An overview of -omics landscapes in prostate cancer

Gaetan De Waele and Bryan Verfaillie
Supervisors: Prof. dr. ir. Tim De Meyer, Prof. dr. ir. Jo Vandesompele
Counsellors: Ir. Louis Coussement

Abstract— **Omics related studies produce a lot of data, many of it readily publicly available. The combination of multiple omics data sources and technologies has the potential to reveal new insights that would not be possible with a single omics study. This scientific report tries a basic multiomics approach to prostate cancer using public data. A genomic, epigenomic and two transcriptomic analyses were performed in this report to obtain an overall view of the molecular mechanisms of prostate cancers.**

We show that there is considerable overlap between results of different data sources and reason that leveraging information between different data styles makes sense from a statistical and biological point of view.

Keywords—**Omics, Multiomics, Prostate Cancer, Tumor, RNA-seq, Microarray, Expression Profiling, Methylation Profiling, Comparative Genomic Hybridisation, CNV**

I. Introduction

Cancer is one of the biggest causes of death in the modern world. It is an umbrella term for diseases, characterised by an abnormal growth and division of cells that form tumors. Cancers have been known to change the genomic, transcriptomic and epigenomic landscape of cells drastically, making omics studies vital for understanding the general mechanisms of cancers.

Prostate cancer is the second most common cancer in men, with 1 278 106 new cases diagnosed in 2018 [1]. Thousands of molecular studies have been performed on the topic of prostate cancer; however, integration of data sources remains a challenge. Data integration of omics studies could provide novel insights in the molecular mechanisms of tumorigenesis and progression of prostate tumors [2]. The goal of this scientific report is to lay the foundation for multiomics approaches to prostate cancer analysis by using 4 public datasets. Our hope is to see that the results of the 4 different experiments are consistent with each other and represent the same molecular mechanisms.

A. Data

This report is based on prostate cancer data that can be used to study the difference between cancer tissue and neighbouring healthy tissue.

The first selected dataset contains RNA sequencing data of prostate cancer cells and adjacent normal tissue cells of 14 Chinese patients. The data was sequenced with a single Illumina HiSeq2000 flowcell. Reads below a quality threshold and with common uncalled characters were filtered out, as well as reads coming from bacterial contamination [3].

The second dataset is a gene profiling analysis using Affymetrix human exon 1.0 ST arrays to identify the differentially expressed genes in tumor vs. normal tissue. Thirty prostate biopsy specimens, tumor and adjacent normal tissues, were collected from 15 European American prostate cancer patients [4].

An epigenomic data source profiling the methylome of both healthy and tumor tissue of British prostate cancer patients was used to complement the transcriptomic analyses. The samples were profiled using Infinium HumanMethylation450k BeadChip arrays [5].

Finally, a Comparative Genomic Hybridisation (CGH) microarray of human prostate adenocarcinoma and normal samples of three patients was used to assess Copy Number Variation (CNV) in prostate cancer. The profiles were obtained via the Agilent 244A CGH array. In the original study, the profiles were used to support transcriptional sequencing data on the same samples [6].

The remainder of this paper is structured as follows. The different approaches to analyse the datasets are explained in section II. The results of these analyses are listed in section III. An elaborated discussion of these results is given in section IV. Section V concludes the paper.

II. Methods

This section describes the different methods used to analyse the different omics datasets.

A. RNA-seq Based Expression Analysis

The first expression analysis is performed on the raw counts of the RNA-seq experiment. Quality control of raw reads is already performed in the original research and is outlined in Ren et al. (2012) [3], as well as when the data was introduced. Patients for which only tumor or only healthy tissue data is available were filtered out, so a paired design could be used. Based on the rationale that lowly expressed genes cannot be accurately determined to be differentially expressed, the genes with low counts in most of the samples can be filtered out. After exploratory goodness-of-fit evaluation, differential expression between tumor and adjacent healthy tissue on the remaining genes is tested according to standard *edgeR* guidelines. *EdgeR* employs a model-based normalization approach and tests differential expression via a likelihood-ratio test. P-values are corrected for multiple testing with the Benjamini-Hochberg method to obtain FDR-values. Gene set analysis is

performed with a standard over-representation test to determine which Gene Ontology (GO) terms are over-represented in the set of differentially expressed genes.

B. Microarray Based Gene Expression Analysis

A gene expression microarray is used to determine differentially expressed genes in prostate tumor tissue compared to normal prostate tissue. The *oligo* package is used for background correction, normalization and summarization in one single step. It uses a deconvolution method for background correction, quantile normalization and the Robust Multichip Average (RMA) algorithm for summarising. The batch effect is corrected by a quantile normalization between the arrays. With the *arrayQualityMetrics* package, quality control metrics are calculated before and after preprocessing. Subsequently, differential expression is tested with the *limma* package. Finally, gene set analysis is performed as described in the previous section, but with a more lax cut-off. More details on this subject can be found in the results sections.

C. Methylation Profiling Analysis

Methylation profiling data generated with the Illumina Infinium HumanMethylation450k BeadChip platform is used for an epigenomic study of prostate cancer. Methylation profiles of both tumor and healthy adjacent prostate tissue of 4 patients are used in the analysis. The *minfi* package is used for quality control, preprocessing and normalization of the data. *Limma* is used to identify differentially methylated positions, and *bumphunter* is used to identify differentially methylated regions. Finally, gene set analysis is performed on both differentially methylated positions and regions in the same way as described in previous sections.

D. Copy number variation analysis

Micro-array based Comparative Genomic Hybridisation data is used to get an overview of copy number variations in prostate cancer. Three profiles of healthy tissue and three profiles of tumor tissue containing log2-ratios of the tissue against a diploid reference are analysed. The standard *CGHcall* procedure is followed for preprocessing, normalization, segmentation and calling of CNV in all six samples. *CGHregions*, which employs dimensionality reduction with minimal information loss, is used to obtain CNV calls based on all tumor samples together instead of samples separately.

E. Integration of Data Sources

The results of the above described analyses are compared to find out differences or similarities between the different analyses. This gives an indication of the robustness of the results. Comparison of results consists of evaluating overlap of the results on both gene level and gene set level. For the comparison of the two gene expression based analyses, a scatter plot of the logFCs of both results is also used to evaluate agreement of results. For the sake of comparison, a cutoff of uncorrected p-values < 0.20 is used to call genes significantly differentially expressed for the microarray based analysis.

III. Results

This section will first outline the results of the individual analyses, after which results of data comparison between analyses are shown. Generally, extra observations and results can be found in the appendices, where code and additional comments are presented in a user-friendly Rmarkdown notebook format.

A. RNA-seq Based Expression Analysis

Data exploration showed a predominant grouping of expression profiles by tissue type (either tumor or healthy adjacent tissue). A paired design with a blocking patient effect is used. A good fit to the *edgeR* GLM model is observed with a qq-plot (data not shown, see appendices). Differential expression of genes between tissue types is tested via a likelihood-ratio test, which found 2 569 and 2 813 out of 18 877 genes to be significantly up- and downregulated respectively (Benjamini-Hochberg FDR < 0.05). An MA-plot of the results is shown in figure 1. Top hits are shown in table I.

Table I
TOP HITS FOR DIFFERENTIAL EXPRESSION BETWEEN TUMOR AND HEALTHY TISSUE IN THE RNA-SEQ ANALYSIS.

Gene Symbol	log2FC	FDR
KRT13	-5.93	1.93e-35
CYP4B1	-3.46	1.46e-32
AMACR	4.50	6.32e-31
GATA3	-2.34	2.52e-25
CD177	-5.84	1.26e-23
ARHGEF38	1.99	2.56e-23
APOBEC3C	-2.08	2.56e-23
SCGB1A1	-4.65	4.43e-22
SLC45A2	6.45	1.19e-19
GSTP1	-1.91	1.22e-19

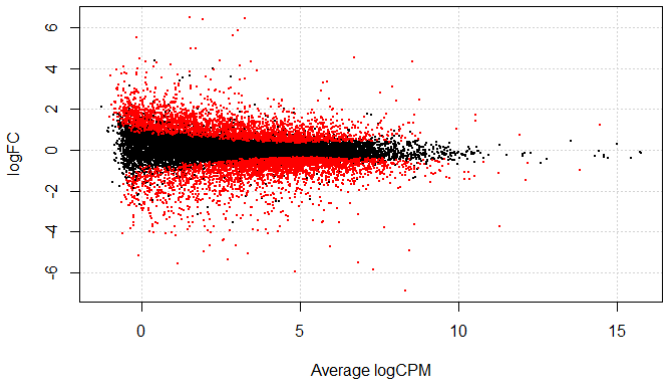


Figure 1. MA-plot of log fold changes between expression values in tumor and healthy adjacent tissue. Genes called significant (with an FDR value $<$ 0.05) are indicated in red.

Gene set analysis is performed on genes called significant via over-representation analysis on GO terms. After Benjamini-Hochberg FDR adjustment, 1 037 GO Biological Process terms are called significantly over-represented in our significant genes. Top hits include: developmental process,

anatomical structure development and tissue development. Further results can be found in appendix A.

B. Microarray Based Expression Analysis

The oligo package is used to preprocess the data. It performs a background correction, normalization and summarization in one single step using a deconvolution method for background correction and quantile normalization, and the RMA (robust multichip average) algorithm for summarising. By setting the target to *core*, the remaining data will only be data on gene level. With normalizeQuantiles, the data can further be normalized to correct for the batch effect. Figure 2 shows the boxplot for the preprocessed data. Before continuing, the probe ID information is replaced by the gene information to make the interpretation easier. The extended code for this analysis can be found in appendix B.

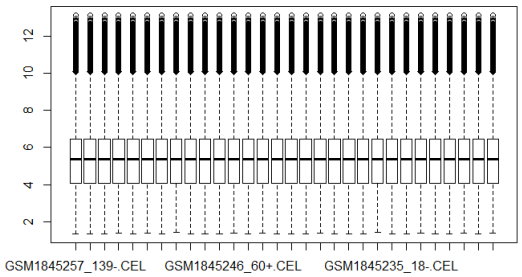


Figure 2. Boxplot showing the data distribution of intensity values of each sample after limma preprocessing and batch effect correction.

To find the differential expression of genes between tumor and normal tissue, *limma* is used. The results obtained by using the *limma* package are shown in Table II. It can be seen that the p-values are quite high which indicates a low power for the test, which is counter-intuitive for a 15vs15 experiment.

Table II
TOP HITS FOR DIFFERENTIAL EXPRESSION BETWEEN TUMOR AND HEALTHY TISSUE IN THE EXPRESSION ARRAY ANALYSIS.

Gene Symbol	log2FC	P-value
PAM	-0.59	2.10E-04
LINC00839	-1.09	4.06E-04
RNF121	0.39	1.13E-03
EDC3	-0.43	1.68E-03
SLC48A1	-0.44	2.26E-03
FBXO5	-0.28	2.30E-03
CSNK1A1	-0.40	2.39E-03
WASF2	0.51	3.20E-03
QSER1	0.46	4.38E-03
RCOR3	0.43	4.64E-03

A visualisation of results can be found in Figure 3. The plot shows the logFC of genes in function of the average log intensity of the gene. This plot shows no significant difference (FDR < 0.05) between gene expressions in healthy vs tumor tissue.

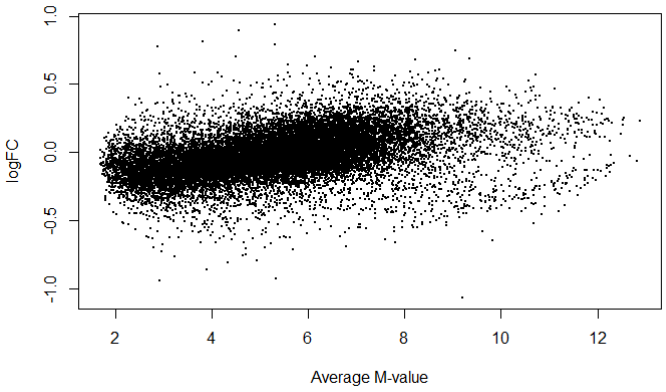


Figure 3. MA-plot of log fold changes between M-values in tumor and healthy adjacent tissue. No significant differences can be found.

C. Methylation Profiling Analysis

Methylation profiles of both tumor and epithelial prostate tissue of four patients are analysed using *minfi*, *limma* and *bumphunter*. Quality control on unprocessed beta values showed no bad quality samples (data not shown). An exploratory MDS plot, shown in figure 4, shows predominant grouping of methylation patterns by patients. Hence, a patient effect is also included in the statistical design as a blocking effect.

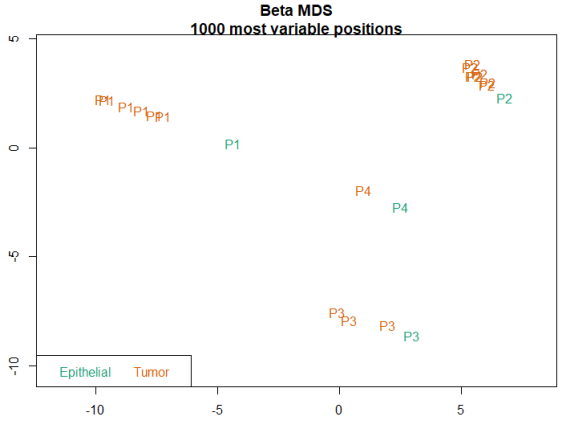


Figure 4. MDS plot of beta-values. Tumor samples are shown in orange and epithelial samples are shown in green. A predominant grouping by patients is observed.

Differential methylation of CpG sites is tested with *limma* after functional normalization and filtering of SNPs. 71 692 (15.32%) CpG probes are found to be significantly (FDR $<$ 0.05) differentially methylated between the two conditions. An MA-plot of the results is shown in Figure 5.

Because the methylation status of individual CpG sites seldom have a big influence, genomic regions are searched where CpG sites have the same differential methylation patterns. For this purpose *bumphunter* is used. A short description of the *bumphunter* algorithm can be found in the appendix

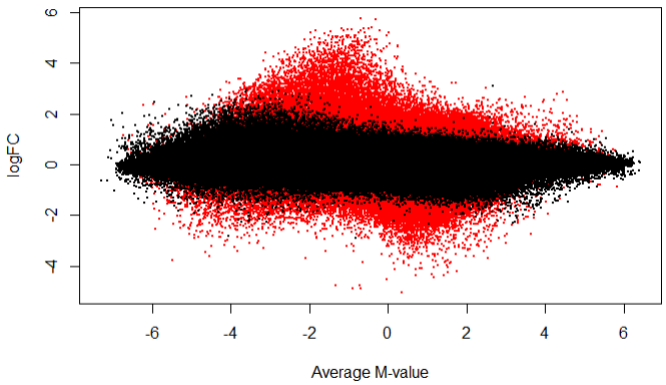


Figure 5. MA-plot of log fold changes between M-values in tumor and healthy adjacent tissue. CpG probes called significant (with an FDR value < 0.05) are indicated in red.

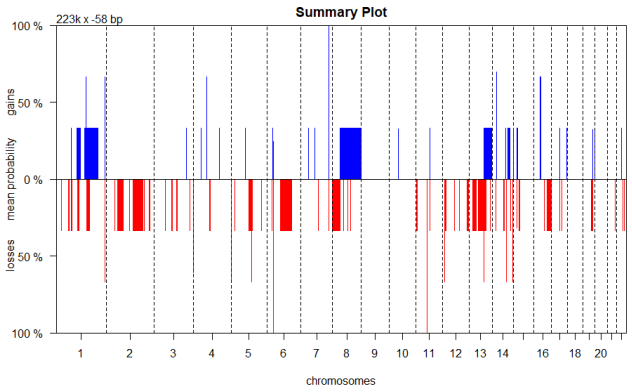


Figure 6. Summary Plot of copy number variation in the three tumor prostate samples. Blue indicates gain of copy number in that genomic region, red indicates loss. The height of the bars represent the average probability that the positions they cover are gained or lost.

C. For the 444 significantly (FWER < 0.10) differentially methylated regions (DMRs), the nearest gene is determined (filtering for regions far away from genes). Some genes for the most significant DMRs are: CYBA, KLF8 and HIF3A. Gene set analysis on genes with a significant DMR in it or nearby is tested via over-representation analysis on GO terms. Resulting p-values are corrected for multiple testing using Benjamini-Hochberg FDR adjustment. Top hits include GO terms that signify responses to growth factors, regulation of cell differentiation and response to endogenous stimulus.

To validate these results, the same gene set analysis procedure was followed with individual CpG sites instead of regions. For each significant CpG site, the nearest gene is determined, filtering for duplicated genes. Top hits for GO Biological Process terms include nervous system development, anatomical structure morphogenesis and developmental process. Of the top 50 gene sets for each gene set analysis, 30 (60%) overlap.

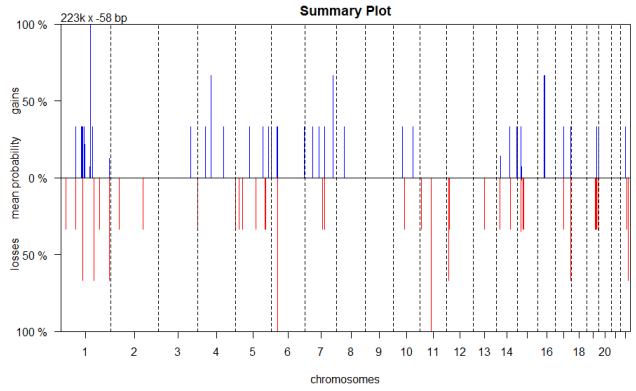


Figure 7. Summary Plot of copy number variation in the three healthy prostate samples. Blue indicates gain of copy number in that genomic region, red indicates loss. The height of the bars represent the average probability that the positions they cover are gained or lost.

D. Copy Number Variation Analysis

Array comparative genomic hybridization data is analysed to obtain a view of copy number variations in tumor tissue. Log2-ratios between either tumor or healthy tissue and a diploid reference are analysed for three tumor and three healthy samples. A procedure following standard *CGHcall* guidelines is followed for preprocessing, normalization and segmentation, as well as calling of CNV. Summary plots for both tumor and healthy prostate samples are shown in figure 6 and 7 respectively.

The *CGHregions* package is used to adjust the segmentation in tumor samples, so that break-points are in similar locations across the three samples. 2885 genes are found in regions for which there is generally copy number variation across the tumor samples. More details and figures can be found in the appendices.

E. Integration of Data Sources

First of all, a comparison of the two expression based analyses is performed. Because of the lower power of the microarray based analysis and for the sake of comparison, a cut-off of uncorrected p-values < 0.20 is used to call genes significant in the microarray based analysis. A scatter plot of the log fold changes of genes in both analyses can be found in figure 8. 52.25% of all genes have the same sign in their log fold changes. Of the 3 318 and 2 331 significant genes in the RNAseq and microarray experiment, 665 genes overlap.

An overlap of results is also evaluated between the two gene expression studies and the methylation study. Overlap of differential methylation results is evaluated for both differentially methylated CpG sites and differentially methylated regions. Since differentially methylated regions gave the best overlap with differential expression results, plus makes more sense biologically, only these results are shown (more data shown in appendices). A Venn diagram of overlapping genes can be found in figure 9. 22 genes are found to be significant in all analyses. Of these common genes, top hits include: SOX7, AOX1, CAV2, CPA6 and ALPL.

Additionally, an overlap including the genes lying in regions where CNV was called is also performed. A clear overlap (428

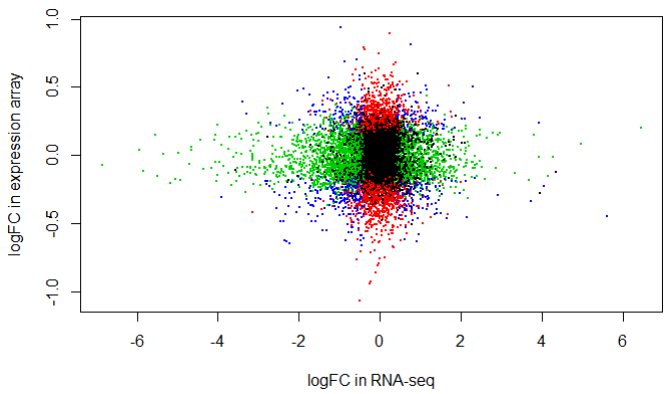


Figure 8. Scatterplot of logFCs in both analyses. For the expression array a significance cut-off of $p<0.20$ was used, whereas for the RNA-seq analysis FDR<0.05 was used. Red and green dots indicate significant genes in the expression array and RNA-seq, respectively. Blue dots indicate significant genes in both analyses.

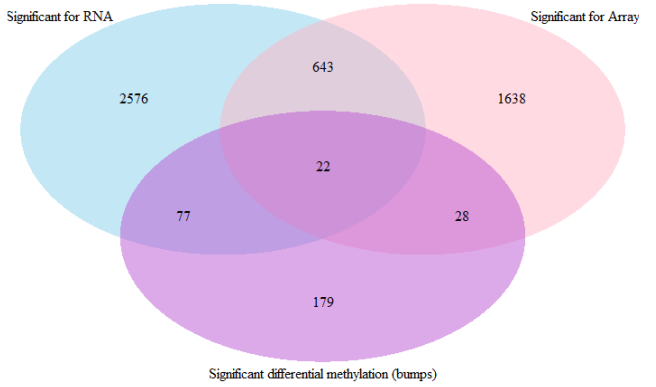


Figure 9. Venn diagram showing overlap of significant genes found to be differentially methylated or differentially expressed according to three analyses.

out of 3 318 and 2 777 genes) with differentially expressed genes is observed, indicating the need for further evaluation. More data is shown in the appendices.

Finally, an overlap between the 100 top GO Biological Process terms is performed for the first three analyses. 15 terms were found to be overlapping between the three. Of these common terms, top hits include: developmental process, anatomical structure development and regulation of cellular process.

IV. Discussion

This section will first discuss the observations and remarks on the generated results. Near the end, a more general discussion of possible data integration for omics studies approaches is presented.

A. Discussion of results

Differential expression of genes was analysed using data coming from two platforms. Since these platforms detect different types of signals, RNAseq uses detection of reads whereas microarrays quantify fluorescent signals, a direct comparison of signals is not possible. However, in ideal conditions, log fold changes for both analyses should follow a trend, e.g. if a gene has a low logFC in one analysis, it is expected to have a low logFC in another analysis as well. Our results, visualised in figure 8, do not show this trend. The ability to compare results in this case is heavily limited because of the use of different patients from different ethnic groups. Tumors are naturally diverse, and it is to be expected that they are even more diverse for individuals whose genetic features are substantially different from each other, like in different ethnic groups. However, our estimation is that not only the effect of different individuals hampers the similarity of results, but also the cleanliness of data. This suggestion is supported by the fact that no significant genes were found in the expression microarray based analysis. Perhaps some errors occurred in sample preparation or fluorescent signal detection, but this is unlikely as quality control of raw data showed normal observations. As authors, we leave this as an open question to the reader.

Differential methylation analysis was performed both for individual CpG sites and for regions comprising multiple CpG sites. Analysis based on regions showed more overlap with differential expression analysis results, and makes more sense biologically as methylation of individual CpGs don't have a big impact. Copy number variation analysis showed more aberrant copy numbers in tumor tissue as compared to healthy tissue, as expected from a tumor. Also here, a design where the same patients are used for CNV analysis and differential methylation analysis, as for differential expression analysis, have the potential to complement each other better and makes more sophisticated integrative genomics methods possible. A very simple example of data integration in the case of CNV analysis, is that one can correct signals or counts of expressed genes for the copy number that is called for those genes. This is not implemented here because of the potentially low comparability between the samples used in CNV analysis and differential gene expression analysis. However, considerable overlap between differentially expressed genes and genes which lie in regions where copy number variation was called, indicates that this suggestion needs more evaluation. Examples and strengths of more complicated integrative genomics methods are discussed in the last section of the discussion.

An evaluation of gene set analysis results is also important. Gene sets for both differential expression analyses were overlapping considerably. Also a considerable overlap was found with over-represented gene sets for differential methylation results, with 15 of the top 100 GO Biological Process terms overlapping between the three results. A critical remark on these results is that a lot of the top over-represented gene sets are very general GO terms. Because general GO terms contain more genes, it improves the power of the Fisher's exact test used in over-representation analysis. This way, it is to be expected that some overlap is found between the different results, as some of these are very general GO Biological Process terms, such as developmental process. A way around

this could be to use an arbitrary cut-off for gene sets with a maximal allowed number of genes in it. One could also try more sophisticated gene set analysis procedures such as Gene Set Enrichment Analysis [7].

