The Transcriptomic Landscape of HIV-TB

Thesis Presented for the Degree of
DOCTOR OF PHILOSOPHY
in the Department of Medicine
UNIVERSITY OF CAPE TOWN

Armin Deffur

25th November 2013
The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.
This work is dedicated to Viliana, who has supported me throughout this venture through some very challenging circumstances, and has taught me the value of balance between work- and home life, and to Alexander, who constantly reminds me of the value of play in discovering the world.
Acknowledgements

First and foremost I wish to thank my sponsor and co-supervisor for this PhD degree, Professor Robert J Wilkinson, for his unwavering support. In his words, I am “an infectious disease physician who writes equations for fun” (there are six in this thesis!). This pretty much sums up a core difficulty for me: where do I, with my interests that range from clinical medicine to computer science actually fit in? The answer is clear: in a research group that is open to, and indeed requires, interdisciplinary approaches in problem solving. My current working environment is the embodiment of this: where else could a clinician develop several thousand lines of R code in order to address a complex biomedical problem like HIV-TB? I trust that this work will justify the investment that Professor Wilkinson was willing to make.

No study can succeed without access to willing study participants. I am extremely fortunate to have had access to a treasure trove of prior data as well as ongoing study subject recruitment and data collection in the context to the long-running IMPI-Africa registry, headed by my co-supervisor and Head of Department, Professor Bongani M Mayosi. Under often difficult conditions, a large dataset that is unique in the world has been steadily accruing. This thesis examines individuals studied in the context of IMPI-Africa, but does not concentrate on the cardiac aspects of the condition, but rather takes advantage of sampling the site of tuberculosis disease.

Development of an analytic strategy for heterogeneous data is tricky at best, and a nightmare at worst. In order to figure out how to approach the IMPI-MA data (a total of approximately 8.9 million probe intensity values) I often relied on sage advice of my main thesis supervisor, Professor Nicola Mulder. My initially lofty, if nebulous analysis goals were slowly but surely converted to practical ideas that I could implement in code. Knowing what to leave out is often as important as deciding what to include, and Prof. Mulder was of great assistance in this regard.

Absolutely vital to the success of this project was the collaboration with Dr Anne O’Garra’s laboratory at the National Institute for Medical Research, London, UK. Building on her existing collaboration with Professor Robert J Wilkinson, I was able to access the expertise in the O’Garra laboratory in the following crucial areas: study design, the choice of negative controls, assessment of RNA quality and preparation of RNA samples for microarray analysis, as well as approaches to data interpretation. Many thanks go to Dr Christine Graham in the O’Garra laboratory for single-
handedly processing my extracted RNA samples and preparing the sample library for microarray. Thanks to her expert contribution we ended up with a very high quality dataset. Discussions with Dr O’Garra and Dr Chloe Bloom were very helpful in shaping analytic approaches. Finally, Harsha Jani in the shared microarray facility at the NIMR hybridised and imaged all arrays. Additional thanks to her for re-scanning my samples so that I could get my hands on the original TIFF image data and bead-level data!

Virtually all of science today relies on work that has gone before, and in this context I would like to thank Kerryn Matthews, who took on the immunology of tuberculous pericarditis several years ago together with her supervisor, Professor Katalin A Wilkinson, and generated a huge amount of data whose value is yet to be fully realised. Many RNA samples that went on to microarray were collected and processed by Kerryn. Mpiko Ntsekhe, now Professor and Head of the Division of Cardiology in the Department of Medicine at UCT, together with his colleagues, contributed a very large proportion of the phenotypic data that enabled me to assign samples to classes and which provides a very rich physiological context for the various samples. Without this phenotypic data many questions could not have been asked.

For the prospective sample collection I relied heavily on the IMPI team, both in the CathLab and the IMPI Research Office. Thank you to Veronica Francis, Unita September, and Sisters Hartnick, Williams, Hare and Bing and all the Cardiology Fellows for your efforts in ensuring that I was able to obtain these precious samples!

A fair amount of laboratory time was required to process the prospectively collected samples. Many thanks to Ronnett Seldon for her help in the BSL-3 facility, and Nzwaki Bangani for her help in extracting the RNA.

Many thanks to Rene Goliath and Kathryn Wood for expert logistical and emotional support. Without their input this project would have been a non-starter.

The R project and its sibling, the bioconductor project, as well as the various projects contributing to the \TeX/ \LaTeX codebase provide us with free software. The fact this this software is free is no way a reflection of its value, rather the opposite is true. R is at the cutting edge of data analytic software, and its role will continue to grow in this era of “Big Data”. In this context I would like to thank Renaud Gaujoux and Cathal Seoighe for contributing the CellMix package, which was key in one of three analytic steps applied to the microarray data, and of course all the unnamed developers
who have contributed to the open source code base.

This project could not have been accomplished without adequate funding. I wish to thank all funders for their vital contributions to this project:

1. The Clinical Infectious Disease Research Initiative, Institute for Infectious Diseases and Molecular Medicine (via Wellcome Trust)

2. Southern African Consortium for Research Excellence (via Wellcome Trust)

3. Medical Research Council of South Africa

4. Hasso Plattner Foundation

5. Harry Crossley Foundation

6. Faculty of Health Sciences, University of Cape Town

Last but not least I again thank Viliana for being there for me all the way, supporting my dream as well as myself, and my son Alexander, who, while younger than this research project, continues to learn and innovate on a daily basis at such a pace that I feel humbled.
Abstract

The Transcriptomic Landscape of HIV-TB

Armin Deffur

August 2013

The thesis consists of four main parts. In Part 1 of this thesis I provide a broad overview of HIV-TB with an emphasis on systems approaches followed by an overview of a systems-level study aiming at addressing hypotheses relating to transcriptional differences in active tuberculosis and HIV-1 infection, measured in blood and the site of disease in tuberculous pericarditis. The final chapter in this part describes the methods used to generate and analyse the systems-level data, with emphasis on microarray data generation and analysis.

Part 2 first presents analysis of transcriptional data generated by RT-PCR at the site of disease compared to blood in study subjects with tuberculous pericarditis, with results showing clear evidence of transcriptional differences between compartments. A Technical Results chapter then provides an overview of the microarray data, and an analytic paradigm based on sample embedding in high-dimensional phenotype space is developed. I then assess the overall quality of the dataset and exclude large-scale systematic bias, while comparison of the IMPI-MA data to exiting TB transcriptomic data shows a close match. Part 2 concludes with a description of a comprehensive analytic framework developed for the IMPI-MA data.

Part 3 presents the results of the analytic pipeline as applied to the transcriptional response in blood to active tuberculosis in an HIV-1 uninfected population. A signature of active tuberculosis is described, and deconvolution analysis finds significant NK cell activation in active tuberculosis. Cell-type specific differential expression identifies CD4 T cells, NK cells and neutrophils as the most likely contributors to the overall “signature” of active tuberculosis. Weighted gene co-expression network analysis reveals multiple modules, whose expression is shown to be differentially regulated based on disease category.

Part 4 summarises the results of analysing contrasts in three main contexts: tuberculosis, HIV-1 infection and compartment. Two novel results are presented: Firstly, NK cells are shown to
be functionally downregulated at the site of disease, suggesting a possible defence mechanism by
*M. tuberculosis*, and secondly, large-scale metabolic pathway dysregulation at the site of disease,
possibly favouring *M. tuberculosis*, is demonstrated.

Part 5 concludes the thesis with a summary and outlines future work.
Contents

Terminology and abbreviations ...................................................... 25

I Introduction and methods .......................................................... 27

1 HIV/TB and studies at the site of disease ....................................... 29
  1.1 Introduction/outline .............................................................. 29
  1.2 Clinical features and importance of HIV-TB ............................ 30
  1.3 Clinical phenotypes and diagnosis: HIV-TB is a heterogeneous syndrome .... 32
  1.4 Immune reconstitution after ART ........................................... 34
  1.5 Disease mechanisms in co-infection ....................................... 35
  1.6 Experimental systems ......................................................... 37
  1.7 Systems approaches ........................................................... 41
  1.8 Conclusion ........................................................................... 49

2 IMPI-MA: a two-compartment survey of the transcriptional landscape in HIV-TB 51
  2.1 Tuberculous pericarditis ....................................................... 51
  2.2 IMPI and related substudies .................................................. 53
  2.3 Development of a systems approach ...................................... 56
  2.4 Hypotheses and study questions .......................................... 57
  2.5 Aims and objectives ............................................................ 57
  2.6 Overall study design ............................................................ 58

3 Materials and methods ............................................................. 61
  3.1 Preliminary data: RT-PCR data generation and analysis ............... 61
## Contents

3.2 IMPI-MA study subjects ........................................ 65  
3.3 RNA and other Samples ........................................ 70  
3.4 Clinical phenotype database .................................... 73  
3.5 RNA processing for microarray ................................. 77  
3.6 Microarray workflow ............................................. 78  
3.7 Microarray data .................................................... 79  
3.8 Data analysis overview ......................................... 82  
3.9 Code descriptions ................................................. 96  
3.10 Comparisons ....................................................... 100  

II General Results 101

4 RT-PCR on matched blood and fluid 103  
4.1 General comments regarding the presentation of results 103  
4.2 Study subjects ..................................................... 104  
4.3 Gene expression analysis ....................................... 104  
4.4 Discussion .......................................................... 112  

5 Technical results 115  
5.1 Included patients and samples ................................. 115  
5.2 Sample distribution along the vertices of a multi-dimensional hypercube 117  
5.3 Validation of deconvolution approach: proof of concept data 123  

6 An unbiased view of all array data 127  
6.1 Data ................................................................. 127  
6.2 Quality control plots ............................................ 128  
6.3 Sample relations .................................................. 132  
6.4 Discussion .......................................................... 138  

7 Relation of IMPI-MA data to previous studies of blood transcriptomics in tuberculosis 139  
7.1 Data ................................................................. 139
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Study subjects</td>
<td>140</td>
</tr>
<tr>
<td>7.3</td>
<td>Analysis</td>
<td>140</td>
</tr>
<tr>
<td>7.4</td>
<td>Comparisons</td>
<td>147</td>
</tr>
<tr>
<td>8</td>
<td>An analytic framework for heterogenous microarray data</td>
<td>155</td>
</tr>
<tr>
<td>8.1</td>
<td>The sample hypercube revisited: what kinds of questions relating to</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>the three main hypotheses can we ask, and how can we answer these</td>
<td></td>
</tr>
<tr>
<td></td>
<td>questions?</td>
<td></td>
</tr>
<tr>
<td>8.2</td>
<td>Code description</td>
<td>158</td>
</tr>
<tr>
<td>8.3</td>
<td>Computational output</td>
<td>162</td>
</tr>
<tr>
<td>8.4</td>
<td>Results as Data: a new resource for generating and analysing</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>HIV-TB related hypotheses</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>IMPI-MA: Results in depth</td>
<td>165</td>
</tr>
<tr>
<td>9</td>
<td>Question 1: Tuberculosis</td>
<td>167</td>
</tr>
<tr>
<td>9.1</td>
<td>Differential gene expression</td>
<td>167</td>
</tr>
<tr>
<td>9.2</td>
<td>Array deconvolution and cell-specific differential gene expression</td>
<td>173</td>
</tr>
<tr>
<td>9.3</td>
<td>Weighted gene co-expression network analysis (WGCNA)</td>
<td>178</td>
</tr>
<tr>
<td>IV</td>
<td>IMPI-MA: Results in breadth</td>
<td>187</td>
</tr>
<tr>
<td>10</td>
<td>Contrasts involving tuberculosis</td>
<td>189</td>
</tr>
<tr>
<td>10.1</td>
<td>Question 1: Tuberculosis</td>
<td>189</td>
</tr>
<tr>
<td>10.2</td>
<td>Question 2: PTB and extrapulmonary TB</td>
<td>197</td>
</tr>
<tr>
<td>10.3</td>
<td>Question 3: Latent TB</td>
<td>204</td>
</tr>
<tr>
<td>10.4</td>
<td>Question 4: Haemodynamic phenotypes in TB-PC</td>
<td>211</td>
</tr>
<tr>
<td>10.5</td>
<td>Pathway analysis</td>
<td>217</td>
</tr>
<tr>
<td>11</td>
<td>Contrasts involving HIV infection status</td>
<td>229</td>
</tr>
<tr>
<td>11.1</td>
<td>Question 5: HIV (not active TB)</td>
<td>229</td>
</tr>
<tr>
<td>11.2</td>
<td>Question 6: HIV (active TB)</td>
<td>236</td>
</tr>
</tbody>
</table>
## Contents

11.3 Pathway analysis .................................................... 243

### 12 Contrasts involving the compartment 251

12.1 Question 7: Blood and pericardial fluid in TB-PC ........... 251
12.2 Pathway analysis .................................................... 260

V Overall conclusions ................................................. 271

13 Conclusions and future work ........................................ 273

13.1 Summary of novel results ........................................ 273
13.2 An approach to heterogeneous microarray data ............... 274
13.3 Choice of methods .................................................. 275
13.4 Results .............................................................. 275
13.5 Future work .......................................................... 277

VI Appendix ..................................................................... 295

A Code ........................................................................... 297

A.1 Code for functions used throughout ............................ 297
A.2 System setup ........................................................... 310
A.3 Data manager (general) ............................................. 312
A.4 RT-PCR ................................................................. 317
A.5 Relation of IMPI-MA data to BERRY dataset ............... 332
A.6 Overview of all data ................................................ 355
A.7 Data manager for 7 questions .................................... 361
A.8 Analysis: IMPI-MA .................................................... 366
List of Figures

1.1 Clinical interactions of tuberculosis and HIV-1 ........................................ 33
1.2 Levels of organisation in biological systems ............................................. 42
1.3 A systems approach is an iterative process ............................................. 44

2.1 Relationship between various IMPI cohorts ............................................. 54
2.2 The difference between the concepts Contrast and Comparison ............... 59
2.3 3D parameter space ................................................................................... 60

3.1 RT-PCR: calculation of $\Delta CT$ ................................................................. 62
3.2 Clinical workflow for retrospective TB-PC cases ..................................... 67
3.3 Clinical workflow for retrospective controls ............................................. 69
3.4 Clinical workflow for prospective TB-PC cases ......................................... 69
3.5 Layering of components after Ficoll separation ........................................ 72
3.6 Workflow for data integration ................................................................... 76
3.7 RNA preparation schema .......................................................................... 79
3.8 Workflow for RNA preparation .................................................................. 80
3.9 Workflow for array hybridisation ............................................................... 80
3.10 Approach for unification of cell type subsets .......................................... 92

4.1 Overall expression values ......................................................................... 105
4.2 Expression values by compartment ......................................................... 107
4.3 Calibrated heatmap of all gene expression values ................................... 108
4.4 Correlation matrices for 42 genes ............................................................. 109
4.5 Calibrated heatmap of filtered gene expression values ............................ 109
List of Figures

4.6 Volcano plot of differentially expressed genes .......................... 110
4.7 Calibrated heatmap of differentially expressed genes .................. 111
4.8 Principal component analysis ........................................ 112
5.1 Study subject recruitment ............................................. 116
5.2 RNA quality metrics .................................................. 117
5.3 Sample classes ........................................................ 118
5.4 Three-dimensional sample phenotype space ............................. 119
5.5 Extension of three-dimensional cube to six dimensions ............... 120
5.6 Six dimensional hypercube ............................................. 121
5.7 Three-dimensional phenotype space for LTBI ......................... 121
5.8 Three-dimensional phenotype space for TB-site ....................... 122
5.9 Three-dimensional phenotype space for HD phenotype ............... 122
5.10 Barplots of proportions in 11 samples ................................ 124
5.11 Correlations of cell proportions ...................................... 125
6.1 PDF plots of all IMPI-MA arrays ...................................... 129
6.2 CDF plots of all IMPI-MA arrays ...................................... 130
6.3 Pairwise sample correlation .......................................... 131
6.4 MA plots for five arrays .............................................. 132
6.5 Box-and-whisker plot of all IMPI-MA arrays ......................... 133
6.6 MDS: All IMPI-MA data .............................................. 135
6.7 MDS: IMPI-MA blood samples (HIV-TB) ............................. 135
6.8 MDS: IMPI-MA blood samples (Sex/steroids) ......................... 136
6.9 MDS: IMPI-MA pericardial fluid samples (HIV-TB) ................. 137
7.1 Box-and-whisker plots of the raw expression data for the three datasets .... 142
7.2 Multidimensional scaling analysis .................................... 143
7.3 Heatmaps of selected probes ........................................ 145
7.4 Multidimensional scaling analysis of selected probes ............... 146
7.5 Heatmaps of differentially expressed probes .......................... 148
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.6</td>
<td>Multidimensional scaling analysis of differentially expressed probes</td>
<td>149</td>
</tr>
<tr>
<td>7.7</td>
<td>Method validation</td>
<td>150</td>
</tr>
<tr>
<td>7.8</td>
<td>Data similarity</td>
<td>151</td>
</tr>
<tr>
<td>7.9</td>
<td>Probability of probe overlap</td>
<td>151</td>
</tr>
<tr>
<td>7.10</td>
<td>Overlap of all three datasets with the 393 probe list</td>
<td>151</td>
</tr>
<tr>
<td>7.11</td>
<td>Heatmaps of 61 overlapping probes applied to three datasets</td>
<td>153</td>
</tr>
<tr>
<td>8.1</td>
<td>Sample hypercube</td>
<td>157</td>
</tr>
<tr>
<td>8.2</td>
<td>Overall code structure for IMPI-MA main script</td>
<td>159</td>
</tr>
<tr>
<td>9.1</td>
<td>Data QC: not normalised (left) and normalised (right)</td>
<td>168</td>
</tr>
<tr>
<td>9.2</td>
<td>Volcano plot of differentially expressed probes</td>
<td>170</td>
</tr>
<tr>
<td>9.3</td>
<td>Heatmaps of significant and top 500 probes</td>
<td>171</td>
</tr>
<tr>
<td>9.4</td>
<td>PCA of differentially expressed probes</td>
<td>172</td>
</tr>
<tr>
<td>9.5</td>
<td>Overlap of differentially expressed probes for two different methods</td>
<td>172</td>
</tr>
<tr>
<td>9.6</td>
<td>Barplots of detected and PBMC types</td>
<td>174</td>
</tr>
<tr>
<td>9.7</td>
<td>Boxplots of detected and PBMC types</td>
<td>176</td>
</tr>
<tr>
<td>9.8</td>
<td>False discovery rate plots after cell-specific DE</td>
<td>178</td>
</tr>
<tr>
<td>9.9</td>
<td>Cell-type specific differential expression</td>
<td>179</td>
</tr>
<tr>
<td>9.10</td>
<td>WGCNA: Traits and modules</td>
<td>180</td>
</tr>
<tr>
<td>9.11</td>
<td>Module/ trait relationships</td>
<td>182</td>
</tr>
<tr>
<td>9.12</td>
<td>Top 3 GS/MM plots</td>
<td>183</td>
</tr>
<tr>
<td>9.13</td>
<td>Three significant modules</td>
<td>184</td>
</tr>
<tr>
<td>9.14</td>
<td>Weighted gene co-expression networks</td>
<td>185</td>
</tr>
<tr>
<td>10.1</td>
<td>Results for contrast TB status in contexts HIV negative and HIV positive</td>
<td>194</td>
</tr>
<tr>
<td>10.2</td>
<td>Cell-type specific differential expression</td>
<td>195</td>
</tr>
<tr>
<td>10.3</td>
<td>Module clustering</td>
<td>196</td>
</tr>
<tr>
<td>10.4</td>
<td>Overlap of differentially expressed probes</td>
<td>196</td>
</tr>
<tr>
<td>10.5</td>
<td>Results for contrast TB site in contexts HIV negative and HIV positive</td>
<td>201</td>
</tr>
<tr>
<td>10.6</td>
<td>Cell-type specific differential expression</td>
<td>202</td>
</tr>
</tbody>
</table>
## List of Figures

10.7 Module clustering ................................................. 202
10.8 Overlap of differentially expressed probes .................. 203
10.9 Results for contrast *LTBI* in contexts *HIV negative* and *HIV positive* 207
10.10 Cell-type specific differential expression .................. 209
10.11 Module clustering ............................................. 210
10.12 Overlap of differentially expressed probes ................ 210
10.13 Results for contrast *Haemodynamic phenotype* in contexts *Blood* and *Fluid* 214
10.14 Cell-type specific differential expression .................. 215
10.15 Module clustering ............................................. 216
10.16 Overlap of differentially expressed probes ................ 216
10.17 Overlap of differentially regulated pathways ............... 221
10.18 Significant pathways in tuberculosis (HIV-1 uninfected) .... 222
10.19 Significant pathways in tuberculosis (HIV-1 infected) ...... 223
10.20 Toll-like receptor pathway ................................. 225
10.21 Antigen processing and presentation pathway .............. 226
10.22 NK cell cytotoxicity pathway (HIV-1 uninfected) ........... 227

11.1 Results for contrast *HIV status* in contexts *Healthy* and *LTBI* ........ 233
11.2 Cell-type specific differential expression .................. 234
11.3 Module clustering ............................................. 235
11.4 Overlap of differentially expressed probes ................ 235
11.5 Results for contrast *HIV status* in contexts *TB-PC Blood* and *TB-PC Fluid* 240
11.6 Cell-type specific differential expression .................. 241
11.7 Module clustering ............................................. 242
11.8 Overlap of differentially expressed probes ................ 242
11.9 Significant pathways in HIV ................................ 246
11.10 Overlap of differentially regulated pathways ............... 247
11.11 Antigen processing and presentation pathway .............. 249
11.12 RIG-I-like receptor signaling pathway ....................... 250
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1</td>
<td>Results for contrast <em>Compartment</em> in contexts TB-PC HIV neg and TB-PC HIV</td>
<td>256</td>
</tr>
<tr>
<td>12.2</td>
<td>Barplots and boxplots of matched blood and pericardial fluid samples</td>
<td>257</td>
</tr>
<tr>
<td>12.3</td>
<td>Cell-type specific differential expression</td>
<td>258</td>
</tr>
<tr>
<td>12.4</td>
<td>Module clustering</td>
<td>259</td>
</tr>
<tr>
<td>12.5</td>
<td>Overlap of differentially expressed probes</td>
<td>259</td>
</tr>
<tr>
<td>12.6</td>
<td>Overlap of differentially regulated pathways</td>
<td>266</td>
</tr>
<tr>
<td>12.7</td>
<td>Antigen processing and presentation pathway</td>
<td>268</td>
</tr>
<tr>
<td>12.8</td>
<td>Natural killer cell mediated cytotoxicity</td>
<td>269</td>
</tr>
</tbody>
</table>
**List of Tables**

3.1 42 genes selected for RT-PCR analysis ................................. 63
3.2 Definitions of terms .................................................. 66
3.3 Sources of clinical phenotype data used in IMPI-MA .................. 74
3.4 Levels of data complexity ............................................. 81
3.5 Methods for selecting differentially expressed genes ................. 85
4.1 Clinical characteristics of study participants ......................... 104
4.2 Gene-by-gene comparison of expression between compartments .... 106
4.3 Differentially expressed genes ......................................... 110
5.1 Reasons for exclusion of RNA samples ............................... 116
7.1 Table of clinical characteristics across datasets and contrasts .... 141
7.2 Effect of non-specific filtering ........................................ 141
7.3 Results of differential expression analysis ........................... 147
8.1 Questions ...................................................................... 156
9.1 Top 30 probes as selected by *limma* .................................. 169
9.2 Significance of differences in cell proportions in active TB vs not active TB .......................... 175
9.3 Significance of modules for contrast “active TB vs no TB” .......... 181
10.1 Clinical characteristics: active TB vs not active TB ................ 191
10.2 Overall results: active TB vs not active TB ........................ 192
10.3 Clinical characteristics: PTB vs TB-PC ............................... 198
10.4 Overall results: PTB vs TB-PC .......................................... 199
List of Tables

10.5 Clinical characteristics: Healthy vs LTBI ........................................... 205
10.6 Overall results: Healthy vs LTBI .......................................................... 206
10.7 Clinical characteristics: Effusive vs Effusive-constrictive pericarditis ...... 211
10.8 Overall results: Effusive vs Effusive-constrictive pericarditis ............... 212
10.9 Top pathways in TB (HIV-1 uninfected) ............................................. 218
10.10 Upregulated pathways in TB (HIV-1 infected) .................................... 219
10.11 Downregulated pathways in TB (HIV-1 infected) .............................. 220

11.1 Clinical characteristics: HIV positive vs HIV negative (not active TB) .... 230
11.2 Overall results: HIV positive vs HIV negative (not active TB) ............... 231
11.3 Clinical characteristics: HIV positive vs HIV negative (active TB) ......... 237
11.4 Overall results: HIV positive vs HIV negative (active TB) .................. 238
11.5 Upregulated pathways in HIV (not active TB) ...................................... 243
11.6 Downregulated pathways in HIV (not active TB) .................................. 244
11.7 Upregulated pathways in HIV (active TB) ........................................... 245
11.8 Downregulated pathways in HIV (active TB) ....................................... 245

12.1 Clinical characteristics: blood vs pericardial fluid .............................. 252
12.2 Overall results: blood vs pericardial fluid .......................................... 254
12.3 Pathways significantly upregulated in pericardial fluid (HIV-1 uninfected) . 261
12.4 Pathways significantly downregulated in pericardial fluid (HIV-1 uninfected) 262
12.5 Pathways significantly upregulated in pericardial fluid (HIV-1 infected) .... 263
12.6 Pathways significantly downregulated in pericardial fluid (HIV-1 infected) .. 264
## Listings

| A.1 | heatmapad.R | 297 |
| A.2 | superHeatmap.R | 302 |
| A.3 | superHeatmap2.R | 304 |
| A.4 | plotSampleRelationsAD.R | 306 |
| A.5 | plotSampleRelations3DAD.R | 308 |
| A.6 | SystemSetup.R | 310 |
| A.7 | Data-manager.R | 312 |
| A.8 | RT-PCR code | 317 |
| A.9 | RelationIMPI-MA to TB-AOG | 332 |
| A.10 | Overview of all data | 355 |
| A.11 | Data-manager-Q.R | 361 |
| A.12 | SevenQuestions | 366 |
## Terminology and abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFB</td>
<td>Smear for acid-fast bacilli (either Ziehl Neelsen or immunofluorescence)</td>
</tr>
<tr>
<td>BSL-3</td>
<td>Biosafety Level 3</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative density function</td>
</tr>
<tr>
<td>CFF</td>
<td>Cell-free pericardial fluid</td>
</tr>
<tr>
<td>FDR</td>
<td>False discovery rate</td>
</tr>
<tr>
<td>HIV+</td>
<td>HIV-1 infected (HIV positive)</td>
</tr>
<tr>
<td>HIV-</td>
<td>HIV-1 uninfected (HIV negative)</td>
</tr>
<tr>
<td>HIV-TB</td>
<td>Co-infection with HIV-1 and <em>Mycobacterium tuberculosis</em></td>
</tr>
<tr>
<td>IMPI-MA</td>
<td>Investigation into the management of tuberculous pericarditis: microarray sub-study</td>
</tr>
<tr>
<td>KEGG</td>
<td>Kyoto Encyclopedia of Genes and Genomes</td>
</tr>
<tr>
<td>LTBI</td>
<td>Latent tuberculosis infection</td>
</tr>
<tr>
<td>NIMR</td>
<td>National Institute for Medical Research</td>
</tr>
<tr>
<td>PBMCs</td>
<td>Peripheral blood mononuclear cells</td>
</tr>
<tr>
<td>PBS</td>
<td>Phosphate-buffered saline solution</td>
</tr>
<tr>
<td>PCF</td>
<td>Pericardial fluid</td>
</tr>
</tbody>
</table>
### Terminology and abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF</td>
<td>Probability density function</td>
</tr>
<tr>
<td>PFC</td>
<td>Pericardial fluid cells</td>
</tr>
<tr>
<td>PTB</td>
<td>Pulmonary tuberculosis</td>
</tr>
<tr>
<td>RPMI+10%FCS</td>
<td>Roswell Park Memorial Institute medium (version 1640) containing 10% heat-inactivated fetal calf serum</td>
</tr>
<tr>
<td>RPMI-1640</td>
<td>Roswell Park Memorial Institute medium (version 1640)</td>
</tr>
<tr>
<td>TB</td>
<td>Tuberculosis</td>
</tr>
<tr>
<td>TB-PC</td>
<td>Tuberculous pericarditis</td>
</tr>
<tr>
<td>TOM</td>
<td>Topological overlap matrix</td>
</tr>
<tr>
<td>WGCNA</td>
<td>Weighed gene network co-expression analysis</td>
</tr>
</tbody>
</table>
Part I

Introduction and methods
1 HIV/TB and studies at the site of disease

Chapter summary

In this chapter I describe in detail the clinical, cellular and molecular characteristics of co-infection with *M. tuberculosis* and HIV-1 (HIV-TB). I go on to describe systems approaches in general and applied to tuberculosis, and finally argue for system-level studies in HIV-TB.

The text of this Chapter is taken verbatim from a review paper submitted to Pathogens and Disease on 17 April 2013 and accepted for publication on 20 June 2013 [1]. Authors of this paper are Armin Deffur (first author), Nicola Mulder and Robert J Wilkinson (senior author); the co-authors provided input in line with their roles as supervisors. The rationale for including this publication in the thesis is that it provides a comprehensive literature review of the problem of HIV-1 and *Mycobacterium tuberculosis* co-infection. Please note that British English spelling rules have been applied to the text where relevant.

1.1 Introduction/outline

Co-infection by HIV-1 and *Mycobacterium tuberculosis* is of great concern worldwide. In this review I present a concise overview of clinical and immunological features of HIV-TB, review experimental systems in which the biology of this co-infection is studied and finally discuss how a systems approach may be employed in advancing understanding of these conditions.
1.2 Clinical features and importance of HIV-TB

About a third of the human population are estimated to be infected with *M. tuberculosis*. In the majority of cases the infection is clinically inapparent, with only approximately 10% of individuals at risk of developing overt clinical tuberculosis in their lifetime.

This indicates a fine balance between effective host defences (resulting in bacterial containment or clearance) and substantive infection that results in significant pathology, potentially killing the host early in the course of infection. Either scenario is clearly not beneficial for the reproductive success of *M. tuberculosis*, and a delicate balance has evolved that maintains the host-pathogen relationship over long time periods. Even in the case of a shift towards active infection with concurrent pulmonary pathology, the resulting disease (pulmonary tuberculosis) can follow a chronic course with 37% 5 year survival if left untreated [2]. The recent arrival of HIV-1 as a human pathogen has shifted the balance towards a dramatic increase in clinical disease and increased mortality associated with tuberculosis in settings where HIV-1 is highly prevalent. This has resulted in a deadly syndemic [3], where the consequences of co-infection manifest variability in clinical and pathologic features and a shift towards worse outcome. The pathobiology of this clinically important co-infection is incompletely understood. While multiple risk factors have substantial impact on the worldwide tuberculosis epidemic, HIV-1 infection is an important driver of tuberculosis. 13% of worldwide incident cases of tuberculosis are HIV-1 co-infected, and HIV-1 remains the most important risk factor in many high-burden settings [3]. HIV-TB accounts for a disproportionate share of tuberculosis-related mortality. It is therefore instructive to review in outline what is known about the interaction of HIV-1, *M. tuberculosis* and the human immune system at the individual level. Population-level interactions will not be reviewed here.

1.2.1 Effect of HIV-1 on tuberculosis

HIV-1 co-infection increases risk for TB susceptibility to primary or re-infection. While in general HIV-TB patients are less frequently sputum positive, their greater numbers in high-burden settings for both tuberculosis and HIV-1 infection, suggests they may contribute substantially to transmission. A study in Harare found that HIV-1 infected individuals had high levels of incident TB, while HIV-1 uninfected individuals had higher levels of prevalent smear-positive tuberculosis, suggesting
that this subpopulation still drives transmission of \textit{M. tuberculosis}, while HIV-1-infected individuals account for the majority of incident cases [4]. Of all opportunistic infections with associated with HIV-1, the risk for developing clinical tuberculosis is increased shortly after HIV-1 infection has taken place, well before the total CD4 count drops to levels below 500 cells per \( \mu l \) [5]. This raises the question whether HIV-1 causes an early tuberculosis-specific defect in host immunity.

HIV-1 also increases the risk of progressing from one stage of tuberculosis to the next [6], including increased risk of primary disease following exposure [7], reactivation of latent (clinically quiescent) lesions [8, 2] and reinfection with \textit{M. tuberculosis} following exposure [9, 10, 3].

Tuberculosis disease phenotypes are also altered by HIV-1 co-infection. While cavitatory pulmonary tuberculosis is the most common clinical manifestation in HIV-1-uninfected individuals, atypical presentations of pulmonary disease are much more prevalent in HIV-1 infected individuals [11, 12, 3]. This tendency to atypical pulmonary disease is linked to CD4 count, inasmuch as individuals with preserved CD4 counts present with upper lobe cavitation much like their HIV-1 uninfected counterparts, whereas low CD4 counts are linked to atypical infiltrates (middle and lower lobe infiltrates, interstitial nodules), mediastinal lymph node involvement and even normal chest radiographs [13]. While chest radiographs are unhelpful in detecting active tuberculosis in HIV-1 infected individuals in many instances, so is the addition of Interferon-gamma release assays to screening strategies aimed at identifying smear-negative tuberculosis [14]. In addition to pulmonary disease, which remains the most important presentation in HIV-1 co-infected individuals, extrapulmonary forms of tuberculosis are more common in this group compared to HIV-1 uninfected individuals [15, 5]. While extrapulmonary tuberculosis usually presents with a single site in the absence of HIV-1 infection, HIV-1-co-infected individuals often present with concurrent pulmonary and extrapulmonary disease.

In a prospective study of extrapulmonary tuberculosis, which consisted of HIV-1 infected and uninfected individuals, the most common extrapulmonary sites were lymph nodes (43%), pleura (23%), central nervous system (8%), musculo-skeletal system (7%), genitourinary tract (5%), and gastrointestinal tract (2%) [6, 16] with other sites making up the remainder (Data was not stratified by HIV-status in original study). Finally, HIV-1-co-infection increases tuberculosis-associated mortality[7, 17].
1.3 Clinical phenotypes and diagnosis: HIV-TB is a heterogeneous syndrome

Tuberculosis infection in humans is classically divided into two distinct phenotypes[18]. In this paradigm, latent tuberculosis infection results from primary infection of the lung that is contained but not eradicated by a host adaptive immune response. This response is characterised by granuloma formation, intracellular (mainly within macrophages) location of mycobacteria, which do not replicate and may be metabolically adapted to persistence, and the absence of any symptoms suggestive of tuberculosis. The other phenotype is active pulmonary tuberculosis, (usually ascribed to reactivation of a previously dormant lesion). This century-old description is being increasingly questioned, and current thinking favours the concept of a spectrum of disease phenotypes [19], ranging from cleared infection without or with immune sensitisation, inactive lesions containing mycobacteria, active lesions with bacterial replication but without symptoms to full-blown active disease (See Figure 1.1).

HIV-1 co-infection complicates this picture by skewing clinical and pathological phenotypes towards more advanced disease, significantly contributing to the heterogeneity of clinical presentations of tuberculosis in the context of HIV-1 co-infection. In order to dissect this complexity, more detailed phenotypes of human disease may be generated by using novel functional imaging techniques [20, 21, 19].

Primary disease following recent tuberculosis exposure is also more common in HIV-1 co-infection, as is an increased risk for reactivation of previously quiescent lesions as well as re-infection of persons already infected with *Mycobacterium tuberculosis*. The question of whether prior or current tuberculosis elicits an immune response that can protect from infection is controversial. While a 2005 review [22] found that evidence for protective immunity (against re-infection) induced by latent tuberculosis was not convincing, a recent systematic review [23] showed that individuals with latent tuberculosis had a 79% lower risk of progressive tuberculosis after re-infection than unsensitised individuals.
1.3 Clinical phenotypes and diagnosis: HIV-TB is a heterogeneous syndrome

Figure 1.1: Clinical interactions of tuberculosis and HIV-1. HIV-1 and tuberculosis interact with each other. The x-axis represents stages of tuberculosis, from infection through to active disease, while the y-axis represents stages of HIV-1 infection. *M. tuberculosis* bacterial burden and CD4 count are shown in blue along the respective axes. The spectrum of “latent” TB is represented as follows: (1) Infection eliminated without priming antigen-specific T-cells. (2) Infection eliminated in association with T-cell priming. (3) Infection contained with some bacteria persisting in a non-replicating form. (4) Bacterial replication maintained at subclinical level by the immune system. Clinical disease (pulmonary and extrapulmonary tuberculosis) occurs in a subset of individuals who are latently infected or who develop primary tuberculosis directly following infection or re-infection. The annual risk is represented as follows: (A) HIV-1 uninfected: approximately 10% lifetime risk, or about 1% per annum (p.a.). (B) Shortly after HIV-1 infection, and prior to substantial CD4 T-cell depletion, the risk for active tuberculosis increases. (C) During the early stages of HIV-1 infection, this risk rises to approximately 10% p.a. (D) In late-stage HIV-1 infection, the risk for active tuberculosis increases to 30% per annum. The effects of HIV-1 on tuberculosis, and of tuberculosis on HIV-1 disease are shown by the red arrows.
1 HIV/TB and studies at the site of disease

1.4 Immune reconstitution after ART

Following initiation of highly active antiretroviral therapy (ART) in patients with HIV-1 infection, tuberculosis incidence is reduced as protection from tuberculosis is restored with recovery of CD4 count. This reduction in incidence is only partial, however, as tuberculosis incidence still remains higher after ART and CD4 restoration compared to HIV-1 uninfected individuals [24]. As reconstitution of a protective immune response against many antigens occurs, individuals may also develop worsening symptoms and signs of opportunistic infections due to a vigorous response against antigens present in the host resulting from immune activation that may be modulated by corticosteroids [25]. This is termed tuberculosis immune reconstitution inflammatory syndrome (TB-IRIS). TB-IRIS appears to be driven by increased recognition of \textit{M. tuberculosis} antigens present shortly after ART initiation; this may be associated with conversion of a negative TST to a strongly positive one.

Pathogenic mechanisms that have been considered in TB-IRIS include purified protein derivative (PPD)-specific expansion of Th1 responses [26] and defective restoration of regulatory T cell function. However, Th1-specific expansions may also occur in individuals who do not develop symptoms and signs of TB-IRIS [27, 25], and numbers of CD4+ Foxp3+ T cells in TB-IRIS cases are similar in controls[28]. More recent work has focused on the interaction of innate and adaptive responses, with evidence of myeloid activation [29, 30], cytolytic action of NK cells [31] as well as killer immunoglobulin receptor (KIR)-negative \(\gamma\delta\)-T cells [32] all contributing to TB-IRIS.

In contrast to the pathologic responses associated with TB-IRIS, functional, non-pathologic immune restoration is responsible for the enhanced protection from active tuberculosis seen following start of ART. This response is associated with an increase in absolute CD4 T cell count, decline in HIV-1 viral load, an absolute increase in effector function but a proportional decrease in the ratio of effector to central memory CD4 T cells due to an even greater expansion of the central memory pool, which provides the strongest correlate of ART-mediated immune restoration [33]. Another study found that the extent of immune restoration following ART initiation depended on the baseline (nadir) CD4 T cell count. Only at nadir CD4 counts greater than 350 cells/\(\mu\)l did the immune restoration result in CD4 T cell populations similar to those in HIV-1 uninfected individuals, suggesting an immunological benefit of starting ART at that threshold [34].
1.5 Disease mechanisms in co-infection

1.5.1 Immune mechanisms in tuberculosis

Immune control of *M. tuberculosis*, an intracellular pathogen relies on both innate and adaptive cell-mediated immunity [35]. Although a vigorous humoral response to multiple mycobacterial antigens is produced, the antibodies appear to be largely ineffective in controlling the disease due to the mainly intracellular localisation of the bacterium; there is some debate whether opsonising antibody may be helpful early in infection. The response therefore requires CD4 and CD8 T cells that respond to *M. tuberculosis* antigens presented by either MHC class I, MHC class II or CD1 molecules on the surface of antigen presenting cells (macrophages and dendritic cells), a Th1 cytokine milieu (IFN-gamma, IL-12, TNF) and appropriate regulatory responses that limit the extent of the inflammatory response. Even in active tuberculosis, there is a delicate balance between pro- and anti-inflammatory responses, that aim to contain the infection while limiting collateral damage. This results in the often very protracted course of pulmonary tuberculosis, recognised as the syndrome of “consumption” prior to the era of antimycobacterial chemotherapy. Co-infection with HIV-1 has a dramatic impact on this balance.

1.5.2 Effect of HIV-1 on tuberculosis immunity

The most obvious effect on tuberculosis immunity by HIV-1 is the severe depletion of the CD4 T cell population, both in blood and in peripheral lymphoid tissues [36, 37]. In addition, HIV-1 induces CD4 T cell dysfunction as evidenced by reduced IL-2 or IFN-gamma production [38, 39, 40]. Early selective depletion of *M. tuberculosis* antigen-restricted CD4 cells was demonstrated in 2008, and likely explains the early increase in risk of tuberculosis following HIV-1 infection [41, 42]. It is hypothesised that HIV-1 may indirectly cause loss of *M. tuberculosis*-specific CD8 T-cells through loss of *M. tuberculosis*-specific CD4 T cells, though this has not yet been experimentally demonstrated in humans. CD8 T cells probably play an important role in containing tuberculosis infection. This view is based on data from in vitro work [43], mouse models [44], non-human primate models [45], and cells isolated from individuals with tuberculosis following anti-TNF therapy [46].
1.5.3 Effect of tuberculosis on HIV-1: viral replication

Infection with *Mycobacterium tuberculosis*, and more directly the immune response directed against it, result in enhanced intracellular HIV-1 replication in blood [47] and at sites of tuberculosis disease, including the lung [48] and pericardial space in the case of tuberculous pericarditis [49]. Multiple mechanisms have been implicated, including cytokine- and chemokine-mediated mechanisms [50, 51], loss of an inhibitory transcription factor C/EBP beta [52], activation of (NF)-κB and signalling involving positive transcription elongation factor (P-TEFβ) [53]. These mechanisms cause activation of transcription of HIV-1, resulting in enhanced viral replication, spread to uninfected cells and their subsequent depletion.

1.5.4 Effect of tuberculosis on HIV-1: clinical disease progression

Active tuberculosis is associated with more rapid progression to severe CD4 T cell depletion and higher risk for opportunistic infections resulting in late stage HIV-1 infection and the development of AIDS [54, 55].

1.5.5 The role of host genetic variation

There is moderate evidence of human genetic susceptibility to tuberculosis [56]. For example, using a genome-wide screening approach in *M. tuberculosis*-infected vs. non-infected dendritic cells, Barreiro et al found a number of expression quantitative trait loci (eQTL) that associate with tuberculosis in this system [57], and imputation of 1000 Genomes Project data [58] into a Ghanaian genome-wide dataset (tuberculosis cases and healthy controls) revealed a resistance locus at 11p13 downstream of WT1 (Wilms tumor 1) [59]. Thus, host genetics introduces an additional factor to consider when discussing the interaction between HIV-1 and tuberculosis. Recently, susceptibility to tuberculosis in HIV-1 infected individuals was associated with a CARD8 genetic variant [60], and a genotype resulting in high IL-10 production also appeared to predispose HIV-1 infected individuals to TB infection [61]. It should be noted however, that studies that infer genetic risk for tuberculosis are often inconsistent in their results, and study design is probably responsible for this [62]. The definition of tuberculosis phenotype often varies across such studies, making them hard to compare directly and to generalise their results. This caveat applies equally to studies of
tuberculosis susceptibility in HIV-1 infected individuals.

1.6 Experimental systems

1.6.1 Isolated model systems

Experimental systems consisting of single cell types (e.g. macrophages) manipulated in vitro are useful for delineating actions of individual mediators (cytokines, chemokines), intracellular processes (signalling cascades, autophagy) and processes that are ultimately reductionist. While yielding invaluable data on the “parts list” of the immune response to tuberculosis, such systems by definition lose higher-order information about the overall orchestration and coordination of the immune response.

1.6.2 Animal models

Animal models of tuberculosis can be very useful in understanding higher-level immune phenomena that are lost in simple systems, but care must be taken when extrapolating these phenomena to human immunobiology without experimental evidence.

Early major contributions of the mouse model to understanding human tuberculosis are insights into the key importance of IFN-γ, TNF-α and CD4 T-cells in containing tuberculosis. The mouse model is utilised to study the effect of \(M. tuberculosis\) genomic variation (including induced mutants), immune response, the effect of drugs and vaccines [63].

An argument against the use of mouse models is that they cannot effectively recapitulate HIV-1 co-infection in tuberculosis. Considerable effort has gone to develop mouse models of HIV-1 infection [64, 65]. Basic models aim to mimic HIV-1 infection by CD4 T-cell depletion. While CD4 depletion is clearly important in HIV-1 pathogenesis, HIV-1 has additional effects on the immune system that are not captured in this model (apoptosis, disruption of lymph node architecture, depletion of \(M. tuberculosis\)-specific T cells and effects on macrophage function). Therefore, more realistic models have been developed, including Nef-transgenic mice [66, 67, 68] and humanised bone marrow-liver-thymus (BLT) mice [69, 70, 71] that support the entire life cycle of HIV-1 and thus may become valuable tools in understanding aspects of tuberculosis and HIV-1 co-infection.
In contrast to their indisputable utility in discovering disease mechanisms in tuberculosis [72], use of mouse models at the system level (by the use of unbiased genomic screens) may also lead to incorrect conclusions if great care is not taken when mapping mouse data onto human orthologs. In a recent study, transcriptomic responses to three broad inflammatory stimuli were studied in humans and mice using a whole-blood transcriptomic approach [73]. In humans, the transcriptomic responses were correlated with each other, but the corresponding mouse models of the inflammatory conditions (burns, trauma, endotoxinaemia) were both uncorrelated with their human counterparts, and with each other. This strongly suggests that overall control in the mouse immune system may be wired differently than in humans, and that large-scale organisational features are not conserved across this particular species barrier. This finding argues strongly to restrict systems-level investigations into tuberculosis and HIV-1 co-infection to models other than mice, such as human (clinical) and non-human primate models.

In contrast to mouse models, other animal models recapitulate more features of human tuberculosis, e.g. guinea pigs and rabbits both produce caseous granulomas, a hallmark of human disease, but are limited by the availability of immunological reagents required to interrogate the models. The non-human primate model is the closest approximation of human tuberculosis [74, 75]. Longitudinal samples including blood, broncho-alveolar lavage fluid and lymph node biopsy tissue are available during the infection stage of the model, and all tissues are available at necropsy. As non-human primates can be infected with simian retroviruses that cause an immunodeficiency syndrome in these animals, use of this model may extend to study the non-human primate equivalent of HIV-TB [65]. Given these features, this model system is able to generate data that is essentially impossible to generate from other animal models or human clinical disease models.

Despite their utility, all animal models are limited when performing system-level studies that aim to understand precise mechanisms of tuberculosis and HIV-1 co-infection in humans. Given these limitations it is imperative that an integrated, systems-level model of tuberculosis and HIV-1 co-infection be developed for humans in an unbiased way using systems approaches.
1.6.3 Difficulty of \textit{in vivo} work in humans

\textit{In vivo} studies of human tuberculosis are difficult, as sampling of disease sites cannot be carried out in the same way as in animal models. Post-mortem studies of human tuberculosis are available, but relatively limited, and the ability to perform experimental challenges are very limited.

The bulk of our existing knowledge comes from technologies that may suffer from low spatial resolution (e.g. chest radiographs may be normal in subclinical pulmonary disease) that may fail to resolve phenotypes along the latency spectrum, and obscure the inherent heterogeneity of human tuberculosis.

Our understanding of this low-resolution data has led to classical descriptions of tuberculosis pathogenesis involving distinct, polarised phenotypes: Infection with \textit{Mycobacterium tuberculosis} leading to either latent tuberculosis or primary progressive disease, and reactivation (usually in the lung apices) of latent infection which leads to full-blown clinical tuberculosis.

Compounding the problem is the fact that the early stages of tuberculosis are hard to study. It is likely that very early events following exposure to aerosols containing \textit{M. tuberculosis} are key in determining the outcome of the exposure. Hypothetical outcomes are threefold:

1. Clearance of inhaled \textit{M. tuberculosis} by innate immune responses (possibly mediated by neutrophils alone or in combination with other innate cells) without presentation of cognate antigen to adaptive immune cells, therefore not resulting in any acquired response.

2. Failure of the innate immune system to clear the infection, resulting in cells of the adaptive immune system becoming actively involved, with antigen presentation on dendritic cells and macrophages to T-lymphocytes and subsequent downstream events. At this stage the infection may be cleared or contained within granulomas. IGRA and/ or TST responses are used as indicators of this immune sensitisation to \textit{M. tuberculosis}, but cannot differentiate between cleared infection with a lasting memory response, a contained infection that may yet reactivate, and subclinical disease.

3. The infection may overwhelm innate and adaptive responses and result in primary progressive disease.
1 HIV/TB and studies at the site of disease

1.6.4 Limitations of reductionist approaches

Current reductionist approaches have defined a multitude of genes, proteins, cellular mechanisms and cell populations that are important in containing mycobacterial infection, account for immune deficits caused by HIV-1 infection and describe the response to vaccines. However, these molecular and cellular-level mechanisms interact to produce high-level phenomena, such as protective immune responses. At this stage, it is unknown what truly constitutes an immune response that protects from infection with \( M. \) \textit{tuberculosis} or from progressive disease following infection. The simple single cell models and even animal models do not recapitulate human HIV-TB co-infection completely. Simple models interrogate only a few dimensions of a very high-dimensional parameter space that describes the complex system fully. As a result, a large portion of this parameter space remains unexplored. HIV-1 co-infection adds to this complexity by introducing one or more orthogonal dimensions to the parameter space. An alternative is a systems approach to HIV-TB co-infection that yields models that are able to predict emergent phenomena.

1.6.5 Evolutionary perspective

The association of \textit{Mycobacterium tuberculosis} with human hosts is not a recent phenomenon, but rather a long-standing one, with co-evolution of host and pathogen [76]. \textit{M. tuberculosis} is found wherever human populations are found, and at least seven distinct lineages have been identified [77]. In addition, humans have evolved while exposed to multiple retroviruses, which remain as genomic remnants in the form of HERVs (human endogenous retroviruses) [78]. Also, HIV-1 infection disables key components of the adaptive immune response to tuberculosis (some specific for \textit{M. tuberculosis} responses) while leaving innate mechanisms more intact. HIV-1 and \textit{M. tuberculosis} may affect each other’s evolution. An increase in HIV-1 heterogeneity may be associated with a TB-mediated increase in HIV-1 replication discussed above, suggesting the possibility that HIV-1 and \textit{M. tuberculosis} co-infection may alter the evolutionary dynamics of HIV-1[79]. Conversely, an increase in multidrug-resistance conferring mutations in \textit{M. tuberculosis} has sometimes been noted in HIV-1 co-infected individuals, suggesting that HIV-1 may directly or indirectly influence \textit{M. tuberculosis} diversity [79].

Worldwide distribution of different lineages of \textit{M. tuberculosis} mirrors that of geographically
defined human population groups, which suggest that the specific strains are adapted to their (geographically) local population. This sympatric relationship is disrupted following HIV-1 infection, in which a much higher likelihood of infection with allopatric strains was seen compared to HIV-1 uninfected individuals [80].

### 1.7 Systems approaches

#### 1.7.1 Introduction to systems

A “system” can be loosely defined as a collection of entities that collectively exhibit particular behaviour. A hallmark of complex systems is that this behaviour is hard to predict when given information about the constituent parts. Systems are organised in levels of organisational complexity. A defining feature of systems is the concept of interaction: any particular level of organisation may be viewed as a set that has members consisting of specific biological entities that interact with each other. Usually, the set of interactions between the various members can be described using network terminology. An intuitive example is the collection of signalling pathways within a specific cell type that consists of interactions between ligands, receptors, downstream linker molecules and ultimately effector molecules (e.g. transcription factors) that alter gene expression. The set of interactions between entities of one level of organisation then give rise to the next level of organisation, a set that has more complex members (e.g. protein complexes, double-layer lipid membranes), which again interact with each other to form a higher-order system. This process is repeated from the subatomic scale to the population level. Figure 1.2 shows the different levels of organisation, components of those levels and examples of emergent properties resulting from members within that level.

Systems biology is an emerging discipline that aims to describe complex systems by studying the interactions within that system, and by modelling these interactions of known system constituents discovering emergent properties of the system that could not be predicted a priori. Using this approach of data-driven modelling, predictions about higher-order system behaviour can be made, and importantly these predictions may be tested in the system of interest.
### Levels of organisation in biological systems

<table>
<thead>
<tr>
<th>Level</th>
<th>Components</th>
<th>Emergent phenomena</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>Human populations</td>
<td>Transmission dynamics of M. tb</td>
</tr>
<tr>
<td>Multicellular Organism</td>
<td>Human</td>
<td>Disease state: Latent tuberculosis, active tuberculosis</td>
</tr>
<tr>
<td>Organ system</td>
<td>Respiratory system, immune system</td>
<td>Respiratory failure, cough, weight loss</td>
</tr>
<tr>
<td>Organ</td>
<td>Lung, lymph nodes, blood</td>
<td>Macrophage pathology: caseating necrosis, cavitation, exudative effusions</td>
</tr>
<tr>
<td>Tissue</td>
<td>Respiratory epithelium, alveolar tissue, lymphoid tissue, recruited cells</td>
<td>Granuloma formation</td>
</tr>
<tr>
<td>Cell</td>
<td>Mycobacterium tuberculosis, Macrophage, dendritic cell, CD4 T cell</td>
<td>Mycobacterium tuberculosis: growth, non-replicative state, Human: immune cell activation, apoptosis</td>
</tr>
<tr>
<td>Organelle</td>
<td>Cell membranes, phagosomes, lysosomes</td>
<td>Phagocytosis, phagosome-lysosome fusion, phagosomal arrest, autophagy</td>
</tr>
<tr>
<td>Macromolecule</td>
<td>DNA, proteins (receptors, signal transduction) phospholipid assemblies</td>
<td>Signalling cascades</td>
</tr>
<tr>
<td>Molecule</td>
<td>Cytokines, chemokines, metabolites</td>
<td>Antigen recognition sites, ligand-receptor binding, metabolic networks</td>
</tr>
<tr>
<td>Atom</td>
<td>Individual atoms</td>
<td>Chemistry, biochemistry</td>
</tr>
</tbody>
</table>

Figure 1.2: **Levels of organisation in biological systems.** At each level, entities of the same class interact in complex interaction networks, and emergent properties of these entities and their interactions define the next level up. The first column lists the hierarchical levels of organisation of biological systems. Components that are studied at each particular level are shown in the second column (these components do not form an exhaustive list). In the third column, examples of emergent phenomena that result from interactions between components are listed.
1.7 Systems approaches

Systems approaches should be hypothesis driven, global/unbiased and multi-scale. They should iterate over cycles of data collection, hypothesis formulation, model construction and model testing [81] (Figure 1.3). Data collection requires unbiased sampling of the system and parallel measurement of high numbers of parameters at multiple levels of organisation. This is facilitated by the advent of various high-throughput biology technologies that measure whole genome gene expression (DNA sequencing, cDNA/cRNA microarrays, RNAseq), protein levels or presence (mass spectrometry), small molecule metabolites and other large-scale phenomena. Such datasets may then be incorporated into models of gene expression, metabolic pathways and other models. Important in this context is the concept of perturbation, where the experimental system is subjected to specific changes, and the broad-based measurements mentioned above repeated. This data may then be used to test hypotheses about emergent properties of the system formulated using the model.

1.7.3 Systems approaches to tuberculosis

An important contribution of the systems-level understanding of tuberculosis was made in 2010[82]. In this study, the authors collected whole blood for RNA extraction from a cohort of individuals with either active pulmonary tuberculosis or not active pulmonary tuberculosis (comprising latent tuberculosis and individuals without evidence of immune sensitisation to *M. tuberculosis* antigens). Extracted and processed samples were hybridised to Illumina Human HT12v3 gene expression microarrays. The resulting data was first subjected to unsupervised analysis followed by statistical testing for significant differential transcript abundance. This approach [83, 84] yielded an interferon-inducible, neutrophil driven blood transcriptional signature consisting of 393 transcripts, shedding light on the potential role of neutrophils in active tuberculosis. Type-1 interferon-inducible genes were shown to be highly expressed in neutrophils. In subsequent work [85, 86] core features of this signature were independently replicated. This has led to new thinking regarding the role of the innate immune system in the response to tuberculosis, and its interaction with the adaptive response. This signature is being evaluated for diagnostic potential of early identification of active tuberculosis, and has also been shown to revert towards the inactive signature following antimycobacterial therapy [87]. An additional contribution of this paper was the finding that some
Figure 1.3: A systems approach is an iterative process. For each system under study, experimental data collection, data interpretation, modelling, prediction and further experimentation follow each other. The key goal is correct prediction of system states following perturbation of such systems using computational models.
individuals who had the clinical phenotype of latent tuberculosis assigned based on the absence of symptoms and evidence of prior immune sensitisation to *M. tuberculosis* clustered together with cases of active tuberculosis, suggesting that their correct phenotype was subclinical active tuberculosis [82], supporting the idea that tuberculosis should be viewed as a spectrum. As this interpretation lacks experimental support, alternative interpretations of the data should be considered. For instance, the misclassified individuals may have another subclinical condition that produces a transcriptional response that is similar to that associated with active tuberculosis. In this context, it is worth noting that the 86-transcript signature described by Berry et al also distinguishes individuals with melioidosis from healthy controls [88], suggesting that this signature may not in fact be specific for active tuberculosis. The paper concludes that type 1 and type 2 interferon-mediated responses dominate the host response to both infections, and that transcriptional signatures based on interferon signalling cannot distinguish between acute melioidosis and tuberculosis.

In developing systems approaches to tuberculosis, novel mathematical methods need to be developed and implemented such as scoring methods for protein interactions [89], investigation of gene co-expression patterns in *M. tuberculosis* using microarray data [90] and modelling latent tuberculosis using time-course microarray data [91]. Another approach to investigating tuberculosis has been developed by Kirschner et al. Here a systems biology approach [92, 93] is used to integrate experimental data (e.g. flow cytometry, enzyme-linked immunosorbent spot (ELISPOT), Luminex, immunohistochemistry, *in situ* hybridisation (ISH), *in vivo* and *in vitro* cytotoxicity assays, gene knockout (deletion) and cell and cytokine depletions, cytotoxicity assays, 2-photon microscopy) and computational models to study emergent properties of a system consisting of lung and draining lymph nodes [94] down to the cellular level, and find that many aspects of granuloma formation are recapitulated in this model [95]. A view of intracellular host factors important for survival of *M. tuberculosis* has been developed based on genome-wide siRNA screens, which yielded a dense interaction network of 275 molecules [96]. A subset of 74 host proteins was shown to be invariably required for survival of *M. tuberculosis*; over half of these act through regulation of autophagy which implies that modulation of autophagy forms a basis for intracellular survival.

An EU-funded consortium (SystemTB) aims to combine experimentally-driven model development and model-driven experimentation to improve the understanding of the biology of *M. tuberculosis* and its interactions with host cells and mediators by elucidating its interaction networks by
various high-throughput screens performed under multiple conditions involving biologically relevant perturbations (chemical and physical stresses, knock-out or knock-down mutants [97]. In the USA, one of four NIH-funded Systems Biology of Infection centres based at the Broad institute has established a collaborative network that will perform systematic profiling using high-throughput screens during both \textit{in vitro} and \textit{in vivo} growth, integrating these data into predictive computational models of the \textit{M. tuberculosis} regulatory and metabolic networks [97].

1.7.4 Systems epidemiology

Systems epidemiology [98] aims to answer questions about the effect strain diversity on disease phenotypes. In tuberculosis, relatively little is known about strain-specific effects. Questions of this nature will be addressed by parallel high-throughput screens: individual strain sequencing combined with high-resolution host phenotyping using transcriptomics, proteomics, metabolomics and other approaches. One challenge of this approach will be to combine and integrate large-scale genomic data from the pathogen with phenotypic data from the host. One approach may utilise host transcriptomics to define the host response at high resolution, and to correlate the host states with pathogen genomics.

1.7.5 Interface of \textit{M. tuberculosis} and host, and how this is modified

Tuberculosis is defined by the presence of \textit{M. tuberculosis} in the host, and the subsequent immune response. In order for a response to occur, information about the pathogen has to reach the host via an afferent information collection system. This information is processed, and a response generated using effector components of the immune system. This flow of information from pathogen to host is crucial, and occurs via the interface between the host and pathogen. Defining the exact nature of this interface is required to understand host-pathogen information exchange. In addition to the siRNA approach outlined above [96] which identified host autophagy machinery as a crucial part of this interface, another method to understand this information exchange is the use of simultaneous host and pathogen transcriptomics, reviewed in [99]. This approach is challenging, as the pathogen RNA occurs in quantities orders of magnitude lower than host RNA. Array-based methods cannot routinely discriminate between different RNA isoforms, uncover unannotated non-coding
RNA sequences, define transcript borders, and suffer from a low signal-to-noise ratio due to non-specific and background hybridisation. In addition, RNA sample preparation is complex. RNAseq overcomes many of these problems. The method requires RNA to be extracted from samples containing both host cells and mycobacteria. This RNA is subjected to RNAseq, and resultant reads are mapped to both genomes, allowing inferences to be made about correlations of transcriptomic responses. As this technique relies on a stochastic sample of RNA from both host and organism, the absolute amount of RNA recovered from the organism is still much lower than that of the host; this has to be taken into account in the analysis of the dual RNAseq data. In order to partially alleviate this, one might employ techniques to specifically enrich for RNA of the under-represented organism; this is not usually recommended. Another strategy consists of depletion of ribosomal RNA (rRNA); this increases the relative information content of the prokaryotic RNA fraction. Again, this approach is not recommended if the sequencing depth can be increased sufficiently in order to identify transcripts of extremely low copy number. This, however, significantly increases costs.

1.7.6 Compartmentalisation and site of disease studies

Systems approaches may also be applied at the level of site of disease. For example, host transcriptomics may be performed in whole blood and site of disease simultaneously (Deffur et al, manuscript in preparation). For concurrent transcriptomics approaches to two compartments, care must be taken to account for profound differences in cell populations in the compartments (e.g. the site of disease is expected to be enriched for cell types germane to dealing with the infection whereas the whole blood compartment serves as the conduit for the efferent arm of the immune response [84]. One method of dealing with this is deconvolution of total compartment signatures into cell-specific signals [100] and performing cell-specific differential expression analyses [101].

1.7.7 Challenges

Some challenges faced by systems approaches are data integration and model integration. Data sets are generated on high-throughput platforms with unique readouts, and the difficulty becomes integrating the datasets at the level of the biological signal, and not the raw data, which contains both signal and noise. Additional integration challenges are file format conventions, different methods
of data storage (flat files, relation databases), the absence of metadata standards and the lack of precise, structured vocabularies that describe the system components exactly. Some existing models perform an adequate function in recreating emergent properties observed in either the host or the bacterium, making some simplifying assumptions about the other player in the two-component system. What is required is the integration of a complete model of mycobacterial metabolism and responses to stress and other perturbations within a multi-scale model of human immunology. In order for this to be accomplished, a complete description of the interface of host and pathogen, with associated transcriptional response networks is required.

1.7.8 Research priorities for systems approaches in co-infection with *Mycobacterium tuberculosis* and HIV

**Define and delineate (new) phenotypes.** Given the added complexity of phenotypes of co-infection with *M. tuberculosis* and HIV, the full range of phenotypes needs to be elucidated. This requires high throughput screens of healthy, immunosensitised and symptomatic individuals, combined with sensitive imaging techniques (e.g. PET/CT). High-resolution phenotyping is required in order to perform meaningful comparisons between groups of samples that measure thousands of parameters concurrently.

**Controlled vocabularies.** Terminology employed in clinical, animal model and molecular studies needs to be standardised, especially where higher order emergent phenomena are concerned. For example, there is lack of consensus in the literature for the appropriate use of terms like “dormant”, “latent” and “inactive” tuberculous lesions. On some occasions, different terms may mean the same thing, and on others, one term may refer to multiple phenomena. What is required is a strict one-to-one mapping of terms to biologic concepts. This is best accomplished by developing a controlled vocabulary (ontology), which can be used in annotating specific aspects of biological and computational models. Development of a TB ontology is underway under the auspices of a collaborative NHLBI-sponsored effort (TB Systems Biology).

**What is the interface, and how does HIV change it?** One of the main difficulties in integrating models of human immunology and mycobacterial biology is definition of the interface that allows bidirectional information flow between the two systems. In a way, this interface allows the
one system to “read” the internal state of the other system. Many features of this interface are known (ligands and receptors, signalling pathways), but others require elucidation (e.g. role of mycobacterial secreted factors).

**Model integration.** Controlled vocabularies and knowledge of the host-pathogen interface are both key to integrating computational models (e.g. metabolic models of *M. tuberculosis* and models of granuloma formation). Many current models consider one side of the host-pathogen relationship in considerable detail, while at the same time simplifying the biology of the other side. What is required is convincing interaction between host and pathogen that incorporates sufficient detail about both. Predictions of such models are again amenable to testing using either clinical samples or realistic animal models.

**Create integrated models of HIV co-infection.** Models of HIV-TB co-infection will require a significant increase in complexity, due to the increased phenotypic variability introduced by the variable immune dysfunction associated with HIV, and will ideally model all three organisms. They should exhibit known emergent properties of co-infection (e.g. effects of TB on HIV, effects of HIV on TB, immune reconstitution inflammatory syndrome).

### 1.8 Conclusion

Systems approaches to TB-HIV will complement on-going reductionist approaches by synthesis of models capable of predicting emergent phenomena related to this disease. In so doing, we will gain a deeper and more complete understanding of the drivers of pathogenetic processes, and will hopefully identify novel ways in which to interfere with this system in order to produce more favourable clinical outcomes. Systems approaches are challenging, and require a new ways of thinking about TB-HIV co-infection specifically, and infectious diseases research more generally[101], with the goal of ultimately redefining the host-pathogen research paradigm.
2 IMPI-MA: a two-compartment survey of the transcriptional landscape in HIV-TB

Chapter summary

In this chapter I describe IMPI-MA, the study that forms the basis for this thesis. First, the problem of tuberculous pericarditis in the context of a major HIV-TB epidemic is presented, as this disease presentation will serve as a two-compartment model of HIV-TB. Next, hypotheses and study questions are developed, followed by specific aims and objectives. Finally, I present the study design and outline my approach for dealing with a very heterogenous dataset.

2.1 Tuberculous pericarditis

2.1.1 Tuberculosis worldwide and in South Africa

Currently, approximately one third of the world’s population is infected with *Mycobacterium tuberculosis*, and one in ten of those will develop clinically overt disease in their lifetime. In the most recent report on the global TB epidemic [102], the WHO provided the following statistics: In 2011, there were an estimated 8.7 million new cases of TB (13% co-infected with HIV) and 1.4 million people died from TB, including almost one million deaths among HIV-negative individuals and 430,000 among people who were HIV-positive. Worldwide incidence rates continue to fall very slowly. 5% of all cases of tuberculosis have multidrug resistant TB (MDR-TB).

In South Africa, the incidence of tuberculosis is amongst the highest in the world, at 993 cases per 100,000 population per year [102]. A high proportion of this disease burden is borne by HIV-infected individuals: approximately 70% of incident TB cases are HIV infected. Manifestations of
tuberculosis in HIV-infected individuals are highly variable, but an increased proportion of cases present with extrapulmonary tuberculosis.

2.1.2 Tuberculous pericarditis: phenotypes and complications

One of the less common but clinically very important presentations of tuberculosis is tuberculous pericarditis (TB-PC). Approximately 1% of cases have cardiac involvement. In the acute setting, tuberculous pericarditis presents with constitutional symptoms, chest pain and a pericardial effusion. Patients may present with cardiac tamponade requiring emergency drainage of the effusion.

Despite adequate antimycobacterial chemotherapy, up to 25% of patients with tuberculous pericarditis develop constrictive pericarditis, which has a high level of morbidity and 25% mortality at 6 months. The mortality rate rises to 40% at 6 months in patients with HIV infection who have clinical features of immunosuppression, and are not on antiretroviral therapy [103, 104]. This makes TB-PC the second-most lethal form of tuberculosis after tuberculous meningitis (TBM).

Strategies are needed to reduce the high complication rate in patients with TB-PC. The use of corticosteroids in tuberculous pericarditis is still controversial [105, 106], but is currently actively being studied in a randomised, controlled clinical trial.

TB-PC exhibits one of several distinct haemodynamic phenotypes. The phenotypes are identified using haemodynamic measurements obtained at the time of pericardiocentesis, and consist of the following:

1. Dry (acute) pericarditis
2. Purely effusive, with normalisation of right atrial pressure when the pericardium is drained to dryness
3. Effusive-constrictive, with evidence of visceral pericardial involvement and failure of right atrial pressure to normalise when the pericardium is drained to dryness [103]
4. Constrictive pericarditis (late-stage complication)

Understanding the sequence of events that leads to constrictive pericarditis might aid in developing tools to predict which patients with tuberculous pericarditis are at risk of developing constriction, and to find novel ways of preventing constriction by targeted disruption of constriction-inducing
2.2 IMPI and related substudies

The study, detailed below, in essence will try to elucidate biological processes that play a role in the pathogenesis of constrictive pericarditis.

It is clear that pericardial involvement in patients with tuberculosis presents a significant problem in South African healthcare setting. Better understanding of this disease is required in order to develop improved diagnostic and therapeutic modalities. In summary, TB-PC is second only to TB meningitis in terms of lethal potential, and it is an excellent model of extrapulmonary TB, with the ability to sample material directly from the disease site.

2.2 IMPI and related substudies

The Investigation of the Management of Pericarditis in Africa (IMPI Africa) registry [106] was established by the Cardiac Clinic, Department of Medicine at the University of Cape Town in order to study clinical, pathophysiological and outcome characteristics of a cohort of patients with tuberculous pericarditis.

Over the years since the original IMPI-Africa Registry was established, a number of studies have continued to collect data on patients with TB-PC. The broader registry can therefore be divided into several “phases” with different research focus.

1. Original IMPI registry
2. Registry 2
3. Clinical trial of adjunctive immunotherapy in TB-PC

In addition to the three sequential studies above, a number of substudies have sampled from the three populations above:

1. Haemodynamic phenotypes in TB-PC (Ntsekhe et al, manuscript in final review with PLOS ONE)
2. Immunology of tuberculous pericarditis
3. Diagnosis of tuberculous pericarditis
4. Microarray substudy (IMPI-MA) - the subject of this thesis
The complex relationship of study populations between studies and substudies is shown in Figure 2.1

Figure 2.1: **Relationship between various IMPI cohorts.** The figure shows that three large cohorts (ovals) of patients were drawn from a the set of patients with TB-PC seen at Groote Schuur Hospital. A subset of patients presenting with TB-PC underwent pericardiocentesis (red circle). Four substudies were performed on overlapping patient populations (rectangles)

### 2.2.1 Immunology substudy

A collaboration between the Cardiology group leading the IMPI projects and the laboratory of Professor Robert J Wilkinson at the IIDMM, University of Cape Town, has been active since 2006. In work for her Masters and PhD degrees, Kerryn Matthews, together with her supervisor Katalin A Wilkinson, has performed extensive immunological characterisation of samples obtained from individuals with TB-PC.

The following assays were performed:

1. Restimulation assay of whole blood and pericardial fluid using ESAT-6 and CFP-10 with measurement of IFN-γ in the supernatant
2.2 IMPI and related substudies

2. ELISPOT analysis of peripheral blood mononuclear cells (PBMC) and pericardial fluid cells (PFC) for IFN-\(\gamma\) using multiple mycobacterial antigens

3. Surface phenotyping of T-cells from PBMC and PFC using fluorescence-activated cell sorting (FACS)

4. Quantitative RT-PCR (TaqMan) for forty-two genes involved in inflammation and regulation of inflammation

5. Measurement of twenty-four cytokines and other mediators involved in inflammation, fibrosis and tissue destruction in serum and cell-free pericardial fluid using ELISA and luminex

Key findings reported in her PhD thesis [107]

1. Increased recognition of \textit{Mycobacterium tuberculosis}-specific antigens by T cells in the pericardial fluid with a phenomenon of “Pericardial” and “Peripheral dominance” in the recognition of \textit{M. tuberculosis}-specific antigens

2. There are more differentiated antigen experienced CD4+ and CD8+ T cells in HIV-1 uninfected TB-PC patients, while HIV skews the phenotype of memory T cells in the pericardium towards less differentiated CD4+ and CD8+ T cells

3. HIV skews the cell-specific cytokine response towards a more polyfunctional T cell profile, probably contributing to pathology of TB pericarditis, rather than protection

4. The pericardial space has been shown to be largely pro-inflammatory, but it is also profibrotic, with compartmentalisation of pro-collagen genes and lack of fibrotic regulators, irrespective of the HIV status of the patients (see Chapter 5 for additional analysis of this data)

5. There is dysregulation between MMP and TIMP molecule production and binding, which may overstimulate the synthesis of inflammatory and profibrotic cytokines

2.2.2 Haemodynamic phenotypes

In a study submitted to PLOS ONE, Mpiko Ntsekhe, et al report a high prevalence (53%) of the effusive-constrictive phenotype of TB-PC in the study population, and this phenotype is accom-
panied by high interleukin 10 (IL-10) levels in blood and pericardial fluid. This suggests that the inflammatory and anti-inflammatory milieu may be differentially regulated in effusive and effusive-constrictive TB-PC.