Finally, some interesting genes, that are overlapping for both differential expression and differential methylation analyses, were evaluated on existing knowledge of their function in cancer. Top genes that were found significant in the three analyses are SOX7, AOX1, CAV2, CPA6 and ALPL. SOX7 is suggested to be a tumor suppressor gene by suppressing beta-catenin-mediated transcriptional activity, a protein that is often found to be mutated in cancers. Its inactivation is suggested to promote the development of approximately half of prostate tumors [8, 9]. Hypermethylation and downregulation of AOX1 is in concordance with previous studies; however, no studies have been published on its role in cancers [10]. CAV2 codes for a caveolae-associated protein, found to be differentially expressed in prostate cancer progression [11]. Our results show that CAV2 is downregulated, contradicting previous observations reported in literature. CPA6 codes for a peptidase, not previously linked with tumor progression. Finally, ALPL expression has been shown to contribute to the pathogenesis of prostate cancer progression [12]. Again, our results contradict this study, as we have found ALPL to be downregulated.

Top biological processes that were overlapping for both differential expression and differential methylation analyses include GO terms that relate to developmental processes. This observation is not out of the ordinary, since tumors reprogram cells to be very active. Also, both up- and downregulation of cellular processes is also found to be one of the top terms. tumors change the transcriptional landscape of cells drastically to progress and develop, e.g. inducing angiogenesis and activating metastasis.

B. Multi-omics approaches

The currently used methods for data comparison were rather simple as each dataset is analysed separately, not leveraging information of other data sources to make statistical inference. A real multiomics approach is more complex, but is required to draw a more complete picture and to account better for the complexity of biological systems. These approaches could provide more meaningful results, leading to potential novel insights in the molecular mechanisms of the disease. Analysis of only one data type is limited to correlations, mostly reflecting reactive processes rather than causative ones. Integration of different omics data types is often used to elucidate potential causative changes that lead to disease, or the treatment targets, that can be then tested in further molecular studies [13]. Combining this omics data requires some linkage, so preferably all samples come from the same patients. If this is not possible a gene based approach can be used. However, Palson et al. describe many remaining challenges in multi-omics analysis [14]. Bersanelli et al. describe different types of data integration methods of multi-omics, both network-based, network-free or bayesian or non-bayesian methods [15]. Network-based methods use graphs for modeling and analysing relationships among variables. Multiple omics data can be naturally embedded in a heterogeneous network framework, where different layers in the network are modelled.

V. Summary & Conclusion

In conclusion, this report provides an overview of -omics landscapes in prostate cancer. We show that there is considerable overlap between results of different data sources and reason that leveraging information between different data styles makes sense from a statistical and biological point of view. The heterogeneity of prostate cancers in different individuals griefly affects similarity of obtained results, so an in-depth integration of data sources on the used data is not recommended. However, this report provides an example of the foundations of using multiple omics data sources to make new biological discoveries. With more data and better experimental designs, we reason that integration of multiple data sources can lead to better results.

VI. Acknowledgement

The authors would like to thank the University of Ghent and in special the teachers and supervisors from the course *Applied High-Throughput Analysis* in the Bioinformatics track for their support and advice during the writing of this paper.

VII. References

[1] Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. "Cancer statistics, 2018". In: *CA: A Cancer Journal for Clinicians* 68.1 (2018), pp. 7–30. DOI: 10.3322/caac.21442. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3322/caac.21442. URL: https://onlinelibrary.wiley.com/doi/abs/10.3322/caac.21442.

[2] Greg Gibson. *A Primer of Human Genetics*. Oxford University Press Inc, 2015. ISBN: 978-1-60535-313-5.

[3] Shancheng Ren et al. "RNA-seq analysis of prostate cancer in the Chinese population identifies recurrent gene fusions, cancer-associated long noncoding RNAs and aberrant alternative splicings". In: *Cell Research* 22.5 (May 2012), pp. 806–821. ISSN: 1001-0602. DOI: 10.1038/cr.2012.30. URL: http://www.nature.com/articles/cr201230.

[4] B.-D. Wang et al. "Identification and Functional Validation of Reciprocal microRNA-mRNA Pairings in African American Prostate Cancer Disparities". In: *Clinical Cancer Research* 21.21 (Nov. 2015), pp. 4970–4984. ISSN: 1078-0432. DOI: 10.1158/1078-0432.CCR-14-1566. URL: http://clincancerres.aacrjournals.org/cgi/doi/10.1158/1078-0432.CCR-14-1566.

[5] Charles E Massie et al. "HES5 silencing is an early and recurrent change in prostate tumourigenesis". In: *Endocrine-Related Cancer* 22.2 (2015), pp. 131 –144. URL: https://erc.bioscientifica.com/view/journals/erc/22/2/131.xml.

[6] Serban Nacu et al. “Deep RNA sequencing analysis of readthrough gene fusions in human prostate adenocarcinoma and reference samples”. In: *BMC medical genomics* 4 (Jan. 2011), p. 11. DOI: 10.1186/1755-8794-4-11.

[7] Aravind Subramanian et al. “Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles”. In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550. ISSN: 0027-8424. DOI: 10.1073/pnas.0506580102. eprint: https://www.pnas.org/content/102/43/15545.full.pdf. URL: https://www.pnas.org/content/102/43/15545.

[8] Yu Zhang et al. “SOX7, down-regulated in colorectal cancer, induces apoptosis and inhibits proliferation of colorectal cancer cells”. In: *Cancer Letters* 277.1 (2009), pp. 29–37. ISSN: 0304-3835. DOI: 10.1016/J.CANLET.2008.11.014. URL: https://www.sciencedirect.com/science/article/pii/S0304383508008938.

[9] Lizheng Guo et al. “Sox7 Is an Independent Checkpoint for -Catenin Function in Prostate and Colon Epithelial Cells”. In: *Molecular Cancer Research* 6.9 (2008), pp. 1421–1430. ISSN: 1541-7786. DOI: 10.1158/1541-7786.MCR-07-2175. eprint: http://mcr.aacrjournals.org/content/6/9/1421.full.pdf. URL: http://mcr.aacrjournals.org/content/6/9/1421.

[10] Lokman Varisli. “Identification of New Genes Down-regulated in Prostate Cancer and Investigation of Their Effects on Prognosis”. In: *Genetic Testing and Molecular Biomarkers* 17.7 (2013). PMID: 23621580, pp. 562–566. DOI: 10.1089/gtmb.2012.0524. eprint: https://doi.org/10.1089/gtmb.2012.0524. URL: https://doi.org/10.1089/gtmb.2012.0524.

[11] M.L. Gould, G. Williams, and H.D. Nicholson. “Changes in caveolae, caveolin, and polymerase 1 and transcript release factor (PTRF) expression in prostate cancer progression”. In: *The Prostate* 70.15 (2010), pp. 1609–1621. DOI: 10.1002/pros.21195. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pros.21195. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/pros.21195.

[12] S R Rao et al. “Tumour-derived alkaline phosphatase regulates tumour growth, epithelial plasticity and disease-free survival in metastatic prostate cancer”. In: *British Journal of Cancer* 116.2 (2017), pp. 227–236. ISSN: 0007-0920. DOI: 10.1038/bjc.2016.402. URL: http://www.nature.com/articles/bjc2016402.

[13] Yehudit Hasin, Marcus Seldin, and Aldons Lusis. “Multi-omics approaches to disease”. In: *Genome Biology* 18.1 (May 2017), p. 83. ISSN: 1474-760X. DOI: 10.1186/s13059-017-1215-1. URL: https://doi.org/10.1186/s13059-017-1215-1.

[14] Bernhard Palsson and Karsten Zengler. “The challenges of integrating multi-omic data sets”. In: *Nature chemical biology* 6 (Nov. 2010), pp. 787–9. DOI: 10.1038/nchembio.462.

[15] Matteo Bersanelli et al. “Methods for the integration of multi-omics data: mathematical aspects”. In: *BMC Bioinformatics* 17.2 (Jan. 2016), S15. ISSN: 1471-2105. DOI: 10.1186/s12859-015-0857-9. URL: https://doi.org/10.1186/s12859-015-0857-9.

APPENDIX A
EXPRESSION ARRAY ANALYSIS

This appendix shows the R markdown code for the RNA sequencing analysis.

RNA-seq data of tumor and adjacent healthy prostate tissue

This analysis tries to identify differentially expressed genes between tumor tissue and paired normal tissue in prostates of 9 Chinese males. Here, the analysis starts from raw counts, already mapped to a genome.

Analysis

Preprocessing and normalisation

Load in necessary packages:

```
library(edgeR)
```

```
library(plyr)
```

Load in counts data.

```
# read in data
counts <- read.table(file = "E-MTAB-567-raw-counts.tsv", sep = '\t', header = TRUE, quote="")

# save gene IDs en gene names in separate variable
geneIDs <- counts[,1:2]

# remove gene names from data
counts$Gene.Name <- NULL

# set gene ID as rownames and remove column from data.
rownames(counts) <- as.character(counts$Gene.ID)
counts$Gene.ID <- NULL

dim(counts)
```

```
## [1] 65217    23
```

```
head(geneIDs)
```

Gene.ID	Gene.Name
ENSG00000000003	TSPAN6
ENSG00000000005	TNMD
ENSG00000000419	DPM1
ENSG00000000457	SCYL3
ENSG00000000460	C1orf112
ENSG00000000938	FGR

Load in sample data:

```
sample_data <- read.table(file = "sample_data_relation.txt", sep = '\t', header = TRUE, quote="")
colnames(sample_data)
```

```
##  [1] "Source.Name"                    "Characteristics.organism."
##  [3] "Characteristics.individual."    "Characteristics.age."
##  [5] "Unit.time.unit."                "Characteristics.sex."
##  [7] "Characteristics.ethnic.group."  "Characteristics.organism.part."
##  [9] "Characteristics.disease."       "Characteristics.sampling.site."
## [11] "Material.Type"                  "Protocol.REF"
## [13] "Protocol.REF.1"                 "Protocol.REF.2"
```

```
## [15] "Extract.Name"                      "Comment.LIBRARY_LAYOUT."
## [17] "Comment.LIBRARY_SOURCE."           "Comment.LIBRARY_STRATEGY."
## [19] "Comment.LIBRARY_SELECTION."        "Comment.ORIENTATION."
## [21] "Comment.NOMINAL.LENGTH."           "Comment.NOMINAL_SDEV."
## [23] "Protocol.REF.3"                    "Performer"
## [25] "Assay.Name"                        "Technology.Type"
## [27] "COMMENT.SEQUENCE_LENGTH."          "COMMENT.SPOT_LENGTH."
## [29] "Comment.READ_INDEX_0_READ_TYPE."   "Comment.READ_INDEX_0_READ_CLASS."
## [31] "Comment.READ_INDEX_0_BASE_COORD." "Comment.READ_INDEX_1_READ_TYPE."
## [33] "Comment.READ_INDEX_1_READ_CLASS." "Comment.READ_INDEX_1_BASE_COORD."
## [35] "Comment.ENA_SAMPLE."               "Scan.Name"
## [37] "Comment.ENA_RUN."                  "Comment.FASTQ_URI."
## [39] "Comment.SUBMITTED_FILE_NAME."      "FactorValue..sampling.site."
```

```
dim(sample_data)
```

```
## [1] 56 40
```

```
# for every sample, there are two rows, we keep every odd row:
keep <- seq(0, nrow(sample_data)-1, 2)+1
sample_data <- sample_data[keep,]

# reorder sample_data to be in the same order as the columns of the counts
sample_data <- sample_data[which(as.character(sample_data$Comment.ENA_RUN.) %in% colnames(counts)),]
```

Create factor variables: we are interested in the tumor vs tissue effect, but also include the patient effect as a blocking factor.

```
individual <- as.factor(sample_data$Characteristics.individual.)
tissue <- sample_data$Characteristics.sampling.site.

# rename factor: H = healthy, T = tumor
tissue <- mapvalues(tissue, from = levels(tissue), to = c('H', 'T'))

interaction(individual, tissue)
```

```
##  [1] 10.H 10.T 11.H 12.T 13.H 13.T 14.H 14.T 1.H  1.T  2.H  2.T  3.H  3.T
## [15] 4.H  5.H  6.T  7.H  7.T  8.H  8.T  9.H  9.T
## 28 Levels: 1.H 2.H 3.H 4.H 5.H 6.H 7.H 8.H 9.H 10.H 11.H 12.H ... 14.T
```

From this we deduct that there is no tumor data for individual 4, 5, 11, and no healthy tissue data for individual 6 & 12. We throw these individuals out of our analysis and keep the other 9 individuals so we can keep a paired design, since losing a couple samples is preferable to giving up a paired design.

```
keep_ind <- table(individual) == 2 # keep individuals for which we have two samples: tumor & healthy
names(keep_ind)[keep_ind == TRUE] # keep these individuals
```

```
## [1] "1"  "2"  "3"  "7"  "8"  "9"  "10" "13" "14"
```

```
# filter sample_data & counts
sample_data <- sample_data[which(as.character(sample_data$Characteristics.individual.) %in%
                           names(keep_ind)[keep_ind == TRUE]),]
counts <- counts[, which(colnames(counts) %in% as.character(sample_data$Comment.ENA_RUN.))]
```

Make new factor variables based on new filtered data and construct a design matrix

```
individual <- as.factor(sample_data$Characteristics.individual.)
tissue <- sample_data$Characteristics.sampling.site.

#rename factor: H = healthy, T = tumor
tissue <- mapvalues(tissue, from = levels(tissue), to = c('H', 'T'))
```

```
design <- model.matrix(~individual+tissue)
design
```

```
##    (Intercept) individual2 individual3 individual7 individual8 individual9
## 1            1           0           0           0           0           0
## 2            1           0           0           0           0           0
## 3            1           0           0           0           0           0
## 4            1           0           0           0           0           0
## 5            1           0           0           0           0           0
## 6            1           0           0           0           0           0
## 7            1           0           0           0           0           0
## 8            1           0           0           0           0           0
## 9            1           1           0           0           0           0
## 10           1           1           0           0           0           0
## 11           1           0           1           0           0           0
## 12           1           0           1           0           0           0
## 13           1           0           0           1           0           0
## 14           1           0           0           1           0           0
## 15           1           0           0           0           1           0
## 16           1           0           0           0           1           0
## 17           1           0           0           0           0           1
## 18           1           0           0           0           0           1
##    individual10 individual13 individual14 tissueT
## 1             1            0           0       0
## 2             1            0           0       1
## 3             0            1           0       0
## 4             0            1           0       1
## 5             0            0           1       0
## 6             0            0           1       1
## 7             0            0           0       0
## 8             0            0           0       1
## 9             0            0           0       0
## 10            0            0           0       1
## 11            0            0           0       0
## 12            0            0           0       1
## 13            0            0           0       0
## 14            0            0           0       1
## 15            0            0           0       0
## 16            0            0           0       1
## 17            0            0           0       0
## 18            0            0           0       1
## attr(,"assign")
##  [1] 0 1 1 1 1 1 1 1 1 2
## attr(,"contrasts")
## attr(,"contrasts")$individual
## [1] "contr.treatment"
##
## attr(,"contrasts")$tissue
## [1] "contr.treatment"
```

Load in a DGEList object:

```
y <- DGEList(counts=counts)
y$sample
```

	group	lib.size	norm.factors

	group	lib.size	norm.factors
ERR031017	1	21545673	1
ERR031018	1	17140929	1
ERR031023	1	17160543	1
ERR031024	1	15878416	1
ERR031025	1	20052038	1
ERR031026	1	20029530	1
ERR031027	1	18555506	1
ERR031028	1	14665900	1
ERR031029	1	16001408	1
ERR031030	1	14625099	1
ERR031031	1	19826227	1
ERR031032	1	19899141	1
ERR031039	1	17885346	1
ERR031040	1	16482318	1
ERR031041	1	20180536	1
ERR031042	1	21749663	1
ERR031043	1	21092844	1
ERR031044	1	19166925	1

Filtering low counts genes out of data:

```
keep <- rowSums(cpm(y)>1) >= 4 # keep genes with more than 1 count per million for at least 4 samples

table(keep)
```

```
## keep
## FALSE  TRUE
## 46340 18877
```

```
y <- y[keep, , keep.lib.sizes=FALSE]
```

Calculating norm factors

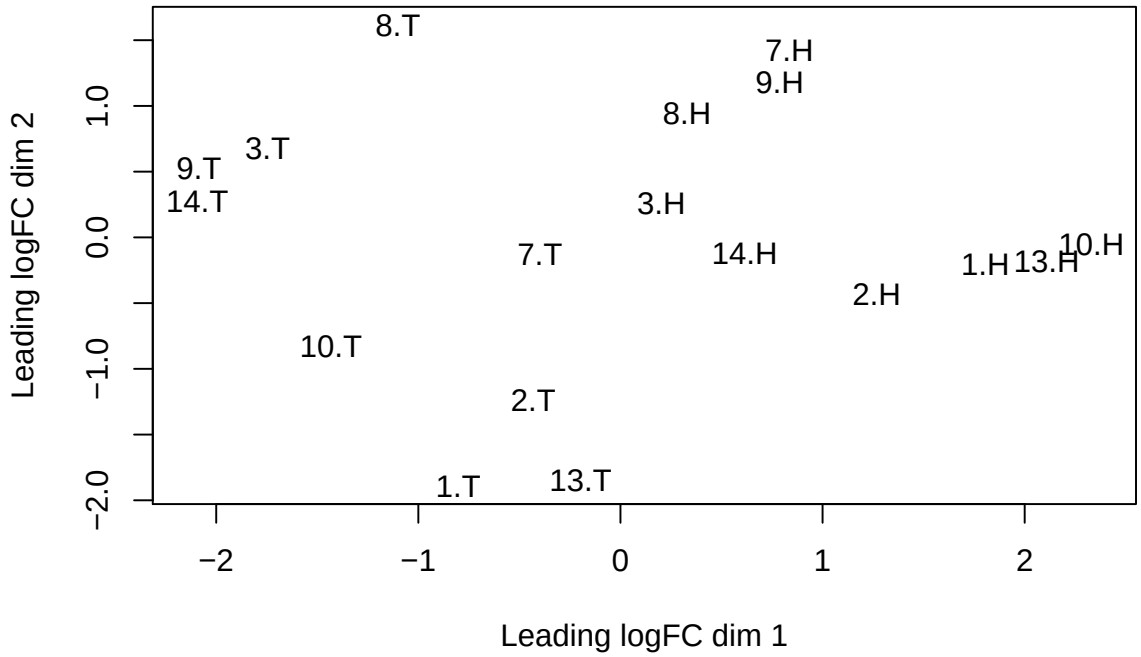
```
y <- calcNormFactors(y)
y$samples
```

	group	lib.size	norm.factors
ERR031017	1	21519703	0.4692668
ERR031018	1	17071657	0.8826677
ERR031023	1	17128237	0.8568919
ERR031024	1	15812852	1.1698495
ERR031025	1	19984612	1.2720499
ERR031026	1	19975278	1.0382591
ERR031027	1	18519963	0.8087744
ERR031028	1	14590216	1.1230212
ERR031029	1	15962097	1.1026694
ERR031030	1	14571493	1.2660051
ERR031031	1	19756855	1.3459756
ERR031032	1	19821137	1.1957800
ERR031039	1	17861919	0.8820501
ERR031040	1	16429391	1.1857422
ERR031041	1	20131064	1.1142277
ERR031042	1	21678620	0.8699901

	group	lib.size	norm.factors
ERR031043	1	21050605	0.9777549
ERR031044	1	19124804	0.9014273

Make an MDS plot, this plot tells you the grouping of samples. From the plot a clear separation between healthy and tumor samples is apparent.

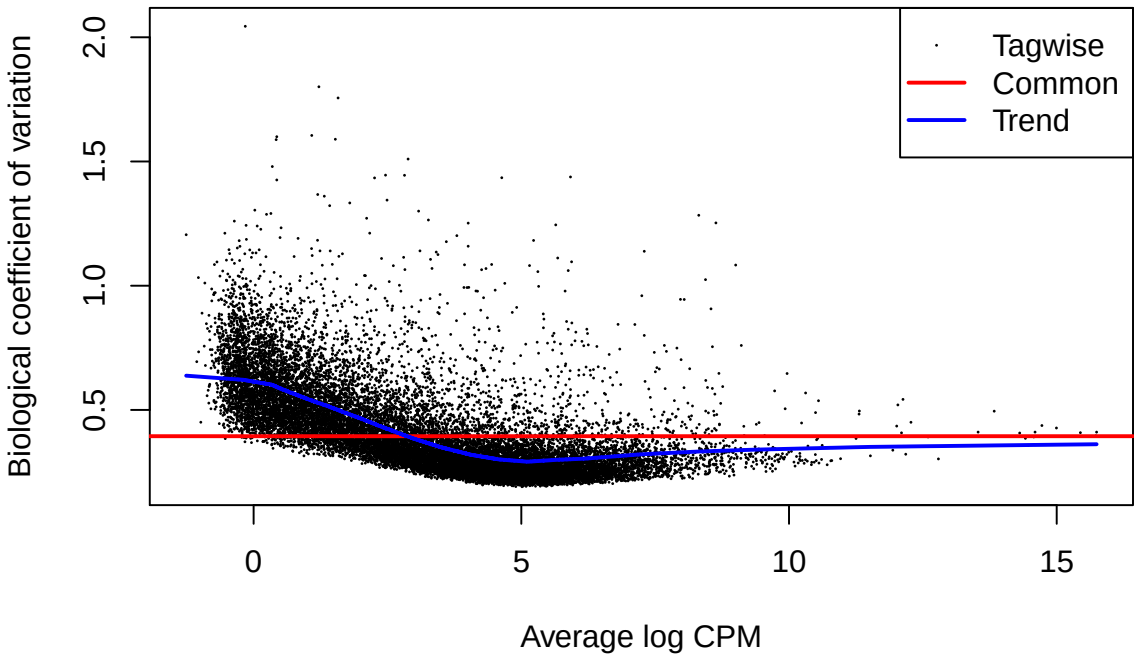
```
colnames(y) <- as.character(interaction(individual,tissue))
plotMDS(y)
```



EdgeR differential expression analysis

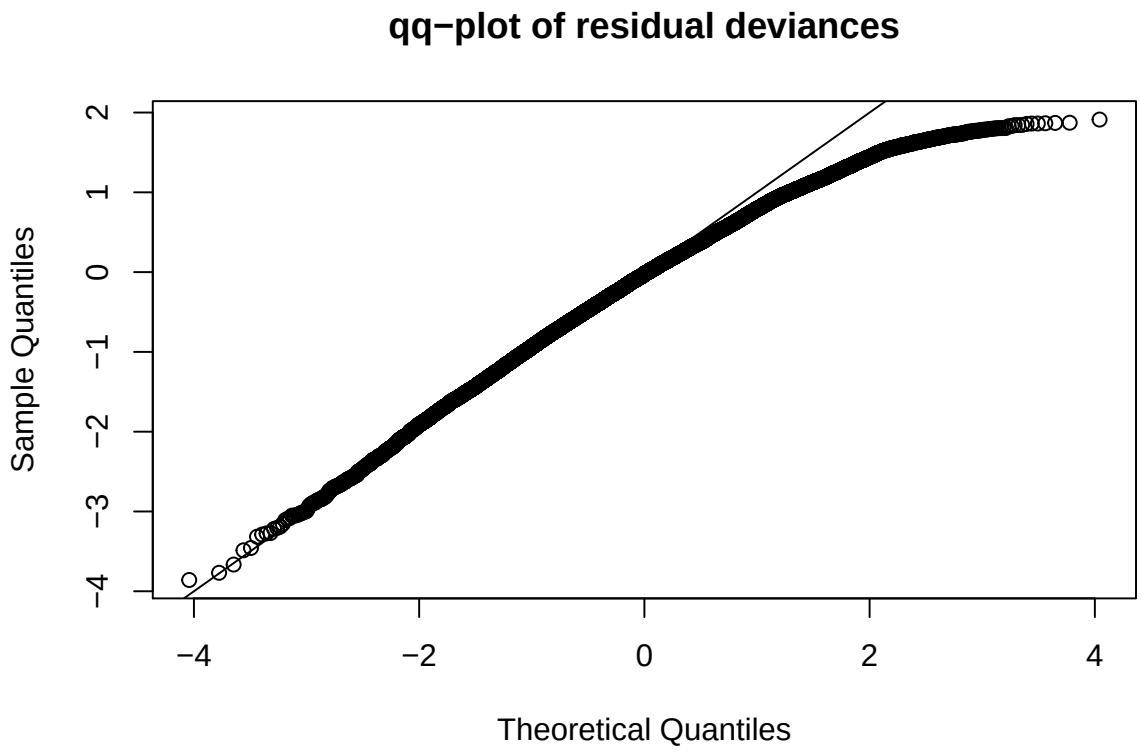
Estimating dispersion and plotting BCV

```
y <- estimateDisp(y, design, robust=TRUE)
plotBCV(y)
```



Fitting model and making a qq-plot to evaluate goodness of fit.

```
fit = glmFit(y,design)
gof(fit,plot=TRUE)
```



Specify contrasts and test differential expression via a likelihood ratio test

```
LRT <- glmLRT(fit)
topTags(LRT)
```

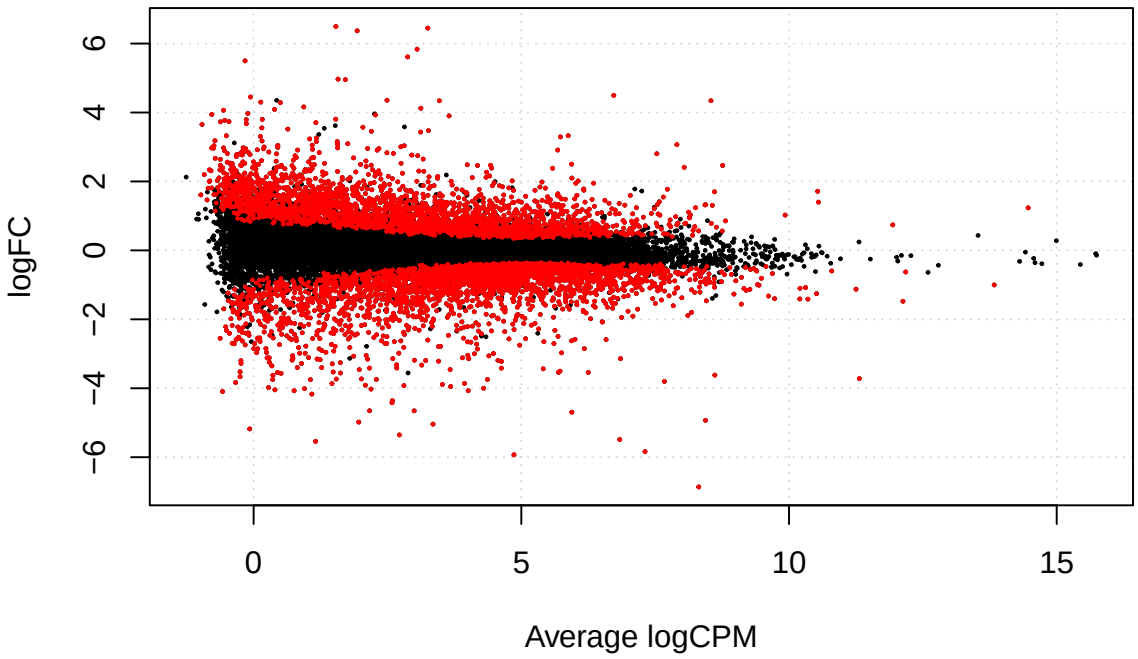
```
## Coefficient:  tissueT
##                     logFC   logCPM        LR       PValue          FDR
## ENSG00000171401 -5.929711 4.863601 173.93834 1.021057e-39 1.927450e-35
## ENSG00000142973 -3.457102 3.670703 159.37568 1.549019e-36 1.462042e-32
## ENSG00000242110  4.496585 6.727269 151.08412 1.004622e-34 6.321418e-31
## ENSG00000107485 -2.335815 4.442908 124.90336 5.343452e-29 2.521708e-25
## ENSG00000204936 -5.836702 7.311235 116.70902 3.324310e-27 1.255060e-23
## ENSG00000236699  1.982991 5.385785 114.73687 8.986461e-27 2.561148e-23
## ENSG00000244509 -2.076255 6.516271 114.62723 9.497293e-27 2.561148e-23
## ENSG00000149021 -4.653921 2.999578 108.71222 1.876445e-25 4.427707e-22
## ENSG00000164175  6.447446 3.253974  97.39789 5.670635e-23 1.189384e-19
## ENSG00000084207 -1.908923 7.084655  97.13184 6.486137e-23 1.224388e-19
```

```
summary(dt <- decideTestsDGE(LRT))
```

```
##        tissueT
## Down      2813
## NotSig   13495
## Up        2569
```

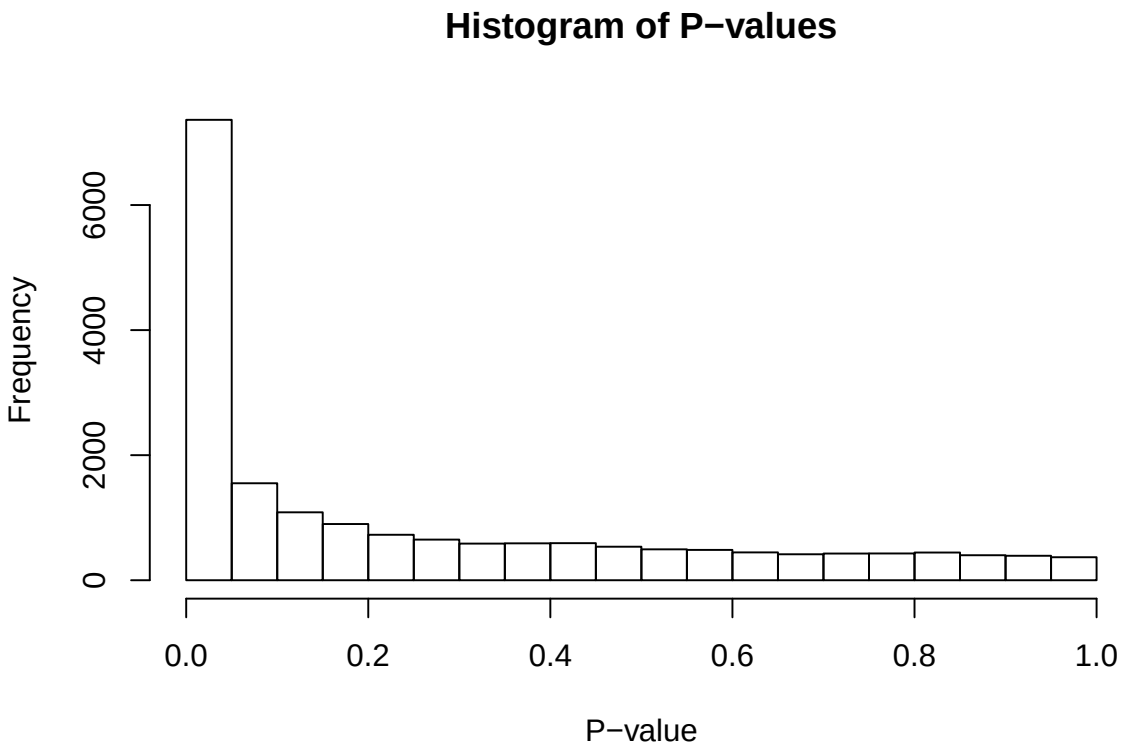
MA-plot:

```
plotSmear(LRT,de.tags=rownames(y)[as.logical(dt)])
```

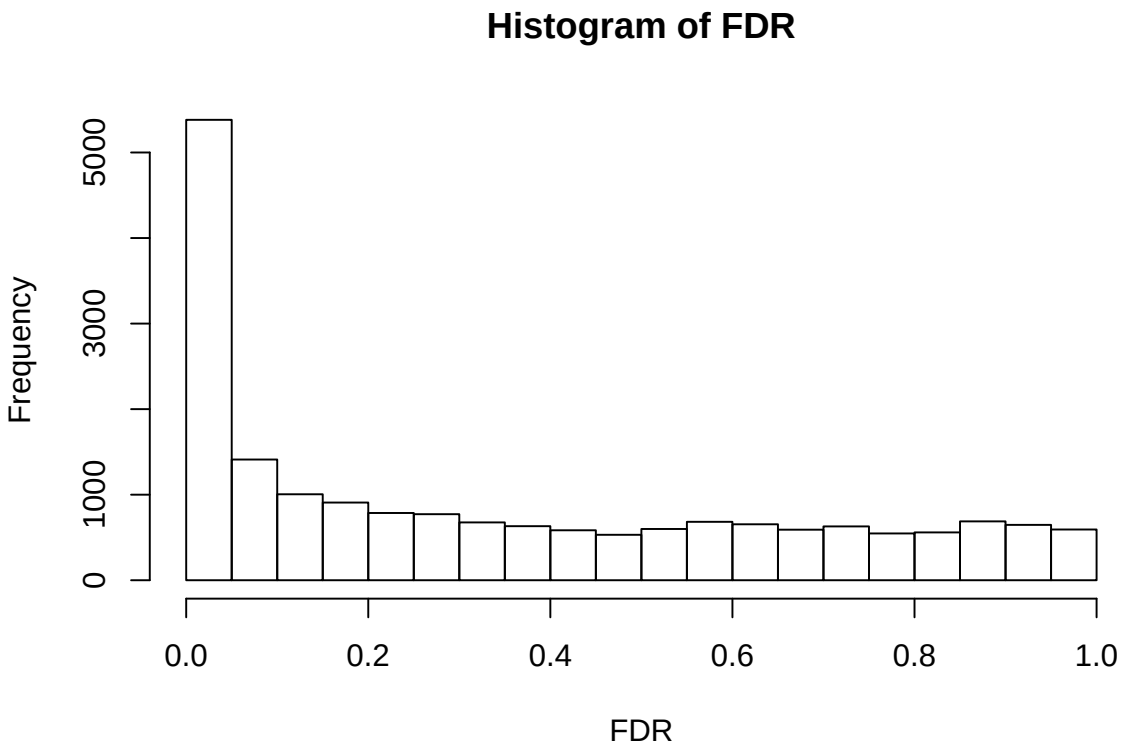


Plotting a histogram of p-values & FDR-values

```
hist(LRT$table$PValue, xlab="P-value", main="Histogram of P-values")
```




```
FDR <- p.adjust(LRT$table$PValue, "fdr")
hist(FDR)
```



Creating a subset of the results with only significant genes

```
LRT$table$FDR <- FDR
sign_genes <- LRT$table[FDR<0.05,]
sign_genes <- sign_genes[order(sign_genes$FDR),]
```

Getting topgenes with their gene symbol instead of ensembl ID

```
library('org.Hs.eg.db')
```

```
topHits <- LRT$table[order(LRT$table$FDR, decreasing=F),]
GeneSymbols <- mapIds(org.Hs.eg.db, rownames(topHits), 'SYMBOL', 'ENSEMBL')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
topHits <- topHits[1:10,c("logFC", "FDR")]
rownames(topHits) <- GeneSymbols[1:10]

topHits
```

	logFC	FDR
KRT13	-5.929711	0
CYP4B1	-3.457101	0
AMACR	4.496585	0
GATA3	-2.335815	0
CD177	-5.836702	0
ARHGEF38	1.982991	0
APOBEC3C	-2.076255	0
SCGB1A1	-4.653921	0
SLC45A2	6.447446	0

	logFC	FDR
GSTP1	-1.908923	0

Gene Set Analysis

Goana uses Entrez gene identifiers, we need to convert our ensemble gene ids to entrez ids. For this purpose we use the org.Hs.eg.db package:

```
EntrezIDs <- mapIds(org.Hs.eg.db, rownames(sign_genes), 'ENTREZID', 'ENSEMBL')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
sign_genes_entrez <- sign_genes[!(duplicated(EntrezIDs) | is.na(EntrezIDs)),]

#make rownames the Entrez gene ID
rownames(sign_genes_entrez) <- EntrezIDs[!(duplicated(EntrezIDs) | is.na(EntrezIDs))]
```

Overrepresentation analysis with goana:

```
library(limma)
goanaOut <- goana(de=rownames(sign_genes_entrez), species="Hs", trend=T)
```

FDR multiple testing adjustment:

```
goanaOut <- goanaOut[order(goanaOut$P.DE, decreasing=FALSE),]
goanaOut$FDR.DE <- p.adjust(goanaOut$P.DE, method="BH")
```

```
topGORNA <- topGO(goanaOut, ontology="BP", number=50)
topGORNA
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0048856	anatomical structure development	BP	5836	1485	0	0
GO:0032502	developmental process	BP	6266	1575	0	0
GO:0007275	multicellular organism development	BP	5351	1368	0	0
GO:0009653	anatomical structure morphogenesis	BP	2656	746	0	0
GO:0009888	tissue development	BP	1968	578	0	0
GO:0048731	system development	BP	4760	1227	0	0
GO:0007399	nervous system development	BP	2296	649	0	0
GO:0022610	biological adhesion	BP	1360	419	0	0
GO:0007155	cell adhesion	BP	1352	416	0	0
GO:0030029	actin filament-based process	BP	734	251	0	0
GO:0007010	cytoskeleton organization	BP	1306	397	0	0
GO:0048869	cellular developmental process	BP	4322	1106	0	0
GO:0030154	cell differentiation	BP	4123	1061	0	0
GO:0048522	positive regulation of cellular process	BP	5142	1287	0	0
GO:0016043	cellular component organization	BP	6269	1535	0	0
GO:0022008	neurogenesis	BP	1536	450	0	0
GO:0042127	regulation of cell proliferation	BP	1668	481	0	0
GO:0048513	animal organ development	BP	3428	889	0	0
GO:0008283	cell proliferation	BP	2077	574	0	0
GO:0048468	cell development	BP	2045	565	0	0
GO:0030036	actin cytoskeleton organization	BP	642	216	0	0
GO:0060429	epithelium development	BP	1264	375	0	0
GO:0009887	animal organ morphogenesis	BP	1001	308	0	0
GO:0050793	regulation of developmental process	BP	2563	684	0	0
GO:0051239	regulation of multicellular organismal process	BP	2969	776	0	0
GO:0006928	movement of cell or subcellular component	BP	2086	570	0	0

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0048699	generation of neurons	BP	1437	414	0	0
GO:0002009	morphogenesis of an epithelium	BP	528	182	0	0
GO:0071840	cellular component organization or biogenesis	BP	6486	1557	0	0
GO:0048518	positive regulation of biological process	BP	5841	1415	0	0
GO:0030030	cell projection organization	BP	1477	422	0	0
GO:0050794	regulation of cellular process	BP	11074	2525	0	0
GO:0034329	cell junction assembly	BP	223	94	0	0
GO:0009790	embryo development	BP	959	293	0	0
GO:0048523	negative regulation of cellular process	BP	4821	1189	0	0
GO:0070887	cellular response to chemical stimulus	BP	3175	819	0	0
GO:0051128	regulation of cellular component organization	BP	2403	639	0	0
GO:0051270	regulation of cellular component movement	BP	985	298	0	0
GO:0048729	tissue morphogenesis	BP	639	209	0	0
GO:0034330	cell junction organization	BP	270	107	0	0
GO:0001655	urogenital system development	BP	316	120	0	0
GO:0022612	gland morphogenesis	BP	119	59	0	0
GO:0045595	regulation of cell differentiation	BP	1761	485	0	0
GO:0048732	gland development	BP	430	151	0	0
GO:0035295	tube development	BP	1065	315	0	0
GO:0030182	neuron differentiation	BP	1303	373	0	0
GO:0120036	plasma membrane bounded cell projection organization	BP	1444	407	0	0
GO:0007167	enzyme linked receptor protein signaling pathway	BP	1028	305	0	0
GO:0072001	renal system development	BP	280	108	0	0
GO:0031325	positive regulation of cellular metabolic process	BP	3074	787	0	0

```
goanaOut_BP <- goanaOut[goanaOut$Ont == "BP",]
print(paste("Amount of significant GO Biological Process terms:",
            as.character(sum(goanaOut_BP$FDR.DE < 0.05))))
```

```
## [1] "Amount of significant GO Biological Process terms: 795"
```

Order topGORNA on number of genes in the GO term, this will show more "specific" GO terms (less genes in the term means a term lower in the hierarchy).

```
topGORNA[order(topGORNA$N),]
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0022612	gland morphogenesis	BP	119	59	0	0
GO:0034329	cell junction assembly	BP	223	94	0	0
GO:0034330	cell junction organization	BP	270	107	0	0
GO:0072001	renal system development	BP	280	108	0	0
GO:0001655	urogenital system development	BP	316	120	0	0
GO:0048732	gland development	BP	430	151	0	0
GO:0002009	morphogenesis of an epithelium	BP	528	182	0	0
GO:0048729	tissue morphogenesis	BP	639	209	0	0
GO:0030036	actin cytoskeleton organization	BP	642	216	0	0
GO:0030029	actin filament-based process	BP	734	251	0	0
GO:0009790	embryo development	BP	959	293	0	0
GO:0051270	regulation of cellular component movement	BP	985	298	0	0
GO:0009887	animal organ morphogenesis	BP	1001	308	0	0
GO:0007167	enzyme linked receptor protein signaling pathway	BP	1028	305	0	0
GO:0035295	tube development	BP	1065	315	0	0
GO:0060429	epithelium development	BP	1264	375	0	0
GO:0030182	neuron differentiation	BP	1303	373	0	0
GO:0007010	cytoskeleton organization	BP	1306	397	0	0

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0007155	cell adhesion	BP	1352	416	0	0
GO:0022610	biological adhesion	BP	1360	419	0	0
GO:0048699	generation of neurons	BP	1437	414	0	0
GO:0120036	plasma membrane bounded cell projection organization	BP	1444	407	0	0
GO:0030030	cell projection organization	BP	1477	422	0	0
GO:0022008	neurogenesis	BP	1536	450	0	0
GO:0042127	regulation of cell proliferation	BP	1668	481	0	0
GO:0045595	regulation of cell differentiation	BP	1761	485	0	0
GO:0009888	tissue development	BP	1968	578	0	0
GO:0048468	cell development	BP	2045	565	0	0
GO:0008283	cell proliferation	BP	2077	574	0	0
GO:0006928	movement of cell or subcellular component	BP	2086	570	0	0
GO:0007399	nervous system development	BP	2296	649	0	0
GO:0051128	regulation of cellular component organization	BP	2403	639	0	0
GO:0050793	regulation of developmental process	BP	2563	684	0	0
GO:0009653	anatomical structure morphogenesis	BP	2656	746	0	0
GO:0051239	regulation of multicellular organismal process	BP	2969	776	0	0
GO:0031325	positive regulation of cellular metabolic process	BP	3074	787	0	0
GO:0070887	cellular response to chemical stimulus	BP	3175	819	0	0
GO:0048513	animal organ development	BP	3428	889	0	0
GO:0030154	cell differentiation	BP	4123	1061	0	0
GO:0048869	cellular developmental process	BP	4322	1106	0	0
GO:0048731	system development	BP	4760	1227	0	0
GO:0048523	negative regulation of cellular process	BP	4821	1189	0	0
GO:0048522	positive regulation of cellular process	BP	5142	1287	0	0
GO:0007275	multicellular organism development	BP	5351	1368	0	0
GO:0048856	anatomical structure development	BP	5836	1485	0	0
GO:0048518	positive regulation of biological process	BP	5841	1415	0	0
GO:0032502	developmental process	BP	6266	1575	0	0
GO:0016043	cellular component organization	BP	6269	1535	0	0
GO:0071840	cellular component organization or biogenesis	BP	6486	1557	0	0
GO:0050794	regulation of cellular process	BP	11074	2525	0	0

Writing data for comparison of results

we write out two dataframes that can be used in a separate file to compare results of the analyses.

Write out the results of edgeR analysis

```
edgeR_res <- LRT$table[order(LRT$table$FDR, decreasing=F),]
write.table(edgeR_res, sep= "\t", file="RNAseq_results.txt")
```

Write out the results of Gene Set analysis

```
RNAseq_GSA_res <- topGO(goanaOut, ontology="BP", number=100)
write.table(RNAseq_GSA_res, sep= "\t", file="RNAseq_GSA_results.txt")
```

APPENDIX B
EXPRESSION ARRAY ANALYSIS

This appendix shows the R markdown code for the Expression Array data analysis.

Analysis of Affybatch Data

AHTA: Prostate Cancer Project

Analysis of AffyBatch data

Data

In this analysis the follwing data is used: E-GEOD-71783 - Gene expression profiling of the prostate biopsy samples from cancer and adjacent normal tissues of European American prostate cancer patients

Setup

```
library(affy)
library(arrayQualityMetrics)
library(ArrayExpress)
library(limma)
library(siggenes)
library(oligo)
library("annotate")
library("affycoretools")
library("huex10sttranscriptcluster.db")
```

Load Data

To load the data, one can either use this fist chunk to start download and load the data immediately from the website of ArrayExpress. If the data already is on the local computer, it can be loaded by using the second chunk. Note that in the latter case the working directory must point to the directory containing the data.

```
rawData <- ArrayExpress("E-GEOD-71783")
```

```
sdrf_location <- file.path("E-GEOD-71783.sdrf.txt")
SDRF <- read.delim(sdrf_location)

rownames(SDRF) <- SDRF$Array.Data.File
SDRF <- AnnotatedDataFrame(SDRF)

celFiles <- list.celfiles()
rawData <- read.celfiles(celFiles, verbose=T, phenoData=SDRF)
```

```
## Loading required package: pd.huex.1.0.st.v2
## Loading required package: RSQLite
## Loading required package: DBI
## Platform design info loaded.
## Reading in : **********.CEL
## Warning in read.celfiles(celFiles, verbose = T, phenoData = SDRF):
## 'channel' automatically added to varMetadata in phenoData.
```

By now, the 30 samples should be loaded. To check if the data is correctly loaded one can simply type the name of the object to get some information out.

```
rawData
```

```
## ExonFeatureSet (storageMode: lockedEnvironment)
## assayData: 6553600 features, 30 samples
```

```
##   element names: exprs
## protocolData
##   rowNames: GSM1845257_139-.CEL GSM1845256_139LM.CEL ...
##     GSM1845228_9.CEL (30 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: GSM1845257_139-.CEL GSM1845256_139LM.CEL ...
##     GSM1845228_9.CEL (30 total)
##   varLabels: Source.Name Comment..Sample_description. ...
##     Comment..Derived.ArrayExpress.FTP.file. (37 total)
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.huex.1.0.st.v2
```

Quality Control on raw data

First thing we can do in the analysis is a quality check on the raw data. Note that this operations require a lot of RAM memory and take quite long to execute.

arrayQualityMetrics (open "index.html" file for a full overview of the output)

```
arrayQualityMetrics(rawData, outdir="output/raw", force=TRUE)
arrayQualityMetrics(rawData, outdir="output/rawlog", force=TRUE, do.logtransform=TRUE)
```

Preprocessing

The next step is to prepare the data for further analysis. The package oligo allows us to perform background correction, normalization and summarization in one single step using a deconvolution method for background correction, quantile normalization and the RMA (robust multichip average) algorithm for summarization. By setting the target to core, the remaining data will only by data on gene level.

```
normdata <- oligo::rma(rawData, target="core")
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

```
dim(rawData)
```

```
## Features  Samples
##  6553600       30
```

```
dim(normdata)
```

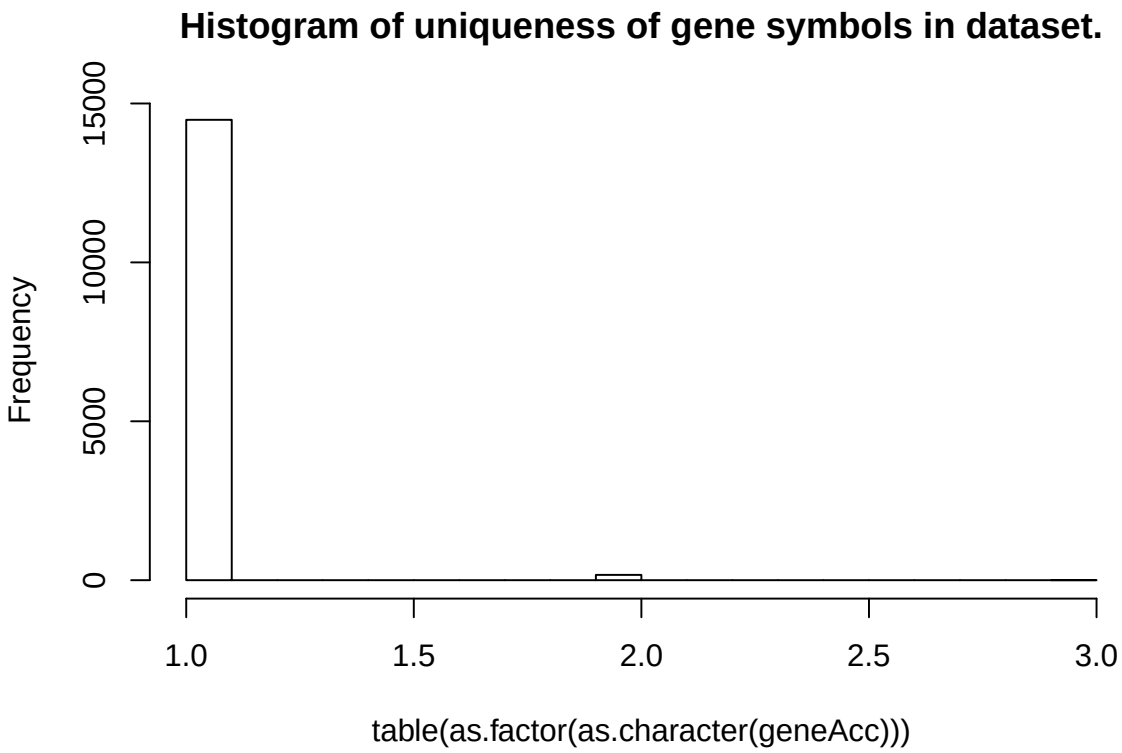
```
## Features  Samples
##    22011       30
```

As the remaining data still contains probeIDs we like to replace these by the actual gene. For this we need an additional library that allows us to map the probeIDs to a gene.