### 2.2.3 Synthesis

The findings from the above work suggest the following:

1. Cellular profiles in blood and pericardial fluid in TB-PC may differ significantly between HIV-1-infected and -uninfected individuals. This may be detectable as differential gene expression at whole sample and cell-specific level

2. Several components of the immune response to tuberculosis may be compartmentalised as evidenced by differential mRNA transcript, protein levels and cell population proportions between blood and pericardial fluid

3. In tuberculous pericarditis, the immune response may be differentially regulated based on the haemodynamic phenotype of the pericardial effusion (in blood, pericardial fluid, or both)

### 2.3 Development of a systems approach

The finding of an interferon-inducible neutrophil-driven blood transcriptional signature in human tuberculosis [82] in HIV-1 uninfected study subjects argues strongly for replicating this approach in an HIV-1 infected cohort. In addition, site of disease data as produced by Matthews et al and Ntsekhe et al argue strongly that gene-expression patterns underlying cellular and cytokine responses at the site of disease are amenable to discovery and may shed significant light on site-of-disease pathogenetic mechanisms in HIV-TB. I therefore embarked on developing a theoretical framework within which questions relating gene expression in tuberculosis, HIV and body compartment could be answered, and the results compared.

In this context, it is worth briefly discussing what tissue-level transcriptomes actually represent. The transcriptome is the set of all RNA molecules, including mRNA, rRNA, tRNA, and other non-coding RNA produced in one or a population of cells [http://en.wikipedia.org/wiki/](http://en.wikipedia.org/wiki/)
Transcriptome). It can be measured using array-based and sequencing-based techniques. Importantly, the transcriptome varies dynamically over time, and is strongly influenced by environmental factors. The unit of measure for the transcriptome is “transcript abundance”, which is influenced by factors such as constitutive expression, regulation of the activity of individual genes due to signal processing within cells and stochastic processes. Individual types of cells exhibit a predilection to express particular genes which, taken together, define and establish the specific properties of the cell type.

2.4 Hypotheses and study questions

Three main hypotheses underlie this thesis.

1. Active tuberculosis is distinguishable from not active tuberculosis on the basis of differential gene expression in both HIV-1 uninfected and HIV-1 infected individuals

2. HIV-1 infection is distinguishable from absence of HIV-1 infection on the basis of differential gene expression in multiple health and disease contexts, including active tuberculosis

3. Differential gene expression between matched samples of blood and pericardial fluid is due to both altered cell proportions between the compartments as well as cell-specific differential gene expression

2.5 Aims and objectives

2.5.1 Aims

The broad aim of this work is to provide a survey of the transcriptional landscape of HIV-TB in a two-compartment model at the level of individual genes, gene modules and individual cell types.

2.5.2 Objectives

1. Retrospective selection and prospective collection of matched blood and pericardial fluid samples from individuals with suspected or confirmed TB-PC, and of blood samples of three classes of controls (healthy, LTBI and pulmonary tuberculosis (PTB))
2. Retrospective curation, prospective collection and synthesis of clinical and phenotype data relating to the above samples

3. Measurement of host transcript abundance in pericardial fluid (PCF) and blood using cDNA microarrays

4. Development of an analytic framework in the R programming language that allows for addressing the stated hypotheses and additional research questions. Importantly, results must be output at gene, gene module and cellular levels

5. Placement of differentially expressed genes and modules in biological context by examining gene modules in co-expression networks as well as biological pathways

2.6 Overall study design

This study, though complex in analytic design, is an observational study of a heterogeneous cohort of individuals, each of whom had contributed either a blood or matched blood and PCF samples, depending on disease status. Throughout this manuscript, this study will be referred to as IMPI-MA (IMPI-Africa registry MicroArray substudy).

2.6.1 Contrasts and comparisons

Two concepts are important to differentiate at this point:

1. Contrast: a contrast is defined as a phenotypic variable that defines exactly two states between which differential gene expression is assessed. The full set of contrasts will be described and justified in Chapter 8, where the full dataset and the associated phenotypes will be described. The three contrasts defined \textit{a priori} (as implied by the hypotheses) are:

   a) \textbf{TB-status}, consisting of \textit{active tuberculosis} (either PTB or TB-PC) or \textit{not active tuberculosis} (either healthy individuals or asymptomatic individuals with evidence of prior sensitisation to \textit{Mycobacterium tuberculosis})

   b) \textbf{HIV-status}, consisting of \textit{HIV-1 infected} and \textit{HIV-1 uninfected} individuals, respectively
2.6 Overall study design

c) **Compartment**, consisting of whole *venous blood* or whole *pericardial fluid* samples

2. Comparison: a comparison is defined as an assessment of similarity or difference in two or more groups following assessment of one particular contrast, based on another phenotypic variable, termed the *context variable*. For example, differential expression analysis may be performed for the contrast TB-status in two sample subsets (HIV-1 infected and HIV-1 uninfected), and the results compared. Conversely, HIV-status may be used as the contrast, and *active tuberculosis* and *not active tuberculosis* status used as basis for the comparison. Figure 2.2 shows this graphically.

**Figure 2.2:** The difference between the concepts **Contrast** and **Comparison**. In panel A, the contrast *TB-status* is used to discover a list of differentially expressed genes in two populations. The two gene lists can then be compared on the basis of HIV-status. In panel B, the converse experiment is shown, with *HIV-status* being the contrast and TB-status the comparator.

2.6.2 Phenotype space as hypercube

Given the three contrasts defined above, we can visualise each sample represented mapped to a corner of a three-dimensional cube. Examination of the edges of the cube, shown in Figure 2.3 reveals all possible contrasts and comparisons involving these three parameters. This concept will be expanded upon in Chapter 8, where we will generalise the parameter space for the full analysis to a subgraph of a six-dimensional hypercube. The cubic graphical parameter space is analogous to a three-dimensional space showing principal components of a dataset defined by three parameters. In an ideal situation, the point clouds for each of the eight groups would be arranged in a cube-like fashion, with the eight centroids forming the vertices of a cube.

The three-dimensional cube immediately suggests a design for an analytic algorithm that will
Figure 2.3: **3D parameter space.** Panel A. Three-dimensional cube representing the main parameter space required to address the three stated hypotheses is shown. Abbreviations: **B** blood; **F** pericardial fluid; **A** active tuberculosis; **N** not active tuberculosis; + HIV-1 infected; - HIV-1 uninfected. There are $2^3 = 8$ combinations of the three variables, yielding the eight corners of the cube as shown. Given the eight corners, there are a total of 12 potential contrasts and 6 potential comparisons (See Figure 2.2). Each face of the cube corresponds to one of $2 \times 3 = 6$ views of the dataset. Panel B shows the three orthogonal contrasts used to address the main hypotheses of this thesis.

potentially demonstrate a complete survey of the transcriptional landscape of HIV-TB in a two-compartment model.
3 Materials and methods

Chapter summary

In this chapter I describe generation and analysis of preliminary data based on an RT-PCR approach, IMPI-MA study subjects, the clinical phenotype database, RNA sample processing, further processing of RNA samples for microarray, microarray data generation, microarray data management and a general approach to microarray data analysis as developed in R.

3.1 Preliminary data: RT-PCR data generation and analysis

One subset of the TB-PC immunology data will be discussed here, as I will present an analysis of this data in Chapter 4. In work done prior to the project that is the subject of this thesis, Kerryn Matthews studied matched blood and pericardial fluid samples from individuals with TB-PC by quantitative real-time polymerase chain reaction (qRT-PCR). The generation of this data was performed in 2006-2010, and the analysis reported in her PhD thesis [107]. For the purposes of developing an analytic framework for the microarray study, I re-analysed this dataset using a workflow designed for microarray data. Here I briefly review the laboratory work done by Kerryn Matthews, and describe the analytic pipeline.

3.1.1 RNA collection and processing

Three mL of matched blood and pericardial fluid samples were collected in PAXgene Blood RNA tubes (PreAnalytiX GmbH, CH, catalog number 762165) at the time of pericardiocentesis, and frozen at –20°C pending RNA extraction.
3 Materials and methods

3.1.2 Data generation

All of the following work was performed by Kerryn Matthews and is hereby fully acknowledged. Forty-two genes were chosen to evaluate the host transcriptional response in two compartments based on known biological significance in tuberculosis or HIV pathogenesis. These genes are listed in Table 3.1. RNA was extracted using the PAXgene Blood RNA Kit (PreAnalytiX, catalog number 762174) according to the manufacturer’s instructions, and stored at −80°C. Quantity and quality of the total RNA was determined using a NanoDrop (ND1000, Thermo Scientific, USA). RNA was reverse transcribed to cDNA using the Quantitect Reverse Transcription kit (Qiagen, USA, catalog number 205313) and a total of 1µg/ml of cDNA was used in each RT-PCR reaction. Primers and probes for RT-PCR were purchased from Applied Biosystems as predesigned inventoried assay reagents. The TaqMan Gene Expression Assays (Applied Biosystems, USA) used to determine the gene expression of forty-two genes are listed in Table 3.1, and the experiments were performed in a 25µl final reaction volume under universal cycling conditions on the ABI 7000 platform (Applied Biosystems, USA) as described [30]. Human beta-actin was used as the single internal endogenous control for each sample assayed. All data values output are ΔCT values, calculated by subtracting the cycle threshold of beta-actin from the cycle threshold of the gene of interest. See Figure 3.1 for illustration of this method. On occasions where the gene of interest failed to amplify after forty cycles, the ΔCT value was recorded as “invalid” by the instrument.

![Figure 3.1: RT-PCR: calculation of ΔCT](image-url)

1This subsection is based on original input by Kerryn Matthews
### Table 3.1: 42 genes selected for RT-PCR analysis

<table>
<thead>
<tr>
<th>HGNC Symbol</th>
<th>HGNC Approved Name</th>
<th>HGNC ID</th>
<th>Entrez Gene ID</th>
<th>Protein name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IL1B</td>
<td>interleukin 1, beta</td>
<td>5992</td>
<td>3553</td>
<td>Interleukin-1 beta</td>
</tr>
<tr>
<td>IL2</td>
<td>interleukin 2</td>
<td>6001</td>
<td>3556</td>
<td>Interleukin-2</td>
</tr>
<tr>
<td>IL6</td>
<td>interleukin 6</td>
<td>6018</td>
<td>3569</td>
<td>Interleukin-6</td>
</tr>
<tr>
<td>IL18</td>
<td>interleukin 18</td>
<td>5986</td>
<td>3606</td>
<td>Interleukin-18</td>
</tr>
<tr>
<td>IL24</td>
<td>interleukin 24</td>
<td>11346</td>
<td>11009</td>
<td>Interleukin-24</td>
</tr>
<tr>
<td>TNF</td>
<td>tumor necrosis factor</td>
<td>11692</td>
<td>7124</td>
<td>Tumor necrosis factor</td>
</tr>
<tr>
<td>IFNG</td>
<td>interferon, gamma</td>
<td>5428</td>
<td>3458</td>
<td>Interferon gamma</td>
</tr>
<tr>
<td>CXCL10</td>
<td>chemokine (C-X-C motif) ligand 10</td>
<td>10637</td>
<td>3627</td>
<td>C-X-C motif chemokine 10</td>
</tr>
<tr>
<td>IL4</td>
<td>interleukin 4</td>
<td>6014</td>
<td>3565</td>
<td>Interleukin-4</td>
</tr>
<tr>
<td>IL13</td>
<td>interleukin 13</td>
<td>5973</td>
<td>3596</td>
<td>Interleukin-13</td>
</tr>
<tr>
<td>IL17A</td>
<td>interleukin 17A</td>
<td>5981</td>
<td>3605</td>
<td>Interleukin-17A</td>
</tr>
<tr>
<td>IL22</td>
<td>interleukin 22</td>
<td>14900</td>
<td>50616</td>
<td>Interleukin-22</td>
</tr>
<tr>
<td>IL23A</td>
<td>interleukin 23, alpha subunit p19</td>
<td>15488</td>
<td>51361</td>
<td>Interleukin-23 subunit alpha</td>
</tr>
<tr>
<td>TGFbeta1</td>
<td>transforming growth factor, beta 1</td>
<td>11766</td>
<td>7040</td>
<td>Transforming growth factor beta-1</td>
</tr>
<tr>
<td>IL10</td>
<td>interleukin 10</td>
<td>5962</td>
<td>3586</td>
<td>Interleukin-10</td>
</tr>
<tr>
<td>FOXP3</td>
<td>forkhead box P3</td>
<td>6106</td>
<td>50943</td>
<td>Forkhead box protein 3</td>
</tr>
<tr>
<td>IL8</td>
<td>Interleukin 8</td>
<td>6025</td>
<td>3576</td>
<td>Interleukin-8</td>
</tr>
<tr>
<td>LCN2</td>
<td>lipocalin 2</td>
<td>6526</td>
<td>3934</td>
<td>Neutrophil gelatinase-associated lipocalin</td>
</tr>
<tr>
<td>CAMP</td>
<td>cathelicidin antimicrobial peptide</td>
<td>1472</td>
<td>820</td>
<td>Cathelicidin antimicrobial peptide</td>
</tr>
<tr>
<td>DEFs4A</td>
<td>defensin, beta 4A</td>
<td>2787</td>
<td>1673</td>
<td>Beta-defensin 4A</td>
</tr>
<tr>
<td>ELANE</td>
<td>elastase, neutrophil expressed</td>
<td>3389</td>
<td>1991</td>
<td>Neutrophil elastase</td>
</tr>
<tr>
<td>TGF</td>
<td>epidermal growth factor</td>
<td>3229</td>
<td>1950</td>
<td>Pro-epidermal growth factor</td>
</tr>
<tr>
<td>CTGF</td>
<td>connective tissue growth factor</td>
<td>2580</td>
<td>1490</td>
<td>Connective tissue growth factor</td>
</tr>
<tr>
<td>CSF2</td>
<td>colony stimulating factor 2 (granulocytes-macrophages)</td>
<td>2434</td>
<td>1437</td>
<td>Granulocyte-macrophage colony-stimulating factor</td>
</tr>
<tr>
<td>IL9</td>
<td>interleukin 9</td>
<td>6029</td>
<td>3578</td>
<td>Interleukin-9</td>
</tr>
<tr>
<td>COL1A1</td>
<td>collagen, type I, alpha 1</td>
<td>2197</td>
<td>1227</td>
<td>Collagen alpha-1(I) chain</td>
</tr>
<tr>
<td>COL1A2</td>
<td>collagen, type I, alpha 2</td>
<td>2198</td>
<td>1228</td>
<td>Collagen alpha-2(I) chain</td>
</tr>
<tr>
<td>COL4A1</td>
<td>collagen, type IV, alpha 1</td>
<td>2202</td>
<td>1282</td>
<td>Collagen alpha-1(IV) chain</td>
</tr>
<tr>
<td>COL4A2</td>
<td>collagen, type IV, alpha 2</td>
<td>2203</td>
<td>2203</td>
<td>Collagen alpha-2(IV) chain</td>
</tr>
<tr>
<td>ARG1</td>
<td>arginase, liver</td>
<td>663</td>
<td>383</td>
<td>Arginase-1</td>
</tr>
<tr>
<td>SPARC</td>
<td>secreted protein, acidic, cysteine-rich (osteonectin)</td>
<td>11219</td>
<td>6678</td>
<td>SPARC</td>
</tr>
<tr>
<td>MMP1</td>
<td>matrix metalloproteinase 1 (interstitial collagenase)</td>
<td>7155</td>
<td>4312</td>
<td>Interstitial collagenase</td>
</tr>
<tr>
<td>MMP2</td>
<td>matrix metalloproteinase 2 (gelatinase A, 72kDa gelatinase, 72kDa type IV collagenase)</td>
<td>7166</td>
<td>4313</td>
<td>72 kDa type IV collagenase</td>
</tr>
<tr>
<td>MMP3</td>
<td>matrix metalloproteinase 3 (stromelysin 1, progelatinase)</td>
<td>7173</td>
<td>4314</td>
<td>Stromelysin-1</td>
</tr>
<tr>
<td>MMP7</td>
<td>matrix metalloproteinase 7 (matrilysin, usterine)</td>
<td>7174</td>
<td>4316</td>
<td>Matrilysin</td>
</tr>
<tr>
<td>MMP8</td>
<td>matrix metalloproteinase 8 (neutrophil collagenase)</td>
<td>7175</td>
<td>4317</td>
<td>Neutrophil collagenase</td>
</tr>
<tr>
<td>MMP9</td>
<td>matrix metalloproteinase 9 (gelatinase B, 92kDa gelatinase, 92kDa type IV collagenase)</td>
<td>7176</td>
<td>4318</td>
<td>Matrix metalloproteinase-9</td>
</tr>
<tr>
<td>MMP10</td>
<td>matrix metalloproteinase 10 (stromelysin 2)</td>
<td>7156</td>
<td>4319</td>
<td>Stromelysin-2</td>
</tr>
<tr>
<td>MMP11</td>
<td>matrix metalloproteinase 11 (stromelysin 3)</td>
<td>7157</td>
<td>4320</td>
<td>Stromelysin-3</td>
</tr>
<tr>
<td>TIMP1</td>
<td>TIMP metalloproteinase inhibitor 1</td>
<td>11820</td>
<td>7076</td>
<td>Metalloproteinase inhibitor 1</td>
</tr>
<tr>
<td>TIMP2</td>
<td>TIMP metalloproteinase inhibitor 2</td>
<td>11821</td>
<td>7077</td>
<td>Metalloproteinase inhibitor 2</td>
</tr>
<tr>
<td>TIMP3</td>
<td>TIMP metalloproteinase inhibitor 3</td>
<td>11822</td>
<td>7078</td>
<td>Metalloproteinase inhibitor 3</td>
</tr>
</tbody>
</table>
3 Materials and methods

3.1.3 Analysis pipeline

In order to support development of an analysis pipeline for the microarray data, I developed a simplified version of this pipeline to deal with the RT-PCR data as this data was analogous to microarray data, but far less complex (by almost four orders of magnitude).

Raw RT-PCR data was expressed as delta cycle threshold ($\Delta CT$) in both blood and pericardial fluid samples. In order to emulate data generated in microarray experiments, the $\Delta CT$ values were transformed by $\log_{10} \times 2^{\Delta CT}$, which yielded expression levels relative to beta-actin on a continuous log scale, with smaller values indicating lower expression levels. No further normalisation was performed, as each value was already normalised to beta-actin, and assumptions that apply to large data sets of thousands of genes, as found in typical microarray experiments can not be assumed to hold true for forty-two individually selected genes. Several genes failed to reach the detection threshold by 40 cycles, and a $\Delta CT$ value of 40 was assigned in those cases, representing extremely low levels of gene expression.

All downstream data analysis was performed in R [108], using packages from the bioconductor suite (http://www.bioconductor.org), and additional packages from CRAN (http://cran.r-project.org). The matrix of expression values was clustered by hierarchical clustering with Euclidean distance and complete linkage. Correlation of gene expression in both blood and pericardial fluid was assessed by computing the matrix of pairwise Pearson correlation coefficients. Genes that had very low expression levels ($\Delta CT$ value > 38 in more than 5% of samples) were removed from the analysis by a non-specific filtering step.

Differential gene expression between blood and pericardial fluid was determined using a linear model approach implemented in R using the package limma [109]. In short, log-ratios were estimated between two or more target RNA samples simultaneously by performing least-squares regression modelling for each gene. The standard errors of each model are moderated using an empirical Bayes model. The output of limma was a moderated t-statistic and a log-odds of differential expression for each gene. The resulting log-fold changes (differential expression) and the corresponding statistical significance were visualised using a volcano plot. Visualisation of various gene subsets by principal component analysis was performed using the ClassDiscovery package from [110]. Heatmaps were calibrated by including one column each of minimum and maximum values.
of the raw data. This preserves the colour hue and intensity of each datapoint across all heatmaps. All values were clustered using hierarchical clustering with average linkage.

3.2 IMPI-MA study subjects

In order to address the three hypotheses above, I required RNA samples from “active tuberculosis” cases and “not active tuberculosis controls”, stratified by HIV status and compartment (blood and pericardial fluid). Instead of attempting to prospectively recruit adequate numbers of study subjects, I had the opportunity to utilise samples collected prior to the start of the IMPI-MA project, referred to below as “retrospective cases” and “retrospective controls”. The study subjects and their respective phenotype data and RNA samples are described in detail. Even with a large retrospective sample collection, it became apparent that prospective study subjects and samples were required to reach adequate numbers of samples. Therefore, an additional cohort of prospectively collected samples was generated.

3.2.1 Definitions of cases and controls

Specific definitions apply for certain terms used in this thesis, including case definitions of disease states. These are summarised in Table 3.2.
**3 Materials and methods**

Table 3.2: **Definitions of terms.** Abbreviations: ADA: Adenosine deaminase, CXR: Chest X-Ray. *Tygerberg score* refers to a numeric score based on clinical (fever, weight loss and night sweats) and laboratory (serum globulin and leukocyte count) [111]

<table>
<thead>
<tr>
<th>Term</th>
<th>Criterion</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB Status</td>
<td>Not active tuberculosis or Active tuberculosis</td>
<td>Healthy or Latent TB infection</td>
</tr>
<tr>
<td>TB Status: Not active tuberculosis</td>
<td></td>
<td>Asymptomatic, and no evidence of prior immune sensitisation to <em>Mycobacterium tuberculosis</em> (TST negative and IGRA negative)</td>
</tr>
<tr>
<td>Healthy</td>
<td>Symptoms, TST and IGRA</td>
<td>Asymptomatic, and evidence of prior immune sensitisation to <em>Mycobacterium tuberculosis</em> (TST positive or IGRA positive)</td>
</tr>
<tr>
<td>Latent TB infection</td>
<td>Symptoms, TST and IGRA</td>
<td></td>
</tr>
<tr>
<td>TB Status: Active tuberculosis</td>
<td></td>
<td>PTB (definite) or TBPC (definite)</td>
</tr>
<tr>
<td>PTB (definite)</td>
<td>Demonstration of <em>Mycobacterium tuberculosis</em> in sputum</td>
<td>Sputum smear positive for acid-fast bacilli OR Positive culture or PCR for <em>Mycobacterium tuberculosis</em> in pericardial fluid</td>
</tr>
<tr>
<td>PTB (probable)</td>
<td>Symptoms, CXR</td>
<td>Symptoms and signs of PTB; compatible CXR changes; response to therapy</td>
</tr>
<tr>
<td>TB-PC (definite)</td>
<td>Demonstration of <em>Mycobacterium tuberculosis</em> in pericardial fluid</td>
<td>Positive culture or PCR for <em>Mycobacterium tuberculosis</em> in pericardial fluid</td>
</tr>
<tr>
<td>TB-PC (probable)</td>
<td>Pericardial effusion AND No evidence of pericardial infection with other bacteria AND Elevated adenosine deaminase level in pericardial fluid OR Elevated IFN$\gamma$ level in pericardial fluid OR Good clinical response to anti-TB chemotherapy OR Demonstration of <em>Mycobacterium tuberculosis</em> in a site other than PCF OR Tygerberg score</td>
<td>Demonstration of significant pericardial effusion on echocardiogram Negative bacterial culture of PCF ADA $&gt; 40$ U/L in PCF IFN$\gamma &gt; 50$ pg/ml in PCF Clinical assessment of response to therapy Positive smear, culture or PCR for <em>Mycobacterium tuberculosis</em> at a site other than PCF A Tygerberg score of 6 or greater</td>
</tr>
<tr>
<td>Effusive-constrictive TBPC</td>
<td>Pre-pericardiocentesis right atrial pressure elevated AND low transmural pressure gradient AND Failure of Post-pericardiocentesis RAP to fall to normal levels</td>
<td>$&gt; 8$ mmHg $\leq 4$ mmHg $&gt; 11$ mmHg</td>
</tr>
<tr>
<td>Effusive TBPC</td>
<td>Post-pericardiocentesis RAP falls to normal levels and the post-pericardiocentesis intrapericardial pressure falls to zero</td>
<td></td>
</tr>
<tr>
<td>HIV Status</td>
<td>HIV-1 infected or HIV-1 uninfected</td>
<td></td>
</tr>
<tr>
<td>HIV-1 infected</td>
<td>ELISA for HIV-1</td>
<td>ELISA for HIV-1 reactive</td>
</tr>
<tr>
<td>HIV-1 uninfected</td>
<td>ELISA for HIV-1</td>
<td>ELISA for HIV-1 non-reactive</td>
</tr>
<tr>
<td>Compartment</td>
<td>Blood or Pericardial Fluid</td>
<td></td>
</tr>
<tr>
<td>Blood</td>
<td></td>
<td>Venous blood</td>
</tr>
<tr>
<td>Pericardial fluid</td>
<td></td>
<td>Pericardial fluid collected at pericardiocentesis</td>
</tr>
</tbody>
</table>
3.2.2 Retrospective TB-PC cases

Starting in 2006, Kerryn Matthews in the laboratory of Robert J Wilkinson at CIDRI, IIDMM, UCT started collecting matched blood and pericardial fluid samples form patients being enrolled in Registry 2 of the IMPI-Africa Registry by Mpiko Ntsekhe and colleagues. This specific sample collection period ended in 2010 when the Biosafety Level 3 (BSL-3) facility at the IIDMM was upgraded and all work requiring BSL-3 laboratory conditions ceased.

The majority of subjects during this time were also enrolled in a “Haemodynamic Phenotype” substudy of the IMPI-Africa Registry. The clinical workflow is described in Figure 3.2. Patients were included in the Registry if they were 18 years or older, gave written informed consent, had an effusion of at least 500mm$^3$ (at least 10 mm free fluid at the cardiac apex on echocardiogram) and had tuberculosis suspected or confirmed as aetiology for their effusion. In addition patients were included in the haemodynamic phenotype substudy if right atrial pressures pre- and post pericardiocentesis were available.

1. Screening: Echocardiogram confirms pericardial effusion
2. Patient consented and enrolled in the IMPI-Africa Registry
3. Pericardiocentesis under fluoroscopic guidance in the Heart Catheterisation Laboratory at Groote Schuur Hospital
   a) Effusion tapped to dryness
   b) Blood and pericardial fluid samples collected
   c) In a subset of cases, simultaneous right-heart catheterisation was performed in order to obtain the measurements required for categorisation of haemodynamic phenotype
4. Phenotyping
   a) History and clinical examination
   b) Appropriate microbiological tests to detect *Mycobacterium tuberculosis*
   c) Additional biochemical and other tests on blood

Figure 3.2: Clinical workflow for retrospective TB-PC cases
3 Materials and methods

3.2.3 Retrospective controls

The ILULU partnership, an EU-funded consortium aiming to develop a novel diagnostic marker for tuberculosis, enrolled children and adults in study sites in South Africa and Malawi in six clinical phenotypes:

1. TB+ HIV-
2. TB+ HIV+
3. LTBI (Latent TB) + HIV+
4. LTBI+ HIV-
5. HIV+ with opportunistic infections other than TB
6. HIV- with acute and chronic infections other than TB

Following completion of the first phase of this study, hereafter referred to as the “EU study” (during which cDNA microarrays were used to generate transcriptomic data), duplicate RNA samples remaining at the study sites were available for subsequent use at the discretion of the site principal investigator. I obtained permission from Professor Robert J Wilkinson, principal investigator for the adult study in Cape Town, to utilise selected samples based on needs for positive and negative controls for the IMPI-MA study.

The study subjects for the adult Cape Town arm of the study were enrolled at Ubuntu Clinic, Site B, Khayelitsha, Cape Town. The workflow of enrolment is described in Figure 3.3. Multiple asymptomatic subjects were screened for inclusion into the LTBI arms of the study; a number of these subjects turned out to have negative TST and IGRA responses, and while they were not included in the ILULU consortium study, their samples remained available for study. These constituted the “healthy” controls (see Table 3.2). LTBI+ HIV+ and LTBI+HIV- constituted the “LTBI” controls. Lastly, TB+ HIV+ and TB+ HIV- constituted the “PTB” controls.

3.2.4 Prospective TB-PC cases

Following completion of refurbishment work at the BSL-3 facility at the IIDMM, prospective sample collection could be restarted. From March 2011 to February 2012 I collected blood and pericardial fluid samples from patients enrolled in the diagnostic substudy of the ongoing IMPI
3.2 IMPI-MA study subjects

1. Screening: History and clinical examination
2. HIV counselling and testing
   a) If HIV-1 infected: CD4 count
   b) If HIV-1 uninfected: no CD4 count
3. TST and IGRA performed on asymptomatic patients
4. Investigation of symptomatic patients
   a) CXR
   b) Sputum for AFB and TB culture
   c) Other tests as indicated (specifically to detect other infections or malignancies)
5. Phenotyping based on results of 1-4

Figure 3.3: Clinical workflow for retrospective controls

Registry and/or the IMPI trial, a prospective clinical trial of immunotherapy in tuberculous pericarditis.

1. Screening: Echocardiogram confirms pericardial effusion
2. Patient consented and enrolled in the IMPI-Africa Registry: Diagnostic substudy and/or IMPI-trial
3. Pericardiocentesis under fluoroscopic guidance in the Heart Catheterisation Laboratory at Groote Schuur Hospital
   a) Effusion tapped to dryness
   b) Blood and pericardial fluid samples collected
4. Phenotyping
   a) History and clinical examination
   b) Appropriate microbiological tests to detect *Mycobacterium tuberculosis*
   c) Additional biochemical and other tests on blood

Figure 3.4: Clinical workflow for prospective TB-PC cases
3 Materials and methods

3.2.5 Ethics clearance for the various cohorts

All subjects provided written informed consent. This included sample storage for subsequent analysis. All subjects with TB-PC who were recruited into one of the IMPI cohorts are covered by UCT HREC REF: 102/2003, 402/2005 & 289/2007, while all subjects recruited by the EU study are covered by UCT HREC REF: 012/2007.

3.3 RNA and other Samples

3.3.1 Retrospective TB-PC cases

For each subject, 3 mL blood and pericardial fluid were collected in PAXgene containers. Following manufacturer’s guidelines, the PAXgene tubes were frozen at −20°C. Subsequently, RNA was extracted using the PAXgene Blood RNA Kit (PreAnalytiX, catalog number 762174) according to the manufacturer’s instructions, and stored at −80°C. RNA of 28 matched blood and pericardial fluid samples was used in the RT-PCR assay described in Section 3.1, and the remainder of extracted RNA was kept frozen at −80°C. RNA yield and quality was assessed using NanoDrop 1000 and RIN as determined by AB Bioanalyzer 2100 (Agilent Technologies, USA).

Other blood and pericardial fluid samples were collected, processed and analysed as previously described [107, 112, 49].

3.3.2 Retrospective controls

For each subject, 3 mL blood was collected in PAXgene containers, respectively. Following manufacturer’s guidelines, the PAXgene tubes were frozen at −20°C. Subsequently, RNA was extracted using PAXgene Blood RNA Kit and the extracted RNA was kept frozen at −80°C. RNA yield and quality was assessed using NanoDrop.

3.3.3 Prospective TB-PC cases

All work below was performed by myself in the BSL-3 facility at the IIDMM.
3.3 RNA and other Samples

**RNA samples**  For each subject, 3 mL blood and pericardial fluid were collected in PAXgene containers, respectively. Following manufacturer’s guidelines, the PAXgene tubes were frozen at $-20^\circ$C after being allowed to stand undisturbed for at least two hours. Subsequently, RNA was extracted using PAXgene Blood RNA Kit. The remainder of extracted RNA was kept frozen at $-80^\circ$C. RNA yield and quality was assessed using NanoDrop (NanoDrop 2000c, Thermo Scientific, USA).

**Serum**  For each subject, venous blood was collected in $2 \times 5.0$ mL SST (BD, USA, Catalog number 367955) tubes. Serum was separated using centrifugation of the SST tube at 3000 rpm for 10 minutes with the brake on; serum the above gel layer was removed by pipette and transferred to a Nunc CryoTube Vial (Thermo Scientific, catalog number 375418) (max. volume per vial = 1.8 mL) using a filter tip. Labeled cryotubes were frozen at $-80^\circ$C and recorded in the freezer log.

**Separation of PBMC**  For each subject, venous blood was collected in $2 \times 9.0$ mL VACUETTE Plasma Sodium Heparin tubes (greiner bio-one, USA, catalog number 455051). Peripheral blood mononuclear cells (PBMC) were separated from plasma using the Ficoll method. Blood was mixed with an equal volume of Dulbecco’s Phosphate-buffered saline (PBS) (Sigma-Aldrich Life Science, USA, catalog number D8537) and carefully layered onto Ficoll-Paque (GE Healthcare, USA, catalog number 17-1440-03) in a 50 mL conical tube (BD Falcon, USA, catalog number 352070). The blood/PBS/Ficoll-Paque layered mixture was then centrifuged at 700g for 20 minutes with the brakes off. Following this, theuffy coat containing PBMC (see Figure 3.5) was removed by Pasteur pipette and transferred into 15mL RPMI-1640 (Sigma-Aldrich Life Science, USA, catalog number R8758) medium, which was then made up to 40 mL for the first wash step by centrifugation at 600 x g for 10 minutes (brakes on). After this, the supernatant was discarded, and the pellet resuspended with 10 mL RPMI containing 10% heat-inactivated foetal calf serum made up from FBS Superior stock solution (RPMI+10%FCS) (Biochrom, Germany, catalog number S 0615). The cells were counted twice using standard counting chambers using Trypan Blue (Sigma-Aldrich Life Science, USA, catalog number T8154) and Crystal Violet (Sigma-Aldrich Life Science, USA, catalog number C3886-25G), respectively, and the results averaged. The former allows to distinguish live from dead cells, and the latter lyse erythrocytes which may still be present. The cells were
3 Materials and methods

finally centrifuged at 600 x g for 10 minutes with brakes on, and frozen as described below.

![Figure 3.5: Layering of components after Ficoll separation](image)

**PFC and Cell-free pericardial fluid**  The method used for separating pericardial fluid cells (PFC) from pericardial fluid depended on blood content of the pericardial fluid. In some cases there was gross blood admixture, and the fluid resembled blood macroscopically, and in other cases the fluid was clear, serous fluid.

In case of no or minimal visible blood contamination, the contents of all three sodium heparin tubes containing pericardial fluid were transferred to a sterile 50 mL conical tube and centrifuged at 600 x g for 10 minutes with the brakes off. The supernatant, cell-free fluid (CFF), was removed by pipette and transferred to a cryovial (max. volume per vial = 1.8 mL); in most cases I aimed to freeze 3 x 1.8 mL. The remainder of the supernatant was discarded. The pellet containing PFCs was resuspended in 40 mL RPMI and centrifuged in the first wash step at 600 x g for 10 minutes with the brakes on. Supernatant was discarded and 10 mL of RPMI+10% HI-FCS added to the cells, which were then counted as described above. The final wash step consisted of centrifuging at 600 x g for 10 minutes with the brakes on, and the cells were frozen as described below.

In case of gross blood contamination (i.e. blood visible to the naked eye), the Ficoll method was used as above in order to cope with the large number of red cells, with the following modification: two of the three tubes were transferred to a 50 mL conical tube *without* addition of PBS, and the remainder of the steps were identical to the ones above in the Ficoll procedure for separating PBMC from blood. The remaining sample tube was processed by the standard PCF method in order to recover cell free fluid; the cells obtained in this process were discarded.
Freezing procedure for PBMC and PFCs After the second wash step in RPMI+10% HI-FCS, the supernatant was discarded. $1 \times 10^6$ cells (PBMC or PFC) were resuspended in 1 mL of freezing solution by carefully adding 0.4 mL of RPMI+10% HI-FCS to the pellet with a pipette, mixing carefully, and adding 0.5 mL of freezing medium (20% dimethylsulfoxide in 10% HI-FCS) drop-wise, while gently agitating the mixture. The cell/ cell-freeze mix was transferred to a labelled cryovial which was in turn transferred to a Mr Frosty and frozen at $-80^\circ$C overnight. After overnight storage, the cells were transferred to the vapour phase of liquid nitrogen for long-term storage, and the storage log completed.

3.3.4 Summary: RNA samples and banked specimens

RNA samples from blood, and in cases with TB-PC also pericardial fluid, were collected in a uniform way as per manufacturer guidelines, stored in identical fashion prior to extraction and extracted using identical kits. This addresses one potential source of technical bias in microarray studies: use of different RNA collection, storage and extraction methods. Banked serum, cell free fluid, PBMC and PFCs are available for future validation work.

3.4 Clinical phenotype database

Clinical phenotype data for the RNA samples used in the IMPI-MA study was obtained by multiple investigators over a five year period. All retrospective data was obtained in electronic format from the investigators who had collected it, and I captured and collated all prospectively collected clinical data (SET-2).

3.4.1 Data sets

Data sets are listed in Table 3.3. The data sets are heterogenous, as each was originally developed for a specific purpose other than provision of clinical phenotype data for IMPI-MA. These datasets required many hours of work to produce, and have been used in supporting numerous publications, and will continue to support others. For the purposes of IMPI-MA, I required one dataset which used identical variables for parameters of interest. This was to be achieved by integrating the data, ensuring the ability to consistently phenotype all samples used in the microarray analysis.
<table>
<thead>
<tr>
<th>Source</th>
<th>ID</th>
<th>Investigator</th>
<th>Time period</th>
<th>N</th>
<th>Description of data types used in IMPI-MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPI phase 2</td>
<td>IMPI phase 2 comprehensive</td>
<td>Mipk Ntsekhe, Faisal Syed</td>
<td>22/1/2006 - 1/3/2008</td>
<td>85</td>
<td>ID, demographic, clinical, clinical laboratory, ECG, microbiology, cytokine, haemodynamic and outcome data</td>
</tr>
<tr>
<td>IMPI phase 2</td>
<td>ECP plus lab variables</td>
<td>Mipk Ntsekhe needs to be inferred from dataset [1] by using identifiers</td>
<td>68</td>
<td>ID, demographic, clinical, haemodynamic, clinical laboratory, ECG and outcome data</td>
<td></td>
</tr>
<tr>
<td>IMPI phase 2</td>
<td>ECP outcomes</td>
<td>Mipk Ntsekhe needs to be inferred from other dataset as dates incorrect</td>
<td>68</td>
<td>ID, demographic, haemodynamic, HIV and outcome data</td>
<td></td>
</tr>
<tr>
<td>IMPI phase 2</td>
<td>ECP cytokines</td>
<td>Mipk Ntsekhe needs to be inferred from dataset [1] by using identifiers</td>
<td>61</td>
<td>ID, demographic, haemodynamic phenotype, cytokines</td>
<td></td>
</tr>
<tr>
<td>ELISPO</td>
<td>ELISPO</td>
<td>Kerryn Matthews needs to be inferred from dataset [1] by using identifiers</td>
<td>41</td>
<td>ID, cytokines, ELISPOT</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Sources of clinical phenotype data used in IMPI-MA
<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Investigator</th>
<th>Time period</th>
<th>N</th>
<th>Description of data types</th>
<th>Format</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACS</td>
<td>7-22</td>
<td>Kerryn Matthews</td>
<td>needs to be inferred from dataset [1] by using identifiers</td>
<td>various</td>
<td>ID, numbers, proportions</td>
<td>xlb, csv, pzf</td>
<td>16 files, no true source data, redundancy, missing dates</td>
</tr>
<tr>
<td>RF-PCR</td>
<td>23</td>
<td>Kerryn Matthews</td>
<td>needs to be inferred from dataset [1] by using identifiers</td>
<td>28</td>
<td>ID, delta CT</td>
<td>xlx</td>
<td>Only some source data (SDS), missing dates</td>
</tr>
<tr>
<td>HIV-viral load</td>
<td>24-25</td>
<td>Kerryn Matthews</td>
<td>needs to be inferred from dataset [1] by using identifiers</td>
<td>16</td>
<td>ID, viral load PCF, HIV data</td>
<td>xls, xlx</td>
<td>Missing dates</td>
</tr>
<tr>
<td>Whole blood assay</td>
<td>26</td>
<td>Kerryn Matthews</td>
<td>needs to be inferred from dataset [1] by using identifiers</td>
<td>59</td>
<td>ID, IFNγ responses to various antigens</td>
<td>xlx</td>
<td>Missing dates</td>
</tr>
<tr>
<td>Immunology metadata</td>
<td>27</td>
<td>Kerryn Matthews</td>
<td>needs to be inferred from dataset [1] by using identifiers</td>
<td>110</td>
<td>Metadata of all immunology samples</td>
<td>xlx</td>
<td>Missing dates</td>
</tr>
<tr>
<td>Prospective TB-PC cases</td>
<td>30</td>
<td>Armin Deffur</td>
<td>1/3/2011 - 2/2/2012</td>
<td>51</td>
<td>ID, demographics, clinical, imaging, BCG, echocardiography, clinical laboratory, microbiology, IMPI-MA sample metadata</td>
<td>xlx</td>
<td>baseline data only</td>
</tr>
</tbody>
</table>

Table 3.3: (cont.) Sources of clinical phenotype data used in IMPI-MA
3 Materials and methods

3.4.2 Integration strategy

The strategy for production of a single clinical phenotype database is outlined in Figure 3.6. This was a complex process requiring multiple rounds of data cleaning and development of computer code in Mathematica [113] that automated the process of TB class assignment based on the complex case definitions in Table 3.2.

1. Merge datasets 1-4; clean data and fully eliminate redundancy. Folder review of study and hospital folders to resolve any conflicts. Output = SET-1_clinical
2. Merge datasets 5-27; clean data and fully eliminate redundancy. Ensure correct matching of identifiers to individuals in clinical dataset. Output = SET-1_immunology
3. Merge SET-1_clinical and SET-1_immunology. Each row in flat-file database contains all data on on study subject. Output = SET-1_full
4. Identify potential microarray candidates using the following criteria: Blood RIN and PCF RIN of matched blood and PCF samples > 6.0
5. Export clinical subset of this data as SET-1 (immunology data removed as not available in SET-2 or controls)
7. Entire SET-2 exported as microarray candidates as RIN data not available (RIN to be performed in UK)
8. In Mathematica, calculate tuberculosis class and export as SET-1 and SET-2
9. Merge SET-1 and SET-2 into full TB-PC clinical phenotype dataset
10. Search EU data for potential array candidates based on RNA availability and clinical phenotype (Healthy, LTBI and PTB)
11. Verify RNA availability for potential samples
12. Match the three groups to identify 20 each of Healthy, LTBI and PTB using age, sex, HIV-status, CD4 count as matching criteria
13. Extract additional relevant clinical phenotype data from EU database and export data as Controls, with identical column headers in the flat file datasheet
14. Manually merge TB-PC and Control datasets into single file
15. Duplicate phenotype entries as required, so that each sample type (blood and pericardial fluid) has a full phenotype record
16. Examine final dataset for missing data, and perform final folder review of various study and hospital folders to close gaps where possible
17. Export to production CSV file, to be used in all microarray analyses.

Figure 3.6: Workflow for data integration
3.5 RNA processing for microarray

3.5.1 Extraction and collation

RNA was extracted as described above, using the PAXgene Blood RNA Kit (PreAnalytiX, catalog number 762174) according to the manufacturer’s instructions. RNA was extracted by different individuals and different times, depending on the original study for which the RNA was intended.

1. Retrospective TB-PC cases: Kerryn Matthews and myself

2. Retrospective controls (EU study): Nzwaki Bangani
3 Materials and methods

3. Prospective TB-PC cases: Nzwaki Bangani and myself

As described previously, RNA samples were selected by different criteria depending on the original study. As all samples in the retrospective TB-PC case series had RIN data available, only high-quality samples were selected. The EU study controls had no RIN data available, so simply sample availability was used as criterion to select sixty microarray candidates. Lastly, the prospective TB-PC case series had no RIN data available, so all available matched blood and pericardial fluid samples were selected as array candidates.

3.5.2 Shipping and storage

The selected samples were shipped on dry ice to the National Institute for Medical Research in London (NIMR), UK, for further processing and microarray analysis, under an agreement with our collaborator there, Dr Anne O’Garra, head of Immunoregulation at the NIMR and senior author of the Nature paper [82] on which our laboratory had previously collaborated.

3.6 Microarray workflow

3.6.1 RNA preparation for microarray

All RNA processing for microarrays, as described in this subsection was performed by Christine Graham in the laboratory of Anne O’Garra, and her contribution to this project is hereby acknowledged. An outline of the process is shown in Figure 3.7.

The RNA preparation process followed the method outlined in [114]. The first step in preparing the extracted RNA samples for microarray was to verify RNA quality and yield, using an AB Bioanalyzer 2100 instrument at the NIMR. This step was used to select samples for further processing, as samples with degraded RNA (i.e. RIN values < 6) or low yield (i.e. yield < 1250 ng) would not be expected to produce useful results. Selected samples were then subjected to a globin reduction step, in order to reduce interference from globin mRNA message with the transcripts of interest [115, 116]. Quality control was then performed on all globin-reduced samples, and all samples passing QC criteria taken forward the following step: cRNA generation. In short, the mRNA in the sample was converted to cDNA, which in turn was converted to cRNA. A final QC
step was performed to confirm high-quality cRNA for further use in microarray data generation. All samples were processed independent of original sample type (blood and pericardial fluid) and clinical phenotype. The samples passing all QC checkpoints were then randomly assigned to positions individual array chips. This array randomisation is vital to reduce the batch effects that are due to the specific chip used [117]. The workflow for the RNA preparation step is summarised in Figure 3.7.

3.7 Microarray data

3.7.1 Array technology

The BeadArray technology was first described in 2004 [118], as was a method used to decode such arrays [119]. All microarray data was generated on the Illumina BeachChip platform, using the current version of the assay and Illumina Human HT12-v4.0 BeadChips. Each BeadChip contains twelve arrays; generating multiple arrays on a single chip further reduces batch effects. Each array contains approximately 1-1.5 million beads, representing in excess of 47,000 probe sequences, randomly distributed across the array, with approximately 30 replicates of each probe type on each array. Given the random bead distribution, each array is unique, and requires decoding [119] prior to use. The decoded array is then used for RNA hybridisation.
3 Materials and methods

1. Starting material: mRNA extracted from human sample
2. Globin reduction using GLOBINclear kit (Ambion, USA, catalog number AM1980) (a method utilising capture of unwanted RNA by magnetic beads)
3. cRNA generation and purification (Illumina TotalPrep -96 RNA Amplification Kit,) (Illumina, USA, catalog number AMIL1791)
   a) Reverse transcription to synthesise first-strand cDNA
   b) Second strand synthesis to generate double-stranded cDNA
   c) cDNA purification to remove RNA and other components that could potentially inhibit or interfere with downstream reactions
   d) in vitro transcription to generate multiple copies biotinylated cRNA from double-stranded cDNA templates
   e) cRNA purification to remove unincorporated NTPs, salts and other residuals which could interfere with the direct hybridisation assay

Figure 3.8: Workflow for RNA preparation

3.7.2 RNA hybridisation and addition of fluorophore

All microarray work, starting with biotinylated cRNA prepared by Christine Graham, was performed by Harsha Jani at the Microarray Core Facility at the NIMR. The workflow for the microarray work is listed in Figure 3.9.

1. Normalised quantities of cRNA are dispensed onto the BeadChip
2. The loaded BeadChips incubated overnight (10-14 hours) at 58 °C
3. Wash BeadChips using high-temperature wash buffer and follow with room-temperature washes
4. Apply streptavidin-Cy3 to BeadChips; this will allow for signal detection using a fluorescent light source as the streptavidin-Cy3 fluorophore will bind to biotinylated cRNA present on the chip
5. Final wash steps and drying

Figure 3.9: Workflow for array hybridisation
3.7 Microarray data

3.7.3 Imaging

After hybridisation, the BeadChips were scanned at the Microarray core facility using an Illumina iScan microarray scanner. This step converted relative transcript abundance into a proportional light signal. This light signal was captured during the imaging step by a powerful CCD camera, which resulted in a high-resolution TIFF image. Each array was imaged in two passes, and the two resulting images were then combined into a single image by the iScan software. At this stage, the chip decoding data, unique to each BeadChip was loaded onto the controlling computer, in order to assign bead identifiers to each bead on the chip.

3.7.4 Image processing and data summarisation for further analysis

The decoded high-resolution images generated by the iScan scanner were processed further using GenomeStudio, resulting in summarised probe-level data. Each step in the summary process reduced the number of observations per array. This is illustrated for a single array in Table 3.4.

<table>
<thead>
<tr>
<th>Level</th>
<th>Data type</th>
<th>Interpretation</th>
<th>Number of observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw image</td>
<td>Pixel intensities</td>
<td>One observation per pixel per array</td>
<td>approx. (1.25 \times 10^6)</td>
</tr>
<tr>
<td>Bead-level</td>
<td>Bead intensities</td>
<td>One observation per bead per array</td>
<td>approx. (1.0 - 1.5 \times 10^6)</td>
</tr>
<tr>
<td>Probe level</td>
<td>bead intensities summarised by bead type</td>
<td>One observation per probe type per array</td>
<td>47,231</td>
</tr>
</tbody>
</table>

The summarised bead intensity value consists of an average intensity value for a particular bead type, together with the number of replicates and the standard deviation for the observation. This is useful for the data transformation step.

The final microarray data was provided by the NIMR Microarray Core Facility as a tab-delimited text file containing summarised probe-level data.
3.8 Data analysis overview

3.8.1 Computing environment

All data analysis was performed on a mid 2010 MacBook Pro with dual-core Intel Core i7 processors and 8GB non-ECC RAM running OS X 10.8.4. R software was used for the entire analysis workflow.

3.8.2 Analysis framework

All data analysis was done in R version 2.15.2 using a set of scripts, and utilising several R and bioconductor packages.

General data analysis revolves around analysis of RT-PCR data (see above), presentation of an unbiased view of all microarray data, and demonstration of the relation of the IMPI-MA data to previous studies of tuberculosis using blood transcriptomics approaches.

The core of the analysis revolves around seven questions (see Table 8.1) pertaining to the three main hypotheses. Multiple analyses of different, overlapping subsets of the data in fact constitutes a meta-analysis, as the goal is not only to evaluate a single contrast and reach a conclusion, but to evaluate multiple contrasts, compare the results and then reach a more generalised conclusion about the pathogenesis of HIV-TB in a two-compartment model. Each of the seven questions is addressed using complementary analytic methods: detection of differential gene expression for the contrast of interest, deconvolution of the data in order to estimate proportions of specific cell types in the sample as well as cell-type-specific differential gene expression, and modular analysis by detection of modules on the basis of co-expression networks in the microarray data. The scientific basis and methodology underlying these analyses will be described in the remainder of this section. The remaining sections of this chapter, together with Chapter 8, outline the R scripts that underlie various aspects of the analysis, and shows how these scripts together constitute an end-to-end solution for reproducible and well-documented (meta-) analysis of heterogenous microarray datasets that consist of data based on samples from different tissue types and individuals with different disease phenotype classes. All source code is reproduced with syntax highlighting in the Appendix (A), and has been annotated to facilitate readability. The scripts produce output in three main classes: RData files for further processing, images and delimited text files that can be imported in other
3.8 Data analysis overview

3.8.3 Microarray analysis in general

The approach to studies involving microarray data should include pre-defined research questions. Overall hypotheses must be stated prior to the analysis of the data, and the results presented in the context of the hypotheses. Pre-stating clear hypotheses facilitates the process of selecting a data analysis goal and therefore tools to achieve this goal. In general supervised and unsupervised methods may be applied to microarray data, and their use depends on the research question (see below).

Analysis of microarray data usually has one or more of three goals [120]: Class Comparison, Class Prediction and Class Discovery. The aim of class comparison is to find genes that are differentially expressed between groups of samples that belong to known and defined classes. The analysis is supervised, as the classes are known and this knowledge is used to find differentially expressed genes. Class prediction uses unsupervised methods to separate classes in the data without prior knowledge. These predicted classes can then be compared to the real classes, and sensitivity and specificity calculated. Usually, the classifier is built using a training set, and then tested on a separate test set. Class discovery again uses unsupervised methods to discover previously unknown classes of samples. This may be used where sample phenotyping is unable to further differentiate phenotypes within a set of samples that is suspected to be heterogenous at the molecular level. This approach has been used in oncology to classify morphologically identical tumours into subclasses that can be linked to survival or other clinico-pathological outcomes [121, 122].

3.8.4 Reduction in technical bias: array transformation and normalisation

As stated previously, there are many factors (which are not due to the controlled factors in an experiment) that may influence the intensity of a particular probe type on a BeadArray chip, for instance quantity of mRNA. In order to reduce the impact of this source of bias, array data should be normalised. Multiple methods as applied to Illumina HumanHT12 v3 BeadArray were compared in a recent paper[123], and on the basis of that paper, as well as recommendations in the documentation of Illumina-specific analysis packages in R [124], variance stabilising transformation followed by
3 Materials and methods

quantile normalisation was performed.

3.8.5 Differential expression analysis

The most fundamental analytic technique applied to microarray datasets is the detection of differential expression of probes or genes between two or more classes of samples. While simple in principle, in practice this problem is quite complex. The individual measured intensity values are affected by multiple sources of fluctuation and noise. This problem is in part addressed by including technical replicates on the microarray chip; the Illumina HumanHT12v4 chip used in this study contains an average of 30 replicates for each probe type on the chip.

Multiple methods for detection of differentially expressed genes have been described. Many rely on some form of statistical hypothesis testing, where the null hypothesis $H_0$ is that there is no difference in transcript abundance for probe (or gene) $x$. A simple method of testing this hypothesis is to perform a simple t-test; all genes that are significant at a pre-specified p-value are labelled as differentially expressed. This approach is likely to yield many false-positives, given the typical number of tests performed (20,000-48,000 for Illumina data). A more rigorous approach would be to subject the results to some form of multiple testing correction. A non-exhaustive list of potential methods for selecting differentially expressed genes is shown in Table 3.5 (Table based on [125]).

In the analysis of the IMPI-MA data, we employ an approach first described by Smyth [109], namely the use of a moderated t-statistic in selecting differentially expressed genes. This method is widely used in the literature, and yields robust results.

This approach aims to fit a linear model to expression data that fully models the systematic part of the data. The model is specified by a design matrix, where each row represents one array, and each column represents one coefficient that describes the RNA source (i.e. a phenotype variable). The purpose of this approach is to describe and account for the variability of the data. For single channel arrays, linear modelling is similar to ordinary ANOVA or multiple regression, except that a model is fitted to every gene (or probe type). Once the model is derived, an empirical Bayes method is used to moderate the standard errors of the estimated log-fold changes. This results in estimators that have robust behaviour, even for small numbers of arrays. The posterior odds statistic is thus reformulated in terms of the moderated t-statistic where the posterior residual standard deviations
### 3.8 Data analysis overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fold Change</td>
<td>Select threshold</td>
<td>Simple and intuitive</td>
<td>arbitrary</td>
</tr>
<tr>
<td>Unusual Ratio</td>
<td>Ratio of experiment to control &gt; 2 standard deviations from mean</td>
<td>Simple</td>
<td>always selects 5% of all genes</td>
</tr>
<tr>
<td></td>
<td>experiment/control ratio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hypothesis testing</td>
<td>t-test followed by multiple testing correction</td>
<td>Powerful</td>
<td>Very conservative, assumes genes are independent</td>
</tr>
<tr>
<td>ANOVA</td>
<td>Explicit model of all sources of variance</td>
<td>Each source of variance is</td>
<td>Requires specific experimental design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accounted for</td>
<td></td>
</tr>
<tr>
<td>Noise sampling</td>
<td>Estimate noise and calculate confidence levels for gene regulation</td>
<td>Better sensitivity than</td>
<td>Provides estimates only for log-ratios of genes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>unusual ratio and better</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>specificity than fold change</td>
<td></td>
</tr>
<tr>
<td>Model-based maximum</td>
<td>Generate mixture model for distribution of observed log ratios</td>
<td>General, powerful, flexible</td>
<td>Require solution of complex non-linear equations;</td>
</tr>
<tr>
<td>maximum likelihood</td>
<td></td>
<td></td>
<td>unreliable for smaller samples</td>
</tr>
<tr>
<td>estimation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAM</td>
<td>Permutation-based to discover genes differentially regulated for a</td>
<td>Useful for small sample sizes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>given false discovery rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moderated t-statistic</td>
<td>Fit linear model to each gene and determine whether coefficient</td>
<td>Useful for small sample sizes;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>differs significantly from zero</td>
<td>powerful</td>
<td></td>
</tr>
</tbody>
</table>
are used in place of ordinary standard deviations, resulting in more stable inference.

In the model, the array intensity values are represented by a continuous variable termed the response variable. The contrast(s) of interest are represented by one or more categorical variables (explanatory variables or predictors). The goal of the procedure is to test the association between predictors and response. Equation 3.1 shows the model:

\[ y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \beta_0 + \epsilon_i \]  

Here, \( y_i \) is the response for the \( i \)th sample, \( \beta_j \) the coefficient for the \( j \)th predictor, \( x_{ij} \) the value of the predictor \( x_j \) of the \( i \)th sample, \( \beta_0 \) the intercept of the model, and \( \epsilon_i \) the error term (or residual), representing noise and the effect of all predictors not considered in the model.

A microarray experiment consists of a large number of probes or genes that have to be modelled, therefore the equation above can be represented in matrix form as

\[ y = X \beta + \epsilon \]  

where \( y \) is the response vector, \( X \) the design matrix, \( \beta \) the vector of coefficients and \( \epsilon \) the vector of residuals. Building a linear model then reduces to finding the values of the \( \beta \)s (coefficients) that would minimise the error, i.e. solving \( y = X \beta \). The R package limma implements an ordinary least squares solution, as \( X \) is usually not a singular square matrix, and makes the following assumptions:

1. The residuals \( \epsilon_i \) are normally distributed
2. The mean (expected value) of the residuals is zero and is independent of the values of the predictors
3. The variance of the residuals does not depend on the values of the predictors (homoscedasticity)
4. The value of the residual for one sample is unrelated to the amount of residual for another sample

A basic limitation of parametric modelling is that this approach only finds the best values for the parameters available in the model.
Once the fitted model with the moderated t-statistic is produced, multiple testing correction is applied in order to reduce the number of false-positive calls. Throughout all analyses, a multiple testing correction method described by Benjamini and Hochberg [126, 127], based on a specified false discovery rate has been applied.

Issues with this study:

This study aims to compare the results of multiple differential expression analyses, each based on a subset of data with an variable number of samples in each contrast category. As each differential expression analysis attempts to identify an effect size that may differ widely between sample subsets, from small (difference between healthy and LTBI in blood in HIV uninfected) vs. large (e.g., differences between gene expression in matched blood and pericardial fluid samples), the statistical power of each DE analysis is different. Some sample subsets consist of large numbers in each group, and the likely effect size is large, while others consist of small numbers and aim to detect a small effect size, therefore the type II error rate for each DE analysis is different.

This is dealt with by limiting the number of called DE probes to a maximum of 2000, and by including as a minimum the top 300 DE probes, regardless of whether the adjusted p-value is significant or not. In the latter case, such an analysis will be regarded as exploratory, but may still offer some biological insight and aid in the development of testable hypotheses.

3.8.6 Deconvolution and cell-specific differential expression

A common criticism of microarray approaches in general is that the expression profiles for a given sample strongly depend on the cellular composition of the samples in question. Now, if the cellular composition between samples varies widely, differences in gene expression profiles are in the first instance ascribable to the differences in cellular composition, and not necessarily to differential use of transcriptional pathways. In addition, the power to detect differentially expressed genes between samples is strongly affected by sample variation in cell-type frequencies [128, 129, 130]. Therefore, comparisons of gene expression between samples of different tissues (like blood and pericardial fluid) are likely to yield many differentially expressed genes, most of which are uninformative regarding true biological differences, and subtle differences in cell-type specific gene expression in closely related conditions may be masked by the noise of the cell-type frequency component in the
expression signal. To solve this dilemma, two approaches can be taken.

First, the samples may be separated into constituent cell populations. For example, a whole blood sample may be separated into pre-defined cell subsets (neutrophils, CD4 T cells, CD8 T cells, and others). Then, RNA is extracted separately and microarrays performed. This approach has a number of disadvantages: it is time-consuming and expensive, must be done prospectively, and the physical separation of the cells may have unexpected effects on gene expression in the separated cells [131, 132]. This approach was not followed in this study.

The second option is to determine the proportions of cell types in the samples, and then perform a cell-type specific differential expression analysis on the expression matrix based on these proportions. Determination of cell-type proportions in each sample may be performed by physical means (e.g. flow cytometry) or computationally, using a method called matrix deconvolution and first described by Lu et al [133]. The analysis of the IMPI-MA dataset implements the matrix deconvolution approach utilising a least-squares fit algorithm described by Abbas et al [100] and implemented in the CellMix package [134].

Estimation of cell-type specific differential gene expression may also be performed computationally. This relies on known (or estimated) proportions of cells for each sample, and subsequent application of differential expression analysis. A method utilising the permutation-based method called significance analysis of microarrays (SAM) has been described [135] and will be used in this analysis. This method is also implemented in the CellMix R package. Estimates of cell-type specific differential expression may uncover true biological differences between samples with variable cellular composition.

A number of methods for gene expression deconvolution have been described (e.g. [100, 136, 137, 138]) and a full review can be found in the comprehensive documentation of the CellMix package. Here I briefly describe the method used by Abbas et al [100], as this method was most suitable to the data to be analysed. The objective of the deconvolution method is to find the solution for a convolution equation

\[
AX = B
\]  

(3.3)

Here, \( B \) is the matrix of gene expression data, \( X \) is the matrix of proportions of cell types in all
3.8 Data analysis overview

samples and $A$ is the matrix of known expression levels of genes in all cellular components of $B$, which is convolved with $X$. This equation is solved for $X$ by standard least squares fitting using the method pioneered by Lu [133], where one solves for $X$ with the constraint that the resulting matrix contains only non-negative entries.

This approach requires knowledge of $A$, and in the Abbas paper [100], the authors describe their method of generating this matrix of cell-type specific signatures, its validation and subsequent application to a whole-blood microarray study of systemic lupus erythematosus. The CellMix package implements multiple deconvolution methods. One method (gedBlood) replicates the Abbas method exactly, and utilises the above cell-type signature matrix; this was used in the analysis of the IMPI-MA data.

Once cellular proportions for all samples represented in the expression matrix are known, differential gene expression for each cell type represented in significant numbers should become possible. A method to achieve this was published in 2010 [135].

The following two assumptions are made in the analysis as reported:

1. Blood and pericardial fluid are assumed to have the same cellular composition, differing only in proportion

2. These cell-types consist of the cell-types isolated and studied by Abbas

In a study published in 2006 [139], Reuter et al performed flow cytometry on whole blood and pericardial fluid in patients with tuberculous pericarditis (including HIV-1 infected and -uninfected groups). Their findings show that the three most commonly cell-types were neutrophils, lymphocytes and monocytes, while NK cells also occurred. The data also show that proportions of cell types in blood and pericardial fluid, in both HIV-1 infected and -uninfected individuals, are markedly different, with neutrophils being the most abundant cell-type in blood, but much less frequent in pericardial fluid.

Despite the promise of using microarray data in order to infer cell proportions, some limitations of this approach need to be recognised. Firstly, this approach only yields indirect evidence of different cell proportions and cell-specific differential expression, and the results require verification prior to accepting the results at face value. An experiment that does this is described below. Secondly, the ability to detect differences in whole blood may be limited due to the dilution of the
signal in mixed cell populations. In published papers[100, 135] however, the authors are able to reliably detect cell proportions and cell-specific differential expression. Indeed, sample heterogeneity may mask true differential gene expression when performing differential gene expression analysis on whole blood transcriptomes, while cell-specific differential expression correctly identifies the cell type and transcript list of differentially expressed genes[135].

Validation of the deconvolution approach  Given the limitations discussed above, validation of the deconvolution method is warranted. This would be best achieved by collecting flow cytometry data on whole blood on the same samples on which array data is available. In the absence of such data (no whole blood flow cytometry data was generated for IMPI-MA), I used existing data generated in the Berry et al study [82] (London Test Set) to partially validate this method.

Flow cytometry data was generated on whole blood samples on 27 patients, approximately equally sampled from healthy controls, LTBI and PTB cases. 17 of these patients also had Illumina HT12v3 microarray data available. All flow cytometry data was provided by Fin McNab in the Anne O’Garra laboratory, NIMR, London, UK.

The flow cytometry data was received as three Microsoft Excel files; one each for healthy controls, LTBI controls and PTB cases. For each workbook, one worksheet was dedicated to one class of cells. I copied the original data into a new Excel workbook with one worksheet for each class of cell, converting all cells containing formulas to static values; each worksheet contained data for all patients. Next, I removed all rows (samples) from this for which no microarray data was available, and all columns which did not contain relevant data for the intended comparison.

The following cell subsets were enumerated: Total T cells, CD4 T cells (and multiple subsets), CD8 T cells (and multiple subsets), other T cells (i.e. double negative for CD4 and CD8), dendritic cells (plasmacytoid and myeloid subtypes), inflammatory and non-inflammatory monocytes, NK cells, and neutrophils. Full methods describing the flow cytometry approach are available in the supplementary data for the Berry et al paper [82]. Based on proportions seen at flow cytometry and total number of white blood cells as based on complete blood counts, numbers of each subtype were calculated. The sum of numbers for all cell types was computed, and compared to the total white blood cell count; this yielded the number (and proportion) of cells unaccounted for by flow cytometry for the particular sample. Finally, the subset of the data representing the cell propor-
3.8 Data analysis overview

A proportions matrix similar to the proportions matrix yielded by deconvolution was exported as a comma-delimited text file and used for further analysis. Only samples where the sum of all proportions represented at least 80% of all cells were included. This aims to exclude noise introduced by additional cell types not measured by flow cytometry but which may influence the expression profiles derived from whole blood.

Deconvolution using the method described above was applied to the full BERRY test set. Cell types identified by deconvolution included CD4 T cells (resting and activated), CD8 T cells (resting and activated), B cells (5 subtypes), plasma cells, monocytes (resting and activated), NK cells (resting and activated), dendritic cells (resting and activated) and neutrophils.

In the analysis step, the two proportions matrices obtained from flow cytometry and deconvolution, using the same samples and containing cell subsets with identical labels, were used. Figure 3.10 shows how the cell subsets from the two approaches were related to each other to study a unified subset of common cell types.

The function `profplot`, implemented in the `CellMix` package, was used to plot the combined proportions matrices. For each cell type, the regression coefficient ($\alpha$), Spearman rho ($\rho$) and Pearson $R^2$ with 95% confidence interval were calculated. The results of the analysis are presented in Chapter 5.

3.8.7 Weighted gene co-expression network analysis (WGCNA)

The techniques described above for differential expression focus on finding individual genes that may be important in a particular biological context. However, as described in Chapter 1, the focus of systems approaches is to find explanations for behaviour and properties of complex systems, i.e. identification of functional units instead of individual parts. Modules of co-expressed genes may be related to emergent and other high-order phenomena of such complex systems, leading to improved understanding of the underlying biology [140]. This leads to a reduction in high-dimensional data, and allows for integration of multi-scale data.

An implementation of such an approach was first described in 2005 [140] and has been implemented as the package `WGCNA` in R [141].

The fundamental concept underlying the functionality of the `WGCNA` package is that genes are
Figure 3.10: **Approach for unification of cell type subsets** For the deconvolution cell subsets, those subsets actually detected by the deconvolution algorithm in the array data are marked in yellow.
expressed in groups called modules, and that such modules are identifiable in microarray expression data. The processes involved in detecting such modules and interpreting their function and importance relative to phenotypic or other information is described below.

The network is constructed by making use of interaction patterns in the raw expression data. Pearson correlation is used as the measure of describing such interaction, as it is intuitive, avoids overfitting, is computationally fast and is reproducible.

The first step in the analysis is therefore construction of the correlation matrix, containing the Pearson correlation coefficient for each pairwise interaction for all genes studied. For large gene lists this is a computationally intensive step, and the package provides some utilities to run the analysis on systems with relatively small amounts of RAM. For instance, a list of 8000 probes or genes generates a square matrix with sixteen million entries. The next step of the analysis transforms the correlation matrix into an adjacency matrix, where each entry represents the weighted connectivity between all pairwise gene interactions, by using the power function

\[ f(x) = x^\beta \] (3.4)

The parameter \( \beta \) is chosen in such a way that the resulting adjacency matrix exhibits scale-free topology, as many biological networks that have been studied experimentally exhibit scale-free topology [142, 143, 144, 145, 146, 147]. This is based on a previously described model fitting index \( R^2 \) [140]. For a perfectly scale-free network, this index is defined to equal one. Here, the parameter \( \beta \) is chosen as the smallest exponent of the above power function such that the model fitting index \( R^2 \geq 0.8 \). This is termed the soft-thresholding procedure. It preserves the continuous information of the co-expression network and yields robust results with regard to different threshold choices [148]. The result of this step is an adjacency matrix, where each entry corresponds to the strength of the interaction between the two genes. This adjacency matrix is then transformed into a topological overlap matrix (TOM) which enables rapid detection of modules. This matrix provides a similarity measure for each pairwise interaction reflecting their relative interconnectedness.

Detection of modules (i.e. groups of genes whose expression profiles are highly correlated across samples) is performed as the next step. First, the TOM-based dissimilarity matrix (i.e. 1-TOM) is clustered using average linkage hierarchical clustering. Modules are then detected based on the
3 Materials and methods

clustering result, and this therefore depends on the choice of branch cut height, i.e. how distant two clusters have to be to belong to separate modules. The WGCNA package [141] implements a dynamic branch-cutting algorithm that allows for variation in branch cut height for different modules.

In order to interpret the modules, they are next related to external information. For the purposes of this study I have limited this to the contrast of interest, in order to identify modules important in that contrast. This allows for the discovery of “interesting” modules that may be analysed further. Module eigengene expression can be linked to clinical data. This is implemented by correlating eigengene expression values with clinical trait data. As clinical traits can be measured using continuous or categorical variables, this presents us with a problem. To address this, categorical clinical traits were first converted to ordinal data by encoding the individual categories using numerical values either arbitrarily (e.g. sex), or where appropriate on scale of increasing severity (e.g. healthy, latent TB, active TB). This allows for associating module eigengene expression with both continuous variables as well as categorical variables.

Finally, module preservation across different datasets can be studied, and the key drivers (hub genes) in the modules identified.

3.8.8 Pathway analysis

Exploratory pathway analysis was performed on several datasets, as proof of concept and for future integration in the analytic framework. Pathway analysis was performed using the R package gage [149], and pathways were visualised using the R package pathview [150]. Since these packages require R version 3.0.1 or higher, the code was not integrated with the other scripts, as the current analytic framework still runs in R version 2.15.2; code integration will require updating the R installation to R version 3.0.1 and all Bioconductor packages to version 12.

Gage implements pathway analysis using publicly available data from KEGG (Kyoto Encyclopedia of Genes and Genomes) [151, 152]. The method extends techniques used for gene set enrichment analysis and provides a more robust environment for finding differentially regulated pathways. It allows for unpaired samples, and case and control groups of different sizes. Gage outputs over-represented pathways ranked by q-values.
3.8 Data analysis overview

All pathways significant after multiple testing correction were visualised in using *pathview*, with log-fold change values mapped to individual pathway components, indicating whether this component was over- or underexpressed in the differential expression analysis.

Pathway analysis has several limitations. These have been reviewed recently in [153] and are summarised here. These limitations can be divided into two main groups: Annotation challenges and methodological challenges.

**Annotation challenges** Genomic annotation databases at present lack the resolution of new-generation technologies used to generate state-of-the-art genomic data. Alternative splicing may, for example, be detected using RNA-seq, but annotation databases may contain only information for a single transcript. Multiple transcripts of the same gene may have different, even opposing, biological functions, but this is not reflected in current databases. So, a query to KEGG will only indicate that a given gene shows altered transcript abundance, but does not identify the biological function of the specific splice variant. In addition, annotations contained in biological knowledge-bases may be incomplete or inaccurate. Many genes are annotated as “hypothetical” genes with no known function. Despite this, transcripts that map to these non-annotated genes appear to be important in biological contexts. Finally, information on specific pathways is usually based on a defined experimental system, but information regarding the experimental context is often missing from pathway resources.

**Methodological challenges** Benchmark data for comparing different pathway analysis methods is not available. Methods are usually compared using simulated data, which yields reproducible results but no indication of which method correctly identifies real biological phenomena. Also, pathways are not independent of each other, yet analytical methods usually make that very assumption. This results in a lack of a model that accounts for dependence among pathways at different time points. This limits our ability to observe changes at a pathway level in a biological system. This may in part be addressed by topology-based approaches.

**Conclusion** Pathway analysis, as implemented for this thesis has several limitations. Care must be taken when interpreting results.
3 Materials and methods

3.8.9 Other methods

Multiple additional methods to analyse microarray data exist. I chose the above to provide a broad view of the transcriptomic landscape in HIV-TB across many different contrasts. An additional method that might have been implemented is transcription factor binding site analysis. This would yield information regarding possible regulatory factors which might explain the observed variation, and would definitely improve the biological insight gained from the various gene lists. Future work may add this and other methods to the pipeline described above.