```
geneAcc <- huex10sttranscriptclusterSYMBOL[rownames(normdata)]
rownames(normdata) <- paste(as.character(c(1:nrow(normdata))), geneAcc,  sep=".")
```

The histogram shows the number of unique values of the genes. As most of the genes occur only occurs once, we are sure that we are working on the gene level.

```
hist(table(as.factor(as.character(geneAcc))), main="Histogram of uniqueness of gene symbols in dataset.")
```

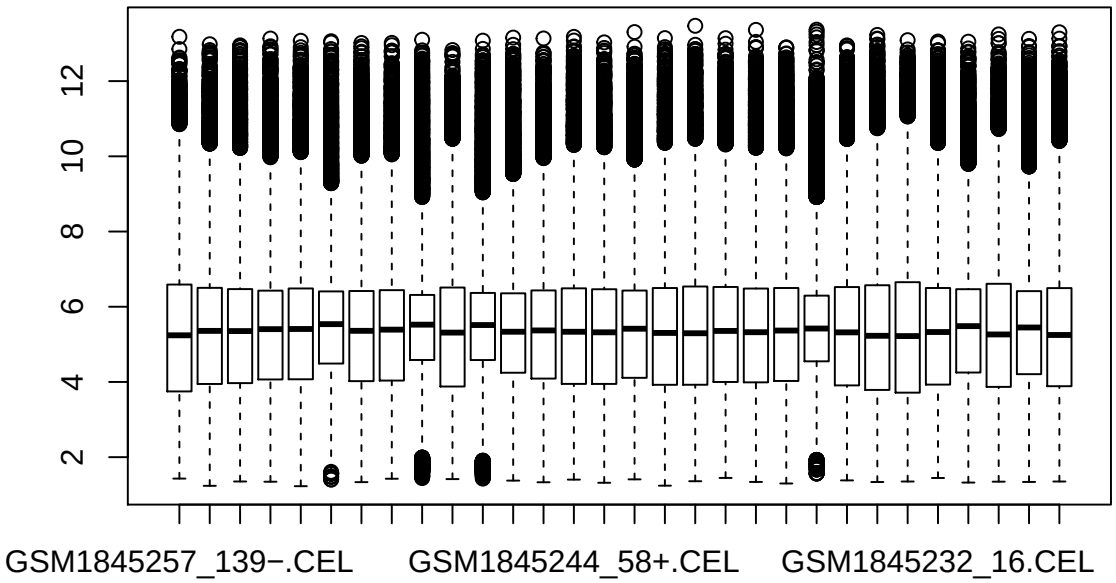


From this data we extract the expression data

```
d <- exprs(normdata)
```

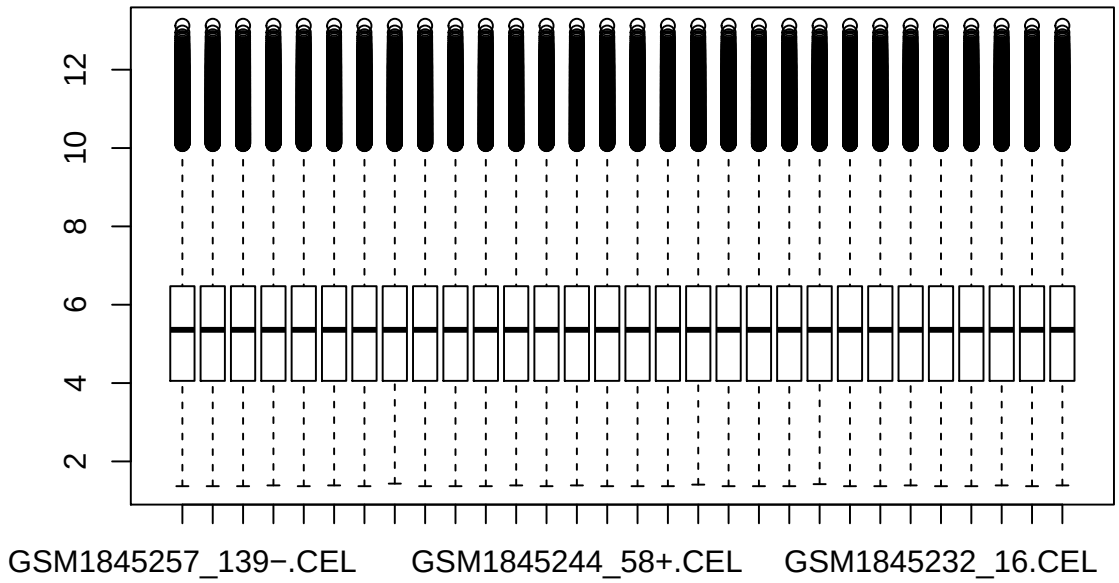
Check data distribution

```
boxplot(d)
```



Although the boxplot looks pretty good, it can be further optimized. Between array normalization with quantile normalization, corrects for the batch effect

```
d2 <- normalizeQuantiles(d)
boxplot(d2)
```




```
humanRMA <- d
```

Quality Control on preprocessed data

QC post preprocessing

```
arrayQualityMetrics(normdata, outdir="output/rma2", force=TRUE) #RMA produces log-transformed data
```

RMA flattens the trend observed for the standard deviation versus rank of the mean, which means the post-processing made the data more homoskedastic

Differential expression analysis with RMA preprocessed data

Quick view on the phenotypic data and the intensity values

```
head(pData(normdata))
```

	Source.Name	Comment..Sample_description.	Comment..Sample_source_name.	Comment..Sample_title.
GSM1845257_139-.CEL	GSM1845257 1	needle biopsy specimen from prostate	prostate biopsy from prostate cancer patient	EA normal 139 [Affymetrix]
GSM1845256_139LM.CEL	GSM1845256 1	needle biopsy specimen from prostate	prostate biopsy from prostate cancer patient	EA tumor 139 [reanalysis Affymetrix]
GSM1845255_124-.CEL	GSM1845255 1	needle biopsy specimen from prostate	prostate biopsy from prostate cancer patient	EA normal 124 [Affymetrix]
GSM1845254_124LA.CEL	GSM1845254 1	needle biopsy specimen from prostate	prostate biopsy from prostate cancer patient	EA tumor 124 [reanalysis Affymetrix]
GSM1845253_121-.CEL	GSM1845253 1	needle biopsy specimen from prostate	prostate biopsy from prostate cancer patient	EA normal 121 [Affymetrix]
GSM1845252_121LM.CEL	GSM1845252 1	needle biopsy specimen from prostate	prostate biopsy from prostate cancer patient	EA tumor 121 [reanalysis Affymetrix]

Char..organism.	Source.REF	Term.Access.Number	Organism.part.	Source.REF.1	Term.Accession.Number.1
Homo sapiens	EFO	http://purl.obolibrary.org/obo/NCBITaxon_9606	prostate	EFO	http://purl.obolibrary.org/obo/UBERON_0002367
Homo sapiens	EFO	http://purl.obolibrary.org/obo/NCBITaxon_9606	prostate	EFO	http://purl.obolibrary.org/obo/UBERON_0002367
Homo sapiens	EFO	http://purl.obolibrary.org/obo/NCBITaxon_9606	prostate	EFO	http://purl.obolibrary.org/obo/UBERON_0002367
Homo sapiens	EFO	http://purl.obolibrary.org/obo/NCBITaxon_9606	prostate	EFO	http://purl.obolibrary.org/obo/UBERON_0002367
Homo sapiens	EFO	http://purl.obolibrary.org/obo/NCBITaxon_9606	prostate	EFO	http://purl.obolibrary.org/obo/UBERON_0002367

Protocol.REF	Term.Source.REF.2	Extract.Name	Material.Type	Protocol.REF.3	Term.Source.REF.5	Labeled.Extract.Name	Label	Assay.Name
P-GSE71783-2	ArrayExpress	GSM1845257 extract 1	total RNA	P-GSE71783-5	ArrayExpress	GSM1845257 LE 1	biotin	GSM1845257
P-GSE71783-2	ArrayExpress	GSM1845256 extract 1	total RNA	P-GSE71783-5	ArrayExpress	GSM1845256 LE 1	biotin	GSM1845256
P-GSE71783-2	ArrayExpress	GSM1845255 extract 1	total RNA	P-GSE71783-5	ArrayExpress	GSM1845255 LE 1	biotin	GSM1845255
P-GSE71783-2	ArrayExpress	GSM1845254 extract 1	total RNA	P-GSE71783-5	ArrayExpress	GSM1845254 LE 1	biotin	GSM1845254
P-GSE71783-2	ArrayExpress	GSM1845253 extract 1	total RNA	P-GSE71783-5	ArrayExpress	GSM1845253 LE 1	biotin	GSM1845253
P-GSE71783-2	ArrayExpress	GSM1845252 extract 1	total RNA	P-GSE71783-5	ArrayExpress	GSM1845252 LE 1	biotin	GSM1845252

```
head(humanRMA)
```

```
##           GSM1845257_139-.CEL GSM1845256_139LM.CEL GSM1845255_124-.CEL
## 1.B3GALT6            4.740862             5.400059            5.420580
## 2.PUSL1              5.939760             6.016387            6.016269
## 3.VWA1               5.354928             5.872004            5.669543
## 4.CALML6             6.077443             6.481294            6.219925
## 5.PRKCZ              7.149398             8.142289            7.577925
## 6.SKI                6.210976             6.505938            6.076239
##           GSM1845254_124LA.CEL GSM1845253_121-.CEL GSM1845252_121LM.CEL
## 1.B3GALT6            5.603492            5.618571             6.321365
## 2.PUSL1              6.314282            6.058429             6.506621
## 3.VWA1               5.863213            5.829449             6.651441
## 4.CALML6             6.586455            6.157135             7.251490
## 5.PRKCZ              8.087997            8.009275             8.894754
## 6.SKI                6.117053            5.835813             6.426919
##           GSM1845251_105-.CEL GSM1845250_105LA.CEL GSM1845249_87-.CEL
## 1.B3GALT6            5.419318             5.787626           6.416717
## 2.PUSL1              5.668390             5.963925           6.433364
## 3.VWA1               5.642401             5.853781           6.575473
## 4.CALML6             5.737526             6.350106           7.039029
## 5.PRKCZ              7.515636             8.392925           8.621100
## 6.SKI                6.042638             6.431134           7.696570
##           GSM1845248_87RB.CEL GSM1845247_60-.CEL GSM1845246_60+.CEL
## 1.B3GALT6            5.450299           6.071902           5.431422
## 2.PUSL1              5.944258           6.387920           5.520074
## 3.VWA1               5.689214           6.612674           5.658708
## 4.CALML6             6.317954           7.047423           5.900312
## 5.PRKCZ              7.365439           8.623979           7.120600
## 6.SKI                5.769907           7.129278           6.487747
##           GSM1845245_58-.CEL GSM1845244_58+.CEL GSM1845243_57-.CEL
## 1.B3GALT6           5.463718           5.141377           5.142780
## 2.PUSL1             5.767370           5.627882           5.386115
## 3.VWA1              5.762056           5.474092           5.405388
## 4.CALML6            6.003272           5.995912           5.775725
## 5.PRKCZ             7.515870           7.364859           7.167553
## 6.SKI               6.221580           5.805472           5.776459
##           GSM1845242_57+.CEL GSM1845241_44-.CEL GSM1845240_44+.CEL
## 1.B3GALT6           5.431181           5.072017           4.759423
## 2.PUSL1             5.879055           5.454986           5.592864
## 3.VWA1              5.922621           5.274370           5.354881
## 4.CALML6            6.019550           5.558634           5.747268
## 5.PRKCZ             7.623111           7.477420           6.784004
## 6.SKI               6.000133           6.071889           5.766392
##           GSM1845239_30-.CEL GSM1845238_30+.CEL GSM1845237_B2.CEL
## 1.B3GALT6           5.142362           5.169361          5.378790
## 2.PUSL1             5.942378           5.694536          5.943683
## 3.VWA1              5.430361           5.549308          5.836599
## 4.CALML6            5.991197           6.046737          6.169629
## 5.PRKCZ             7.555513           7.163980          7.484252
## 6.SKI               5.721471           5.940389          5.957303
##           GSM1845236_B1.CEL GSM1845235_18-.CEL GSM1845234_18+.CEL
## 1.B3GALT6          5.726183           5.044110           4.887805
## 2.PUSL1            5.686606           5.489169           5.020123
## 3.VWA1             5.714242           5.315527           5.047059
## 4.CALML6           6.024104           5.972433           5.539224
```

```
## 5.PRKCZ            7.345204           7.057427           6.945484
## 6.SKI              6.585367           6.150419           5.616095
##           GSM1845233_16+.CEL GSM1845232_16.CEL GSM1845231_15-.CEL
## 1.B3GALT6           4.832423           5.118476           6.247789
## 2.PUSL1             5.341960           5.469858           5.972546
## 3.VWA1              5.048010           5.327593           6.398959
## 4.CALML6            5.828234           6.088116           6.607092
## 5.PRKCZ             6.990207           8.136957           7.976590
## 6.SKI               5.881399           5.688687           6.088617
##           GSM1845230_15+.CEL GSM1845229_9+.CEL GSM1845228_9.CEL
## 1.B3GALT6           4.678653          6.109371         4.887288
## 2.PUSL1             5.312582          5.994868         5.438280
## 3.VWA1              5.185412          6.120709         5.315677
## 4.CALML6            5.492197          6.105776         5.698908
## 5.PRKCZ             7.058911          7.405994         6.981052
## 6.SKI               5.730845          6.749787         5.854827
```

We store the effect of interest (normal vs tumor tissue) as a factor, as well as the blocking patient effect.

```
library("plyr")
```

```
tissue <- factor(pData(rawData)[, 4])
tissue <- mapvalues(as.factor(grepl("tumor", tissue)),
                    from=levels(as.factor(grepl("tumor", tissue))),
                    to= c("healthy", "tumor"))
patient <- as.factor(sapply(strsplit(as.character(pData(rawData)[,4]), " "), "[[", 3))
```

Define design matrix

```
design <- model.matrix(~tissue+patient)

cont.matrix <- makeContrasts(TvsN=tissuetumor, levels=design)
```

```
## Warning in makeContrasts(TvsN = tissuetumor, levels = design): Renaming
## (Intercept) to Intercept
```

```
rownames(cont.matrix)[1] <- "(Intercept)"
```

Fitting the model:

```
fit <- lmFit(humanRMA, design)
fit2 <- contrasts.fit(fit, cont.matrix)
fit2 <- eBayes(fit2)
```

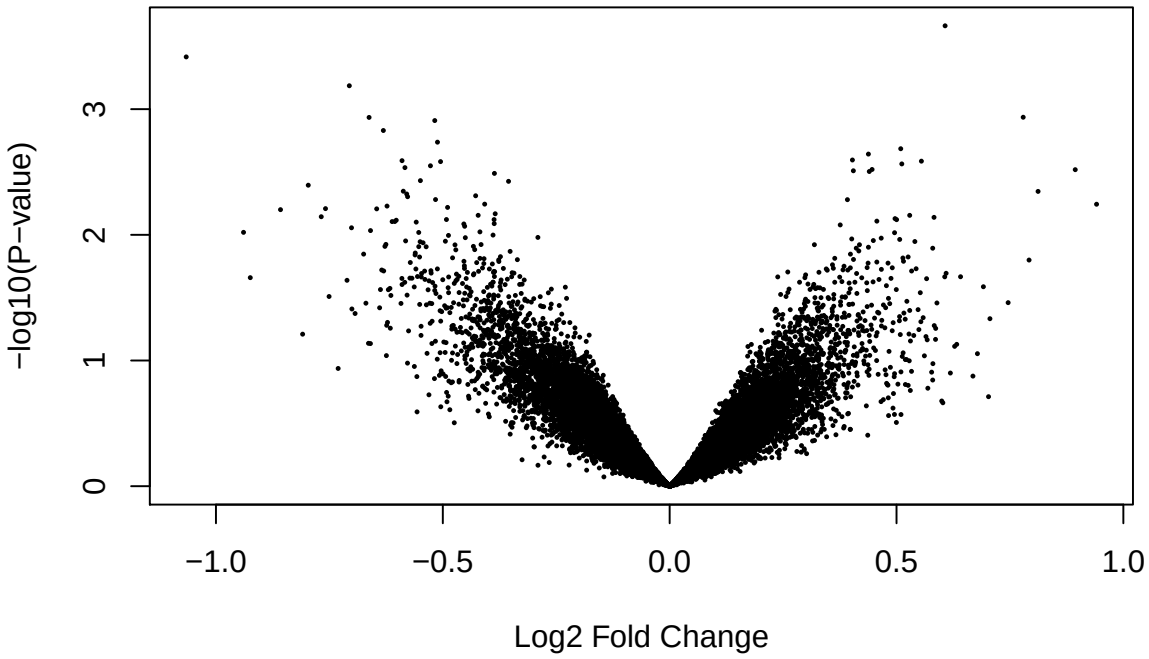
Getting top genes:

```
LIMMAout_RMA <- topTable(fit2, adjust="BH", number=nrow(normdata))
head(LIMMAout_RMA, n=10)
```

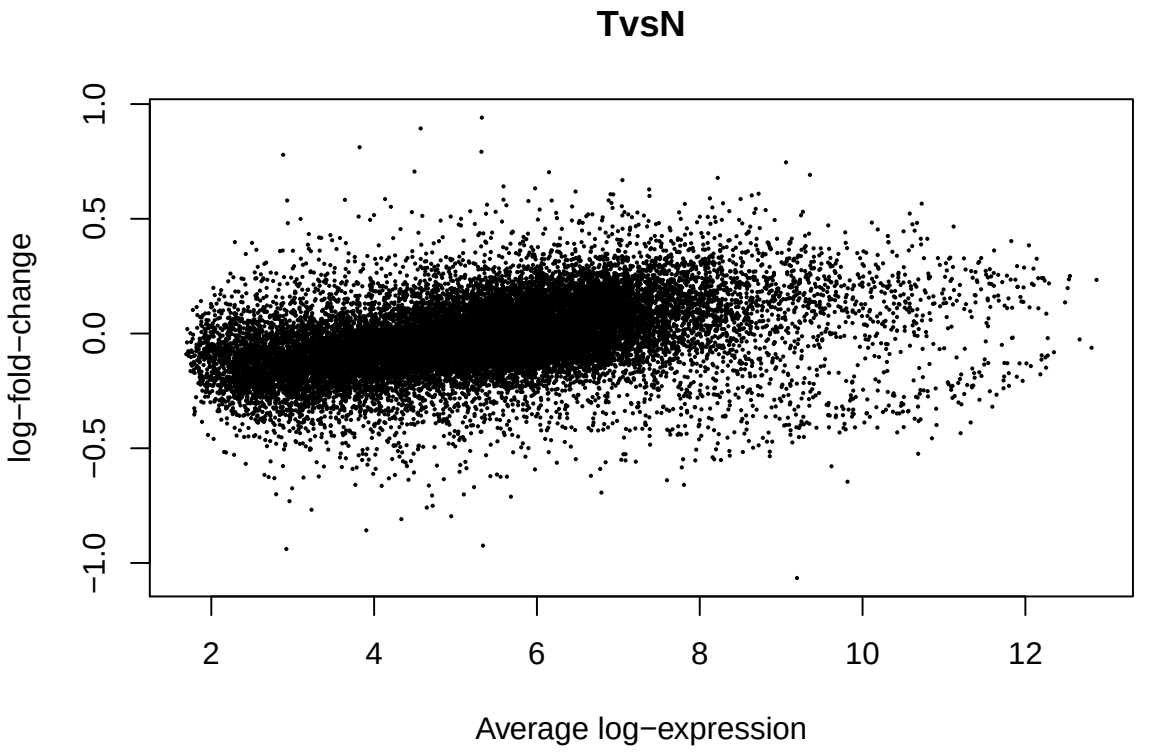
	logFC	AveExpr	t	P.Value	adj.P.Val	B
21027.SLC52A2	0.6070928	6.904212	4.498916	0.0002172	0.9988071	-2.999836
21909.LINC00839	-1.0655598	9.194311	-4.255301	0.0003839	0.9988071	-3.106784
1457.FAM89A	-0.7060515	4.711975	-4.029767	0.0006513	0.9988071	-3.209778
14630.MAGT1	0.7791750	2.882731	3.783337	0.0011595	0.9988071	-3.326386
18567.ADH4	-0.6625130	4.671936	-3.781837	0.0011636	0.9988071	-3.327108
13538.UBE2D3	-0.5177846	5.707743	-3.757274	0.0012323	0.9988071	-3.338950
18832.PAM	-0.6310810	4.180816	-3.679069	0.0014791	0.9988071	-3.376898
2938.CCDC51	-0.5117337	5.180326	-3.587530	0.0018306	0.9988071	-3.421766
1353.DENND1B	0.5093135	8.001218	3.535745	0.0020648	0.9988071	-3.447352
667.RCOR3	0.4380086	5.688188	3.493420	0.0022779	0.9988071	-3.468367

Plotting results

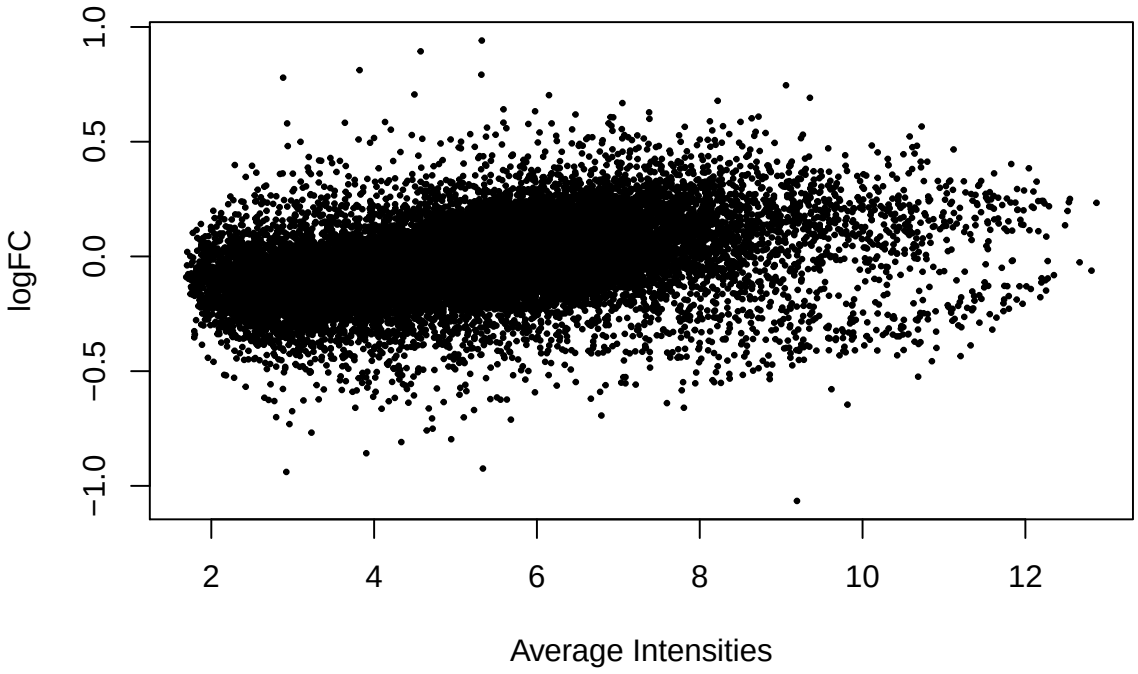
```
volcanoplot(fit2, col=as.factor(LIMMAout_RMA$adj.P.Val < 0.15), style="p-value")
```




```
limma::plotMA(fit2)
```



```
plot(LIMMAout_RMA$AveExpr, LIMMAout_RMA$logFC,
     col=as.factor(LIMMAout_RMA$adj.P.Val < 0.05), pch=20, cex=0.50,
     xlab="Average Intensities", ylab="logFC")
```



Gene set analysis

Goana uses Entrez gene identifiers, we need to convert our gene symbol to entrez ids. For this purpose we use the org.Hs.eg.db package. Because we have no genes that are significantly differentially expressed at FDR < 0.05. We must choose an ad hoc cut-off from our top genes. Here we choose a cut-off of uncorrected p-values < 0.10.

```
library('org.Hs.eg.db')
LIMMAout_filtered <- LIMMAout_RMA[LIMMAout_RMA$P.Value < 0.20, ]

EntrezIDs <- mapIds(org.Hs.eg.db, gsub('.*\\.', '', rownames(LIMMAout_filtered)), 'ENTREZID', 'SYMBOL')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
IDs_unique <- EntrezIDs[!(duplicated(EntrezIDs) | is.na(EntrezIDs))]
```

Overrepresentation analysis with goana:

```
goanaOut <- goana(de=IDs_unique, species="Hs", trend=T)
```

FDR multiple testing adjustment:

```
goanaOut <- goanaOut[order(goanaOut$P.DE, decreasing=FALSE), ]
goanaOut$FDR.DE <- p.adjust(goanaOut$P.DE, method="BH")
```

```
topGOarray <- topGO(goanaOut, ontology="BP", number=50)
topGOarray
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0048518	positive regulation of biological process	BP	5841	972	0.00e+00	0.0000030
GO:0048522	positive regulation of cellular process	BP	5142	864	0.00e+00	0.0000063
GO:0009790	embryo development	BP	959	202	0.00e+00	0.0000069
GO:0008283	cell proliferation	BP	2077	386	0.00e+00	0.0000125
GO:0023051	regulation of signaling	BP	3575	620	0.00e+00	0.0000197
GO:0010646	regulation of cell communication	BP	3522	610	0.00e+00	0.0000281
GO:0044237	cellular metabolic process	BP	10957	1702	0.00e+00	0.0000296
GO:0009987	cellular process	BP	16158	2412	0.00e+00	0.0000428
GO:0071704	organic substance metabolic process	BP	11249	1741	0.00e+00	0.0000447
GO:0008152	metabolic process	BP	11746	1810	1.00e-07	0.0000517
GO:0010033	response to organic substance	BP	3218	560	1.00e-07	0.0000527
GO:0065009	regulation of molecular function	BP	3373	581	1.00e-07	0.0001083
GO:0031325	positive regulation of cellular metabolic process	BP	3074	533	2.00e-07	0.0001697
GO:0065008	regulation of biological quality	BP	3770	640	2.00e-07	0.0001697
GO:0048856	anatomical structure development	BP	5836	950	3.00e-07	0.0002062
GO:0044238	primary metabolic process	BP	10905	1684	3.00e-07	0.0002089
GO:0065007	biological regulation	BP	12351	1886	4.00e-07	0.0002617
GO:0009653	anatomical structure morphogenesis	BP	2656	465	5.00e-07	0.0003356
GO:0071310	cellular response to organic substance	BP	2656	465	5.00e-07	0.0003356
GO:0006807	nitrogen compound metabolic process	BP	10390	1608	5.00e-07	0.0003464
GO:0044093	positive regulation of molecular function	BP	1742	320	6.00e-07	0.0003688
GO:0007275	multicellular organism development	BP	5351	873	8.00e-07	0.0004913
GO:0043085	positive regulation of catalytic activity	BP	1414	266	8.00e-07	0.0004913
GO:0009893	positive regulation of metabolic process	BP	3329	567	9.00e-07	0.0005097
GO:0048468	cell development	BP	2045	366	1.20e-06	0.0006517
GO:0009966	regulation of signal transduction	BP	3215	548	1.30e-06	0.0006826
GO:0051173	positive regulation of nitrogen compound metabolic process	BP	2965	509	1.50e-06	0.0007594
GO:0071840	cellular component organization or biogenesis	BP	6486	1037	2.00e-06	0.0009511
GO:0010941	regulation of cell death	BP	1696	309	2.00e-06	0.0009511
GO:0050789	regulation of biological process	BP	11709	1788	2.10e-06	0.0009515
GO:0048598	embryonic morphogenesis	BP	586	125	2.10e-06	0.0009515
GO:0043067	regulation of programmed cell death	BP	1579	290	2.20e-06	0.0009659
GO:0032502	developmental process	BP	6266	1004	2.20e-06	0.0009783
GO:0042127	regulation of cell proliferation	BP	1668	304	2.40e-06	0.0010194
GO:0043170	macromolecule metabolic process	BP	9622	1491	2.40e-06	0.0010230
GO:0044260	cellular macromolecule metabolic process	BP	8550	1336	2.80e-06	0.0011805
GO:0042981	regulation of apoptotic process	BP	1566	287	2.90e-06	0.0011805
GO:0050790	regulation of catalytic activity	BP	2292	402	2.90e-06	0.0011805
GO:0051239	regulation of multicellular organismal process	BP	2969	507	3.00e-06	0.0011914
GO:0010604	positive regulation of macromolecule metabolic process	BP	3106	528	3.10e-06	0.0011938
GO:0031329	regulation of cellular catabolic process	BP	814	163	3.60e-06	0.0013684
GO:0008150	biological_process	BP	17653	2590	6.70e-06	0.0024185
GO:0016043	cellular component organization	BP	6269	997	9.90e-06	0.0035086
GO:0009792	embryo development ending in birth or egg hatching	BP	578	120	1.27e-05	0.0043836
GO:0007155	cell adhesion	BP	1352	248	1.36e-05	0.0046073
GO:0040020	regulation of meiotic nuclear division	BP	29	14	1.38e-05	0.0046083
GO:0022610	biological adhesion	BP	1360	249	1.49e-05	0.0049201
GO:0050794	regulation of cellular process	BP	11074	1688	1.52e-05	0.0049428
GO:0048583	regulation of response to stimulus	BP	4243	693	1.77e-05	0.0056811
GO:0009611	response to wounding	BP	654	132	2.03e-05	0.0064005

```
goanaOut_BP <- goanaOut[goanaOut$Ont == "BP",]
print(paste("Amount of significant GO Biological Process terms:", as.character(sum(goanaOut_BP$FDR.DE < 0.05))))
```

```
## [1] "Amount of significant GO Biological Process terms: 117"
```

Order topGORNA on number of genes in the GO term, this will show more "specific" GO terms (less genes in the term means a term lower in the hierarchy).

```
topGOarray[order(topGOarray$N), ]
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0040020	regulation of meiotic nuclear division	BP	29	14	1.38e-05	0.0046083
GO:0009792	embryo development ending in birth or egg hatching	BP	578	120	1.27e-05	0.0043836
GO:0048598	embryonic morphogenesis	BP	586	125	2.10e-06	0.0009515
GO:0009611	response to wounding	BP	654	132	2.03e-05	0.0064005
GO:0031329	regulation of cellular catabolic process	BP	814	163	3.60e-06	0.0013684
GO:0009790	embryo development	BP	959	202	0.00e+00	0.0000069
GO:0007155	cell adhesion	BP	1352	248	1.36e-05	0.0046073
GO:0022610	biological adhesion	BP	1360	249	1.49e-05	0.0049201
GO:0043085	positive regulation of catalytic activity	BP	1414	266	8.00e-07	0.0004913
GO:0042981	regulation of apoptotic process	BP	1566	287	2.90e-06	0.0011805
GO:0043067	regulation of programmed cell death	BP	1579	290	2.20e-06	0.0009659
GO:0042127	regulation of cell proliferation	BP	1668	304	2.40e-06	0.0010194
GO:0010941	regulation of cell death	BP	1696	309	2.00e-06	0.0009511
GO:0044093	positive regulation of molecular function	BP	1742	320	6.00e-07	0.0003688
GO:0048468	cell development	BP	2045	366	1.20e-06	0.0006517
GO:0008283	cell proliferation	BP	2077	386	0.00e+00	0.0000125
GO:0050790	regulation of catalytic activity	BP	2292	402	2.90e-06	0.0011805
GO:0009653	anatomical structure morphogenesis	BP	2656	465	5.00e-07	0.0003356
GO:0071310	cellular response to organic substance	BP	2656	465	5.00e-07	0.0003356
GO:0051173	positive regulation of nitrogen compound metabolic process	BP	2965	509	1.50e-06	0.0007594
GO:0051239	regulation of multicellular organismal process	BP	2969	507	3.00e-06	0.0011914
GO:0031325	positive regulation of cellular metabolic process	BP	3074	533	2.00e-07	0.0001697
GO:0010604	positive regulation of macromolecule metabolic process	BP	3106	528	3.10e-06	0.0011938
GO:0009966	regulation of signal transduction	BP	3215	548	1.30e-06	0.0006826
GO:0010033	response to organic substance	BP	3218	560	1.00e-07	0.0000527
GO:0009893	positive regulation of metabolic process	BP	3329	567	9.00e-07	0.0005097
GO:0065009	regulation of molecular function	BP	3373	581	1.00e-07	0.0001083
GO:0010646	regulation of cell communication	BP	3522	610	0.00e+00	0.0000281
GO:0023051	regulation of signaling	BP	3575	620	0.00e+00	0.0000197
GO:0065008	regulation of biological quality	BP	3770	640	2.00e-07	0.0001697
GO:0048583	regulation of response to stimulus	BP	4243	693	1.77e-05	0.0056811
GO:0048522	positive regulation of cellular process	BP	5142	864	0.00e+00	0.0000063
GO:0007275	multicellular organism development	BP	5351	873	8.00e-07	0.0004913
GO:0048856	anatomical structure development	BP	5836	950	3.00e-07	0.0002062
GO:0048518	positive regulation of biological process	BP	5841	972	0.00e+00	0.0000030
GO:0032502	developmental process	BP	6266	1004	2.20e-06	0.0009783
GO:0016043	cellular component organization	BP	6269	997	9.90e-06	0.0035086
GO:0071840	cellular component organization or biogenesis	BP	6486	1037	2.00e-06	0.0009511
GO:0044260	cellular macromolecule metabolic process	BP	8550	1336	2.80e-06	0.0011805
GO:0043170	macromolecule metabolic process	BP	9622	1491	2.40e-06	0.0010230
GO:0006807	nitrogen compound metabolic process	BP	10390	1608	5.00e-07	0.0003464
GO:0044238	primary metabolic process	BP	10905	1684	3.00e-07	0.0002089
GO:0044237	cellular metabolic process	BP	10957	1702	0.00e+00	0.0000296
GO:0050794	regulation of cellular process	BP	11074	1688	1.52e-05	0.0049428
GO:0071704	organic substance metabolic process	BP	11249	1741	0.00e+00	0.0000447
GO:0050789	regulation of biological process	BP	11709	1788	2.10e-06	0.0009515
GO:0008152	metabolic process	BP	11746	1810	1.00e-07	0.0000517
GO:0065007	biological regulation	BP	12351	1886	4.00e-07	0.0002617
GO:0009987	cellular process	BP	16158	2412	0.00e+00	0.0000428

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0008150	biological_process	BP	17653	2590	6.70e-06	0.0024185

Writing data for comparison of results

we write out two dataframes that can be used in a separate file to compare results of the analyses.

Write out the results of limma analysis

```
write.table(LIMMAout_RMA, sep= "\t", file="limmaExprsArray_results.txt")
```

Write out the results of Gene Set analysis

```
array_GSA_res <- topGO(goanaOut, ontology="BP", number=100)
write.table(array_GSA_res, sep= "\t", file="array_GSA_results.txt")
```

APPENDIX C
METHYLATION PROFILING ANALYSIS

This appendix shows the R markdown code for the Methylation Profiling Analysis.

Methylation Array Analysis

A methylation array dataset was analysed to assess methylation changes in tumors versus healthy prostate tissue. The data was collected from prostatectomy patients in the UK. DNA from the patients were analysed with a Illumina Infinium HumanMethylation450 BeadChip. The following data-analysis in this notebook largely follows the guidelines provided with the minfi package.

Load in necessary packages:

```
library("lumi")
library("wateRmelon")
library("minfi")
library("plyr")
library("IlluminaHumanMethylation450kmanifest")
library("shinyMethyl")
library("FDb.InfiniumMethylation.hg19")
library('org.Hs.eg.db')
library("GenomicRanges")
```

Reading in the data

Make sure your working directory is set where the .idat data is before running this chunk. Load in data and annotation data: Note: The .idat files for the blood samples were removed because these are not of interest for this analysis

```
annot <- read.table("data_info_exp_design/sample_data_relation.txt",header=T,sep="\t")
annot$Slide <- substring(annot$Array.Data.File,1,10) #adding annotation for Slide
annot$Array <- substring(annot$Array.Data.File,12,17) #adding annotation for Array
annot$Basename <- substr(annot$Array.Data.File,1,17) #adding annotation for Basename
annot <-annot[!duplicated(annot$Basename),] #remove duplicated row, due to Cy3 & Cy5 taking up separate rows
annot <- annot[!(annot$Characteristics.cell.type. == "blood cell"),] #remove rows with blood cells

RGset <- read.metharray.exp(targets = annot)
```

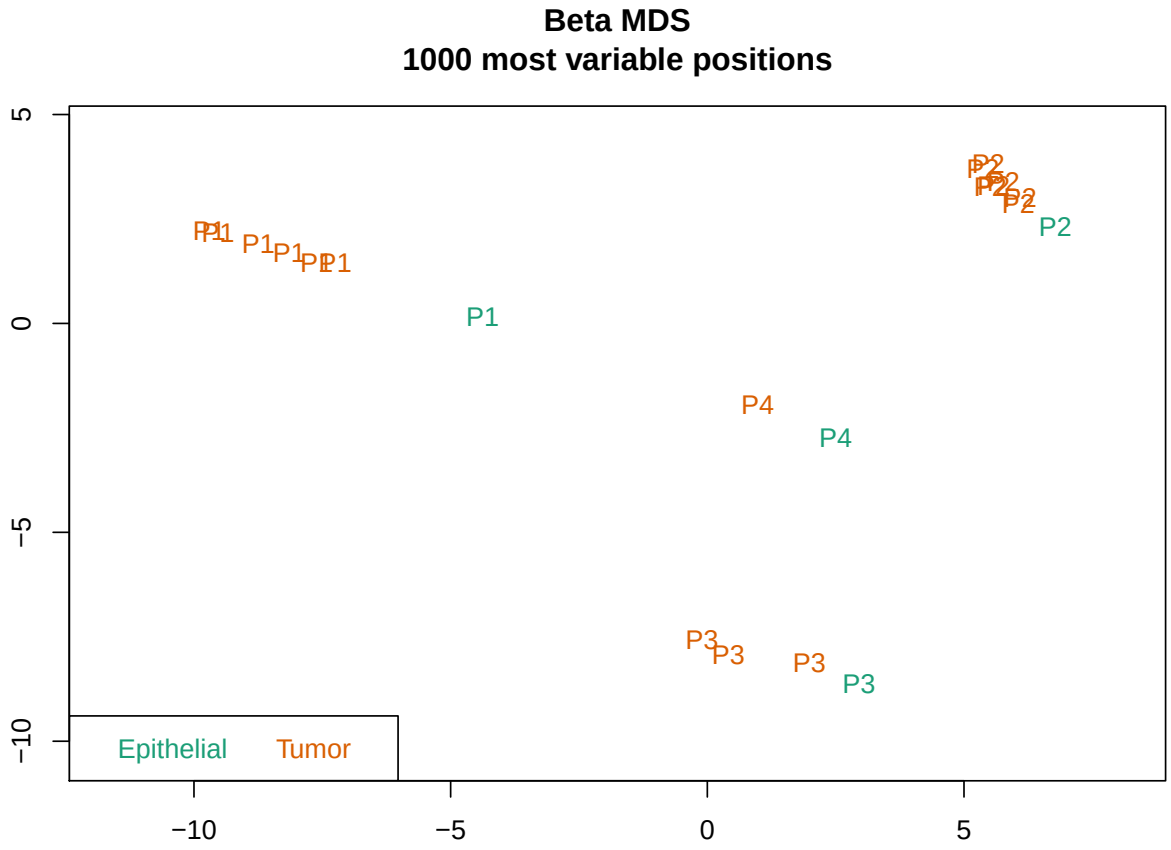
Quality control

Defining factors of interest: patient effect as a blocking effect and tissue type as the main effect.

```
patient <- paste("P",as.numeric(pData(RGset)$Characteristics.individual.), sep="")
tissue <- droplevels(pData(RGset)$Characteristics.cell.type.)
tissue <- mapvalues(tissue, from = levels(tissue), to = c("Epithelial", "Tumor"))
```

MDS plot of betas:

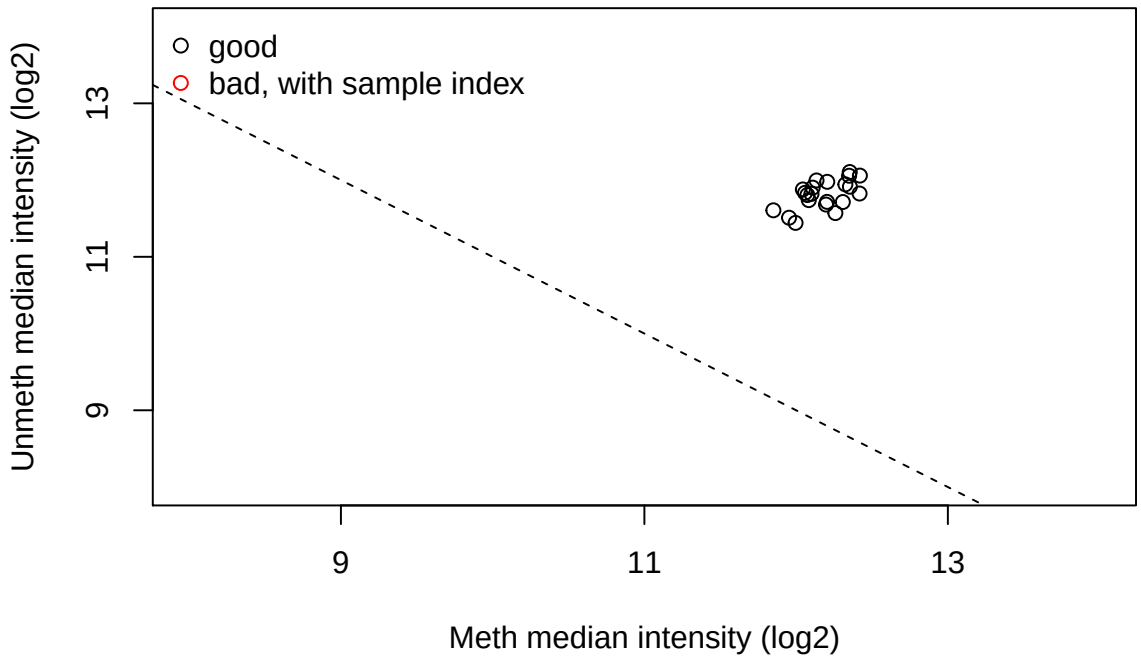
```
mdsPlot(RGset, sampNames=patient, sampGroups=tissue)
```



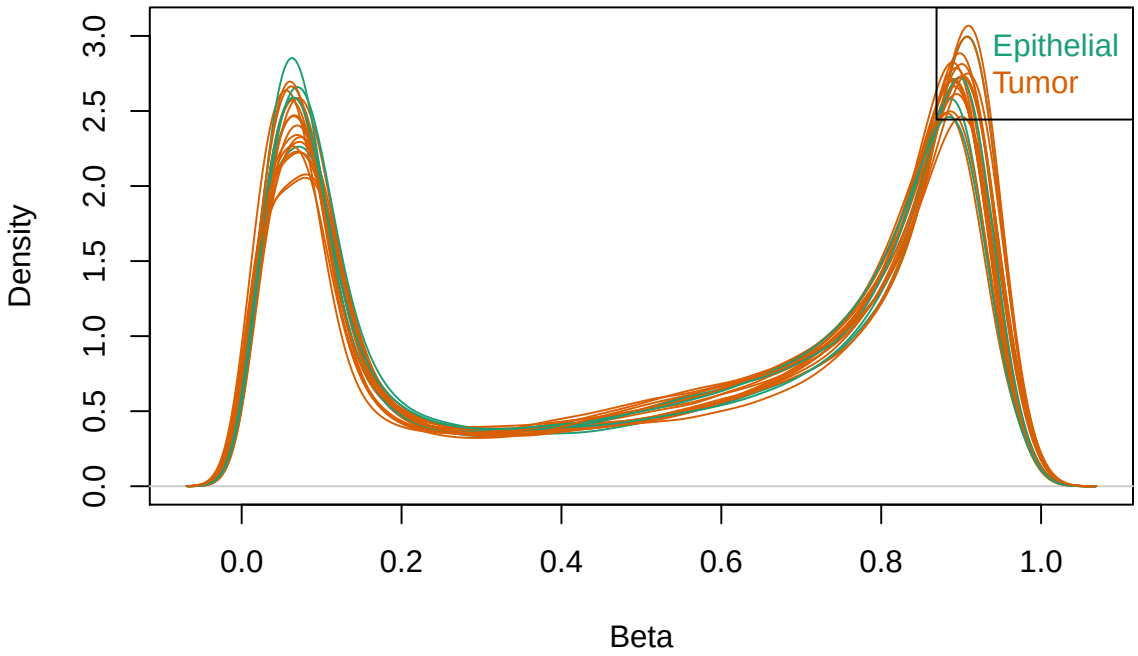
From this plot it is clear that the data is predominantly grouped by patient, but a clear distinction can be made between tumor and epithelial tissue.

Quality control plot:

```
MSet <- preprocessRaw(RGset)
qc <- getQC(MSet)
plotQC(qc)
```




```
phenoData <- pData(RGset)
densityPlot(MSet, sampGroups = tissue)
```



For extra quality control, shinyMethyl can be used:

```
summary <- shinySummarize(RGset)
runShinyMethyl(summary)
```

Preprocessing

Perform functional normalization as described in JFortin et al. (2014). It is an extension to quantile normalization that uses the internal control probes to infer between-array technical variation. It also addresses the problem of normalizing methylation data with global epigenetic changes, making it useful for studies comparing conditions with known large-scale differences, such as cancer/normal studies, or between-tissue studies.

```
GRset <- preprocessFunnorm(RGset)
```

```
## [preprocessFunnorm] Background and dye bias correction with noob
```

```
## [preprocessFunnorm] Mapping to genome
```

```
## [preprocessFunnorm] Quantile extraction
```

```
## Warning in .getSex(CN = CN, xIndex = xIndex, yIndex = yIndex, cutoff
## = cutoff): An inconsistency was encountered while determining sex. One
## possibility is that only one sex is present. We recommend further checks,
## for example with the plotSex function.
```

```
## [preprocessFunnorm] Normalization
```

Remove SNPs. SNPs are a confounder in methylation arrays because they can mess up the hybridisation of the DNA with the probes and result in false positives/negatives.