3.9 Code descriptions

In this section, the R code used for the analyses is described in outline. Full code listings with syntax highlighting are provided in the Appendix. Additional information on customised R functions used in the setup and analysis scripts can also be found in the appendix, together with the code for the functions.

Some of the principles that are used in the code, specifically the idea of setting up the workspace environment, as well as the general workflow for the the code in subsection A.9 are based on a tutorial for performing the analysis of the Berry et al data, reported in 2010 [82] in R instead of GeneSpring. I adapted the code as required; the utility functions mentioned on the website have been incorporated in the system setup script.

IMPI-MA refers to data generated for this PhD project; BERRY refers to expression data published in 2010 [82].

3.9.1 IMPI-MA: System setup

The code for this analysis is reproduced in full in Appendix Section A.6. This script is sourced by all other analysis scripts, and standardises the analysis and output environment for each analysis. The script performs the following actions:

1. Initialise the session by clearing the workspace and resetting the graphics devices
2. Configure local folders (project root, R-scripts folder, data folder)
3. Load utility functions

4. Set global parameters (e.g. image export formats, graphics parameters)

5. Load all libraries required by the analysis scripts

3.9.2 IMPI-MA: Data manager

The code for this analysis is reproduced in full in Appendix Section A.7. This script performs the following actions:

1. Source system setup script

2. Specify local data and output folders

3. Import expression (IMPI-MA and BERRY) and phenotype data using lumi

4. Make subsets of the datasets using CSC (compartment, subset, contrast) scheme

5. Export datasets to RData files; any subsequent use of the data is greatly sped up using RData

6. Import 393 and 86 transcript lists, and export for later use as RData files

3.9.3 RT-PCR data analysis as a prototype for more complex analyses

The code for this analysis is reproduced in full in Appendix Section A.8. The script performs the following actions:

1. Setup

2. Source setup script
   a) Make specific data and versioned output folders
   b) Define additional functions redgreen and interleave
3 Materials and methods

3. Read in RT-PCR and phenotype data and make single data structure
4. Calculate statistics for summary table of participant clinical characteristics
5. Select phenotype categories used for colour in plots
6. Make data matrix used in DE analysis and matrix of relevant phenotype information
7. Plot raw data: box-and-whisker plots, heatmap and correlation matrix plots
8. Filter data based on low expression values
9. Plot filtered data
10. Differential expression analysis for contrast: compartment
11. Plot and export results
12. Principal component analysis of all, filtered data and selected genes
13. Protein analysis
   a) Select protein data
   b) Boxplots and statistics for differential protein abundance
14. Additional analysis for HIV status and lowCD4 contrasts in blood and pericardial fluid

3.9.4 Relation of IMPI-MA data to previously published work

The code for this analysis is reproduced in full in Appendix Section A.5. The code for the differential expression analysis (Baylor Method) is based on concepts described in an online tutorial (found here: http://www.bigre.ulb.ac.be/courses/statistics_bioinformatics/practicals/microarrays_berry_2010/) and adapted for use in this framework, but the remainder is original work.

The script performs the following actions:

1. System setup
2. Source setup script
a) Create local data folders and versioned output folders

3. Load data and place in a list which is iterated over for each step of the analysis

4. Average normalisation (replicating a procedure originally carried out in BeadStudio)

5. Calculations for two-way table of clinical characteristics, comparing variables between datasets, and between different disease classes within datasets

6. Show raw data using box-and-whisker charts, multidimensional scaling and principal component analysis

7. Data preprocessing: threshold low expression values to minimum of 10 and perform median scaling; visualise data

8. Data filtering: detection filter and fold-change filter followed by selection of probes passing both filtering criteria; visualise data

9. Calculate differential expression statistics

10. Generate lists of differentially expressed probes

11. Plot results and export lists of probes for further analysis

The code for the main analysis will be discussed in the Chapter 8, as development of this code depended on specific technical results; the overall code structure was conceived only after the microarray data had become available, together with information regarding the numbers of samples in various classes with successful arrays.

3.9.5 An unbiased view of all IMPI-MA and BERRY data

The code for this analysis is reproduced in full in Appendix Section A.10. This simple linear script performs the following actions:

1. Source setup script

2. Create local data and versioned output folders

3. Load data (IMPI-MA and BERRY)
3 Materials and methods

4. Generate quality control plots, including density plots, boxplots and MA plots

5. Multidimensional scaling analysis of IMPI-MA data (all samples, blood and fluid)

3.10 Comparisons

Comparisons of gene lists were made by calculating the intersection of two or more gene lists, and visualising these using Venn diagrams in R; Code examples are shown in Appendix Section A.9.
Part II

General Results
4 RT-PCR on matched blood and fluid

Chapter summary

This chapter describes the results of an analysis of RT-PCR data from patients with tuberculous pericarditis, with particular focus on contrasting gene expression between the blood and pericardial fluid compartments. We show that significant differential expression of genes is detectable between compartments, and that this difference may be biologically informative.

Disclosure: The text is largely based on a manuscript in the early stages of preparation. All text was written by myself, but incorporates significant input from the senior author of the manuscript, Prof. Katalin A Wilkinson. For this reason, this chapter is written in the first person plural form. As described in Section 3.1, all data was generated by Kerryn Matthews and the IMPI Registry team.

4.1 General comments regarding the presentation of results

Throughout this thesis results of increased or decreased transcript abundance are reported, be it generated by RT-PCR or microarray. The terms “upregulation” and “downregulation” are frequently employed to refer to increased and decreased transcript abundance as a form of convenient shorthand. In reality, the observed changes are usually due to factors over and above changes in the regulation of expression, and the use of the terms up- or downregulation should be read in their (intended) wider sense.

Many images depicting clustered versions or PCA plots of pre-selected features are shown in the results. These heatmaps and plots essentially show only that the selected features classify the input data correctly. Future work will include experiments where the signatures obtained from the IMPI-MA data will be validated against unrelated datasets.
Interpretation of PCA plots can be rather subjective. As it stands, the current work does not provide more objective metrics of similarity of datasets based on visual inspection of the scatterplots. Future work will incorporate calculation of Euclidean distances between centroids and drawing ellipses. R packages that have these features exist.

## 4.2 Study subjects

27 patients with TB pericarditis (15 definite and 12 probable cases) were included in this analysis (Table 4.1). Ten were HIV-1 uninfected and 17 were HIV-1 infected, with a median CD4 count of 167.0 (98.0-343.0) cells/µl. There was no difference in proportions of definite or probable TB diagnosis, gender or age between HIV status groups, but the CD4 count was significantly lower in the HIV-1 infected patients (p=0.025). One HIV-1-infected participant was on antiretroviral therapy at the time of sampling; duration of antiretroviral therapy was not recorded.

<table>
<thead>
<tr>
<th>Comparison of clinical characteristics of patients with tuberculous pericarditis, stratified by HIV-status</th>
<th>All participants (N=27)</th>
<th>HIV-1 uninfected (N=10)</th>
<th>HIV-1 infected (N=17)</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (Median [IQR])</td>
<td>31.6 (28.7-41.3)</td>
<td>47.4 (28.9-68.2)</td>
<td>31.4 (28.9-35.8)</td>
<td>0.155</td>
</tr>
<tr>
<td>Gender</td>
<td>10:17</td>
<td>3:7</td>
<td>7:10</td>
<td>0.866</td>
</tr>
<tr>
<td>Male</td>
<td>10:7</td>
<td>6:3</td>
<td>7:10</td>
<td>0.866</td>
</tr>
<tr>
<td>TB status</td>
<td>0.397</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probable: Definite</td>
<td>12:15</td>
<td>6:4</td>
<td>6:11</td>
<td>0.397</td>
</tr>
<tr>
<td>CD4 count (cells/µl)</td>
<td>167.0 (96.0-343.0)</td>
<td>396.5 (190-418.5)</td>
<td>118 (89-266)</td>
<td>0.025</td>
</tr>
<tr>
<td>Concurrent steroids</td>
<td>Yes:No</td>
<td>18:8</td>
<td>5:4</td>
<td>13:4</td>
</tr>
<tr>
<td>Fluid ADA (U/L)</td>
<td>77.0 (54.5-91.5)</td>
<td>91.0 (65.0-137.0)</td>
<td>75.0 (52.0-86.0)</td>
<td>0.374</td>
</tr>
<tr>
<td>Fluid IFN-γ (pg/ml)</td>
<td>1531.0 (205.5-4212.0)</td>
<td>2084.25 (0.0-3993.0)</td>
<td>1308.0 (498.0-4512.0)</td>
<td>0.907</td>
</tr>
<tr>
<td>Number on ART (yes:no)</td>
<td>NA</td>
<td>NA</td>
<td>1:16</td>
<td>NA</td>
</tr>
</tbody>
</table>

## 4.3 Gene expression analysis

Overall expression values in the two compartments were similar (Figure 4.1).

As a first step in our analysis, we grouped the 42 genes studied into eight functional categories: Th1, Th2, TH17, genes associated with immunoregulation, neutrophil associated, growth factors, fibrosis associated and matrix metalloproteinase associated genes. This categorised gene expres-
4.3 Gene expression analysis

Figure 4.1: Overall expression values. Box-and-whisker plot showing the median, upper and lower quartile of raw ∆CT values for blood (red) and pericardial fluid samples (yellow).

Expression in blood and pericardial fluid is summarised in Figure 4.2. Gene-by-gene comparison (with Benjamini-Hochberg multiple testing correction) yielded 21 genes that are differentially expressed in blood compared to pericardial fluid following application of Benjamini-Hochberg multiple testing correction (Table 4.2).

Overall, differences between compartments for the expression of Th1, Th2 and Th17-associated genes were not evident, whereas genes associated with immunoregulation did show a difference. In addition, overall differences in expression levels of fibrosis-associated genes were evident, as well as some neutrophil-associated genes, growth factors and MMP genes, as well as their regulators.

As a second step, we employed cluster analysis in order to explore the data further. By combining the expression values of all samples and all genes (as described in methods) we generated a heatmap shown in Figure 4.3. While there was no differentiation between blood and fluid, we found that strikingly, most blood and pericardial fluid samples from individual patients clustered together. This appeared to be driven by highly correlated gene expression patterns between the two compartments, specific for each individual, thus obscuring differences of gene expression between the two compartments. Four clusters of genes can be seen based on very low, low, intermediate and high expression regardless of compartment. Many genes appear to be expressed in groups. This phenomenon of co-expression was further explored in these samples and is shown in the correlation matrices of blood and pericardial fluid samples in Figure 4.4. Gene co-expression patterns in blood were different from pericardial fluid, where pronounced co-expression of fibrosis-associated
Table 4.2: **Gene-by-gene comparison of expression between compartments.** All comparisons statistically significant (P ≤ 0.05) are highlighted in yellow.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Pval</th>
<th>Bonf</th>
<th>BH</th>
</tr>
</thead>
<tbody>
<tr>
<td>IL1B</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>IL2</td>
<td>0.18</td>
<td>1.00</td>
<td>0.30</td>
</tr>
<tr>
<td>IL6</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>IL18</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>IL24</td>
<td>0.65</td>
<td>1.00</td>
<td>0.81</td>
</tr>
<tr>
<td>TNF</td>
<td>0.17</td>
<td>1.00</td>
<td>0.29</td>
</tr>
<tr>
<td>IFNG</td>
<td>0.02</td>
<td>0.63</td>
<td>0.03</td>
</tr>
<tr>
<td>CXCL10</td>
<td>0.30</td>
<td>1.00</td>
<td>0.45</td>
</tr>
<tr>
<td>IL4</td>
<td>0.05</td>
<td>1.00</td>
<td>0.09</td>
</tr>
<tr>
<td>IL13</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>IL17A</td>
<td>0.94</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>IL22</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>IL23A</td>
<td>0.59</td>
<td>1.00</td>
<td>0.76</td>
</tr>
<tr>
<td>TGFB1</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>FOXP3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>IL10</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
</tr>
<tr>
<td>IL8</td>
<td>0.00</td>
<td>0.10</td>
<td>0.01</td>
</tr>
<tr>
<td>LCN2</td>
<td>0.00</td>
<td>0.05</td>
<td>0.00</td>
</tr>
<tr>
<td>CAMP</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Defb4a</td>
<td>0.48</td>
<td>1.00</td>
<td>0.65</td>
</tr>
<tr>
<td>Elane</td>
<td>0.07</td>
<td>1.00</td>
<td>0.12</td>
</tr>
<tr>
<td>Egf</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>Ctgf</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Csf2</td>
<td>0.54</td>
<td>1.00</td>
<td>0.71</td>
</tr>
<tr>
<td>Il9</td>
<td>0.76</td>
<td>1.00</td>
<td>0.89</td>
</tr>
<tr>
<td>Col1a1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Col1a2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Col4a1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Col4a2</td>
<td>0.04</td>
<td>1.00</td>
<td>0.07</td>
</tr>
<tr>
<td>Arg1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Sparc</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mmp1</td>
<td>0.22</td>
<td>1.00</td>
<td>0.35</td>
</tr>
<tr>
<td>Mmp2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mmp3</td>
<td>0.34</td>
<td>1.00</td>
<td>0.49</td>
</tr>
<tr>
<td>Mmp7</td>
<td>0.73</td>
<td>1.00</td>
<td>0.87</td>
</tr>
<tr>
<td>Mmp8</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mmp9</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mmp10</td>
<td>0.85</td>
<td>1.00</td>
<td>0.96</td>
</tr>
<tr>
<td>Mmp11</td>
<td>0.95</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Timp1</td>
<td>0.00</td>
<td>0.09</td>
<td>0.00</td>
</tr>
<tr>
<td>Timp2</td>
<td>0.00</td>
<td>0.07</td>
<td>0.00</td>
</tr>
<tr>
<td>Timp3</td>
<td>0.48</td>
<td>1.00</td>
<td>0.65</td>
</tr>
</tbody>
</table>
4.3 Gene expression analysis

Figure 4.2: Expression values by compartment. Blood and pericardial fluid expression values for all 42 genes are shown, grouped into eight categories. Expression values are \( \log_2 \)-fold change relative to \( \beta - actin \), and as such all values are negative. More negative values correspond to lower expression.
and neutrophil-associated genes could be seen, as well as co-expression of some pro- and anti-inflammatory genes. The expression of several genes in fluid also exhibited negative correlation (e.g. IL-9 is negatively correlated with IL-2, IL-4, ELANE and MMP1)

Figure 4.3: **Calibrated heatmap of all gene expression values.** This heatmap shows all expression values. Samples and genes are clustered by hierarchical clustering with average linkage. Strikingly, most blood and pericardial fluid samples from the same individual cluster together.

Given the very low expression levels of some genes, we next applied a non-specific filter to the data, which removed genes with delta-CT values greater than 38 in 5% or more of the samples (see Section 3.1.3). This step left 22 genes that separate blood and pericardial fluid to a large extent, with three samples misclassified (Figure 4.5). This result indicates that overall gene expression in the two compartments is very different and confirmed our previous suggestion that genes with low levels of expression drive the patient-specific clustering, shown earlier.

Following computation of differential gene expression with multiple testing correction using a linear models approach (see Section 3.1.3), 17 genes were found to be differentially expressed between the blood and pericardial fluid compartments. These genes are summarised in Table 4.3 and illustrated on a volcano plot in Figure 9.2. The calibrated heatmap shown in Figure 4.7 illustrates that the selected genes collectively cluster the samples into two separate compartments of
4.3 Gene expression analysis

Figure 4.4: Correlation matrices for 42 genes. Colours indicate the value of $R^2$ for each pairwise correlation of gene expression (raw $\Delta CT$ values) in the range -0.5 to 1; red = positive correlation, blue = negative correlation.

Figure 4.5: Calibrated heatmap of all filtered expression values. This heatmap shows expression values of filtered genes. Samples and genes are clustered by hierarchical clustering with average linkage. At this stage, blood and pericardial fluid form two separate clusters.
blood and pericardial fluid.

Figure 4.6: Volcano plot of differentially expressed genes.

Table 4.3: Differentially expressed genes. The logFC value represents increased or decreased transcript abundance in pericardial fluid relative to blood. A positive value indicates higher in fluid, and a negative value indicates higher in blood.

<table>
<thead>
<tr>
<th>Differentially expressed genes</th>
<th>ID</th>
<th>logFC</th>
<th>P.Value</th>
<th>adj.P.Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL1A1</td>
<td>2.86</td>
<td>6.51E-23</td>
<td>1.43E-21</td>
<td></td>
</tr>
<tr>
<td>COL1A2</td>
<td>1.80</td>
<td>3.41E-14</td>
<td>3.75E-13</td>
<td></td>
</tr>
<tr>
<td>MMP2</td>
<td>1.37</td>
<td>1.29E-12</td>
<td>9.46E-12</td>
<td></td>
</tr>
<tr>
<td>COL4A1</td>
<td>1.26</td>
<td>4.71E-09</td>
<td>1.15E-08</td>
<td></td>
</tr>
<tr>
<td>CTGF</td>
<td>1.19</td>
<td>3.40E-11</td>
<td>1.25E-10</td>
<td></td>
</tr>
<tr>
<td>IL6</td>
<td>1.15</td>
<td>1.89E-11</td>
<td>9.30E-11</td>
<td></td>
</tr>
<tr>
<td>IL8</td>
<td>0.85</td>
<td>1.49E-03</td>
<td>2.53E-03</td>
<td></td>
</tr>
<tr>
<td>FOXP3</td>
<td>0.77</td>
<td>8.07E-05</td>
<td>1.61E-04</td>
<td></td>
</tr>
<tr>
<td>IL10</td>
<td>0.53</td>
<td>6.67E-03</td>
<td>8.63E-03</td>
<td></td>
</tr>
<tr>
<td>TIMP1</td>
<td>0.29</td>
<td>1.87E-03</td>
<td>2.94E-03</td>
<td></td>
</tr>
<tr>
<td>TIMP2</td>
<td>-0.23</td>
<td>5.50E-03</td>
<td>7.56E-03</td>
<td></td>
</tr>
<tr>
<td>TGF81</td>
<td>-0.25</td>
<td>4.08E-04</td>
<td>7.48E-04</td>
<td></td>
</tr>
<tr>
<td>IL1B</td>
<td>-0.52</td>
<td>4.44E-03</td>
<td>6.51E-03</td>
<td></td>
</tr>
<tr>
<td>SPARC</td>
<td>-0.95</td>
<td>5.32E-09</td>
<td>1.17E-08</td>
<td></td>
</tr>
<tr>
<td>MMP9</td>
<td>-1.36</td>
<td>4.38E-11</td>
<td>1.32E-10</td>
<td></td>
</tr>
<tr>
<td>CAMP</td>
<td>-1.57</td>
<td>4.80E-11</td>
<td>1.32E-10</td>
<td></td>
</tr>
<tr>
<td>ARG1</td>
<td>-1.66</td>
<td>2.11E-11</td>
<td>9.30E-11</td>
<td></td>
</tr>
</tbody>
</table>

Finally, we performed principal components analysis of the data using the three gene sets (all 42 genes, subset of 22 ‘filtered genes’ obtained after non-specific filtering, and the subset of 17 ‘selected genes’ obtained after differential gene expression analysis). The results are represented in Figure 4.8 (panels A, B, and C). While there was no separation of blood and fluid on the first two principal components using ‘all genes’ (panel A), the 22 ‘filtered genes’ (panel B) did indicate a
Figure 4.7: **Calibrated heatmap of differentially expressed genes.** This heatmap shows expression values of genes that are differentially expressed. Samples and genes are clustered by hierarchical clustering with average linkage. As expected, blood and pericardial fluid form two separate clusters.
separation of the two compartments. Finally, the first principal component of the 17 differentially expressed ‘selected genes’ (panel C) explained 60% of the observed variance.

![Figure 4.8: Principal component analysis.](image)

Figure 4.8: **Principal component analysis.** The first panel shows the results of principal component analysis of the whole dataset. There is no principal component that clearly allows for separation of blood and pericardial fluid samples. The second panel shows the same analysis applied to the filtered genes. Now, blood and fluid do separate, but not in a single dimension. Finally, PCA applied to only the differentially expressed genes shows clear separation of blood and pericardial fluid. Colours: red = blood; blue = pericardial fluid.

### 4.4 Discussion

In this Chapter I demonstrate that significant differential abundance of gene transcripts can be detected in tuberculous pericarditis when contrasting blood and pericardial fluid. Transcripts associated with a Th1 response and fibrosis are more abundant in the pericardial fluid compartment, suggesting that larger-scale transcriptomic studies contrasting the blood transcriptome with that of the site of disease may be informative regarding pathogenetic mechanisms of tuberculosis.

Specifically, transcripts associated with fibrosis (COL1A1, COL1A2 and COL4A1) are more abundant, providing evidence at mRNA level of a profibrotic environment even without pericardial constriction present and suggesting that collagen synthesis takes place much in advance of clinical features of constrictive pericarditis. Increase of procollagen transcripts is accompanied by increase of transcripts of the growth factor CTGF. CTGF protein is secreted by vascular endothelial cells and promotes proliferation of chondrocytes, mediates cell adhesion in many cell types and enhances fibroblast growth factor-induced DNA synthesis.

ARG1 and SPARC, two negative regulators of fibrosis are less abundant at transcript level, again
underscoring the overall pro-fibrotic environment in pericardial fluid.

In addition to the pro-fibrotic environment in pericardial fluid there is some support for a Th1-mediated inflammatory response in the pericardial compartment, with increase of IL-6 transcripts in pericardial fluid. Other Th1-related markers are not increased, though, so this increase should be interpreted with caution. The increase of IL-10 and FOXP3 transcripts may indicate a concurrent anti-inflammatory response. Interpretation of pro-and anti-inflammatory responses in tuberculosis is not without difficulty, as it is unclear whether Th1 responses are protective, injurious, or both, and whether anti-inflammatory responses are protective or detrimental in the context of inflammation.

Neutrophil-associated genes IL-8 and CAMP are up-and downregulated in pericardial fluid, respectively; this does not provide a clear indication of the role of neutrophil-specific activity in tuberculous pericarditis.

Finally, matrix metallocproteinases and their negative regulators appear to be dysregulated, with upregulation of MMP2 and TIMP1 in pericardial fluid, and concurrent downregulation of MMP9 and TIMP1.

This proof-of-concept analysis demonstrates that informative and biologically plausible results may be generated from analysis of matched blood and site-of-disease gene expression data.
5 Technical results

Chapter summary

In this chapter I provide an overview of all patients included in IMPI-MA, and develop the idea of ordering the study subjects along the vertices of multi-dimensional hypercubes in order to define contrasts and contexts for comparisons. This serves as the underpinning for a framework of analysing heterogeneous microarray data.

5.1 Included patients and samples

Figure 5.1 outlines the recruitment of study subjects into the IMPI-MA study, and the availability of RNA samples for these subjects. As stated previously, recruitment into the study consisted of three components: retrospective controls (EU Study), retrospective cases (recruited by Kerryn Matthews for her PhD work) and prospective cases (recruited by myself for this study). A total of 140 study subjects were recruited to contribute RNA samples for microarray analysis, and these individuals contributed a total of 220 blood and/or pericardial fluid samples for quality assessment and microarrays. These samples were shipped to the NIMR in London. At the NIMR, the RNA was further processed and microarrays performed as described in Sections 3.6 and 3.7. Following assessment of RNA yield and microarray quality metrics at the NIMR in London (see Figure 5.2), 39 samples were excluded, leaving 181 samples for analysis. Table 5.1 lists the reasons for exclusion of the 39 samples.

In order to plan the analysis, the 181 available microarrays were linked to phenotype information and stratified by three main criteria, based on the main hypotheses of the study: compartment (blood and pericardial fluid), HIV status (HIV+ and HIV-) and TB status (Healthy, LTBI, PTB and TB-PC).
Figure 5.1: Study subject recruitment. The first three branches of the flowchart consist of the subjects from the EU Study whose whole-blood derived RNA samples were included as controls (N=20 × 3. The TB-PC arm of the study consists of retrospective (N=40 matched blood and pericardial fluid) as well as prospective (N=40 matched blood and pericardial fluid) samples. The retrospective controls were selected on the basis of sample and data availability, the retrospective cases on the basis of sample and data availability as well as RNA quality (hence the large drop in numbers following screening), and the prospective samples on the basis of availability of matched blood and pericardial fluid samples. Tygerberg index refers to a score based on clinical (fever, weight loss, night sweats) and laboratory (serum globulin, blood leukocyte count) parameters [111].

Table 5.1: Reasons for exclusion of RNA samples

<table>
<thead>
<tr>
<th></th>
<th>Low yield (&lt;1250ng)</th>
<th>Low RIN (&lt;6)</th>
<th>Array QC fail</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB-PC blood</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>TB-PC fluid</td>
<td>21</td>
<td>12</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>PTB</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>LTBI</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>“Healthy”</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>34</strong></td>
<td><strong>15</strong></td>
<td><strong>1</strong></td>
<td><strong>39</strong></td>
</tr>
</tbody>
</table>
5.2 Sample distribution along the vertices of a multi-dimensional hypercube

As is evident from Figure 5.3, the choice of possible contrasts and comparisons for analysis of this heterogenous and complex dataset is not immediately clear. The main problem lies with the fact that any hierarchical ordering of the sample subsets, as shown in the figure is completely arbitrary. Unsupervised analysis using principal component analysis may suggest an ordering, but even then the biological relevance of such a hierarchy is unclear. There is another option for representing the data, first discussed in Subsection 2.6.2, which allows for the fact that each phenotypic variable is independent of all others, and thus lies orthogonally oriented to them in phenotype space. Representing this graphically for three dimensions is trivial, as shown in Figure 2.3 for the three phenotype variables linked to the three main study hypotheses. Figure 5.4 (left panel) now shows this same 3-cube, while the right panel shows each of the cube’s six faces, together with their respective sample numbers. The edges joining the four vertices of each face can now be interpreted as contrasts or comparisons (see Figure 2.2), where each contrast has a

Figure 5.2: RNA quality metrics. The top panel shows the RNA yield (ng, in log_{10} scale) for each of the 220 samples, and the bottom panel the corresponding RNA quality (RIN, linear scale). Green bars correspond to EU study samples; red (blood) and yellow (pericardial fluid) bars correspond to retrospective (first pair) and prospective (second pair) TB-PC samples. Blue lines indicate pre-specified cutoff values used for including samples in further analysis.

The stratification of the 181 samples based on these criteria is shown in Figure 5.3.
Figure 5.3: **Sample classes.** This figure illustrates the breakdown of sample numbers when stratified by three criteria (Compartment, HIV status and TB status). It further shows that subclasses may be recombined in different ways to yield additional groupings of potential interest for analysis.
5.2 Sample distribution along the vertices of a multi-dimensional hypercube

Hamming distance [154] of 1, i.e. the phenotypes between the two vertices for each contrast differ by exactly one character, with all others equal.

Figure 5.4: Three-dimensional sample phenotype space. The left panel shows the three-dimensional phenotype space relevant to the main study hypotheses (A) and the three orthogonal contrasts (B). The right panel (C) shows each face of the cube, with each vertex labeled by sample phenotype and number of available samples for that class. The edges represent potential contrasts or comparisons. Solid lines represent contrasts included in this study, and dotted lines represent contrasts excluded from the analysis due to insufficient numbers. Abbreviations: B blood; F pericardial fluid; A active tuberculosis; N not active tuberculosis; + HIV-1 infected; - HIV-1 uninfected; m matched samples exist.

This idea can of course be extended into higher dimensions, where one additional dimension is required for each additional phenotype variable to be used as a contrast. As there are a number of interesting subgroups of phenotypes in the dataset that may shed additional light on HIV-TB, three additional dimensions were explored: effect of TB site (pulmonary vs pericardial TB), early TB (latent tuberculosis vs healthy) and haemodynamic phenotype in TB-PC (effusive non-constrictive vs effusive-constrictive). Figure 5.5 illustrates the resulting higher-dimensional 3-cubes, and Figure 5.6 illustrates how the additional dimensions may be mapped to a three-dimensional representation of a six-dimensional hypercube (hexeract). This structure contains 160 cubic cells, the four which are of interest are shown. To understand this more easily, one should consider the original three-dimensional phenotype space, cube A in Figure 5.5. The left face of cube A consists of all samples mapping to the “not active TB” phenotype. This phenotype may be divided into

---

119
two groups ("Healthy" and "LTBI") based on TST and IGRA results. This division “extends” the two-dimensional left face of cube A into a three-dimensional cube of its own (cube B, Figure 5.5). The six faces and associated sample numbers are shown in Figure 5.7. In contrast, the right face of cube A consists of all samples mapping to the “active TB” phenotype, which may be divided into two groups based on clinical presentation ("Pulmonary TB" and “Tuberculous pericarditis”), again “extending” the two-dimensional right face of cube A into cube C (Figure 5.5 and Figure 5.8). Finally, the right face of cube C ("Tuberculous pericarditis") may yet again be “extended” into cube D based on haemodynamic phenotype ("effusive non-constrictive pericarditis" and “effusive-constrictive pericarditis”), shown in Figure 5.5 and Figure 5.9.

Figure 5.5: Extension of three-dimensional cube to six dimensions. Cube A is the cube originally discussed; and will serve as the reference point.

This concept of adding dimensions to a phenotype hypercube for each measured variable may be further generalised to very high dimensions. The full description of the “system state” of a complex system such as a human being requires a very large number of dimensions. This has important implications for sampling. When drawing a sample from the study population, one ideally requires a sample where samples for each contrast are present in equal numbers in each of the other contrasts, in order to avoid bias and confounding.

Such a sampling strategy would then result in a balanced dataset with (1) similar numbers of samples on all vertices of the n-dimensional hypercube, and (2) similar distributions for values of
5.2 Sample distribution along the vertices of a multi-dimensional hypercube

Figure 5.6: Six dimensional hypercube. The four cubes shown in Figure 5.5 may be mapped to vertices in six-dimensional phenotype space, shown here as a three-dimensional “shadow” of the six-dimensional space.

Figure 5.7: Three-dimensional phenotype space for LTBI.
5 Technical results

Figure 5.8: Three-dimensional phenotype space for TB-site.

Figure 5.9: Three-dimensional phenotype space for HD phenotype.
5.3 Validation of deconvolution approach: proof of concept data

As pointed out in section 3.8.6, an attempt at validating the deconvolution method using flow cytometry data was made. Figures 5.10 and 5.11 show the results.

The barplots indicate an important feature of the data. All proportions obtained by deconvolution sum to one (by design) whereas the proportions obtained by flow cytometry do not.

Overall, the two methods correlate reasonably well ($R^2=0.77$). Some cell types (CD8 T cells, B cells, monocytes) correlate well, a second group (NK cells) less convincingly, and a third group (CD4 T cells, neutrophils and dendritic cells) correlates poorly. With the data available, the reasons for this are not clear. Dendritic cells and NK cells are observed at much higher proportions in the deconvolution data, whereas CD4 T cells and neutrophils are observed at higher proportions in the flow cytometry data.

The above results suggest systematic effects for these phenomena. A few potential reasons will be discussed below, but I should emphasise that this analysis should not be considered definitive. Firstly, neither of the proportion matrices contain “pure” proportions of single cell types but rather “composite” cell types obtained by adding proportions of different cell types together, as shown in Figure 3.10. Therefore the proportion signals do not necessarily represent the exact same cell populations. Secondly, in the deconvolution method used, the basis matrix based on the Abbas paper was generated on the Affymetrix platform, and mapping of transcript IDs (via Entrez IDs) to...
5 Technical results

Figure 5.10: **Barplots of proportions in 11 samples**  While the proportions in the top panel all sum to one (by design), the proportions in the bottom plot do not. Each sample therefore has an additional category of unassigned cells, which belong to one of the measured categories, or to a different category (e.g. basophils, eosinophils)
5.3 Validation of deconvolution approach: proof of concept data

Figure 5.11: Correlations of cell proportions The CellMix function profplot was used to generate this plot. Prior to calculation of the correlations, the flow cytometry proportions matrix was rescaled to one, which mirrors the deconvolution matrix. Different cell populations are indicated by different colours. Clear systematic effects are visible for NK cells, dendritic cells, neutrophils and CD4 T cells. The values for the overall plot and the individual cell types represent the regression coefficient ($\alpha$), Spearman rho ($\rho$) and Pearson $R^2$ with 95% confidence interval.
the Illumina data recovered only a part of the original basis matrix (324 features of a total of 515, i.e. 62.9%). This incomplete mapping may also influence the determination of cell proportions using deconvolution. Finally, log₂-transforming and quantile normalising the expression data may also skew the reported proportions; indeed it has been reported that using non-transformed, non-normalised data achieves more accurate results (R. Gaujoux, personal communication).

In conclusion, the deconvolution method is promising, but requires further validation, ideally by generating a bespoke basis matrix based on locally generated flow cytometry data and single cell type expression profiles using Illumina HT12v4 arrays, followed by prospectively collecting samples from a diverse range of volunteers and subjecting these samples to whole-blood flow cytometry and Illumina microarray with deconvolution.
6 An unbiased view of all array data

Chapter summary

In this chapter I provide an overview of all IMPI-MA array data, for the purpose of defining overt patterns in the data and to identify potential batch effects caused by technical factors. I compare some aspects of this dataset to three datasets published previously[82], in order to argue that the IMPI-MA dataset is similar to this previously published data.

6.1 Data

6.1.1 IMPI-MA

All data was generated as described in Section 3.7. The data consists of summarised, background-corrected, non-normalised probe-level data for all 181 arrays. The term “IMPI-MA” will be used to refer to this data.

6.1.2 BERRY

The data consist of the original summarised probe-level data used for the analysis of the Berry et al paper [82], specifically the training, test and validation sets. In personal correspondence with the first author of that paper I was able to obtain the raw, non-normalised probe-level data; this was necessary as the publicly available version of the data includes some preprocessing steps in the form of average normalisation (see Section 3.9.4). The term “BERRY” will be used to refer to this data. Two BERRY subsets were included: the “training set” consisted of individuals with active tuberculosis, LTBI and individuals without evidence of prior immune sensitisation; the same groups were available in the IMPI-MA data. The BERRY validation set only included individuals
with active tuberculosis or LTBI, but the study participants were recruited in South Africa and therefore possibly more closely matched to the IMPI-MA data.

6.2 Quality control plots

Figure 6.1 shows plots of the probability density function for six groups of arrays. This is a quick way to visually assess the similarity of expression profiles between arrays. As is clear from the plots, the overall shape of the probability density function (PDF) is the same for all arrays in each group but differs to some extent between groups of arrays. These differences are likely due to technical factors rather than biology, and indicate the need for a normalisation step. These plots highlight differences in array intensities for low to middle ranges of expression. In none of the plots are individual arrays visible that vary dramatically from the rest.

Figure 6.2 shows plots of the cumulative density function for six groups of arrays. Similarly to the PDF plots, these plots allow rapid assessment of all arrays for outliers in terms of overall array intensities. In contrast to the PDF plots, the CDF plots allow for better visualisation of intensity differences between arrays at the high and middle ranges of expression. Again, the need for normalisation becomes evident by inspection of the plots.

After reviewing the PDF and CDF plots we concluded that no samples needed to be excluded from analysis as no individual array differed dramatically from the rest. As the IMPI-MA and BERRY arrays looked significantly different, we concluded that direct comparison of these two studies might be problematic. Further comparison of the IMPI-MA and BERRY data is discussed in Chapter 7.

We now review some more aspects of the IMPI-MA data. In Figure 6.3 we show an example of pairwise correlation plots for five arrays from this dataset. These plots show that expression levels for individual probes for different arrays are highly correlated, an expected finding. A plot for all arrays was not done as the output would be illegible. As the data are not normalised, a high number of probes for each correlation pair differs by a factor of 2. Again, this expected.

MA plots are used to visualise intensity-dependent ratios of raw microarray data; microarrays typically show a bias here, with higher A resulting in higher M, i.e. the brighter the spot the more likely an observed difference between sample and control. The MA plot uses M as the y-axis and
6.2 Quality control plots

Figure 6.1: **PDF plots of all IMPI-MA arrays.** The first three plots show the superimposed probability density function (PDF) plots of all IMPI-MA data, and the blood and pericardial fluid subsets, separately. The fluid arrays in general appear to have similar expression distributions as blood, and important finding when considering batch effects. The last three plots show the PDFs for the BERRY training, test and validation sets, respectively. The x-axis shows the \( \log_2 \)-intensity values, and the y-axis the corresponding probability for that intensity.
Figure 6.2: **CDF plots of all IMPI-MA arrays.** The first three plots show the superimposed cumulative density function (CDF) plots of all IMPI-MA data, and the blood and pericardial fluid subsets, separately. The last three plots show the CDF plots for the BERRY training, test and validation sets, respectively.
6.2 Quality control plots

Figure 6.3: **Pairwise sample correlation.** Pairwise correlation of all probes on the first five arrays in the IMPI-MA dataset are shown.

A as the x-axis and gives another quick overview of the distribution of the data. Figure 6.4 shows M versus A (MA) plots for the first five arrays as an example. The M and A values are defined as follows:

\[ M_k = \log_2 \left( \frac{x_{ki}}{x_{kj}} \right) \]  

(6.1)

\[ A_k = \frac{1}{2} \log_2 (x_{ki} \cdot x_{kj}) \]  

(6.2)

The A values thus correspond to the mean of the two log values of two arrays \( i \) and \( j \) for all probes \( k \) on each array, while the M values correspond to the difference of the two log values. A Loess normalisation curve (shown in red) is fitted to the plot. In an ideal situation, this line should be straight. As it is not, it highlights the need for between-array normalisation.

Lastly, a box-and-whisker plot can highlight individual samples that differ significantly from all others. Such a plot is shown in Figure 6.5. The two ends of the box (“hinges”) are versions
An unbiased view of all array data

6.3 Sample relations

The previous subsection focused on various methods to assess overall intensity distributions in order to identify arrays that differed significantly from all others. This subsection now focuses on the discovery of systematic effects within the array data which may indicate that further downstream data analysis will be problematic. Relationships between the individual samples were assessed using a technique called metric multidimensional scaling (MDS). This methodology belongs to the class of methods called ordination, which aim to visualise information contained in

---

1 The hinges equal the quartiles for odd n (where n <- length(x)) and differ for even n. Whereas the quartiles only equal observations for n mod 4 == 1 (n = 1 mod 4), the hinges do so additionally for n mod 4 == 2 (n = 2 mod 4), and are in the middle of two observations otherwise.
6.3 Sample relations

Figure 6.5: **Box-and-whisker plot of all IMPI-MA arrays.** Blood arrays are coloured grey, and pericardial fluid arrays are coloured yellow. Each box shows the upper “hinge”, median and lower “hinge” for the log₂-transformed expression data.

Microarray data are of very high dimension (approx. 47,000 in the case of summarised probe-level Illumina data), therefore dimension reduction techniques like MDS are useful for showing how close or distant two samples are to each other. While the algorithm for performing MDS will not be discussed here, some general principles are worth mentioning. The data object consist of the matrix of expression values for all probes and all samples. A distance matrix for all entries is computed (usually using a Euclidean metric), and this distance matrix is used to compute the two-dimensional representation of the data by the use of singular value decomposition. The result is a list of coordinates for each sample, on principal component axes. These coordinates preserve the distance information of the original data matrix, and can be plotted in a scatterplot. In this analysis, MDS is performed on all genes that are somewhat variable between samples (defined as
PCA in contrast computes a covariance matrix of the input data and performs eigenvalue decomposition of that matrix, resulting in a list of principal components. The first principal component is defined as the one that accounts for as much variability in the data as possible (i.e. the normalised eigenvector with the largest associated eigenvalue of the sample covariance matrix). Each successive component is constrained by the requirement that it is orthogonal to the preceding one, and given this constraint explains as much of the remaining variance as possible. These two techniques are similar in several respects. While we only use MDS here to visualise data relationships, PCA will be used in later chapters of this thesis as well. In summary, MDS visualises dissimilarity between samples, and PCA identifies linearly uncorrelated components that explain most of the variance.

It is important to note that all of the analysis to follow was performed on non-normalised data. This was done to highlight potential technical bias. True biological differences between samples will be evaluated in later chapters, where that data has been normalised in order to deal with this technical bias.

Figure 6.6 shows the results of MDS analysis of all 181 microarrays. It is immediately clear that pericardial fluid arrays as a group behave very differently from blood arrays, and that the blood arrays do not show distinct subgroups based on TB status; indeed, there is complete overlap between the four categories (Healthy, LTBI, PTB and TB-PC).

MDS of blood samples is shown in Figure 6.7. Here, the samples are identified according to TB status and HIV Status. Again we see significant overlap of all groups, and no significant pattern suggesting another factor (e.g. technical) important in producing the observed differences.

Potential factors potentially biasing gene expression are sex [155] and the use of corticosteroids. Sex and corticosteroids may result in multiple gene expression changes that may bias differential expression analyses if the two groups contain unequal ratios of men to women. Figure 6.8 shows that no sex-specific or corticosteroid-specific signal appears to dominate the data.

In pericardial fluid, HIV status, and TB culture status (separating definite from probable TB-PC cases) may influence the overall pattern of gene expression. Figure 6.9 shows that neither TB culture status or HIV status dominate the overall expression data; differences that may exist will need to be sought by differential expression analysis.
6.3 Sample relations

Figure 6.6: MDS: All IMPI-MA data. Each point consists of a sample. The sample classes are shown in the figure legend. There is striking dissimilarity between blood and pericardial fluid arrays.

Figure 6.7: MDS: IMPI-MA blood samples (HIV-TB). Shown here are all blood samples, coloured by TB-status and HIV Status.
Figure 6.8: MDS: IMPI-MA blood samples (Sex/steroids). No clear separation appears between men (blue) and women (red), and between individuals who were taking corticosteroids (filled triangles) at the time of blood collection versus those who were not (hollow triangles). Squares represent samples where the steroid status was unclear.
Figure 6.9: **MDS: IMPI-MA pericardial fluid samples (HIV-TB)** No clear separation appears between HIV positive (filled squares) and HIV negative (hollow squares) fluid samples, or between samples that were culture positive for *Mycobacterium tuberculosis* (red) versus those who were not (purple).
6 An unbiased view of all array data

6.4 Discussion

The unbiased view of all IMPI-MA array data shows the data to be of high quality, and the need for normalisation between samples has been made evident. Multi-dimensional scaling views of the data clearly demonstrate that fluid samples as a whole behave differently from blood samples. Within the blood samples, I failed to detect gross technical bias. The individual sample classes do not separate on the two axes shown, despite the fact that the TB-PC and non-TB-PC samples were collected and processed at different times by different individuals. In addition, none of the parameters used to label the data showed any evidence of grouping by any of the studied parameters. This allowed me to proceed with confidence to subsequent analyses.
7 Relation of IMPI-MA data to previous studies of blood transcriptomics in tuberculosis

Chapter summary

In this chapter I demonstrate the results of an analysis pipeline in R that replicates the exact method used by Berry et al in the 2010 paper [82]. This is followed by application of the same method to a subset of the IMPI-MA data, and the results are compared.

7.1 Data

This analysis considers similarities in differential expression analyses of conceptually similar data-sets. Therefore, the data included in this Section consists of the BERRY training and validation sets, and the subset of IMPI-MA data generated from blood samples in HIV-1 uninfected individuals who either had no clinical or microbiological evidence for active tuberculosis (Healthy, LTBI), or who had culture-confirmed tuberculosis (either PTB or TB-PC). The BERRY validation set is of interest, as the samples were collected in South Africa from a population demographically very similar to the IMPI-MA cohort. After due consideration, the BERRY training sets was selected in addition to the BERRY validation set as it was used to derive the 393 probe signature for active tuberculosis, and was closer to the IMPI-MA data in terms of the specified phenotype classes. The BERRY training set consisted of three sample classes (healthy, LTBI and PTB), the BERRY validation set only two (LTBI and PTB) and the IMPI-MA subset four (healthy, LTBI, PTB and TB-PC). Despite this, it is reasonable to expect that differential expression profiles contrasting active and
not-active tuberculosis should be comparable.

### 7.2 Study subjects

Table 7.1 shows comparisons between and within the three datasets across four clinical variables: age, sex, TST size and ethnicity. The comparison between datasets for all classes shows significant differences for age and ethnicity, but the datasets are matched for proportions of active to not active TB and sex. For each of the subsets (healthy, LTBI and active TB) we find the following: The *healthy* subset is mismatched w.r.t. ethnicity, the LTBI subset w.r.t. age, TST size and ethnicity, and the active TB subset w.r.t. ethnicity and TB type.

Comparisons within datasets find that the following datasets are not matched with respect to the following variables are not matched: BERRY training set: ethnicity; BERRY validation set: age. These differences between and within datasets are likely to influence differential expression analysis to some extent, but this is unavoidable.

### 7.3 Analysis

The raw data, following average normalisation looks similar across all three datasets (Figure 7.1).

When viewed using multidimensional scaling, none of the raw datasets show complete separation along the contrast of interest. Both the BERRY datasets show some extent of separation along at least one axis in the plot, suggesting a significant influence of the contrast variable.

Following the raw data visualisation, the data was processed as originally described in the supplementary methods section of [82]; see Section 3.9.4 for details of implementation and links to the relevant code. Table 7.2 shows the effects of applying the two non-specific filters to the three median-scaled data sets. Strikingly, 2.9 times the number of probes are selected from the IMPI-MA dataset (which used HT12 v4 arrays) when compared to the BERRY training set.

As the basis for the two filters may be dependent on technology, this may not be a fair comparison. For instance, newer technology may result in expression for more probes becoming detectable, and improved dynamic range of new array imaging technology may result in improved fold-change estimates. It is interesting that the BERRY validation set has results that are inter-
### Table 7.1: Table of clinical characteristics across datasets and contrasts

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>BERRY training</th>
<th>BERRY validation</th>
<th>IMPI-MA</th>
<th>p value</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>med</td>
<td>0-18</td>
<td>18-30</td>
<td>30-55</td>
<td>55-75</td>
<td>75</td>
</tr>
<tr>
<td>Sex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethnicity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Abbreviations:
- **n** = number
- **p** = proportion
- **med** = median
- **iqr** = interquartile range

#### P-values given for comparisons between datasets (second-last column) as well as within datasets (last three rows), with the statistical test used for each comparison listed.

### Table 7.2: Effect of non-specific filtering

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Original number</th>
<th>Pass detection filter</th>
<th>Pass Fold change filter</th>
<th>Pass both filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPI-MA</td>
<td>47,231</td>
<td>19,027</td>
<td>5,316</td>
<td>5,316</td>
</tr>
<tr>
<td>BERRY validation</td>
<td>48,802</td>
<td>15,418</td>
<td>3,388</td>
<td>2,824</td>
</tr>
<tr>
<td>BERRY training</td>
<td>48,802</td>
<td>15,391</td>
<td>1,836</td>
<td>1,835</td>
</tr>
</tbody>
</table>
Figure 7.1: **Box-and-whisker plots of the raw expression data for the three datasets.** Colours indicate sample classes: green=healthy, blue=LTBI, orange=PTB and red=TB-PC. Overall expression patterns look similar between the three datasets, and no extreme outliers are obvious. Boxes indicate lower quartile, median and upper quartile, and whiskers the extremes of data.
Figure 7.2: Multidimensional scaling analysis. The three datasets are shown following multidimensional scaling of the raw data.
mediate to the other two datasets, and that overall the number of selected probes are closer to the BERRY training set value. The two datasets based on samples collected in South Africa appear to be more variable than the original BERRY training set; the reason for this is unclear.

Figure 7.3 shows heatmaps of the selected probes. The BERRY validation set is already well separated into active TB and LTBI clusters, whereas the other datasets still show high levels of misclassification. The heatmaps in question demonstrate that by only using non-specific filtering to remove non-informative probes, much variability remains in the data obscuring potential differences that may exist between probes belonging to different phenotypic classes. Essentially this shows that in an unsupervised analysis, the various phenotypic groups do not cluster together in the IMPI-MA dataset. This could be due to smaller sample size, or additional variability between samples not explained by disease phenotype. This could include for instance concurrent viral infections in otherwise healthy individuals.

Multidimensional scaling applied to the selected probes (i.e. the probes passing both filters) show separation into clusters that does not relate well to the clinical phenotype.

Differential expression analysis was performed using the Kruskal Wallis test (IMPI-MA, BERRY validation) for the contrast active TB vs. not active TB, and Kruskal Wallis ANOVA for BERRY training set, as in the original paper. All results were corrected for multiple testing using the Benjamini Hochberg procedure. Table 7.3 list the results. This comparison is problematic at multiple levels. First, the number of conditions used for the comparisons is not identical for the three analyses. The original BERRY training set used Kruskal Wallis ANOVA to distinguish between three conditions (healthy, LTBI and PTB). This yields 404 DE probes at an FDR of 0.01, a result that is very similar to the original paper. Two conditions were used for the two other sets, as the validation set did not include healthy controls, and the IMPI-MA subset was too small to yield significant results for three conditions. Also of note is the fact that very different levels of FDR were used for the three analyses. This was done to obtain a roughly similar number of DE probes, even though the proportion of false positives in each results set was likely to be different due to this.

Given these results, and the caveats that accompany them, we now focus on the probes. The differentially expressed probes are shown in heatmaps in Figure 7.5 and multidimensional scaling plots in Figure 7.6. Of interest is whether the selected probes classify their input data correctly. As can be seen in Figure 7.5, the respective lists of differentially expressed probes classify the input
7.3 Analysis

Figure 7.3: **Heatmaps of selected probes.** Each heatmap shows the expression values of the selected probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours: green: *healthy*, blue: *LTBI*, orange: *PPBS*, red: *TB-PC*. Plot titles reflect the number of included probes.
Figure 7.4: Multidimensional scaling analysis of selected probes.
data reasonably well. In Figure 7.5a we find that one sample (“healthy” phenotype) is misclassified in the IMPI-MA data; in case of the BERRY validation data, two samples are misclassified (one “active TB” and one “LTBI” phenotype). Finally, in the case of the BERRY training set, three samples (one each of “healthy”, “LTBI” and “active TB”) are misclassified.

Table 7.3: Results of differential expression analysis

<table>
<thead>
<tr>
<th>Dataset</th>
<th>N conditions</th>
<th>FDR</th>
<th>N DE probes</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPI-MA</td>
<td>2</td>
<td>0.08</td>
<td>394</td>
</tr>
<tr>
<td>BERRY validation</td>
<td>2</td>
<td>0.0001</td>
<td>484</td>
</tr>
<tr>
<td>BERRY training</td>
<td>3</td>
<td>0.01</td>
<td>404</td>
</tr>
</tbody>
</table>

7.4 Comparisons

The main objective for the Berry et al analysis was to derive a predictive signature, whereas the main objective for the analysis in this chapter is to assess, given independent input data, how robustly does the method employed in the Berry et al analysis produce a characteristic signature, and how do signatures produced overlap with the original 393 signature? We can regard this approach as repeating the training set analysis three times, and comparing the results.

Firstly, how does the implementation of the Berry et al analysis method in R compare to the original results? Figure 7.7 shows a two-way Venn diagram demonstrating the overlap. One would expect perfect overlap if the method is indeed identical. We see that all probes from the 393 signature are reproduced by the R implementation of the method. An additional eleven probes are also called significant; this may be due to rounding errors due to machine precision settings used on two different systems; this difference may be caused by inclusion of probes at either of the non-specific filters or at the time of DE analysis using Kruskal Wallis ANOVA. In any event, the R implementation of the gene selection procedure reproducibly recreates the 393 probe list with a few additional probes.

The second question we now address is how do the probes selected from the IMPI-MA data overlap with the original 393 signature. Figure 7.8 shows this overlap. Interpretation of this extent of overlap requires some context before we proceed. In the Gedankenexperiment (“thought experiment”) that follows, we consider two sets A and B of 30,000 unique elements, and ask the question:
Figure 7.5: **Heatmaps of differentially expressed probes.** Each heatmap shows the expression values of the selected probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.
Figure 7.6: Multidimensional scaling analysis of differentially expressed probes.
Figure 7.7: **Method validation.** Overlap of “393” signature with DE probes from “BERRY training set (“TRAIN”) as found using R. Each set is labelled on the circle boundary, and the venn counts are centred within each circle. Please note that the *oval sizes are not scaled* to the number of probes for the purpose of better legibility.

how probable is an overlap of size $n$ of two subsets of A and B, each of size $k$? Using *Mathematica*, I created two sets A and B, each consisting of 30,000 elements, randomly subset each by 400 elements, and calculated the overlap (strictly the size of the intersection of the two subsets $A_k \cap B_k$). This was repeated 20,000 times, and the resulting data charted as a histogram (Figure 7.9). I then fit three types of distribution to the data (a normal distribution, a kernel mixture distribution and a Pareto distribution (Figure 7.9)). Using this distribution, I then produced a probability plot for different sizes $n$; this shows that any $n>15$ is very unlikely, with a probability of $0.000655873$. An overlap size exceeding 100 is extremely unlikely to occur by chance ($\text{probability} = 2.59115 \times 10^{-176}$). I conclude that most probes in the overlapping region are present because of biological reasons. The non-overlapping probes may be false positives and false negatives, respectively, depending which dataset is used as reference.

The two datasets were generated using different versions of the array technology, consisted of different sample sizes, were drawn from different populations and were analysed using different values for false discovery rate. Despite this, a statistically highly significant overlap of differentially expressed probes gives me confidence that overall, the IMPI-MA data is comparable to the Berry et al data.

Finally, we show the overlap of the 393 probe set with the BERRY training set results using R, the BERRY validation set and the IMPI-MA data in Figure 7.10.

Despite the fact that this degree of overlap between three datasets is extremely unlikely to occur by chance, we find that 61 probes are present as differentially expressed probes in the BERRY
Figure 7.8: **Data similarity.** Overlap of 393 probe set and the DE probes selected from the IMPI-MA data using the same analytic method from a dataset that differs in many respects from the BERRY training set.

Figure 7.9: **Probability of probe overlap.**

Figure 7.10: **Overlap of all three datasets with the 393 probe list**
training set, validation set and IMPI-MA subset. I infer that these probes are particularly interesting from a biological point of view. Do these probes allow for clear separation of active TB and not active TB? To investigate this, I show in Figure 7.11 three heatmaps, each consisting of the 61 probes present in all result sets, and demonstrate that many samples are still correctly classified as active or not active TB. Of note is that assignment to either the active or not active TB cluster is now more frequently done incorrectly. In the original analysis, the samples of IMPI-MA, BERRY validation and BERRY training sets are misclassified in one, two, and three instances, respectively, while using the 61 set of overlapping probes results in two, five and eight samples misclassified. However, active and not-active TB samples still cluster together in each of the two main clusters.

While the 61 probe subset does not appear to improve the classification of the input data, further investigation of the actual probes that constitute the set will be required to determine their biological significance. For instance, the probes may capture a conserved feature of the immune response to *M. tuberculosis* detectable in blood that is independent of site or severity of disease.
Figure 7.11: **Heatmaps of 61 overlapping probes applied to three datasets.** Each heatmap shows the expression values of the selected probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.
8 An analytic framework for heterogenous microarray data

Chapter summary

In this chapter I develop the idea of the data in relation to a multi-dimensional hypercube further, and show how this was used to develop an analytic framework for the IMPI-MA data. Finally, the code structure for this analytic framework is described, and an overview of the output of the analysis is given.

8.1 The sample hypercube revisited: what kinds of questions relating to the three main hypotheses can we ask, and how can we answer these questions?

Having verified that the data are sound (Chapter 6) and comparable to previously published data (Chapter 7), we now turn our attention to the addressing the three main hypotheses stated earlier.

8.1.1 Questions

Given the three hypotheses, the questions we aim to ask are related to the detection of differential transcript abundance for three main contrasts: TB status, HIV status and compartment. Each of these questions may be posed in different contexts, e.g. differential gene expression (DE) in active TB vs. not active TB in HIV-1 infected and HIV-1 uninfected individuals, respectively. The results of DE experiments may then be compared between these contexts, allowing for assessment of the effect of HIV-1 infection on the transcriptional response to tuberculosis. Table 8.1 lists the
questions and contexts developed for the analysis of the IMPI-MA data, as well as various metadata variables that will be required to subset the main dataset, set up the contrasts and specify annotation variables, if different from the contrast variable. Figure 8.1 shows the questions (1-7) mapped to 3D phenotype space, represented as the hypercube graph discussed earlier.

Table 8.1: **Questions.** Seven questions addressed in IMPI-MA analysis, listing metadata relevant to the implementation in R.

<table>
<thead>
<tr>
<th>Question</th>
<th>1: Tuberculosis</th>
<th>2: TB site</th>
<th>3: LTBI</th>
<th>4: HD phenotype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contrast variable</strong></td>
<td>class3</td>
<td>class</td>
<td>class</td>
<td>ECP</td>
</tr>
<tr>
<td><strong>Contrast 1</strong></td>
<td>ActiveTB</td>
<td>PTB</td>
<td>Healthy</td>
<td>Eff TB-PC</td>
</tr>
<tr>
<td><strong>Contrast 2</strong></td>
<td>notActive TB</td>
<td>TB-PC</td>
<td>LTBI</td>
<td>EC TB-PC</td>
</tr>
<tr>
<td><strong>Colour variable</strong></td>
<td>class</td>
<td>class</td>
<td>class</td>
<td>ECP</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>B.Neg.AN</td>
<td>B.Pos.PTB_TBPC</td>
<td>B.Pos.HL</td>
<td>B.C.Pos.EffEC</td>
</tr>
<tr>
<td></td>
<td>B.Pos.AN</td>
<td>B.Neg.PTB_TBPC</td>
<td>B.Neg.HL</td>
<td>PF.C.Pos.EffEC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question</th>
<th>5: HIV (not active TB)</th>
<th>6: HIV (active TB)</th>
<th>7: Compartment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contrast variable</strong></td>
<td>HIV.Status</td>
<td>HIV.Status</td>
<td>Compartment</td>
</tr>
<tr>
<td><strong>Contrast 1</strong></td>
<td>Positive</td>
<td>Positive</td>
<td>Blood</td>
</tr>
<tr>
<td><strong>Contrast 2</strong></td>
<td>Negative</td>
<td>Negative</td>
<td>Fluid</td>
</tr>
<tr>
<td><strong>Colour variable</strong></td>
<td>HIV.Status</td>
<td>HIV.Status</td>
<td>Compartment</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>B.N.PosNeg</td>
<td>B.A.PosNeg</td>
<td>C.BPF (matched)</td>
</tr>
<tr>
<td></td>
<td>B.L.PosNeg</td>
<td>PF.A.PosNeg</td>
<td>C.Neg.matchedBPF</td>
</tr>
<tr>
<td></td>
<td>B.H.PosNeg</td>
<td>B.PTB.PosNeg</td>
<td>C.Pos.matchedBPF</td>
</tr>
<tr>
<td></td>
<td>B.TBPC.PosNeg</td>
<td>C.Pos.EFF.matchedBPF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C.Pos.EC.matchedBPF</td>
</tr>
</tbody>
</table>

Datasets are subsets of all IMPI-MA data, named in triads or tetrads using . as separator. Each triad (tetrad) specifies SET.SUBSET.(SUBSET).CONTRAST. Abbreviations for datasets: B=blood, PF=pericardial fluid, Neg=HIV-1 uninfected, Pos=HIV-1 infected, A=active TB, N=not active TB, L=LTBI, H=healthy, PTB=pulmonary tuberculosis, TBPC=tuberculous pericarditis, C=tuberculous pericarditis (cardiac), Eff=effusive pericarditis, EC=effusive-constrictive pericarditis. Contrast variable refers to the name of the variable in the clinical phenotype dataset used to define the contrast of interest. Colour variable refers to the name of the variable in the clinical phenotype dataset used to define the factor used for identifying subsets in graphics by colour.

### 8.1.2 Analyses

Each dataset listed in Table 8.1 was analysed by the three complementary approaches described in Methods 3.8.
8.1 The sample hypercube revisited: what kinds of questions relating to the three main hypotheses can we ask, and how can we answer these questions?

Figure 8.1: Sample hypercube. Questions listed in Table 8.1 mapped to edges of the six-dimensional phenotype space rendered in three dimensions. Yellow labels indicate the question number; labelled edges represent the contrasts examined; greyed-out edges and vertices represent contrasts and sample classes not available or inappropriate for analysis. Cubes: A Main cube (Tuberculosis, HIV-1 infection, Compartment) B Not active tuberculosis C Active tuberculosis D Tuberculous pericarditis Abbreviations: B blood; F pericardial fluid; A active tuberculosis; N not active tuberculosis; + HIV-1 infected; - HIV-1 uninfected; m matched samples exist
8.2 Code description

8.2.1 IMPI-MA: Data manager

The code for this analysis is reproduced in full in Appendix Section A.11. Using Table 8.1 as reference, the script is based on a similar script described earlier (A.7), but extracts all data subsets based on the vertices of the extended hypercube and the selected contrasts and contexts. This data is saved to RData files for later access.

8.2.2 IMPI-MA: Analysis

The code for this analysis is reproduced in full in Appendix Section A.12. Structurally, the code implements a for-loop, wherein each of the seven questions is addressed. Within each question, one or several datasets are each subjected to the same analytic workflow of differential expression, deconvolution and cell-specific differential expression, and weighted gene co-expression network analysis. Multiple mechanisms for dealing with code exceptions have been implemented. These deal with missing or incomplete data and insufficient sample size. In addition, memory management has been optimised, allowing the code to run with a remarkably small memory footprint. A single run of this script takes approximately three and a half hours to complete on a 2010 intel dual core i7 machine with 8GB of non-ECC RAM running Mac OSX 10.8.4.

An important feature of this code is that the output is written to disk using versioned folders, and the naming of individual output files is based on the question, analysis type, dataset and contrast and output type. This fully automated process generates output files where the file name contains a significant amount of metadata allowing for interpretation of the output without reference to external material.

Outer for-loop

The outer for-loop iterates over a list containing seven “questions”. Each question in turn is a list, containing all metadata required to perform the analyses and correctly label all output. The overall code structure is shown in Figure 8.2.
for (question in questions) (n=7)
for (dataset in datalist) (n=1-4)
preprocess
DE
DC / csDE
output
WGCNA
output
output
Clean up
Clean up & finish

Figure 8.2: Overall code structure for IMPI-MA main script
Generation of clinical characteristics tables  An important consideration for microarray analysis of clinical samples is the description of the clinical characteristics of the patients included in the particular analysis. This is required in order to identify potential confounding effects, and to provide confidence that the two groups identified by the contrast variable are similar, and differ only in the value of the contrast variable. This essentially means that the two groups to be contrasted have a Hamming distance of one. In addition, where an analysis of the same contrast is performed in more than one dataset (i.e. context), it is also useful to show how the different datasets vary according to clinical parameters.

Given the complexity of this task, the following phenotypic variables were chosen for the clinical characteristics tables: age, sex and CD4 count. Each clinical table compares these variables within datasets (across contrasts) and between datasets (across contexts).

The comparisons for each variable are computed and stored in temporary variables. The results of the various computations are then combined into a single table and exported to raw LaTeX.
Differential gene expression

The DE section of the script performs the following actions:

1. Show raw data (boxplots, MDS, PCA)
2. Preprocess the data: variance stabilising transform followed by quantile normalisation
3. Show pre-processed data
4. Apply filter to remove poor-quality probes
5. Show filtered probes
6. Perform differential expression analysis
7. Show DE results
8. Export the probe lists

Deconvolution and cell-specific differential expression

The deconvolution section of the script performs the following actions:

1. Perform deconvolution based on method in [100]
2. Show proportion matrix as stacked barplot
3. Show boxplots of cell-type proportions by condition
4. Calculate statistics to identify cell-types that differ in proportion by condition (includes multiple testing correction)
5. Perform cell-type specific differential expression analysis using csSAM [135] and the proportion matrices calculated in 1.
6. Generate top tables for each cell type and export

Weighted gene co-expression network analysis

The weighted gene co-expression network analysis performs the following actions:
8 An analytic framework for heterogenous microarray data

1. Setup and linking of clinical trait data to expression data
2. Block-wise network construction and module detection
3. Relate modules to external information and identify important genes
4. Output module gene lists and perform gene-ontology analysis
5. Visualise networks of module eigengenes in relation to clinical traits
6. Export module networks in Cytoscape format
7. Plot module heatmaps and eigengene expression
8. Plot boxplots of eigengene expression values by condition (contrast)
9. Calculate statistics to identify modules that differ in expression by condition (includes multiple testing correction)

8.3 Computational output

8.3.1 Output tree

All output is saved in a single directory named using the date and start time of the run in R that produced the output. Two folders are made within the output folder; one contains all images, and the other all text files.

8.3.2 Images

Images can be output in multiple raster or vector formats; in order to conserve disk space all images were output as pdf for the run that produced the output for Chapters 9 to 12. This may be changed by modifying parameters in the setup script. A single run outputs 2,182 images (at the time of writing). All image files are labeled automatically with four numbers and a text string in the format “question#”, “analysis#”, “dataset#”, “figure#” followed by text that specifies the question, analysis, dataset and type of output. This allows anyone to understand a particular output file simply by inspecting the filename. In addition, most plots contain some annotation on the image (usually the image title) that allows the viewer to understand what (s)he is looking at.
8.3.3 Session Output log

All output usually written to `stdout`, i.e. the R console under usual configurations of R, is captured in a text file entitled `sessionOutput.txt`. This text file captures all warning or error messages created while running the code, as well as the output of print statements and any output normally sent to `stdout`. This file serves two main purposes: (1) Debugging the script by catching informative error messages and (2) capturing all output not written to file.

The file produced in a single run of the main analysis script is 84 pages long when printed. Table 10.2 and other similar tables were produced based on output in this file. For such a comprehensive analysis, this type of output serves as a useful reference point when evaluating the overall results of an analysis run.

8.3.4 Spreadsheets

314 spreadsheet-like objects were produced during the course of the analysis. These objects are comma-delimited text files and can easily viewed in any spreadsheet programme. The following analytic steps produce these objects:

1. Differential expression analysis
2. Cell-type specific differential expression analysis (one sheet for each cell type)
3. Gene Ontology enrichment analysis, as produced during WGCNA analysis
4. Results of statistical calculations (e.g. evaluating statistical significance of of differential module regulation or differences in cell-type proportions between sample classes)
5. Results of gene significance and module membership calculations

8.3.5 Plain text files

1373 plain text files were produced in the course of the analysis. They are simpler in structure than the csv files, and contain the following classes of data:

1. Entrez IDs for all genes used in the WGCNA analysis
2. Entrez IDs for all genes in modules identified in the WGCNA analysis
3. Nodes and edges for all modules in weighted gene correlation networks for all analyses. These files can be imported into Cytoscape (a network visualisation and analysis tool) for further processing.

8.3.6 \LaTeX{} tables

Tabular data for display purposes was formatted in the script as \LaTeX{} tables, and written to tex files. These files could then be incorporated directly in documents like this thesis. Table 10.1 and other similar tables were produced in this way. A single run of the script at the time of writing produces 27 ready to use \LaTeX{} documents.

8.4 Results as Data: a new resource for generating and analysing HIV-TB related hypotheses

I have shown that the output of a complex analytic pipeline may yield very comprehensive output that requires further systematic evaluation in order to understand fully. In this way, one output described in this thesis is a large dataset consisting of multiple output files that as a whole addresses a range of questions. In this way, the results of running the analytic pipeline produces new classes of data that are amenable to further in-depth analysis.
Part III

IMPI-MA: Results in depth
9 Question 1: Tuberculosis

Chapter summary

Matthew Berry et al have previously published a transcriptomic signature for tuberculosis in HIV uninfected individuals [82], and others have since confirmed many properties of the signature that was found [86, 85]. Here I show the results of an analysis that refines and extends the work of these authors. I present the results analysis for the dataset of Question 1: Blood, HIV negative with the contrast activeTB vs not active TB as an example of the depth of the analytic framework and the associated output. As mentioned in Chapter 8, the entire analysis output is extremely comprehensive, and will not be reproduced in full for the purposes of this thesis. This chapter is divided into three parts, presenting results for differential expression analysis, deconvolution and cell-specific differential expression analysis, and weighted gene co-expression network analysis. Pathway analysis will be presented in the next Chapter.

9.1 Differential gene expression

In this section I present the results of differential expression analysis using quantile normalised data and a linear models approach. The contrast used was active TB vs. not active TB. Active TB includes PTB and TB-PC, while not active TB includes healthy and LTBI classes.

9.1.1 Data quality

Boxplots of the non-normalised data show that no sample differs significantly from all others (Figure 9.1) and the quantile normalised data distributions for each sample are identical. PCA analysis of the non-normalised as well as normalised data demonstrate no significant batch effects; there is
large overlap of the four groups displayed.

![Boxplot](image1)

![Boxplot](image2)

![PCA](image3)

![PCA](image4)

Figure 9.1: Data QC: not normalised (left) and normalised (right).

### 9.1.2 DE probes

Following normalisation and filtering of probes based on probe quality, differential expression analysis was performed using the R package `limma`, with false discovery rate set to 0.05. 369 probes were selected as differentially expressed. Table 9.1 lists the top 30 differentially expressed probes, ordered by the variable logFC (log₂-fold change). A positive value indicates overexpression in active tuberculosis, and a negative value underexpression in active tuberculosis.

The results of the differential expression analysis can be visualised using a volcano plot (Figure 9.2), which, for each probe, plots the log₂-fold change value against the log-odds for differential expression (B). We conclude from the plot that the majority of probes are unlikely to be differentially expressed, and have low logFC values. The significant probes are labeled by identifier in blue.
### Table 9.1: Top 30 probes as selected by limma

<table>
<thead>
<tr>
<th>ProbeID</th>
<th>TargetID</th>
<th>logFC</th>
<th>CI.025</th>
<th>CI.975</th>
<th>AveExpr</th>
<th>t</th>
<th>P.Value</th>
<th>adj.P.Val</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>4150270</td>
<td>ANKRD22</td>
<td>1.77</td>
<td>1.23</td>
<td>2.34</td>
<td>6.78</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.98</td>
</tr>
<tr>
<td>3696088</td>
<td>SIGDA7</td>
<td>1.75</td>
<td>1.23</td>
<td>2.34</td>
<td>6.91</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.18</td>
</tr>
<tr>
<td>130181</td>
<td>ANKRD22</td>
<td>1.56</td>
<td>1.06</td>
<td>2.07</td>
<td>8.21</td>
<td>0</td>
<td>0.00</td>
<td>0.01</td>
<td>4.39</td>
</tr>
<tr>
<td>6620259</td>
<td>FGFR1B</td>
<td>1.32</td>
<td>0.89</td>
<td>1.86</td>
<td>7.85</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>7.58</td>
</tr>
<tr>
<td>530369</td>
<td>MS4A6A</td>
<td>1.05</td>
<td>0.72</td>
<td>1.38</td>
<td>10.06</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.69</td>
</tr>
<tr>
<td>589766</td>
<td>CCR2</td>
<td>1.02</td>
<td>0.72</td>
<td>1.33</td>
<td>8.86</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.30</td>
</tr>
<tr>
<td>830168</td>
<td>CYPB</td>
<td>0.99</td>
<td>0.73</td>
<td>1.26</td>
<td>12.24</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>6.83</td>
</tr>
<tr>
<td>6606711</td>
<td>CCR2</td>
<td>0.98</td>
<td>0.68</td>
<td>1.32</td>
<td>8.72</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.17</td>
</tr>
<tr>
<td>450491</td>
<td>CASP1</td>
<td>0.89</td>
<td>0.62</td>
<td>1.25</td>
<td>9.68</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.20</td>
</tr>
<tr>
<td>6206536</td>
<td>DUSP5</td>
<td>0.88</td>
<td>0.59</td>
<td>1.16</td>
<td>9.84</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.30</td>
</tr>
<tr>
<td>7050382</td>
<td>CASP1</td>
<td>0.86</td>
<td>0.60</td>
<td>1.13</td>
<td>9.84</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.82</td>
</tr>
<tr>
<td>20452</td>
<td>TYMP</td>
<td>0.85</td>
<td>0.58</td>
<td>1.12</td>
<td>8.54</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>6.64</td>
</tr>
<tr>
<td>3570039</td>
<td>LOC92794</td>
<td>0.55</td>
<td>0.37</td>
<td>0.79</td>
<td>5.34</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>8.53</td>
</tr>
<tr>
<td>1770152</td>
<td>MS4A6A</td>
<td>0.74</td>
<td>0.51</td>
<td>0.97</td>
<td>10.12</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.93</td>
</tr>
<tr>
<td>830035</td>
<td>ATG1</td>
<td>0.72</td>
<td>0.49</td>
<td>0.94</td>
<td>9.27</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.88</td>
</tr>
<tr>
<td>23055</td>
<td>OCHS</td>
<td>0.63</td>
<td>0.44</td>
<td>0.83</td>
<td>7.64</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.99</td>
</tr>
<tr>
<td>4060022</td>
<td>CD86</td>
<td>0.58</td>
<td>0.40</td>
<td>0.76</td>
<td>9.98</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.97</td>
</tr>
<tr>
<td>2360739</td>
<td>BEAK3</td>
<td>0.54</td>
<td>0.37</td>
<td>0.72</td>
<td>7.84</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.33</td>
</tr>
<tr>
<td>7260933</td>
<td>YLR11</td>
<td>0.51</td>
<td>0.35</td>
<td>0.66</td>
<td>7.59</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.53</td>
</tr>
<tr>
<td>839480</td>
<td>B15</td>
<td>0.48</td>
<td>0.32</td>
<td>0.63</td>
<td>7.35</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.22</td>
</tr>
<tr>
<td>675602</td>
<td>SEST1</td>
<td>0.47</td>
<td>0.32</td>
<td>0.64</td>
<td>7.31</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.33</td>
</tr>
<tr>
<td>766015</td>
<td>CCR2</td>
<td>0.20</td>
<td>0.14</td>
<td>0.25</td>
<td>6.88</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>6.20</td>
</tr>
<tr>
<td>7050328</td>
<td>SEMA4C</td>
<td>-0.16</td>
<td>-0.22</td>
<td>-0.11</td>
<td>6.97</td>
<td>0</td>
<td>0.00</td>
<td>0.01</td>
<td>4.20</td>
</tr>
<tr>
<td>7580189</td>
<td>VCAM1</td>
<td>-0.21</td>
<td>-0.36</td>
<td>-0.14</td>
<td>5.22</td>
<td>0</td>
<td>0.00</td>
<td>0.01</td>
<td>4.39</td>
</tr>
<tr>
<td>580341</td>
<td>LAX1</td>
<td>-0.46</td>
<td>-0.59</td>
<td>-0.32</td>
<td>9.94</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.61</td>
</tr>
<tr>
<td>5460621</td>
<td>R1S1322</td>
<td>-0.46</td>
<td>-0.61</td>
<td>-0.32</td>
<td>7.47</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.08</td>
</tr>
<tr>
<td>2550088</td>
<td>BMP21</td>
<td>-0.55</td>
<td>-0.70</td>
<td>-0.39</td>
<td>8.10</td>
<td>0</td>
<td>0.00</td>
<td>0.01</td>
<td>6.73</td>
</tr>
<tr>
<td>3994084</td>
<td>MEF2D</td>
<td>-0.58</td>
<td>-0.73</td>
<td>-0.42</td>
<td>8.98</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.92</td>
</tr>
<tr>
<td>199009</td>
<td>HCDHP</td>
<td>-0.85</td>
<td>-1.08</td>
<td>-0.68</td>
<td>8.48</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>5.09</td>
</tr>
<tr>
<td>3611057</td>
<td>TFCR19</td>
<td>-0.90</td>
<td>-1.10</td>
<td>-0.69</td>
<td>8.10</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>4.90</td>
</tr>
</tbody>
</table>

Key to variables: **ProbeID**: Illumina probe identifier, **TargetID**: Gene name, **logFC**: log$_2$-fold change, **CI**: lower and upper bounds of 95% confidence interval for logFC variable, **AveExpr**: average expression value for all arrays, **t**: t-statistic, **P.Value**: raw P-value for linear model coefficient, **adj.P.Val**: Benjamini-Hochberg corrected P value for linear model coefficient, **B**: log-odds for differential expression.
Based on the differential expression analysis of this dataset, we conclude that an informative signature for active tuberculosis, including non-pulmonary tuberculosis, has been identified in blood.
Figure 9.3: **Heatmaps of significant and top 500 probes** Each heatmap shows the expression values of the selected (a) or top 500 (b) probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours of samples: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.
Figure 9.4: **PCA of differentially expressed probes.** Principal component analysis shows that the selected probes clearly separate active tuberculosis from not active tuberculosis.

Figure 9.5: **Overlap of differentially expressed probes for two different methods.**
Further work is required to fully characterise the biological meaning of this signature.

9.2 Array deconvolution and cell-specific differential gene expression

In this section we examine cellular composition of the samples, and the effect that cellular composition may have on differential gene expression.

9.2.1 Deconvolution results

The results of the deconvolution analysis are shown in Figure 9.6. The algorithm, based on the least-squares algorithm described by Abbas et al [100], and implemented in the CellMix package [134], returns a matrix of proportions of cell types in each sample. Some important caveats apply: Firstly, the matrix of proportions only includes cell types for which expression profiles were provided. Therefore, even if other cell-types are present in the sample, these will not contribute to the proportions matrix. The original paper by Abbas described expression profiles for 18 cell types, and only 12 are reported in the output. This is probably due to the fact that not all probes in the Abbas dataset (Affymetrix data) are directly mappable to the Illumina data. Indeed, of 359 features defining 17 cell-types in the CellMix package, 212 are mapped to target IDs, and after removal of duplicate probes, only 143 features are used for the final deconvolution analysis. It is unclear whether an improved probe mapping will substantially alter the results. Future work will re-examine this issue, and also compare these results to other deconvolution methods.

The proportions for each cell-type were then summarised for each group, and the results plotted as pairwise box-and-whisker plots (Figure 9.7). These plots are more informative than the stacked barcharts, as they allow for easy identification of large systematic differences between sample groups. The median proportions for each cell-type were statistically compared between the two disease phenotypes, and the results are presented in Table 9.2.

An interesting and unexpected finding is the highly significant difference in the proportion of natural killer cells (NK cells) that are activated. NK cells are bone-marrow derived mononucleated cells that originate from lymphocyte progenitor cells. In contrast to lymphocytes, NK cells have invariant receptors, and respond to a broad range of intracellular pathogens. In order to respond
9 Question 1: Tuberculosis

Figure 9.6: Barplots of detected and PBMC types. The stacked bar charts show the proportion of cell-types in each sample (top panel, neutrophils included, bottom panel neutrophils excluded, representing the PBMC population). Each bar is normalised to 1 (100%). Samples are ordered according to disease phenotype (not active TB followed by active TB). The legend lists the cell types identified. Abbreviations: neutro: neutrophils, DC act: activated dendritic cells, DC: non-activated dendritic cells, mono: monocytes, NK act: activated natural killer cells, NK: non-activated natural killer cells, B aIGM: BCR-ligated B cells, B: resting B cells, Tc act: activated CD8 T cell, Tc: resting CD8 T cell, Th act: activated CD4 T cell, Th: resting CD4 T cell
to such pathogens, they require activation by interferons and cytokines derived from macrophages and dendritic cells (e.g. interferon-\(\alpha\), interferon-\(\beta\) (type I interferons), and interleukin-12 (IL-12). Once bound to cells with intracellular pathogens they release cytotoxic granules (containing granzymes and perforin) which kill the infected cell.

In previous studies, NK cells were shown to be activated at the site of disease in a mouse model of tuberculosis, but this did not have an effect on pulmonary bacterial burden [156], questioning the role NK cells might play in protection against active tuberculosis and controlling the disease. An in vitro study showed that, in the presence of glutathione and IL-2/IL-12 stimulation, NK cells inhibit growth of *Mycobacterium tuberculosis* in monocytes; this growth inhibition itself was inhibited when FasL and CD40L, important mediators of NK cell cytotoxicity, were inhibited by antibodies [157]. This leaves open the question whether NK cells are protective in active tuberculosis in humans. In a recent study performed on human tuberculosis site-of-disease samples (lung resection samples) the authors demonstrated that NK cells infiltrated pulmonary granulomatous lesions. In addition, NK cells from healthy volunteers could be stimulated to release IFN\(\gamma\) and TNF\(\alpha\) by *M. tuberculosis* H37Rv or *M. bovis* BCG in the presence of IL-2 [158]. The authors also demonstrated variation in responses to TB antigens that correlated with variation in the KIR locus.

Given the increased proportion of activated NK cells in active tuberculosis suggested by the deconvolution results in light of the studies cited above, the functional consequences of this phenomenon require further exploration.

Table 9.2: **Significance of differences in cell proportions in active TB vs not active TB.** Significant differences are highlighted in yellow.

<table>
<thead>
<tr>
<th>names</th>
<th>P-value</th>
<th>Bonferroni</th>
<th>Benjamini Hochberg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th</td>
<td>0.2543</td>
<td>1.0000</td>
<td>0.3987</td>
</tr>
<tr>
<td>Th act</td>
<td>0.0728</td>
<td>0.8735</td>
<td>0.1747</td>
</tr>
<tr>
<td>Tc</td>
<td>0.2990</td>
<td>1.0000</td>
<td>0.3987</td>
</tr>
<tr>
<td>Tc act</td>
<td>0.0044</td>
<td>0.0531</td>
<td>0.0266</td>
</tr>
<tr>
<td>B</td>
<td>0.0358</td>
<td>0.4299</td>
<td>0.1075</td>
</tr>
<tr>
<td>B IgM</td>
<td>0.6990</td>
<td>1.0000</td>
<td>0.6990</td>
</tr>
<tr>
<td>NK</td>
<td>0.0112</td>
<td>0.1338</td>
<td>0.0446</td>
</tr>
<tr>
<td>NK act</td>
<td>0.0004</td>
<td>0.0052</td>
<td>0.0052</td>
</tr>
<tr>
<td>mono</td>
<td>0.4176</td>
<td>1.0000</td>
<td>0.5011</td>
</tr>
<tr>
<td>DC</td>
<td>0.0943</td>
<td>1.0000</td>
<td>0.1886</td>
</tr>
<tr>
<td>DC act</td>
<td>0.2817</td>
<td>1.0000</td>
<td>0.3987</td>
</tr>
<tr>
<td>neutro</td>
<td>0.6744</td>
<td>1.0000</td>
<td>0.6990</td>
</tr>
</tbody>
</table>
Figure 9.7: **Boxplots of detected and PBMC types.** The top panel shows box-and-whisker plots for cell proportions by disease phenotype for all 12 detected cell types, and the bottom panel for the PBMC subset. *Tc act*, *NK act* and *NK* are statistically significant after multiple testing correction.
9.2.2 Cell-type specific differential expression

Analysis of cell-type specific differential expression was limited to the top 500 differentially expressed genes combined with the probes in the Abbas data that uniquely identifies each cell type. By doing so, sensitivity for detecting subtle changes in gene expression in specific cell types was reduced, but the likelihood of detecting any cell-type specific differential expression was increased, as we focus on probes known to be differentially expressed, and remove the majority of probes most likely to contribute to false positive signals caused by excessive noise. So, in the first instance the cell-type specific differential expression output is principally useful to identify cell-types, or groups of cell-types, that are of prime interest for the contrast under investigation.

In several instances, members of the list of identified cell-types were combined into functional classes. This was done where the number of samples in either arm of the contrast was smaller then the number of cell-types. By using a proportion matrix containing fewer classes, even small data subsets were amenable to analysis, assuming that the broader classes used were biologically relevant. Given this caveat, much of the output of cell-type specific differential expression analysis should be regarded as hypothesis-generating at best, and not necessarily a reflection of biological reality.

Figure 9.8 demonstrates some results characteristic for many questions and data subsets. Differential expression of the analysed probes appears to be restricted to mainly NK cells, CD4 T cells and neutrophils, while other lymphocytes, monocytes and dendritic cells do not show evidence in favour of a cell-type specific signal. Given that these latter cell types often become active with regard to tuberculosis at the site of disease, this is not surprising. Deconvolution analysis above showed an increased proportion of activated NK cells, so it is reasonable to hypothesise that the differentially regulated probes for this cell subset are involved in NK cell activation and cytotoxicity. CD4 cells are central to the immune response to tuberculosis. Detection of a differential expression signal in CD4 T cells but not in other lymphocytes may indicate differential recirculation dynamics following exposure to antigen at the site of disease. Figure 9.9 shows the overlap of the top 100 differentially expressed probes for three cell types. 36-46% of probes are unique to these cell-types, highlighting the possibility that this method may be useful for defining specific biological roles in different cell types in tuberculosis using transcriptomic profiles. I will return to
9 Question 1: Tuberculosis

Figure 9.8: False discovery rate plots after cell-specific DE

NK cell activation in tuberculosis in the next Chapter.

9.3 Weighted gene co-expression network analysis (WGCNA)

Modular analysis using weighted gene co-expression network analysis was performed next. Using a soft-thresholding power of 5 and an $R^2$ cutoff of 0.8, sixteen modules were identified. Of the 8000 probes considered for this analysis, 7440 were assigned to one of the modules. The modules vary widely in the number of probes they contain, ranging from 3217 to 24. Figure 9.10 shows the sample dendrogram resulting from hierarchical clustering of the 8000 probes, and below this a heatmap of phenotypic traits.

In Figure 9.11 we clearly see that the contrast variable (class3) is strongly correlated with all modules. This is expected, given that the 8000 probes used for module detection were selected

1 Modules are arbitrarily named by the software using colour names as labels. As such, the module names are meaningless.

2 The variable class3 encodes “activeTB” and “notActiveTB”, while the variable class encodes “healthy”, “LTBI”, “PTB” and “TB-PC”. The former is used for the contrast in the differential expression analysis while the latter is used mainly for applying the correct colour to the classes in various figures. Correlations are opposite for these two variables due to the way the categorical variables are converted to numeric values used in the correlation calculations.
9.3 Weighted gene co-expression network analysis (WGCNA)

based on their probability of differential expression. Of interest is the fact that some additional variables also show significant correlation with most modules. These variables (like fever, weight loss, night sweats, pulse rate) are expected to correlate with presence and/or severity of clinical tuberculosis disease. Interestingly, this relationship is not seen with CD4 count, but this may be due to incomplete data as CD4 counts were not measured in many HIV-1 uninfected individuals.

Next, we highlight some features of interesting modules. Associations of individual genes with our trait of interest (clinical tuberculosis) are quantified by the measure “Gene Significance” (GS) (the absolute value of the correlation between the gene and the trait). In addition, a quantitative measure of module membership (MM) is defined as the correlation of the module eigengene and the gene expression profile. For significant modules, the GS and MM measures are correlated; examples for this are shown in Figure 9.12. Highly connected network hub genes tend to have high GS and MM values.

Another way to visualise the modules is to plot the values of the module eigengenes underneath a heatmap of expression data for the particular module, as in Figure 9.13. The barplots suggest that, for these three modules at least, the expression of the module genes is associated with the contrast variable (active TB or not active TB). Boxplots summarise the module eigengene expression data further, and Table 9.3 summarises these comparisons for all detected modules.

Modules can also be visualised in a network context. Figure 9.14 shows this by the example of the Red and Lightgreen modules. In both cases, results of differential expression analysis have
Figure 9.10: **WGCNA: Traits and modules.** In Subfigure a, categorical variables in the phenotype data were converted to numerical variables, and the matrix rescaled to the interval 0 to 1 for each variable. Missing data is shown in grey. Subfigure b shows the data used as basis for selecting the soft-thresholding power used in module detection, and the mean connectivity associated with various soft-thresholding powers. Subfigure c shows the results of module detection by clustering and dynamic branch cutting.
### 9.3 Weighted gene co-expression network analysis (WGCNA)

Table 9.3: **Significance of modules for contrast “active TB vs no TB”**. Results for comparing expression values of module eigengenes by Kruskal Wallis test. Uncorrected, Bonferroni-corrected and Benjamini-Hochberg-corrected P-values/ Q-values are shown. The *Midnightblue* module is not significant and has been highlighted in yellow. All module names are simply arbitrary colour labels.

<table>
<thead>
<tr>
<th>Module name</th>
<th>P-value</th>
<th>Bonferroni</th>
<th>Q-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 green</td>
<td>0.0090</td>
<td>0.1971</td>
<td>0.0104</td>
</tr>
<tr>
<td>2 turquoise</td>
<td>0.0003</td>
<td>0.0066</td>
<td>0.0007</td>
</tr>
<tr>
<td>3 black</td>
<td>5.8765 · 10^{-5}</td>
<td>0.0013</td>
<td>0.0004</td>
</tr>
<tr>
<td>4 lightyellow</td>
<td>0.0034</td>
<td>0.0756</td>
<td>0.0047</td>
</tr>
<tr>
<td>5 cyan</td>
<td>9.2786 · 10^{-5}</td>
<td>0.0020</td>
<td>0.0004</td>
</tr>
<tr>
<td>6 red</td>
<td>9.2786 · 10^{-5}</td>
<td>0.0020</td>
<td>0.0004</td>
</tr>
<tr>
<td>7 blue</td>
<td>9.2786 · 10^{-5}</td>
<td>0.0020</td>
<td>0.0004</td>
</tr>
<tr>
<td>8 midnightblue</td>
<td>0.0591</td>
<td>1.0000</td>
<td>0.0591</td>
</tr>
<tr>
<td>9 darkred</td>
<td>0.0112</td>
<td>0.2454</td>
<td>0.0123</td>
</tr>
<tr>
<td>10 tan</td>
<td>0.0006</td>
<td>0.0132</td>
<td>0.0010</td>
</tr>
<tr>
<td>11 pink</td>
<td>0.0006</td>
<td>0.0132</td>
<td>0.0010</td>
</tr>
<tr>
<td>12 brown</td>
<td>0.0011</td>
<td>0.0248</td>
<td>0.0018</td>
</tr>
<tr>
<td>13 purple</td>
<td>0.0001</td>
<td>0.0031</td>
<td>0.0005</td>
</tr>
<tr>
<td>14 magenta</td>
<td>0.0003</td>
<td>0.0066</td>
<td>0.0007</td>
</tr>
<tr>
<td>15 yellow</td>
<td>0.0090</td>
<td>0.1971</td>
<td>0.0104</td>
</tr>
<tr>
<td>16 greenyellow</td>
<td>0.0026</td>
<td>0.0582</td>
<td>0.0039</td>
</tr>
<tr>
<td>17 grey60</td>
<td>0.0056</td>
<td>0.1242</td>
<td>0.0073</td>
</tr>
<tr>
<td>18 lightcyan</td>
<td>0.0169</td>
<td>0.3718</td>
<td>0.0177</td>
</tr>
<tr>
<td>19 lightgreen</td>
<td>0.0006</td>
<td>0.0132</td>
<td>0.0010</td>
</tr>
<tr>
<td>20 salmon</td>
<td>0.0003</td>
<td>0.0066</td>
<td>0.0007</td>
</tr>
<tr>
<td>21 royalblue</td>
<td>0.0004</td>
<td>0.0095</td>
<td>0.0009</td>
</tr>
<tr>
<td>22 grey</td>
<td>9.2786 · 10^{-5}</td>
<td>0.0020</td>
<td>0.0004</td>
</tr>
</tbody>
</table>
The results of WGCNA-analysis demonstrate that modules significant in tuberculosis can be identified. These modules await further characterisation at a more detailed level by identifying the genes central to the modules’ functioning, as well as any pathways or groups of pathways associated with the modules.
Figure 9.12: **Top 3 GS/MM plots** This figure demonstrates correlation of the measures *Gene Significance vs. Module Membership* (GS/MM). The results for three modules are shown.
Figure 9.13: Three significant modules. Results for three modules are demonstrated in more detail. Bar plots of eigengene expression combined with a heatmap of the probes making up the module are shown on the left. The right panel shows a box and whisker chart of a summary of all eigengenes (one per sample) in each of the two contrast classes. Test statistics for the median difference are shown in Table 9.3. The sample names are coloured by TB class (green = healthy, blue = LTBI, orange = PTB, red = TB-PC). Note that for these three modules their respective eigengenes are downregulated in not active TB and upregulated in active TB.
9.3 Weighted gene co-expression network analysis (WGCNA)

Figure 9.14: **Weighted gene co-expression networks.** Networks of gene co-expression were generated in Cytoscape. Nodes reported as differentially expressed in earlier analysis are coloured on a green (downregulated in active TB) to red (upregulated in active TB) colour scale. Network edges are weighted by the strength of association between any of their nodes.
Part IV

IMPI-MA: Results in breadth


10 Contrasts involving tuberculosis

In this and the following two chapters I present a broad impression of results for all questions that
form part of the analysis. The results are mainly presented in tabular form and summary images.
The emphasis here is on comparison of results (rendered in broad brushstrokes as opposed to the
level of detail used in the preceding chapter); in particular, the results for two datasets in each
question will be juxtaposed, in order to highlight similarities and differences.

This Chapter deals with questions relating to active and latent tuberculosis, effects of pulmonary
vs. extrapulmonary tuberculosis and differences between haemodynamic phenotypes of TB-PC.

General comment: Due to heterogeneity of the IMPI-MA data, many analyses lack adequate
statistical power. Some data subsets lend themselves to splitting into “training” and “test” sets, as
sufficient numbers of samples are available. Future work will address this by splitting IMPI-MA
data subsets into training and test sets or applying the signatures to other extant microarray datasets
(Berry et al, Maertzdorf et al). The current work does not provide for validation of signatures, and
for this reason should be considered as an exploration of the “space” of signatures in HIV-TB.

10.1 Question 1: Tuberculosis

In this Section I show the results for the contrast “active TB vs. not-active TB”, in order to address
questions regarding the biology of active tuberculosis disease. Related to this question are differ-
ences in biology that are due to the site of active disease (lung vs. pericardium), biological features
of LTBI, and differences in cellular and transcriptional responses in two different haemodynamic
presentations of TB-PC. These three additional questions will be addressed in the following sec-
tions.
10 Contrasts involving tuberculosis

10.1.1 Included patients and samples

I examined the contrast active vs not active TB in blood in two contexts: HIV-1 uninfected and HIV-1 co-infected individuals. Table 10.1 show the baseline demographic and clinical characteristics of the study participants on whose samples the microarray analysis is based. This table has been structured to facilitate comparisons between the two sample sets as well as for the contrast variable within each sample set. Three variables were examined: age, sex and CD4 count.

In comparisons across the two study groups, HIV-1/M. tuberculosis co-infected individuals were younger than their HIV-1 uninfected counterparts, but this difference failed to reach statistical significance. HIV-1 infected individuals were predominantly female and HIV-1/M. tuberculosis co-infected individuals had lower CD4 counts than their HIV-1 uninfected counterparts. Comparing these three variables within each study group, the HIV-1 uninfected group was well matched for age and sex (CD4 count could not be assessed), but the HIV-1 infected group was only matched for age, as all cases of not active TB were female, and the active TB group had significantly lower median CD4 counts. The sex bias in the HIV-1 infected not active TB group is explained by patterns of health-seeking behaviour in asymptomatic HIV-1 infected persons; in South Africa men infrequently attend wellness clinics if asymptomatic, therefore females are much more readily recruited into this category. Future studies should employ stratified sampling in order to avoid this type of bias. In summary, when examining the contrast activeTB vs not active TB we should be aware of possible bias introduced by different CD4 counts and male to female ratios in the data in the HIV-1 infected group, and when comparing the results of the two analyses, we should consider the possibility of sex and CD4 count contributing to any differences in results.

10.1.2 Overall results

In Table 10.2 I show the comparison of the main results of the analytic pipeline. A much higher number of probes (5404 vs 369) were differentially expressed in the HIV-1 infected group at a false discovery rate cutoff of 0.05. This is likely due to several factors. Firstly, the second analysis had greater statistical power to detect differences, as 3.3 times the number of study participants in the HIV-1 uninfected group were available. Secondly, as noted above, the HIV-1 infected study

---

1While the table is hard to read, it summarises a lot of data, and allows the comparison of variables within and between datasets. Comparisons between datasets are read horizontally, and comparisons within datasets are read vertically.
Table 10.1: **Clinical characteristics: active TB vs not active TB.** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th></th>
<th>Blood HIVneg activeTB</th>
<th>Blood HIVpos activeTB</th>
<th>Pval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Median</td>
<td>IQR</td>
<td>Median</td>
<td>IQR</td>
</tr>
<tr>
<td>age_All</td>
<td>33.83</td>
<td>26.55</td>
<td>31.38</td>
<td>32.98</td>
</tr>
<tr>
<td>age_activeTB</td>
<td>40.67</td>
<td>33.69</td>
<td>52.86</td>
<td>33.3</td>
</tr>
<tr>
<td>age_notActiveTB</td>
<td>29.9</td>
<td>22.98</td>
<td>34.2</td>
<td>29.5</td>
</tr>
<tr>
<td>age_P value</td>
<td>0.086</td>
<td>0.084</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_Test</td>
<td>Kruskal-Wallis rank</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sex</th>
<th>ratio F/M</th>
<th>Female</th>
<th>Male</th>
<th>ratio F/M</th>
<th>Female</th>
<th>Male</th>
<th>Pval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>sex_All</td>
<td>0.467</td>
<td>7</td>
<td>15</td>
<td>2.850</td>
<td>23</td>
<td>20</td>
<td>0.001</td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_activeTB</td>
<td>0.714</td>
<td>5</td>
<td>7</td>
<td>1.300</td>
<td>26</td>
<td>20</td>
<td>0.553</td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_notActiveTB</td>
<td>0.250</td>
<td>2</td>
<td>8</td>
<td>Inf</td>
<td>27</td>
<td>9</td>
<td>0.000</td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_P value</td>
<td>0.531</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sex_Test</td>
<td>2-sample test for eq</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CD4</th>
<th>Median</th>
<th>IQR</th>
<th>Median</th>
<th>IQR</th>
<th>Median</th>
<th>IQR</th>
<th>Pval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD4_All</td>
<td>566</td>
<td>404.5</td>
<td>653</td>
<td>260</td>
<td>113</td>
<td>361.5</td>
<td>0.007</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>CD4_activeTB</td>
<td>566</td>
<td>404.5</td>
<td>653</td>
<td>167</td>
<td>86</td>
<td>291.5</td>
<td>0.001</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>CD4_notActiveTB</td>
<td>333</td>
<td>303.5</td>
<td>467</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>CD4_P value</td>
<td>NA</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD4_Test</td>
<td>NA</td>
<td>Kruskal-Wallis rank</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
participants grouped by contrast variable were unmatched for sex and CD4 count, and finally, it is plausible that HIV-1 co-infection introduces additional variability to gene expression. Following deconvolution analysis, proportions for 3 of 12 cell types were significantly different in HIV-1 uninfected individuals, whereas 8 of 14 cell types differed in proportion in HIV-1 infected individuals. For the WGCNA analysis the same soft thresholding power was selected; a similar number of modules was detected in both cases, with at least 93% of probes assigned to modules. The following subsection examines some results in more detail.

Table 10.2: **Overall results: active TB vs not active TB.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 1: Tuberculosis</th>
<th>Contrast: Active TB vs. not Active TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>1</td>
</tr>
<tr>
<td>Context</td>
<td>Blood, HIV neg</td>
</tr>
<tr>
<td>N: Active TB</td>
<td>12</td>
</tr>
<tr>
<td>N: not Active TB</td>
<td>10</td>
</tr>
</tbody>
</table>

**Analysis 1: Differential expression**

| N significant probes (unadjusted) | 4121 | 8847 |
| N significant probes (BH adjusted) | 369  | 5405 |
| N probes used (heatmaps, csDE)    | 369  | 2000 |

**Analysis 2: Deconvolution**

| Deconvolution successful | yes | yes |
| N of N significant for contrast (BH) | 3 of 12 | 8 of 14 |
| N of N PBMC significant for contrast (BH) | 2 of 11 | 5 of 13 |
| Cell-specific DE successful | yes | yes |
| Number of cell types | 7 | 7 |
| N cell types that reach FDR < 0.4 | 2 | 3 |

**Analysis 3: WGCNA**

| R² cutoff for scale-free approximation | 0.8 | 0.8 |
| Soft threshold power                  | 5   | 5   |
| Number of modules                     | 21  | 16  |
| Probes assigned to modules (N, %)     | 7937 (99) | 7440 (93) |
| GO: Entrez IDs submitted              | 4382 | 4730 |
| GO: Entrez IDs mapped                 | 2920 | 3182 |
| N (%) modules cor GS/MM sig          | 17 (81) | 12 (75) |
| N (%) modules sig for contrast (BH)   | 20 (95) | 16 (100) |
10.1 Question 1: Tuberculosis

10.1.3 Results for context: HIV (datasets: Blood, HIV negative and Blood, HIV positive)

How well do these differentially expressed probes classify the input data? Figure 10.1 shows heatmaps of the significantly differentially expressed probes for the HIV-1 uninfected group, and the top 2000 differentially expressed probes for the HIV-1 infected group. In the first case the input data is classified well (one “healthy” sample is misclassified), and in the second case, two distinct clusters are seen: one cluster only contains active TB cases, and the second six active TB cases in addition to all the not-active TB cases. The reason for this misclassification may lie in the fact that either an insufficient number of probes were selected, or that the signal due to HIV-1 infection interferes with the active TB signal.

False discovery rate (FDR) plots for the two analyses (Figure 10.2) offer an additional perspective on the results. While the signal for cell-specific differential expression is seen in CD4 T cells, NK cells and to a lesser extent neutrophils in HIV-1 uninfected individuals (see also Chapter 9), we find stronger signals in CD8 T cells, neutrophils, monocytes and dendritic cells in HIV-1 infected individuals, while the signal for differential expression in NK cells is much less convincing. This may be due to effects of HIV-1 on gene expression in these cell subsets in the context of a TH1-predominant inflammatory background. It is well described that *Mycobacterium tuberculosis* and HIV-1 interact significantly (See Chapter 1). Both monocytes and dendritic cells may become infected with HIV-1 [159], and it is possible that the differential expression in these cell types is due to effects of HIV-1 which are only seen in the context of active tuberculosis. A possible effect of HIV-1 on monocytes and dendritic cells is the induction of apoptosis [160], and detection of induction of pro-apoptotic factors in these cell subsets would provide additional evidence for this.

Module detection was similar in both groups as see in Figure 10.3. The functional characteristics of the identified modules are still to be determined.

Finally I looked at the overlap of differentially expressed probes in the two contexts. 237 probes were found to overlap between the 369 probes found in HIV-1 uninfected individuals and the top 2000 probes in HIV-1 infected individuals, accounting for 64% of the probes in the first set. The non-overlapping probes may represent noise or HIV-1 specific effects, while the overlapping probes represent effects driven by active tuberculosis regardless of HIV-1 infection.
Figure 10.1: Results for contrast TB status in contexts HIV negative and HIV positive. The heatmaps show good separation of samples regardless of HIV status. The HIV negative group was discussed in Chapter 9. In comparison, the HIV positive group had 14.6 times the number of differentially regulated probes; only the top 2000 probes are shown here. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: healthy, blue: LTBI, orange: PTB, red: TB-PC. Plot titles reflect the number of included probes.
Figure 10.2: **Cell-type specific differential expression.** False discovery rate plots for the two analyses. For each plot the x-axis shows the number of differentially expressed probes called, given the false discovery rate shown on the y-axis.
Figure 10.3: **Module clustering.** The plots show the dendrograms resulting from hierarchical clustering of the 8000 probes used in the analysis. Colours indicate the modules to which probes were assigned; these colours correspond to the module names.

(a) Module clustering HIV neg  
(b) Module clustering HIV pos

Figure 10.4: **Overlap of differentially expressed probes.**

(a) 369 significantly differentially expressed probes  
(b) Top 500 differentially expressed probes
10.2 Question 2: PTB and extrapulmonary TB

In this Section I examine the contrast “PTB vs TB-PC”. This looks for differences in transcriptional response detectable in blood based on the site of MTB infection. The underlying hypothesis for this contrast is that the site of disease elicits an additional signal in the transcriptomic data over and above that of TB itself.

10.2.1 Included patients and samples

Table 10.3 summarises the variables age, sex and CD4 count across two groups of study subjects (HIV-1 uninfected and -infected) as well as within each of the two groups across the contrast variable PTB vs TB-PC. HIV-1 infected individuals presenting with TB-PC were younger than their HIV-1 uninfected counterparts, and HIV-1 infected individuals presenting with PTB were all female while the female to male ratio in the HIV-1 uninfected group was 0.33. As expected, median CD4 counts in HIV-1 infected individuals presenting with TB-PC were significantly lower than in HIV-1 uninfected cases. The HIV-1 infected group was matched for age, but not sex or CD4 count while HIV-1 uninfected group was matched for age and sex (matching for CD4 count could not be assessed due to missing data). The finding that HIV-1 infected individuals presenting with TB-PC, a form of extrapulmonary TB, had lower CD4 counts than those presenting with pulmonary disease is in line with expectation. In summary, differences in CD4 count and sex should be considered as potential biasing factors in this analysis.

10.2.2 Overall results

Table 10.4 lists the overall statistics of running the analytic pipeline on the two datasets. It should be noted that the sample sizes in the HIV-1 uninfected group are very small, and as such, any results should be regarded as hypothesis generating at best. Another potentially important biasing factor is the fact that the RNA from the two groups was collected at different times in different study contexts, and was extracted at different time points. Of all analyses, this one is probably most susceptible to technical bias.

The number of differentially expressed probes found in the two datasets after multiple testing correction was 38 (HIV-1 infected) and 0 (HIV-1 uninfected), respectively. In order to explore
Table 10.3: Clinical characteristics: PTB vs TB-PC. Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th></th>
<th>Blood HIVpos</th>
<th>Blood HIVneg</th>
<th>Pval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PTB-TBPC</td>
<td>PTB-TBPC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>Median</td>
<td>Median</td>
<td>LQ</td>
<td>UQ</td>
</tr>
<tr>
<td>age_All</td>
<td>33.3</td>
<td>33.69</td>
<td>29.6</td>
<td>38.6</td>
</tr>
<tr>
<td>age_CON_PTB</td>
<td>34.25</td>
<td>33.85</td>
<td>31.78</td>
<td>39.6</td>
</tr>
<tr>
<td>age_TBPC</td>
<td>33.13</td>
<td>35.3</td>
<td>29.7</td>
<td>38.32</td>
</tr>
<tr>
<td>age_Test</td>
<td>0.749</td>
<td>0.749</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Sex      | ratio F/M | Female | Male |          |          |          |          |          |
|----------|-----------|--------|------|----------|----------|----------|----------|
| sex_All  | 1.300     | 26     | 20   | 0.744    | 5        | 3        | 0.551    |
| sex_CON_PTB | Inf       | Inf    | Inf  | 0.333    | 1        | 3        | 0.018    |
| sex_TBPC | 0.840     | 16     | 20   | 1.000    | 4        | 4        | 1.000    |
| sex_Test | 0.006     | 0.836  |      |          |          |          |          |

| CD4      | Median     | Median     | LQ   | UQ     |        |          |          |          |
|----------|------------|------------|------|--------|--------|----------|----------|
| C4D_All  | 167        | 566        | 404.5| 653    | 0.001  | Kruskal- | rank    |
| C4D_CON_PTB | 322        | 454.5      | NA   | NA     |
| C4D_TBPC | 110        | 237.2      | 566  | 404.5  | 653    | 0.000    | Kruskal- | rank    |
| C4D_P value | 0.000       | NA         |      |        |        |          |          |
| C4D_Test | Kruskal-   | NA         |      |        |        | rank     |          |

<table>
<thead>
<tr>
<th></th>
<th>Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10 Contrasts involving tuberculosis
classification of the input data and probe overlap, I selected the top 300 differentially expressed probes (using as ranking measure the log-odds of differential expression), bearing in mind that many of these probes are potentially (but not definitely) false positives. The sample size in the HIV-1 uninfected group was too small for meaningful cell-specific differential expression; this will not be reported here. Finally, a high number of modules was identified in the HIV-1 uninfected group (67), with all but two probes assigned to modules. In contrast, only 62.1% of probes were assigned to modules in the HIV-1 infected group.

Table 10.4: **Overall results: PTB vs TB-PC.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 2: TB Site</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contrast:</strong> PTB vs. TB-PC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Context</td>
<td>Blood, HIV pos</td>
<td>Blood, HIV neg</td>
</tr>
<tr>
<td>N: PTB</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>N: TB-PC</td>
<td>36</td>
<td>8</td>
</tr>
</tbody>
</table>

**Analysis 1: Differential expression**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N significant probes (unadjusted)</td>
<td>3593</td>
<td>2260</td>
</tr>
<tr>
<td>N significant probes (BH adjusted)</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>N probes used (heatmaps, csDE)</td>
<td>300</td>
<td>300</td>
</tr>
</tbody>
</table>

**Analysis 2: Deconvolution**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Deconvolution successful</td>
<td>yes</td>
</tr>
<tr>
<td>N of N significant for contrast (BH)</td>
<td>3 of 13</td>
</tr>
<tr>
<td>N of N PBMC significant for contrast (BH)</td>
<td>2 of 12</td>
</tr>
<tr>
<td>Cell-specific DE successful</td>
<td>yes</td>
</tr>
<tr>
<td>Number of cell types</td>
<td>7</td>
</tr>
<tr>
<td>N cell types that reach FDR &lt; 0.4</td>
<td>1</td>
</tr>
</tbody>
</table>

**Analysis 3: WGCNA**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R² cutoff for scale-free approximation</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>Soft threshold power</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Number of modules</td>
<td>18</td>
<td>67</td>
</tr>
<tr>
<td>Probes assigned to modules (N, %)</td>
<td>4971 (62.1)</td>
<td>7998 (99.99)</td>
</tr>
<tr>
<td>GO: Entrez IDs submitted</td>
<td>4406</td>
<td>4048</td>
</tr>
<tr>
<td>GO: Entrez IDs mapped</td>
<td>2925</td>
<td>2662</td>
</tr>
<tr>
<td>N (%) modules cor GS/MM sig</td>
<td>11 (61)</td>
<td>44 (66)</td>
</tr>
<tr>
<td>N (%) modules sig for contrast (BH)</td>
<td>18 (100)</td>
<td>37 (55)</td>
</tr>
</tbody>
</table>
10.2.3 Results for context HIV: (datasets: Blood, HIV negative and Blood, HIV positive)

The heatmaps in Figure 10.5 show that in both cases, the top 300 selected probes do quite well in classifying the input data. There are no misclassifications in the HIV-1 uninfected group, and two cases of PTB are misclassified as TB-PC in the HIV-1 infected group. Given the higher risk for extrapulmonary tuberculosis in HIV-1 co-infection, concurrent pulmonary and extrapulmonary tuberculosis in these cases can not be excluded as an explanation for this. In any event, any difference in transcriptional profiles that are due to the site of TB disease is likely to be small, with a much more subtle signal than active TB vs. not active TB.

Cell-type specific differential expression analysis offers additional clues about the significance of any detectable differential expression between sites of tuberculosis disease (Figure 10.6). Due to small sample numbers, no estimate could be made in the HIV-1 uninfected group, but in HIV-1 infected individuals the cell-type specific signal is strongest in CD8 T cells. Importantly, given that the TB-PC cases had lower CD4 counts than the PTB cases, the signal may in fact be due to a significant effect of HIV, rather than a signal due to tuberculosis disease. With this interpretation, extrapulmonary tuberculosis would serve as a marker of more advanced HIV disease. More likely, the signal is of mixed origin, with differential severity of HIV-1 infection, \textit{M. tuberculosis} organism burden and location-specific changes.

Figure 10.7 highlights differences in module detection in the two datasets. A much large number of modules is found in HIV-1 uninfected individuals.

Figure 10.8 is informative regarding the biological significance of the two signatures; of the top 300 differentially expressed probes, only 14 probes overlap. While it is likely that the selected probes are biologically informative, the added effect of HIV-1 infection in the context of extrapulmonary tuberculosis probably leads to the lack of overlap.
Figure 10.5: **Results for contrast TB site in contexts HIV negative and HIV positive.** The heatmaps show good separation of samples based on site of TB infection. Few or no probes for significantly differentially expressed after Benjamini Hochberg correction, but despite this the top 300 differentially expressed probes (based on ranking by log-odds of differential expression) show good separation of the samples. The question of whether we are seeing a batch effect (the samples were collected in different studies) or a true biological effect remains currently an open one. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: orange: PTB, red: TB-PC. Plot titles reflect the number of included probes.
Figure 10.6: **Cell-type specific differential expression.** Only the result for HIV-1 infected individuals is shown, as the HIV-1 uninfected group was too limited in sample numbers for meaningful analysis.

Figure 10.7: **Module clustering.**
10.2 Question 2: PTB and extrapulmonary TB

(a) Overlap of top 300 differentially expressed probes  
(b) Overlap of top 500 differentially expressed probes

Figure 10.8: Overlap of differentially expressed probes.
10.3 Question 3: Latent TB

This subsection addresses the question of whether LTBI leaves a detectable signal in the blood transcriptome relative to no prior TB sensitisation.

10.3.1 Included patients and samples

Table 10.5 summarises the variables age, sex and CD4 count across two groups of study subjects (HIV-1 uninfected and -infected) as well as within each of the two groups across two groups defined by the contrast variable LTBI vs healthy. The two groups, as well as the two comparator groups within each group were matched for age. Both comparator groups within each set were also matched for sex, but the two sets differed significantly w.r.t. this variable; no male patients were included in the HIV-1 infected group, whereas both men and women were included in the HIV negative group. Finally, assessment for matching for CD4 count was only possible in the HIV-1 infected group (where the healthy and LTBI groups were matched for CD4 count), as CD4 count data was unavailable for the HIV-1 uninfected individuals. In summary, some bias due to sex may be expected when comparing the results of the two analyses.

10.3.2 Overall results

Table 10.6 lists the results of the analysis for the two datasets of interest. Once again, few to no probes (0 and 1, respectively) were significantly differentially expressed after multiple testing correction. Deconvolution and cell-specific differential expression was successful for both datasets, as was module detection.

10.3.3 Results for context HIV: (datasets: Blood, HIV negative and Blood, HIV positive)

With selection of the top 300 differentially expressed probes, both HIV-1 uninfected and uninfected input data sets are separated perfectly into two clusters, suggesting that the probes are biologically meaningful.

Cell-type specific differential expression in HIV-1 uninfected individuals only exhibited a very weak signal in natural killer cells and neutrophils, while in HIV-1 infected individuals a very strong
Table 10.5: Clinical characteristics: Healthy vs LTBI. Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th></th>
<th>Blood HIV pos</th>
<th>Healthy-LTBI</th>
<th>Blood HIV neg</th>
<th>Healthy-LTBI</th>
<th>P-val</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Median</td>
<td>LQ</td>
<td>UQ</td>
<td>Median</td>
<td>LQ</td>
<td>UQ</td>
</tr>
<tr>
<td>age_All</td>
<td>29.3</td>
<td>25.75</td>
<td>37.3</td>
<td>29.9</td>
<td>22.98</td>
<td>34.2</td>
</tr>
<tr>
<td>age_CON_LTBI</td>
<td>29.3</td>
<td>25.5</td>
<td>35.3</td>
<td>29.1</td>
<td>22.7</td>
<td>34.7</td>
</tr>
<tr>
<td>age_CON_healthy</td>
<td>29.3</td>
<td>26.08</td>
<td>39.47</td>
<td>30.7</td>
<td>23.8</td>
<td>32.7</td>
</tr>
<tr>
<td>age_P value</td>
<td>0.544</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Female</th>
<th>Male</th>
<th>Female</th>
<th>Male</th>
<th>P-val</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>sex_All</td>
<td>Inf</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0.250</td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_CON_LTBI</td>
<td>Inf</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0.250</td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_CON_healthy</td>
<td>Inf</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0.250</td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_P value</td>
<td>NaN</td>
<td>1.000</td>
<td>1.000</td>
<td></td>
<td></td>
<td>2-sample test for eq</td>
</tr>
<tr>
<td>sex_Test</td>
<td>2-sample test for eq</td>
<td>2-sample test for eq</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Median</th>
<th>LQ</th>
<th>UQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD4_All</td>
<td>387.5</td>
<td>346</td>
<td>NA</td>
</tr>
<tr>
<td>CD4_CON_LTBI</td>
<td>356</td>
<td>346</td>
<td>NA</td>
</tr>
<tr>
<td>CD4_CON_healthy</td>
<td>322.5</td>
<td>342.5</td>
<td>NA</td>
</tr>
<tr>
<td>CD4_P value</td>
<td>0.846</td>
<td></td>
<td>NA</td>
</tr>
<tr>
<td>CD4_Test</td>
<td></td>
<td></td>
<td>Kruskal-Wallis rank</td>
</tr>
</tbody>
</table>
Table 10.6: **Overall results: Healthy vs LTBI** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 3: Latent TB infection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast: LTBI vs. Healthy</td>
</tr>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>Context</td>
</tr>
<tr>
<td>N: LTBI</td>
</tr>
<tr>
<td>N: Healthy</td>
</tr>
</tbody>
</table>

**Analysis 1: Differential expression**

| N significant probes (unadjusted) | 1329 | 2111 |
| N significant probes (BH adjusted)| 0    | 1    |
| N probes used (heatmaps, csDE)    | 300  | 300  |

**Analysis 2: Deconvolution**

| Deconvolution successful | yes | yes |
| N of N significant for contrast (BH) | 0 of 12 | 0 of 13 |
| N of N PBMC significant for contrast (BH) | 0 of 11 | 0 of 12 |
| Cell-specific DE successful | yes | yes |
| Number of cell types | 7    | 4    |
| N cell types that reach FDR < 0.4 | 2    | 0    |

**Analysis 3: WGCNA**

| R² cutoff for scale-free approximation | 0.8 | 0.8 |
| Soft threshold power                  | 3   | 10  |
| Number of modules                     | 21  | 32  |
| Probes assigned to modules (N, %)     | 7182 (89.8) | 6783 (84.8) |
| GO: Entrez IDs submitted              | 3900 | 4201 |
| GO: Entrez IDs mapped                 | 2573 | 2805 |
| N (%) modules cor GS/MM sig          | 8 (38) | 21 (65) |
| N (%) modules sig for contrast (BH)   | 20 (95) | 24 (75) |
10.3 Question 3: Latent TB

(a) Heatmap of differentially expressed probes, HIV-1 uninfected

(b) Heatmap of differentially expressed probes, HIV-1 infected

Figure 10.9: Results for contrast LTBI in contexts HIV negative and HIV positive. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: healthy, blue: LTBI. Plot titles reflect the number of included probes.
Contrasts involving tuberculosis

neutrophil signal combined with weaker signals for CD8 T cells and NK cells are evident.

Almost all probes were assigned to modules in both cases.

Once again, the top 300 differentially expressed probes hardly overlap (3 out of 300), again highlighting the possibility that HIV-1 infection in the context of LTBI drives the differential expression seen.
10.3 Question 3: Latent TB

Figure 10.10: Cell-type specific differential expression.

(a) False discovery rate plot HIV-1 uninfected

(b) False discovery rate plot HIV-1 infected
10 Contrasts involving tuberculosis

Figure 10.11: Module clustering.

(a) Module clustering HIV neg

(b) Module clustering HIV pos

Figure 10.12: Overlap of differentially expressed probes.

(a) Overlap of top 300 differentially expressed probes

(b) Overlap of top 500 differentially expressed probes
10.4 Question 4: Haemodynamic phenotypes in TB-PC

The final question relating to tuberculosis deals with haemodynamic phenotypes. Here I examine the contrast “effusive TB-PC vs. effusive-constrictive TB-PC”. Here I compare the results of this analysis for two compartments: blood and pericardial fluid, the site of disease. It is anticipated that the signal for effusive-constrictive disease, if any, is stronger in the pericardial fluid compartment.

10.4.1 Included patients and samples

As the comparison focuses on matched blood and pericardial fluid samples, all comparisons across the two datasets are not significant, as they consider the same study subjects. In addition, the samples are matched for age, sex and CD4 count across the contrast variable. See Table 10.7 for details.

Table 10.7: Clinical characteristics: Effusive vs Effusive-constrictive pericarditis. Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.
10.4.2 Overall results

No probes were significantly differentially expressed after multiple testing correction, so again the top 300 differentially expressed probes ranked by log-odds for differential expression were used for downstream analyses and visualisations. Deconvolution, cell-type specific differential expression analysis and module detection were performed successfully. Table 10.8 lists the main results of this analysis.

Table 10.8: Overall results: Effusive vs Effusive-constrictive pericarditis. Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 4: Haemodynamic phenotype</th>
<th>Contrast: Effusive-constrictive pericarditis (EC) vs. Effusive pericarditis (Eff)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>1</td>
</tr>
<tr>
<td>Context</td>
<td>Blood, TB-PC, HIV pos</td>
</tr>
<tr>
<td>N: EC</td>
<td>9</td>
</tr>
<tr>
<td>N: Eff</td>
<td>11</td>
</tr>
<tr>
<td>Analysis 1: Differential expression</td>
<td></td>
</tr>
<tr>
<td>N significant probes (unadjusted)</td>
<td>1195</td>
</tr>
<tr>
<td>N significant probes (BH adjusted)</td>
<td>0</td>
</tr>
<tr>
<td>N probes used (heatmaps, csDE)</td>
<td>300</td>
</tr>
<tr>
<td>Analysis 2: Deconvolution</td>
<td></td>
</tr>
<tr>
<td>Deconvolution successful</td>
<td>yes</td>
</tr>
<tr>
<td>N of N significant for contrast (BH)</td>
<td>0 of 13</td>
</tr>
<tr>
<td>N of N PBMC significant for contrast (BH)</td>
<td>0 of 12</td>
</tr>
<tr>
<td>Cell-specific DE successful</td>
<td>yes</td>
</tr>
<tr>
<td>Number of cell types</td>
<td>7</td>
</tr>
<tr>
<td>N cell types that reach FDR &lt; 0.4</td>
<td>4</td>
</tr>
<tr>
<td>Analysis 3: WGCNA</td>
<td></td>
</tr>
<tr>
<td>R² cutoff for scale-free approximation</td>
<td>3</td>
</tr>
<tr>
<td>Soft threshold power</td>
<td>0.8</td>
</tr>
<tr>
<td>Number of modules</td>
<td>34</td>
</tr>
<tr>
<td>Probes assigned to modules (N, %)</td>
<td>7808 (97.6)</td>
</tr>
<tr>
<td>GO: Entrez IDs submitted</td>
<td>3759</td>
</tr>
<tr>
<td>GO: Entrez IDs mapped</td>
<td>2452</td>
</tr>
<tr>
<td>N (% modules cor GS/MM sig</td>
<td>18 (53)</td>
</tr>
<tr>
<td>N (% modules sig for contrast (BH)</td>
<td>33 (97)</td>
</tr>
</tbody>
</table>
10.4.3 Results for context: Compartment (datasets: TB-PC HIVpos Blood, TB-PC HIVpos Fluid)

The heatmaps shown in Figure 10.13 show that the top 300 differentially expressed probes cluster their respective input data perfectly.

Cell-type specific differential expression yielded stronger signals in blood (for CD4 T cells, B cells, NK cells, monocytes and dendritic cells) than in pericardial fluid (for neutrophils and CD8 T cells). This interesting finding warrants further investigation.

Module clustering is shown in Figure 10.15. Of interest is the fact that most probes were not assigned to any module.

As seen previously, the overlap of differentially expressed probes is very small when comparing blood and pericardial fluid. Blood appears to be a very poor mirror for events at the site of disease.
Figure 10.13: **Results for contrast Haemodynamic phenotype in contexts Blood and Fluid.** Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: yellow: *effusive-constrictive phenotype*, blue: *effusive phenotype*. Plot titles reflect the number of included probes.
Figure 10.14: Cell-type specific differential expression.
Module clustering.

Figure 10.15: **Module clustering.**

Overlap of top 300 differentially expressed probes

Figure 10.16: **Overlap of differentially expressed probes.**
10.5 Pathway analysis

Pathway analysis was performed as described in Section 3.8.8. Tables 10.9, 10.10 and 10.11 list the pathways found to be significantly differentially regulated in active tuberculosis in HIV-1 uninfected and -infected groups, respectively. Figures 10.18 and 10.19 show heatmaps for the differentially regulated pathways in tuberculosis cases.
Table 10.9: **Top pathways in TB (HIV-1 uninfected).** Positive results refer to upregulation of the pathway in active tuberculosis, and negative numbers refer to downregulation. Colours: *green* = also downregulated in HIV-TB, *red* = also upregulated in HIV-TB, *yellow* = downregulated in HIV-TB. The *Mean logFC* value indicates the extent to which the pathway is up- or downregulated by providing the log$_2$-fold change value.

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa00190 Oxidative phosphorylation</td>
<td>2.356</td>
</tr>
<tr>
<td>hsa03050 Proteasome</td>
<td>1.573</td>
</tr>
<tr>
<td>hsa04145 Phagosome</td>
<td>1.365</td>
</tr>
<tr>
<td>hsa04610 Complement and coagulation cascades</td>
<td>1.256</td>
</tr>
<tr>
<td>hsa04260 Cardiac muscle contraction</td>
<td>1.073</td>
</tr>
<tr>
<td>hsa04142 Lysosome</td>
<td>0.930</td>
</tr>
<tr>
<td>hsa04620 Toll-like receptor signaling pathway</td>
<td>0.894</td>
</tr>
<tr>
<td>hsa04130 SNARE interactions in vesicular transport</td>
<td>0.888</td>
</tr>
<tr>
<td>hsa04612 Antigen processing and presentation</td>
<td>0.764</td>
</tr>
<tr>
<td>hsa03060 Protein export</td>
<td>0.773</td>
</tr>
<tr>
<td>hsa04660 T cell receptor signaling pathway</td>
<td>-1.418</td>
</tr>
<tr>
<td>hsa04630 Jak-STAT signaling pathway</td>
<td>-1.148</td>
</tr>
<tr>
<td>hsa04310 Wnt signaling pathway</td>
<td>-1.133</td>
</tr>
<tr>
<td>hsa04070 Phosphatidylinositol signaling system</td>
<td>-1.106</td>
</tr>
<tr>
<td>hsa04662 B cell receptor signaling pathway</td>
<td>-0.937</td>
</tr>
<tr>
<td>hsa04144 Endocytosis</td>
<td>-0.884</td>
</tr>
<tr>
<td>hsa04012 ErbB signaling pathway</td>
<td>-0.853</td>
</tr>
<tr>
<td>hsa04720 Long-term potentiation</td>
<td>-0.827</td>
</tr>
<tr>
<td>hsa00562 Inositol phosphate metabolism</td>
<td>-0.818</td>
</tr>
<tr>
<td>hsa04742 Taste transduction</td>
<td>-0.810</td>
</tr>
<tr>
<td>hsa00532 Glycosaminoglycan biosynthesis - chondroitin sulfate</td>
<td>-0.779</td>
</tr>
<tr>
<td>hsa04010 MAPK signaling pathway</td>
<td>-0.765</td>
</tr>
<tr>
<td>hsa04330 Notch signaling pathway</td>
<td>-0.766</td>
</tr>
<tr>
<td>hsa04960 Aldosterone-regulated sodium reabsorption</td>
<td>-0.748</td>
</tr>
<tr>
<td>hsa00514 Other types of O-glycan biosynthesis</td>
<td>-0.740</td>
</tr>
<tr>
<td>hsa00563 Glycosylphosphatidylinositol(GPI)-anchor biosynthesis</td>
<td>-0.726</td>
</tr>
<tr>
<td>hsa03040 Spliceosome</td>
<td>-0.720</td>
</tr>
<tr>
<td>hsa03013 RNA transport</td>
<td>-0.712</td>
</tr>
<tr>
<td>hsa04910 Insulin signaling pathway</td>
<td>-0.706</td>
</tr>
<tr>
<td>hsa04520 Adherens junction</td>
<td>-0.689</td>
</tr>
<tr>
<td>hsa04110 Cell cycle</td>
<td>-0.678</td>
</tr>
<tr>
<td>hsa00270 Cysteine and methionine metabolism</td>
<td>-0.662</td>
</tr>
<tr>
<td>hsa04062 Chemokine signaling pathway</td>
<td>-0.640</td>
</tr>
</tbody>
</table>
Table 10.10: **Upregulated pathways in TB (HIV-1 infected).** Colours: *red* = also upregulated in TB without HIV

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa04130 SNARE interactions in vesicular transport</td>
<td>1.219</td>
</tr>
<tr>
<td>hsa04620 Toll-like receptor signaling pathway</td>
<td>1.136</td>
</tr>
<tr>
<td>hsa04380 Osteoclast differentiation</td>
<td>1.133</td>
</tr>
<tr>
<td>hsa00190 Oxidative phosphorylation</td>
<td>1.096</td>
</tr>
<tr>
<td>hsa04610 Complement and coagulation cascades</td>
<td>1.082</td>
</tr>
<tr>
<td>hsa00564 Glycerophospholipid metabolism</td>
<td>0.968</td>
</tr>
<tr>
<td>hsa04140 Regulation of autophagy</td>
<td>0.923</td>
</tr>
<tr>
<td>hsa00760 Nicotinate and nicotinamide metabolism</td>
<td>0.898</td>
</tr>
<tr>
<td>hsa04966 Collecting duct acid secretion</td>
<td>0.887</td>
</tr>
<tr>
<td>hsa00600 Sphingolipid metabolism</td>
<td>0.711</td>
</tr>
<tr>
<td>hsa00770 Pantothenate and CoA biosynthesis</td>
<td>0.715</td>
</tr>
<tr>
<td>hsa04740 Olfactory transduction</td>
<td>0.663</td>
</tr>
<tr>
<td>hsa04976 Bile secretion</td>
<td>0.623</td>
</tr>
<tr>
<td>hsa02010 ABC transporters</td>
<td>0.622</td>
</tr>
<tr>
<td>hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series</td>
<td>0.623</td>
</tr>
<tr>
<td>hsa04142 Lysosome</td>
<td>0.622</td>
</tr>
<tr>
<td>hsa00512 Mucin type O-Glycan biosynthesis</td>
<td>0.604</td>
</tr>
<tr>
<td>hsa04920 Adipocytokine signaling pathway</td>
<td>0.570</td>
</tr>
<tr>
<td>hsa04145 Phagosome</td>
<td>0.561</td>
</tr>
<tr>
<td>hsa03320 PPAR signaling pathway</td>
<td>0.532</td>
</tr>
<tr>
<td>hsa00980 Metabolism of xenobiotics by cytochrome P450</td>
<td>0.511</td>
</tr>
<tr>
<td>hsa04260 Cardiac muscle contraction</td>
<td>0.509</td>
</tr>
<tr>
<td>hsa04614 Renin-angiotensin system</td>
<td>0.484</td>
</tr>
<tr>
<td>hsa00982 Drug metabolism - cytochrome P450</td>
<td>0.462</td>
</tr>
<tr>
<td>hsa00900 Terpenoid backbone biosynthesis</td>
<td>0.456</td>
</tr>
<tr>
<td>hsa03022 Basal transcription factors</td>
<td>0.443</td>
</tr>
<tr>
<td>hsa00480 Glutathione metabolism</td>
<td>0.422</td>
</tr>
<tr>
<td>hsa04975 Fat digestion and absorption</td>
<td>0.416</td>
</tr>
<tr>
<td>hsa04210 Apoptosis</td>
<td>0.409</td>
</tr>
<tr>
<td>hsa00531 Glycosaminoglycan degradation</td>
<td>0.370</td>
</tr>
<tr>
<td>hsa00590 Arachidonic acid metabolism</td>
<td>0.352</td>
</tr>
<tr>
<td>hsa00910 Nitrogen metabolism</td>
<td>0.344</td>
</tr>
<tr>
<td>hsa04710 Circadian rhythm - mammal</td>
<td>0.333</td>
</tr>
<tr>
<td>hsa04621 NOD-like receptor signaling pathway</td>
<td>0.314</td>
</tr>
<tr>
<td>hsa0380 Tryptophan metabolism</td>
<td>0.313</td>
</tr>
</tbody>
</table>
Table 10.11: **Downregulated pathways in TB (HIV-1 infected)**. Colours: green = also down in TB without HIV, yellow = upregulated in TB without HIV.

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa03040 Spliceosome</td>
<td>-1.779</td>
</tr>
<tr>
<td>hsa03013 RNA transport</td>
<td>-1.501</td>
</tr>
<tr>
<td>hsa04660 T cell receptor signaling pathway</td>
<td>-1.384</td>
</tr>
<tr>
<td>hsa00970 Aminoacyl-tRNA biosynthesis</td>
<td>-1.259</td>
</tr>
<tr>
<td>hsa00290 Valine, leucine and isoleucine biosynthesis</td>
<td>-1.220</td>
</tr>
<tr>
<td>hsa03008 Ribosome biogenesis in eukaryotes</td>
<td>-1.154</td>
</tr>
<tr>
<td>hsa00270 Cysteine and methionine metabolism</td>
<td>-1.084</td>
</tr>
<tr>
<td>hsa03030 DNA replication</td>
<td>-1.079</td>
</tr>
<tr>
<td>hsa04110 Cell cycle</td>
<td>-1.029</td>
</tr>
<tr>
<td>hsa03018 RNA degradation</td>
<td>-1.024</td>
</tr>
<tr>
<td>hsa04612 Antigen processing and presentation</td>
<td>-0.851</td>
</tr>
<tr>
<td>hsa00620 Pyruvate metabolism</td>
<td>-0.830</td>
</tr>
<tr>
<td>hsa04520 Adherens junction</td>
<td>-0.822</td>
</tr>
<tr>
<td>hsa00020 Citrate cycle (TCA cycle)</td>
<td>-0.793</td>
</tr>
<tr>
<td>hsa04662 B cell receptor signaling pathway</td>
<td>-0.753</td>
</tr>
<tr>
<td>hsa00640 Propanoate metabolism</td>
<td>-0.747</td>
</tr>
<tr>
<td>hsa00532 Glycosaminoglycan biosynthesis - chondroitin sulfate</td>
<td>-0.729</td>
</tr>
<tr>
<td>hsa00010 Glycolysis / Gluconeogenesis</td>
<td>-0.707</td>
</tr>
<tr>
<td>hsa04514 Cell adhesion molecules (CAMs)</td>
<td>-0.700</td>
</tr>
<tr>
<td>hsa00280 Valine, leucine and isoleucine degradation</td>
<td>-0.687</td>
</tr>
<tr>
<td>hsa04672 Intestinal immune network for IgA production</td>
<td>-0.668</td>
</tr>
<tr>
<td>hsa04310 Wnt signaling pathway</td>
<td>-0.651</td>
</tr>
<tr>
<td>hsa00250 Alanine, aspartate and glutamate metabolism</td>
<td>-0.644</td>
</tr>
<tr>
<td>hsa03010 Ribosome</td>
<td>-0.665</td>
</tr>
<tr>
<td>hsa00670 One carbon pool by folate</td>
<td>-0.509</td>
</tr>
<tr>
<td>hsa00330 Lysine degradation</td>
<td>-0.478</td>
</tr>
<tr>
<td>hsa04114 Oocyte meiosis</td>
<td>-0.476</td>
</tr>
<tr>
<td>hsa03015 mRNA surveillance pathway</td>
<td>-0.472</td>
</tr>
<tr>
<td>hsa03430 Mismatch repair</td>
<td>-0.468</td>
</tr>
<tr>
<td>hsa04720 Long-term potentiation</td>
<td>-0.465</td>
</tr>
<tr>
<td>hsa00051 Fructose and mannose metabolism</td>
<td>-0.457</td>
</tr>
<tr>
<td>hsa04062 Chemokine signaling pathway</td>
<td>-0.447</td>
</tr>
<tr>
<td>hsa04070 Phosphatidylinositol signaling system</td>
<td>-0.435</td>
</tr>
<tr>
<td>hsa04540 Gap junction</td>
<td>-0.413</td>
</tr>
<tr>
<td>hsa00030 Pentose phosphate pathway</td>
<td>-0.418</td>
</tr>
<tr>
<td>hsa00410 beta-Alanine metabolism</td>
<td>-0.406</td>
</tr>
<tr>
<td>hsa00350 Tyrosine metabolism</td>
<td>-0.392</td>
</tr>
<tr>
<td>hsa04530 Tight junction</td>
<td>-0.385</td>
</tr>
<tr>
<td>hsa04330 Notch signaling pathway</td>
<td>-0.372</td>
</tr>
<tr>
<td>hsa04510 Focal adhesion</td>
<td>-0.365</td>
</tr>
<tr>
<td>hsa00360 Phenylalanine metabolism</td>
<td>-0.365</td>
</tr>
<tr>
<td>hsa04960 Aldosterone-regulated sodium reabsorption</td>
<td>-0.359</td>
</tr>
<tr>
<td>hsa00040 Pentose and glucuronate interconversions</td>
<td>-0.357</td>
</tr>
<tr>
<td>hsa04742 Taste transduction</td>
<td>-0.350</td>
</tr>
<tr>
<td>hsa00920 Sulfur metabolism</td>
<td>-0.354</td>
</tr>
<tr>
<td>hsa04640 Hematopoietic cell lineage</td>
<td>-0.346</td>
</tr>
<tr>
<td>hsa04744 Phototransduction</td>
<td>-0.335</td>
</tr>
<tr>
<td>hsa00630 Glyoxylate and dicarboxylate metabolism</td>
<td>-0.336</td>
</tr>
<tr>
<td>hsa04650 Natural killer cell mediated cytotoxicity</td>
<td>-0.322</td>
</tr>
<tr>
<td>hsa03410 Base excision repair</td>
<td>-0.312</td>
</tr>
<tr>
<td>hsa04350 TGF-beta signaling pathway</td>
<td>-0.306</td>
</tr>
<tr>
<td>hsa00340 Histidine metabolism</td>
<td>-0.303</td>
</tr>
<tr>
<td>hsa03450 Non-homologous end-joining</td>
<td>-0.300</td>
</tr>
<tr>
<td>hsa04722 Neurotrophin signaling pathway</td>
<td>-0.275</td>
</tr>
</tbody>
</table>
Figure 10.17 demonstrates the overlap in up- and downregulated pathways in active tuberculosis in HIV-1 infected and uninfected individuals. 7 and 15 pathways are up and downregulated in active TB, respectively, regardless of HIV status. Note that one pathway (hsa04612 Antigen processing and presentation) that is upregulated in HIV-1 uninfected individuals is downregulated in HIV-1 co-infection. It is clear that in addition to the pathways up- or downregulated in active tuberculosis in HIV-1 uninfected individuals, HIV-1 co-infection results in additional differential pathway regulation in the presence of active tuberculosis. It is unclear whether this association is causal, and if so, in which direction.

Broadly speaking, upregulated pathways are concerned with meeting increased energy requirements in tuberculosis (oxidative phosphorylation), phagocytosis and downstream events like vesicular transport (phagosome, lysosome) and innate immune responses (Toll-like receptor signalling and complement pathways). In contrast, downregulated pathways may indicate possible mechanisms by which *Mycobacterium tuberculosis* evades internalisation and killing (e.g endocytosis, chemokine signalling) and subverts the host metabolic response to one more favourable to its survival (four metabolic pathways are downregulated). Most pathways up- or downregulated in HIV-1 uninfected individuals show the same pattern of regulation in HIV-1 infection (see Tables 10.10 and 10.11). Multiple additional pathways are also up- or downregulated in HIV-1 co-infected individuals, including additional metabolic pathways related to lipid metabolism, apoptosis and natural killer cell mediated cytotoxicity. The latter is of interest as HIV-1 has been shown to affect NK cell activation by Mycobacteria [161].
Figure 10.18: **Significant pathways in tuberculosis (HIV-1 uninfected).** The heatmap shows mean estimates for pathway expression for all active TB samples (not active TB samples are not shown). Colours: *red* = pathway upregulated; *green* = pathway downregulated. *(Note: Pathway names are cut off by the heatmap rendering algorithm; some manual editing of the source is required to fix this. The relevant tables provide the full pathway names)*
Figure 10.19: Significant pathways in tuberculosis (HIV-1 infected). The heatmap shows mean estimates for pathway expression for all active TB samples (not active TB samples are not shown). Colours: red = pathway upregulated; green = pathway downregulated.
10.5.1 Pathway visualisation

Figure 10.20 shows details of the Toll-like receptor pathway as an example of a pathway reported as upregulated in active tuberculosis in both HIV-1 uninfected and -infected individuals. While overall the pathway is reported as upregulated relative to controls, it appears that the biological effects of this upregulation may differ, as different individual genes in this pathway are over- or underexpressed. In Figure 10.21 we can see the effect of tuberculosis on the antigen processing and presentation pathway. In HIV-1 uninfected individuals, the MHC class I pathway is upregulated, while in HIV-1 infected individuals, the MHC class II pathway appears to be downregulated. This sheds some light on potential pathogenetic mechanisms in HIV-TB co-infection. Finally, Figure 10.22 shows the NK cell cytotoxicity pathway in blood for HIV-1 uninfected individuals. The set of top 100 probes called as differentially expressed in NK cells using cell-type specific differential expression analysis was intersected with the top 2000 differentially expressed probes in the standard differential expression analysis in order to provide logfold change estimates for the NK cell “specific” probes. Three of these probes are mapped to genes in the NK cell cytotoxicity pathway; two of these lie downstream of NKp30, NKp46 and FcγRIII receptors, suggesting that these receptors are involved in activating NK cells in tuberculosis. TNF-related apoptosis inducing ligand (TRAIL) is also upregulated in NK cells in tuberculosis, suggesting the effector pathway by which these NK cells may mediate their effect by inducing apoptosis in cells infected with *M. tuberculosis*. These results suggest numerous downstream experiments for future work.
Figure 10.20: **Toll-like receptor pathway.** Canonical pathway *hsa04620* from KEGG, rendered by *pathview* software. The log-fold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in active tuberculosis; *green* = downregulated in active tuberculosis.
Figure 10.21: Antigen processing and presentation pathway. Canonical pathway hsa04612 from KEGG, rendered by pathview software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: red = upregulated in active tuberculosis; green = downregulated in active tuberculosis.
Figure 10.22: NK cell cytotoxicity pathway (HIV-1 uninfected). Canonical pathway hsa04650, rendered using pathview. Genes associated with differential expression in NK cells and shown to be differentially expressed globally in blood are highlighted in colour. Colours: red = upregulated in active tuberculosis; green = downregulated in tuberculosis. See text for interpretation.
Chapter overview

This Chapter deals with questions relating to the effect of HIV-1 infection in two contexts: not active tuberculosis and active tuberculosis.

11.1 Question 5: HIV (not active TB)

The effect of HIV-1 co-infection on tuberculosis is complex, and causal relationships are not always clear. In order to isolate the effect of HIV-1 infection from the effect of *M. tuberculosis* disease, I present an overview of the analysis of two datasets: blood, LTBI and blood, healthy.

11.1.1 Included patients and samples

Table 11.1 shows that the healthy and LTBI subsets of the class not active TB were matched for age, sex and CD4 count. Both datasets were individually matched for age for the contrast HIV negative vs HIV positive. In both cases, a higher proportion of female individuals constituted the HIV-1 infected class.

11.1.2 Overall results

The overall results (Table 11.2) show similar numbers of differentially expressed probes prior to multiple testing correction, with no or few (6) probes significant after multiple testing. This is likely due to the small number of included samples, and the results should be treated with reserve.
Table 11.1: Clinical characteristics: HIV positive vs HIV negative (not active TB). Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant. This table includes data for the combined group (“not active TB”), LTBI and healthy.

<table>
<thead>
<tr>
<th>Age</th>
<th>Blood not activeTB HIVpos-HIVneg Median</th>
<th>LQ</th>
<th>UQ</th>
<th>Blood LTBI HIVpos-HIVneg Median</th>
<th>LQ</th>
<th>UQ</th>
<th>Blood Healthy HIVpos- HIVneg Median</th>
<th>LQ</th>
<th>UQ</th>
<th>Pval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>age_</td>
<td>29.5</td>
<td>25</td>
<td>36.7</td>
<td>29.3</td>
<td>24.22</td>
<td>35.3</td>
<td>29.8</td>
<td>25.5</td>
<td>38.55</td>
<td>0.844</td>
</tr>
<tr>
<td>neg</td>
<td>age_</td>
<td>29.9</td>
<td>22.98</td>
<td>34.2</td>
<td>29.1</td>
<td>22.7</td>
<td>34.7</td>
<td>30.7</td>
<td>23.8</td>
<td>32.7</td>
<td>0.994</td>
</tr>
<tr>
<td>pos</td>
<td>age_</td>
<td>29.5</td>
<td>25.75</td>
<td>37.3</td>
<td>29.3</td>
<td>25.5</td>
<td>35.5</td>
<td>29.5</td>
<td>26.08</td>
<td>39.47</td>
<td>0.829</td>
</tr>
<tr>
<td>P value</td>
<td>age_All</td>
<td>0.584</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sex_</td>
<td>sex_All</td>
<td>3.625</td>
<td>29</td>
<td>8</td>
<td>3.300</td>
<td>14</td>
<td>4</td>
<td>3.750</td>
<td>15</td>
<td>4</td>
<td>1.000</td>
</tr>
<tr>
<td>neg</td>
<td>sex_</td>
<td>0.250</td>
<td>2</td>
<td>8</td>
<td>0.250</td>
<td>1</td>
<td>4</td>
<td>0.250</td>
<td>1</td>
<td>4</td>
<td>1.000</td>
</tr>
<tr>
<td>pos</td>
<td>sex_</td>
<td>Inf</td>
<td>27</td>
<td>0</td>
<td>Inf</td>
<td>13</td>
<td>0</td>
<td>Inf</td>
<td>14</td>
<td>0</td>
<td>1.000</td>
</tr>
<tr>
<td>P value</td>
<td>sex_All</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sex_</td>
<td>2-sample test for eq</td>
<td>2-sample test for eq</td>
<td>2-sample test for eq</td>
<td>2-sample test for eq</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD4</td>
<td>CD4:</td>
<td>333</td>
<td>307.5</td>
<td>467</td>
<td>333</td>
<td>316</td>
<td>416</td>
<td>372</td>
<td>273</td>
<td>542.5</td>
<td>0.981</td>
</tr>
<tr>
<td>All</td>
<td>CD4_</td>
<td>333</td>
<td>307.5</td>
<td>467</td>
<td>333</td>
<td>316</td>
<td>416</td>
<td>372</td>
<td>273</td>
<td>542.5</td>
<td>0.981</td>
</tr>
<tr>
<td>neg</td>
<td>CD4_</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>pos</td>
<td>CD4_P</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>Test</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>
The top 300 differentially expressed probes based on log-likelihood of differential expression were used for subsequent analysis.