```
snps <- getSnpInfo(GRset)
head(snps,10)
```

```
## DataFrame with 10 rows and 6 columns
##               Probe_rs Probe_maf      CpG_rs   CpG_maf      SBE_rs
##            <character> <numeric> <character> <numeric> <character>
## cg13869341          NA        NA          NA        NA          NA
## cg14008030          NA        NA          NA        NA          NA
## cg12045430          NA        NA          NA        NA          NA
## cg20826792          NA        NA          NA        NA          NA
## cg00381604          NA        NA          NA        NA          NA
## cg20253340          NA        NA          NA        NA          NA
## cg21870274          NA        NA          NA        NA          NA
## cg03130891  rs77418980  0.305556          NA        NA          NA
## cg24335620 rs147502335    0.0128          NA        NA          NA
## cg16162899          NA        NA          NA        NA          NA
##              SBE_maf
##            <numeric>
## cg13869341        NA
## cg14008030        NA
## cg12045430        NA
## cg20826792        NA
## cg00381604        NA
## cg20253340        NA
## cg21870274        NA
## cg03130891        NA
## cg24335620        NA
## cg16162899        NA
```

```
GRset <- addSnpInfo(GRset)
GRset <- dropLociWithSnps(GRset, snps=c("SBE","CpG"), maf=0)
```

Identify differentially methylated positions: limma

```
Mvalues <- getM(GRset)

design <- model.matrix(~tissue+patient)

cont.matrix <- makeContrasts(TvsH=tissueTumor,levels=design)
```

```
## Warning in makeContrasts(TvsH = tissueTumor, levels = design): Renaming
## (Intercept) to Intercept
```

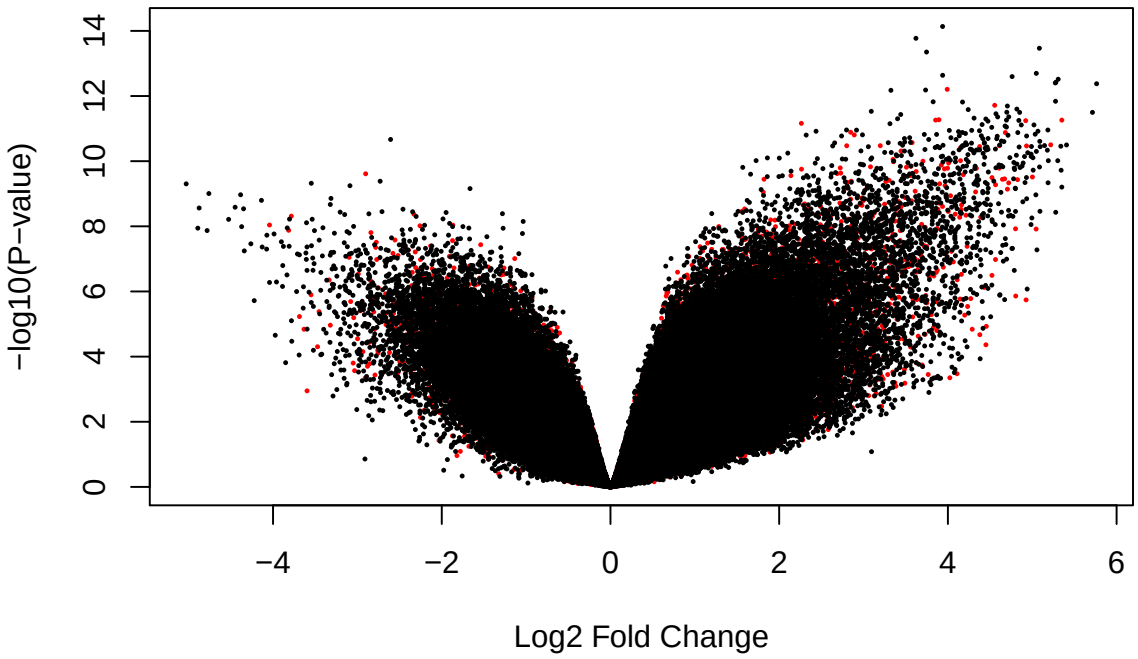
```
rownames(cont.matrix)[1] <- "(Intercept)"
```

```
fit <- lmFit(Mvalues,design)
fit2 <- contrasts.fit(fit,cont.matrix)
fit2 <- eBayes(fit2)
```

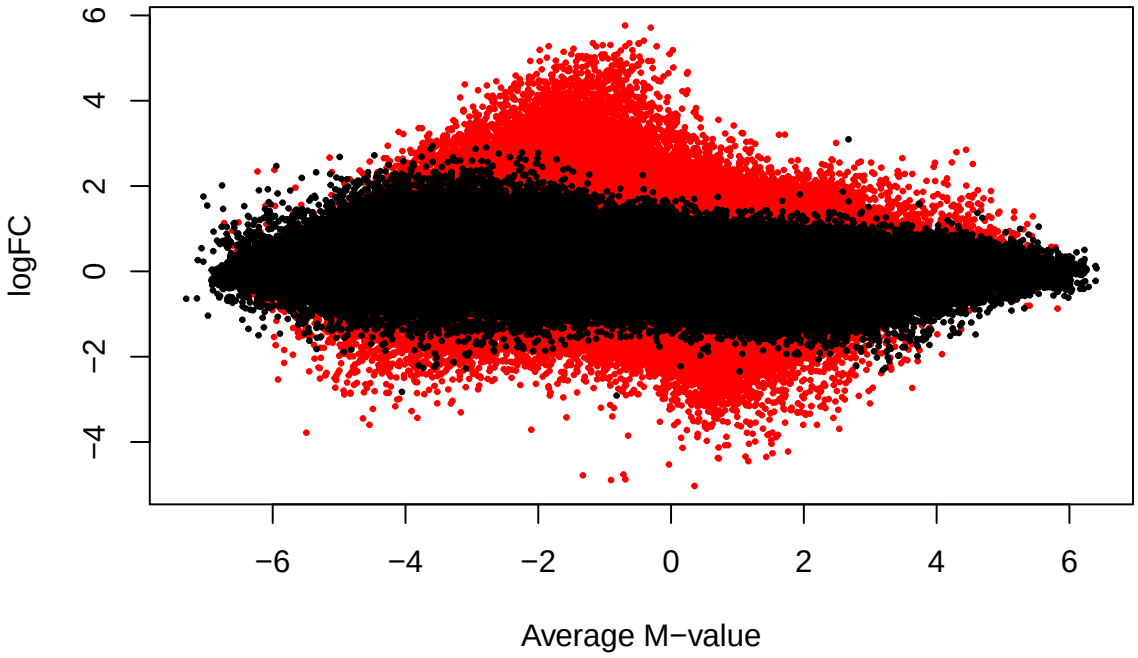
```
LIMMAout<-topTable(fit2,adjust="BH",number=nrow(fit2))
head(LIMMAout)
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
cg03156893	3.937212	-1.4029783	21.89826	0	0	22.59524
cg14361409	3.621003	-1.2674965	20.91155	0	0	21.96075
cg02879662	5.084721	-1.0438878	20.11445	0	0	21.41595
cg00027083	3.747097	-0.8292163	19.81721	0	0	21.20496
cg07198194	5.048362	-0.5706298	18.22331	0	0	19.99508
cg19729649	3.937883	-1.4943552	18.08171	0	0	19.88079

```
volcanoplot(fit2, style="p-value", col=as.factor(LIMMAout$adj.P.Val < 0.05))
```



```
plot(LIMMAout$AveExpr, LIMMAout$logFC, col= as.factor(LIMMAout$adj.P.Val<0.05), pch=20,
      xlab="Average M-value", ylab="logFC", cex=0.50)
```




```
sum(LIMMAout$adj.P.Val<0.05)
```

```
## [1] 71692
```

```
sum(LIMMAout$adj.P.Val<0.05)/length(LIMMAout$adj.P.Val)
```

```
## [1] 0.1531975
```

Gene Set Analysis

With Gene Set Analysis, we can figure out which biological processes have a different methylation status in tumors as opposed to healthy prostate tissue.

Subset for significant probes, a more strict cut-off for significant probes is used here as to not over-saturate with probes.

```
sign_probes <- LIMMAout[LIMMAout$adj.P.Val<0.01,]
```

For gene set analysis, we need to find out the nearest Gene for each CpG probe.

```
hm450 <- get450k()
```

```
## Fetching coordinates for hg19...
```

```
probenames <- rownames(sign_probes)
probes <- hm450[probenames]

ProbeToGene <- getNearestGene(probes)
head(ProbeToGene)
```

	queryHits	subjectHits	distance	nearestGeneSymbol
cg03156893	1	16384	0	PRKCB
cg14361409	2	13030	246	MIR203A
cg02879662	3	2248	0	RNU6-66P
cg00027083	4	7519	0	EPB41L3
cg07198194	5	15000	657	PFKP
cg19729649	6	5246	669	NOTUM

Some probes are very far away from a gene, because these probably do not have a very large effect anymore

```
table(ProbeToGene$distance < 500)
```

```
##
## FALSE  TRUE
## 11377 26968
```

```
ProbeToGene <- ProbeToGene[ProbeToGene$distance < 500,]
```

Goana uses Entrez gene identifiers, we need to convert our gene symbols to entrez ids. For this purpose we use the org.Hs.eg.db package:

```
EntrezIDs <- mapIds(org.Hs.eg.db, ProbeToGene$nearestGeneSymbol, 'ENTREZID', 'SYMBOL')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
EntrezIDs <- EntrezIDs[!(duplicated(EntrezIDs) | is.na(EntrezIDs))]
```

Overrepresentation analysis with goana:

```
goanaOut <- goana(de=EntrezIDs, species="Hs", trend=T)
```

FDR multiple testing adjustment:

```
goanaOut <- goanaOut[order(goanaOut$P.DE, decreasing=FALSE),]
goanaOut$FDR.DE <- p.adjust(goanaOut$P.DE, method="BH")
```

```
topGOcpg <- topGO(goanaOut, ontology="BP", number=50)
topGOcpg
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0048856	anatomical structure development	BP	5810	3034	0	0
GO:0032502	developmental process	BP	6212	3207	0	0
GO:0007275	multicellular organism development	BP	5330	2800	0	0
GO:0009653	anatomical structure morphogenesis	BP	2604	1523	0	0
GO:0048731	system development	BP	4783	2534	0	0
GO:0007399	nervous system development	BP	2296	1358	0	0
GO:0048869	cellular developmental process	BP	4282	2269	0	0
GO:0032501	multicellular organismal process	BP	7510	3696	0	0
GO:0030154	cell differentiation	BP	4100	2183	0	0
GO:0022008	neurogenesis	BP	1555	964	0	0
GO:0065008	regulation of biological quality	BP	3891	2081	0	0
GO:0048699	generation of neurons	BP	1459	910	0	0
GO:0048468	cell development	BP	2093	1225	0	0
GO:0048513	animal organ development	BP	3455	1864	0	0
GO:0030182	neuron differentiation	BP	1317	829	0	0
GO:0051239	regulation of multicellular organismal process	BP	3093	1683	0	0
GO:0051179	localization	BP	6555	3229	0	0
GO:0023052	signaling	BP	6475	3194	0	0

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0007154	cell communication	BP	6522	3213	0	0
GO:0050794	regulation of cellular process	BP	10828	5027	0	0
GO:0032989	cellular component morphogenesis	BP	1079	689	0	0
GO:0023051	regulation of signaling	BP	3567	1888	0	0
GO:0000902	cell morphogenesis	BP	980	635	0	0
GO:0010646	regulation of cell communication	BP	3532	1868	0	0
GO:0050896	response to stimulus	BP	9106	4302	0	0
GO:0048666	neuron development	BP	1071	674	0	0
GO:0050793	regulation of developmental process	BP	2607	1424	0	0
GO:0007267	cell-cell signaling	BP	1584	931	0	0
GO:0065007	biological regulation	BP	12576	5698	0	0
GO:0048518	positive regulation of biological process	BP	6054	2966	0	0
GO:0009887	animal organ morphogenesis	BP	982	619	0	0
GO:0045595	regulation of cell differentiation	BP	1788	1020	0	0
GO:0000904	cell morphogenesis involved in differentiation	BP	712	475	0	0
GO:0051716	cellular response to stimulus	BP	7432	3549	0	0
GO:0048522	positive regulation of cellular process	BP	5330	2634	0	0
GO:0031175	neuron projection development	BP	941	593	0	0
GO:2000026	regulation of multicellular organismal development	BP	2064	1146	0	0
GO:0051128	regulation of cellular component organization	BP	2457	1331	0	0
GO:0032990	cell part morphogenesis	BP	655	439	0	0
GO:0030030	cell projection organization	BP	1497	868	0	0
GO:0048523	negative regulation of cellular process	BP	4759	2372	0	0
GO:0120036	plasma membrane bounded cell projection organization	BP	1462	850	0	0
GO:0007165	signal transduction	BP	6031	2928	0	0
GO:0048858	cell projection morphogenesis	BP	636	427	0	0
GO:0120039	plasma membrane bounded cell projection morphogenesis	BP	635	426	0	0
GO:0051960	regulation of nervous system development	BP	883	557	0	0
GO:0032879	regulation of localization	BP	2746	1457	0	0
GO:0060284	regulation of cell development	BP	910	570	0	0
GO:0009966	regulation of signal transduction	BP	3183	1656	0	0
GO:0048812	neuron projection morphogenesis	BP	621	416	0	0

Order topGOcpg on number of genes in the GO term, this will show more "specific" GO terms (less genes in the term means a term lower in the hierarchy).

```
topGOcpg[order(topGOcpg$N),]
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0048812	neuron projection morphogenesis	BP	621	416	0	0
GO:0120039	plasma membrane bounded cell projection morphogenesis	BP	635	426	0	0
GO:0048858	cell projection morphogenesis	BP	636	427	0	0
GO:0032990	cell part morphogenesis	BP	655	439	0	0
GO:0000904	cell morphogenesis involved in differentiation	BP	712	475	0	0
GO:0051960	regulation of nervous system development	BP	883	557	0	0
GO:0060284	regulation of cell development	BP	910	570	0	0
GO:0031175	neuron projection development	BP	941	593	0	0
GO:0000902	cell morphogenesis	BP	980	635	0	0
GO:0009887	animal organ morphogenesis	BP	982	619	0	0
GO:0048666	neuron development	BP	1071	674	0	0
GO:0032989	cellular component morphogenesis	BP	1079	689	0	0
GO:0030182	neuron differentiation	BP	1317	829	0	0
GO:0048699	generation of neurons	BP	1459	910	0	0
GO:0120036	plasma membrane bounded cell projection organization	BP	1462	850	0	0

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0030030	cell projection organization	BP	1497	868	0	0
GO:0022008	neurogenesis	BP	1555	964	0	0
GO:0007267	cell-cell signaling	BP	1584	931	0	0
GO:0045595	regulation of cell differentiation	BP	1788	1020	0	0
GO:2000026	regulation of multicellular organismal development	BP	2064	1146	0	0
GO:0048468	cell development	BP	2093	1225	0	0
GO:0007399	nervous system development	BP	2296	1358	0	0
GO:0051128	regulation of cellular component organization	BP	2457	1331	0	0
GO:0009653	anatomical structure morphogenesis	BP	2604	1523	0	0
GO:0050793	regulation of developmental process	BP	2607	1424	0	0
GO:0032879	regulation of localization	BP	2746	1457	0	0
GO:0051239	regulation of multicellular organismal process	BP	3093	1683	0	0
GO:0009966	regulation of signal transduction	BP	3183	1656	0	0
GO:0048513	animal organ development	BP	3455	1864	0	0
GO:0010646	regulation of cell communication	BP	3532	1868	0	0
GO:0023051	regulation of signaling	BP	3567	1888	0	0
GO:0065008	regulation of biological quality	BP	3891	2081	0	0
GO:0030154	cell differentiation	BP	4100	2183	0	0
GO:0048869	cellular developmental process	BP	4282	2269	0	0
GO:0048523	negative regulation of cellular process	BP	4759	2372	0	0
GO:0048731	system development	BP	4783	2534	0	0
GO:0007275	multicellular organism development	BP	5330	2800	0	0
GO:0048522	positive regulation of cellular process	BP	5330	2634	0	0
GO:0048856	anatomical structure development	BP	5810	3034	0	0
GO:0007165	signal transduction	BP	6031	2928	0	0
GO:0048518	positive regulation of biological process	BP	6054	2966	0	0
GO:0032502	developmental process	BP	6212	3207	0	0
GO:0023052	signaling	BP	6475	3194	0	0
GO:0007154	cell communication	BP	6522	3213	0	0
GO:0051179	localization	BP	6555	3229	0	0
GO:0051716	cellular response to stimulus	BP	7432	3549	0	0
GO:0032501	multicellular organismal process	BP	7510	3696	0	0
GO:0050896	response to stimulus	BP	9106	4302	0	0
GO:0050794	regulation of cellular process	BP	10828	5027	0	0
GO:0065007	biological regulation	BP	12576	5698	0	0

Identify differentially methylated regions (DMRs): bumphunter

Bumphunter searches genomic regions that are differentially methylated between two conditions, instead of looking for individual CpG that are differentially methylated. The algorithm first makes clusters of probes. Then it computes a t-statistic at each genomic location. Candidate regions are then clusters of probes for which all t-statistics exceed a certain threshold. To test for significance of the candidate regios, bootstraps are used.

Parallellisation:

```
library(doParallel)
registerDoParallel(cores = 4)
```

Run bumphunter on cutoff=0.25, this means that only those regions with a 25% difference in beta-values are evaluated. Ran with 500 bootstraps. This takes a while.

```
dmrs <- bumphunter(GRset, design = design, coef=2, cutoff = 0.25, B=500, maxGap=500, verbose=TRUE,
                   nullMethod="bootstrap")
```

```
## [bumphunterEngine] Parallelizing using 4 workers/cores (backend: doParallelSNOW, version: 1.0.14).
```

```
## [bumphunterEngine] Computing coefficients.
## [bumphunterEngine] Performing 500 bootstraps.
## [bumphunterEngine] Computing marginal bootstrap p-values.
## [bumphunterEngine] cutoff: 0.25
## [bumphunterEngine] Finding regions.
## [bumphunterEngine] Found 6891 bumps.
## [bumphunterEngine] Computing regions for each bootstrap.
## [bumphunterEngine] Estimating p-values and FWER.
```

```
names(dmrs)
```

```
## [1] "table"           "coef"            "fitted"          "pvaluesMarginal"
## [5] "null"            "algorithm"
```

```
head(dmrs$table, n=6)
```

	chr	start	end	value	area	cluster	indexStart	indexEnd	L	clusterL	p.value	fwer
1687	chr16	88717134	88717755	0.43	5.17	77339	374940	374951	12	14	0	0
4232	chrX	56258440	56259373	0.42	4.61	195568	460839	460849	11	14	0	0
1308	chr14	23834808	23836012	0.37	4.05	56642	327403	327413	11	13	0	0
3315	chr5	180017689	180018722	0.38	3.76	154910	145759	145768	10	14	0	0
2084	chr19	46800054	46800602	0.46	3.65	97185	427544	427551	8	8	0	0
2667	chr3	25469694	25469991	0.39	3.52	125629	81888	81896	9	16	0	0

Gene Set Analysis

Subset: take only significant bumps (with a FWER<0.10 for the area).

```
sign_bumps <- dmrs$table[dmrs$table$fwerArea<0.10,]
```

For the significant bumps, construct a GRanges object which can then be used to get the nearest gene symbol.

```
dmrs_GR <- makeGRangesFromDataFrame(sign_bumps[,1:3])

bumpsToGenes <- getNearestGene(dmrs_GR)
head(bumpsToGenes)
```

	queryHits	subjectHits	distance	nearestGeneSymbol
1687	1	5635	0	CYBA
4232	2	3582	0	KLF8
1308	3	2644	0	EFS
3315	4	24117	0	SCGB3A1
2084	5	18533	0	HIF3A
2667	6	17687	0	RARB

Filter out bumps that are far away (>=500bp) from their nearest gene, their impact is probably not as high on the gene as closely differentially methylated regions.

```
table(bumpsToGenes$distance < 500)
```

```
##
## FALSE  TRUE
##    70   374
```

```
bumpsToGenes <- bumpsToGenes[bumpsToGenes$distance < 500,]
```

Goana uses Entrez gene identifiers, we need to convert our gene symbols to entrez ids. For this purpose we use the org.Hs.eg.db package:

```
EntrezIDs_bmp <- mapIds(org.Hs.eg.db, bumpsToGenes$nearestGeneSymbol, 'ENTREZID', 'SYMBOL')
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
EntrezIDs_bmp <- EntrezIDs_bmp[!(duplicated(EntrezIDs_bmp) | is.na(EntrezIDs_bmp))]
```

Overrepresentation analysis with goana:

```
goanaOut_bmp <- goana(de=EntrezIDs_bmp, species="Hs", trend=T)
```

FDR multiple testing adjustment:

```
goanaOut_bmp <- goanaOut_bmp[order(goanaOut_bmp$P.DE, decreasing=FALSE),]
goanaOut_bmp$FDR.DE <- p.adjust(goanaOut_bmp$P.DE, method="BH")
```

```
topGObumps <- topGO(goanaOut_bmp, ontology="BP", number=50)
topGObumps
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0032502	developmental process	BP	6212	156	0.0e+00	0.0000000
GO:0048856	anatomical structure development	BP	5810	149	0.0e+00	0.0000000
GO:0007275	multicellular organism development	BP	5330	140	0.0e+00	0.0000000
GO:0048869	cellular developmental process	BP	4282	118	0.0e+00	0.0000000
GO:0030154	cell differentiation	BP	4100	114	0.0e+00	0.0000000
GO:0032501	multicellular organismal process	BP	7510	171	0.0e+00	0.0000001
GO:0009719	response to endogenous stimulus	BP	1628	60	0.0e+00	0.0000002
GO:0071495	cellular response to endogenous stimulus	BP	1373	54	0.0e+00	0.0000002
GO:0048731	system development	BP	4783	122	0.0e+00	0.0000004
GO:0051239	regulation of multicellular organismal process	BP	3093	90	0.0e+00	0.0000004
GO:0023051	regulation of signaling	BP	3567	98	0.0e+00	0.0000011
GO:0071310	cellular response to organic substance	BP	2595	78	0.0e+00	0.0000021
GO:0010646	regulation of cell communication	BP	3532	96	0.0e+00	0.0000026
GO:0045595	regulation of cell differentiation	BP	1788	60	0.0e+00	0.0000047
GO:0050793	regulation of developmental process	BP	2607	77	0.0e+00	0.0000050
GO:0070848	response to growth factor	BP	724	34	0.0e+00	0.0000071
GO:0009653	anatomical structure morphogenesis	BP	2604	76	0.0e+00	0.0000099
GO:0023052	signaling	BP	6475	145	0.0e+00	0.0000138
GO:0009966	regulation of signal transduction	BP	3183	86	0.0e+00	0.0000255
GO:0071363	cellular response to growth factor stimulus	BP	694	32	0.0e+00	0.0000256
GO:0070887	cellular response to chemical stimulus	BP	3144	85	0.0e+00	0.0000280
GO:0007154	cell communication	BP	6522	144	0.0e+00	0.0000365
GO:2000026	regulation of multicellular organismal development	BP	2064	63	0.0e+00	0.0000398
GO:0048513	animal organ development	BP	3455	90	0.0e+00	0.0000475
GO:0090287	regulation of cellular response to growth factor stimulus	BP	282	19	1.0e-07	0.0000581
GO:0007399	nervous system development	BP	2296	67	1.0e-07	0.0000663
GO:0009888	tissue development	BP	1961	60	1.0e-07	0.0000710
GO:0051093	negative regulation of developmental process	BP	1017	39	1.0e-07	0.0000715
GO:0048523	negative regulation of cellular process	BP	4759	112	1.0e-07	0.0001126
GO:0090288	negative regulation of cellular response to growth factor stimulus	BP	160	14	2.0e-07	0.0001149
GO:0010033	response to organic substance	BP	3172	83	2.0e-07	0.0001240
GO:0051241	negative regulation of multicellular organismal process	BP	1260	44	2.0e-07	0.0001240
GO:0051128	regulation of cellular component organization	BP	2457	69	2.0e-07	0.0001356
GO:0045596	negative regulation of cell differentiation	BP	725	31	2.0e-07	0.0001356

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0050794	regulation of cellular process	BP	10828	207	3.0e-07	0.0001714
GO:0022008	neurogenesis	BP	1555	50	3.0e-07	0.0001798
GO:0009887	animal organ morphogenesis	BP	982	37	3.0e-07	0.0001798
GO:0048880	sensory system development	BP	357	20	6.0e-07	0.0003213
GO:0048699	generation of neurons	BP	1459	47	7.0e-07	0.0003619
GO:0007167	enzyme linked receptor protein signaling pathway	BP	1016	37	7.0e-07	0.0003707
GO:0007165	signal transduction	BP	6031	131	7.0e-07	0.0003913
GO:0048646	anatomical structure formation involved in morphogenesis	BP	1151	40	8.0e-07	0.0003950
GO:0051716	cellular response to stimulus	BP	7432	153	1.1e-06	0.0005735
GO:0065007	biological regulation	BP	12576	229	1.4e-06	0.0006261
GO:0044093	positive regulation of molecular function	BP	1736	52	1.4e-06	0.0006261
GO:0042326	negative regulation of phosphorylation	BP	448	22	1.4e-06	0.0006261
GO:0032879	regulation of localization	BP	2746	72	1.4e-06	0.0006261
GO:0030182	neuron differentiation	BP	1317	43	1.5e-06	0.0006299
GO:0010648	negative regulation of cell communication	BP	1363	44	1.5e-06	0.0006299
GO:0035295	tube development	BP	1095	38	1.5e-06	0.0006354

Order topGObumps on number of genes in the GO term, this will show more "specific" GO terms (less genes in the term means a term lower in the hierarchy).