Table 11.2: Overall results: HIV positive vs HIV negative (not active TB). Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 5: HIV (no TB)</th>
<th>Contrast: HIV neg vs. HIV pos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dataset</td>
</tr>
<tr>
<td></td>
<td>Context</td>
</tr>
<tr>
<td>N: HIV neg</td>
<td></td>
</tr>
<tr>
<td>N: HIV pos</td>
<td></td>
</tr>
<tr>
<td><strong>Analysis 1: Differential expression</strong></td>
<td></td>
</tr>
<tr>
<td>N significant probes (unadjusted)</td>
<td>2299</td>
</tr>
<tr>
<td>N significant probes (BH adjusted)</td>
<td>0</td>
</tr>
<tr>
<td>N probes used (heatmaps, csDE)</td>
<td>300</td>
</tr>
<tr>
<td><strong>Analysis 2: Deconvolution</strong></td>
<td></td>
</tr>
<tr>
<td>Deconvolution successful</td>
<td>yes</td>
</tr>
<tr>
<td>N of N significant for contrast (BH)</td>
<td>2 of 13</td>
</tr>
<tr>
<td>N of N PBMC significant for contrast (BH)</td>
<td>2 of 12</td>
</tr>
<tr>
<td>Cell-specific DE successful</td>
<td>yes</td>
</tr>
<tr>
<td>Number of cell types</td>
<td>4</td>
</tr>
<tr>
<td>N cell types that reach FDR &lt; 0.4</td>
<td>2</td>
</tr>
<tr>
<td><strong>Analysis 3: WGCNA</strong></td>
<td></td>
</tr>
<tr>
<td>$R^2$ cutoff for scale-free approximation</td>
<td>0.8</td>
</tr>
<tr>
<td>Soft threshold power</td>
<td>6</td>
</tr>
<tr>
<td>Number of modules</td>
<td>23</td>
</tr>
<tr>
<td>Probes assigned to modules (N, %)</td>
<td>7968 (99.6)</td>
</tr>
<tr>
<td>GO: Entrez IDs submitted</td>
<td>4114</td>
</tr>
<tr>
<td>GO: Entrez IDs mapped</td>
<td>2720</td>
</tr>
<tr>
<td>N (%) modules cor GS/MM sig</td>
<td>14 (61)</td>
</tr>
<tr>
<td>N (%) modules sig for contrast (BH)</td>
<td>23 (100)</td>
</tr>
</tbody>
</table>

11.1.3 Results for context: not Active TB (datasets: Blood Healthy, Blood LTBI)

Figure 11.1 shows reasonable clustering of the input data from HIV-1 infected and -uninfected samples for the top 300 differentially expressed probes. This shows that there probably is a specific signal of HIV-1 infection, but requires larger numbers to detect reliably. Cell-type specific differential expression was detectable, and shows different results depending on whether prior immune sensitisation to *M. tuberculosis* had occurred (LTBI). In both groups, NK cell and neutrophil sig-
Contrasts involving HIV infection status

Signals are evident, based on the FDR plots shown in Figure 11.2. In addition, “antigen presenting cells” (comprising monocytes and dendritic cells), show a significant FDR drop in HIV-1 uninfected samples, while lymphocytes show possible signal in HIV-1 infected individuals. Please note that due to small sample numbers in this analysis we show cell-type specific responses for groups of cells. Module detection was similar in both groups as shown by the module detection plots in Figure 11.3. Surprisingly, very few probes of either the top 300 or top 500 differentially expressed probes in either set overlap. A number of possible explanations for exist, but remain speculative given the high probability of false positives in the probes shown.
11.1 Question 5: HIV (not active TB)

(a) Heatmap of differentially expressed probes, Healthy

(b) Heatmap of differentially expressed probes, LTBI

Figure 11.1: Results for contrast HIV status in contexts Healthy and LTBI. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: HIV negative, red: HIV positive. Plot titles reflect the number of included probes.
Figure 11.2: Cell-type specific differential expression.
11.1 Question 5: HIV (not active TB)

Figure 11.3: Module clustering.

(a) Module clustering Healthy

(b) Module clustering LTBI

Figure 11.4: Overlap of differentially expressed probes.

(a) Overlap of significantly differentially expressed probes

(b) Overlap of top 500 differentially expressed probes

235
11.2 Question 6: HIV (active TB)

11.2.1 Included patients and samples

This subsection addresses HIV-1 and Mycobacterium tuberculosis co-infection. Four datasets were included in this analysis: “blood, active TB”, “pericardial fluid, active TB-PC”, “blood, PTB” and “blood, active TB-PC”. Table 11.3 shows the clinical characteristics of the included study subjects. Between datasets, the variables sex (HIV-positive) and CD4 count (HIV-positive) were not matched. Within the two datasets for which results are presented in this thesis (“pericardial fluid, active TB-PC” and “blood, active TB-PC”, the variables age and CD4 count are not matched. The HIV-1 infected groups were younger and had lower CD4 counts, in keeping with a higher risk for extrapulmonary TB at a younger age driven by lower CD4 counts.

11.2.2 Overall results

The overall results (Table 11.4) show similar numbers of differentially expressed probes prior to multiple testing correction, with no probes significant after multiple testing. This is likely due to the small number of included samples, and the results should be treated with reserve. The top 300 differentially expressed probes based on log-likelihood of differential expression were used for subsequent analysis.

11.2.3 Results for context: Compartment (datasets: Blood TB-PC, Fluid TB-PC)

Figure 11.5 shows reasonable clustering of the input data from HIV-1 infected and -uninfected samples for the top 300 differentially expressed probes in both blood and pericardial fluid. In both cases, the HIV-1 uninfected individuals are assigned to a subcluster of HIV-1 infected individuals. This shows that there possibly is a specific signal of HIV-1 infection, but requires larger numbers to detect reliably, and this signal may be influenced by active tuberculosis. Cell-type specific differential expression was detectable, and shows different results depending on compartment. In blood, the strongest signal is found in neutrophils, while in pericardial fluid all four groups of cells show evidence of cell-specific differential expression (see Figure 11.6). Again, due to small sample num-
Table 11.3: Clinical characteristics: HIV positive vs HIV negative (active TB). Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Median LQ UQ</td>
<td>Median LQ UQ</td>
<td>Median LQ UQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_All</td>
<td>31.37 29.84 32.33</td>
<td>31.13 30.9 31.51</td>
<td>31.95 31.33 31.97</td>
<td>0.584</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>age_negative</td>
<td>40.67 33.69 52.36</td>
<td>53.92 35.95 56.07</td>
<td>31.85 31.78 30.6</td>
<td>0.425</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>age_positive</td>
<td>31.3 29.6 38.6</td>
<td>31.5 28.31 37.53</td>
<td>34.25 28.12 42.27</td>
<td>0.103</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>age_Pval</td>
<td>0.065</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_Ptest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sex</td>
<td>ratio F/M Female Male</td>
<td>ratio F/M Female Male</td>
<td>ratio F/M Female Male</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sex_All</td>
<td>1.148 31 27</td>
<td>0.965 13 17</td>
<td>3.667 11 3</td>
<td>0.173</td>
<td>Fisher's Exact Test</td>
</tr>
<tr>
<td>sex_negative</td>
<td>0.714 5 7</td>
<td>1.386 3 2</td>
<td>2.343 1 3</td>
<td>0.814</td>
<td>Fisher's Exact Test</td>
</tr>
<tr>
<td>sex_positive</td>
<td>1.00 26 30</td>
<td>0.667 10 15</td>
<td>Inf 10 0</td>
<td>0.003</td>
<td>Fisher's Exact Test</td>
</tr>
<tr>
<td>sex_Pval</td>
<td>0.452</td>
<td>0.046</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sex_Ptest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD4</td>
<td>Median LQ UQ</td>
<td>Median LQ UQ</td>
<td>Median LQ UQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD4_All</td>
<td>210 92 0.45</td>
<td>111 86 273</td>
<td>322 267 135</td>
<td>0.016</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>CD4_negative</td>
<td>566 406.5 653</td>
<td>600.5 492.4 651</td>
<td>666 282 640</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>CD4_positive</td>
<td>167 86 291.3</td>
<td>108 81 184</td>
<td>322 267 135</td>
<td>0.001</td>
<td>Kruskal-Wallis rank</td>
</tr>
<tr>
<td>CD4_Pval</td>
<td>0.004</td>
<td>0.004</td>
<td>NA</td>
<td>0.002</td>
<td></td>
</tr>
<tr>
<td>CD4_Ptest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 11.4: **Overall results: HIV positive vs HIV negative (active TB).** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 6: HIV (active TB)</th>
<th>Dataset</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>N: HIV neg</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>N: HIV pos</td>
<td>32</td>
<td>25</td>
</tr>
</tbody>
</table>

**Analysis 1: Differential expression**

| N significant probes (unadjusted) | 2098 | 1778 |
| N significant probes (BH adjusted) | 0 | 0 |
| N probes used (heatmaps, csDE) | 300 | 300 |

**Analysis 2: Deconvolution**

| Deconvolution successful | yes | yes |
| N of N significant for contrast (BH) | 0 of 13 | 1 of 13 |
| N of N PBMC significant for contrast (BH) | 0 of 12 | 1 of 12 |
| Cell-specific DE successful | yes | yes |
| Number of cell types | 4 | 4 |
| N cell types that reach FDR < 0.4 | 0 | 2 |

**Analysis 3: WGCNA**

| R² cutoff for scale-free approximation | 0.8 | 0.8 |
| Soft threshold power | 6 | 7 |
| Number of modules | 20 | 16 |
| Probes assigned to modules (N, %) | 4768 (60) | 4425 (55) |
| GO: Entrez IDs submitted | 4100 | 4014 |
| GO: Entrez IDs mapped | 2698 | 2653 |
| N (%) modules cor GS/MM sig | 6 (30) | 8 (50) |
| N (%) modules sig for contrast (BH) | 19 (95) | 12 (75) |
bers in this analysis we show cell-type specific responses for groups of cells. Module detection was similar in both groups as shown by the module detection plots in Figure 11.7. A large number of probes were not assigned to modules. Very few probes of either the top 300 or top 500 differentially expressed probes in either set overlap. This is not surprising, as the site of disease has been shown in previous work to cause changes in HIV-1 viral dynamics relative to blood [49].
11 Contrasts involving HIV infection status

Figure 11.5: Results for contrast HIV status in contexts TB-PC Blood and TB-PC Fluid. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: HIV negative, red: HIV positive. Plot titles reflect the number of included probes.
11.2 Question 6: HIV (active TB)

(a) False discovery rate plot blood

(b) False discovery rate plot pericardial fluid

Figure 11.6: Cell-type specific differential expression.
11 Contrasts involving HIV infection status

Figure 11.7: Module clustering.

(a) Module clustering Blood
(b) Module clustering Fluid

Figure 11.8: Overlap of differentially expressed probes.

(a) Overlap of significantly differentially expressed probes
(b) Overlap of top 500 differentially expressed probes probes
11.3 Pathway analysis

Pathway analysis was performed as described in Section 3.8.8. Tables 11.5, 11.6, 11.7 and 11.8 list the pathways found to be significantly differentially regulated in HIV-1 infection in two groups: not active TB and active TB (see Sections 11.1 and 11.2). Figure 11.9 shows heatmaps for the differentially regulated pathways in HIV-1 infection.

Table 11.5: Upregulated pathways in HIV (not active TB). Colours: red = also upregulated in HIV-1 infection in the active TB group, yellow = downregulated in HIV-1 infection in the active TB group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa03050 Proteasome</td>
<td>1.490</td>
</tr>
<tr>
<td>hsa04612 Antigen processing and presentation</td>
<td>1.305</td>
</tr>
<tr>
<td>hsa00190 Oxidative phosphorylation</td>
<td>1.212</td>
</tr>
<tr>
<td>hsa04672 Intestinal immune network for IgA production</td>
<td>0.868</td>
</tr>
<tr>
<td>hsa04623 Cytosolic DNA-sensing pathway</td>
<td>0.853</td>
</tr>
<tr>
<td>hsa03040 Spliceosome</td>
<td>0.790</td>
</tr>
<tr>
<td>hsa04622 RIG-I-like receptor signaling pathway</td>
<td>0.777</td>
</tr>
<tr>
<td>hsa03030 DNA replication</td>
<td>0.756</td>
</tr>
<tr>
<td>hsa04145 Phagosome</td>
<td>0.743</td>
</tr>
<tr>
<td>hsa04514 Cell adhesion molecules (CAMs)</td>
<td>0.727</td>
</tr>
<tr>
<td>hsa00970 Aminoacyl-tRNA biosynthesis</td>
<td>0.677</td>
</tr>
<tr>
<td>hsa00010 Glycolysis / Gluconeogenesis</td>
<td>0.662</td>
</tr>
<tr>
<td>hsa00240 Pyrimidine metabolism</td>
<td>0.633</td>
</tr>
<tr>
<td>hsa04110 Cell cycle</td>
<td>0.623</td>
</tr>
<tr>
<td>hsa03008 Ribosome biogenesis in eukaryotes</td>
<td>0.624</td>
</tr>
<tr>
<td>hsa00020 Citrate cycle (TCA cycle)</td>
<td>0.602</td>
</tr>
<tr>
<td>hsa04260 Cardiac muscle contraction</td>
<td>0.543</td>
</tr>
<tr>
<td>hsa03060 Protein export</td>
<td>0.538</td>
</tr>
<tr>
<td>hsa00051 Fructose and mannose metabolism</td>
<td>0.489</td>
</tr>
<tr>
<td>hsa00100 Steroid biosynthesis</td>
<td>0.485</td>
</tr>
<tr>
<td>hsa00480 Glutathione metabolism</td>
<td>0.456</td>
</tr>
<tr>
<td>hsa00290 Valine, leucine and isoleucine biosynthesis</td>
<td>0.462</td>
</tr>
<tr>
<td>hsa00360 Phenylalanine metabolism</td>
<td>0.434</td>
</tr>
</tbody>
</table>
Table 11.6: **Downregulated pathways in HIV (not active TB).** Colours: green = also downregulated in HIV-1 infection in the active TB group, yellow = upregulated in HIV-1 infection in the active TB group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa04012 ErbB signaling pathway</td>
<td>-0.824</td>
</tr>
<tr>
<td>hsa04910 Insulin signaling pathway</td>
<td>-0.751</td>
</tr>
<tr>
<td>hsa04150 mTOR signaling pathway</td>
<td>-0.749</td>
</tr>
<tr>
<td>hsa04144 Endocytosis</td>
<td>-0.729</td>
</tr>
<tr>
<td>hsa04010 MAPK signaling pathway</td>
<td>-0.693</td>
</tr>
<tr>
<td>hsa04360 Axon guidance</td>
<td>-0.681</td>
</tr>
<tr>
<td>hsa04960 Aldosterone-regulated sodium reabsorption</td>
<td>-0.680</td>
</tr>
<tr>
<td>hsa00590 Arachidonic acid metabolism</td>
<td>-0.669</td>
</tr>
<tr>
<td>hsa03010 Ribosome</td>
<td>-0.694</td>
</tr>
<tr>
<td>hsa04630 Jak-STAT signaling pathway</td>
<td>-0.617</td>
</tr>
<tr>
<td>hsa04320 Dorso-ventral axis formation</td>
<td>-0.610</td>
</tr>
<tr>
<td>hsa04310 Wnt signaling pathway</td>
<td>-0.600</td>
</tr>
<tr>
<td>hsa04720 Long-term potentiation</td>
<td>-0.583</td>
</tr>
<tr>
<td>hsa04916 Melanogenesis</td>
<td>-0.557</td>
</tr>
<tr>
<td>hsa04722 Neurotrophin signaling pathway</td>
<td>-0.555</td>
</tr>
<tr>
<td>hsa04510 Focal adhesion</td>
<td>-0.538</td>
</tr>
<tr>
<td>hsa04666 Fc gamma R-mediated phagocytosis</td>
<td>-0.535</td>
</tr>
<tr>
<td>hsa04664 Fc epsilon RI signaling pathway</td>
<td>-0.506</td>
</tr>
<tr>
<td>hsa04070 Phosphatidylinositol signaling system</td>
<td>-0.505</td>
</tr>
<tr>
<td>hsa04710 Circadian rhythm - mammal</td>
<td>-0.507</td>
</tr>
<tr>
<td>hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series</td>
<td>-0.507</td>
</tr>
<tr>
<td>hsa04350 TGF-beta signaling pathway</td>
<td>-0.500</td>
</tr>
<tr>
<td>hsa04340 Hedgehog signaling pathway</td>
<td>-0.500</td>
</tr>
<tr>
<td>hsa04210 Apoptosis</td>
<td>-0.500</td>
</tr>
<tr>
<td>hsa04912 GnRH signaling pathway</td>
<td>-0.463</td>
</tr>
<tr>
<td>hsa04810 Regulation of actin cytoskeleton</td>
<td>-0.454</td>
</tr>
<tr>
<td>hsa04730 Long-term depression</td>
<td>-0.445</td>
</tr>
<tr>
<td>hsa04920 Adipocytokine signaling pathway</td>
<td>-0.443</td>
</tr>
<tr>
<td>hsa04370 VEGF signaling pathway</td>
<td>-0.430</td>
</tr>
<tr>
<td>hsa00562 Inositol phosphate metabolism</td>
<td>-0.430</td>
</tr>
<tr>
<td>hsa04740 Olfactory transduction</td>
<td>-0.422</td>
</tr>
<tr>
<td>hsa04330 Notch signaling pathway</td>
<td>-0.409</td>
</tr>
<tr>
<td>hsa04966 Collecting duct acid secretion</td>
<td>-0.409</td>
</tr>
</tbody>
</table>
### Table 11.7: Upregulated pathways in HIV (active TB)

Colours: red = also upregulated in HIV-1 infection in the not active TB group, yellow = downregulated in HIV-1 infection in the not active TB group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa04740 Olfactory transduction</td>
<td>0.769</td>
</tr>
<tr>
<td>hsa04623 Cytosolic DNA-sensing pathway</td>
<td>0.742</td>
</tr>
<tr>
<td>hsa04622 RIG-I-like receptor signaling pathway</td>
<td>0.714</td>
</tr>
<tr>
<td>hsa00240 Pyrimidine metabolism</td>
<td>0.623</td>
</tr>
<tr>
<td>hsa02010 ABC transporters</td>
<td>0.569</td>
</tr>
<tr>
<td>hsa00564 Glycerophospholipid metabolism</td>
<td>0.539</td>
</tr>
<tr>
<td>hsa04920 Adipocytokine signaling pathway</td>
<td>0.485</td>
</tr>
<tr>
<td>hsa00760 Nicotinate and nicotinamide metabolism</td>
<td>0.488</td>
</tr>
<tr>
<td>hsa04620 Toll-like receptor signaling pathway</td>
<td>0.458</td>
</tr>
<tr>
<td>hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series</td>
<td>0.439</td>
</tr>
<tr>
<td>hsa04975 Fat digestion and absorption</td>
<td>0.433</td>
</tr>
<tr>
<td>hsa04976 Bile secretion</td>
<td>0.396</td>
</tr>
<tr>
<td>hsa04380 Osteoclast differentiation</td>
<td>0.382</td>
</tr>
<tr>
<td>hsa04140 Regulation of autophagy</td>
<td>0.377</td>
</tr>
<tr>
<td>hsa00982 Drug metabolism - cytochrome P450</td>
<td>0.359</td>
</tr>
</tbody>
</table>

### Table 11.8: Downregulated pathways in HIV (active TB)

Colours: green = also downregulated in HIV-1 infection in the not active TB group, yellow = upregulated in HIV-1 infection in the not active TB group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa03013 RNA transport</td>
<td>-0.735</td>
</tr>
<tr>
<td>hsa00640 Propanoate metabolism</td>
<td>-0.581</td>
</tr>
<tr>
<td>hsa04810 Regulation of actin cytoskeleton</td>
<td>-0.560</td>
</tr>
<tr>
<td>hsa04510 Focal adhesion</td>
<td>-0.546</td>
</tr>
<tr>
<td>hsa00270 Cysteine and methionine metabolism</td>
<td>-0.531</td>
</tr>
<tr>
<td>hsa00970 Aminoacyl-tRNA biosynthesis</td>
<td>-0.532</td>
</tr>
<tr>
<td>hsa00290 Valine, leucine and isoleucine biosynthesis</td>
<td>-0.522</td>
</tr>
<tr>
<td>hsa00010 Glycolysis / Gluconeogenesis</td>
<td>-0.473</td>
</tr>
<tr>
<td>hsa00280 Valine, leucine and isoleucine degradation</td>
<td>-0.471</td>
</tr>
<tr>
<td>hsa03040 Spliceosome</td>
<td>-0.455</td>
</tr>
<tr>
<td>hsa00620 Pyruvate metabolism</td>
<td>-0.438</td>
</tr>
<tr>
<td>hsa04520 Adherens junction</td>
<td>-0.425</td>
</tr>
<tr>
<td>hsa04612 Antigen processing and presentation</td>
<td>-0.421</td>
</tr>
<tr>
<td>hsa00020 Citrate cycle (TCA cycle)</td>
<td>-0.408</td>
</tr>
<tr>
<td>hsa03060 Protein export</td>
<td>-0.394</td>
</tr>
<tr>
<td>hsa04670 Leukocyte transendothelial migration</td>
<td>-0.362</td>
</tr>
<tr>
<td>hsa04530 Tight junction</td>
<td>-0.354</td>
</tr>
<tr>
<td>hsa00410 beta-Alanine metabolism</td>
<td>-0.352</td>
</tr>
</tbody>
</table>
Figure 11.9: **Significant pathways in HIV**. The heatmaps show mean estimates for pathway expression for all HIV-1 infected samples (HIV-1 uninfected samples are not shown). Colours: *red* = pathway upregulated; *green* = pathway downregulated.
Figure 11.10 shows the extent to which pathways up- or downregulated by HIV-1 infection are shared by individuals with and without active tuberculosis. This provides additional information about possible disease mechanisms in HIV-TB. In contrast to the shared pathways affected by active tuberculosis, pathways affected by HIV show very little overlap in the two groups (not active TB and active TB). This suggests that active tuberculosis is an important co-regulator of HIV-1-related pathogenesis. In fact, more pathways are up- or downregulated in opposite directions depending on TB status than in the same direction. This is discussed further below.

![Venn diagram showing overlap of differentially regulated pathways in active tuberculosis stratified by HIV status.](image)

**Figure 11.10:** Overlap of differentially regulated pathways in active tuberculosis stratified by HIV status.

### 11.3.1 Pathway visualisation

The antigen processing and presentation pathway which is strongly upregulated by HIV-1 in the absence of active tuberculosis is downregulated in HIV-TB co-infection. Figure 11.11 shows the extent of this differential regulation at gene level. This should be compared to Figure 10.21 which demonstrates the converse situation, i.e. the effect of active tuberculosis on the same pathway, with and without HIV-1 co-infection.

Another interesting pathway concerns innate defences against viral infections mediated by the RIG-1-like pathway (hsa04622). This pathway is an important sensor for cytoplasmic viral RNA, essential for generating a downstream signal that results in type I interferon secretion. RIG-1 is induced by type 1 interferon signalling as well as double-stranded viral RNA. While purified HIV-1 RNA indeed produces a RIG-1 dependent interferon response, actual HIV-1 infection does not result in increased interferon production *in vitro*. This was shown to be due to inhibition of
11 Contrasts involving HIV infection status

the RIG-1 sensor (a DExD/H box RNA helicase) by HIV-1 protease by reducing phosphorylation of IFN regulatory factor 3 (IRF-3) phosphorylation [162]. The proximal pathway appears to be upregulated during HIV-1 infection in both active TB and not active TB contexts (see Figure 11.12), but IRF-3, however, is not upregulated. A possible interpretation of the pathway result is that failure of upregulating IRF-3 may impair the type I interferon response, which would relate to a previously published result, in which a similar picture of downregulation of IRF-3 activity, but not IRF-7 was reported in vitro. [163].

In summary, pathway analysis with mapping of differentially expressed genes onto differentially regulated pathways provides substantial insight into the complex relationships between HIV-1 and M. tuberculosis infections.
Figure 11.11: **Antigen processing and presentation pathway.** Canonical pathway hsa04612 from KEGG, rendered by *pathview* software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in HIV-1 infection; *green* = downregulated in HIV-1 infection.
Figure 11.12: RIG-I-like receptor signaling pathway. Canonical pathway hsa04622 from KEGG, rendered by pathview software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: red = upregulated in HIV-1 infection; green = downregulated in HIV-1 infection.
12 Contrasts involving the compartment

Chapter overview

In this Chapter I discuss the results of transcriptomic analysis at the site of active tuberculosis in individuals with tuberculous pericarditis.

12.1 Question 7: Blood and pericardial fluid in TB-PC

While blood transcriptomic responses are informative regarding the nature of the immune response to tuberculosis with and without HIV-1 co-infection, the site of disease is of considerable interest in studies of tuberculosis pathogenesis, as it is at the site of infection that the disease develops and progresses. Given our understanding of transcriptional responses detectable in blood, it is instructive to assess transcriptional responses in pericardial fluid relative to those in blood.

12.1.1 Included patients and samples

Five datasets were included in this analysis: all matched blood/pericardial fluid samples, and matched blood pericardial fluid samples for HIV-1 uninfected, HIV-1 infected, Effusive phenotype and effusive-constrictive phenotype subgroups\(^1\). All five sets/subsets were matched for age and sex; the HIV-1 infected subsets had significantly lower CD4 counts. Results for two contexts will be reported here (TB-PC HIV negative, TB-PC HIV positive). As only matched blood and pericardial fluid samples were included, there is no bias due to mismatched demographic or phenotypic variables, setting this analysis apart from all others.

\(^1\)The much smaller sample sizes in the haemodynamic phenotype subsets \((\text{EFF blood - fluid and EC blood - fluid})\) are due to the limited number of study subjects included in this study who had right heart studies at baseline.
Table 12.1: Clinical characteristics: blood vs pericardial fluid. Two-way comparison of demographic variables and CD4 count between and within groups of study participants. P-values less than 0.05 are regarded as significant. This table includes data for the combined group ("not active TB"), LTBI and healthy.

<table>
<thead>
<tr>
<th>Test</th>
<th>Age (median [LQ, UQ])</th>
<th>Sex (F/M)</th>
<th>CD4 (median [LQ, UQ])</th>
<th>P value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.08 [29.6, 38.7]</td>
<td>40/62</td>
<td>182 [86, 326]</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>33.08 [29.6, 38.7]</td>
<td>5/5</td>
<td>10 [86, 46]</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>33.08 [29.6, 38.7]</td>
<td>10/12</td>
<td>8 [8, 174]</td>
<td>0.000</td>
</tr>
</tbody>
</table>

*Note: All comparisons were done using the Kruskal-Wallis rank test for continuous variables and Fisher’s Exact Test for categorical variables.*
12.1.2 Overall results

Table 12.2 lists the overall results of the three analyses for two datasets. Of all analyses, the contrast “blood vs. pericardial fluid” yielded the highest number of significantly differentially expressed probes. This is likely due to the markedly different cellular composition of pericardial fluid compared to blood (see below). For this reason, cell-type specific differential expression is of great interest here. However, as I will show in the pathway analysis section, global transcriptional changes at the site of disease regardless of cellular origin are still informative about biological processes at the site of disease, which demonstrate a very strong signal indicative of major reprogramming of host metabolism, which is likely to benefit *M. tuberculosis*. Given the significant differences in cellular composition, the modules identified using weighted gene co-expression network analysis possibly represent cell-types. This requires further analysis.
Table 12.2: Overall results blood vs pericardial fluid. Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

<table>
<thead>
<tr>
<th>Question 7: Compartment</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>TB-PC HIV neg</td>
<td>N: Blood 10</td>
<td>N: Fluid 10</td>
</tr>
<tr>
<td>TB-PC HIV pos</td>
<td>N: Blood 41</td>
<td>N: Fluid 41</td>
</tr>
</tbody>
</table>

**Analysis 1: Differential expression**

| N significant probes (unadjusted) | 7895 | 12688 |
| N significant probes (BH adjusted) | 4211 | 10434 |
| N probes used (heatmaps, csDE) | 2000 | 2000 |

**Analysis 2: Deconvolution**

| Deconvolution successful | yes | yes |
| N of N significant for contrast (BH) | 5 of 15 | 8 of 15 |
| N of N PBMC significant for contrast (BH) | 5 of 14 | 9 of 14 |
| Cell-specific DE successful | yes | yes |
| Number of cell types | 7 | 7 |
| N cell types that reach FDR < 0.4 | 1 | 3 |

**Analysis 3: WGCNA**

| R² cutoff for scale-free approximation | 0.75 | 0.8 |
| Soft threshold power | 8 | 7 |
| Number of modules | 7 | 13 |
| Probes assigned to modules (N, %) | 8000 (100) | 7987 (99.84) |
| GO: Entrez IDs submitted | 4665 | 4875 |
| GO: Entrez IDs mapped | 3131 | 3275 |
| N (%) modules cor GS/MM sig | 6 | 12 |
| N (%) modules sig for contrast (BH) | 7 (100) | 13 (100) |
12.1.3 Results for context: HIV status (datasets: TB-PC HIV neg, TB-PC HIV pos)

The top 2000 differentially expressed probes in both HIV-1 uninfected and -infected individuals separate the input data into two distinct clusters (blood and pericardial fluid) (Figure 12.1). Of interest is the large overlap between HIV-1 infected and -uninfected individuals when considering the top 2000 differentially expressed probes; in contrast, there is very little overlap between the top 500 differentially expressed probes. Figure 12.2 shows the barplots for proportions of detected cell types. The most striking difference is the low proportion of neutrophils in pericardial fluid. This is in keeping with the most common presentation of tuberculous pericardial effusions as predominantly lymphocytic (see [139] for data in a South African population). As the neutrophils skew the proportions of all other cells (mainly PBMC), barplots of PBMC proportions (normalised to 1.0) are also shown.

Cell-specific differential expression differed between the two groups studied. In HIV-1 uninfected individuals, the strongest signal was detected in B cells, followed by CD8 T cells, with weaker signals in NK cells and dendritic cells. In contrast, HIV-1 infected individuals exhibited strong signals for cell-type specific differential expression in CD4 T cells, B cells, dendritic cells, and to a lesser extent also in NK cells and monocytes.

Virtually all probes examined were assigned to modules. These require further analysis to characterise and relate to functional classes like cell types and pathways.
12 Contrasts involving the compartment

Figure 12.1: Results for contrast *Compartment* in contexts *TB-PC HIV neg* and *TB-PC HIV pos*. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: red: *blood*, yellow: *pericardial fluid*. Plot titles reflect the number of included probes.
12.1 Question 7: Blood and pericardial fluid in TB-PC

Figure 12.2: Barplots and boxplots of matched blood and pericardial fluid samples. Stacked bar charts for cell proportions for the analysed samples are shown in subfigures a and b. Note the striking visual difference between blood and pericardial fluid. Box-and-whisker charts are shown for all detected cell-types, and for PBMC only in subfigures c, d and e, f, respectively. Each bar is normalised to 1 (100%). Samples are ordered according to disease phenotype (not active TB followed by active TB). The legend lists the cell types identified. Abbreviations: neutro: neutrophils, DC act: activated dendritic cells, DC: non-activated dendritic cells, mono: monocytes, NKact: activated natural killer cells, NK: non-activated natural killer cells, B alGM: BCR-ligated B cells, B: resting B cells, Tc act: activated CD8 T cell, Tc: resting CD8 T cell, Th act: activated CD4 T cell, Th: resting CD4 T cell.
Figure 12.3: Cell-type specific differential expression.
12.1 Question 7: Blood and pericardial fluid in TB-PC

Figure 12.4: **Module clustering.**

(a) Module clustering HIV neg

(b) Module clustering HIV pos

---

Figure 12.5: **Overlap of differentially expressed probes.**

(a) Overlap of significantly differentially expressed probes
(b) Overlap of top 500 differentially expressed probes
12.2 Pathway analysis

Given the massive differences in differential expression at probe level, pathway analysis is a useful method to reduce the data to more intuitive biological differences. Tables 12.3, 12.4, 12.5 and 12.6 list the pathways up- and downregulated in pericardial fluid (relative to blood) with and without HIV-1 co-infection. Pathway heatmaps are not shown,
Table 12.3: **Pathways significantly upregulated in pericardial fluid (HIV-1 uninfected).** Colours: red = also upregulated in pericardial fluid in the HIV-1 infected group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa00970 Aminoacyl-tRNA biosynthesis</td>
<td>2.808</td>
</tr>
<tr>
<td>hsa03008 Ribosome biogenesis in eukaryotes</td>
<td>2.587</td>
</tr>
<tr>
<td>hsa00010 Glycolysis / Gluconeogenesis</td>
<td>2.136</td>
</tr>
<tr>
<td>hsa03040 Spliceosome</td>
<td>2.054</td>
</tr>
<tr>
<td>hsa03030 DNA replication</td>
<td>2.106</td>
</tr>
<tr>
<td>hsa04142 Lysosome</td>
<td>2.068</td>
</tr>
<tr>
<td>hsa03050 Proteasome</td>
<td>1.966</td>
</tr>
<tr>
<td>hsa03013 RNA transport</td>
<td>1.896</td>
</tr>
<tr>
<td>hsa00020 Citrate cycle (TCA cycle)</td>
<td>1.932</td>
</tr>
<tr>
<td>hsa04612 Antigen processing and presentation</td>
<td>1.769</td>
</tr>
<tr>
<td>hsa00190 Oxidative phosphorylation</td>
<td>1.724</td>
</tr>
<tr>
<td>hsa00290 Valine, leucine and isoleucine biosynthesis</td>
<td>1.657</td>
</tr>
<tr>
<td>hsa04672 Intestinal immune network for IgA production</td>
<td>1.495</td>
</tr>
<tr>
<td>hsa0310 Lysine degradation</td>
<td>1.488</td>
</tr>
<tr>
<td>hsa00280 Valine, leucine and isoleucine degradation</td>
<td>1.364</td>
</tr>
<tr>
<td>hsa00640 Propanoate metabolism</td>
<td>1.365</td>
</tr>
<tr>
<td>hsa00270 Cysteine and methionine metabolism</td>
<td>1.355</td>
</tr>
<tr>
<td>hsa03018 RNA degradation</td>
<td>1.325</td>
</tr>
<tr>
<td>hsa00670 One carbon pool by folate</td>
<td>1.296</td>
</tr>
<tr>
<td>hsa03060 Protein export</td>
<td>1.308</td>
</tr>
<tr>
<td>hsa04141 Protein processing in endoplasmic reticulum</td>
<td>1.188</td>
</tr>
<tr>
<td>hsa00051 Fructose and mannose metabolism</td>
<td>1.201</td>
</tr>
<tr>
<td>hsa00510 N-Glycan biosynthesis</td>
<td>1.086</td>
</tr>
<tr>
<td>hsa00052 Galactose metabolism</td>
<td>1.038</td>
</tr>
<tr>
<td>hsa00520 Amino sugar and nucleotide sugar metabolism</td>
<td>0.921</td>
</tr>
<tr>
<td>hsa00511 Other glycan degradation</td>
<td>0.932</td>
</tr>
<tr>
<td>hsa03410 Base excision repair</td>
<td>0.878</td>
</tr>
<tr>
<td>hsa03010 Ribosome</td>
<td>0.889</td>
</tr>
<tr>
<td>hsa03020 RNA polymerase</td>
<td>0.820</td>
</tr>
<tr>
<td>hsa03430 Mismatch repair</td>
<td>0.814</td>
</tr>
<tr>
<td>hsa00620 Pyruvate metabolism</td>
<td>0.773</td>
</tr>
<tr>
<td>hsa00630 Glyoxylate and dicarboxylate metabolism</td>
<td>0.752</td>
</tr>
<tr>
<td>hsa00410 beta-Alanine metabolism</td>
<td>0.739</td>
</tr>
<tr>
<td>hsa00350 Tyrosine metabolism</td>
<td>0.732</td>
</tr>
<tr>
<td>hsa04260 Cardiac muscle contraction</td>
<td>0.730</td>
</tr>
<tr>
<td>hsa04146 Peroxisome</td>
<td>0.692</td>
</tr>
<tr>
<td>hsa00450 Selenocompound metabolism</td>
<td>0.675</td>
</tr>
<tr>
<td>hsa00100 Steroid biosynthesis</td>
<td>0.650</td>
</tr>
<tr>
<td>hsa00250 Alanine, aspartate and glutamate metabolism</td>
<td>0.638</td>
</tr>
</tbody>
</table>
12 Contrasts involving the compartment

Table 12.4: **Pathways significantly downregulated in pericardial fluid (HIV-1 uninfected).** Colours: green = also downregulated in pericardial fluid in the HIV-1 infected group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa04650 Natural killer cell mediated cytotoxicity</td>
<td>-2.537</td>
</tr>
<tr>
<td>hsa04670 Leukocyte transendothelial migration</td>
<td>-2.120</td>
</tr>
<tr>
<td>hsa04810 Regulation of actin cytoskeleton</td>
<td>-1.855</td>
</tr>
<tr>
<td>hsa04210 Apoptosis</td>
<td>-1.784</td>
</tr>
<tr>
<td>hsa04666 Fc gamma R-mediated phagocytosis</td>
<td>-1.607</td>
</tr>
<tr>
<td>hsa00564 Glycerophospholipid metabolism</td>
<td>-1.566</td>
</tr>
<tr>
<td>hsa04144 Endocytosis</td>
<td>-1.475</td>
</tr>
<tr>
<td>hsa04380 Osteoclast differentiation</td>
<td>-1.481</td>
</tr>
<tr>
<td>hsa04920 Adipocytokine signaling pathway</td>
<td>-1.338</td>
</tr>
<tr>
<td>hsa00910 Nitrogen metabolism</td>
<td>-1.343</td>
</tr>
<tr>
<td>hsa04740 Olfactory transduction</td>
<td>-1.312</td>
</tr>
<tr>
<td>hsa04966 Collecting duct acid secretion</td>
<td>-1.299</td>
</tr>
<tr>
<td>hsa00760 Nicotinate and nicotinamide metabolism</td>
<td>-1.202</td>
</tr>
<tr>
<td>hsa04010 MAPK signaling pathway</td>
<td>-1.135</td>
</tr>
<tr>
<td>hsa04270 Vascular smooth muscle contraction</td>
<td>-1.121</td>
</tr>
<tr>
<td>hsa04140 Regulation of autophagy</td>
<td>-1.118</td>
</tr>
<tr>
<td>hsa04664 Fc epsilon RI signaling pathway</td>
<td>-1.105</td>
</tr>
<tr>
<td>hsa04720 Long-term potentiation</td>
<td>-1.097</td>
</tr>
<tr>
<td>hsa04640 Hematopoietic cell lineage</td>
<td>-1.039</td>
</tr>
<tr>
<td>hsa00770 Pantothenate and CoA biosynthesis</td>
<td>-1.007</td>
</tr>
<tr>
<td>hsa04062 Chemokine signaling pathway</td>
<td>-0.953</td>
</tr>
<tr>
<td>hsa04360 Axon guidance</td>
<td>-0.933</td>
</tr>
<tr>
<td>hsa04070 Phosphatidylinositol signaling system</td>
<td>-0.904</td>
</tr>
<tr>
<td>hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series</td>
<td>-0.891</td>
</tr>
<tr>
<td>hsa04350 TGF-beta signaling pathway</td>
<td>-0.872</td>
</tr>
<tr>
<td>hsa04630 Jak-STAT signaling pathway</td>
<td>-0.838</td>
</tr>
<tr>
<td>hsa00562 Inositol phosphate metabolism</td>
<td>-0.837</td>
</tr>
<tr>
<td>hsa04730 Long-term depression</td>
<td>-0.824</td>
</tr>
<tr>
<td>hsa04614 Renin-angiotensin system</td>
<td>-0.817</td>
</tr>
<tr>
<td>hsa04530 Tight junction</td>
<td>-0.804</td>
</tr>
<tr>
<td>hsa04510 Focal adhesion</td>
<td>-0.786</td>
</tr>
<tr>
<td>hsa04520 Adherens junction</td>
<td>-0.774</td>
</tr>
<tr>
<td>hsa04310 Wnt signaling pathway</td>
<td>-0.714</td>
</tr>
<tr>
<td>hsa04662 B cell receptor signaling pathway</td>
<td>-0.687</td>
</tr>
<tr>
<td>hsa04370 VEGF signaling pathway</td>
<td>-0.657</td>
</tr>
<tr>
<td>hsa04115 p53 signaling pathway</td>
<td>-0.649</td>
</tr>
</tbody>
</table>
### 12.2 Pathway analysis

Table 12.5: Pathways significantly upregulated in pericardial fluid (HIV-1 infected). Colours:

- red = also upregulated in pericardial fluid in the HIV-1 uninfected group

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa04142 Lysosome</td>
<td>3.400</td>
</tr>
<tr>
<td>hsa00970 Aminoacyl-tRNA biosynthesis</td>
<td>2.944</td>
</tr>
<tr>
<td>hsa04612 Antigen processing and presentation</td>
<td>2.564</td>
</tr>
<tr>
<td>hsa03050 Proteasome</td>
<td>2.532</td>
</tr>
<tr>
<td>hsa00010 Glycolysis / Gluconeogenesis</td>
<td>2.300</td>
</tr>
<tr>
<td>hsa04190 Oxidative phosphorylation</td>
<td>2.236</td>
</tr>
<tr>
<td>hsa03008 Ribosome biogenesis in eukaryotes</td>
<td>2.147</td>
</tr>
<tr>
<td>hsa00020 Citrate cycle (TCA cycle)</td>
<td>2.112</td>
</tr>
<tr>
<td>hsa04672 Intestinal immune network for IgA production</td>
<td>2.056</td>
</tr>
<tr>
<td>hsa03040 Splicesome</td>
<td>1.704</td>
</tr>
<tr>
<td>hsa03030 DNA replication</td>
<td>1.728</td>
</tr>
<tr>
<td>hsa03013 RNA transport</td>
<td>1.666</td>
</tr>
<tr>
<td>hsa00290 Valine, leucine and isoleucine biosynthesis</td>
<td>1.770</td>
</tr>
<tr>
<td>hsa00310 Lysine degradation</td>
<td>1.620</td>
</tr>
<tr>
<td>hsa00280 Valine, leucine and isoleucine degradation</td>
<td>1.588</td>
</tr>
<tr>
<td>hsa09640 Propanoate metabolism</td>
<td>1.499</td>
</tr>
<tr>
<td>hsa00511 Other glycan degradation</td>
<td>1.514</td>
</tr>
<tr>
<td>hsa03060 Protein export</td>
<td>1.402</td>
</tr>
<tr>
<td>hsa00051 Fructose and mannose metabolism</td>
<td>1.364</td>
</tr>
<tr>
<td>hsa00670 One carbon pool by folate</td>
<td>1.333</td>
</tr>
<tr>
<td>hsa00520 Amino sugar and nucleotide sugar metabolism</td>
<td>1.273</td>
</tr>
<tr>
<td>hsa00270 Cysteine and methionine metabolism</td>
<td>1.225</td>
</tr>
<tr>
<td>hsa04141 Protein processing in endoplasmic reticulum</td>
<td>1.143</td>
</tr>
<tr>
<td>hsa00052 Galactose metabolism</td>
<td>1.151</td>
</tr>
<tr>
<td>hsa00510 N-Glycan biosynthesis</td>
<td>0.983</td>
</tr>
<tr>
<td>hsa00410 beta-Alanine metabolism</td>
<td>0.990</td>
</tr>
<tr>
<td>hsa00630 Glyoxylate and dicarboxylate metabolism</td>
<td>0.988</td>
</tr>
<tr>
<td>hsa04145 Phagosome</td>
<td>0.938</td>
</tr>
<tr>
<td>hsa00351 Glycosaminoglycan degradation</td>
<td>0.946</td>
</tr>
<tr>
<td>hsa00620 Pyruvate metabolism</td>
<td>0.906</td>
</tr>
<tr>
<td>hsa04146 Peroxisome</td>
<td>0.858</td>
</tr>
<tr>
<td>hsa04260 Cardiac muscle contraction</td>
<td>0.857</td>
</tr>
<tr>
<td>hsa00350 Tyrosine metabolism</td>
<td>0.853</td>
</tr>
<tr>
<td>hsa03018 RNA degradation</td>
<td>0.816</td>
</tr>
<tr>
<td>hsa03410 Base excision repair</td>
<td>0.810</td>
</tr>
<tr>
<td>hsa00480 Glutathione metabolism</td>
<td>0.762</td>
</tr>
<tr>
<td>hsa00360 Phenylalanine metabolism</td>
<td>0.751</td>
</tr>
<tr>
<td>hsa00071 Fatty acid metabolism</td>
<td>0.694</td>
</tr>
<tr>
<td>hsa00250 Alanine, aspartate and glutamate metabolism</td>
<td>0.694</td>
</tr>
<tr>
<td>hsa03430 Mismatch repair</td>
<td>0.685</td>
</tr>
<tr>
<td>hsa00450 Selenocompound metabolism</td>
<td>0.623</td>
</tr>
<tr>
<td>hsa00600 Sphingolipid metabolism</td>
<td>0.602</td>
</tr>
<tr>
<td>hsa04610 Complement and coagulation cascades</td>
<td>0.587</td>
</tr>
<tr>
<td>hsa00380 Tryptophan metabolism</td>
<td>0.575</td>
</tr>
<tr>
<td>hsa00340 Pentose and glucuronate interconversions</td>
<td>0.564</td>
</tr>
<tr>
<td>hsa09012 Primary bile acid biosynthesis</td>
<td>0.555</td>
</tr>
<tr>
<td>hsa03020 RNA polymerase</td>
<td>0.523</td>
</tr>
<tr>
<td>hsa00330 Arginine and proline metabolism</td>
<td>0.509</td>
</tr>
<tr>
<td>hsa03320 PPAR signaling pathway</td>
<td>0.508</td>
</tr>
<tr>
<td>hsa04621 NOD-like receptor signaling pathway</td>
<td>0.504</td>
</tr>
<tr>
<td>hsa08660 Glycosphingolipid biosynthesis - ganglio series</td>
<td>0.503</td>
</tr>
<tr>
<td>hsa08621 Glycosphingolipid biosynthesis - globo series</td>
<td>0.486</td>
</tr>
<tr>
<td>hsa04973 Carbohydrate digestion and absorption</td>
<td>0.476</td>
</tr>
<tr>
<td>hsa00030 Pentose phosphate pathway</td>
<td>0.473</td>
</tr>
<tr>
<td>hsa00340 Histidine metabolism</td>
<td>0.448</td>
</tr>
<tr>
<td>hsa03440 Homologous recombination</td>
<td>0.437</td>
</tr>
<tr>
<td>hsa00850 Butanate metabolism</td>
<td>0.425</td>
</tr>
<tr>
<td>hsa03529 Nucleotide excision repair</td>
<td>0.423</td>
</tr>
<tr>
<td>hsa00014 Glycosaminoglycan biosynthesis - heparan sulfate</td>
<td>0.359</td>
</tr>
<tr>
<td>hsa00351 Cell adhesion molecules (CAMs)</td>
<td>0.334</td>
</tr>
<tr>
<td>hsa04977 Vitamin digestion and absorption</td>
<td>0.281</td>
</tr>
</tbody>
</table>
### Table 12.6: Pathways significantly downregulated in pericardial fluid (HIV-1 infected).

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean logFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa04650 Natural killer cell mediated cytotoxicity</td>
<td>-2.279</td>
</tr>
<tr>
<td>hsa04740 Olfactory transduction</td>
<td>-1.930</td>
</tr>
<tr>
<td>hsa04210 Apoptosis</td>
<td>-1.810</td>
</tr>
<tr>
<td>hsa04670 Leukocyte transendothelial migration</td>
<td>-1.778</td>
</tr>
<tr>
<td>hsa00564 Glycophospholipid metabolism</td>
<td>-1.619</td>
</tr>
<tr>
<td>hsa04010 MAPK signaling pathway</td>
<td>-1.582</td>
</tr>
<tr>
<td>hsa04920 Adipocytokine signaling pathway</td>
<td>-1.503</td>
</tr>
<tr>
<td>hsa04810 Regulation of actin cytoskeleton</td>
<td>-1.477</td>
</tr>
<tr>
<td>hsa00910 Nitrogen metabolism</td>
<td>-1.415</td>
</tr>
<tr>
<td>hsa04144 Endocytosis</td>
<td>-1.330</td>
</tr>
<tr>
<td>hsa04140 Regulation of autophagy</td>
<td>-1.328</td>
</tr>
<tr>
<td>hsa04720 Long-term potentiation</td>
<td>-1.299</td>
</tr>
<tr>
<td>hsa04350 TGF-beta signaling pathway</td>
<td>-1.246</td>
</tr>
<tr>
<td>hsa04660 Fc gamma R-mediated phagocytosis</td>
<td>-1.226</td>
</tr>
<tr>
<td>hsa04380 Osteoclast differentiation</td>
<td>-1.228</td>
</tr>
<tr>
<td>hsa04270 Vascular smooth muscle contraction</td>
<td>-1.199</td>
</tr>
<tr>
<td>hsa04070 Phosphatidylinositol signaling system</td>
<td>-1.197</td>
</tr>
<tr>
<td>hsa00760 Nicotinate and nicotinamide metabolism</td>
<td>-1.163</td>
</tr>
<tr>
<td>hsa04360 Axon guidance</td>
<td>-1.076</td>
</tr>
<tr>
<td>hsa04664 Fc epsilon RI signaling pathway</td>
<td>-1.042</td>
</tr>
<tr>
<td>hsa00562 Inositol phosphate metabolism</td>
<td>-1.020</td>
</tr>
<tr>
<td>hsa04310 Wnt signaling pathway</td>
<td>-1.001</td>
</tr>
<tr>
<td>hsa04966 Collecting duct acid secretion</td>
<td>-1.003</td>
</tr>
<tr>
<td>hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series</td>
<td>-0.997</td>
</tr>
<tr>
<td>hsa04730 Long-term depression</td>
<td>-0.965</td>
</tr>
<tr>
<td>hsa04630 Jak-STAT signaling pathway</td>
<td>-0.938</td>
</tr>
<tr>
<td>hsa00770 Pantothenate and CoA biosynthesis</td>
<td>-0.912</td>
</tr>
<tr>
<td>hsa04370 VEGF signaling pathway</td>
<td>-0.894</td>
</tr>
<tr>
<td>hsa04120 Ubiquitin mediated proteolysis</td>
<td>-0.860</td>
</tr>
<tr>
<td>hsa04520 Adherens junction</td>
<td>-0.811</td>
</tr>
<tr>
<td>hsa04012 ErbB signaling pathway</td>
<td>-0.809</td>
</tr>
<tr>
<td>hsa04115 p53 signaling pathway</td>
<td>-0.805</td>
</tr>
<tr>
<td>hsa04662 B cell receptor signaling pathway</td>
<td>-0.768</td>
</tr>
<tr>
<td>hsa04976 Bile secretion</td>
<td>-0.723</td>
</tr>
<tr>
<td>hsa04062 Chemokine signaling pathway</td>
<td>-0.699</td>
</tr>
<tr>
<td>hsa00565 Glycosylphosphatidylinositol(GPI)-anchor biosynthesis</td>
<td>-0.660</td>
</tr>
<tr>
<td>hsa00340 Steroid hormone biosynthesis</td>
<td>-0.643</td>
</tr>
<tr>
<td>hsa04817 Tight junction</td>
<td>-0.630</td>
</tr>
<tr>
<td>hsa04640 Hematopoietic cell lineage</td>
<td>-0.627</td>
</tr>
<tr>
<td>hsa00512 Mucin type O-Glycan biosynthesis</td>
<td>-0.596</td>
</tr>
<tr>
<td>hsa00590 Arachidonic acid metabolism</td>
<td>-0.570</td>
</tr>
<tr>
<td>hsa00430 Taurine and hypotaurine metabolism</td>
<td>-0.557</td>
</tr>
<tr>
<td>hsa02010 ABC transporters</td>
<td>-0.544</td>
</tr>
<tr>
<td>hsa04510 Focal adhesion</td>
<td>-0.537</td>
</tr>
<tr>
<td>hsa04742 Taste transduction</td>
<td>-0.532</td>
</tr>
<tr>
<td>hsa04540 Gap junction</td>
<td>-0.518</td>
</tr>
<tr>
<td>hsa04971 Gastric acid secretion</td>
<td>-0.515</td>
</tr>
<tr>
<td>hsa04910 Insulin signaling pathway</td>
<td>-0.515</td>
</tr>
<tr>
<td>hsa00260 Glycine, serine and threonine metabolism</td>
<td>-0.508</td>
</tr>
<tr>
<td>hsa04614 Renn-angiotensin system</td>
<td>-0.507</td>
</tr>
<tr>
<td>hsa04330 Notch signaling pathway</td>
<td>-0.470</td>
</tr>
<tr>
<td>hsa04660 T cell receptor signaling pathway</td>
<td>-0.461</td>
</tr>
<tr>
<td>hsa04975 Fat digestion and absorption</td>
<td>-0.437</td>
</tr>
<tr>
<td>hsa04912 GmRH signaling pathway</td>
<td>-0.432</td>
</tr>
<tr>
<td>hsa04150 mTOR signaling pathway</td>
<td>-0.426</td>
</tr>
<tr>
<td>hsa04340 Hedgehog signaling pathway</td>
<td>-0.412</td>
</tr>
<tr>
<td>hsa00533 Glycosaminoglycan biosynthesis - keratan sulfate</td>
<td>-0.415</td>
</tr>
<tr>
<td>hsa03022 Basal transcription factors</td>
<td>-0.408</td>
</tr>
<tr>
<td>hsa04970 Salivary secretion</td>
<td>-0.399</td>
</tr>
<tr>
<td>hsa00860 Porphyrin and chlorophyll metabolism</td>
<td>-0.390</td>
</tr>
<tr>
<td>hsa04964 Proximal tubule bicarbonate reclamation</td>
<td>-0.377</td>
</tr>
<tr>
<td>hsa04710 Circadian rhythm - mammal</td>
<td>-0.342</td>
</tr>
<tr>
<td>hsa00790 Folate biosynthesis</td>
<td>-0.346</td>
</tr>
<tr>
<td>hsa04620 Toll-like receptor signaling pathway</td>
<td>-0.331</td>
</tr>
<tr>
<td>hsa04520 Dorso-ventral axis formation</td>
<td>-0.300</td>
</tr>
<tr>
<td>hsa00591 Lipoic acid metabolism</td>
<td>-0.292</td>
</tr>
<tr>
<td>hsa04622 RIG-I-like receptor signaling pathway</td>
<td>-0.276</td>
</tr>
</tbody>
</table>
The most striking finding in pathways differentially regulated at the site of disease is the emphasis on multiple metabolic pathways. There is concerted upregulation of pathways important in central carbon metabolism (e.g. glyoxylate and dicarboxylate metabolism, citrate cycle, propanoate metabolism), pathways critical to growth in *M. tuberculosis*. Previous *in vitro* findings suggest alignment of MTB metabolism with its current environment [164] as well as large-scale manipulation of host metabolism by trehalose dimycolate, a mycobacterial cell wall lipid [165]. The latter results in dysregulated host lipid metabolism, a finding associated with caseation of granulomas.

In addition to large-scale metabolic rearrangements, there is upregulation of transcriptional and translational machinery, indicating site-specific protein synthesis, possibly in direct response to the infection. Upregulated DNA duplication pathways suggest cell proliferation at the site of disease. Glycan biosynthesis is also upregulated. As glycans form an important part of mycobacterial cell walls, one can speculate whether this host pathway is actively utilised by MTB to meet its requirements.

Upregulation of the proteasome pathway is probably in part related to upregulated MHC class I activity, as the proteasome plays a critical role in the function of the adaptive immune system by producing peptides to display on MHC class I molecules [166].

Downregulated pathways are also of significant interest. Natural killer cell cytotoxicity is strongly downregulated, perhaps indicating a defensive strategy by the mycobacterium. Other pathways which intuitively should be upregulated in a protective response are leukocyte transendothelial migration, Fc- gamma receptor mediated phagocytosis, regulation of autophagy (both ATG-1 and ATG-8 are downregulated), again hinting at the possibility that the immune response at the site of disease is inappropriately downregulated.

Finally, the overlap of upregulated and downregulated pathways in HIV-1 uninfected and infected individuals is near complete with additional pathways dysregulated in HIV-1 infected individuals (Figure 12.6). Surprisingly, there are no pathways that are regulated in opposite directions depending on HIV status.
Figure 12.6: Overlap of differentially regulated pathways stratified by HIV status
12.2 Pathway analysis

12.2.1 Pathway visualisation

Figure 12.7 shows the KEGG pathway *Antigen processing and presentation*. This pathway is universally upregulated in pericardial fluid in individuals with tuberculous pericarditis, but its effects may be different depending on HIV-1 infection status. Regardless of HIV status the pathway *Natural killer cell-mediated cytotoxicity* is strongly downregulated, raising questions about mechanisms involved in this counterintuitive phenomenon. One question that is suggested by the data is whether downregulation of NK cell cytotoxicity is required for the pathogenesis of tuberculosis.
Figure 12.7: Antigen processing and presentation pathway. Canonical pathway hsa04612 from KEGG, rendered by pathview software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: red = upregulated in pericardial fluid; green = downregulated in pericardial fluid.
12.2 Pathway analysis

Figure 12.8: **Natural killer cell mediated cytotoxicity.** Canonical pathway hsa04650 from KEGG, rendered by pathview software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in pericardial fluid; *green* = downregulated in pericardial fluid.

(a) HIV-1 uninfected

(b) HIV-1 infected
Part V

Overall conclusions
13 Conclusions and future work

13.1 Summary of novel results

Below I summarise the novel aspects of this work:

1. Creation of a unique and comprehensive microarray dataset (with detailed phenotype information) enabling investigation of multiple aspects of HIV-TB in blood and the site of disease. To my knowledge similar datasets do not exist.

2. Development of a comprehensive analysis pipeline using only open-source software, enabling research reproducibility once the dataset and analysis pipeline are jointly published. In addition to standard differential expression analysis, this pipeline also extracts information regarding cell proportions, modular organisation of co-expressed genes and pathway enrichment, resulting in a comprehensive overview of biological phenomena in the system under study.

3. Application of the analytic pipeline to multiple data subsets which are placed in clear relation to each other using the hypercube analogy that describes sample embedding in phenotype space. This maximises the information that can be extracted from the complex signals embedded in the data.

4. New insights into the biology of HIV-TB were attained using the above methods, in particular the strong suggestion that NK cells may play an important role in tuberculosis and the transcriptional response at the site of disease resulting in major metabolic rearrangements that may support mycobacterial growth.
13 Conclusions and future work

13.2 An approach to heterogeneous microarray data

Biology is complicated. Acknowledging this complexity in experimental designs opens the door to potentially deep insights, at the price of significantly increased analytic difficulty. Given the potential for interactions between biological processes that define “phenotype classes” like active tuberculosis or HIV-1 infection, it is important to investigate these processes individually and jointly.

With linear increases in phenotypic complexity of study samples there is a concurrent exponential increase in signal complexity, as shown using the hypercube analogy. To make analysis feasible, the data needs to be understood in relation to the phenotype classes in potentially high-dimensional parameter spaces.

It is tempting to draw conclusions regarding disease pathogenesis from lists of over- and under-expressed transcripts. Care must be taken not to over-interpret results, however. Gene expression is a very complex process, and attempts at measuring this in heterogeneous tissues (blood, pericardial fluid) sampled from a heterogeneous study population will result in data in which multiple signals are embedded in a background influenced by stochastic processes. Therefore, while differential gene expression undoubtedly provides us with the ability to classify samples it also provides us with information directly related to the biological process. Retrieving this “biology signal” from the background remains challenging, but I hope that methods that aim to isolate multiple signals from heterogeneous datasets will allow us understand the biology better.

For the study under discussion, transcriptomes of whole blood and pericardial fluid were established. Each tissue type has a characteristic cellular composition which strongly influences the overall transcript abundance. Differences in transcript abundance between conditions of interest are typically found in studies of this nature. Indeed, differences in transcript abundance may be identified even when comparing groups created randomly, and the challenge becomes identifying biologically relevant differences. This strongly argues for use of a combination of analytic methods that aim to understand the meaning of the tissue (blood and pericardial fluid) transcriptomes.

Transcriptomes vary over time. In this study, a single timepoint was used to generate transcriptomic data. In terms of the two disease groups (pulmonary and pericardial tuberculosis), the timepoint was determined by the study subject presenting to medical care, and undergoing dia-
gnostic workup for tuberculosis. This has the consequence that the disease group may contain significant heterogeneity due to individuals presenting in different stages of the disease (e.g. pulmonary TB with a single disease focus vs. pulmonary tuberculosis with concomitant disseminated disease), potentially confounding the results. Therefore, a “snapshot” transcriptome may yield differences between tuberculosis cases that are simply due to different disease stages, to some extent limiting the generalisability of the results.

13.3 Choice of methods

I chose three methods of analysis: differential expression analysis using linear models, deconvolution and cell-type specific differential expression, and weighted gene co-expression network analysis. Results were put into biological context using pathway analysis. The three methods complement each other, as different features of the biological problem are explored. Together, these methods provide results that may serve as input data for further analyses.

13.4 Results

Expectedly, the largest differences in gene expression can be found when comparing the site of disease to blood. This is mainly due to different proportions of cell types in the two compartments, but cell-type specific differential expression strongly suggests that within multiple cell types biologically relevant differences can be identified. Furthermore, the site of disease exhibits markedly different metabolic activity when compared to blood. It remains to be determined which of these metabolic alterations is caused by *Mycobacterium tuberculosis* to facilitate its survival by creating an environment optimised for its own growth, which are of a defensive nature, and which are unrelated to the presence of the bacterium. Finally, several pathways thought to play a protective role in the response to *M. tuberculosis* are downregulated at the site of disease, possibly indicating additional defence mechanisms of the pathogen.

The effect of HIV-1 infection on transcriptional profiles is a lot more subtle than the effect of compartment. For instance, few or no probes were significantly differentially expressed in individuals with or without concurrent active tuberculosis. This is puzzling, and may be a result of
13 Conclusions and future work

small numbers of HIV-1 uninfected samples in the various analyses investigating this contrast. Despite this, the top 300 and top 500 differentially regulated probes do successfully cluster the input data, and cell-type specific differential gene expression is detectable. Pathway analysis recapitulates some key results (e.g. differential regulation in the RIG-1-like pathway), and also show that the differential regulation of pathways in HIV-1 infection is strongly influenced by concurrent active tuberculosis.

The transcriptional response to active tuberculosis is to a large extent conserved regardless of HIV-1 infection status. This response consists of upregulation of pathways central to energy metabolism, phagocytosis and downstream events and innate immune responses (like Toll-like receptor signalling). In HIV-1 co-infection, a number of additional pathways are up- or downregulated, indicating that active tuberculosis and HIV-1 interact on many levels.

In each of the analyses, modules were detected using weighted gene co-expression network analysis. These modules have not been characterised on a functional level (see Future Work, below), but is is interesting to note that module size varies widely. What do these modules represent? While the answer to this is not yet known, I speculate that study of the modules may reveal pathway groups, expressed in concert, given that some modules contain thousands of genes. In some cases, the modules may represent an even higher level of organisation, namely cell types.

The strategy employed in this analysis has afforded us a view of different aspects contributing to HIV-TB pathobiology. In all three main contrasts, a common dual theme emerges: disease (be it HIV-1 infection or tuberculosis) is associated with striking alterations in immune system function at many levels as well as significant changes in host metabolism. While this has been known for a long time, the results presented here in this thesis give a detailed look at the idea that host metabolism and immune response are part of a higher-order phenomenon: that of pathogens utilising humans as an ecologic niche by evading immune responses and modifying host metabolism to serve their respective needs. This opens up new avenues of investigation, in which immune responses and host metabolic response are assessed as one functional unit. The pathogens’ reliance on host metabolic pathways may well turn out to be their Achilles heel, and therefore I argue that host metabolic responses should be studied with the potential of developing new therapeutic strategies.

As a final word on this, as a clinician I would like to state that these results should not come as a big surprise. On a whole system level (i.e. patient physiology) we are deeply familiar with
the metabolic consequences of HIV-1 infection (the entity “wasting disease” is AIDS defining, after all) as well as tuberculosis (or “consumption”, a term no longer in widespread use). Maybe the focus of future investigations should shift to include mechanisms of wholesale manipulation of energy metabolism by both pathogens, in addition to defining mechanisms leading to immune system evasion and immunopathology. This addresses the following speculative scenario: what if the contribution that HIV-1 infection makes to the increased risk for developing active TB is not immune dysfunction alone (although this doubtlessly plays a crucial role) but also re-engineering of host metabolism to better suit *M. tuberculosis* growth and survival.

### 13.5 Future work

#### 13.5.1 Informatics

This thesis presents only a few results for a subset of all contrasts examined. The remaining results, constituting a dataset in their own right require ongoing analysis and interpretation. Specifically, the modules identified may point to ways how various pathways are connected and co-expressed. I have shown that some of the results, obtained in a deep sweep of transcriptional activity in active TB and HIV-1 infection replicate previously published findings. I intend to match all results of this analysis with the extant TB literature. The potential outcome for each result is threefold:

1. Agreement with published results serving as validation of prior knowledge

2. Disagreement with published results serving as potential alternative hypothesis which may require experimental assessment

3. Novel finding, indicating the need for validation experiments

I plan to prepare a catalogue of targeted experiments based on the above analysis which may aid our understanding of HIV-1 infection, tuberculosis and co-infection.

In addition to the present dataset, the opportunity to re-analyse existing datasets using the pipeline presented here will be utilised, as several groups have published data that is freely accessible.

Ongoing efforts will be made to optimise the analytic pipeline. Pathway analysis, currently implemented in separate code, will be integrated into the pipeline, and the entire pipeline will be upgraded to function under the latest release of R and bioconductor.
13 Conclusions and future work

13.5.2 Validation

All results of this analysis are at the RNA level. Validation experiments are required to assess these results in the context of protein-level and cellular level findings. To start this process, targeted experiments will be designed to utilise stored serum, cell-free pericardial fluid and frozen, unstimulated PBMC and pericardial fluid cells. Validation will include, but not be limited to assessment of cell proportions by flow cytometry, and comparing these to the results obtained by deconvolution, as well as determination of key proteins (cytokines, chemokines, enzymes) in the various contexts. An additional class of validation experiments may be performed at the metabolite level, in order to define the metabolic niche utilised by *M. tuberculosis*.

13.5.3 New projects

Not all potential contrasts could be examined; Figure 8.1 illustrates this graphically. This framework can be used to design a new study comparing blood and site of disease samples even in LTBI, PTB and possibly other forms of tuberculosis, in contexts with or without HIV-1 co-infection. This may be achieved by careful study design and stratified sampling to ensure that all vertices in high-dimensional phenotype space are as evenly populated as possible.
Bibliography


61. Ramaseri Sunder, S. et al. IL-10 high producing genotype predisposes HIV infected individuals to TB infection. *Hum Immunol* 73, 605–611 (June 2012).


Bibliography


Bibliography

Colophon

This document was typeset in 11pt Times Roman using LyX. Typesetting and layout utilised \LaTeX\ 2e. Bibliographic data was stored in Bib\TeX\ format, and processed using \texttt{biber} and \texttt{biblatex}.

The pdf version has live hyperlinks for references and table of contents which may aid in reading.
Part VI

Appendix
A Code

A.1 Code for functions used throughout

A.1.1 Modified heatmap (heatmap_ad.R)

Changes made to original “heatmap.plus” command from heatmap.plus package:

Added options noan (number of annotations) and rowlab (row labels) to increase the width (lhei) of column annotation rows for the Colv matrix by the factor noan. This is useful because a large number of column annotations (e.g. based on phenodata) normally make the Colv matrix unreadable as it has a fixed width. The rowlab option switches the labelling of the column annotations on and off as desired. Credit for noan parametrisation of lhei is this post:


```r
heatmap_ad <- function (x, Rowv = NULL, Colv = if (symm) "Rowv" else NULL,
    distfun = dist, hclustfun = hclust, reorderfun = function (d,
        w) reorder (d, w), add.expr = symm = FALSE, revC = identical (Colv,
        "Rowv"), scale = c("row", "column", "none"), na.rm = TRUE,
    margins = c(5, 5), ColSideColors, RowSideColors, cexRow = 0.2 +
    1/log10(nr), cexCol = 0.2 + 1/log10(nc), labRow = NULL,
    labCol = NULL, main = NULL, xlab = NULL, ylab = NULL, keep.dendro =
    FALSE, noan = NULL, rowlab = TRUE,
    verbose = getOption("verbose"), ...) {
    scale <- if (symm & missing (scale))
        "none"
    else match.arg (scale)
    if (length (di <- dim (x)) != 2 || !is.numeric (x))
        stop ("'x' must be a numeric matrix")
    nr <- di [1]
    nc <- di [2]
    if (nr <= 1 || nc <= 1)
        stop ("'x' must have at least 2 rows and 2 columns")
    if (!is.numeric (margins) || length (margins) != 2)
        stop ("'margins' must be a numeric vector of length 2")
    doRdend <- ! identical (Rowv, NA)
    doCdend <- ! identical (Colv, NA)
    if (is.null (Rowv))
        Rowv <- rowMeans (x, na.rm = na.rm)
    if (is.null (Colv))
        Colv <- colMeans (x, na.rm = na.rm)
    if (doRdend) {
        if (inherits (Rowv, "dendrogram"))
            Rowv <- reorder (Rowv, Rowv)
    ```

297
ddr <- Rowv
else {
  hcr <- hclustfun(distfun(x))
  ddr <- as.dendrogram(hcr)
  if (!is.logical(Rowv) || Rowv)
    ddr <- reorderfun(ddr, Rowv)
}
if (nr != length(rowInd <- order.dendrogram(ddr)))
  stop("row dendrogram ordering gave index of wrong length")
} else rowInd <- 1:nr
if (doCdend) {
  if (inherits(Colv, "dendrogram"))
    ddc <- Colv
  else if (identical(Colv, "Rowv") {  
    if (nr != nc)
      stop("Colv = \"Rowv\" but nrow(x) != ncol(x)")
    ddc <- ddr
  } else {
    hcc <- hclustfun(distfun(if (symm)
      x
    else t(x)))
    ddc <- as.dendrogram(hcc)
    if (!is.logical(Colv) || Colv)
      ddc <- reorderfun(ddc, Colv)
  }
  if (nc != length(colInd <- order.dendrogram(ddc)))
    stop("column dendrogram ordering gave index of wrong length")
} else colInd <- 1:nc
x <- x[rowInd, colInd]
labRow <- if (rowlab == TRUE) {
  if (is.null(labRow))
    if (is.null(rownames(x)))
      (1:nr)[rowInd]
  else rownames(x)
} else labRow[rowInd]
else NULL
labCol <- if (is.null(labCol)) {
  if (is.null(colnames(x)))
    (1:nc)[colInd]
  else colnames(x)
} else labCol[colInd]
if (scale == "row") {
  x <- sweep(x, 1, rowMeans(x, na.rm = na.rm))
  sx <- apply(x, 1, sd, na.rm = na.rm)
  x <- sweep(x, 1, sx, "/")
} else if (scale == "column") {
  x <- sweep(x, 2, colMeans(x, na.rm = na.rm))
  sx <- apply(x, 2, sd, na.rm = na.rm)
  x <- sweep(x, 2, sx, "/")
}
lmat <- rbind(c(NA, 3), 2:1)
lwid <- c(if (doRdend) 1 else 0.05, 4)
lhei <- c(if (doCdend) 1 else 0.05) + if (!is.null(main)) 0.2 else 0,4)
A.1 Code for functions used throughout

# lhei <- c((if (doCdend) 1 else 0.05) + if (!is.null(main)) 0.8 else 0, 4)
if (!missing(ColSideColors)) {
  if (!is.matrix(ColSideColors))
    stop("'ColSideColors' must be a matrix")
  if (!is.character(ColSideColors) || dim(ColSideColors)[1] != nc)
    stop("'ColSideColors' dim()[2] must be of length ncol(x)")
  lmat <- rbind(lmat[1, ] + 1, c(NA, 1), lmat[2, ] + 1)
  # lmat <- rbind(lmat[1, ] + 1, c(NA, 1), lmat[3, ] + 1)
  # lhei <- c(lhei[1], 0.2, lhei[2])
  lhei <- c(lhei[1], 0.1*noan, lhei[2])
}
if (!missing(RowSideColors)) {
  if (!is.matrix(RowSideColors))
    stop("'RowSideColors' must be a matrix")
  if (!is.character(RowSideColors) || dim(RowSideColors)[1] != nr)
    stop("'RowSideColors' must be a character vector of length nrow(x)")
  lmat <- cbind(lmat[, 1] + 1, c(rep(NA, nrow(lmat) - 1), 1), lmat[, 2] + 1)
  # lwid <- c(lwid[1], 0.2, lwid[2])
  lwid <- c(lwid[1], 0.2, lwid[2])
}
lmat[is.na(lmat)] <- 0
if (verbose) {
  cat("layout: widths = ", lwid, ", heights = ", lhei, "; lmat=
  print(lmat)
}
op <- par(no.readonly = TRUE)
on.exit(par(op))
layout(lmat, widths = lwid, heights = lhei, respect = FALSE)

#1 draw row side col
if (!missing(RowSideColors)) {
  par(mar = c(margins[1], 0, 0, 0.5))
  rsc = RowSideColors[rowInd, ]
  rsc.colors = matrix()
  rsc.names = names(table(rsc))
  rsc.i = 1
  for (rsc.name in rsc.names) {
    rsc.colors[rsc.i] = rsc.name
    rsc[rsc == rsc.name] = rsc.i
    rsc.i = rsc.i + 1
  }
  rsc = matrix(as.numeric(rsc), nrow = dim(rsc)[1])
  image(t(rsc), col = as.vector(rsc.colors), axes = FALSE)
  if (length(colnames(RowSideColors)) > 0) {
    axes(1, 0:(dim(rsc)[2] - 1)/(dim(rsc)[2] - 1), colnames(RowSideColors),
      las = 2, tick = FALSE)
  }
}

#2 draw col side col
if (!missing(ColSideColors)) {
  par(mar = c(0.5, 0, 0, margins[2]))
  csc = ColSideColors[colInd, ]
csc.colors = matrix()
csc.names = names(table(csc))
csc.i = 1
for (csc.name in csc.names) {
  csc.colors[csc.i] = csc.name
  csc[csc == csc.name] = csc.i
  csc.i = csc.i + 1
}
csc = matrix(as.numeric(csc), nrow = dim(csc)[1])
image(csc, col = as.vector(csc.colors), axes = FALSE)
if (length(colnames(ColSideColors)) > 0) {
  axis(2, 0:(dim(csc)[2] - 1)/(dim(csc)[2] - 1), colnames(ColSideColors),
       las = 2, tick = FALSE)
}

#3 draw heatmap
par(mar = c(margins[1], 0, 0, margins[2]))
if (!symm || scale != "none") {
x <- t(x)
}
if (revC) {
  iy <- nr:1
  ddr <- rev(ddr)
  x <- x[, iy]
} else iy <- 1:nr
image(1:nc, 1:nr, x, xlim = 0.5 + c(0, nc), ylim = 0.5 +
c(0, nr), axes = FALSE, xlab = "", ylab = "", ...
axis(1, 1:nc, labels = labCol, las = 2, line = -0.5, tick = 0,
     cex.axis = cexCol)
if (!is.null(xlab))
  mtext(xlab, side = 1, line = margins[1] - 1.25)
axis(4, iy, labels = labRow, las = 2, line = -0.5, tick = 0,
     cex.axis = cexRow)
if (!is.null(ylab))
if (!is.null(add.expr))
  eval(substitute(add.expr))
#4 draw row dendro
par(mar = c(margins[1], 0, 0, 0))
if (doRdend)
  plot(ddr, horiz = TRUE, axes = FALSE, yaxs = "i", leaflab = "none")
else frame()
#5 draw col dendro
par(mar = c(0, 0, if (!is.null(main)) 2 else 0, margins[2]))
if (doCdend)
  plot(ddc, axes = FALSE, xaxs = "i", leaflab = "none")
else if (!is.null(main))
  frame()
#6 title
if (!is.null(main))
  title(main, cex.main = 1.3 * op["cex.main"])
invisible(list(rowInd = rowInd, colInd = colInd, Rowv = if (keep.dendro &
                      doRdend) ddr, Colv = if (keep.dendro && doCdend) ddc))
Listing A.1: heatmapad.R
A.1.2 Comprehensive heatmap (1) (superHeatmap.R)

Convenient wrapper function for drawing heatmaps in the style used in the Berry et al paper [82]. This function adds options for correlation clustering and coloured dendrograms to the function parameters, and then draws the heatmap using heatmap_ad.R.

Correlation code is based on this tutorial on the ULB BIGRe lab site, and the coloured dendrogram solution is based on this post.

```r
superHeatmap <- function (x,y=selected.probes, phenomatrix=phenomatrix, scale=scale, addTit=NULL) {
  #cluster
  genes.cor <- cor(t(exprs(x)[y,]), use="pairwise.complete.obs", method="pearson")
  genes.cor.dist <- as.dist(1-genes.cor)
  genes.tree <- hclust(genes.cor.dist, method='average')
  samples.cor.spearman <- cor(exprs(x)[y,], use="pairwise.complete.obs", method="spearman")
  samples.cor.spearman.dist <- as.dist(1-samples.cor.spearman)
  samples.tree <- hclust(samples.cor.spearman.dist, method='average')
  #dendrogram colors
  # coloured dendrogram
  smpCol <- as.dendrogram(samples.tree)
  local({
    colLab <- function(n) {
      a <- attributes(n)
      i <<- i+1
      # attr(n, "nodePar") <-
      # c(a$nodePar, list(lab.col = mycols[i], lab.font = i%%3),
      pch=NULL)
      attr(n, "edgePar") <-
      c(a$edgePar, list(col = mycols[i]), lwd=3)
    }
    n
  })
  mycols <- as.vector(phenomatrix[,1][order.dendrogram(smpCol)])
  i <- 0
})
  dL <- dendrapply(smpCol, colLab)
  max.level <- max(abs(range(exprs(x)[y,])))
  br <- c(seq(from=0, to=1, by=1/127), seq(from=1, to=5, by=4/126), max.level)
  #heatmap
  heatmap.ad(as.matrix(exprs(x)[y,]),
             scale=scale, ## AVOID RESCALING THE VALUES OF ALL COLUMNS
             col=palette.BYR(),
             breaks=br,
             Rowv=as.dendrogram(genes.tree),
             Colv=dL,
             main=paste(nrow(exprs(x)[y,]), "probes", addTit),
             ColSideColors=phenomatrix,
```

302
A.1 Code for functions used throughout

```r
margins=c(9,5),
#RowSideColors=rsc,
noan=2,labRow="",verbose=TRUE
```

Listing A.2: superHeatmap.R
A.1.3 Comprehensive heatmap (2) (superHeatmap.R)

Improves on superHeatmap.R by automating the selection of colour levels in the blue-yellow-red colour ramp.

```r
superHeatmap2 <- function(x,y=selected.probes, phenomatrix=phenomatrix, scale=scale, addTit=NULL, low=low, high=high) {
  #cluster
  genes.cor <- cor(t(exprs(x)[y,]), use="pairwise.complete.obs", method="pearson")
  genes.cor.dist <- as.dist(1-genes.cor)
  genes.tree <- hclust(genes.cor.dist, method='average')
  samples.cor.spearman <- cor(exprs(x)[y,], use="pairwise.complete.obs", method="spearman")
  samples.cor.spearman.dist <- as.dist(1-samples.cor.spearman)
  samples.tree <- hclust(samples.cor.spearman.dist, method='average')
  #dendrogram colors
  smpCol <- as.dendrogram(samples.tree)
  local({
    colLab <- function(n) {
      if(is.leaf(n)) {
        a <- attributes(n)
        i <<- i+1
        # attr(n, "nodePar") <-
        # c(a$nodePar, list(lab.col = mycols[i], lab.font = i%%3),
        pch=NULL)
        attr(n, "edgePar") <-
        c(a$edgePar, list(col = mycols[i]), lwd=3)
      }
    }
    mycols <- as.vector(phenomatrix[,1])[order.dendrogram(smpCol)]
    i <<- 0
  })
  dl <- dendrapply(smpCol, colLab)
  max.level <- max(abs(range(exprs(x)[y,])))
  min.level <- min(abs(range(exprs(x)[y,])))
  low<-1.05*min.level
  high<-0.6*max.level
  br <- c(seq(from=min.level, to=low, length.out=30),
    seq(from=low, to=high, length.out=105),
    seq(from=high, to=max.level, length.out=120),
    max.level
  )
  #heatmap
  heatmap_ad(as.matrix(exprs(x)[y,]),
    scale=scale, ## AVOID RESCALING THE VALUES OF ALL COLUMNS
    col=palette.BYR(),
    breaks=br,
    Rowv=as.dendrogram(genes.tree),
    Colv=dl,
    main=paste(nrow(exprs(x)[y,]),"probes",addTit),
    ColSideColors=phenomatrix,
}
```

Listing A.3: superHeatmap2.R

```r
margins=c(9,5),
#RowSideColors=rsc,
noan=2,labRow="",verbose=TRUE
```
A.1.4 Modified Sample Relations plot (MDS) (plotSampleRelationsAD.R)

Modifies the function `plotSampleRelation` from the `lumi` package to plot points using specified plot character `plotchar` instead of the sample names; this makes the plot more readable. A background colour can also be specified.

```r
plotSampleRelationsAD <- function (x, subset = NULL, cv.Th = 0.1, standardize = TRUE,
            method = c("cluster", "mds"), dimension = c(1, 2), color = NULL,
            backgr = NULL, plotchar=24,
            main = NULL, ...) {
  if (is(x, "ExpressionSet")) {
    dataMatrix <- exprs(x)
  } else if (is.matrix(x)) {
    dataMatrix <- x
  } else {
    stop("The class of \"x\" should be matrix or LumiBatch!")
  }
  if (standardize)
    dataMatrix <- scale(dataMatrix)
  if (is.null(subset)) {
    probeList <- rownames(dataMatrix)
    if (is.null(probeList))
      probeList <- 1:nrow(dataMatrix)
    if (cv.Th > 0) {
      cv.gene <- apply(dataMatrix, 1, function(x) sd(x)/mean(x))
      subset <- probeList[abs(cv.gene) > cv.Th]
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
          "genes with sd/mean >", cv.Th)
    } else {
      subset <- probeList
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
          "genes")
    }
  } else {
    if (length(subset) == 1 && is.numeric(subset)) {
      subset <- sample(1:nrow(dataMatrix), min(subset,
          nrow(dataMatrix)))
    } if (is.null(main))
      main <- paste("Sample relations based on", length(subset),
        "selected genes")
  }
  dd <- dist(t(dataMatrix[, subset]))
  method <- match.arg(method)
  if (method == "cluster") {
    hc = hclust(dd, "ave")
    plot(hc, xlab = "Sample", main = main, ...)
    attr(hc, "geneNum") <- length(subset)
    attr(hc, "threshold") <- cv.Th
    return(invisible(hc))
  }
```

306
else {
    mds.result <- cmdscale(dd, k = max(dimension), eig = TRUE)
    ppots <- mds.result$points
    eig <- mds.result$eig
    percent <- round(eig/sum(eig) * 100, 1)
    if (is.null(color)) {
        color <- 1
    }
    if (!is.numeric(color)) {
        allColor <- colors()
        if (!all(is.element(color, allColor))) {
            color <- as.numeric(factor(color, levels = unique(color)))
        }
    }
    plot(ppots[, dimension[1]], ppots[, dimension[2]],
         type = "n", xlab = paste("Principal Component ",
         dimension[1], " (", percent[dimension[1]], 
         ")", sep = ""), ylab = paste("Principal Component ",
         dimension[2], " (", percent[dimension[2]], 
         ")", sep = ""), main =
         points(ppots[, dimension[1]], ppots[, dimension[2]],
         col = color, bg=backgr, cex = 1,pch=plotchar)
    attr(ppots, "geneNum") <- length(subset)
    attr(ppots, "threshold") <- cv.Th
    return(invisible(ppots))
}

Listing A.4: plotSampleRelationsAD.R
A.1.5 Modified 3D Sample Relations plot

Modifies the function `plotSampleRelationsAD.R` to plot points in 3D space using the first three principal components. The `plot3d` command in this function requires the rgl library, which is loaded in the setup script.

```r
plotSampleRelations3D_AD <- function(x, subset = NULL, cv.Th = 0.1,
  standardize = TRUE,
  method = c("cluster", "mds"), dimension = c
  (1, 2,3), color = NULL, backgr = NULL, plotchar=24, spheresize=2,
  main = NULL, ...)
{
  if (is(x, "ExpressionSet")) {
    dataMatrix <- exprs(x)
  } else if (is.matrix(x)) {
    dataMatrix <- x
  } else {
    stop("The class of \"x\" should be matrix or LumiBatch!")
  }
  if (standardize)
    dataMatrix <- scale(dataMatrix)
  if (is.null(subset)) {
    probeList <- rownames(dataMatrix)
    if (is.null(probeList))
      probeList <- 1:nrow(dataMatrix)
    if (cv.Th > 0) {
      cv.gene <- apply(dataMatrix, 1, function(x) sd(x)/mean(x))
      subset <- probeList[abs(cv.gene) > cv.Th]
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
                      "genes with sd/mean >", cv.Th)
    } else {
      subset <- probeList
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
                      "genes")
    }
  } else {
    if (length(subset) == 1 & is.numeric(subset)) {
      subset <- sample(1:nrow(dataMatrix), min(subset,
                      nrow(dataMatrix)))
    } if (is.null(main))
      main <- paste("Sample relations based on", length(subset),
                      "selected genes")
  }
  dd <- dist(t(dataMatrix[subset, ]))
  method <- match.arg(method)
  if (method == "cluster") {
    hc = hclust(dd, "ave")
    plot(hc, xlab = "Sample", main = main, ...)
    attr(hc, "geneNum") <- length(subset)
    attr(hc, "threshold") <- cv.Th
    return(invisible(hc))
  }
}
```
else {
    mds_result <- cmdscale(dd, k = max(dimension), eig = TRUE)
    ppoints <- mds_result$points
    eig <- mds_result$eig
    percent <- round(eig/sum(eig) * 100, 1)
    if (is.null(color)) {
        color <- 1
    } else {
        if (!is.numeric(color)) {
            allColor <- colors()
            if (!all(is.element(color, allColor))) {
                color <- as.numeric(factor(color, levels = unique(color)))
            }
        }
    }
    plot3d(ppoints[, dimension[1]], ppoints[, dimension[2]], ppoints[, dimension[3]],
           type = "s", size=spheresize,
           xlab = paste("Principal Component ",dimension[1], " (", percent[dimension[1]], ", ")",sep = "",
           ylab = paste("Principal Component ",dimension[2], " (", percent[dimension[2]], ", ")",sep = "",
           zlab = paste("Principal Component ",dimension[3], " (", percent[dimension[3]], ", ")",sep = "",
           main = main, col=color)
    }
}

Listing A.5: plotSampleRelations3DAD.R
# IMPI_MA microarray analysis pipeline
# Universal System Setup
# Armin Deffur
# File created 18-02-2013
# Version 1.0

## System Setup

init session
```
rm(list=ls())
graphics.off()
```

# Local folder configuration

- **Projects root (scripts and output)**
  ```
dir.root <- '/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects'
  ```

- **R scripts root**
  ```
dir.R.files <- file.path(dir.root, "01_scripts/06_IMPI-MA/Analysis/03_Functions")
print(paste("R scripts source ", dir.R.files))
```

- **R utilities folder**
  ```
dir.util <- file.path(dir.R.files, 'util')
```

# Load utilities
```
source(file.path(dir.util, 'util.R'))
source(file.path(dir.util, 'util_chip_analysis.R'))
source(file.path(dir.R.files, "plotSampleRelationsAD.R"))
source(file.path(dir.R.files, "plotSampleRelations3D_AD.R"))
source(file.path(dir.R.files, "heatmap_ad.R"))
source(file.path(dir.R.files, "heatmap_ad2.R"))
```

# Global parameters
```
verbosity <- 1
export.formats.plots <- c("eps","pdf","png","jpg")
export.formats.plots <- c("pdf","png","jpg")
export.formats.obj <- c("table")
height=16##use for printing?
par(mfrow=c(1,1),mar=c(5, 4, 4, 2) + 0.1,xaxt="s")
parbackup <- par(mfrow=c(1,1),mar=c(5, 4, 4, 2) + 0.1,xaxt="s",cex.main=1)
plotnumber <- 1

## Specify if drawings should be done in colors or not.
in.colors <- T
# A.2 System setup

```r
# load all libraries
library(lumi)
library(annotate)
library(lumiHumanAll.db)
library(lumiHumanIDMapping)
library(rgl)
library(limma)
library(xtable)
library(gplots)
library(VennDiagram)
library(tools)
library(vegan)
library(CellMix)
library(csmSAM)
library(RColorBrewer)
library(pwr)
library(ClassDiscovery)
library(WGCNA)
library(RColorBrewer)
library(beadarray)
library(illuminaHumanv3.db)
library(illuminaHumanv4.db)
```

Listing A.6: SystemSetup.R
A.3 Data manager (general)

```r
# System_setup.R####
source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/
IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System
_setup.R")

# Local Setup####

## Individual data folders (specific for each part)
dir.gx <- file.path(dir.data_root,'09_IMPI-MA_DATA/04_GX_format')
print(paste("GX12-format data repository", dir.gx))
dir.gs <- file.path(dir.data_root,'09_IMPI-MA_DATA-unversioned/05_
GenomeStudio_reports')
print(paste("GenomeStudio report data repository", dir.gs))
dir.bgx <- file.path(dir.data_root,'09_IMPI-MA_DATA-unversioned/06_Illumina
Manifest files/HT12v4")
print(paste("Illumina manifest file repository", dir.bgx))
dir.jpg <- file.path(dir.data_root,'09_IMPI-MA_DATA-unversioned/07_
imageData/JPG")
print(paste("Raw image data (.jpg) repository", dir.jpg))
dir.tif <- file.path(dir.data_root,'09_IMPI-MA_DATA-unversioned/07_
imageData/TIFF")
print(paste("Raw image data (.tif) repository", dir.tif))
dir.pheno <- file.path(dir.data_root,'09_IMPI-MA_DATA/08_PhenoData")
print(paste("Phenodata data repository", dir.pheno))

# Output folders

# Current date string used for versioning output
ds <- Sys.time()

dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version,"results")
print(paste("Results will be saved to", dir.results))

## Define folder for saving figures
dir.figures <- file.path(dir.output.version,"figures")
print(paste("Figures will be saved to", dir.figures))

## Define folder for saving RData files
dir.rdata <- file.path(dir.data_root,'09_IMPI-MA_DATA/05_RData")
print(paste("Data will be saved to", dir.rdata))

for (dir in c(dir.main, dir.figures, dir.results)) {
```
if (! file.exists(dir)) {
  dir.create(dir, recursive=T, showWarnings=T)
}

figure<-1

#WD####
setwd(dir.main)

#Data for Differential Expression

#Data for Deconvolution

# Data for WGCNA

#Import expression and phenotype data using lumi####

fileName<-file.path(dir.gx,"Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm__Sample__Probe_Profile.txt")
fileNameAnno<-file.path(dir.gs,"Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm_FinalReport.txt")
bgxfile<-file.path(dir.bgx,"HumanHT-12_V4_0_R1_15002873_B.bgx")

# PHENODATA ORIGINAL
# the phenodata below assigns TB status according to Pericardial culture.
# This misclassifies TB-PC suspects who were culture positive elsewhere as
# "probable" TB (wrong)
# phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_25_9_2012.csv")

# PHENODATA EDIT 1
# the phenodata (class3 variable) below defines active TB as ANY positive
# site. This means that some probable TBPC cases are definite TB cases.
# This is relevant as the !blood! signature will be determined by any (??)
# TB site. This needs to be tested where fluid is negative but another
# site positive. It is unclear if this applies to TBM, though. (issues of
# blood-brain-barrier, etc.)
# Important: when changing this, factors of diagnostic class may change,
# depending which "probable" case is made "definite".
# phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_28_10_2012.csv")

# PHENODATA EDIT 2
# With this in mind, the phenodata source file has been updated to change
# sample 20868 control back to probableActiveTB, as the only positive
# culture was CSF. It seems unwise to include only one case of TBM, while
# there are many TB-PC and PTB.
# the phenodata file was edited: sample PCA141 blood and fluid assignments to
# chips have been exchanged, as earlier mds plots strongly suggested this
# All results may change that depend on the correct assignment.

#IMPI-MA data
phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_17_01_2013.csv")
sampleInfoDF<-read.csv(phenodataPath, header = TRUE, colClasses = "character",
  comment.char = "", check.names = FALSE)
allArrays<-lumi.batch(fileName,lib.mapping='lumiHumanIDMapping',
sampleInfoFile=sampleInfoDF,verbose=TRUE)

313
A Code

colnames(exprs(allArrays))<-.allArrays$sample_name

# Berry data
berry.sampleInfoDF.train<-.read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_train.csv", header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
berry.train<-.lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MathewRequest_02Feb11_TrainingSet.txt", lib.mapping = 'lumiHumanIDMapping', sampleInfoFile = berry.sampleInfoDF.train, verbose=TRUE)

berry.sampleInfoDF.test<-.read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_test.csv", header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
berry.test<-.lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MathewRequest_02Feb11_TestSet.txt", lib.mapping = 'lumiHumanIDMapping', sampleInfoFile = berry.sampleInfoDF.test, verbose=TRUE)

berry.sampleInfoDF.val<-.read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_val.csv", header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
berry.val<-.lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MathewRequest_02Feb11_VALIDATIONSet.txt", lib.mapping = 'lumiHumanIDMapping', sampleInfoFile = berry.sampleInfoDF.val, verbose=TRUE)

# subset allArrays by compartment
fluidArrays<-.allArrays[, .allArrays$Compartment == "fluid"]
bloodArrays<-.allArrays[, .allArrays$Compartment == "blood"]

# Data subsetting####

# make subsets

# CSC scheme: compartment, subset, contrast
save(berry.train, file = file.path(dir.rdata, 'berry_train.RData'))
save(berry.test, file = file.path(dir.rdata, 'berry_test.RData'))
save(berry.val, file = file.path(dir.rdata, 'berry_val.RData'))
save(allArrays, file = file.path(dir.rdata, 'IMPI_MA.RData'))

hivNegBloodArrays<-.bloodArrays[, .bloodArrays$HIV.Status == "negative"] #tb
B.N.AX<-.hivNegBloodArrays[, .hivNegBloodArrays$class3 == "activeTB" | .hivNegBloodArrays$class3 == "notActiveTB"]#training
B.N.ax<-.hivNegBloodArrays[, .hivNegBloodArrays$class3 == "probableActiveTB" | .hivNegBloodArrays$class3 == "notActiveTB"]#validation
save(B.N.AX, file = file.path(dir.rdata, 'blood_hivNeg_atB_noTB.RData'))
save(B.N.ax, file = file.path(dir.rdata, 'blood_hivNeg_probTB_noTB.RData'))

hivPosBloodArrays<-.bloodArrays[, .bloodArrays$HIV.Status == "positive"]
B.P.AX<-.hivPosBloodArrays[, .hivPosBloodArrays$class3 == "activeTB" | .hivPosBloodArrays$class3 == "notActiveTB"]
B.P.ax<-.hivPosBloodArrays[, .hivPosBloodArrays$class3 == "probableActiveTB" | .hivPosBloodArrays$class3 == "notActiveTB"]
A.3 Data manager (general)

```r
save(B.P.AX, file=file.path(dir.rdata, 'blood_hivPos_aTB_noTB.RData'))
save(B.P.aX, file=file.path(dir.rdata, 'blood_hivPos_probTB_noTB.RData'))
##
B.aTB.PN<- bloodArrays[, bloodArrays$class3=="activeTB"] #tb
save(B.aTB.PN, file=file.path(dir.rdata, 'blood_aTB_HIVposNeg.RData'))
B.noTB.PN<- bloodArrays[, bloodArrays$class3=="notActiveTB"] #tb
save(B.noTB.PN, file=file.path(dir.rdata, 'blood_noTB_HIVposNeg.RData'))
##
B.C<- bloodArrays[, bloodArrays$type=="PC"] #hiv,hd
PF.C<- fluidArrays[, fluidArrays$type=="PC"] #hiv, hd
save(B.C, file=file.path(dir.rdata, 'blood_TBPC.RData'))
save(PF.C, file=file.path(dir.rdata, 'fluid_TBPC.RData'))
C.BPF<- allArrays[, allArrays$matching=="m"] #comp
C.N.BPF<-C.BPF[, C.BPF$HIV.Status=="negative"]
C.P.BPF<-C.BPF[, C.BPF$HIV.Status=="positive"]
save(C.BPF, file=file.path(dir.rdata, 'TBPC_matchedBloodFluid.RData'))
save(C.N.BPF, file=file.path(dir.rdata, 'TBPC_hivNeg_matchedBloodFluid.RData'))
save(C.P.BPF, file=file.path(dir.rdata, 'TBPC_hivPos_matchedBloodFluid.RData'))

datasets<- list(
  "Berry training set"=berry.train,
  "Berry test set"=berry.test,
  "Berry validation set"=berry.val,
  "IMPI_MA"=allArrays,
  "Blood, HIV neg, active or not active TB"=B.N.AX,
  "Blood, HIV neg, probable or not active TB"=B.N.aX,
  "Blood, HIV pos, active or not active TB"=B.P.AX,
  "Blood, HIV pos, probable or not active TB"=B.P.aX,
  "Blood, all TBPC"=B.C,
  "Pericardial Fluid, all TBPC"=PF.C,
  "Matched blood and pericardial fluid"=C.BPF,
  "Matched blood and pericardial fluid, HIV neg"=C.N.BPF,
  "Matched blood and pericardial fluid, HIV pos"=C.P.BPF)

#Berry signatures
#file=file.path(dir.gx,"Nature_paper_transcripts","P22_42_04Mar09_Training_393_GX11.txt")
#np393<- read.table(file,header=TRUE,sep="\t")
#char393<-as.character(np393[,1])

#berry.sig<- char393
#save(berry.sig, file=file.path(dir.rdata, 'berry.sig.RData'))

#need to get 86 transcript list; downloaded as xls

#import
file86 = file.path(dir.gx,"Nature_paper_transcripts","nature09247-s4.csv")
np86<- read.csv(file86, header=TRUE, sep="","")
berry.86<-np86
save(berry.86, file=file.path(dir.rdata, 'berry.86.RData'))

#not done, need to fix format...
```
```r
file393 <- file.path(dir.gx,"Nature_paper_transcripts","nature09247-s2.csv")
np393 <- read.csv(file393, header=TRUE, sep="","")
berry.393 <- np393
save(berry.393, file = file.path(dir.rdata,'berry.393.RData'))

# convert illumina ID to nuID
# as character
# save as .RData
```

Listing A.7: Data-manager.R
A.4 RT-PCR

### setup####
```r
source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/03_RT-PCR/11_RTPCR_paper/RTPCR_System_setup.R")
```

```r
dir.rtpcrdata <- file.path(dir.data_root,'02_IMPI_immunology data/Assays/RT-PCR')
```

```r
dir.phenodata <- file.path("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/02_output/02_Phenotype exploration/")
```

# Output folders
```r
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/03_RT-PCR/11_RTPCR_paper')
```

# Current date string used for versioning output
ds<-Sys.time()

# versioned output
```r
dir.output.version<-file.path(dir.main,paste("ver_",ds))
```

```r
## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
```

```r
## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
```

```r
for (dir in c(dir.main, dir.figures, dir.results)) {
  if (!file.exists(dir)) {
    dir.create(dir, recursive=T,showWarnings=T)
  }
}
```

```r
sink(type="output",file=file.path(dir.results,"session_output.txt"))
```

```r
sessionInfo()
```

```r
print(paste("Begin time: ",ds))
```

```r
print(paste("Results will be saved to", dir.results))
```

```r
print(paste("Figures will be saved to", dir.figures))
```

### additional functions####
```r
redgreen <- function(n){c(hsv(h=2/6,v=seq(1,0,length=n/2)),hsv(h=0/6,v=seq(0,1,length=n/2)))}
```

```r
bluered <- colorRampPalette(c("blue","lightblue","white","pink","red")) ## (n)
```

```r
```
interleave <- function(v1, v2)
{
  ord1 <- 2*(1:length(v1)) - 1
  ord2 <- 2*(1:length(v2))
  c(v1, v2)[order(c(ord1, ord2))]
}

# Data

# delta CT values
data2 <- read.xls(file.path(dir.rtpcrdata, "rtDat2.XLS"))
# ELISA and luminex
data2.prot <- read.xls(file.path(dir.rtpcrdata, "elDat.XLS"))
# Phenotype information
# make sure to use the latest! The below can vary as the output from the
# phenotype explorer in Mathematica changes!
phenodata <- read.xls(file.path(dir.phenodata, "SET1_RT-PCRsumdata +
  class20130409.xls"))
# Combine two datasets
data3 <- merge(phenodata, data2, by.x="ID_KERRYN", by.y="ID")
data3.prot <- merge(phenodata, data2.prot, by.x="ID_KERRYN", by.y="PersonID")
# About the patients: Table 1 calculations

pts2 <- phenodata
# variables
hiv <- pts2$HIV
tabs <- ftable(hiv)
tabs2 <- ftable(hiv, hiv)
tabsp <- chisq.test(tabs2)$p.value

sex <- pts2$SEX
table(sex)
tabsex <- ftable(sex, hiv)
tabsexp <- chisq.test(tabsex)$p.value

fluid <- pts2$Fluid
table(fluid)
tablefluid <- ftable(fluid, hiv)
tablefluidp <- chisq.test(tablefluid)$p.value

onart <- pts2$ARVS
onarttable <- ftable(onart, hiv)
onartp <- chisq.test(onarttable)$p.value

onster <- pts2$STEROIDS
onster <- ftable(onster, hiv)
onsterp <- chisq.test(onsterp)$p.value

m <- tapply(pts2$AGE_CAL, hiv, median, na.rm=T)
i <- tapply(pts2$AGE_CAL, hiv, summary)
x <- tapply(pts2$AGE_CAL, hiv, length)
A.4 RT-PCR

```r
summary(pts2$AGE_CALC)
#boxplot(pts2$AGE_CALC)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$AGE_CALC~hiv)

m <- tapply(pts2$CD4,hiv,median,na.rm=T)
i <- tapply(pts2$CD4,hiv,summary)
n <- tapply(pts2$CD4,hiv,length)
summary(pts2$CD4)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$CD4~hiv)

m <- tapply(pts2$ADA,hiv,median,na.rm=T)
i <- tapply(pts2$ADA,hiv,summary)
n <- tapply(pts2$ADA,hiv,length)
summary(pts2$ADA)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$ADA~hiv)

m <- tapply(pts2$IFN.gamma,hiv,median,na.rm=T)
i <- tapply(pts2$IFN.gamma,hiv,summary)
n <- tapply(pts2$IFN.gamma,hiv,length)
summary(pts2$IFN.gamma)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$IFN.gamma~hiv)

# Select compartment ####

# Blood only
data3blood <- data3[data3$compart == "Blood",]
# Fluid only
data3fluid <- data3[data3$compart == "Fluid",]

#the phenotype categories ####
compart <- as.factor(data3$compart); levels(compart) <- c("red","yellow")
sex <- as.factor(data3$SEX); levels(sex) <- c("white","blue","pink")
h <- as.factor(data3$HIV); levels(h) <- c("green","orange","white")
lowcd <- as.factor(data3$CD4.200); levels(lowcd) <- c("white","black","gray")
cd4 <- as.factor(data3$CD4); levels(cd4) <- sequential_hcl(28, h = 100, c = 100, power = 1.2)
hd <- as.factor(data3$ECP); levels(hd) <- brewer.pal(3,"Set1")
tbs <- as.factor(data3$TB.PC); levels(tbs) <- brewer.pal(3,"Set2")
tbc <- as.factor(data3$PCF.TB.culture); levels(tbc) <- brewer.pal(4,"Set3")
fluid <- as.factor(data3$Fluid); levels(fluid) <- c("white","white","red","pink","yellow")

#phenomatrix definitions
phenomatrix <- matrix(cbind(as.character(tbc),as.character(tbs),as.character(hd),as.character(lowcd),as.character(cd4),as.character(sex),as.character(compart),as.character(fluid)),ncol=9)
rownames(phenomatrix) <- c("TBC","TBS","HD","HIV","lowCD4","CD4","SEX","COMP","Fluid")
phenomatrix <- matrix(cbind(as.character(h),as.character(compart)),ncol=2)
rownames(phenomatrix) <- c("HIV","COMP")

# for blood
compart.blood <- as.factor(data3blood$compart); levels(compart.blood) <- c("red","yellow")
sex.blood <- as.factor(data3blood$SEX); levels(sex.blood) <- c("white","blue","pink")
```
h. blood <- as.factor(data3blood$HIV); levels(h. blood) <- c("green", "orange", "white")

lowcd. blood <- as.factor(data3blood$CD4 < 200); levels(lowcd. blood) <- c("white", "black")

cd4. blood <- as.factor(data3blood$CD4); levels(cd4. blood) <- sequential_hcl(28, h=100, c=100, power = 1.2)

hd. blood <- as.factor(data3blood$ECP); levels(hd. blood) <- brewer.pal(3, "Set1")

tbs. blood <- as.factor(data3blood$TB.PC); levels(tbs. blood) <- brewer.pal(3, "Set2")

tbc. blood <- as.factor(data3blood$PCF.TB.culture); levels(tbc. blood) <- brewer.pal(4, "Set3")

hd. blood <- as.factor(data3blood$ECP); levels(hd. blood) <- brewer.pal(4, "Set3")

tbs. blood <- as.factor(data3blood$TB.PC); levels(tbs. blood) <- brewer.pal(4, "Set3")

tbc. blood <- as.factor(data3blood$PCF.TB.culture); levels(tbc. blood) <- brewer.pal(4, "Set3")

# Unfiltered data ####

# datamat <- as.matrix(data3[, -40:-1])

# transform RT-PCR data to LFC

datamat = log10(2^- datamat)

# Normalization methods ####

# datamat <- t(normalizeQuantileRank.matrix(t(datamat), robust = TRUE))

# datamat <- t(normalizeQuantile(t(datamat)))

# datamat <- t(normalizeBetweenArrays(t(datamat)))

# datamat <- t(normalizeMedianValues(t(datamat)))

# datamat <- t(normalizeAverage.matrix(t(datamat)))

# # Start here: separate quantile normalization of blood and fluid, then combining to one matrix which is median scaled, then transformed to expression levels####

bloodmat <- as.matrix(data3blood[, -40:-1])

rownames(bloodmat) <- data3blood$ID2

# bloodmatN <- t(normalizeQuantileRank.matrix(t(bloodmat), robust = TRUE))

# bloodmatN <- t(normalizeBetweenArrays(t(bloodmat), method = "quantile"))

# bloodmatN <- log10(2^- bloodmatN)
fluidmat <- as.matrix(data3fluid[, -40: -1])
rownames(fluidmat) <- data3fluid$ID2
# fluidmatN <- t(normalizeQuantileRank(matrix(t(fluidmat), robust=TRUE)))
# fluidmatN <- t(normalizeBetweenArrays(t(fluidmat), method="quantile"))
# fluidmatN <- log10(2^-fluidmatN)
#
datamat <- rbind(bloodmatN, fluidmatN)
datamat <- t(normalizeMedianValues(t(datamat)))
#
# not normalized at all (argument for this approach is that all values are
# normalized to beta Actin already, which is universally constant):
datamat <- rbind(bloodmat, fluidmat)

# transformation
datamat = log10(2^-datamat)

ref.min <- min(datamat)
ref.max <- max(datamat)

# phenomatrix####
phenomatrix = cbind(c(rep("red",27), rep("yellow",27)), c(rep("red",27), rep("yellow",27)))
calphenomatrix = rbind(c("blue","blue"), c("blue","blue"), phenomatrix)

# gene categories
genecol = matrix(c(cv, cv), ncol=2)
rownames(genecol) <- colnames(datamat)

## Unfiltered raw data: Boxplots####
figure <- 1
setwd(dir.figures)

par(parbackup)
boxplot(datamat~factor(c(rep("blood",nrow(data3)/2), rep("fluid",nrow(data3)/2))),col=c("red","yellow"),main="Overall expression values for blood and pericardial fluid")
export.plot(file.prefix=paste("fig",figure,"."), boxplot overall expression blood and fluid), export.formats=export.formats.plots, height=height*1.5, width=(1+sqrt(5))/2*height*1.5)
figure <- figure +1

m.blood <- median(datamat[1:27,])
m.fluid <- median(datamat[28:54,])
blood.fluid.overall.comparison <- wilcox.test(datamat~factor(c(rep("blood", nrow(data3)/2), rep("fluid", nrow(data3)/2))))

boxplot(t(datamat), las=2, col=c(rep("red",nrow(data3)/2), rep("yellow",nrow(data3)/2)), main="Boxplot of expression values by sample")
export.plot(file.prefix=paste("fig",figure,"."), boxplot overall expression blood and fluid), export.formats=export.formats.plots, height=height*1.5, width=(1+sqrt(5))/2*height*1.5)
figure <- figure +1

npar <- par(mar=c(6,3,3,19), xpd=TRUE)
boxplot(datamat, las=2, col=cv, main="Expression in blood and fluid")
legend(45, -5, c("TH1", "TH2", "TH17", "Immunoregulation", "Neutrophil", "Growth factors", "Fibrosis", "MMPs"), fill=c("red", "green", "yellow", "blue", "white", "purple", "pink", "gray"), cex=.8)

export.plot(file.prefix=paste("fig", figure, ".", "boxplot"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

boxplot(datamat[1:27,], las=2, col=cv, main="Expression in blood")
legend(45, -5, c("TH1", "TH2", "TH17", "Immunoregulation", "Neutrophil", "Growth factors", "Fibrosis", "MMPs"), fill=c("red", "green", "yellow", "blue", "white", "purple", "pink", "gray"), cex=.8)
export.plot(file.prefix=paste("fig", figure, ".", "boxplot"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

boxplot(datamat[28:54,], las=2, col=cv, main="Expression in fluid")
legend(89, -5, c("blood", "fluid"), fill=c("red", "yellow"), cex=.8)
export.plot(file.prefix=paste("fig", figure, ".", "boxplot"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

ord<-interleave(seq(1, 42), seq(43, 84))
newdat<-cbind(datamat[1:27,], datamat[28:54,])
boxplot(newdat[, ord], col=c("red", "yellow"), las=2, main="Expression levels of all genes")
legend(89, -5, c("blood", "fluid"), fill=c("red", "yellow"), cex=.8)
export.plot(file.prefix=paste("fig", figure, ".", "boxplot"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

mpar<-par(mar=c(5, 3, 3, 3), xpd=TRUE, mfrow=c(2, 2))
boxplot(newdat[, ord][, 1:16], col=c("red", "yellow"), ylab="expression", las=2, main="TH1")
boxplot(newdat[, ord][, 17:20], col=c("red", "yellow"), ylab="expression", las=2, main="TH2")
boxplot(newdat[, ord][, 21:26], col=c("red", "yellow"), ylab="expression", las=2, main="TH17")
boxplot(newdat[, ord][, 27:32], col=c("red", "yellow"), ylab="expression", las=2, main="Immunoregulation")
#legend("bottomleft", c("blood", "fluid"), fill=c("red", "yellow"), cex=.8)
export.plot(file.prefix=paste("fig", figure, ".", "boxplot"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

boxplot(newdat[, ord][, 33:42], col=c("red", "yellow"), ylab="expression", las=2, main="Neutrophil")
boxplot(newdat[, ord][, 43:48], col=c("red", "yellow"), ylab="expression", las=2, main="Growth factors")
boxplot(newdat[, ord][, 49:62], col=c("red", "yellow"), ylab="expression", las=2, main="Fibrosis")
boxplot(newdat[, ord][, 63:84], col=c("red", "yellow"), ylab="expression", las=2, main="MMPs")
#legend(15.5, -10, c("blood", "fluid"), fill=c("red", "yellow")) par(parbackup)
export.plot(file.prefix=paste("fig", figure, ".", "boxplot"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
A.4 RT-PCR

```r
figure <- figure + 1
par(parbackup)

npar <- par(mar=c(5.5,3,3,3),xpd=TRUE,mfrow=c(4,2))

boxplot(newdat[,ord][,1:16],col=c("red","yellow"),ylab="expression",las=2,
main="TH1")
boxplot(newdat[,ord][,17:20],col=c("red","yellow"),ylab="expression",las=2,
main="TH2")
boxplot(newdat[,ord][,21:26],col=c("red","yellow"),ylab="expression",las=2,
main="TH17")
boxplot(newdat[,ord][,27:32],col=c("red","yellow"),ylab="expression",las=2,
main="Immunoregulation")

# legend("bottomleft",c("blood","fluid"),fill=c("red","yellow"),cex=.8,horiz=T)

boxplot(newdat[,ord][,33:42],col=c("red","yellow"),ylab="expression",las=2,
main="Neutrophil")
boxplot(newdat[,ord][,43:48],col=c("red","yellow"),ylab="expression",las=2,
main="Growth factors")
boxplot(newdat[,ord][,49:62],col=c("red","yellow"),ylab="expression",las=2,
main="Fibrosis")
boxplot(newdat[,ord][,63:84],col=c("red","yellow"),ylab="expression",las=2,
main="MMPs")

# legend(15.5,-10,c("blood","fluid"),fill=c("red","yellow"))par(parbackup)
export.plot(file.prefix=paste("fig",figure,"boxplot"),export.formats=export.formats.plots,height=height*3,width=(1+sqrt(5))/2*height*1.5)

figure <- figure + 1
par(parbackup)

genes <- colnames(datamat)
pvals.gene.by.compartment <- vector()
for (i in 1:42) {
  pvals.gene.by.compartment[i] <- wilcox.test(datamat[1:27,i],datamat[28:54,i])$p.value
}
bonf <- p.adjust(pvals.gene.by.compartment,method="bonferroni")
bb <- p.adjust(pvals.gene.by.compartment,method="BH")
simple.comp.results <- data.frame(genes,pvals.gene.by.compartment,bonf,bb)
write.csv(simple.comp.results,file.path(dir.results,"genes_by_compartement.csv"))

# FIGURE 1 A: Clustered raw data heatmaps (microarray style) ####
heatmap.plus(t(datamat),scale="none",ColSideColors=phenomatrix,
RowSideColors=genecol,col=redgreen(30),cexCol=1.2,cexRow=1,margins=c(10,30))
par(parbackup)
heatmap.plus(t(datamat),scale="none",ColSideColors=phenomatrix,col=redgreen
(30),cexCol=1.2,cexRow=1,margins=c(10,24))
#HC using euclidean distance, complete linkage
s=round(seq(from=min(datamat),to=max(datamat),by=1),1)
colvec=redgreen(length(s))
legend("right",as.character(s),fill=colvec,ncol=1,cex=.8,y.intersp=.8,title="LFC")
```
```r
# experimental calibrated heatmap
nr <- nrow(t(datamat))
hi <- rep(ref.max, nr)
lo <- rep(ref.min, nr)
caldata <- cbind(hi, lo, t(datamat))
heatmap.plus(caldata, scale="none", ColSideColors=calphenomatrix, col=redgreen(30), cexCol=1.2, cexRow=1, margins=c(10, 24))
s <- round(seq(from=min(datamat), to=max(datamat), by=1), 1)
colvec <- redgreen(length(s))
legend("right", as.character(s), fill=colvec, ncol = 1, cex = .8, y.intersp = .8, title="LFC")
export.plot(file.prefix=paste("fig", figure., ".", "all_genes_heatmap"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure <- figure +1
par(parbackup)
# FIGURE 1 B and C: Correlation matrices####
# GENES
# the gene expression levels are both positively and negatively correlated
# cormat <- cor(datamat, method="pearson")
# heatmap_2(cormat, col=redgreen(300), scale="none", legend=2, legfrac=8, trim = .01)

cormatB <- cor(bloodmat, method="pearson")
heatmap_2(cormatB, col=bluered(34), scale="none", legend=2, legfrac=8)
heatmap_2(cormatB, col=bluered(34), symkey=F, symbreaks=T, trace="none", density.info="none")
export.plot(file.prefix=paste("fig", figure., ".", "cormat_B"), export.formats=export.formats.plots, height=height*1.5, width=height*1.5)
figure <- figure +1

cormatF <- cor(fluidmat, method="pearson")
heatmap_2(cormatF, col=bluered(34), scale="none", legend=2, legfrac=8)
heatmap_2(cormatF, col=bluered(34), symkey=F, symbreaks=T, trace="none", density.info="none")
export.plot(file.prefix=paste("fig", figure., ".", "cormat_F"), export.formats=export.formats.plots, height=height*1.5, width=height*1.5)
figure <- figure +1

library(plotrix)
cormatB2 <- rescale(cormatB, c(-1,1))

# Data filtering step####
# non-specific filtering: all
ct=38
fl <- kOverA(floor(.95*nrow(datamat)), log(2^-ct, 10))
list <- filterfun(fl)
ans <- genefilter(t(datamat), flist)
datamat.filter <- t(t(datamat)[ans,])
fl2 <- kOverA(floor(.95*nrow(datamat[1:27,])), log(2^-ct, 10))
list.2 <- filterfun(fl2)
ans.2 <- genefilter(t(datamat[1:27,]), flist.2)
bloodmat.filter <- t(t(datamat[1:27,])[ans.2,])
fl3 <- kOverA(floor(.95*nrow(datamat[28:54,])), log(2^-ct, 10))
```
A.4 RT-PCR

```r
flist.fluid <- filterfun(f3)
ans.fluid <- genefilter(t(damat)[28:54,], flist.fluid)
fluidmat.filter <- t(t(damat)[28:54,][ans.fluid,])

#FIGURE S1A Clustered raw data heatmaps (microarray style) ##Filtered raw data####
par(parbackup)
heatmap.plus(t(damat), scale="none", ColSideColors=phenomatrix,
RowSideColors=genecol, col=redgreen(50), cexCol=1.2, cexRow=1, margins=c(10,30))
heatmap.plus(t(damat.filter), scale="none", ColSideColors=phenomatrix, col=
redgreen(50), cexCol=1.2, cexRow=1, margins=c(10,24))

# HC using euclidean distance, complete linkage
s=round(seq(from=min(damat.filter), to=max(damat.filter), by=1),1)
colvec=redgreen(length(s))
legend("right", as.character(s), fill=colvec, ncol=1, cex=.8, y.intersp=.8, title="LFC")
	nr.filter <- nrow(t(damat.filter))
hi.filter <- rep(ref.max, nr.filter)
lo.filter <- rep(ref.min, nr.filter)
caldata.filter <- cbind(hi.filter, lo.filter, t(damat.filter))
heatmap.plus(caldata.filter, scale="none", ColSideColors=calphenomatrix, col=
redgreen(50), cexCol=1.2, cexRow=1, margins=c(10,24))
s=round(seq(from=min(damat), to=max(damat), by=1),1)
colvec=redgreen(length(s))
legend("right", as.character(s), fill=colvec, ncol=1, cex=.8, y.intersp=.8, title="LFC")

export.plot(file.prefix=paste("fig", figure, "." ,"filtered_genes_heatmap"),
export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5) )/2*height*1.5)

#FIGURE S1 B and C: Correlation matrices ##Filtered raw data####

cormatB.filter <- cor(bloodmat.filter, method="pearson")
heatmap_2(cormatB.filter, col=bluered(34), scale="none", legend=2, legfrac=8)
export.plot(file.prefix=paste("fig", figure, "." ,"filtered_genes_correlationMatrix_blood"),export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5) )/2*height*1.5)

figure <- figure+1

cormatF.filter <- cor(fluidmat.filter, method="pearson")
heatmap_2(cormatF.filter, col=bluered(34), scale="none", legend=2, legfrac=8, trim=.01)
export.plot(file.prefix=paste("fig", figure, "." ,"filtered_genes_correlationMatrix_fluid"),export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5) )/2*height*1.5)

figure <- figure+1

#Moderated test statistics for blood versus fluid####

design = model.matrix(~-1+factor(comp))
colnames(design) <- levels(comp)
```
# Linear model: blood vs fluid
fit <- lmFit(t(datamat.filter),design)
fit2 <- contrasts.fit(fit, contrast)
fit2 <- eBayes(fit2)

# Results
result1 <- topTable(fit2, number = vc2[,2][2], adjust.method = "BH", coef = 1, sort.by = "P", resort.by = "logFC")
result1

# Table 2
result1table <- xtable(result1[,c(1,2,4,5)], digits = c(NA,NA,2,-3,-3))
write.csv(result1table, file.path(dir.results, "SelectedGeneTopTable.csv"))

# FIGURE 2A: Clustered result data heatmap (microarray style)
heatmap.plus(t(datamat.filter)[result1$ID,], scale = "none", ColSideColors = phenomatrix, col = redgreen(30), cexCol = 1.2, cexRow = 1, margins = c(10,24))

# HC using euclidean distance, complete linkage
s=round(seq(from=min(datamat[,,result1$ID]),to=max(datamat[,,result1$ID]),by=1),1)
colvec=redgreen(length(s))
legend("right", as.character(s), fill = colvec, ncol = 1, cex = .8, y.intersp = .8, title = "LFC")

nr.select <- nrow(t(datamat.filter)[result1$ID,])
hi.select <- rep(ref.max, nr.select)
lo.select <- rep(ref.min, nr.select)
caldata.select <- cbind(hi.select, lo.select, t(datamat.filter)[result1$ID,])
heatmap.plus(caldata.select, scale = "none", ColSideColors = calphenomatrix, col = redgreen(30), cexCol = 1.2, cexRow = 1, margins = c(10,24))
s=round(seq(from=min(datamat), to=max(datamat), by=1),1)
colvec=redgreen(length(s))
469 legend("right", as.character(s), fill=colvec, ncol = 1, cex = .8, y.intersp = .8, title="LFC")
470 export.plot(file.prefix=paste("fig", figure, ".", "selected_genes_heatmap"),
471 export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/(2*height*1.5))
472 figure<-figure+1
473 #FIGURE 2 B and C: Correlation matrices for selected genes####
474 475 cormatB.select<-cor(bloodmat.filter[,result1$ID], method="pearson")
476 heatmap.2(cormatB.select, col=bluered(34), scale="none", legend=2, legfrac=8, trim=.01)
477 export.plot(file.prefix=paste("fig", figure, ".", "selected_genes_correlationMatrix_blood"),
478 export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/(2*height*1.5))
479 figure<-figure+1
480 481 cormatF.select<-cor(fluidmat.filter[,result1$ID], method="pearson")
482 heatmap.2(cormatF.select, col=bluered(34), scale="none", legend=2, legfrac=8, trim=.01)
483 export.plot(file.prefix=paste("fig", figure, ".", "selected_genes_correlationMatrix_fluid"),
484 export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/(2*height*1.5))
485 figure<-figure+1
486 487 #FIGURE 3 PCA (oompa package)####
488 par(parbackup)
489 par(mfrow=c(1,3), mar=c(10,10,6,6))
490 par(xaxt="s")
491
492 #all genes
493 l=rownames(datamat)
494 trueClasses <- as.factor(comp)
495 spca <- SamplePCA(t(datamat), trueClasses)
496 plot(spca, col=c("red", "blue"),main="All genes", cex=2, cex.axis=2, cex.main =3, cex.lab=2.5)
497 mtext(sprintf("%.2f", spca@variances[1]/sum(spca@variances)), side=1, line=3, adj=1, cex=1.5)
498 mtext(sprintf("%.2f", spca@variances[2]/sum(spca@variances)), side=2, line=3, adj=1, cex=1.5)
499 # mark the group centers
500 x1 <- predict(spca, matrix(apply(t(datamat[grep("b", rownames(datamat))]), 1, mean), ncol=1))
501 points(x1[1], x1[2], col = 'red', cex=6, pch=9)
502 x2 <- predict(spca, matrix(apply(t(datamat[grep("f", rownames(datamat))]), 1, mean), ncol=1))
503 points(x2[1], x2[2], col = 'blue', cex=6, pch=9)
504
505 #filtered genes
506 l=rownames(datamat.filter)
507 trueClasses1 <- as.factor(comp)
508 spca1 <- SamplePCA(t(datamat.filter), trueClasses1)
509 plot(spca1, col=c("red", "blue"),main="Filtered genes", cex=2, cex.axis=2, cex.main =3, cex.lab=2.5)
510 mtext(sprintf("%.2f", spca1@variances[1]/sum(spca1@variances)), side=1, line=3, adj=1, cex=1.5)
511 mtext(sprintf("%.2f", spca1@variances[2]/sum(spca1@variances)), side=2, line=3, adj=1, cex=1.5)
512 # mark the group centers
x1 <- predict(spca1, matrix(apply(t(datamat.filter[grep("b", rownames(datamat))]), 1, mean), ncol=1))
points(x1[1], x1[2], col='red', cex=6, pch=9)

x2 <- predict(spca1, matrix(apply(t(datamat.filter[grep("f", rownames(datamat))]), 1, mean), ncol=1))
points(x2[1], x2[2], col='blue', cex=6, pch=9)

# selected genes
trueClasses2 <- as.factor(comp)
spca2 <- SamplePCA(t(datamat.filter)[result1$ID,], trueClasses2)
plot(spca2, col = c("red", "blue"),main="Selected genes",cex=2, cex.axis=2, cex.main=3, cex.lab=2.5)
mtext(sprintf("%.2f", spca2@variances[1]/sum(spca2@variances)),side=1,line=3, adj=1,cex=1.5)
mtext(sprintf("%.2f", spca2@variances[2]/sum(spca2@variances)),side=2,line=3, adj=1,cex=1.5)

# mark the group centers
x1 <- predict(spca2, matrix(apply(t(datamat.filter[grep("b", rownames(datamat))])[result1$ID,], 1, mean), ncol=1))
points(x1[1], x1[2], col='red', cex=6, pch=9)

x2 <- predict(spca2, matrix(apply(t(datamat.filter[grep("f", rownames(datamat))])[result1$ID,], 1, mean), ncol=1))
points(x2[1], x2[2], col='blue', cex=6, pch=9)

sprintf("%.10f",0.25)

export.plot(file.prefix=paste("fig",figure,".","PCA of 3 data subsets"), export.formats=export.formats.plots,height=height*1.5,width=2*(1 + sqrt(5))/2*height*1.5)

figure <- figure +1

par(parbackup)
# screeplot(spca)
# screeplot(spca1)
# screeplot(spca2,type="lines")
# screeplot(spca2)

#TABLE S1: Phenodata
phenodataTS2 <- phenodata[,c(2,4,5,12,13,15,14,26,27,28,35,36,37,38)]
write.csv(phenodataTS2, file.path(dir.results,"TABLES2.csv"))

# print(pdat)

# Protein
subjects <- rownames(datamat)

#Blood only
data3blood.prot <- data3.prot[data3.prot$compart=="blood",]

#Fluid only
data3fluid.prot <- data3.prot[data3.prot$compart=="fluid",]

bloodmat.p <- matrix(as.numeric(as.matrix(data3blood.prot[,40:-1])),nrow=25)
rownames(bloodmat.p) <- data3blood.prot$SampleID
fluidmat.p <- matrix(as.numeric(as.matrix(data3fluid.prot[,40:-1])),nrow=25)
rownames(fluidmat.p) <- data3fluid.prot$SampleID

328
```r
# RT-PCR

datamat.p <- as.matrix(rbind(bloodmat.p, fluidmat.p))
colnames(datamat.p) <- c("IL1B", "IL6", "IL10", "TNF", "IFNG", "CXCL10", "IL13", "IL17", "IL22", "IL23a", "TGFB1", "IL10", "IL8", "IL12P70", "GAL3", "ACSDKP", "TIMP1", "TIMP2", "MMP2TIMP1", "MMP9TIMP1", "MMP1", "MMP2", "MMP3", "MP7", "MMP8", "MMP9")

data.mat.p.sel <- datamat.p[, is.element(colnames(datamat.p), result1$ID)]
ord.p <- interleave(seq(1,9), seq(10,18))
newdat.p <- cbind(datamat.p.sel[,1:25,], datamat.p.sel[,26:50,])
par(parbackup)
boxplot(newdat.p[,ord.p], col=c("red", "yellow"), las=2, main="Levels of 9 proteins")
legend("topright", c("blood", "fluid"), fill=c("red", "yellow"), cex=.8)

data.mat.p.notsel <- datamat.p[, !is.element(colnames(datamat.p), result1$ID)]
ord.p.notsel <- interleave(seq(1,17), seq(18,34))
newdat.p.notsel <- cbind(datamat.p.notsel[,1:25,], datamat.p.notsel[,26:50,])
par(parbackup)
boxplot(newdat.p.notsel[,ord.p.notsel], col=c("red", "yellow"), las=2, main="Levels of other proteins")
legend("topright", c("blood", "fluid"), fill=c("red", "yellow"), cex=.8)

# plot both on one page (log10 scale)
par(mfrow=c(2,1))
boxplot(log10(newdat.p[,ord.p]+1), col=c("red", "yellow"), las=2, main="Levels of 9 proteins (where mRNA was DE)")
legend("topleft", c("bl", "fl"), fill=c("red", "yellow"), cex=.7)
boxplot(log10(newdat.p.notsel[,ord.p.notsel]+1), col=c("red", "yellow"), las=2, main="Levels of other proteins (where mRNA was not DE)")
legend("topleft", c("bl", "fl"), fill=c("red", "yellow"), cex=.7)
par(parbackup)
export.plot(file.prefix=paste("fig", figure, ".", "boxplots of proteins by compartment"), export.formats=export.formats.plots, height=height*1.5, width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

proteins<-colnames(datamat.p)
pvals.protein.by.compartment<-vector()
for (i in 1:26){
  try(pvals.protein.by.compartment[i]<-wilcox.test(datamat.p[,1:25,i], datamat.p[,26:50,i])$p.value)
}
bonf.prot<-p.adjust(pvals.protein.by.compartment, method="bonferroni")
bh.prot<-p.adjust(pvals.protein.by.compartment, method="BH")
simple.comp.results.prot<-data.frame(proteins, pvals.protein.by.compartment, bonf.prot, bh.prot)
write.csv(simple.comp.results.prot, file.path(dir.results, "proteins_by_compartments.csv"))

proteins.sel<-colnames(datamat.p.sel)
pvals.protein.by.compartment.sel<-vector()
for (i in 1:ncol(datamat.p.sel)){
  try(pvals.protein.by.compartment.sel[i]<-wilcox.test(datamat.p.sel[,1:25,i], datamat.p.sel[,26:50,i])$p.value)
}
bonf.prot.sel<-p.adjust(pvals.protein.by.compartment.sel, method="bonferroni")
bh.prot.sel<-p.adjust(pvals.protein.by.compartment.sel, method="BH")
```
```r
simple.comp.results.prot.sel <- data.frame(proteins.sel, pvals.protein.by.compartment.sel, bonf.prot.sel, bh.prot.sel)
write.csv(simple.comp.results.prot.sel, file.path(dir.results, "selected_proteins_by_compartment.csv"))

proteins.notsel <- colnames(datamat.p.notsel)
pvals.protein.by.compartment.notsel <- vector()
for (i in 1:ncol(datamat.p.notsel)){
  try(pvals.protein.by.compartment.notsel[i] <- wilcox.test(datamat.p.notsel[1:25,i], datamat.p.notsel[26:50,i])$p.value)
}
bonf.prot.notsel <- p.adjust(pvals.protein.by.compartment.notsel, method="bonferroni")
bh.prot.notsel <- p.adjust(pvals.protein.by.compartment.notsel, method="BH")
simple.comp.results.prot.notsel <- data.frame(proteins.notsel, pvals.protein.by.compartment.notsel, bonf.prot.notsel, bh.prot.notsel)
write.csv(simple.comp.results.prot.notsel, file.path(dir.results, "non_selected_proteins_by_compartment.csv"))

# median ratios of proteins
apply(datamat.p.sel[26:50,],2,median,na.rm=T)/apply(datamat.p.sel[1:25,],2,median,na.rm=T)

# THE END OF TABLES AND FIGURES FOR THE PAPER ####
# NOTHING BELOW IS SHOWN IN PAPER BUT LACK OF HIV EFFECT NEEDS TO BE MENTIONED!

# Moderated test statistics for blood ONLY ####

# Contrast for blood
# HIV status
comp <- as.factor(data3blood$HIV)
levels(comp) <- c("Neg", "Pos")
design = model.matrix(~ -1 + factor(comp))
colnames(design) <- levels(comp)
design
contrast <- makeContrasts(Pos-Neg, levels=design)
contrast

# Linear model: blood
fit <- lmFit(t(bloodmat.filter), design)
fit2 <- contrasts.fit(fit, contrast)
fit2 <- eBayes(fit2)
gr <- decideTests(fit2, adjust.method="none")
vennDiagram(gr)
vc <- vennCounts(gr)
gr2 <- decideTests(fit2, adjust.method="fdr")
vc2 <- vennCounts(gr2)
vennDiagram(gr2)

# this shows that 2 genes are DE after BH correction

# Results
# DE without correction
result1 <- topTable(fit2, number=vc[,2][2], adjust="fdr", coef=1, sort.by="P")
result1

# Clustered result data heatmap (microarray style) blood only####
```
# Moderated test statistics for fluid ONLY###

# Contrasts and linear models for fluid

# HIV status

comp <- as.factor(data3fluid$HIV)
levels(comp) <- c("Neg", "Pos")
design <- model.matrix(~ -1 + factor(comp))
colnames(design) <- levels(comp)
design
contrast <- makeContrasts(Pos - Neg, levels = design)
contrast

fit <- lmFit(t(fluidmat.filter), design)
fit2 <- contrasts.fit(fit, contrast)
fit2 <- eBayes(fit2)
gr <- decideTests(fit2, adjust.method = "none")
vennDiagram(gr)
vc <- vennCounts(gr)
gr2 <- decideTests(fit2, adjust.method = "fdr")
vc2 <- vennCounts(gr2)
vennDiagram(gr2)

# Low CD4

comp <- as.factor(data3fluid$CD4 < 200)
levels(comp) <- c("high", "low")
design <- model.matrix(~ -1 + factor(comp))
colnames(design) <- levels(comp)
design
contrast <- makeContrasts(high - low, levels = design)
contrast

fit <- lmFit(t(fluidmat.filter), design)
fit2 <- contrasts.fit(fit, contrast)
fit2 <- eBayes(fit2)
gr <- decideTests(fit2, adjust.method = "none")
vennDiagram(gr)
vc <- vennCounts(gr)
gr2 <- decideTests(fit2, adjust.method = "fdr")
vc2 <- vennCounts(gr2)
vennDiagram(gr2)

# Results: Nothing is DE by HIV status or low CD4. The genes are not HIV genes...

Listing A.8: RT-PCR code
A.5 Relation of IMPI-MA data to BERRY dataset

```r
# TO DO
#1. table 1  
#2. export probe lists (not annotated)  
#3. annotate probe lists and export  

# setup###
source("/Users/ArminDeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System_setup.R")
source(file.path(dir.R.files,"superHeatmap.R"))
par("xpd"=FALSE)
## Individual data folders (specific for each part)
dir.data <- file.path(dir.data_root,'09_IMPI-MA_DATA/05_RData')

# Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA/DifferentialExpression','DE_Baylor')

# Current date string used for versioning output
ds<-Sys.time()

# versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
for (dir in c(dir.main, dir.figures, dir.results)) {
  if (!file.exists(dir)) {
    dir.create(dir, recursive=T,showWarnings=T)
  }
}

# load data###
load(file.path(dir.rdata,'berry_train.RData'))
load(file.path(dir.rdata,'berry_val.RData'))
load(file.path(dir.rdata,'blood_hivNeg_aTB_noTB.RData'))

# Average Normalization
# This replicates the data output by GenomeStudio, starting with raw, non-normalized summary data
```
A.5 Relation of IMPI-MA data to BERRY dataset

datalist <- list("SA blood IMPI hiv neg"=B.N.AX,"SA blood VAL hiv neg"=berry.val,"Berry Training Set"=berry.train)
# scaling normalization (as in SOFT data)
for (i in 1:length(datalist)){
datalist[[i]] <- ssn(datalist[[i]], scaling=TRUE, bgMethod='none', fgMethod='mean')
}

# Make Table 1###
# Table 1: Compare groups across the 3 sets###

# Make Table 1###
# Table 1: Compare groups across the 3 sets###

agedata <- list(as.numeric(pData(berry.train)$age),
as.numeric(pData(berry.val)$age),
as.numeric(pData(B.N.AX)$AGE_CALC))
lapply(agedata,length)
lapply(agedata,summary)
kruskal.test(agedata)

sexdata <- matrix(rbind(
table(pData(berry.train)$sex),
table(pData(berry.val)$sex),
table(pData(B.N.AX)$SEX)
),ncol=2)
rownames(sexdata) <- c("Berry.train","Berry.val","B.N.AX"); colnames(sexdata) <- c("Female","Male")
prop.test(sexdata)
table(pData(berry.train)$CrudeEthnicity)
table(pData(berry.val)$CrudeEthnicity)
table(pData(B.N.AX)$ETHNICITY)

ethnicity <- matrix(rbind(
c(11,17,7,6,0,0,1),
c(51,0,0,0,0,0,0),
c(21,0,0,0,1,0)
),ncol=7)
rownames(ethnicity) <- c("Berry.train","Berry.val","B.N.AX"); colnames(ethnicity) <- c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity, simulate.p.value=TRUE)
table(pData(berry.train)$threeclass)
table(pData(berry.val)$threeclass)
table(pData(B.N.AX)$class)
twoclass <- matrix(rbind(
c(13,29),
c(20,31),
c(12,10)
```r
# A Code

colnames(twoclass) <- c("Active TB","Not active TB")
rownames(twoclass) <- c("Berry.train","Berry.val","B.N.AX")
prop.test(twoclass)

fourclass <- matrix(
  rbind(
    c(12,17,13,0),
    c(0,31,20,0),
    c(5,5,4,8)
  ),ncol=4)
colnames(fourclass) <- c("Healthy","LTBI","PTB","TB-PC")
rownames(fourclass) <- c("Berry.train","Berry.val","B.N.AX")
fisher.test(fourclass, simulate.p.value = TRUE)

### healthy
agedata.h <- list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass == "Control",]$age),
  as.numeric(pData(berry.val)[pData(berry.val)$threeclass == "Control",]$age),
  as.numeric(pData(B.N.AX)[pData(B.N.AX)$class == "CON_healthy",]$AGE_CALC))
lapply(agedata.h, length)
lapply(agedata.h, summary)
kruskal.test(agedata.h)

tstdata.h <- list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass == "Control",]$TST),
  as.numeric(pData(berry.val)[pData(berry.val)$threeclass == "Control",]$age),
  as.numeric(pData(B.N.AX)[pData(B.N.AX)$class == "CON_healthy",]$TST))
lapply(tstdata.h, length)
lapply(tstdata.h, summary)
kruskal.test(tstdata.h)

sexdata.h <- matrix(
  rbind(
    table(pData(berry.train)[pData(berry.train)$threeclass == "Control",]$sex),
    table(pData(berry.val)$sex),
    table(pData(B.N.AX)[pData(B.N.AX)$class == "CON_healthy",]$SEX)
  ),ncol=2)
rownames(sexdata.h) <- c("Berry.train","B.N.AX"); colnames(sexdata.h) <- c("Female","Male")
prop.test(sexdata.h)

table(pData(berry.train)[pData(berry.train)$threeclass == "Control",]$CrudeEthnicity)
#table(pData(berry.val)$CrudeEthnicity)
table(pData(B.N.AX)[pData(B.N.AX)$class == "CON_healthy",]$ETHNICITY)

ethnicity.h <- matrix(
  rbind(
    c(0,12,0,0,0,0,0,0),
    c(0,0,0,0,0,0,0,0)
  ),ncol=2)
colnames(ethnicity.h) <- c("Healthy","LTBI","PTB","TB-PC")
rownames(ethnicity.h) <- c("Berry.train","Berry.val","B.N.AX")
fisher.test(ethnicity.h, simulate.p.value = TRUE)
```

334
A.5 Relation of IMPI-MA data to BERRY dataset

```r
#c(51,0,0,0,0,0,0),
c(5,0,0,0,0,0,0)
rownames(ethnicity.h)<-c("Berry.train","B.N.AX"); colnames(ethnicity.h)<-c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity.h,simulate.p.value=TRUE)

# #### LTBI
agedata.l<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$age),
as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$age),
as.numeric(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$AGE_CALC))
lapply(agedata.l,length)
lapply(agedata.l,summary)
kruskal.test(agedata.l)

tstdata.l<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$TST),
as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$TST),
as.numeric(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$TST))
lapply(tstdata.l,length)
lapply(tstdata.l,summary)
kruskal.test(tstdata.l)

sexdata.l<-matrix(
  rbind(
    table(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$sex),
    table(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$sex),
    table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$SEX)
  ),ncol=2)
rownames(sexdata.l)<-c("Berry.train","Berry.val","B.N.AX"); colnames(sexdata.l)<-c("Female","Male")
prop.test(sexdata.l)

table(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$CrudeEthnicity)
table(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$CrudeEthnicity)
table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$ETHNICITY)

ethnicity.l<-matrix(
  rbind(
    c(7,2,5,3,0,0,0),
    c(31,0,0,0,0,0,0),
    c(5,0,0,0,0,0,0)
  ),ncol=7)
rownames(ethnicity.l)<-c("Berry.train","Berry.val","B.N.AX"); colnames(ethnicity.l)<-c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity.l,simulate.p.value=TRUE)
```

335
```r
### TB
agedata.t<-list(
  as.numeric(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$age),
  as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$age),
  as.numeric(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$AGE_CALC))

lapply(agedata.t,length)
lapply(agedata.t,summery)
kruskal.test(agedata.t)

sexdata.t<-matrix(
  rbind(
    table(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$sex),
    table(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$sex),
    table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$SEX)
  ),ncol=2)
rownames(sexdata.t)<-c("Berry.train","Berry.val","B.N.AX");colnames(sexdata.t)<-c("Female","Male")
prop.test(sexdata.t)

table(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$CrudeEthnicity)

ethnicity.t<-matrix(
  rbind(
    c(4,3,2,3,0,0,1),
    c(20,0,0,0,0,0,0),
    c(11,0,0,0,0,1,0)
  ),ncol=7)
rownames(ethnicity.t)<-c("Berry.train","Berry.val","B.N.AX");colnames(ethnicity.t)<-c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity.t,simulate.p.value=TRUE)

classdata.t<-matrix(
  rbind(
    table(factor(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$threeclass,levels=c("PTB","TB-PC"))),
    table(factor(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$threeclass,levels=c("PTB","TB-PC"))),
    table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$class)
  ),ncol=2)
rownames(classdata.t)<-c("Berry.train","Berry.val","B.N.AX");colnames(classdata.t)<-c("PTB","TB-PC")
prop.test(classdata.t)

technology<-matrix(
  rbind(
    table(factor(rep("Tempus",42),levels=c("Tempus","paxGene"))),
    table(factor(rep("Tempus",51),levels=c("Tempus","paxGene"))),
    table(factor(rep("paxGene",22),levels=c("Tempus","paxGene")))
  ),ncol=2)
```

A Code
A.5 Relation of IMPI-MA data to BERRY dataset

```r
rownames(tech)<-c("Berry.train","Berry.val","B.N.AX");
colnames(tech)<-c("Tempus","paxGene")
prop.test(tech)

#Table 1: Compare groups within each set####

# age
# train
age.train<-list(agedata.h[[1]],agedata.l[[1]],agedata.t[[1]])
lapply(age.train,length)
lapply(age.train,summary)
kruskal.test(age.train)

# val
age.val<-list(agedata.l[[2]],agedata.t[[2]])
lapply(age.val,length)
lapply(age.val,summary)
kruskal.test(age.val)

# bnax
age.bnax<-list(agedata.h[[2]],agedata.l[[3]],agedata.t[[3]])
lapply(age.bnax,length)
lapply(age.bnax,summary)
kruskal.test(age.bnax)

# sex
# train
sexdata.train<-matrix(rbind(table(pData(berry.train)[pData(berry.train)$threeclass=="Control",]$sex),
    table(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$sex),
    table(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$sex)
  ),ncol=2)
rownames(sexdata.train)<-c("Healthy","Latent","PTB");
colnames(sexdata.train)<-c("Female","Male")
prop.test(sexdata.train)

# val
sexdata.val<-matrix(rbind(
    table(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$sex),
    table(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$sex)
  )
  ,ncol=2)
rownames(sexdata.val)<-c("Latent","PTB");
colnames(sexdata.val)<-c("Female","Male")
prop.test(sexdata.val)

# bnax
sexdata.bnax<-matrix(rbind(
    table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_ healthy",]$SEX),
    table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$SEX),
    table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$SEX)
  ),ncol=2)
rownames(sexdata.bnax)<-c("Healthy","Latent","TB");
colnames(sexdata.bnax)<-c("Female","Male")
```

337
prop.test(sexdta.bnax)

# ethnicity
ethnicity.train <- matrix(
  rbind(
    c(0,12,0,0,0,0,0),
    c(7,2,5,3,0,0,0),
    c(4,3,2,3,0,0,1)
  ), ncol = 7)
rownames(ethnicity.train) <- c("Healthy","Latent","TB");
colnames(ethnicity.train) <- c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity.train, simulate.p.value = TRUE)

ethnicity.val <- matrix(
  rbind(
    # c(0,12,0,0,0,0,0),
    c(31,0,0,0,0,0,0),
    c(20,0,0,0,0,0,0)
  ), ncol = 7)
rownames(ethnicity.val) <- c("Latent","TB");
colnames(ethnicity.val) <- c("Black","White","Asian other","South Asian","Coloured","Indian","Other")