```
topGObumps[order(topGObumps$N),]
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0090288	negative regulation of cellular response to growth factor stimulus	BP	160	14	2.0e-07	0.0001149
GO:0090287	regulation of cellular response to growth factor stimulus	BP	282	19	1.0e-07	0.0000581
GO:0048880	sensory system development	BP	357	20	6.0e-07	0.0003213
GO:0042326	negative regulation of phosphorylation	BP	448	22	1.4e-06	0.0006261
GO:0071363	cellular response to growth factor stimulus	BP	694	32	0.0e+00	0.0000256
GO:0070848	response to growth factor	BP	724	34	0.0e+00	0.0000071
GO:0045596	negative regulation of cell differentiation	BP	725	31	2.0e-07	0.0001356
GO:0009887	animal organ morphogenesis	BP	982	37	3.0e-07	0.0001798
GO:0007167	enzyme linked receptor protein signaling pathway	BP	1016	37	7.0e-07	0.0003707
GO:0051093	negative regulation of developmental process	BP	1017	39	1.0e-07	0.0000715
GO:0035295	tube development	BP	1095	38	1.5e-06	0.0006354
GO:0048646	anatomical structure formation involved in morphogenesis	BP	1151	40	8.0e-07	0.0003950
GO:0051241	negative regulation of multicellular organismal process	BP	1260	44	2.0e-07	0.0001240
GO:0030182	neuron differentiation	BP	1317	43	1.5e-06	0.0006299
GO:0010648	negative regulation of cell communication	BP	1363	44	1.5e-06	0.0006299
GO:0071495	cellular response to endogenous stimulus	BP	1373	54	0.0e+00	0.0000002
GO:0048699	generation of neurons	BP	1459	47	7.0e-07	0.0003619
GO:0022008	neurogenesis	BP	1555	50	3.0e-07	0.0001798
GO:0009719	response to endogenous stimulus	BP	1628	60	0.0e+00	0.0000002
GO:0044093	positive regulation of molecular function	BP	1736	52	1.4e-06	0.0006261
GO:0045595	regulation of cell differentiation	BP	1788	60	0.0e+00	0.0000047
GO:0009888	tissue development	BP	1961	60	1.0e-07	0.0000710
GO:2000026	regulation of multicellular organismal development	BP	2064	63	0.0e+00	0.0000398
GO:0007399	nervous system development	BP	2296	67	1.0e-07	0.0000663
GO:0051128	regulation of cellular component organization	BP	2457	69	2.0e-07	0.0001356
GO:0071310	cellular response to organic substance	BP	2595	78	0.0e+00	0.0000021
GO:0009653	anatomical structure morphogenesis	BP	2604	76	0.0e+00	0.0000099
GO:0050793	regulation of developmental process	BP	2607	77	0.0e+00	0.0000050
GO:0032879	regulation of localization	BP	2746	72	1.4e-06	0.0006261
GO:0051239	regulation of multicellular organismal process	BP	3093	90	0.0e+00	0.0000004
GO:0070887	cellular response to chemical stimulus	BP	3144	85	0.0e+00	0.0000280

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0010033	response to organic substance	BP	3172	83	2.0e-07	0.0001240
GO:0009966	regulation of signal transduction	BP	3183	86	0.0e+00	0.0000255
GO:0048513	animal organ development	BP	3455	90	0.0e+00	0.0000475
GO:0010646	regulation of cell communication	BP	3532	96	0.0e+00	0.0000026
GO:0023051	regulation of signaling	BP	3567	98	0.0e+00	0.0000011
GO:0030154	cell differentiation	BP	4100	114	0.0e+00	0.0000000
GO:0048869	cellular developmental process	BP	4282	118	0.0e+00	0.0000000
GO:0048523	negative regulation of cellular process	BP	4759	112	1.0e-07	0.0001126
GO:0048731	system development	BP	4783	122	0.0e+00	0.0000004
GO:0007275	multicellular organism development	BP	5330	140	0.0e+00	0.0000000
GO:0048856	anatomical structure development	BP	5810	149	0.0e+00	0.0000000
GO:0007165	signal transduction	BP	6031	131	7.0e-07	0.0003913
GO:0032502	developmental process	BP	6212	156	0.0e+00	0.0000000
GO:0023052	signaling	BP	6475	145	0.0e+00	0.0000138
GO:0007154	cell communication	BP	6522	144	0.0e+00	0.0000365
GO:0051716	cellular response to stimulus	BP	7432	153	1.1e-06	0.0005735
GO:0032501	multicellular organismal process	BP	7510	171	0.0e+00	0.0000001
GO:0050794	regulation of cellular process	BP	10828	207	3.0e-07	0.0001714
GO:0065007	biological regulation	BP	12576	229	1.4e-06	0.0006261

Look at the overlap between GO terms for the CpG based analysis & the region based analysis.

```
intersect(topGObumps$Term, topGOcpg$Term)
```

```
##  [1] "developmental process"
##  [2] "anatomical structure development"
##  [3] "multicellular organism development"
##  [4] "cellular developmental process"
##  [5] "cell differentiation"
##  [6] "multicellular organismal process"
##  [7] "system development"
##  [8] "regulation of multicellular organismal process"
##  [9] "regulation of signaling"
## [10] "regulation of cell communication"
## [11] "regulation of cell differentiation"
## [12] "regulation of developmental process"
## [13] "anatomical structure morphogenesis"
## [14] "signaling"
## [15] "regulation of signal transduction"
## [16] "cell communication"
## [17] "regulation of multicellular organismal development"
## [18] "animal organ development"
## [19] "nervous system development"
## [20] "negative regulation of cellular process"
## [21] "regulation of cellular component organization"
## [22] "regulation of cellular process"
## [23] "neurogenesis"
## [24] "animal organ morphogenesis"
## [25] "generation of neurons"
## [26] "signal transduction"
## [27] "cellular response to stimulus"
## [28] "biological regulation"
## [29] "regulation of localization"
## [30] "neuron differentiation"
```

```
length(intersect(topGObumps$Term, topGOcpg$Term))/50
```

```
## [1] 0.6
```

```
which(topGObumps$Term %in% intersect(topGObumps$Term, topGOcpg$Term))
```

```
##  [1]  1  2  3  4  5  6  9 10 11 13 14 15 17 18 19 22 23 24 26 29 33 35 36
## [24] 37 39 41 43 44 47 48
```

```
which(topGOcpg$Term %in% intersect(topGObumps$Term, topGOcpg$Term))
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 12 14 15 16 18 19 20 22 24 27 29 31 32
## [24] 34 37 38 41 43 47 49
```

Writing data for comparison of results

we can write out some dataframes that can be used in a separate Rmd file to compare results of the analyses.

Write out the Entrez IDs of genes belonging to both differentially methylated CpG and bumps

```
write.table(EntrezIDs, sep= "\t", file="EntrezIDs_CpG_results.txt")
write.table(EntrezIDs_bmp, sep= "\t", file="EntrezIDs_bmp_results.txt")
```

Write out the results of Gene Set analysis for both levels of analysis

```
CpG_GSA_res <- topGO(goanaOut, ontology="BP", number=100)
bmp_GSA_res <- topGO(goanaOut_bmp, ontology="BP", number=100)
write.table(CpG_GSA_res, sep= "\t", file="CpG_GSA_results.txt")
write.table(bmp_GSA_res, sep= "\t", file="bmp_GSA_results.txt")
```

APPENDIX D
COPY NUMBER VARIATION ANALYSIS

This appendix shows the R markdown code for the Copy Number Variation Analysis.

Copy Number Variation Analysis

```
library("CGHcall")
library("GEOquery")
library("CGHregions")
```

Read in data

```
data <- getGEO("GSE24282", destdir="Data/raw_data")
data <- data[[1]]

dim(fData(data))
```

```
## [1] 243430     13
```

```
dim(exprs(data))
```

```
## [1] 243430      6
```

```
dim(pData(data))
```

```
## [1]  6 45
```

Making data ready for CGHcall analysis.

The CGHcall pipeline requires a cghRaw object. We can make this object with the make_cghRaw function, which requires a dataframe where the first 4 columns are the name, chromosome and the start and end position of each target, and the other columns contain log2 ratios between target tissue and diploid reference for all of the samples.

```
head(exprs(data))
```

```
##      GSM597198    GSM597199   GSM597200  GSM597201    GSM597202
## 1  0.007747488  0.050739481  0.03613069 0.03908782  0.043336203
## 2  0.000000000 -0.017089667  0.33760363 0.00000000  0.000000000
## 3  0.000000000 -0.155284340  0.00000000 0.00000000  0.000000000
## 4 -0.015094979  0.003188478 -0.21272196 0.03350370  0.002166552
## 5  0.327186031  0.053453732  0.06008776 0.02570279  0.058134110
## 6 -0.002508904 -0.098609973 -0.07469510 0.02064522 -0.171498367
##      GSM597203
## 1  0.019856378
## 2 -0.385537858
## 3  0.000000000
## 4  0.026997013
## 5  0.060407406
## 6 -0.000694373
```

```
annot <- fData(data)
head(annot)
```

ID	COL	ROW	SPOT_ID	CONTROL_TYPE	GB_ACC	GENE_SYMBOL	GENE_NAME
1	267	912	HsCGHBrightCorner	pos			
2	267	910	DarkCorner	pos			
3	267	908	DarkCorner	pos			
4	267	906	A_16_P20527812	FALSE			
5	267	904	A_16_P01708709	FALSE	NM_138295	PKD1L1	polycystic kidney disease 1 like 1
6	267	902	A_16_P36062310	FALSE	NM_005235	ERBB4	v-erb-a erythroblastic leukemia viral oncogene homolog 4 (avian)

ACCESSION_STRING	CHROMOSOMAL_LOCATION	CYTOBAND	DESCRIPTION	GB_RANGE
	chr16:076331867-076331926	hs|q23.1		NC_000016.8[076331867..076331926]
ref|NM_138295|ref|NM_025031	chr7:047626734-047626793	hs|p12.3	Homo sapiens polycystic kidney disease 1 like 1 (PKD1L1), mRNA.	NC_000007.11[047626734..047626793]
ref|NM_005235	chr2:213225788-213225847	hs|q34	Homo sapiens verba erythroblastic leukemia viral oncogene homolog 4 (avian) (ERBB4), mRNA.	NC_000002.9[213225788..213225847]

```
Chr <- gsub(":.*","",annot$CHROMOSOMAL_LOCATION)
Loc <- gsub(".*:","",annot$CHROMOSOMAL_LOCATION)
Start <- as.numeric(gsub(".*-","",Loc))
```

```
## Warning: NAs introduced by coercion
```

```
End <- as.numeric(gsub("-.*","",Loc))
```

```
## Warning: NAs introduced by coercion
```

```
annot <- data.frame(annot$ID, Chr, Start, End)
colnames(annot)[1] <- "ID"

full_data <- cbind(annot, exprs(data))

head(full_data)
```

ID	Chr	Start	End	GSM597198	GSM597199	GSM597200	GSM597201	GSM597202	GSM597203
1		NA	NA	0.0077475	0.0507395	0.0361307	0.0390878	0.0433362	0.0198564
2		NA	NA	0.0000000	-0.0170897	0.3376036	0.0000000	0.0000000	-0.3855379
3		NA	NA	0.0000000	-0.1552843	0.0000000	0.0000000	0.0000000	0.0000000
4	chr16	76331926	76331867	-0.0150950	0.0031885	-0.2127220	0.0335037	0.0021666	0.0269970
5	chr7	47626793	47626734	0.3271860	0.0534537	0.0600878	0.0257028	0.0581341	0.0604074
6	chr2	213225847	213225788	-0.0025089	-0.0986100	-0.0746951	0.0206452	-0.1714984	-0.0006944

Only keep data coming from chromosomes 1 to 22.

```
dim(full_data)
```

```
## [1] 243430     10
```

```
full_data <- full_data[substr(Chr,4,6) %in% c(1:22),]
dim(full_data)
```

```
## [1] 225388     10
```

Make cghRaw object:

```
full_data$Chr <- substr(as.character(full_data$Chr),4,5)
full_data$Start <- as.integer(full_data$Start)
full_data$End <- as.integer(full_data$End)

#Filter out duplicated spots: spots with the same chromosome, start and end position.
full_data <- full_data[!(duplicated(paste(full_data$Chr,full_data$Start,full_data$End, sep="_"))),]
rownames(full_data) <- NULL
colnames(full_data)[5:10] <- paste(paste("s", c(1:6), sep=""), pData(data)$subtype)
            #give appropriate names to samples
```

```
cgh <- make_cghRaw(full_data)
```

Preprocessing & normalization

Preprocessing

Preprocessing

```
cgh_preproc <- preprocess(cgh)
```

Normalization

```
cgh_norm <- normalize(cgh_preproc)
```

```
## Applying median normalization ...
## Smoothing outliers ...
```

Segmentation

```
cgh_seg <- segmentData(cgh_norm)
```

```
## Start data segmentation ..
## Analyzing: Sample.1
## Analyzing: Sample.2
## Analyzing: Sample.3
## Analyzing: Sample.4
## Analyzing: Sample.5
## Analyzing: Sample.6
```

Normalization post segmentation

```
cgh_segPN <- postsegnormalize(cgh_seg)
```

Calling of CNV

Calling

```
cgh_res <- CGHcall(cgh_segPN, ncpus=3)
```

```
## EM algorithm started ...
## [1] "Total number of segments present in the data: 888"
## [1] "Number of segments used for fitting the model: 533"
## 236010830746821126.1234.6607018049877686324.2380.6433226249877169231.4380.6
## Calling iteration1:
## R Version:  R version 3.5.1 (2018-07-02)
## snowfall 1.84-6.1 initialized (using snow 0.4-3): parallel execution on 3 CPUs.
## Library CGHcall loaded.
## Library CGHcall loaded in cluster.
##
## Stopping cluster
## optim results
## time: 7
## minimum: 632980.219143658
## 6632971.341022809-0.73659305611179-0.193169423015762-0.001997658485672650.1458541284234020.2493392794300310.7
## 237817230791463127.1235607018049877686324.2380.6607018049877169324.2380.6
## Calling iteration2:
## snowfall 1.84-6.1 initialized (using snow 0.4-3): parallel execution on 3 CPUs.
```

```
## Library CGHcall loaded.
## Library CGHcall loaded in cluster.
##
## Stopping cluster
## optim results
## time: 6
## minimum: 632972.067538597
## 6632968.754314966-0.795613059334832-0.193129000758478-0.002339448801898270.1377417352225630.2354710516533130.
## EM algorithm done ...
## Computing posterior probabilities for all segments ...
## Total time:0minutes
cgh_res <- ExpandCGHcall(cgh_res,cgh_segPN)
## Adjusting segmented data for cellularity ...
## Cellularity sample1: 1
## Cellularity sample2: 1
## Cellularity sample3: 1
## Cellularity sample4: 1
## Cellularity sample5: 1
## Cellularity sample6: 1
## Adjusting normalized data for cellularity ...
## Cellularity sample1: 1
## Cellularity sample2: 1
## Cellularity sample3: 1
## Cellularity sample4: 1
## Cellularity sample5: 1
## Cellularity sample6: 1
## 1
## 238080932001330127.2244.2607018049877686324.2380.6607018049877169324.2380.6
## 238083333118710127.2252.7607018049877686324.2380.6607018049877169324.2380.6
## 238083933118730127.2252.7607018049877686324.2380.6607018049877169324.2380.6
## 238087334906530127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238089134906565127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238090534906590127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238091934906615127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238093334906640127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238094734906665127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238095834906689127.2266.4607018059933223324.2457.3607018049877169324.2380.6
## 238098435577159127.2271.5607018059933223324.2457.3607018049877169324.2380.6
```

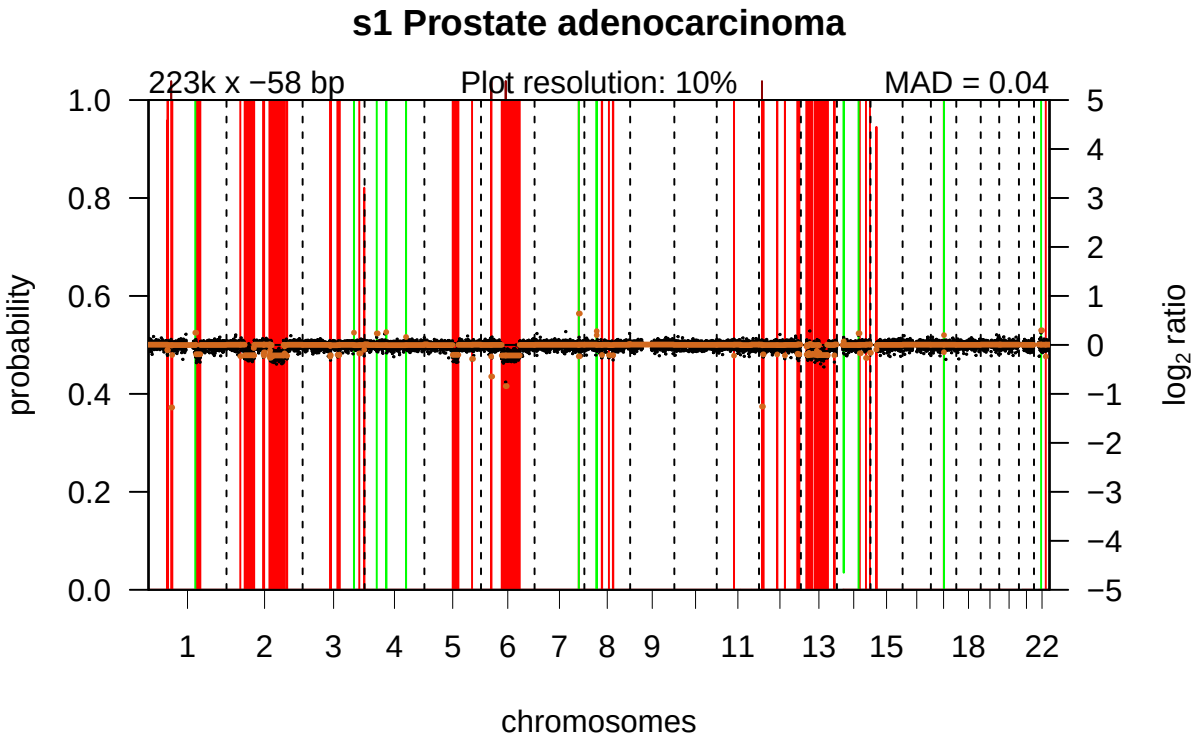
```
## 238141835577548127.2271.5607018059933223324.2457.3607018049877169324.2380.6
## 2
## 238144437141880127.2283.4607018059933223324.2457.3607018049877169324.2380.6
## 238145537141905127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238146137141925127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238147137365410127.2285.1607018059933223324.2457.3607018059898829324.2457
## 238147737141961127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238149137141986127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238150537142011127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238151937142036127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238153337142061127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238154437142085127.2283.4607018059933223324.2457.3607018059898829324.2457
## 238157037812555127.2288.5607018059933223324.2457.3607018059898829324.2457
## 238474839607940127.4302.2607018059933223324.2457.3607018059929802324.2457.3
## 3
## 238477040948798127.4312.5607018071999867324.2549.4607018059929802324.2457.3
## 238478140948823127.4312.5607018071999867324.2549.4607018071969212324.2549.1
## 238478740948843127.4312.5607018071999867324.2549.4607018071969212324.2549.1
## 238479741172338127.4314.2607018071999867324.2549.4607018071969212324.2549.1
## 238480040725416127.4310.8607018071999867324.2549.4607018071969212324.2549.1
## 238481440725441127.4310.8607018071999867324.2549.4607018071969212324.2549.1
## 238482840725466127.4310.8607018071999867324.2549.4607018071969212324.2549.1
## 238484240725491127.4310.8607018071999867324.2549.4607018071969212324.2549.1
## 238485640725516127.4310.8607018071999867324.2549.4607018071969212324.2549.1
## 238486740725540127.4310.8607018071999867324.2549.4607018071969212324.2549.1
## 238489341396010127.4315.9607018071999867324.2549.4607018071969212324.2549.1
## 238485241396003127.4315.9607018071999867324.2549.4607018071971326324.2549.1
## 4
## 238487645865472127.435060701807199867324.2549.4607018071971326324.2549.1
## 238488745865497127.435060701807199867324.2549.4607018071992355324.2549.3
## 238489545865519127.435060701808647984O324.2659.8607018071992355324.2549.3
## 238490345865541127.435060701808647984O324.2659.8607018086410553324.2659.3
## 238490945865561127.435060701808647984O324.2659.8607018086410553324.2659.3
## 238491946535996127.4355.1607018086479840324.2659.8607018086410553324.2659.3
## 238492246089082127.4351.7607018086479840324.2659.8607018086410553324.2659.3
## 238493646089107127.4351.7607018086479840324.2659.8607018086410553324.2659.3
## 238495046089132127.4351.7607018086479840324.2659.8607018086410553324.2659.3
## 238496446089157127.4351.7607018086479840324.2659.8607018086410553324.2659.3
```

```
## 238497846089182127.4351.7607018086479840324.2659.8607018086410553324.2659.3
## 238498946089206127.4351.7607018086479840324.2659.8607018086410553324.2659.3
## 238501548100502127.4367607018086479840324.2659.860701808641055324.2659.3
## 238497448100520127.4367607018086479840324.2659.8607018086426165324.2659.4
## FINISHED!
## Total time:2minutes
```

Visual representation

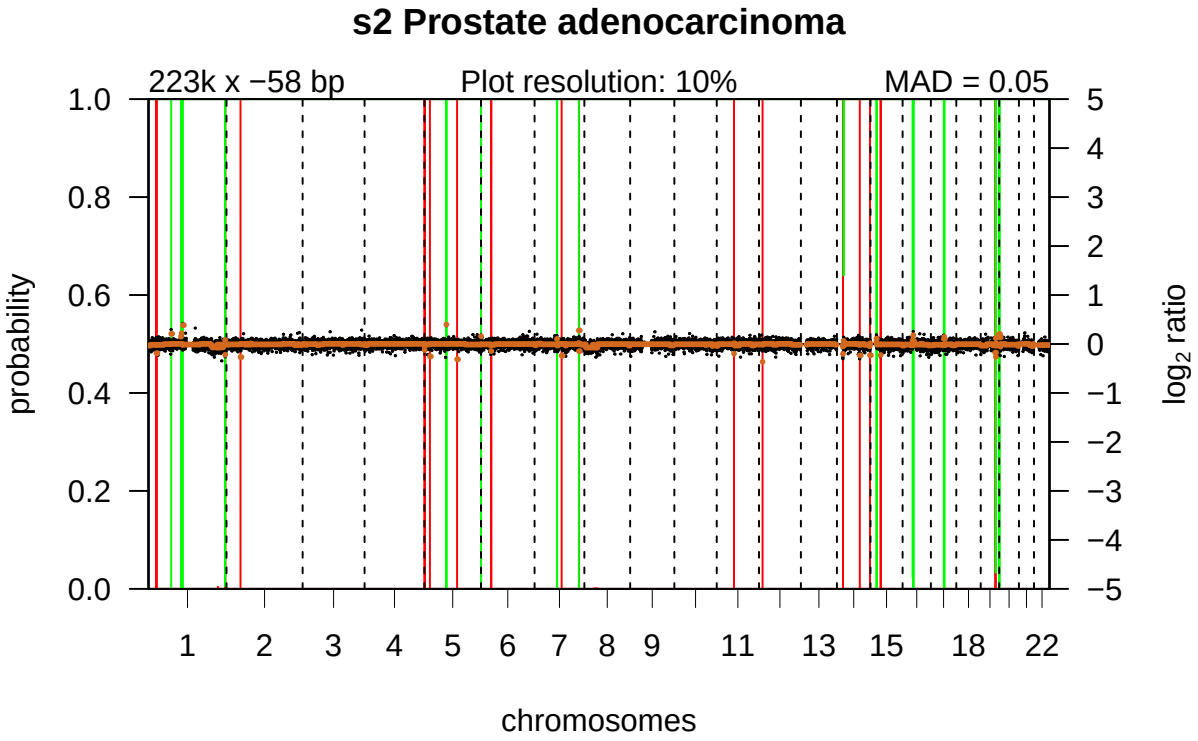
```
plot(cgh_res[,1])
```

```
## Plotting sample s1 Prostate adenocarcinoma
```