# prop.test(ethnicity.val) # fails

ethnicity.bnax <- matrix(
  rbind(
    c(5,0,0,0,0,0,0),
    c(5,0,0,0,0,0,0),
    c(11,0,0,0,0,1,0)
  ), ncol = 7)
rownames(ethnicity.bnax) <- c("Healthy","Latent","TB");
colnames(ethnicity.bnax) <- c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity.bnax, simulate.p.value = FALSE)

tabledata <- read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Documents/PhD/10_Thesis/Thesis_all/Tables/DE_1_table1.csv", sep = ",")

# show data ####

# boxplots and stripcharts
figure <- 1
setwd(dir.figures)

# Need to figure out how to do this automatically. this fails if the order of samples in the input data is changed

tbstat = list(as.factor(B.N.AX$class2), as.factor(berry.val$confirmed_diagnosis), as.factor(berry.train$confirmed_diagnosis))
levels(tbstat[[1]]) <- c("green","deepskyblue","orange","red","red")
levels(tbstat[[2]]) <- c("deepskyblue","orange")
levels(tbstat[[3]]) <- c("green","green","deepskyblue","orange")
colnames(exprs(datalist[[1]])) <- B.N.AX$sample_name
colnames(exprs(datalist[[2]])) <- berry.val$sample_name
colnames(exprs(datalist[[3]])) <- berry.train$sample_name

# the order of this depends completely on the order in the csv file - it should not impact on the actual analysis. Just need to make sure that the class assignments for DE are correct

plotnum <- 1
for (i in 1:length(datalist)){
  boxplot(datalist[[i]], main=names(datalist)[i], col=as.character(tbstat[[i]]))
  export.plot(file.prefix=paste('fig', figure, '.', plotnum, names(datalist)[i], 'boxplot'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height); plotnum<-plotnum+1
}

figure<-figure+1

# Outlier plot
plotnum<-1
for (i in 1:length(datalist)){
  plot(datalist[[i]], what="outlier", main=names(datalist)[i])
  export.plot(file.prefix=paste('fig', figure, '.', plotnum, names(datalist)[i], 'outlier_plot'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height); plotnum<-plotnum+1
}
figure<-figure+1

# MDS all genes
plotnum<-1
for (i in 1:length(datalist)){
  plotSampleRelationsAD(datalist[[i]], method="mds", color=as.character(tbstat[[i]]), plotchar=16)
  legend("bottomleft",
    list(c("Healthy", "Latent", "PTB", "TB-PC"), c("Latent", "PTB"), c("Healthy", "Latent", "PTB"))[i],
    pch=16,
    col=list(c("green", "deepskyblue", "orange", "red"), c("deepskyblue", "orange"), c("green", "deepskyblue", "orange"))[i],
    cex=.8)
  export.plot(file.prefix=paste('fig', figure, '.', plotnum, 'mds_all', names(datalist[[i]])), export.formats=export.formats.plots, height=height*0.7, width=(1 + sqrt(5))/2*height*0.7); plotnum<-plotnum+1
}
figure<-figure+1

# PCA all genes
plotnum<-1
for (i in 1:length(datalist)){
  l=colnames(exprs(datalist[[i]]))
  trueClasses <-list(as.factor(B.N.AX$class2), as.factor(berry.val$confirmed_diagnosis), as.factor(berry.train$confirmed_diagnosis))
  spca <- SamplePCA(exprs(datalist[[i]]), trueClasses[[i]])
  plot(spca, col=list(c("deepskyblue", "orange", "green", "red", "red"), c("deepskyblue", "orange"), c("green", "deepskyblue", "orange"))[i],
    main="All genes"#, cex=2, cex.axis=2, cex.main=3, cex.lab=2.5, pch=16)
  mtext(sprintf("%.2f", spca@variances[1]/sum(spca@variances)), side=1, line=3, adj=1, cex=1.5)
  mtext(sprintf("%.2f", spca@variances[2]/sum(spca@variances)), side=2, line=3, adj=1, cex=1.5)
  legend("bottomleft",
    list(c("Healthy", "Latent", "PTB", "TB-PC"), c("Latent", "PTB"), c("Healthy", "Latent", "PTB"))[i],
    pch=16,
    col=list(c("green", "deepskyblue", "orange", "red"), c("deepskyblue", "orange"), c("green", "deepskyblue", "orange"))[i],
    cex=0.8)
# mark the group centers

# points(x1[1], x1[2], col='red', cex=6,pch=9)

# points(x2[1], x2[2], col='blue', cex=6,pch=9)

export.plot(file=paste('fig',figure,'.',plotnum,'pca_all',names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1

}

figure<-figure+1
##

# #######################################.

# Baylor method

# #######################################.

preprocess####

# threshold then median scale raw data

for (i in 1:length(datalist)){

lower.thresh <- 10
data.raw <- exprs(datalist[[i]])
data.raw.thresh<-data.raw
data.raw.thresh[data.raw<lower.thresh]<-lower.thresh
genemedians <- apply(data.raw.thresh, 1, median)
data.ms <- data.raw.thresh / genemedians

exprs(datalist[[i]])<-data.ms
}

# boxplots of median scaled data

plotnum<-1

for (i in 1:length(datalist)){

boxplot(datalist[[i]],main=names(datalist)[i],col=as.character(tbstat[[i]]))

export.plot(file=paste('fig',figure,'.',',boxplot.ms'),plotnum,names(datalist)[i],
'boxplot.ms'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1

}

figure<-figure+1

# profile plots of median scaled data (not implemented)

#plotnum<-1

#for (i in 1:length(datalist)){

#par(mar=c(10,7,2,1))

#matplot(log2(t(exprs(datalist[[i]]))),type="l",ylim=c(-7,7),ylab="Expression (log2)",xaxt='n',main=paste(names(datalist)[i],'profile plot all probes'));

#abline(a=1,b=0,col="yellow");

#abline(a=-1,b=0,col="yellow");

#abline(a=-0,b=0,col="yellow");

#axis(1,at=1:dim(exprs(datalist[[i]]))[2],lab=colnames(exprs(datalist[[i]]))),las=2)

#par(parbackup)

#export.plot(file=paste('fig',figure,'.',',profile_plot_all_probes'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
# Relation of IMPI-MA data to BERRY dataset

```r
# Detection filter
detection.threshold <- 0.01
samples.threshold.fraction <- 0.1
present.probes<-list()
plotnum<-1
for (i in 1:length(datalist)){
  data.detection<-detection(datalist[[i]])
samples.threshold <- samples.threshold.fraction * ncol(data.detection)
is.present <- data.detection < detection.threshold
# is.present is Boolean matrix with present/absent status for individual
gene and chip
present.samples.per.probe <- apply(is.present, 1, sum)
# present.samples.per.probe is vector with number of chips that have present
call for
# each probe
present.probes[[i]] <- present.samples.per.probe > samples.threshold
# present.probes is Boolean vector with present/absent status for each probe
taken over all
# chips
print(paste(sum(present.probes[[i]]), " probes present in at least 10% of the
samples"))

sample.breaks <- seq(-0.5, ncol(exprs(datalist[[i]]))+0.5, by=1)

h <- hist(present.samples.per.probe, breaks=sample.breaks,
  main=paste('Number of samples with "present" call per probe\n',
    names(datalist)[[i]]),
  xlab="Number of samples with present call", ylab="Number of probes",
  col="#BBFFBB", labels=TRUE)
abline(v=round(samples.threshold)+0.5, col='#880000', lwd=2)
export.plot(file.prefix=paste('fig',figure,'.','present_call_histogram'),
  export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
}

# Fold-change filter
samples.threshold.fraction <- 0.1
fold.threshold <- 2
updown.probes<-list()
updata<-list()
downdata<-list()
selected.probes<-list()
plotnum<-1
for (i in 1:length(datalist)){
  samples.threshold <- samples.threshold.fraction * ncol(exprs(datalist[[i]]))
  is.up <- exprs(datalist[[i]] > fold.threshold;
  print(paste(sum(is.up), " up-regulated probes in",names(datalist)[[i]]));
  up.samples.per.probe <- apply(is.up, 1, sum);
```

341
up.probes <- up.samples.per.probe > samples.threshold;
print(paste(sum(up.probes), " probes upregulated in at least 10% of the samples in", names(datalist)[[i]]));
updata[[i]] <- apply(is.up, 2, sum);

is.down <- exprs(datalist[[i]]) < 1/fold.threshold;
print(paste(sum(is.down), "down-regulated probes in", names(datalist)[[i]]));
down.samples.per.probe <- apply(is.down, 1, sum);
down.probes <- down.samples.per.probe > samples.threshold;
print(paste(sum(down.probes), " probes downregulated in at least 10% of the samples in", names(datalist)[[i]]));
downdata[[i]] <- apply(is.down, 2, sum);

is.changed <- is.up | is.down;
print(paste(sum(is.changed), " changed probes in", names(datalist)[[i]]));
changed.samples.per.probe <- apply(is.changed, 1, sum);
changed.probes <- changed.samples.per.probe > samples.threshold;
print(paste(sum(changed.probes), " probes changed in at least 10% of the samples in", names(datalist)[[i]]));
vardata <- apply(is.changed, 2, sum);
updown.probes[[i]] <- up.probes | down.probes;
print(paste(sum(updown.probes[[i]]), " probes upregulated or downregulated in at least 10% of the samples in", names(datalist)[[i]]));

# final selection of probes passing both filters
selected.probes[[i]] <- present.probes[[i]] & updown.probes[[i]]
print(paste(sum(selected.probes[[i]]), " probes pass both filters in", names(datalist)[[i]]));

par(mar=c(7,5,1,1))
barplot(updata[[i]], las=2, col=as.character(tbstat[[i]]), ylab="Number of probes with FC > 2", cex.lab=.8, ylim=c(0,9000), main=names(datalist)[[i]])
export.plot(file.prefix=paste('fig', figure, '.', plotnum, names(datalist)[i], 'probes with FC > 2'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height); plotnum <- plotnum + 1
barplot(downdata[[i]], las=2, col=as.character(tbstat[[i]]), ylab="Number of probes with FC < .5", cex.lab=.8, ylim=c(0,9000), main=names(datalist)[[i]])
export.plot(file.prefix=paste('fig', figure, '.', plotnum, names(datalist)[i], 'probes with FC < .5'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height); plotnum <- plotnum + 1
par(parbackup)
}
figure <- figure + 1

#visualise selected probes####

# for (i in 1:length(datalist)){
# par(mar=c(10,7,2,1))
# matplot(log2(t(exprs(datalist[[i]])[selected.probes[[i]],])), type="l", ylim=c(-8,8), ylab="Fold Change (log2)", xaxt="n", main=paste(names(datalist)[i], 'profile plot selected probes'));
# abline(a=1, b=0, col="yellow");
# abline(a=-1, b=0, col="yellow");
# abline(a=0, b=0, col="yellow");
# }

}
A.5 Relation of IMPI-MA data to BERRY dataset

```r
#axis(1,at=1:dim(exprs(datalist[[i]]))[2],lab=colnames(exprs(datalist[[i]])),las=2)
#par(parbackup)

#export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
profile plot selected probes'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
#

#figure<-figure+1

#heatmaps
plotnum<-1
for (i in 1:length(datalist)){
  phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
  superHeatmap(x=datalist[i],y=selected.probes[[i]],phenomatrix=phenomatrix,
scale="none")
  export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
heatmap selected probes'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
}
figure<-figure+1

#Data PCA and MDS: selected probes
plotnum<-1
for (i in 1:length(datalist)){
  plotSampleRelationsAD(datalist[[i]][selected.probes[[i]]],,method="mds",
  color=as.character(tbstat[[i]]),plotchar=16)
  legend("bottomleft",
  list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("Healthy","Latent","PTB"))[i],
pch=16,
  col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","orange"),c("green","deepskyblue","orange"))[i],
cex=.8)
  export.plot(file.prefix=paste('fig',figure,'.',plotnum,"mds_selected",names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
}
figure<-figure+1

#selected probes
plotnum<-1
for (i in 1:length(datalist)){
  l=colnames(exprs(datalist[[i]][selected.probes[[i]]]))
  trueClasses <-list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_diagnosis),as.factor(berry.train$confirmed_diagnosis))
  spca <- SamplePCA(exprs(datalist[[i]][selected.probes[[i]]]),trueClasses)
  plot(spca, col=list(c("deepskyblue","orange","green","red","red"),c("deepskyblue","orange"),c("green","green","deepskyblue","orange"))[i],
  main=paste(sum(selected.probes[[i]]),"selected probes",sep=" "))#,cex=2,
cex.axis=2,cex.main=3,cex.lab=2.5,pch=16)
  mtext(sprintf("%2f",spca$variances[1]/sum(spca$variances)),side=1,line=3,
  adj=1,cex=1.5)
  mtext(sprintf("%2f",spca$variances[2]/sum(spca$variances)),side=2,line=3,
  adj=1,cex=1.5)
```

343
legend("bottomleft", 
  list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"))[i], 
  pch=16, 
  col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","orange"),c("green","deepskyblue","orange"))[i], 
  cex=.8)

# mark the group centers
#x1 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("b",rownames(exprs(datalist[[i]])))]),1,mean),ncol=1))
# points(x1[1], x1[2], col = 'red', cex=6, pch =9)
#x2 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("f",rownames(exprs(datalist[[i]])))]),1,mean),ncol=1))
# points(x2[1], x2[2], col = 'blue', cex=6, pch =9)
export.plot(file.prefix=paste('fig',figure,'.',plotnum,'pca_selected',names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,width=(1 + sqrt(5))/2*height*0.7);plotnum <- plotnum +1
}
figure <- figure +1

# statistics ####

# classes

tbstat.class = list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_diagnosis),as.factor(berry.train$confirmed_diagnosis))

## this needs to be sorted out!
# this depends on the order of samples in the RData file
# consider automating this instead of manually defining
levels(tbstat.class[[1]])<c("no TB","no TB","active TB","active TB","active TB")
levels(tbstat.class[[2]])<c("no TB","active TB")
levels(tbstat.class[[3]])<c("no TB","no TB","latent TB","active TB")
levels(tbstat.class[[3]])<c("no TB","no TB","no TB","active TB")

# levels (tbstat.class[[3]])<-c("no TB","no TB","no TB","active TB") #2
# levels (tbstat.class[[3]])<-c("no TB","no TB","no TB","active TB") #3

data.fil =list ()
for (i in 1:length(datalist)){
data.fil[[i]]<-datalist[[i]][selected.probes[[i]],]
par(mar=c(7,5,1,1))
boxplot(data.fil[[i]],las=2,col=as.character(tbstat[[i]]),main=paste(names(datalist[[i]]),"Boxplot of selected probes")(par(parbackup)
export.plot(file.prefix=paste('fig',figure,'.',i,names(datalist[[i]]),'
  boxplot selected probes'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
}
figure <- figure +1

#KW for loop
data.fil.probe.names <- list()
data.fil.probenumber <- list()
for (i in 1:length(datalist)){
data.fil.probe.names[[i]]<-rownames(exprs(data.fil[[i]]))
}
A.5 Relation of IMPI-MA data to BERRY dataset

```r
# Let’s first set alpha:
kruskal.wallis.alpha <- 0.01 # nil significant here... and limma also set to 0.05, so it’s only fair

kruskal.wallis.table <- list(data.frame(), data.frame(), data.frame())

## Run the KW test on gene
for (i in 1:length(datalist)) {
  ksdat <- as.matrix(exprs(data.fil[[i]]))
  g <- tbstat.class[[i]]
  for (j in 1:nb.probenumber[[i]]) {
    x <- as.vector(ksdat[j,])
    ks.test <- kruskal.test(x, g)
    ## Store the result in the data frame
    kruskal.wallis.table[[i]] <- rbind(kruskal.wallis.table[[i]], data.frame(id=data.fil.probe.names[[i]][j], p.value=ks.test$p.value))
  }
}

nb.tests <- list()
for (i in 1:length(datalist)) {
  nb.tests[[i]] <- nb.probenumber[[i]]
  kruskal.wallis.table[[i]]$E.value <- kruskal.wallis.table[[i]]$p.value * nb.tests[[i]]
  kruskal.wallis.table[[i]]$FWER <- pbinom(q=0, p=kruskal.wallis.table[[i]]$p.value, size=nb.tests[[i]], lower.tail=FALSE)
  kruskal.wallis.table[[i]] <<- kruskal.wallis.table[[i]]$order(kruskal.wallis.table[[i]]$p.value.decreasing=FALSE)
  kruskal.wallis.table[[i]]$q.value.factor <- nb.tests[[i]] / 1:nb.tests[[i]]
  kruskal.wallis.table[[i]]$q.value <- kruskal.wallis.table[[i]]$p.value * kruskal.wallis.table[[i]]$q.value.factor
}

head(kruskal.wallis.table[[1]])
head(kruskal.wallis.table[[2]])
head(kruskal.wallis.table[[3]])

for (i in 1:length(datalist)) {
  plot(kruskal.wallis.table[[i]]$p.value, kruskal.wallis.table[[i]]$E.value, main=paste('Multitest corrections: ', names(datalist)[i]), xlab='Nominal p-value', ylab='Multitest-corrected statistics', log='xy', col='blue', panel.first=grid(col='#BBBBBB', lty='solid'))
  lines(kruskal.wallis.table[[i]]$p.value, kruskal.wallis.table[[i]]$FWER, pch=20, col='darkgreen', type='p')
  lines(kruskal.wallis.table[[i]]$p.value, kruskal.wallis.table[[i]]$q.value,
...
A Code

```r
pch = '+' , col = 'darkred' , type = 'p'
)
abline(h = kruskal.wallis.alpha , col = 'red' , lwd = 2)
legend('topleft' , legend = c('E-value', 'p-value' , 'q-value') , col = c('blue', '
darkgreen', '
darkred'
), lwd = 2 , bg = 'white' , bty = 'o')
}

# results ####
# calculation of effect size for original nature paper
effect.size <- pwr.t2n.test(n1 = 13 , n2 = 29 , d = NULL , sig.level = 0.01 , power = 0.9 , alternative = "two.sided")$d

# calculation of significance level, using effect size calculated above with 90% power for t-test # note this does not work when doing stats using Kruskal Wallis ANOVA for 3 conditions...

# sig <- list(
# pwr.t2n.test(n1 = 12 , n2 = 10 , d = effect.size , sig.level = NULL , power = 0.9 , alternative = "two.sided")$sig.level,
# pwr.t2n.test(n1 = 20 , n2 = 31 , d = effect.size , sig.level = NULL , power = 0.9 , alternative = "two.sided")$sig.level,
# pwr.t2n.test(n1 = 13 , n2 = 29 , d = effect.size , sig.level = NULL , power = 0.9 , alternative = "two.sided")$sig.level
#)

# alt sig - empiric
sig <- list(0.08 , 0.0001 , 0.01)

diff.genes <- list()
for (i in 1: length(datalist)){
  last.significant.element <- max(which(kruskal.wallis.table[[i]]$q.value <= sig[[i]]))
  selected <- 1:last.significant.element
diff.genes.factor <- kruskal.wallis.table[[i]]$id[selected]
diff.genes[[i]] <- as.vector(diff.genes.factor)
}

# this can be compared to limma, also at 0.05 given sample size

# heatmaps ####
plotnum <- 1
for (i in 1:length(datalist)){
  phenomatrix <- matrix(cbind(as.character(tbstat[[i]]), as.character(tbstat[[i]])), ncol = 2)
superHeatmap(x = datalist[[i]] , y = diff.genes[[i]] , phenomatrix = phenomatrix , scale = "none" , addTit = paste("diff. exp.", names(datalist)[i]))
  export.plot(file.prefix = paste('fig', figure, '.' , plotnum , names(datalist)[i], ', heatmap DE probes'), export.format = export.format.plot , height = height , width = (1 + sqrt(5))/2*height); plotnum <- plotnum + 1
}
figure <- figure + 1

# Two-way overlap heatmaps not run####
```
A.5 Relation of IMPI-MA data to BERRY dataset

```r
# plotnum <- 1
# for (i in 1:length(datalist)){
# phenomatrix <- matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
# superHeatmap(x=datalist[[i]],y=intersect12,phenomatrix=phenomatrix,scale ="none",addTit=paste("overlap IMPI VAL",names(datalist[[i]])),
# heatplot DE probes: overlap IMPI VAL),export.formats=export.formats.
# plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
# }
# figure<-figure+1

# plotnum <- 1
# for (i in 1:length(datalist)){
# phenomatrix <- matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
# superHeatmap(x=datalist[[i]],y=intersect13,phenomatrix=phenomatrix,scale ="none",addTit=paste("overlap IMPI TRAIN",names(datalist[[i]])),
# heatmap DE probes: overlap IMPI TRAIN'),export.formats=export.formats.
# plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
# }
# figure<-figure+1

# # the below gives an interesting error using intersect23: intersect23[25]
do[es not exist in datalist[[i]]]; it's a nuID not found on newer chips
# # fix:
# intersect23b <- intersect(row.names(exprs(datalist[[1]])),intersect23)
# #
# # plotnum <- 1
# for (i in 1:length(datalist)){
# phenomatrix <- matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
# superHeatmap(x=datalist[[i]],y=intersect23b,phenomatrix=phenomatrix,
# scale ="none",addTit=paste("overlap VAL TRAIN",names(datalist[[i]])),
# heatmap DE probes: overlap VAL TRAIN'),export.formats=export.formats.
# plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
# }
# figure<-figure+1

# Data PCA and MDS####

plotnum<-1
for (i in 1:length(datalist)){
plotSampleRelationsAD(datalist[[i]][diff.genes[[i]]],method="mds",color =
as.character(tbstat[[i]]),plotchar=16)
legend("bottomleft",
list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c(" Healthy","Latent","PTB"))[[i]],
pch=16,
col=list(c("green","deepskyblue","orange","red"),c("deepskyblue"," orange"),c("green","deepskyblue","orange"))[[i]],
cex=.8)
export.plot(file.prefix=paste("fig","figure",',',plotnum,'mds selected',names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,
width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
```

A Code

```r
# selected genes
plotnum <- 1
for (i in 1:length(datalist)){
  l=colnames(exprs(datalist[[i]][diff.genes[[i]],]))
  trueClasses <- list(as.factor(B.N AX$class2), as.factor(berry.val$confirmed_diagnosis), as.factor(berry.train$confirmed_diagnosis))
  spca <- SamplePCA(exprs(datalist[[i]][diff.genes[[i]],]), trueClasses[[i]])
  plot(spca, col=list(c("deepskyblue", "orange", "green", "red", "red"), c("deepskyblue", "orange"), c("green", "green", "deepskyblue", "orange"))[[i]],
       main=paste(length(diff.genes[[i]])," selected probes", sep=" ")#, cex=2, cex.axis=2, cex.main=3, cex.lab=2.5, pch=16)
  mtext(sprintf("%.2f", spca@variances[1]/sum(spca@variances)), side=1, line=3, adj=1, cex=1.5)
  mtext(sprintf("%.2f", spca@variances[2]/sum(spca@variances)), side=2, line=3, adj=1, cex=1.5)
  legend("bottomleft",
         list(c("Healthy", "Latent", "PTB", "TB-PC"), c("Latent", "PTB"), c("Healthy", "Latent", "PTB"))[[i]],
         pch=16,
         col=list(c("green", "deepskyblue", "orange", "red"), c("deepskyblue", "orange"), c("green", "deepskyblue", "orange"))[[i]],
         cex=8)
  # mark the group centers
  #x1 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("b", rownames(exprs(datalist[[i]]))]])), 1, mean), ncol=1)
  #points(x1[1], x1[2], col='red', cex=6,pch=9)
  #x2 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("f", rownames(exprs(datalist[[i]]))]])), 1, mean), ncol=1)
  #points(x2[1], x2[2], col='blue', cex=6,pch=9)
  export.plot(file.prefix=paste('fig',figure,'.',plotnum,'pca_selected',
                               names(datalist[[i]])), export.formats=export.formatsplots
             ,height=height*0.7, width=(1 + sqrt(5))/2*height*0.7); plotnum <- plotnum+1
}
```
A.5 Relation of IMPI-MA data to BERRY dataset

```
dict3.bnax<-merge(dict2.bnax,dict.bnax.entrez,by.x="nuID",by.y=0)
dict3.val<-merge(dict2.val,dict.val.entrez,by.x="nuID",by.y=0)
dict3.train<-merge(dict2.train,dict.train.entrez,by.x="nuID",by.y=0)
write.csv(dict3.bnax,file=file.path(dir.results,"DE_Baylor_BNAX.csv"))
write.csv(dict3.val,file=file.path(dir.results,"DE_Baylor_VAL.csv"))
write.csv(dict3.train,file=file.path(dir.results,"DE_Baylor_train.csv"))

#overlap computations for 393

#import probe lists

load(file.path(dir.rdata,'berry.sig.RData'))
load(file.path(dir.rdata,'berry.86.RData'))
load(file.path(dir.rdata,'berry.393.RData'))

#annotate probe lists
dict<-probeID2nuID(berry.sig,species="Human")
dict.86<-IlluminaID2nuID(as.character(berry.86[,1]),species="Human")
dict.393<-IlluminaID2nuID(as.character(berry.393[,1]),species="Human")

#intersectOfOverlapWith393<-intersect(intersect13,dict[,7])
#intersectBerryWith393<-intersect(diff.genes[[2]],dict[,7])
#intersectSAWith393<-intersect(diff.genes[[1]],dict[,7])

## Venn 4-way at probe level####
x.1<-diff.genes[[1]]
x.2<-diff.genes[[2]]
x.3<-diff.genes[[3]]
x.4<-as.character(dict[,7])
x.5<-as.character(dict.86[,7])

venndata.x.hivNeg<-list(x.1,x.2,x.3,x.4)
sixtyone<-intersect(intersect(intersect(x.1,x.2),x.3),x.4)
s.o.<-nuID2IlluminaID(as.character(sixtyone),species="Human")
s.o.b.<-nuID2EntrezID(as.character(sixtyone),filterTh = NULL,lib.mapping="lumiHumanIDMapping", returnAllInfo = TRUE)
s.o.c<-merge(s.o.a,s.o.b,by.x="nuID",by.y=0)
write.csv(s.o.c,file=file.path(dir.results,"TripleOverlap61probes.csv"))

temp.venn<-venn.diagram(x = venndata.x.hivNeg,
category=c("SA","VAL","TRAIN","393"),
filename = file.path(dir.figures,"fig71_Venn_4set_lh_al_alah.tiff"),
filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set Probe.tiff",sep="")),
filename = NULL,
filename = "~/Users/armindeffur/Desktop/file.tiff",
scaled = T, ext.text = TRUE, ext.line.lwd = 2,
ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
inverted = TRUE,
```
```
cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
#main = "Overlap", sub = "MA, VAL, TRAIN", "393",
# main.cex = 2, sub.cex = 1,
fill=c("blue","green","red","gray"),
alpha=c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000

pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep ="")),height=7,width=(1 + sqrt(5))/2*7)
ggplot(temp.venn)
dev.off()
figure<-figure+1

## Venn 5-way at probe level####

dx.1<-diff.genes[[1]]
dx.2<-diff.genes[[2]]
dx.3<-diff.genes[[3]]
dx.4<-as.character(dict[,7])
dx.5<-as.character(dict.86[,7])

venndatav.hivNeg<-list(x.1,x.2,x.3,x.4,x.5)

temp.venn<-venn.diagram(
  x = venndatav.hivNeg,
  category=c("SA","VAL","TRAIN","393","86"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_aah.tiff"),
  #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.tiff",sep="")),
  filename = NULL,
  #filename = "/Users/armindeffur/Desktop/file.tiff",
  scaled = T, ext.text = TRUE, ext.line.lwd = 2,
  ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
  inverted = TRUE,
  cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
  #main = "Overlap", sub = "MA, VAL, TRAIN", "393",
  # main.cex = 2, sub.cex = 1,
  fill=c("blue","green","red","gray","orange"),
  alpha=c(.4,.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)

pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep ="")),height=7,width=(1 + sqrt(5))/2*7)
ggplot(temp.venn)
dev.off()
figure<-figure+1

## Venn 4-way at gene level####

gx.1<-unique(as.character(nuID2IlluminaID(x.1)[,2]))
gx.2<-unique(as.character(nuID2IlluminaID(x.2)[,2]))
gx.3<-unique(as.character(nuID2IlluminaID(x.3)[,2]))
gx.4<-unique(as.character(nuID2IlluminaID(x.4)[,2]))

venndatav.hivNeg.genes<-list(gx.1,gx.2,gx.3,gx.4)

temp.venn.2<-venn.diagram(
  x = venndatav.hivNeg.genes,
  category=c("SA","VAL","TRAIN","393"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_aah.tiff"),
A.5 Relation of IMPI-MA data to BERRY dataset

```r
# filename = file.path(dir.figures, paste("fig",figure,"_Venn_4set_gene.tif" ,sep="")),
filename = NULL,
scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
inverted = TRUE,
cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
# main = "Overlap",  sub = "MA, VAL, TRAIN","393",
# main.cex = 2,  sub.cex = 1,
filled = c("blue","green","red","gray"),
alpha = c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)
pdf(file=file.path(dir.figures, paste("fig",figure,"_Venn_4set_engle.pdf" ,sep="")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn.2)
dev.off()
figure <- figure +1
## Venn 4-way at gene level for top 500 probes####

g500x.1<-as.character(kruskal.wallis.table[[1]][1:500,1])
g500x.2<-as.character(kruskal.wallis.table[[2]][1:500,1])
g500x.3<-as.character(kruskal.wallis.table[[3]][1:500,1])
x.4<-as.character(dict[,7]);
venndatax.hivNeg.500<-list(x500x.1,x500x.2,x500x.3,x.4)
temp.venn.3<-venn.diagram(
x = venndatax.hivNeg.500,
category=c("SA","VAL","TRAIN","393"),
# filename = file.path(dir.figures,"fig71_Venn_3set_1h_ah_al_lh.tif") ,
# filename = file.path(dir.figures,"fig71_Venn_3set_lh_ah_lh.tif") ,
filename = NULL,
scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
exh.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
inverted = TRUE,
cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
# main = "Overlap",  sub = "MA, VAL, TRAIN","393",
# main.cex = 2,  sub.cex = 1,
alpha = c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)
pdf(file=file.path(dir.figures, paste("fig",figure,"_Venn_4set_engle.pdf" ,sep="")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn.3)
dev.off()
figure <- figure +1
## Venn 4-way at gene level for top 500 probes####
g500x.1<-unique(as.character(nuID2Illumina(xEQ.1)[,2])
g500x.2<-unique(as.character(nuID2Illumina(xEQ.2)[,2])
g500x.3<-unique(as.character(nuID2Illumina(xEQ.3)[,2])
g500x.4<-unique(as.character(nuID2Illumina(x.4)[,2])
venndatax.hivNeg.genes.500<-list(g500x.1,g500x.2,g500x.3,g500x.4)
temp.venn.4<-venn.diagram(
x = venndatax.hivNeg.genes.500,
category=c("SA","VAL","TRAIN","393"),
```

## Heatmaps: full intersect at probe and gene level ####

```r
in.common.probe <- intersect(x.1, intersect(x.2, intersect(x.3, x.4)))
in.common <- intersect(gx.1, intersect(gx.2, intersect(gx.3, gx.4)))
```

```r
plotnum <- 1
for (i in 1:length(datalist)){
    phomenatrix <- matrix(cbind(as.character(tbstat[[i]]), as.character(tbstat[[i]])), ncol=2)
    superHeatmap(x=datalist[[i]], y=in.common.probe, phomenatrix=phomenatrix, scale="none", addTit=paste("overlap", names(datalist)[i]))
    export.plot(file.prefix=paste('fig', figure, '.', names(datalist)[i], 'heatmap in common probes'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height); plotnum <- plotnum+1
}
```

## Venn: 393 and AOG_R_method_train probe (validation of implementation of Baylor method in R) ####

```r
venndatax.val.meth <- list(x.3, x.4)
```

```r
temp.venn.5 < - venn.diagram(
    x = venndatax.val.meth,
    category=c("TRAIN","393"),
    #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ab.tiff"),
    #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff","_set","sep"="")),
    filename = NULL,
    scaled = T, ext.text = TRUE, ext.line.lwd = 2,
    ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
    inverted = TRUE,
    cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
    main = "Overlap", sub = "MA, VAL, TRAIN","393",
    # main.cex = 2, sub.cex = 1,
    fill=c("blue","green", "red", "gray"),
    alpha=c(0.4, 0.4, 0.4), height=3000, width=(1 + sqrt(5))/2*3000)
```

```r
grid.draw(temp.venn.4)
dev.off()
```

```r
figure <- figure +1
```

```r
## Venn: 393 and AOG_R_method_train probe (validation of implementation of Baylor method in R) ####

```r
venndatax.val.meth <- list(x.3, x.4)
```

```r
temp.venn.5 < - venn.diagram(
    x = venndatax.val.meth,
    category=c("TRAIN","393"),
    #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ab.tiff"),
    #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff","_set","sep"="")),
    filename = NULL,
    scaled = T, ext.text = TRUE, ext.line.lwd = 2,
    ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
    inverted = TRUE,
    cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
    main = "Overlap", sub = "MA, VAL, TRAIN", "393",
    # main.cex = 2, sub.cex = 1,
    fill=c("blue", "green"),
    alpha=c(0.4, 0.4), height=3000, width=(1 + sqrt(5))/2*3000)
```

```r
grid.draw(temp.venn.4)
dev.off()
```

```r
figure <- figure +1
```

```r
```
A.5 Relation of IMPI-MA data to BERRY dataset

```r
figure <- figure + 1

## Venn : 393 and AOG_R_method_train gene (validation of implementation of Baylor method in R) #####
venndata.val.meth.g <- list(gx.3, gx.4)
temp.venn.6 <- venn.diagram(x = venndata.val.meth.g, category = c("TRAIN", "393"),
# filename = file.path(dir.figures, "fig71_Venn_3set_lh_al_ah.tiff"),
# filename = file.path(dir.figures, paste("fig", figure, "_Venn_4set_gene.tiff", sep = "")),
filename = NULL, scaled = T, ext.text = TRUE, ext.line.lwd = 2,
ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
inverted = TRUE, cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
# main = "Overlap", sub = "MA, VAL, TRAIN", "393",
# main.cex = 2, sub.cex = 1,
fill = c("blue", "green"), alpha = c(0.4, 0.4), height = 3000, width = (1 + sqrt(5)) / 2 * 3000 pdf(file = file.path(dir.figures, paste("fig", figure, "_Venn_2set_gene.pdf", sep = "")), height = 7, width = (1 + sqrt(5)) / 2 * 7)
grid.draw(temp.venn.6)
dev.off()
figure <- figure + 1

## Venn : SA and 393 probe ####
venndata.val.SA <- list(x.1, x.4)
temp.venn.7 <- venn.diagram(x = venndata.val.SA, category = c("SA", "393"),
# filename = file.path(dir.figures, "fig71_Venn_3set_lh_al_ah.tiff"),
# filename = file.path(dir.figures, paste("fig", figure, "_Venn_4set_gene.tiff", sep = "")),
filename = NULL, scaled = F, ext.text = TRUE, ext.line.lwd = 2,
ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
inverted = TRUE, cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
# main = "Overlap", sub = "MA, VAL, TRAIN", "393",
# main.cex = 2, sub.cex = 1,
fill = c("blue", "green"), alpha = c(0.4, 0.4), height = 3000, width = (1 + sqrt(5)) / 2 * 3000 pdf(file = file.path(dir.figures, paste("fig", figure, "_Venn_2set_probe.pdf", sep = "")), height = 7, width = (1 + sqrt(5)) / 2 * 7)
grid.draw(temp.venn.7)
dev.off()
figure <- figure + 1

## Venn : SA and 393 gene ####
venndata.val.SA.g <- list(gx.1, gx.4)
temp.venn.8 <- venn.diagram(x = venndata.val.SA.g, category = c("SA", "393"),
# filename = file.path(dir.figures, "fig71_Venn_3set_lh_al_ah.tiff"),
Listing A.9: RelationIMPI-MA to TB-AOG

```r
# filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff ",sep="")),
filename = NULL,
scaled = T, ext.text = TRUE, ext.line.lwd = 2,
ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
inverted = TRUE,
cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
#main = "Overlap", sub = "MA, VAL, TRAIN","393",
# main.cex = 2, sub.cex = 1,
fill=c("blue","green"),
alpha=c(.4,.4),height=3000, width=(1 + sqrt(5))/2*3000
pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_probe.pdf",sep 
="")),height=7, width=(1 + sqrt(5))/2*7)
ggrid.draw(temp.venn.8)
dev.off()
figure<-figure+1
```
A.6 Overview of all data

```r
# setup####
source("/Users/arnindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System_setup.R")
source(file.path(dir.R.files,"superHeatmap.R"))
par("xpd"=FALSE)
export.formats.plots <- c("pdf","png")
# phenodata directory
dir.pheno <- file.path(dir.data_root,'09_IMPI_MA_DATA/08_PhenoData')
print(paste("Phenodata data repository", dir.pheno))

## Individual data folders (specific for each part)
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/05_RData')

# Output folders
dir.home<dir.root
dir.main <- file.path(dir.root,'02_output/06_IMPI-MA/AllData','QC_and_Unsupervised')

# Current date string used for versioning output
ds<-date()

# versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version,"results")

## Define folder for saving figures
dir.figures <- file.path(dir.output.version,"figures")
for (dir in c(dir.main,dir.figures,dir.results)) {
    if (!file.exists(dir)) {
        dir.create(dir,recursive=T,showWarnings=T)
    }
}

sink(type="output",file=file.path(dir.results,"session_output.txt"))
sessionInfo()
print(paste("Begin time:" ,ds))
print(paste("Results will be saved to", dir.results))
print(paste("Figures will be saved to", dir.figures))

figure<-1

# data
load(file.path(dir.rdata,'IMPI_MA.RData'))
load(file.path(dir.rdata,'berry_train.RData'))
load(file.path(dir.rdata,'berry_test.RData'))
load(file.path(dir.rdata,'berry_val.RData'))

# subset allArrays by compartment
fluidArrays<-allArrays[,allArrays$Compartment=="fluid"]
bloodArrays<-allArrays[,allArrays$Compartment=="blood"]

# treemap####
```

355
datacats <- read.csv(file.path(dir.pheno, "phenodf_simple.csv"))

tmPlot(datacats, index=c("class2", "Compartment", "matching", "HIV.Status"),
       vSize="N", vColor="N", type="value", pal="Set3")

tmPlot(datacats, index=c("class2", "Compartment", "HIV.Status"), vSize="N",
       vColor="N", type="value", pal="Set3")

# QC and plots####

#QC summary
allarraysum <- summary(allArrays, "QC")
allArrays

berry.trainsum <- summary(berry.train, "QC")
berry.train

berry.testsum <- summary(berry.test, "QC")
berry.test

berry.valsum <- summary(berry.val, "QC")
berry.val

par(mfrow=c(1,1), mar=c(5, 4, 4, 2) + 0.1)
setwd(dir.figures)

array.list <- list("All IMPI Arrays" = allArrays, "IMPI blood Arrays" = bloodArrays,
                   "IMPI fluid Arrays" = fluidArrays, "Berry train" = berry.train,
                   "Berry test" = berry.test, "Berry SA validation" = berry.val)

# #PDF plots
# plotnumber<-1
# for (i in 1:length(array.list)) {
#  density(array.list[[i]], addLegend=FALSE, main=paste("Density plot", names(array.list)[i]))
#  export.plot(file.prefix=paste('fig', figure, '.', plotnumber, names(array.list)), export.format=export.format, height=height/2, width=(1 + sqrt(5))/2*height/2)
#  plotnumber<-plotnumber+1
# }
# figure<-figure+1
#
# #CDF plots
# plotnumber<-1
# for (i in 1:length(array.list)) {
#  plotCDF(array.list[[i]], addLegend=FALSE, main=paste("CDF plot", names(array.list)[i]))
#  export.plot(file.prefix=paste('fig', figure, '.', plotnumber, names(array.list)), export.format=export.format, height=height/2, width=(1 + sqrt(5))/2*height/2)
#  plotnumber<-plotnumber+1
# }
# figure<-figure+1
#
# #thesis versions:
# #PDF plots
# plotnumber<-1
# for (i in 1:length(array.list)) {

A.6 Overview of all data

density(array.list[[i]], addLegend=FALSE, main=NULL)  
export.plot(file.prefix=paste('fig', figure, '.png'), plotnumber, names(array.list[[i]]), exportFormats=export.formats.plots, height=height/1.5, width=(1 + sqrt(5))/2*height/1.5)  
plotnumber<-plotnumber+1  
}
figure<-figure+1

#CDF plots
plotnumber<-1
for (i in 1:length(array.list)) {
plotCDF(array.list[[i]], addLegend=FALSE, main=NULL)  
export.plot(file.prefix=paste('fig', figure, '.png'), exportFormats=export.formats.plots, height=height/1.5, width=(1 + sqrt(5))/2*height/1.5)  
plotnumber<-plotnumber+1  
}
figure<-figure+1

#example of a pairplot of 5 samples
pairs(allArrays[,1:5], smoothScatter=TRUE, main="Smooth scatter, arrays 1 to 5")  
export.plot(file.prefix=paste('fig', figure, 'smooth_scatter'), exportFormats=export.formats.plots, height=height/1.5, width=(1 + sqrt(5))/2*height)  
figure<-figure+1

#examples of MA plot of 5 samples
MAplot(allArrays[,1:5], smoothScatter=TRUE, main="MA plot, arrays 1 to 5")  
export.plot(file.prefix=paste('fig', figure, 'MA_1_to_5'), exportFormats=export.formats.plots, height=height/1.5, width=(1 + sqrt(5))/2*height)  
figure<-figure+1

#colour factors for labeling plots by compartment
comlist<-as.factor(allArrays$Compartment)  
levels(comlist)<-c("gray","yellow")  
comlist2<-comlist  
levels(comlist2)<-c("red","tan1")  
comlist3<-as.factor(allArrays$class2)  
levels(comlist3)<-c("deepskyblue","orange","orange","green","red","red","red")  
pchlist<-comlist  
pchlist2<-as.numeric(as.vector(pchlist))  
levels(pchlist)<-c(21,24)  
pchlist3<-as.numeric(as.vector(pchlist))  
levels(pchlist)<-c(16,1)

#boxplots of all IMPI data
boxplot(allArrays, col=as.character(comlist), subset=NULL, main="Boxplot of all samples", boxwex=.8, cex=2)  
export.plot(file.prefix=paste('fig', figure, 'boxplot_all_IMPI'), exportFormats=export.formats.plots, height=height*1.0, width=(1 + sqrt(5))/2*height*1.0)  
figure<-figure+1

#Sample relations of all IMPI_MA data####
labs<-allArrays$sample_name  
plotSampleRelation(allArrays, method="mds", color=as.character(comlist))
A Code

```r
export.plot(file.prefix=paste('fig', figure, 'mds_all.IMPI'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
figure<-figure+1

plotSampleRelation(allArrays, method="cluster", labels=labs)
export.plot(file.prefix=paste('fig', figure, 'sample hc'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
figure<-figure+1

plotSampleRelationsAD(allArrays, method="mds", color=as.character(comlist),
backgr=as.character(comlist2))
export.plot(file.prefix=paste('fig', figure, 'mds_all.IMPI_v2'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
figure<-figure+1

plotSampleRelationsAD(allArrays, subset=NULL, method="mds")
plotSampleRelationsAD(allArrays, method="mds", color=as.character(comlist3),
plotchar=pchlist3)

legend("bottomright",
c("fluid", "blood TBPC", "blood PTB", "blood LTBI", "blood healthy"),
pch=c(1,16,16,16,16),
col=c("red", "red", "orange", "deepskyblue", "green"),
cex=.8)
export.plot(file.prefix=paste('fig', figure, 'mds_all.IMPI_v3'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)

#3d PCA and scatterplot to check how blood and fluid separate in 3 dimensions
plotSampleRelations3D_AD(allArrays, method="mds", color=as.character(comlist),
backgr=as.character(comlist2))
rgl.snapshot(file.path(dir.figures,paste('fig', figure, 'mds_all.IMPI_3D')), fmt="png", top=TRUE)
figure<-figure+1

#Sample relations of blood data

#colour factors for labeling plots by category
comlist.blood.diag<-as.factor(bloodArrays$class2)
levels(comlist.blood.diag)<-c("deepskyblue", "orange", "orange", "green", "red", "red", "red")
comlist.blood.HIV<-as.factor(bloodArrays$HIV.Status)
levels(comlist.blood.HIV)<-c("green", "red")
pchlist<-comlist.blood.HIV
levels(pchlist)<-c(21,24)
pchlist2<-as.numeric(as.vector(pchlist))
levels(pchlist)<-c(0,7)
pchlist3<-as.numeric(as.vector(pchlist))

plotSampleRelationsAD(bloodArrays, method="mds", color=as.character(comlist.blood.diag), plotchar=pchlist3)
legend("topright",
c("Healthy +", "LTBI +", "PTB +", "TBPC +", "Healthy -", "LTBI -", "PTB -", "TBPC -" ),
pch=c(7,7,7,7,0,0,0,0),
col=c("green", "deepskyblue", "orange", "red"),

```
A.6 Overview of all data

```r
ncol=2, cex=.8

export.plot(file.prefix=paste('fig',figure,'mds_blood_dx_hiv'), export.
formats=export.formats.plots, height=height, width=(1 + sqrt(5)) / 2 * height)
figure<-figure+1

comlist.blood.sex<-as.factor(bloodArrays$SEX)
levels(comlist.blood.sex)<-c("red","deepskyblue")
comlist.blood.ster<-as.factor(bloodArrays$STEROIDS)
levels(comlist.blood.ster)<-c("grey","green","red")
pchlist<-comlist.blood.ster
levels(pchlist)<-c(0,6,17)
pchlist2<-as.numeric(as.vector(pchlist))

plotSampleRelationsAD(bloodArrays,method="mds",color=as.character(comlist.
blood.sex),plotchar=pchlist2)
legend("topright",
c("F?S","F-S","F+S","M?S","M-S","M+S"),
pch=c(0,6,17,0,6,17),
col=c("red","red","red","deepskyblue","deepskyblue","deepskyblue"),
ncol=2,
cex=.8
)

export.plot(file.prefix=paste('fig',figure,'mds_blood_sex_ster'), export.
formats=export.formats.plots, height=height, width=(1 + sqrt(5)) / 2 * height)
figure<-figure+1

#XP
comlist.blood.set<-bloodArrays$sample_name
pca<-grep("PCA",comlist.blood.set)
pc<-grep("PC[0-9]",comlist.blood.set)
eu<-grep("control",comlist.blood.set)
comlist.blood.set[eu]<-"blue"
comlist.blood.set[pc]<-"red"
comlist.blood.set[pca]<-"green"

plotSampleRelationsAD(bloodArrays,method="mds",color=as.character(comlist.
blood.set),plotchar=15)
legend("topright",
c("EU","SET2","SET1"),
pch=c(15,15,15),
col=c("blue","red","green"),
ncol=2,
cex=.8
)

export.plot(file.prefix=paste('fig',figure,'mds_blood_SETs'), export.formats=
export.formats.plots, height=height, width=(1 + sqrt(5)) / 2 * height)
figure<-figure+1

#XP

#Sample relations of fluid data

#colour factors for labeling plots by category
comlist.fluid.diag<-as.factor(fluidArrays$class2)
```
Listing A.10: Overview of all data

```r
levels(comlist.fluid.diag) <- c("red", "purple")
comlist.fluid.HIV <- as.factor(fluidArrays$HIV.Status)
levels(comlist.fluid.HIV) <- c("green", "red")
pchlist <- comlist.fluid.HIV
levels(pchlist) <- c(21, 24)
pchlist2 <- as.numeric(as.vector(pchlist))
levels(pchlist) <- c(0, 7)
pchlist3 <- as.numeric(as.vector(pchlist))

plotSampleRelationsAD(fluidArrays, method = "mds", color = as.character(comlist.fluid.diag), plotchar = pchlist3)
legend("bottomleft", 
c("def +", "prob +", "def -", "prob -"),
pch = c(7, 7, 0, 0),
col = c("red", "purple", "red", "purple"),
cex = .8
)

export.plot(file.prefix = paste('fig', figure, 'mds_fluid_TBdx_hiv'), export.formats = export.formats.plots, height = height, width = (1 + sqrt(5))/2*height)
figure <- figure + 1
```
A.7 Data manager for 7 questions

```r
# metadata####
# IMPI_MA microarray analysis pipeline: question-based version
# Universal Data Manager
# Armin Deffur
# File created 10-06-2013
# Version 1.0

# System_setup.R####
source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/
   IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System
   _setup.R")

# Local Setup####

## Individual data folders (specific for each part)
dir.gx <- file.path(dir.data_root, '09_IMPI_MA_DATA/04_GX_format')
print(paste("GX12-format data repository", dir.gx))
dir.gs <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/05_
   GenomeStudio_reports")
print(paste("GenomeStudio report data repository", dir.gs))
dir.bgx <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/06_Illumina
   Manifest files/HT12v4")
print(paste("Illumina manifest file repository", dir.bgx))
dir.jpg <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/07_
   imageData/JPG")
print(paste("Raw image data (.jpg) repository", dir.jpg))
dir.tif <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/07_
   imageData/TIFF")
print(paste("Raw image data (.tif) repository", dir.tif))
dir.pheno <- file.path(dir.data_root, '09_IMPI_MA_DATA/08_Phenodata')
print(paste("Phenodata data repository", dir.pheno))

# Output folders
dir.home <-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA','AllData')

#Current date string used for versioning output
ds <-Sys.time()

dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
print(paste("Results will be saved to", dir.results))

## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
print(paste("Figures will be saved to", dir.figures))

## Define folder for saving RData files
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/06_RData_Q')
print(paste("Data will be saved to", dir.rdata))

for (dir in c(dir.main, dir.figures, dir.results)) {
```

361
```r
if (!file.exists(dir)) {
  dir.create(dir, recursive=T, showWarnings=T)
}

figure<-1  

#WD###
setwd(dir.main)

#Data for Differential Expression

#Data for Deconvolution

#Data for WGCNA

#Import expression and phenotype data using lumi###

fileName <- file.path(dir.gx, "Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm__Sample__Probe_Profile.txt")
fileNameAnno <- file.path(dir.gs, "Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm_FinalReport.txt")
bgxfile <- file.path(dir.bgx, "HumanHT-12_V4_0_R1_15002873_B.bgx")

#PHENODATA ORIGINAL
#the phenodata below assigns TB status according to Pericardial culture. This misclassifies TB-PC suspects who were culture positive elsewhere as "probable" TB (wrong)
#phenodataPath <- file.path(dir.pheno,"phenoGS_edited_csv_colour_25_9_2012.csv")

#PHENODATA EDIT 1
#the phenodata (class3 variable) below defines active TB as ANY positive site. This means that some probable TBPC cases are definite TB cases. This is relevant as the blood! signature will be determined by any (??) TB site. This needs to be tested where fluid is negative but another site positive. It is unclear if this applies to TBM, though. (issues of blood-brain-barrier, etc.)
#Important: when changing this, factors of diagnostic class may change, depending which "probable" case is made "definite".
#phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_28_10_2012.csv")

#PHENODATA EDIT 2
#With this in mind, the phenodata source file has been updated to change sample 20868 control back to probableActiveTB, as the only positive culture was CSF. It seems unwise to include only one case of TBM, while there are many TB-PC and PTB.
#the phenodata file was edited: sample PCA141 blood and fluid assignments to chips have been exchanged, as earlier mds plots strongly suggested this. All results may change that depend on the correct assignment.

#IMPI-MA data
phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_17_01_2013.csv")
sampleInfoDF<-read.csv(phenodataPath, header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
allArrays<lumiR.batch(fileName, lib.mapping='lumiHumanIDMapping', sampleInfoFile=sampleInfoDF,verbose=TRUE)
```

362
A.7 Data manager for 7 questions

```r
colnames(exprs(allArrays))<-allArrays$sample_name

# Berry data
berry.sampleInfoDF.train<-read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_train.csv", header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
berry.train<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/02Feb11_TrainingSet.txt", lib.mapping='lumiHumanIDMapping', sampleInfoFile = berry.sampleInfoDF.train, verbose=TRUE)
berry.sampleInfoDF.test<-read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_test.csv", header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
berry.test<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/02Feb11_TestSet.txt", lib.mapping='lumiHumanIDMapping', sampleInfoFile = berry.sampleInfoDF.test, verbose=TRUE)
berry.sampleInfoDF.val<-read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_val.csv", header = TRUE, colClasses = "character", comment.char = "", check.names = FALSE)
berry.val<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/02Feb11_ValidationSet.txt", lib.mapping='lumiHumanIDMapping', sampleInfoFile = berry.sampleInfoDF.val, verbose=TRUE)

# subset allArrays by compartment
fluidArrays<-allArrays[,allArrays$Compartment == "fluid"]
bloodArrays<-allArrays[,allArrays$Compartment == "blood"]

# Data subsetting####
# CSC scheme: compartment, subset, contrast
save(berry.train, file=file.path(dir.rdata,'berry_train.RData'))
save(berry.test, file=file.path(dir.rdata,'berry_test.RData'))
save(berry.val, file=file.path(dir.rdata,'berry_val.RData'))

save(allArrays, file=file.path(dir.rdata,'IMPI_MA.RData'))

# Question 1: Tuberculosis
hivNegBloodArrays<-bloodArrays[,bloodArrays$HIV.Status == "negative"] # tb
B.Neg.AN<-hivNegBloodArrays[,hivNegBloodArrays$class3 == "activeTB"]
hivNegBloodArrays$class3="notActiveTB" # training
B.Neg.AN<-hivNegBloodArrays[,hivNegBloodArrays$class3 == "probableActiveTB"]
hivNegBloodArrays$class3="notActiveTB" # validation
save(B.Neg.AN, file=file.path(dir.rdata,'blood_hivNeg_aTB_noTB.RData'))
save(B.Neg.AN, file=file.path(dir.rdata,'blood_hivNeg_proTB_noTB.RData'))
hivPosBloodArrays<-bloodArrays[,bloodArrays$HIV.Status == "positive"]
B.Pos.AN<-hivPosBloodArrays[,hivPosBloodArrays$class3 == "activeTB"]
hivPosBloodArrays$class3="notActiveTB"
B.Pos.AN<-hivPosBloodArrays[,hivPosBloodArrays$class3 == "probableActiveTB"]
hivPosBloodArrays$class3="notActiveTB"
save(B.Pos.AN, file=file.path(dir.rdata,'blood_hivPos_aTB_noTB.RData'))
```

363
# Question 2: HIV in active TB

```r
B.A.PosNeg <- bloodArrays[, bloodArrays$class3 == "activeTB"]
save(B.A.PosNeg, file = file.path(dir.rdata, 'blood_aTB_HIVPosNeg.RData'))
```

```r
PF.A.PosNeg <- fluidArrays[, fluidArrays$class3 == "activeTB"]
save(PF.A.PosNeg, file = file.path(dir.rdata, 'fluid_aTB_HIVPosNeg.RData'))
```

# Question 3: Compartment

```r
C.BPF <- allArrays[, allArrays$matching == "m"]
save(C.BPF, file = file.path(dir.rdata, 'TBPC_matchedBloodFluid.RData'))
```

```r
C.Neg.matchedBF <- C.BPF[, C.BPF$HIV.Status == "negative"]
save(C.Neg.matchedBF, file = file.path(dir.rdata, 'TBPC_hivNeg_matchedBloodFluid.RData'))
```

```r
C.Pos.matchedBF <- C.BPF[, C.BPF$HIV.Status == "positive"]
save(C.Pos.matchedBF, file = file.path(dir.rdata, 'TBPC_hivPos_matchedBloodFluid.RData'))
```

```r
C.Pos.EFF.matchedBF <- C.Pos.matchedBF[, C.Pos.matchedBF$ECP == "Eff"]
save(C.Pos.EFF.matchedBF, file = file.path(dir.rdata, 'TBPC_hivPos_EFF_matchedBloodFluid.RData'))
```

```r
C.Pos.EC.matchedBF <- C.Pos.matchedBF[, C.Pos.matchedBF$ECP == "EC"]
save(C.Pos.EC.matchedBF, file = file.path(dir.rdata, 'TBPC_hivPos_EC_matchedBloodFluid.RData'))
```

# Question 4: LTBI

```r
B.Neg.HL <- hivNegBloodArrays[, hivNegBloodArrays$class3 == "notActiveTB"]
save(B.Neg.HL, file = file.path(dir.rdata, 'blood_hivNeg_H_L.RData'))
```

```r
B.Pos.HL <- hivPosBloodArrays[, hivPosBloodArrays$class3 == "notActiveTB"]
save(B.Pos.HL, file = file.path(dir.rdata, 'blood_hivPos_H_L.RData'))
```

# Question 5: HIV in not active TB

```r
B.N.PosNeg <- bloodArrays[, bloodArrays$class3 == "notActiveTB"]
save(B.N.PosNeg, file = file.path(dir.rdata, 'blood_notActiveTB_PosNeg.RData'))
```

```r
B.H.PosNeg <- bloodArrays[, bloodArrays$class2 == "CON_healthy"]
save(B.H.PosNeg, file = file.path(dir.rdata, 'blood_healthy_PosNeg.RData'))
```

```r
B.L.PosNeg <- bloodArrays[, bloodArrays$class2 == "CON_LTBI"]
save(B.L.PosNeg, file = file.path(dir.rdata, 'blood_LTBI_PosNeg.RData'))
```

# Question 6: TB site

```r
B.Pos.PTB_TBPC <- hivPosBloodArrays[, hivPosBloodArrays$class3 == "activeTB"]
save(B.Pos.PTB_TBPC, file = file.path(dir.rdata, 'blood_hivPos_PTB_TBPC.RData'))
```

```r
B.Pos.PTBPB <- hivPosBloodArrays[, hivPosBloodArrays$class3 == "activeTB"]
save(B.Pos.PTBPB, file = file.path(dir.rdata, 'blood_hivPos_PTB_TBPC.RData'))
```

# Question 7: Haemodynamic phenotype

```r
B.C.Pos.EffEC <- hivPosBloodArrays[, hivPosBloodArrays$ECP == "Eff"]
save(B.C.Pos.EffEC, file = file.path(dir.rdata, 'blood_TBPC_hivPos_EffEC.RData'))
```

```r
PF.C.Pos.EffEC <- fluidArrays[, fluidArrays$ECP == "Eff"]
save(PF.C.Pos.EffEC, file = file.path(dir.rdata, 'fluid_TBPC_hivPos_EffEC.RData'))
```
A.7 Data manager for 7 questions

```r
# Signatures
# Berry signatures
# file = file.path(dir.gx, "Nature_paper_transcripts", "P22_42_04Mar09_Training_393_GX11.txt")
# np393 <- read.table(file, header = TRUE, sep = "\t")
# char393 <- as.character(np393[,1])
# berry.sig <- char393
# save(berry.sig, file = file.path(dir.rdata, 'berry.sig.RData'))

# need to get 86 transcript list; downloaded as xls

# import
# file86 = file.path(dir.gx, "Nature_paper_transcripts", "nature09247-s4.csv")
# np86 <- read.csv(file86, header = TRUE, sep = "",
# berry.86 <- np86
# save(berry.86, file = file.path(dir.rdata, 'berry.86.RData'))

# import manual berry393

# not done, need to fix format...
# file393 = file.path(dir.gx, "Nature_paper_transcripts", "nature09247-s2.csv")
# np393 <- read.csv(file393, header = TRUE, sep = "",
# berry.393 <- np393
# save(berry.393, file = file.path(dir.rdata, 'berry.393.RData'))
```

Listing A.11: Data-manager-Q.R
A.8 Analysis: IMPI-MA

```r
## TO DO
# 1. table 1 code

# setup####
# Current date string used for versioning output
source("/Users/armindefur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System_setup.R")
source(file.path(dir.R.files,"superHeatmap.R"))
source(file.path(dir.R.files,"superHeatmap2.R"))
par("xpd"=FALSE)
## Individual data folders (specific for each part)
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/06_RData_Q')

#Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root,'02_output/06_IMPI-MA/DE_DC_CN','DE_AD_TB')
ds<-Sys.time()
# versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version,"results")

## Define folder for saving figures
dir.figures <- file.path(dir.output.version,"figures")

for (dir in c(dir.main,dir.figures,dir.results)) {
  if (!file.exists(dir)) {
    dir.create(dir,recursive=T,showWarnings=T)
  }
}

#sink(type="output",file=file.path(dir.results,"session_output.txt"))
sessionInfo()
print(paste("Begin time:",ds))
print(paste("Results will be saved to",dir.results))
print(paste("Figures will be saved to",dir.figures))

#all iterators and their definitions
#q.i: loop over questions (n=7)
#filename: loop over filenames when loading RDdata files
#dataindex: loop over datasets when making datalist, colourmaps

#Make questions
#Questions constructs####

questions=list(
  "Tuberculosis"=list(
    "datalist_filenames"=list("blood_hivNeg_aTB_noTB.RData","blood_hivPos_aTB_noTB.RData"),
    "dataset_variables"=list("B.Neg.AN","B.Pos.AN"),
    "dataset_names"=list("Blood HIVneg activeTB-notActiveTB","Blood HIVpos activeTB-notActiveTB"),
    "contrast_variable"="class3",
    "dataset_names"=list("Blood HIVneg activeTB-notActiveTB","Blood HIVpos activeTB-notActiveTB"),
    "contrast_variable"="class3",
```

A.8 Analysis: IMPI-MA

```
"colour_variable"="class",
"colourmap"=list("deepskyblue"="CON_LTBI","orange"="CON_PTB","green"="CON_healthy","red"="TBPC"),
"pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
"wgcna_pheno"=c(),#column numbers in phenodata specific for the question; may be empty
"wgcna_contrast"=list()
),
"TBSite"=list(
  "dlist_filenames"=list("blood_hivPos_PTB_TBPC.RData","blood_hivNeg_PTB_TBPC.RData"),
  "dataset_variables"=list("B.Pos.PTB_TBPC","B.Neg.PTB_TBPC"),
  "dataset_names"=list("Blood HIVpos PTB-TBPC","Blood HIVneg PTB-TBPC"),
  "contrast_variable"="class",
  "colour_variable"="class",
  "colourmap"=list("orange"="CON_PTB","red"="TBPC"),
  "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
  "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
),
"LTBI"=list(
  "dlist_filenames"=list("blood_hivPos_H_L.RData","blood_hivNeg_H_L.RData"),
  "dataset_variables"=list("B.Pos.HL","B.Neg.HL"),
  "dataset_names"=list("Blood HIVpos Healthy-LTBI","Blood HIVneg Healthy-LTBI"),
  "contrast_variable"="class",
  "colour_variable"="class",
  "colourmap"=list("green"="CON_healthy","deepskyblue"="CON_LTBI"),
  "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
  "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
),
"hdPhenotype"=list(
  "dlist_filenames"=list("blood_TBPC_hivPos_EffEC.RData","fluid_TBPC_hivPos_EffEC.RData"),
  "dataset_variables"=list("B.C. Pos.EffEC","PF.C. Pos.EffEC"),
  "dataset_names"=list("Blood TBPC HIVPos Eff-EC","Fluid TBPC HIVPos Eff-EC"),
  "contrast_variable"="ECP",
  "colour_variable"="ECP",
  "colourmap"=list("blue"="Eff","yellow"="EC"),
  "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
  "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
),
"hiv_noTB"=list(
  "dlist_filenames"=list("blood_notActiveTB_PosNeg.RData","blood_LTBI_PosNeg.RData","blood_healthy_PosNeg.RData"),
  "dataset_variables"=list("B.N.PosNeg","B.L.PosNeg","B.H.PosNeg"),
  "dataset_names"=list("Blood not activeTB HIVpos-HIVneg","Blood LTBI HIVpos-HIVneg","Blood Healthy HIVpos-HIVneg"),
  "contrast_variable"="HIV.Status",
  "colour_variable"="HIV.Status",
  "colourmap"=list("green"="negative","red"="positive"),
  "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
  "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
```
A Code

```
"hiv_TB"=list(
  "datalist_filenames"=list("blood_aTB_HIVPosNeg.RData","fluid_aTB_HIVPosNeg.RData","blood_defPTB_HIVPosNeg.RData"),
  "dataset_variables"=list("B.A.PosNeg","PF.A.PosNeg","B.TB.PosNeg"),
  "dataset_names"=list("Blood activeTB HIVpos-HIVneg","Blood PTB HIVpos-HIVneg","Blood TB-PC HIVpos-HIVneg"),
  "contrast_variable"="HIV.Status",
  "colour_variable"="HIV.Status",
  "colourmap"=list("green"="negative","red"="positive"),
  "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
  "wgcna_pho"=c()#column numbers in phenodata specific for the question; may be empty
),
"Compartment"=list(
  "datalist_filenames"=list("TBPC_matchedBloodFluid.RData","TBPC_hivNeg_matchedBloodFluid.RData","TBPC_hivPos_matchedBloodFluid.RData","TBPC_hivPos_EFF_matchedBloodFluid.RData","TBPC_hivPos_EC_matchedBloodFluid.RData"),
  "dataset_variables"=list("C.BPF","C.Neg.matchedBF","C.Pos.matchedBF","C.Pos.EF.matchedBF","C.Pos.EC.matchedBF"),
  "dataset_names"=list("TB-PC all blood-fluid","TB-PC HIVneg blood-fluid","TB-PC HIVpos blood-fluid","TB-PC HIVpos EFF blood-fluid","TB-PC HIVpos EC blood-fluid"),
  "contrast_variable"="Compartment",
  "colour_variable"="Compartment",
  "colourmap"=list("red"="blood","yellow"="fluid"),
  "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
  "wgcna_pho"=c()#column numbers in phenodata specific for the question; may be empty
)
)

# for loops####
#q.i is question iterator; length = 7
#debug
#q.i<-1
#real
for (q.i in 1:length(questions)){
  #Question (c.q: "current question")
c.q=questions[[q.i]]
  #definitions for the current question####
datalist_filenames=c.q$datalist_filenames
dataset.variables=c.q$dataset_variables
dataset.names=c.q$dataset_names
contrast.variable=c.q$contrast_variable
colour.variable=c.q$colour_variable
colourmap=c.q$colourmap
pqlist=c.q$pqlist
extra.traits<-c.q$wgcna_pho
```

print(paste("Results for Question ",q.i," : ",names(questions)[q.i],sep=""))

# load data and make datalist####
for (filename in datalist.filenames){
  load(file.path(dir.rdata,filename))
}
datalist<-list()
for (dataindex in 1:length(datalist.filenames)){
datalist[[dataindex]]<-eval(parse(text=dataset.variables[[dataindex]]))
names(datalist)[[dataindex]]<-dataset.names[[dataindex]]
}

# make pqlist####
pqlist<-c.q$pqlist
print(paste(c("Data to be analysed:",names(datalist))))

# generate table 1####
print("Table 1 data generation")
# Table 1: Compare groups across datasets
# contrasts
contr<-lapply(dataset.variables,function(any){
as.factor(eval(parse(text=paste(any,"$",contrast.variable,sep=""))))
})

# AGE####
# AGE across all
ageAA<-lapply(datalist,function(part){as.numeric(pData(part)$AGE_CALC)})
n.age<-lapply(ageAA,length)
sum.age<-lapply(ageAA,summary)
if(class(test.age)=="try-error"){
test.age$p.value=NA
test.age$method="NA"
}

# AGE across (contrast1)
ageAC1<-lapply(datalist,function(part){
as.numeric(pData(part)[
    eval(parse(text=paste("pData(part)$",contrast.variable,sep=""))
])
  ==levels(
as.factor(eval(parse(text=paste(dataset.variables,"$",contrast.variable,sep=""))))[[1]],]
$AGE_CALC)
}

n.age.C1<-lapply(ageAC1,length)
sum.age.C1<-lapply(ageAC1,summary)
test.age.C1<-try(kruskal.test(ageAC1))
if(class(test.age.C1)=="try-error"){
test.age.C1$p.value=NA
test.age.C1$method="NA"}
### AGE across (contrast 1)

```r
ageAC2 <- lapply(datalist, function(part) {
  as.numeric(pData(part))
  [eval(parse(text=paste("pData(part)\$", contrast.variable, sep=""))
   == levels(as.factor(eval(parse(text=paste(dataset.variables,"\$", contrast.variable, sep="")))) [[2]],]
   $AGE_CALC)
}
```

```r
n.age.C2 <- lapply(ageAC2, length)
sum.age.C2 <- lapply(ageAC2, summary)
test.age.C2 <- try(kruskal.test(ageAC2))
```

```r
if(class(test.age.C2) == "try-error") {
  test.age.C2$p.value = NA
  test.age.C2$method = "NA"
}
```

### AGE down (all sets)

```r
age.down.data <- list()
age.down.n <- list()
age.down.sum <- list()
age.down.test <- list()
for(set in 1:length(datalist)) {
  age.down.data[[set]] <- list(ageAC1[[set]], ageAC2[[set]])
  age.down.n[[set]] <- lapply(age.down.data[[set]], length)
  age.down.sum[[set]] <- lapply(age.down.data[[set]], summary)
  age.down.test[[set]] <- try(kruskal.test(age.down.data[[set]]))
  if(class(age.down.test[[set]]) == "try-error") {
    age.down.test[[set]]$p.value = NA
    age.down.test[[set]]$method = "NA"
  }
}
```

### SEX all across

```r
sexAA <- lapply(datalist, function(part) {as.factor(pData(part)\$SEX)})
for(sexindex in 1:length(sexAA)) {
  levels(sexAA[[sexindex]]) <- c("Female", "Male")
}
```

```r
sexAA.data <- t(matrix(as.numeric(unlist(lapply(sexAA, function(z) {rbind(as.vector(table(z)))}))), ncol=length(sexAA), dimnames=list(c("Female", "Male"), unlist(dataset.names))))
sexAA.test <- try(if(length(sexAA) == 2) {prop.test(sexAA.data)} else {fisher.test(sexAA.data)})
```

```r
if(class(sexA.test) == "try-error") {
  sexAA.test$p.value = NA
  sexAA.test$method = "NA"
}
```

### SEX across (contrast 1)

```r
sexAC1 <- lapply(datalist, function(part) {
  as.factor(pData(part))
  [eval(parse(text=paste("pData(part)\$", contrast.variable, sep=""))
})
```

```r
ageAC2 <- lapply(datalist, function(part) {
  as.numeric(pData(part))
  [eval(parse(text=paste("pData(part)\$", contrast.variable, sep="")))
  == levels(as.factor(eval(parse(text=paste(dataset.variables,"\$", contrast.variable, sep="")))) [[2]],]
  $AGE_CALC)
```

```r
test.age.C2 <- try(kruskal.test(ageAC2))
```

```r
if(class(test.age.C2) == "try-error") {
  test.age.C2$p.value = NA
  test.age.C2$method = "NA"
}
```

### AGE down (all sets)

```r
age.down.data <- list()
age.down.n <- list()
age.down.sum <- list()
age.down.test <- list()
for(set in 1:length(datalist)) {
  age.down.data[[set]] <- list(ageAC1[[set]], ageAC2[[set]])
  age.down.n[[set]] <- lapply(age.down.data[[set]], length)
  age.down.sum[[set]] <- lapply(age.down.data[[set]], summary)
  age.down.test[[set]] <- try(kruskal.test(age.down.data[[set]]))
  if(class(age.down.test[[set]]) == "try-error") {
    age.down.test[[set]]$p.value = NA
    age.down.test[[set]]$method = "NA"
  }
}
```

### SEX all across

```r
sexAA <- lapply(datalist, function(part) {as.factor(pData(part)\$SEX)})
for(sexindex in 1:length(sexAA)) {
  levels(sexAA[[sexindex]]) <- c("Female", "Male")
}
```

```r
sexAA.data <- t(matrix(as.numeric(unlist(lapply(sexAA, function(z) {rbind(as.vector(table(z)))}))), ncol=length(sexAA), dimnames=list(c("Female", "Male"), unlist(dataset.names))))
sexAA.test <- try(if(length(sexAA) == 2) {prop.test(sexAA.data)} else {fisher.test(sexAA.data)})
```

### SEX across (contrast 1)

```r
sexAC1 <- lapply(datalist, function(part) {
  as.factor(pData(part))
  [eval(parse(text=paste("pData(part)\$", contrast.variable, sep=""))
})
```

```r
test.age.C2 <- try(kruskal.test(ageAC2))
```

```r
if(class(test.age.C2) == "try-error") {
  test.age.C2$p.value = NA
  test.age.C2$method = "NA"
}
```
== levels (as.factor(eval(parse(text=paste(dataset.variables,"$",contrast.variable,sep=""))))[[1]],]
$SEX)
)
)

for (sexindex in 1:length(sexAC1)){levels(sexAC1[[sexindex]])<-c("Female","Male")
sexAC1data<-t(matrix(as.numeric(unlist(lapply(sexAC1,function(z){rbind(as.vector(table(z))))})),ncol=length(sexAC1),dimnames=list(c("Female","Male"),unlist(dataset.names))))
sexAC1.test<-try(if (length(sexAC1)==2){prop.test(sexAC1data)}else{fisher.test(sexAC1data)})

if(class(sexAC1.test)="try-error"){
  sexAC1.test$p.value=NA
  sexAC1.test$method="NA"}

### Sex across (contrast2)
sexAC2<-lapply(datalist,function(part){
as.factor(pData(part)[
  eval(parse(text=paste("pData(part) ",contrast.variable,sep=""))]
  ==levels(
    as.factor(eval(parse(text=paste(dataset.variables,"$",contrast.variable,sep=""))))[[2]],]
$SEX)
)}

for (sexindex in 1:length(sexAC2)){levels(sexAC2[[sexindex]])<-c("Female","Male")
sexAC2data<-t(matrix(as.numeric(unlist(lapply(sexAC2,function(z){rbind(as.vector(table(z))))})),ncol=length(sexAC2),dimnames=list(c("Female","Male"),unlist(dataset.names))))
sexAC2.test<-try(if (length(sexAC2)==2){prop.test(sexAC2data)}else{fisher.test(sexAC2data)})

if(class(sexAC2.test)="try-error"){
  sexAC2.test$p.value=NA
  sexAC2.test$method="NA"}

### Sex down
sex.down.data<-list()
sex.down.test<-list()
for (set in 1:length(datalist)){
  sex.down.data[[set]]<-matrix(rbind(sexAC1data[[set]],sexAC2data[[set]]),ncol =2,dimnames=list(levels(contr[[1]]),c("Female","Male")))
  sex.down.test[[set]]<-try(prop.test(sex.down.data[[set]]))
  if(class(sex.down.test[[set]])="try-error"){
    sex.down.test[[set]]$p.value=NA
    sex.down.test[[set]]$method="NA"
  }
}

# CD4
```r
# CD4 across all
CD4AA <- lapply(datalist, function(part) {as.numeric(pData(part)$CD4)})
n.CD4 <- lapply(CD4AA, length)
sum.CD4 <- lapply(CD4AA, summary)
test.CD4 <- try(kruskal.test(CD4AA))

if (class(test.CD4) == "try-error") {
  test.CD4$p.value = NA
  test.CD4$method = "NA"
}

# CD4 across (contrast1)
CD4AC1 <- lapply(datalist, function(part) {
  as.numeric(pData(part)[
    eval(parse(text = paste("pData(part)$", contrast.variable, sep=""))]
  ]$CD4)
})
n.CD4.C1 <- lapply(CD4AC1, length)
sum.CD4.C1 <- lapply(CD4AC1, summary)
test.CD4.C1 <- try(kruskal.test(CD4AC1))

if (class(test.CD4.C1) == "try-error") {
  test.CD4.C1$p.value = NA
  test.CD4.C1$method = "NA"
}

# CD4 across (contrast2)
CD4AC2 <- lapply(datalist, function(part) {
  as.numeric(pData(part)[
    eval(parse(text = paste("pData(part)$", contrast.variable, sep=""))]
  ]$CD4)
})
n.CD4.C2 <- lapply(CD4AC2, length)
sum.CD4.C2 <- lapply(CD4AC2, summary)
test.CD4.C2 <- try(kruskal.test(CD4AC2))

if (class(test.CD4.C2) == "try-error") {
  test.CD4.C2$p.value = NA
  test.CD4.C2$method = "NA"
}

# CD4 down (all sets)
CD4.down.data <- list()
CD4.down.n <- list()
CD4.down.sum <- list()
CD4.down.test <- list()
for (set in 1:length(datalist)) {
  CD4.down.data[[set]] <- lapply(CD4AC1[[set]], CD4AC2[[set]])
  CD4.down.n[[set]] <- lapply(CD4.down.data[[set]], length)
}
```
CD4.down.sum[[set]]<-lapply(CD4.down.data[[set]],summary)
CD4.down.test[[set]]<-try(kruskal.test(CD4.down.data[[set]]))
if(class(CD4.down.test[[set]])=="try-error"){
  CD4.down.test[[set]]$p.value=NA
  CD4.down.test[[set]]$method="NA"
}

# print and export table 1 for each question####

names.across<-c(unlist(lapply(dataset.variables,function(x){c(paste(x," Med"),paste(x,"LQ"),paste(x,"UQ"))}))," P val"," Test")
age.line0<-c(rep(c(" Median","LQ","UQ"),length(dataset.names)),"Pval"," Test")
age.line1<-c(unlist(lapply(n.age,function(x){c(x,"",""))}),"",""
age.line2<-unlist(list(lapply(sum.age,function(x){x[c(3,2,5)]}),list(sprintf("%.3f",test.age$p.value)),list(substr(test.age$method,1,20))))
age.line3<-unlist(list(lapply(sum.age.C1,function(x){x[c(3,2,5)]}),list(sprintf("%.3f",test.age.C1$p.value)),list(substr(test.age.C1$method,1,20))))
age.line4<-unlist(list(lapply(sum.age.C2,function(x){x[c(3,2,5)]}),list(sprintf("%.3f",test.age.C2$p.value)),list(substr(test.age.C2$method,1,20))))
age.line5<-unlist(list(unlist(lapply(age.down.test,function(x){c(sprintf("%.3f",x$p.value),"",""))}),list("",""),recursive=T)
age.line6<-unlist(list(unlist(lapply(age.down.test,function(x){c(substr(x$method,1,20),"",""))}),list("",""),recursive=T)

sex.line0<-c(rep(c("prop F"," Female"," Male"),length(dataset.names)),"Pval"," Test")

sex.line1<-c(as.character(apply(sexAAdata,1,function(x){
  c(
    sprintf("%.3f",x[1]/x[2]),
    x[1],
    x[2]
  )
)},sprintf("%.3f",sexAA.test$p.value),
    substr(sexAA.test$method,1,20))
sex.line2<-c(as.character(apply(sexAC1data,1,function(x){
  c(
    sprintf("%.3f",x[1]/x[2]),
    x[1],
    x[2]
  )
)},sprintf("%.3f",sexAC1.test$p.value),
    substr(sexAC1.test$method,1,20))

sex.line3<-c(as.character(apply(sexAC2data,1,function(x){
  c(
    sprintf("%.3f",x[1]/x[2]),
    x[1],
    x[2]
  )
}),sprintf("%.3f",sexAC2.test$p.value),
    substr(sexAC2.test$method,1,20))

sex.line4<-c(unlist(lapply(sex.down.test,function(x){c(sprintf("%.3f",x$p.value),"",""))}),list("",""),recursive=T)
sex.line5<-unlist(list(unlist(lapply(sex.down.test,function(x){c(substr(x$method,1,20),"",""))}),list("",""),recursive=T)

CD4.line0<-c(rep(c("Median","LQ","UQ"),length(dataset.names)),"Pval"," Test")
CD4.line1<-c(unlist(lapply(n.CD4,function(x){c(x,"",""))}),"",""
A Code

```r
c4d.line2<-unlist(list(lapply(sum.CD4,function(x){x[c(3,2,5)]}),list(sprintf("%.3f",test.CD4$p.value)),list(substr(test.CD4$method,1,20))))
c4d.line3<-unlist(list(lapply(sum.CD4.C1,function(x){x[c(3,2,5)]}),list(sprintf("%.3f",test.CD4.C1$p.value)),list(substr(test.CD4.C1$method,1,20))))
c4d.line4<-unlist(list(lapply(sum.CD4.C2,function(x){x[c(3,2,5)]}),list(sprintf("%.3f",test.CD4.C2$p.value)),list(substr(test.CD4.C2$method,1,20))))
c4d.line5<-unlist(list(unlist(lapply(CD4.down.test,function(x){c(sprintf("%.3f",x$p.value),"","",""))})),list("",""),recursive=T))
c4d.line6<-unlist(list(unlist(lapply(CD4.down.test,function(x){c(substr(x$method,1,20),"",""))})),list("",""),recursive=T))

dft<-data.frame(rbind(age.line0,age.line2,age.line3,age.line4,age.line5,age.line6,sex.line0,sex.line1,sex.line2,sex.line3,sex.line4,sex.line5,CD4.line0,CD4.line2,CD4.line3,CD4.line4,CD4.line5,CD4.line6))
names.down<-c("Age","age_All",paste("age_",levels(contr[[1]])),"age_P value","age_Test","Sex","sex_All",paste("sex_",levels(contr[[1]])),"sex_P value","sex_Test","CD4","CD4_All",paste("CD4_",levels(contr[[1]])),"CD4_P value","CD4_Test")
row.names(dft)<-c(names.down)
names(dft)<-unlist(c(lapply(dataset.names,function(x){c(x,"",""))}),'Pval","Test'))

#hardcoded!
#xt<-xtable(dft,align=c("p{1cm}|","p{1cm}|","p{0.8cm}|","p{0.8cm}|","p{1cm}|","p{0.8cm}|","p{0.8cm}|","p{1cm}|"))
xt<-xtable(dft,align=c("|",rep(c("p{1cm}|"),nrow(dft)),"p{1cm}|"))
print(xt,hline.after=-1:nrow(dft),include.colnames=T,size=c("tiny"),file=  
  file.path(dir.results,paste("Table 1 for question ",q.i," ",names( 
  questions)[[q.i]],".tex",sep=""))

#temporary q.i closure
#
#start analysis####
#analysis for loop
for (i in 1:length(datalist)){
  print(paste("current dataset: ",names(datalist)[[i]]))
  #DE analysis####
  analysis="DE"
  an.count<-1
  print("Analysis 1: Differential expression")
```
```r
# Define empty lists required for DE analysis

# Make class levels and colours

# Show raw data

# Boxplots and stripcharts

# Improve this by integrating into for loop
```

A.8 Analysis: IMPI-MA
```r
# for (i in 1:length(datalist)) {
  boxplot(datalist[[i]], main = names(datalist)[i], col = as.character(classcolours[[i]]))
  export.plot(file.prefix = paste('fig', q.i, '.', an.count, '.', i, '.', 'boxplot'),
              names(questions)[[q.i]], '.', analysis, names(datalist)[i], 'boxplot'),
  export.formats = export.formats.plots, height = height, width = (1 + sqrt(5))/2*height)
#}
figure <- figure + 1

# Outlier plot
# for (i in 1:length(datalist)) {
plot(datalist[[i]], what = "outlier", main = names(datalist)[i])
export.plot(file.prefix = paste('fig', q.i, '.', an.count, '.', i, '.', 'outlier_plot'),
names(questions)[[q.i]], '.', analysis, names(datalist)[i], "outlier_plot"),
export.formats = export.formats.plots, height = height, width = (1 + sqrt(5))/2*height)
#}
figure <- figure + 1

# MDS all probes
# for (i in 1:length(datalist)) {
plotSampleRelationsAD(datalist[[i]], method = "mds", color = as.character(classcolours[[i]])), plotchar = 16)
legend("bottomleft",
  levels(tbstat[[i]]),
  pch = 16,
  col = levels(classcolours[[i]]),
  cex = .8)
export.plot(file.prefix = paste('fig', q.i, '.', an.count, '.', i, '.', " MDS_all"),
names(questions)[[q.i]], '.', analysis, names(datalist)[i], "MDS_all"),
export.formats = export.formats.plots, height = height, width = (1 + sqrt(5))/2*height)
#}
figure <- figure + 1

# PCA all probes
# for (i in 1:length(datalist)) {
  l = colnames(exprs(datalist[[i]]))
  #trueClasses <- tbstat
  spca <- SamplePCA(exprs(datalist[[i]]), tbstat[[i]])
  plot(spca, col = levels(classcolours[[i]]), main = paste(nrow(exprs(datalist[[i]]))), "probes")#
  mtext(sprintf("%.2f", spca@variances[1]/sum(spca@variances)), side = 1, line = 3, adj = 1, cex = 1.5)
  mtext(sprintf("%.2f", spca@variances[2]/sum(spca@variances)), side = 2, line = 3, adj = 1, cex = 1.5)
  legend("bottomleft",
    levels(tbstat[[i]]),
    pch = 16,
    col = levels(classcolours[[i]]),
    cex = .8)
  # mark the group centers
  for (type in 1:length(levels(tbstat[[i]]))) {
```

A Code

```r
```
A.8 Analysis: IMPI-MA

```r
x1 <- predict(spcP, matrix(apply(t(exprs(datalist[[i]]))[,grep(levels(tbstat[[i]])[[type]], eval(parse(text=paste("datalist[[i]]$",colour.variable,seps=" ")))]), 2, mean), ncol=1))
points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6, pch=9)

export.plot(file.prefix=paste('fig',q.i,'.','an.count','.',i,'.',figure,names(questions)[[q.i]],'.','analysis,names(datalist)[i],"PCA_all"),
export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
```

```r
figure <- figure+1

# preprocess####

# Variance stabilising transformation
dataList.v <- list()
for (i in 1:length(datalist)){
dataList.v[[i]] <- lumiT(datalist[[i]],simpleOutput=FALSE)
}

# Quantile normalise
dataList.q <- list()
for (i in 1:length(datalist.v)){
dataList.q[[i]] <- lumiN(datalist.v[[i]],method="quantile")
}

# show preprocessed data####

# boxplots of quantile normalised data
for (i in 1:length(datalist.q)){
boxplot(datalist.q[[i]],main=names(datalist.q)[i],col=as.character(classcolours[[i]]))
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.',i,'.',figure,names(questions)[[q.i]],'.','analysis,names(datalist)[i],"boxplot (q norm )")
export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
}
figure <- figure+1

# density plots of quantile normalised data
for (i in 1:length(datalist.q)){
density(datalist.q[[i]],main=names(datalist)[i],col=as.character(classcolours[[i]]),addLegend=FALSE)
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.',i,'.',figure,names(questions)[[q.i]],'.','analysis,names(datalist)[i],"density plot qnorm")
export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
}
figure <- figure+1

# MDS all probes (normalised)
plotSampleRelationsAD(datalist.q[[i]],method="mds",color=as.character(classcolours[[i]]),plotchar=16)
legend("bottomleft", levels(tbstat[[i]]), pch=16, col=levels(classcolours[[i]]), cex=.8)
```
```r
export.plot(file.prefix=paste('fig',q,i,'.',an.count,'.',i,'.',figure,
   names(questions)[[q,i]],'.','analysis',names(datalist)[i],"MDS_norm"),
   exportformats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
#
figure<--figure+1

#PCA all probes (normalised)
#for (i in 1:length(datalist.q)){
l=colnames(exprs(datalist.q[[i]]))
spca <- SamplePCA(exprs(datalist.q[[i]]),tbstat[[i]])
plot(spca, col=levels(classcolours[[i]]),main=paste(nrow(exprs(datalist.q
[[i]]))," probes (normalised")#,cex=2,cex.axis=2,cex.main=3,cex.lab
   =2.5,pch=16)
mtext(sprintf("%.2f",spca@variances[[1]]/sum(spca@variances)),side=1,line=3,
   adj=1,cex=1.5)
mtext(sprintf("%.2f",spca@variances[[2]]/sum(spca@variances)),side=2,line=3,
   adj=1,cex=1.5)
legend("bottomleft",
   levels(tbstat[[i]]),
pch=16,
col=levels(classcolours[[i]]),
cex=.8)
# mark the group centers
for (type in 1:length(levels(tbstat[[i]]))){
x1 <- predict(spca, matrix(apply(t(exprs(datalist.q[[i]]))[grep(levels(tbstat[[i]])[[type]],eval(parse(text=paste("datalist[[i]]$","variable",sep=""))))])
   ,2,mean), ncol=1))
points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6,pch=9)
}

export.plot(file.prefix=paste('fig',q,i,'.',an.count,'.',i,'.',figure,
   names(questions)[[q,i]],'.','analysis',names(datalist)[i],"PCA_all"),
   exportformats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
#
figure<--figure+1

#filter####

# new filter (requires beadarray; boxplot now uses ggplot2) (libraries
require integration with system setup)

#filter qnorm data
data.fil<-list()
#for (i in 1:length(datalist.q)){
x <- pqlist[[i]]
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
ids <- as.character(nuID2IlluminaID(as.character(featureNames(datalist.q[[
   i]])),chipVersion=getChipInfo(datalist.q[[i]])$chipVersion[[1]]))
is<--intersect(names(unlist(xx)),ids)#remove probes not in database, then
the mget call will not fail
qual <- unlist(mget(is, pqlist[[i]]))
table(qual)
rem <- qual == "No match" | qual == "Bad" | is.na(qual)
data.fil[[i]]<- datalist.q[[i]][!rem, ]
```
A.8 Analysis: IMPI-MA

```r
#show filtered probes ####

#filtered probes (PROBEQUALITY)
# for (i in 1:length(data.fil)) {
  par(mar=c(7,5,1,1))
  boxplot(data.fil[[i]], las=2, col=as.character(classcolours[[i]]), main=paste(names(datalist)[[i]], 'Boxplot of selected probes'))
  par(parbackup)
  export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure, names(questions))[[q.i]], '.', analysis, names(datalist)[i], 'boxplot filtered probes'), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
#}

figure <- figure + 1

# MDS selected probes

# for (i in 1:length(data.fil)){
  plotSampleRelationsAD(data.fil[[i]], method="mds", color=as.character(classcolours[[i]]), plotchar=16)
  legend("bottomleft",
          levels(tbstat[[i]]),
          pch=16,
          col=levels(classcolours[[i]]),
          cex=.8)
  export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure, names(questions))[[q.i]], '.', analysis, names(datalist)[i], "MDS_filtered"),
  export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
#}

figure <- figure + 1

# PCA all probes (filtered)

# for (i in 1:length(data.fil)){
  l=colnames(exprs(data.fil[[i]])
  spca <- SamplePCA(exprs(data.fil[[i]]), tbstat[[i]])
  plot(spca, col=levels(classcolours[[i]]), main=paste(nrow(exprs(data.fil[[i]]))," probes (filtered")#
         mtext(sprintf("%.2f",spca@variances[1]/sum(spcac@variances)),side=1,line=3, adj=1,cex=1.5)
         mtext(sprintf("%.2f",spca@variances[2]/sum(spcac@variances)),side=2,line=3, adj=1,cex=1.5)
         legend("bottomleft",
               levels(tbstat[[i]]),
               pch=16,
               col=levels(classcolours[[i]]),
               cex=.8)
  # mark the group centers
  for (type in 1:length(levels(tbstat[[i]]))){
    x1 <- predict(spca, matrix(apply(t(exprs(data.fil[[i]]))[,grep(levels(tbstat[[i]])[type], eval(parse(text=paste("datalist[[i]]$变量.
                    variable, sep="\"\")]))), 2, mean), ncol=1))
    points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6,pch=9)
```

379
```r
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.',i,'.',figure,names(questions)[[q.i]],'.','analysis.names(datalist)[i],"PCA_filtered"),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
#
figure<-figure+1
#DE statistics####
tbstat.class=list()
tbstat.class[[i]]<-as.factor(eval(parse(text=paste(dataset.variables[[i]],"$","contrast.variable",sep=""))))
contrast.levels<-levels(tbstat.class[[i]])
print(table(tbstat.class[[i]]))
#for (i in 1:length(data.fil)){
design[[i]]=model.matrix(~-1+factor(tbstat.class[[i]]))
colnames(design[[i]])<-levels(tbstat.class[[i]])
design[[i]]
#contrast[[i]]<-makeContrasts(activeTB-notActiveTB,levels=design[[i]])
contrast[[i]]<-makeContrasts(paste(contrast.levels[[1]],"-",contrast.levels[[2]]),
levels=design[[i]])
contrast[[i]]
#This is not ideal: use vanill pvalue of 0.05 and deal with very large or small results
# effect.size<-pwr.t2n.test(n1=13,n2=29,d=NULL,sig.level=0.01,power=0.9,
# alternative="two.sided")$d
# pval<-pwr.t2n.test(n1=table(tbstat.class[[i]])[[1]],n2=table(tbstat.class[[i]])[[2]],d=effect.size,sig.level=NULL,power=0.9,alternative="two.sided")$sig.level*3
# pwr.t2n.test(n1=5,n2=13,d=effect.size,sig.level=NULL,power=0.9,alternative ="two.sided")$sig.level*effect.size
# pwr.t2n.test(n1=20,n2=31,d=effect.size,sig.level=NULL,power=0.9,
# alternative ="two.sided")$sig.level*effect.size
#Linear model: tb status
fit[[i]]<-lmFit(data.fil[[i]],design[[i]])
fit2[[i]]<-contrasts.fit(fit[[i]],contrast[[i]])
fit2[[i]]<-eBayes(fit2[[i]])
gr[[i]]<-decideTests(fit2[[i]],adjust.method="none",p.value=0.05)
vc[[i]]<-vennCounts(gr[[i]])
gr2[[i]]<-decideTests(fit2[[i]],adjust.method="fdr",p.value=0.05)
vc2[[i]]<-vennCounts(gr2[[i]])
#Venn diagrams of DE probes
vennDiagram(gr[[i]])
vennDiagram(gr2[[i]])
#print unmodified results
print(paste("There are",vc2[[i]][[4]],"probes significant at P=0.05 (unadusted) and",vc2[[i]][[4]],"probes significant at P=0.05 (BH adjusted)"))
if(sum(mapply(abs,gr2[[i]]))<300) {

```

A.8 Analysis: IMPI-MA

```r
print(paste("with p value = 0.05", "there are", sum(mapply(abs, gr2[[i]]))," DE probes and top 200 results will be used for the remainder", sep = " ");
gr2[[i]]=is.element(rownames(exprs(data.fil[[i]])), as.character(probeID2nuID(topTable(fit2[[i]], coef=1, number=300, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")[[1]])[,7]))

if(sum(mapply(abs, gr2[[i]]))>2000) {
print(paste("with p value = 0.05", "there are", sum(mapply(abs, gr2[[i]]))," DE probes and top 200 results will be used for the remainder", sep = " ");
gr2[[i]]=is.element(rownames(exprs(data.fil[[i]])), as.character(probeID2nuID(topTable(fit2[[i]], coef=1, number=2000, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")[[1]])[,7]))
}
vc2[[i]]<-vennCounts(gr2[[i]])

#Venn diagrams of DE probes
vennDiagram(gr[[1]])
vennDiagram(gr2[[1]])

#results
reslist[[i]]<-featureNames(data.fil[[i]][gr2[[i]]!=0,])
reslistTTall[[i]]<-topTable(fit2[[i]], coef=1, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")
reslistTT30[[i]]<-topTable(fit2[[i]], coef=1, number=30, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")
reslistTT50[[i]]<-topTable(fit2[[i]], coef=1, number=500, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")
reslistTT100[[i]]<-topTable(fit2[[i]], coef=1, number=100, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")
reslistTT2000[[i]]<-topTable(fit2[[i]], coef=1, number=2000, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")
reslistTT8000[[i]]<-topTable(fit2[[i]], coef=1, number=8000, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")
reslistTT16000[[i]]<-topTable(fit2[[i]], coef=1, number=16000, adjust.method="BH", confint=T, sort.by="B", resort.by="logFC")

# can also restrict by pvalue, lfc, etc.
#
#volcanoplot
#for (i in 1:length(fit2)){
volcanoplot(fit2[[i]], coef=1, highlight=vc2[[i]][,2][2], cex=1, pch=20, col="red")
extport.plot(file.prefix=paste('fig', q.i, '.', 'an.count', '.', 'i', '.', 'figure', names(questions)[[4]], '.', 'analysis', names(datalist)[i], 'volcanoplot DE probes'), exportFormats=exportFormats.plots, height=height, width=(1 + sqrt(5))/2*height)
#
figure<-figure+1

#show DE probes####
#heatmaps
#DE probes
#for (i in 1:length(data.fil)){
```

phenomatrix <- matrix(cbind(as.character(classcolours[[i]]), as.character(classcolours[[i]])), ncol=2)
superHeatmap2(x=data.fil[[i]], y=gr2[[i]]!=0, phenomatrix=phenomatrix, scale="row")
export.plot(file.prefix=paste('fig', q.i, '.', 'analysis', names(datalist)[i], 'heatmap DE probes by pval'), export.formats=exportformats.plots, height=height, width=(1 + sqrt(5))/2*height)

for (i in 1:length(data.fil)){
  phenomatrix <- matrix(cbind(as.character(classcolours[[i]]), as.character(classcolours[[i]])), ncol=2)
  subset <- is.element(rownames(exprs(data.fil[[i]])), as.character(probeID2nuID(reslistTT[[i]][[1]]][,7]))
  superHeatmap2(x=data.fil[[i]], y=subset, phenomatrix=phenomatrix, scale="row")
  export.plot(file.prefix=paste('fig', q.i, '.', 'analysis', names(datalist)[i], 'heatmap diff probes top 500 by B'), export.formats=exportformats.plots, height=height, width=(1 + sqrt(5))/2*height)
}

# for (i in 1:length(data.fil)){
  plotSampleRelationsAD(data.fil[[i]][gr2[[i]]!=0,], method="mds", color=as.character(classcolours[[1]]), plotchar=16)
  legend("bottomleft", levels(tbstat[[i]]), pch=16, col=levels(classcolours[[1]]), cex=.8)
  export.plot(file.prefix=paste('fig', q.i, '.', 'analysis', names(datalist)[i], "MDS_DE"), export.formats=exportformats.plots, height=height, width=(1 + sqrt(5))/2*height)
}

for (i in 1:length(data.fil)){
  l=colnames(exprs(data.fil[[i]]))
  spca <- SamplePCA(exprs(data.fil[[i]][gr2[[i]]!=0,]), tbstat[[i]])
  plot(spca, col=levels(classcolours[[i]]), main=paste(nrow(exprs(data.fil[[i]][gr2[[i]]!=0,])), "probes")#, cex=2, cex.axis=2, cex.main=3, cex.lab=2.5, pch=16)
  mtext(sprintf("%.2f", spca@variances[1]/sum(spca@variances)), side=1, line=3, adj=1, cex=1.5)
  mtext(sprintf("%.2f", spca@variances[2]/sum(spca@variances)), side=2, line=3, adj=1, cex=1.5)
  legend("bottomleft", levels(tbstat[[i]]), cex=1.5)
}
pch = 16,
col = levels(classcolours[[i]]),
cex = 0.8)

# mark the group centers
for (type in 1:length(levels(tbstat[[i]]))){
  x1 <- predict(spca, matrix(apply(t(exprs(data.fil[[i]][gr2[[i]]]!=0,]),
        grep(levels(tbstat[[i]])[[type]], eval(parse(text=paste("datalist[[i]]$",
         colour.variable,sep=""))))), 2, mean), ncol=1)
  points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6, pch=9)
}

export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',',analysis,names(datalist)[i],"PCA_DE"),
        export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height
)
}

figure <- figure + 1

# Export the probe lists ####

dict <- nuID2IlluminaID(as.character(reslist[[i]]),species="Human")
dict.entrez <- nuID2EntrezID(as.character(reslist[[i]]),filterTh = NULL,
        lib.mapping='lumiHumanIDMapping', returnAllInfo = TRUE)
dict3 <- merge(dict, dict.entrez, by.x="nuID", by.y=0)
reslistTT2000[[i]]$ProbeID <- sapply(reslistTT2000[[i]]$ProbeID,
        function(x){sprintf("%010d",as.numeric(x)))})
full <- merge(dict3, reslistTT2000[[i]], by.x="Array_Address_Id", by.y="ProbeID",
        all.x=T)
write.csv(dict3, file=file.path(dir.results,paste(analysis,dataset.names[[i]],
        ",".csv")))
write.csv(reslistTT[[i]], file=file.path(dir.results,paste(analysis,dataset.
        names[[i]],"_top500.csv")))
write.csv(full, file=file.path(dir.results,paste(analysis,dataset.names[[i]],
        ",_sigWithFC.csv")))
write.csv(reslistTT2000[[i]], file=file.path(dir.results,paste(analysis,
        dataset.names[[i]],"_top2000.csv")))
xtabTop30 <- xtable(reslistTT30[[i]], align="c(rep("|p{1.2cm}"),ncol(
        reslistTT30[[i]])","p{1cm}"))
print(xtabTop30, hline.after=-1:nrow(reslistTT30[[i]]),include.colnames=T,
        include.rownames=F, size="tiny", file=
        file.path(dir.results,paste("Top 30 probes for question ",q.i,"",names(questions)[[q.i]],",",dataset.names[[i]],".tex",sep="")))
}

# temp closure for DE code
#
# Comparison
#
# # check overlap HIVneg and HIVpos (i.e. this is a comparison, not a contrast)####
#
# # this can be moved to venn script later
#
# # Venn diagram (this is actually pretty good)
# venndatax = list()
```r
# venndata[[i]] <- reslist[[i]]
# venndata500 <- list()
# for (i in 1:length(reslistTT)){
#    venndata500[[i]] <- reslistTT[[i]][,2]
# }
# venndata100 <- list()
# for (i in 1:length(reslistTT100)){
#    venndata100[[i]] <- reslistTT100[[i]][,2]
# }
# temp.venn.7 <- venn.diagram(
#   x = venndatax,
#   category=c("HIVneg","HIVpos"),
#   # filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
#   # filename = file.path(dir.figures,paste("fig","figure","_Venn_4set_gene.tiff","sep="")),
#   # filename = NULL,
#   # scaled = F, ext.text = TRUE, ext.line.lwd = 2,
#   # ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
#   # inverted = TRUE,
#   # cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
#   # main = "Overlap", sub = "MA, VAL, TRAIN", "393",
#   # main.cex = 2, sub.cex = 1,
#   # fill=c("green","red"),
#   # alpha=c(0.4,0.4),height=3000,width=(1 + sqrt(5))/2*3000
# )
# pdf(file=file.path(dir.figures,paste("fig","figure","_Venn_2set_probe.pdf","sep="")),height=7,width=(1 + sqrt(5))/2*7)
# grid.draw(temp.venn.7)
# dev.off()
# figure <- figure + 1
#
# temp.venn.8 <- venn.diagram(
#   x = venndata500,
#   category=c("HIVneg","HIVpos"),
#   # filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
#   # filename = file.path(dir.figures,paste("fig","figure","_Venn_4set_gene.tiff","sep="")),
#   # filename = NULL,
#   # scaled = F, ext.text = TRUE, ext.line.lwd = 2,
#   # ext.dist = -0.15, ext.length = 0.9, ext.pos = -4,
#   # inverted = TRUE,
#   # cex = 2.5, cat.cex = 2.5, rotation.degree = 0,
#   # main = "Overlap", sub = "MA, VAL, TRAIN", "393",
#   # main.cex = 2, sub.cex = 1,
#   # fill=c("green","red"),
#   # alpha=c(0.4,0.4),height=3000,width=(1 + sqrt(5))/2*3000
# )
# pdf(file=file.path(dir.figures,paste("fig","figure","_Venn_2set_top500_probe.pdf","sep="")),height=7,width=(1 + sqrt(5))/2*7)
# grid.draw(temp.venn.8)
# dev.off()
# figure <- figure + 1
#
# temp.venn.9 <- venn.diagram(
#   x = venndata100,
#   category=c("HIV neg","HIVpos"),
#   # filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
#   # filename = file.path(dir.figures,paste("fig","figure","_Venn_4set_gene.tiff","sep="")),
```
A.8 Analysis: IMPI-MA

# deconvolution####
analysis = "Deconvolution"
an.count <- 2
print("Analysis 2: Deconvolution and csDE")

# setup####
# questions:
do I use VST data?
do I use filtered subset for proportions and csDE?

for (i in 1:length(datalist.v)){
  figure <- 1
  #VST already done as well as filtering
dtd <- as(datalist.v[[i]], "ExpressionSet")
  #colnamelist <- contrast.variable
  meth <- list("Abbas"= gedBlood(dtd, verbose=T))
  for (h in 1:1) {
    h <- 1
    # deconvolution####
    decdat1 <- meth[[h]]
    colname <- contrast.variable
    colnames(coef(decdat1)) <- eval(parse(text=paste("dtd$", colname, sep="")))
    decdat2 <- asCBC(decdat1)
    decmat <- as.matrix(coef(decdat1))
    decmat.red <- decmat[apply(decmat, 1, sum)>0,]
    numbers <- as.vector(table(colnames(decmat.red)))
    names <- names(table(colnames(decmat.red)))
    nstring = paste("N=(", names[1], ",

    # stacked bar plot all types####
    par(parbackup)
    par(mar=c(8, 4.5, 4.5, 8))
    barplot(coef(decdat1), las=2, col=brewer.pal(12, "Set3"), legend.text=rownames(coef(decdat1)),
            main=paste("Proportions of ", names(meth)[[h]], names(datalist)[[i]], nstring), beside=F,
            ylab="Proportion", args.legend=c(x=ncol(decmat.red)*1.35, y=1, cex=.7),
            cex.names=.8)
    par(parbackup)
A Code

```r
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.' ,i,'.' ,figure,
 names(questions)[[q.i]],'.' ,analysis, names(datalist)[i],"stacked_barplot _all",names(meth)[[h]]), export.formats=export.formats.plots,height=
height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

# stacked barplot detected types####
par(parbackup)
par(par=mar=c(8,4.5,4,5.8))
barplot(decmat.red[,las=2,col=brewer.pal(12,"Set3"),legend.text=rownames(decmat.red),main=paste("Proportions of" ,rownames(decmat.red),"cell types",names(meth)[[h]],names(datalist)[[i]],nstring),beside=F,ylab="Proportion",args.legend=c(x=ncol(decmat.red)*1.35,y=1,cex=.7),cex.names=.8)
par(parbackup)
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.' ,i,'.' ,figure,
 names(questions)[[q.i]],'.' ,analysis, names(datalist)[i],"stacked_barplot _detected",names(meth)[[h]]), export.formats=export.formats.plots,height=
height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

# stacked barplot CBC####
par(parbackup)
par(par=mar=c(8,4.5,4.5,8))
barplot(coef(decdat2),las=2,col=brewer.pal(12,"Set3"),legend.text=rownames(coef(decdat2)),main=paste("Proportions of CBC cell types",names(meth)[[h]],names(datalist)[[i]],nstring),beside=F,ylab="Proportion",args.legend=c(x=ncol(coef(decdat2))*1.35,y=1,cex=.7),cex.names=.8)
par(parbackup)
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.' ,i,'.' ,figure,
 names(questions)[[q.i]],'.' ,analysis, names(datalist)[i],"stacked_barplot (CBC)",names(meth)[[h]]), export.formats=export.formats.plots,height=
height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

# stacked barplot detected types (PBMC only) ####
decmat.red.PBMC<-matrix(data=NA,nrow=nrow(decmat.red)-1,ncol=ncol(decmat.red))
rownames(decmat.red.PBMC)<-rownames(decmat.red[-nrow(decmat.red),])
colnames(decmat.red.PBMC)<-colnames(decmat.red)
for(l in 1:ncol(decmat.red)){
decmat.red.PBMC[,l]<-decmat.red[-nrow(decmat.red),l]/sum(decmat.red[-nrow(decmat.red),l])
}

# stacked barplot PBMC####
par(parbackup)
par(par=mar=c(8,4.5,4.5,8))
barplot(decmat.red.PBMC[,las=2,col=brewer.pal(12,"Set3"),legend.text=rownames(decmat.red.PBMC),main=paste("Proportions of" ,rownames(decmat.red.PBMC),"PBMC cell types",names(meth)[[h]],names(datalist)[[i]],nstring),beside=F,ylab="Proportion",args.legend=c(x=ncol(decmat.red.PBMC)*1.35,y=1,cex=.7),cex.names=.8)
par(parbackup)
export.plot(file.prefix=paste('fig',q.i,'.','an.count','.' ,i,'.' ,figure,
 names(questions)[[q.i]],'.' ,analysis, names(datalist)[i],"stacked_barplot (PBMC)",names(meth)[[h]]), export.formats=export.formats.plots,height=
height*1.5,width=(1 + sqrt(5))/2*height*1.5)
```
```r
figure <- figure + 1

# boxplot by condition ####

## convert and calculate

c1N <- colnames(decmat.red)
dN <- t(decmat.red)
VN <- as.vector(dN)
c1vN <- rep(c1N, nrow(decmat.red))
typesN <- rownames(decmat.red)
typevN <- list()
for (j in 1:nrow(decmat.red)) { typevN[[j]] <- rep(typesN[[j]], ncol(decmat.red)) }
type2vN <- unlist(typevN)

dN <- data.frame(vN, c1vN, type2vN)
levs <- levels(as.factor(colnames(decmat.red)))

# plot

par(mar = c(10.5, 4.5, 4.5, 8))
boxplot(vN ~ c1vN * type2vN, data = dN, las = 2, col = c("red", "green", "blue", "yellow")[[1:length(levs)]], main = paste("Relative abundance of cell types by ", colname, "in ", names(datalist)[[i]], 

varwidth = FALSE)
legend("topleft", legend = (levs), fill = c("red", "green", "blue", "yellow"))
par(parbackup)

export.plot(file.prefix = paste("fig", q.i, ".", an.count, ".", i, ".", figure, "boxplot_by_celltype", names(meth)[[h]]), export.formats = exportformats.plots, height = height * 1.5, width = (1 + sqrt(5)) / 2 * height * 1.5)
figure <- figure + 1

# exp

# par(mar = c(10.5, 4.5, 4.5, 8))
# boxplotBy(decmat.red, as.factor(colnames(decmat.red)), col = c("blue", "yellow"), main = paste("func : boxplotBy ", colnamelist[[i]], " in ", names(datalist)[[i]], nstring, names(meth)[[h]]), scale = TRUE)
# par(parbackup)
# export.plot(file.prefix = paste("fig", figure, ".", i, "boxplot_by_celltype", names(datalist)[[i]], names(meth)[[h]]), export.formats = exportformats.plots, height = height * 1.5, width = (1 + sqrt(5)) / 2 * height * 1.5)
# exp
#
#
# stats for comparisons ####

statlist <- list()
for (k in 1:nrow(decmat.red)) { statlist[[k]] <- wilcox.test(decmat.red[k, colnames(decmat.red) == levs[i]], decmat.red[k, colnames(decmat.red) == levs[2]])
} p.value)
tests <- data.frame("names" = rownames(decmat.red), "pval" = as.vector(unlist(statlist)), "man_B" = as.vector(unlist(statlist)) * nrow(decmat.red), "bonf" = p.adjust(as.vector(unlist(statlist)), method = "bonferroni"), "BH" = p.adjust(as.vector(unlist(statlist)), method = "BH")
write.csv(tests, file = file.path(dir.results, paste("Stats_", q.i, ".", an.count, ".", i, ", ", names(datalist)[[i]], names(meth)[[h]], ", ", colname, ", wilcox.csv", sep = "")))
```

387
# boxplot by condition for matched blood/fluid for proportions normalised as PBMCs (i.e. neutrophils removed)####

convert

c1N <- colnames(decmat.red.PBMC)
dN <- t(decmat.red.PBMC)
vN <- as.vector(dN)
c1vN <- rep(c1N, nrow(decmat.red.PBMC))
typesN <- rownames(decmat.red.PBMC)
typevN <- list()
for (j in 1:nrow(decmat.red.PBMC)){typevN[[j]]<-rep(typesN[[j]], ncol(decmat.red.PBMC))}
typev2N <- unlist(typevN)
dN <- data.frame(vN, c1vN, typev2N)
levs <- levels(as.factor(colnames(decmat.red.PBMC)))
par(mar = c(10.5, 4.5, 4.5, 8))
boxplot(vN ~ c1vN * typev2N, data = dN, las = 2, col = c("red", "green", "blue", "yellow"))[1:length(levs)], main = paste("Relative abundance of PBMC types by ", colname, " in ", names(datalist)[[i]], " nstring", names(meth)[[h]]))
legend("topleft", legend = (levs), fill = c("red", "green", "blue", "yellow"))
par(parbackup)
export.plot(file.prefix = paste('fig', q.i, '.', an.count, '.', i, '.', figure, names(questions)[[i]], '"', i, '"', figure, names(datalist)[[i]], " boxplot _by _PBMC _celltype", names(meth)[[h]]), export.formats = export.formats.plots, height = height*1.5, width = (1 + sqrt(5)) / 2*height*1.5)
figure <- figure + 1

# stats for comparisons####

statlist <- list()
for (k in 1:nrow(decmat.red.PBMC)) {statlist[[k]]<-wilcox.test(decmat.red.PBMC[k, colnames(decmat.red.PBMC)==levs[1]], decmat.red.PBMC[k, colnames(decmat.red.PBMC)==levs[2]])$p.value}

tests.PBMC <- data.frame("names"=rownames(decmat.red.PBMC), "pval" = as.vector(unlist(statlist)) * nrow(decmat.red.PBMC))
# bonferroni
tests.PBMC <- data.frame("names"=rownames(decmat.red.PBMC), "pval" = as.vector(unlist(statlist)) * nrow(decmat.red.PBMC), "man_B" = as.vector(unlist(statlist)) * nrow(decmat.red.PBMC), "bonf" = p.adjust(as.vector(unlist(statlist)), method = "bonferroni"), "BH" = p.adjust(as.vector(unlist(statlist)), method = "BH"))
write.csv(tests.PBMC, file = file.path(dir.results, paste("Stats_PBMC_", q.i, ", ", an.count, ", ", i, ", ", figure, names(datalist)[[i]], "", names(meth)[[h]], "", colname, "_wilcox.csv", sep="")))

#csde ####

#PC added to B cells
th <- apply(coef(decdat1)[1:2,], 2, sum)
tc <- apply(coef(decdat1)[3:4,], 2, sum)
b <- apply(coef(decdat1)[5:10,], 2, sum)
#pc <- coef(decdat1)[10]

nk <- apply(coef(decdat1)[11:12,], 2, sum)
mo <- apply(coef(decdat1)[13:14,], 2, sum)
dc <- apply(coef(decdat1)[15:16,], 2, sum)
neut <- coef(decdat1)[17]

props <- matrix(
  rbind(th, tc, b, nk, mo, dc, neut)
), nrow=7)
1081 rownames(props)<-c("th","tc","b","nk","mo","dc","neut")
1082 colnames(props)<-colnames(coef(decdat1))

# Smallprops
1085 l<-apply(coef(decdat1)[1:10,],2,sum)
1086 nk<-apply(coef(decdat1)[11:12,],2,sum)
1087 apc<-apply(coef(decdat1)[13:16,],2,sum)
1088 neut<-coef(decdat1)[17,]

1089 smallprops<-matrix(c(l,nk,apc,neut),nrow =4)
1090 rownames(smallprops)<-c("lym","nk","apc","neut")
1091 colnames(smallprops)<-colnames(coef(decdat1))

1099 subset2<-is.element(rownames(exprs(datalist.v[[i]])),as.character(probeID2nuID(reslistTT[[i]][,7])))
1100 subset3<-is.element(rownames(exprs(datalist.v[[i]])),as.character(nuID2targetID(rownames(exprs(datalist.v[[i]]))),featureData(Abbas)$SYMBOL))
1101 subset4<-is.element(rownames(datalist.v[[i]]),as.character(probeID2nuID(reslistTT2000[[i]][,7])))
1102 subset5<-is.element(rownames(datalist.v[[i]]),as.character(probeID2nuID(reslistTT8000[[i]][,7])))

1109 if(min(table(tbstat.class[[i]]))<nrow(props)) { props<-smallprops print("smallprops used")} else { print("props used")}

1123 try (tt<-csTopTable(cs.de,decreasing=F))
1124 if(tt != "no csDE result "){ names(tt$th)
1125 dict.th<-nuID2IlluminaID(as.character(names(tt$th)),species="Human")
1126 write.csv(dict.th,file=file.path(dir.results,paste("csDE_Th_",q.i,'.',an.count,'.',i,'.',figure,names(questions)[[q.i]],'.',',analysis,names(datalist)[i],"csDE_FDR_by_celltype",names(datalist)[[i]],names(meth)[[h]]),export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5);
}

1132 "no csDE result"
1134 if(tt != "no csDE result "){ names(tt$neut)
1135 dict.th<-nuID2IlluminaID(as.character(names(tt$th)),species="Human")
1137 write.csv(dict.th,file=file.path(dir.results,paste("csDE_Th_",q.i,'.',',',an.count,'.',',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,"_",csv",sep=""'))
A Code

```r
1128  dict.tc <- nuID2IlluminaID(as.character(names(tt$tc)), species="Human")
1129  write.csv(dict.tc, file=file.path(dir.results, paste("csDE_Tc_*", q.i, ".", an.
1130  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1131  ", sep=""))
1132  dict.b <- nuID2IlluminaID(as.character(names(tt$b)), species="Human")
1133  write.csv(dict.b, file=file.path(dir.results, paste("csDE_B_*", q.i, ",", an.
1134  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1135  ", sep=""))
1136  dict.nk <- nuID2IlluminaID(as.character(names(tt$nk)), species="Human")
1137  write.csv(dict.nk, file=file.path(dir.results, paste("csDE_NK_*", q.i, ",", an.
1138  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1139  ", sep=""))
1140  dict.mo <- nuID2IlluminaID(as.character(names(tt$mo)), species="Human")
1141  write.csv(dict.mo, file=file.path(dir.results, paste("csDE_MO_*", q.i, ",", an.
1142  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1143  ", sep=""))
1144  dict.dc <- nuID2IlluminaID(as.character(names(tt$dc)), species="Human")
1145  write.csv(dict.dc, file=file.path(dir.results, paste("csDE_DC_*", q.i, ",", an.
1146  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1147  ", sep=""))
1148  dict.neut <- nuID2IlluminaID(as.character(names(tt$neut)), species="Human")
1149  write.csv(dict.neut, file=file.path(dir.results, paste("csDE_neut_*", q.i, ",", an.
1150  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1151  ", sep=""))
1152  dict.lym <- nuID2IlluminaID(as.character(names(tt$lym)), species="Human")
1153  write.csv(dict.lym, file=file.path(dir.results, paste("csDE_lym_*", q.i, ",", an.
1154  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1155  ", sep=""))
1156  dict.apc <- nuID2IlluminaID(as.character(names(tt$apc)), species="Human")
1157  write.csv(dict.apc, file=file.path(dir.results, paste("csDE_apc_*", q.i, ",", an.
1158  .count, ",", i, ",", names(datalist)[i], names(meth)[h], ",", colname, ".csv
1159  ", sep="")))} else print(tt)
1160
1161  # } methods in deconvolution for loop
1162  # deconvolution for loop
1163
1164  # wgcna ####
1165  analysis = "wgcna"
1166  an.count <- 3
1167  print("Analysis 3: WGCNA")
1168
1169  # setup ####
1170  # for (i in 1:length(datalist)){
1171  # reset figure
1172  figure <- 1
1173  # select data for WGCNA: use top 8000 DE genes
1174  subset <- is.element(rownames(exprs(data.fil[[i]])), as.character(probeID2nuID(reslistTT8000[[i]][[i]][[7]]))
1175  usedata <- data.fil[[i]][subset,]
1176  datExpr0 = data.frame(t(exprs(usedata)))
1177
1178  # Take a quick look at what is in the data set:
1179  dim(datExpr0);
1180  names(datExpr0)[1:10];#probes
1181  rownames(datExpr0)#samples
1182
1183  #1.b Checking data for excessive missing values and identification of outliers microarray samples####
```
# check for missing values
```
gsg = goodSamplesGenes(datExpr0, verbose=3);
gsg$allOK
```

# remove offending samples
```
if (!gsg$allOK)
{
  # Optionally, print the gene and sample names that were removed:
  if (sum(!gsg$goodGenes)>0)
    printFlush(paste("Removing genes:" , paste(names(datExpr0)[!gsg$goodGenes ], collapse="", "")));
  if (sum(!gsg$goodSamples)>0)
    printFlush(paste("Removing samples:" , paste(rownames(datExpr0)[!gsg$goodSamples], collapse="", "")));
  # Remove the offending genes and samples from the data:
  datExpr0 = datExpr0[gsg$goodSamples,gsg$goodGenes]
}
```

# cluster samples to look for outliers####
```
sampleTree = flashClust(dist(datExpr0), method = "average");
```

# sizeGrWindow(12,9)
```
pdf(file = "Plots/sampleClustering.pdf", width = 12, height = 9);
par(cex = 0.6);
par(mar = c(0,4,2,0))
plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="",
cex.lab = 1.5, cex.axis = 1.5, cex.main = 2)
# Plot a line to show the cut
abline(h = 80, col = "red")
export.plot(file.prefix=paste("fig",q.i,."","an.count",".",figure,names(questions)[[q.i]],'.' ,analysis, names(datalist)[i],'outlier plot all samples'), exportformats=exportformats.plots, height=height, width=(1 + sqrt(5))/2*height)
par(parbackup)
```

# Determine cluster under the line
```
clust = cutreeStatic(sampleTree, cutHeight = 80, minSize = 10)
table(clust)
# clust 1 contains the samples we want to keep.
keepSamples = (clust==1)
datExpr = datExpr0[keepSamples,]
genExpr = ncol(datExpr)
SampExpr = nrow(datExpr)
```

#removing outliers breaks the code further down. For now, don't remove outliers at all
```
datExpr=datExpr0
```

#1.c Prepare clinical trait data#####
```
traitData = pData(usedata);
dim(traitData)
names(traitData)
```

# remove columns that hold information we do not need.
```
base.traits<-c(2,7,19,45,47,9,10,11,12,13,14,15,16,17,18,20,22,38)
```
allTraits = traitData[, c(base.traits, extra.traits)]

#allTraits = traitData
#allTraits = allTraits[, c(2, 11:36)];
dim(allTraits)
names(allTraits)

pcSamples = rownames(datExpr);
traitRows = match(pcSamples, allTraits$sample_name);
datTraits = allTraits[traitRows, -1];
rownames(datTraits) <- traitData$sample_name

datTraits <- data.frame(apply(datTraits, 2, function(x){replace(x, x == "",NA)}) )

datTraits[, c(1,2,3,4,5,13,14,17)] <- lapply(datTraits[, c(1,2,3,4,5,13,14,17)], as.factor)
datTraits[, c(1,2,3,4,5,13,14,17)] <- lapply(datTraits[, c(1,2,3,4,5,13,14,17)], as.numeric)
datTraits <- as.matrix(datTraits)
datTraits <- apply(datTraits,2,as.numeric)

#datTraits <- data.frame(lapply(datTraits,as.numeric))
rownames(datTraits) = allTraits[traitRows, 1];
collectGarbage();
datTraits <- datTraits[,colSums(is.na(datTraits))<nrow(datTraits)]#this removes all columns where all values are NA
uniquelength <- apply(datTraits,2,function(x) length(unique(x[!is.na(x)])))
datTraits <- subset(datTraits, select=uniquelength>1)#this removes all columns where all values are the same after removing NAs
datTraits <- data.frame(datTraits)

#expression data and phenotype data are now in analagous data frames

#Visualisation of how the traits relate to the sample dendrogram####
# Re-cluster samples
sampleTree2 = flashClust(dist(datExpr), method = "average")

# Convert traits to a color representation: white means low, red means high, grey means missing entry
traitColors = numbers2colors(as.matrix(datTraits), signed = FALSE);

# Plot the sample dendrogram and the colors underneath.
plotDendroAndColors(sampleTree2, traitColors,cex.dendroLabels = 0.5,cex.axis=.7,
                      groupLabels = colnames(datTraits),
                      main = "Sample dendrogram and trait heatmap")

export.plot(file.prefix=paste('fig',q.i, '.' ,an.count, '.' ,i, '.' ,figure,names(questions))[[q.i]],'
',analysis, names(datalist)[i],'
samples_trait_combo_plot'),export.formats=export.formats.plots,height=height, width=(1 + sqrt(5))/2*height)

par(parbackup)
figure<--figure+1

#2.c.1 Choosing the soft-thresholding power: analysis of network topology####
# Choose a set of soft-thresholding powers
powers = c(c(1:10), seq(from = 12, to=20, by=2))

# Call the network topology analysis function
```r
sft = pickSoftThreshold(datExpr, powerVector = powers, RsquaredCut = 0.80, verbose = 5)

# Plot the results:
#sizeGrWindow(9, 5)
par(mfrow = c(1,2));par(mar=c(4,4,2,2))
cex1 = 0.9;
# Scale-free topology fit index as a function of the soft-thresholding power
plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
    xlab="Soft Threshold (power)",ylab="Scale Free Topology Model Fit, signed R^2",type="n",
    main = paste("Scale independence"));
text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
    labels=powers, cex=cex1, col="red");
# this line corresponds to using an R^2 cut-off of h
abline(h=0.80, col="red")
# Mean connectivity as a function of the soft-thresholding power
plot(sft$fitIndices[,1], sft$fitIndices[,5],
    xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",
    main = paste("Mean connectivity")
text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1, col="red")
par(mfrow = c(1,1));

export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.'i,'.'figure,names(questions)[[q.i]],'.',analysis,names(datalist)[i],'scale_independence_and_mean_connectivity'),exportformats=exportformats.plots,height=height,width=(1+sqrt(5))/2*height)
par(parbackup)
figure<-figure+1
print(paste("Soft-thresholding power: ",sft$powerEstimate ))

#2.c.2 Block-wise network construction and module detection####
thispower = sft$powerEstimate
if (is.na(thispower)){
    thispower=min(sft$fitIndices[sft$fitIndices$SFT.R.sq>0.75,]$Power)
    print("No power estimate for R squared of .80, using .75")
    print(paste("New soft-thresholding power: ",thispower )))
bnnet = blockwiseModules(datExpr, maxBlockSize = 8000,
    power = thispower,
    networkType='signed',TOMType="signed",
    minModuleSize = 20,
    reassignThreshold = 0, mergeCutHeight = 0.25,
    numericLabels = TRUE,
    saveTOMs = FALSE,
    saveTOMFileBase = "TBPC_TOM_blockwise_10K",
    verbose = 3)

# Convert labels to colors for plotting
mergedColors = labels2colors(bnnet$colors)
# Plot the dendrogram and the module colors underneath
plotDendroAndColors(bnnet$Dendrograms[[1]], mergedColors[bnnet$blockGenes[[1]]],
    "Module colors",
    dendroLabels = FALSE, hang = 0.03,
    addGuide = TRUE, guideHang = 0.05,
    main=paste(names(datalist[[1]]),nrow(datExpr),"samples",ncol(datExpr),"probes")
```

```r
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(questions)[[q.i]],'.','analysis.names(datalist)[i],',dendrogram_and_module_colours'),export.formats=export.formats.plots,height=height,width=(1+sqrt(5))/2*height)
par(parbackup)
figure<-figure+1

#plotColorUnderTree(bwnet$dendrograms[[1]], mergedColors[bwnet$blockGenes[[1]]])
#save the tree (module assignment and module eigengene information)
bwmoduleLabels = bwnet$colors
bwmoduleColors = labels2colors(bwnet$colors)
bwMEs = bwnet$MEs;
bwgeneTree = bwnet$dendrograms[[1]];
# Relabel blockwise modules
bwLabels = matchLabels(bwnet$colors, bwmoduleLabels);
# Convert labels to colors for plotting
bwModuleColors = labels2colors(bwLabels)

print("number of genes in each module")
print(table(bwLabels))

#3. Relating modules to external information and identifying important genes
###

#3.a Quantifying module trait associations####

# Define numbers of genes and samples
nGenes = ncol(datExpr);
nSamples = nrow(datExpr);
# Recalculate MEs with color labels
MEs0 = moduleEigengenes(datExpr, bwmoduleColors)$eigengenes
MEs = orderMEs(MEs0)
moduleTraitCor = cor(MEs, datTraits, use = "p");
moduleTraitPvalue = corPvalueStudent(moduleTraitCor, nSamples);

# Will display correlations and their p-values
textMatrix = paste(signif(moduleTraitCor, 2), "\n(signif(moduleTraitPvalue, 1), ", sep = "\n");
dim(textMatrix) = dim(moduleTraitCor)
par(mar = c(5,9,2,2));par(mfrow=c(1,1))
# Display the correlation values within a heatmap plot
labeledHeatmap(Matrix = moduleTraitCor,
  xLabels = names(datTraits),
  yLabels = names(MEs),
  ySymbols = names(MEs),
  colorLabels = FALSE,
  colors = greenWhiteRed(50),
  textMatrix = textMatrix,
  setStdMargins = FALSE,
  cex.text = 0.5,
  zlim = c(-1,1),
  main = paste("Module-trait relationships"))
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(questions)[[q.i]],'.','analysis.names(datalist)[i],',module_trait_relationships'),export.formats=export.formats.plots,height=height,width =(1+sqrt(5))/2*height)
```

394
A.8 Analysis: IMPI-MA

```r
par(parbackup)
figure<-figure+1

# clustered version, with NA columns removed
par(mar = c(2,2,2,5));par(cex.main=.5);
heatmap(moduleTraitCor[,!apply(moduleTraitCor,2,is.na)[1,]],col=
greenWhiteRed(50),main="Clustered module-trait relationships",mar=c(6,1)
,cexRow=.5,cexCol=.5)

export.plot(file.prefix=paste('fig',q.i,':',an.count,':',i,':',figure,names(
questions)[[q.i]],',',analysis,names(datalist)[i],':clust_module-trait_
relationships'),exportformats=export.formats.plots,height=height,width
=(1 + sqrt(5))/2*height)
par(parbackup)
figure<-figure+1

#3.b Gene relationship to trait and important modules: Gene Significance and
Module Membership###

# We quantify associations of individual genes with our trait of interest (which depends on the question/contrast) by defining Gene Significance GS
# as (the absolute value of) the correlation between the gene and the trait.
# For each module, we also define a quantitative measure of module membership MM as the correlation of the module eigengene and the gene expression
# profile. This allows us to quantify the similarity of all genes on the array to every module.

# in this context 'class' refers to the contrast variable!
class=as.data.frame(eval(parse(text=paste("datTraits$",contrast.variable))))
names(class)=contrast.variable

# names (colors) of the modules
geneModuleMembership = as.data.frame(cor(datExpr, MEs, use = "p"));
MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership),
nSamples));
names(geneModuleMembership) = paste("MM", modNames, sep=" ");
names(MMPvalue) = paste("p.MM", modNames, sep=" ");
geneTraitSignificance = as.data.frame(cor(datExpr, class, use = "p"));
GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance),
nSamples));
names(geneTraitSignificance) = paste("GS.", names(class), sep=" ");
names(GSPvalue) = paste("p.GS.", names(class), sep=" ");

#3.c Intramodular analysis: identifying genes with high GS and MM####

# Using the GS and MM measures, we can identify genes that have a high significance for class as well as high module membership in interesting modules. We plot a scatterplot of Gene Significance vs. Module Membership in all modules:
par(parbackup);
for (mods in modNames){
  module = mods
column = match(module, modNames);
  moduleGenes = bwmoduleColors==module;
  ...}
```

if(length(abs(geneModuleMembership[moduleGenes, column]))>19){
    verboseScatterplot(abs(geneModuleMembership[moduleGenes, column]),
                        abs(geneTraitSignificance[moduleGenes, 1]),
                        abline=T, abline.color = "red",
                        xlab = paste("Module Membership in", module, " module"),
                        ylab = paste("Gene significance for", contrast.variable),
                        main = paste(mods," module membership vs. gene significance\n"),
                        cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col =
                        module)
    export.plot(file.prefix=paste('fig',q.i, '.',an.count,'.',i,'.',figure,names(questions))[q.i],',',analysis,names(datalist)[i],',MM_vs_GS',module),
    exportformats=exportformats.plots,height=height,width=(1 + sqrt(5))/2* height)
    par(parbackup)
}
}

#3.d Summary output of network analysis results####

features<-featureNames(usedata)
annotSYM<-as.character(getSYMBOL(features, "lumiHumanAll.db"))
annotACC<-as.character(lookUp(features, "lumiHumanAll.db", "ACCNUM"))
annotCHR<-as.character(lookUp(features, "lumiHumanAll.db", "CHR"))
annotCHRLOC<-as.character(lookUp(features, "lumiHumanAll.db", "CHRLOC"))
annotENSEMBL<-as.character(lookUp(features, "lumiHumanAll.db", "ENSEMBL"))
annotENTREZID<-as.character(lookUp(features, "lumiHumanAll.db", "ENTREZID"))
annotGO<-as.character(lookUp(features, "lumiHumanAll.db", "GO"))
annotREFSEQ<-as.character(lookUp(features, "lumiHumanAll.db", "REFSEQ"))
annotUNIPROT<-as.character(lookUp(features, "lumiHumanAll.db", "UNIPROT"))
annot<-data.frame(cbind(features, annotSYM, annotACC, annotCHR, annotCHRLOC,
                        annotENSEMBL, annotENTREZID, annotGO, annotREFSEQ, annotUNIPROT))

#-- need to do this manually:
dim(annot)
names(annot)
probes = names(datExpr)
probes2annot = match(probes, annot$features)
# The following is the number of probes without annotation:
sum(is.na(probes2annot))
# Should return 0. It doesn’t. Not a surprise.

# Create the starting data frame
geneInfo0 = data.frame(substanceBXH = probes,
                         geneSymbol = annot$annotSYM[probes2annot],
                         LocusLinkID = annot$annotENTREZID[probes2annot],
                         moduleColor = bwmoduleColors,
                         geneTraitSignificance,
                         GSPvalue)

# Order modules by their significance for class
modOrder = order(-abs(cor(MEs, class, use = "p")));# Add module membership information in the chosen order
for (mod in 1:ncol(geneModuleMembership)){
    oldNames = names(geneInfo0)
    geneInfo0 = data.frame(geneInfo0, geneModuleMembership[, modOrder[mod]],
                        geneSymbol = annot$annotSYM[probes2annot],
                        LocusLinkID = annot$annotENTREZID[probes2annot],
                        moduleColor = bwmoduleColors,
                        geneTraitSignificance,
                        GSPvalue)
A.8 Analysis: IMPI-MA

MMPvalue[, modOrder[mod]];

names(geneInfo0) = c(oldNames, paste("MM." , modNames[modOrder[mod]], sep=" " ),

paste("p.MM." , modNames[modOrder[mod]], sep=" " ))
}

# Order the genes in the geneInfo variable first by module color, then by
geneTraitSignificance
geneOrder = order(geneInfo0$moduleColor, -abs(eval(parse(text=paste(" geneInfo0$GS." , contrast.variable , sep=" ")))));
geneInfo = geneInfo0[geneOrder, ]
write.csv(geneInfo, file=file.path(dir.results, paste('table', q.i, '.', an.count , ',i', '.' , names(questions)[[q.i]], '.', analysis, names(datalist)[i], " geneInfoClass.csv"))

#4 Interfacing network analysis with other data such as functional
annotation and gene ontology####

#4.a Output gene lists for use with online software and services####

# Match probes in the data set to the probe IDs in the annotation file
probes = names(datExpr)
probes2annot = match(probes, annot$features)
# Get the corresponding Locus Link IDs
allLLIDs = as.numeric(annot$annotENTREZID[probes2annot]);
allLLIDs = allLLIDs[!is.na(allLLIDs)]
# $ Choose interesting modules
intModules = unique(bwmoduleColors)

for (module in intModules){
    # Select module probes
    modGenes = (bwmoduleColors==module)
    # Get their entrez ID codes
    modLLIDs = allLLIDs[modGenes];
    # Write them into a file
    fileName = file.path(dir.results, paste('LLID ', q.i, '.', an.count, '.i', '.', names(datalist)[[i]], "_LocusLinkIDs -", module, ".txt" ));
    write.table(as.data.frame(modLLIDs), file = fileName, row.names = FALSE, col.names = FALSE)
}

# As background in the enrichment analysis, we will use all probes in the
analysis.
fileName = file.path(dir.results, paste('ALL-LLID', q.i, '.', an.count, '.', i, '.', 
,names(datalist)[[i]], "_entrez_ids-all.txt" ));
write.table(as.data.frame(allLLIDs), file = fileName, row.names = FALSE, col.names = FALSE)

#4.b Enrichment analysis directly within R####
GGenr = GGenrichmentAnalysis(bwmoduleColors, allLLIDs, organism = "human", nBestP = 20,includeOffspring=FALSE);

tab = GGenr$bestPTerms[[4]]$enrichment
names(tab)
write.table(tab, file=file.path(dir.results, paste('GO', q.i, '.', an.count, '.', i 
,'.' ,names(datalist)[[i]], "_GOEnrichmentTable.csv")), sep=" ",quote=TRUE, row.names=FALSE)

#on-screen
keepCols = c(1, 2, 5, 6, 7, 12, 13);
screenTab = tab[, keepCols];
# Round the numeric columns to 2 decimal places:
numCols = c(3, 4);
screenTab[, numCols] = signif(apply(screenTab[, numCols], 2, as.numeric), 2)
# Truncate the the term name to at most 40 characters
screenTab[, 7] = substring(screenTab[, 7], 1, 40)
# Shorten the column names:
colnames(screenTab) = c("module", "size", "p-val", "Bonf", "nInTerm", "ont", 
  "term name");
rownames(screenTab) = NULL;
# Set the width of R output. The reader should play with this number to 
# obtain satisfactory output.
options(width=95)
# Finally, display the enrichment table:
screenTab

#5. Network visualization using WGCNA functions####
#5a. make TOMplot # only run this if lots of memory. don’t include in RData 
####
# problems:
#1. not complete network, but only one block at a time
#2. despite this, it is too big
# Calculate topological overlap anew: this could be done more efficiently by 
saving the TOM
# get hold of TOMs inRData files in “
# calculated during module detection, but let us do it again here. #!No!
# dissTOM = 1-TOMsimilarityFromExpr(datExpr, power = 11);
# do this for eachj block, i.e.TOM?
# #don’t do TOM plots at this stage
# # load("TBPC_TOM_blockwise_10K-block.1.RData")
# TOM<-as.matrix(TOM)
# dissTOM1=1-TOM
# # Transform dissTOM with a power to make moderately strong connections 
# more visible in the heatmap
# plotTOM1 = dissTOM1^7;
# # Set diagonal to NA for a nicer plot
# diag(plotTOM1) = NA;
# # Call the plot function
# #sizeGrWindow(9,9)
# #-- can’t plot whole tom due to memory constraints...
# plotDendroAndColors(bwnet$dendrograms[[1]], bwModuleColors[bwnet$ 
  blockGenes[[1]]], 
  "Module colors", main = "Gene dendrogram and module 
  colors in block 1", 
  dendroLabels = FALSE, hang = 0.03, 
  addGuide = TRUE, guideHang = 0.05)
# TOMplot(plotTOM1,bwnet$dendrograms[[1]], bwModuleColors[bwnet$blockGenes 
  [[1]]], main = "Network heatmap plot, all genes, block1")
# #Plot only some genes..
# nSelect = 1500
# For reproducibility, we set the random seed
set.seed(10);

# There is no simple way of restricting a clustering tree to a subset of
genes, so we must re-cluster.

# selectTree = flashClust(as.dist(selectTOM), method = "average")
# selectColors = bwmoduleColors[select];
# # Open a graphical window
# sizeGrWindow(9,9)
# Taking the dissimilarity to a power, say 10, makes the plot more
# informative by effectively changing
# the color palette; setting the diagonal to NA also improves the clarity
# of the plot

# plotDiss = selectTOM^7;
# diag(plotDiss) = NA;
# TOMplot(plotDiss, selectTree, selectColors, main = "Network heatmap plot,
# selected genes")
# rm(TOM)

#5.b Visualizing the network of eigengenes####
# Recalculate module eigengenes
MEs = moduleEigengenes(datExpr, bwmoduleColors)$eigengenes
# Isolate numeric traits from the clinical traits and add the traits to
# existing module eigengenes

MEtraits <- list()
for (trait in 1:length(datTraits)){
tmp <- as.data.frame(datTraits[[trait]])
names(tmp) <- names(datTraits[[trait]])
string = paste("ME network for " ,names(datTraits[[trait]]) ,sep="")
data = orderMEs(cbind(MEs, tmp))
MEtraits[[trait]] = data	names(MEtraits)[[trait]] = string
}

# Plot the relationships among the eigengenes and the trait
par(cex = 0.9)
for (m in 1:length(MEtraits)){
plotEigengeneNetworks(MEtraits[[m]], names(MEtraits)[[m]], marDendro = c
(0.3,1.5), marHeatmap = c(3.4,1,1), cex.lab = 0.8, xLabelsAngle = 90)
export.plot(file.prefix=paste("fig", q.i, ",", an.count, ",", i, ",", questions)[[q.i]], ",", analysis, names(datalist)[i], m, names(MEtraits)[[m]]), exportformats=exportformats.plots,height=height,width=(1 + sqrt(5) )/2*height)
par(parbackup)
}
figure <- figure+1

#6 Exporting network data to network visualization software####
#6.a Exporting to Cytoscape

# Recalculate topological overlap if needed
TOM = TOMsimilarityFromExpr(datExpr, power = thispower);
# Read in the annotation file - #_# not needed
annot = read.csv(file = file.path(dir.results, "GeneAnnotation.csv"));
# Select modules
modules = int Modules
probesM = features[bwnet$blockGenes[[1]]]
# Select module probes
for ( cytmod in modules ){
  inModule = is.finite(match(bwModuleColors[bwnet$blockGenes[[1]]], cytmod));
  modGenes = annot$annotSYM[match(modProbes, annot$features)];
  # Select the corresponding Topological Overlap
  modTOM = TOM[inModule, inModule];
  dimnames(modTOM) = list(modProbes, modProbes)
  # Export the network into edge and node list files Cytoscape can read
  setwd(dir.results)
  exportNetworkToCytoscape(modTOM, edgeFile = paste('CS ',q.i, '.', an.count, '.', i, '\n', names(datalist)[[i]], 'CS-edges-', paste(cytmod, collapse='--'), '.txt', sep=''),
                          nodeFile = paste('CS ',q.i, '.', an.count, '.', i, '\n', names(datalist)[[i]], 'CS-nodes-', paste(cytmod, collapse='--'), '.txt', sep=''),
                          weighted = TRUE,
                          threshold = 0.02,
                          nodeNames = modProbes,
                          altNodeNames = modGenes,
                          nodeAttr = bwmoduleColors[bwnet$blockGenes[[1]]][inModule]);
  rm(modTOM)
  collectGarbage()
  setwd(dir.figures)
}

#7. Plot module heatmap and eigengene expression####

datME = MEs
for ( module in int Modules ){ #( see above )
  # which.module = module
  ME= datME[, paste('ME', which.module, sep='')]  
  # par(mfrow=c(2,1), mar=c(0.3, 5.5, 10, 2))
  # plotDendroAndColors(sampleTree2, traitColors, cex.dendroLabels = 0.5, cex.colorLabels = .5, cex.lab = .7, cex.axis = .7,
  # groupLabels = names(datTraits), saveMar=F)
  # plotMat(t(scale(datExpr[, bwmoduleColors==which.module ][sampleTree2$order,])), nrgcols=30, rlabels=F, clabels=rownames(datExpr)[sampleTree2$order], rcols=which.module, ccols=as.character(classcolours[[1]])[sampleTree2$order], main=which.module, cex.main=1)
  # par(mar=c(4, 3.8, 0, 0.7))
  # barplot(ME[sampleTree2$order], col=which.module, main='', cex.main=2, ylab="eigengene expression", xlab="array sample")
  # export.plot(file.prefix=paste('fig',q.i, '.', an.count, '', i, '', figure, names(questions)[[q.i]], '.'), analyse, names(datalist)[[1]], module, "dend_order"), exportformats=export.format.plots, height=height, width=(1 + sqrt(5))/2*height)
  # par(parbackup)
  # figure<-figure+1
}
for ( module in int Modules ){ #( see above )
  which.module=module
A.8 Analysis: IMPI-MA

ME = data ME[, paste("ME", which.module, sep="")]  
par(mfrow=c(2,1), mar=c(0.3, 5.5, 13, 2))
# plotDendroAndColors(sampleTree2, traitColors, cex.dendroLabels = 0.5, cex.colorLabels = 0.5, cex.lab = 0.7,  
# cex.axis = 0.7,  
# groupLabels = names(datTraits), saveMar=F)
plotMat(t(scale(dataExpr[, bwmoduleColors == which.module ])), nrncols=30,  
# labels=F, nlabels=rownames(dataExpr), labcols=which.module, cclabels=as.character(classcolours[[i]]), main=which.module, cex.main=1)
par(mar=c(4, 3.8, 0, 0.7))
barplot(ME, col=which.module, main="", cex.main=2,  
# lab="eigengene expression", xlab="array sample")
export.plot(file.prefix=paste("fig", q.i, ",", an.count, ",", i, ",", figure,  
# names(questions)[[q.i]], ",", analysisNames(datalist)[i], module, "orig-order"), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
par(parbackup)
}
figure <- figure + 1
par(parbackup)
# boxplots and stats for DE modules
par(parbackup)
for (module in intModules) {  
which.module = module
ME = data ME[, paste("ME", which.module, sep="")]  
boxplot(list(ME[tsstat.class[[i]] == levels(tsstat.class[[i]])[[1]]], ME[  
	tsstat.class[[i]] == levels(tsstat.class[[i]])[[2]]]), col=c("blue", "red"),  
mains=levels(tsstat.class[[i]]))
export.plot(file.prefix=paste("fig", q.i, ",", an.count, ",", i, ",", figure,  
# names(questions)[[q.i]], ",", analysisNames(datalist)[i], module, "boxplot"  
), export.formats=export.formats.plots, height=height, width=(1 + sqrt(5))/2*height)
par(parbackup)
figure <- figure + 1
# stats for comparisons
statlist.modules <- list()
for (mod in 1:length(intModules)) {
which.module = intModules[[mod]]
ME = data ME[, paste("ME", which.module, sep="")]  
statlist.modules[[mod]] <- wilcox.test(  
ME[tsstat.class[[i]] == levels(tsstat.class[[i]])[[1]]], ME[tsstat.class[[i]]  
== levels(tsstat.class[[i]])[[2]]])
$p.value"

# the end####
# wgcna for loop

# clean up
rm(TOM)
Listing A.12: SevenQuestions

```r
collectGarbage()
}

rm(list = as.character(dataset.variables))
collectGarbage()
}

print(paste("End time:", Sys.time()))
```