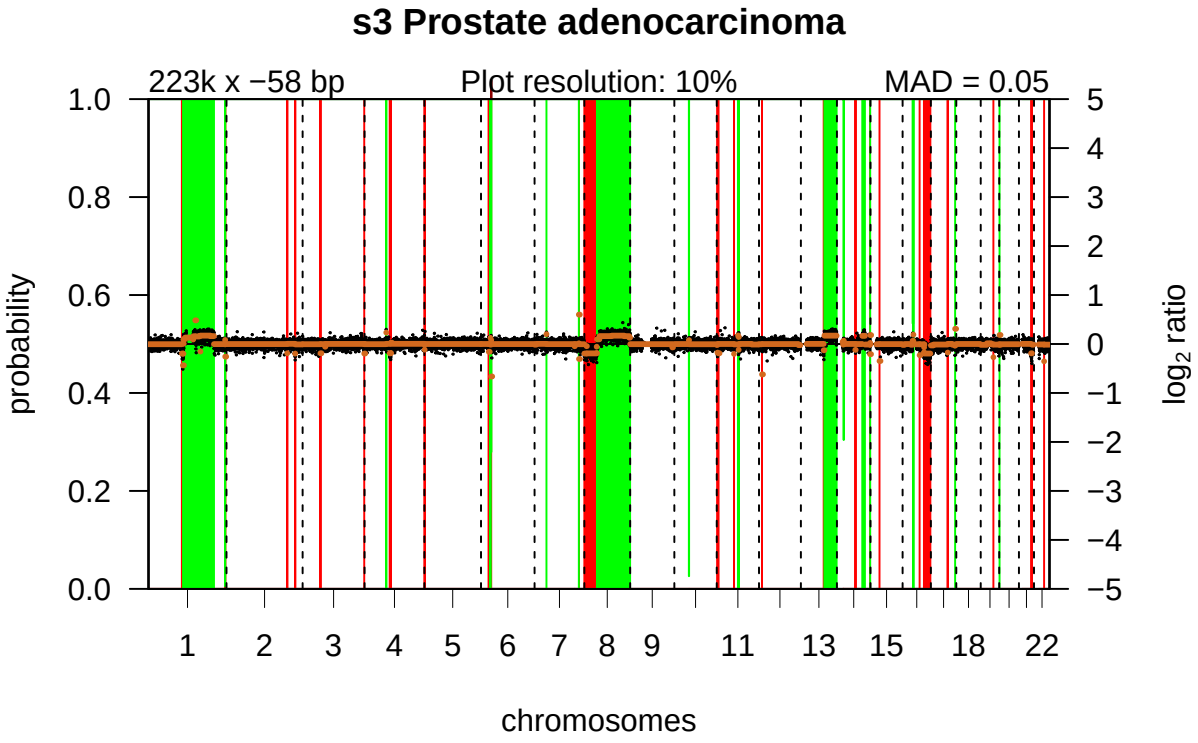
```
plot(cgh_res[,2])
```

```
## Plotting sample s2 Prostate adenocarcinoma
```



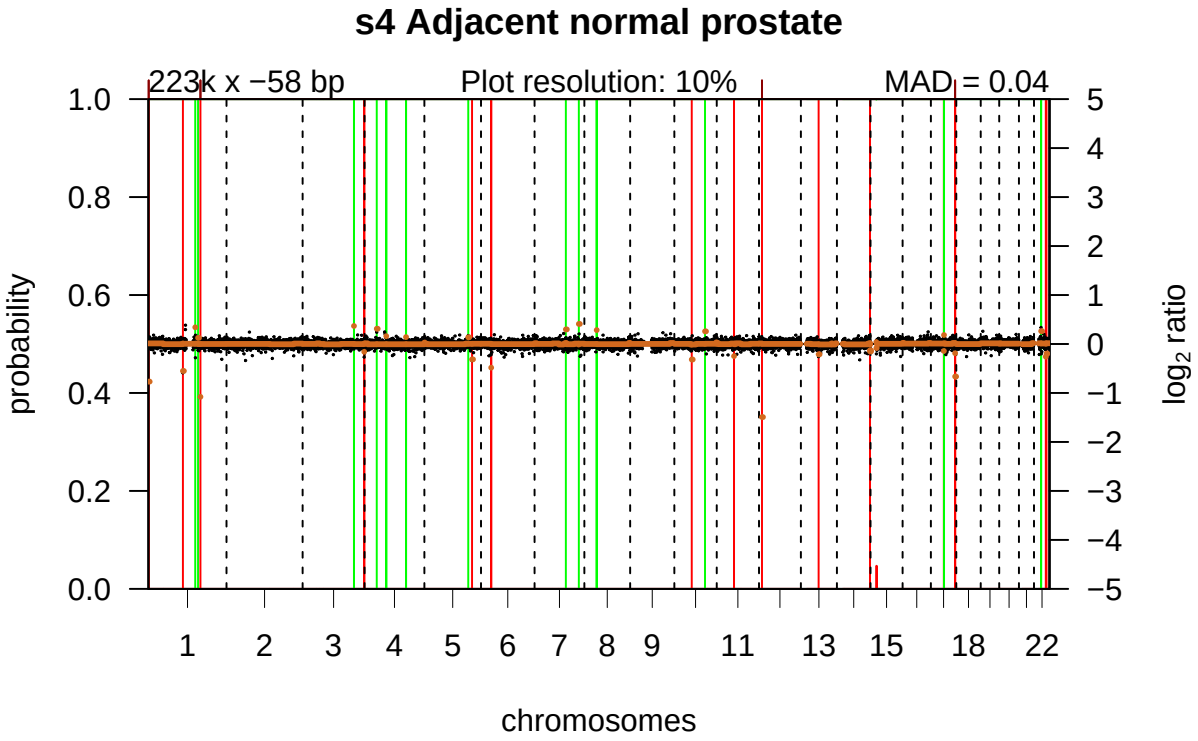
```
plot(cgh_res[,3])
```

```
## Plotting sample s3 Prostate adenocarcinoma
```



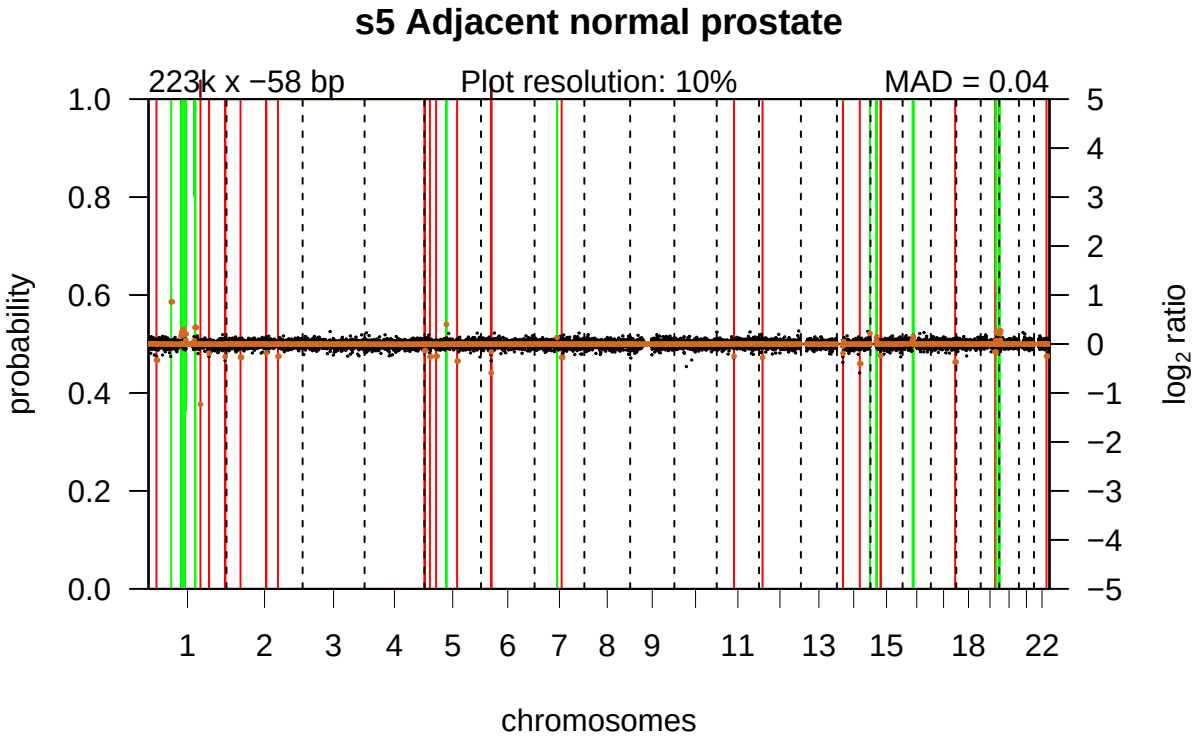
```
plot(cgh_res[,4])
```

```
## Plotting sample s4 Adjacent normal prostate
```



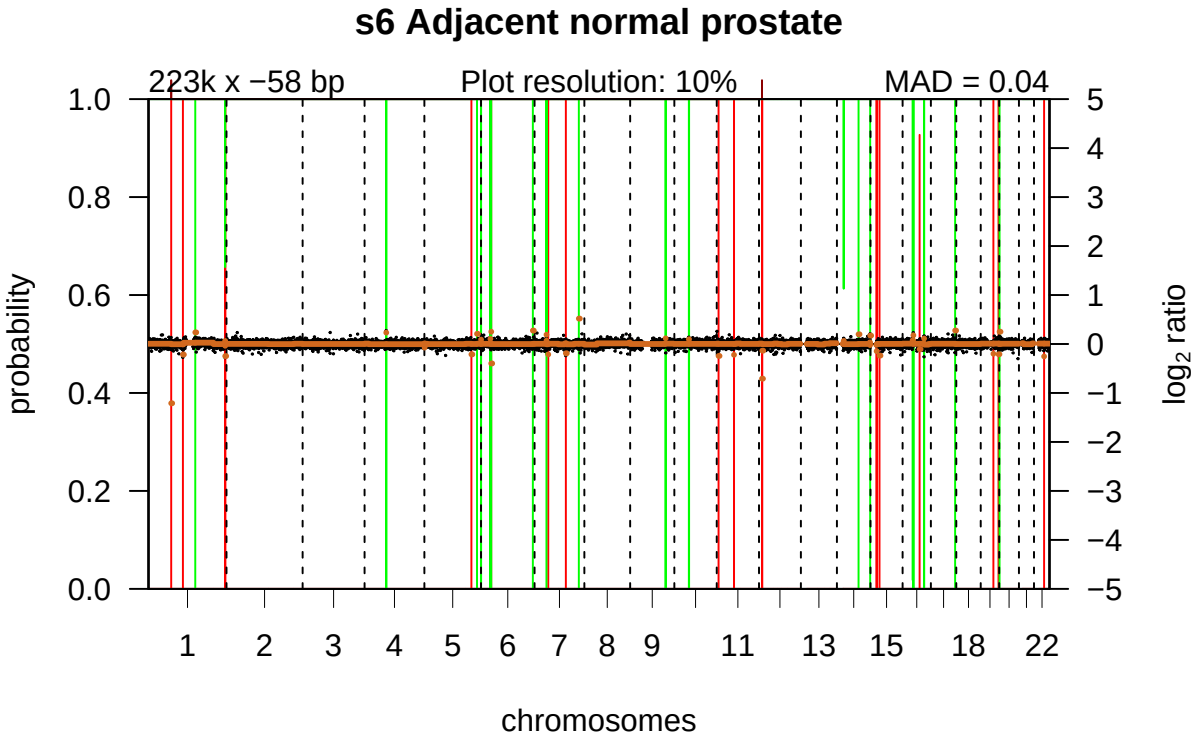
```
plot(cgh_res[,5])
```

```
## Plotting sample s5 Adjacent normal prostate
```



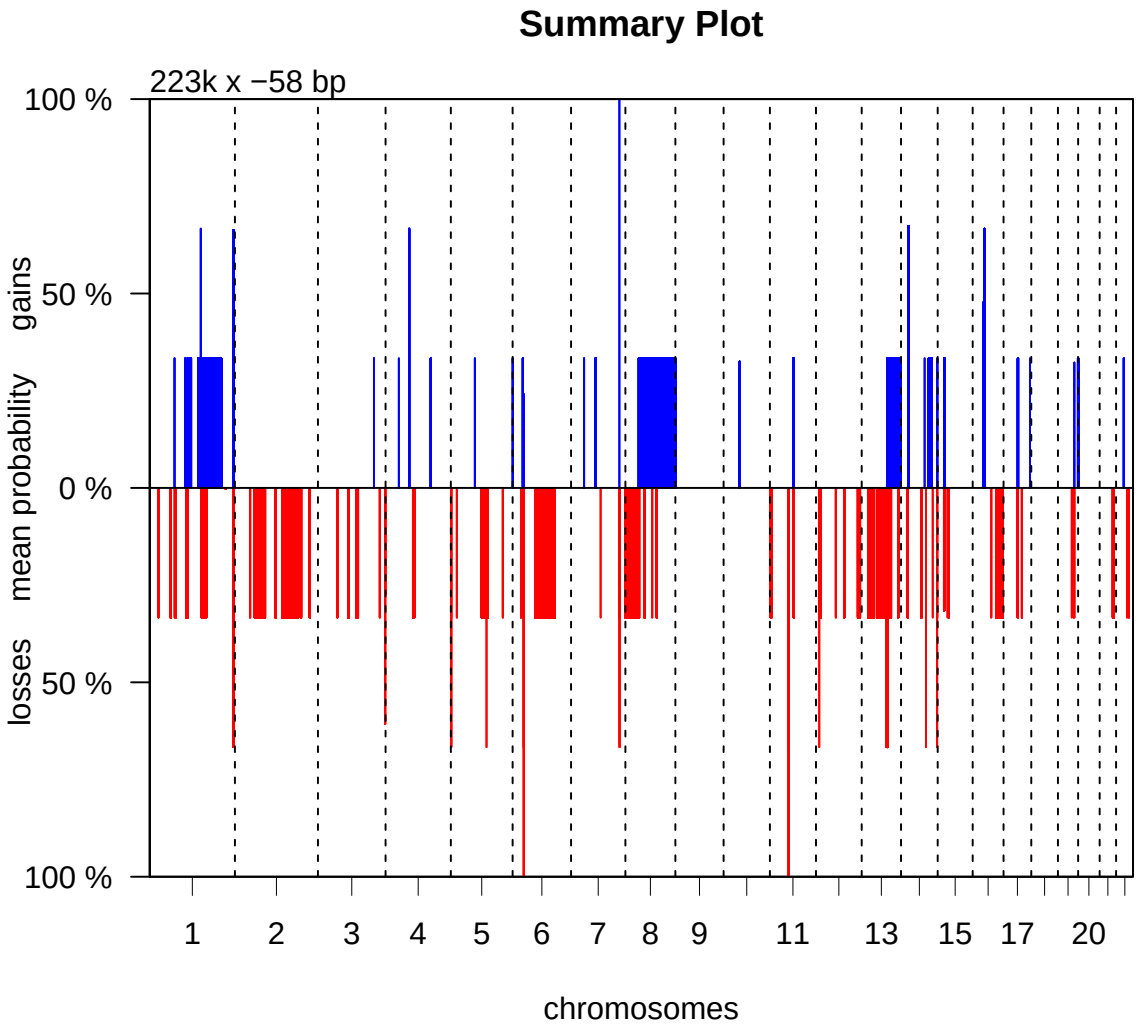
```
plot(cgh_res[,6])
```

```
## Plotting sample s6 Adjacent normal prostate
```

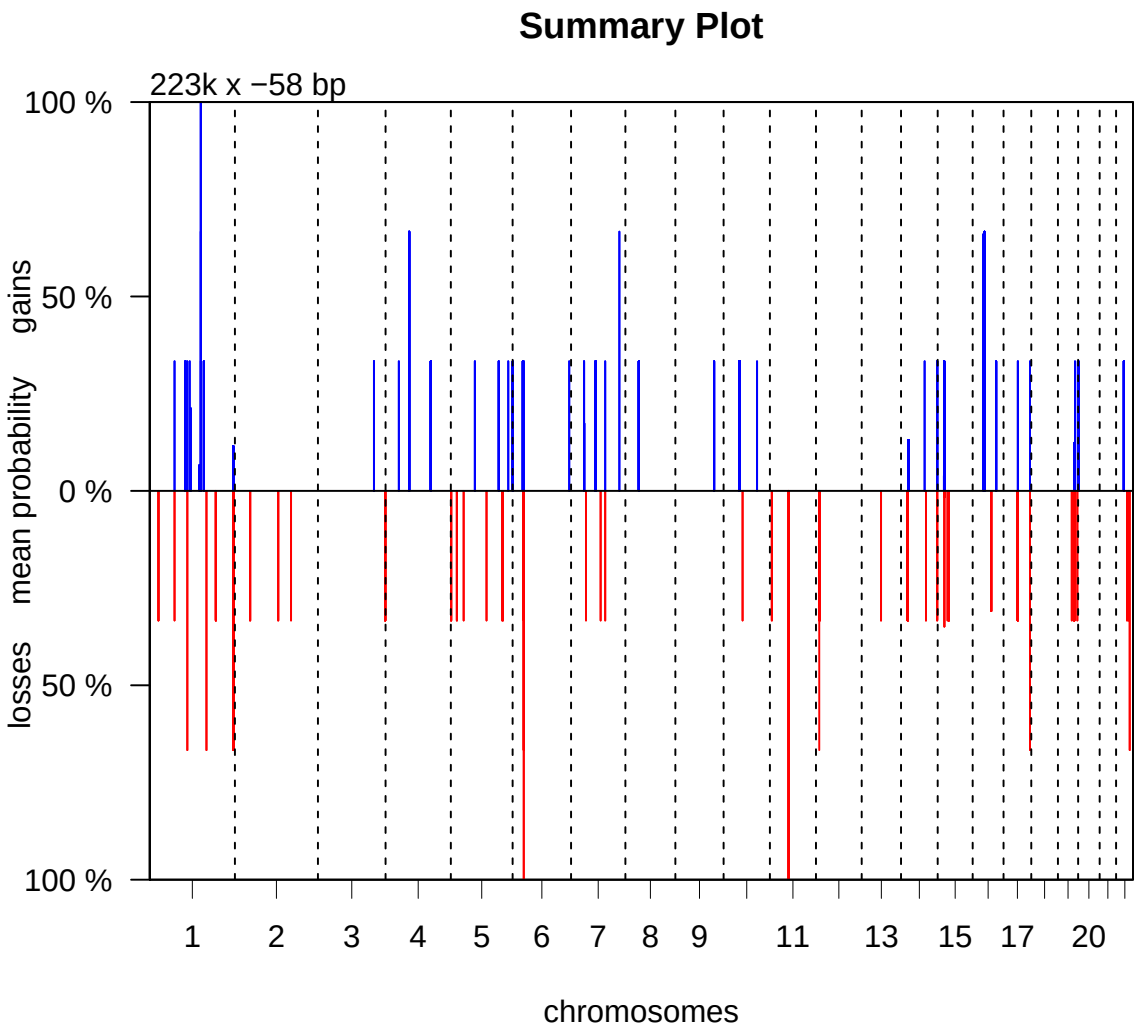


Plot summaries separately for tumor and healthy samples

```
summaryPlot(cgh_res[,1:3])
```



```
summaryPlot(cgh_res[,4:6])
```



CGHregions

CGHregions does dimensionality reduction of array CGH data. It takes as input an array CGH that has been segmented and called. It adjusts the segmentation so that break-points are in similar locations across multiple samples. It facilitates downstream analysis.

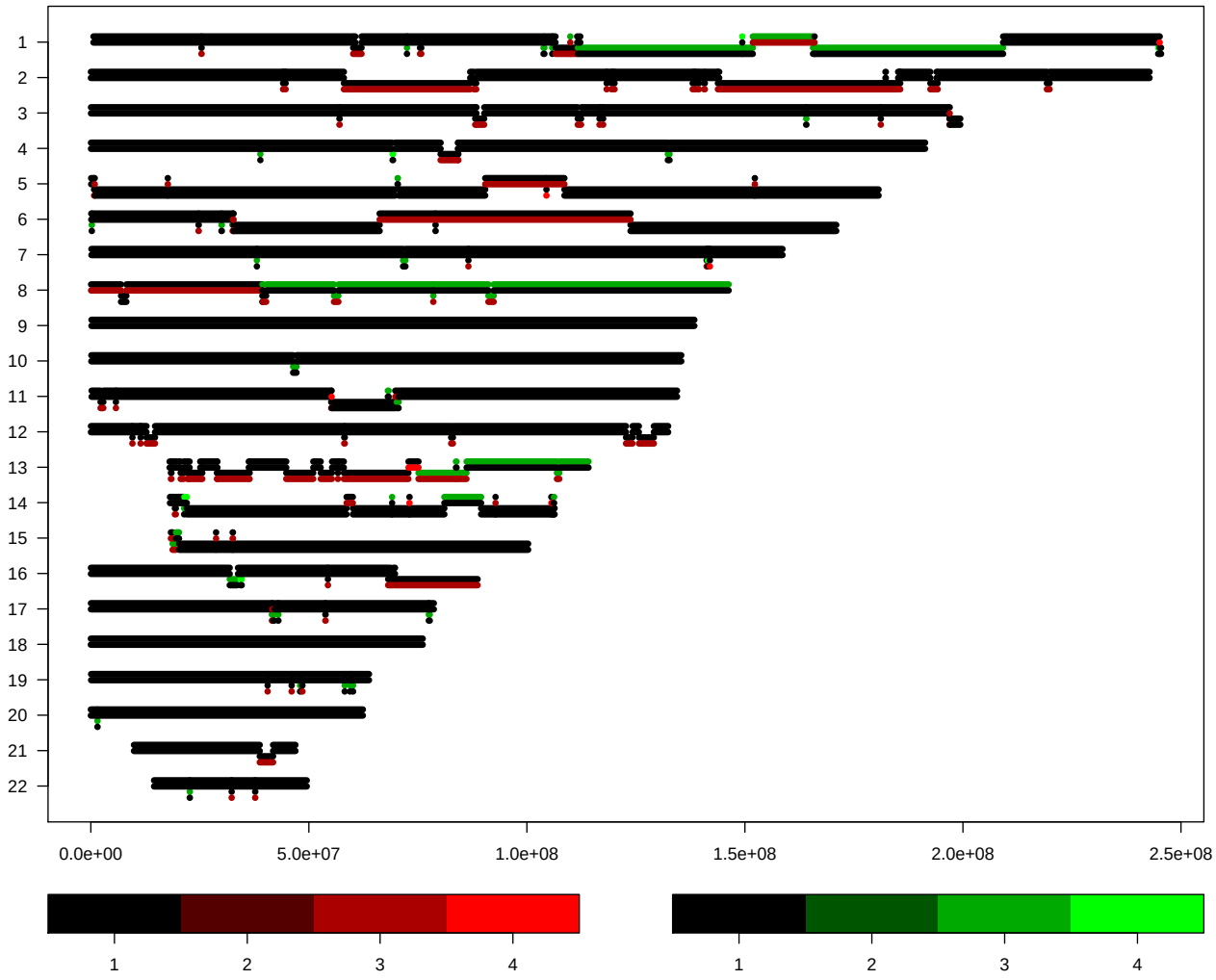
We can use CGHregions to determine genomic regions which have aberrant chromosomal copy number in our tumor samples.

```
regions <- CGHregions(cgh_res[,1:3]) #our first 3 samples are tumor samples
```

```
##         Samples
##       1       0       6
##         Samples
##       2       0       6
## [1] "Tuning on small data set finished...started with entire data set"
##               Samples
##  2.0000000  0.2003223 31.0000000  1.0000000
##               Samples
```

```
##  1.0000000  0.1489577 69.0000000
##         Samples
##       0       0     303
## [1] "c = 0, nr of regions: 303"
## [1] "Finished with entire data set."
```

```
plot(regions)
```



Look at output

```
head(res_regions_val <- regions(regions))
```

```
##   s1 Prostate adenocarcinoma s2 Prostate adenocarcinoma
## 1                          0                          0
## 2                          0                         -1
## 3                          0                          0
## 4                         -1                          0
## 5                          0                          0
## 6                         -1                          0
##   s3 Prostate adenocarcinoma
## 1                          0
## 2                          0
## 3                          0
## 4                          0
```

```
## 5                           0
## 6                           0
```

```
head(res_regions_loc <- fData(regions))
```

Chromosome	Start	End	Nclone	AveDist
1	604327	25330693	2377	0
1	25354824	25408867	6	0
1	25414785	60152251	3346	0
1	60161487	60352968	15	0
1	60407669	60651552	8	0
1	60685254	62079055	105	0

The Genomic regions where there is CNV (loss or gain) are useful in downstream analysis. We define a genomic region where there is CNV arbitrarily by subsetting for regions where the sum of region values is different from zero.

```
res_regions_val_CNV <- res_regions_val[abs(rowSums(res_regions_val))>0,]
res_regions_loc_CNV <- res_regions_loc[abs(rowSums(res_regions_val))>0,]
```

```
res_regions_loc_CNV$Chromosome <- paste("chr", res_regions_loc_CNV$Chromosome, sep="") #Add "chr" to chromosome
head(res_regions_loc_CNV)
```

	Chromosome	Start	End	Nclone	AveDist
2	chr1	25354824	25408867	6	0
4	chr1	60161487	60352968	15	0
6	chr1	60685254	62079055	105	0
8	chr1	72489740	72507501	2	0
10	chr1	75490712	75780238	29	0
12	chr1	103819610	104018710	8	0

We are interested in finding out which genes are in the regions where CNV was detected We can get gene symbols for these regions via use of annotation packages and genomicranges objects:

```
library(Homo.sapiens)
```

```
## Loading required package: AnnotationDbi
## Loading required package: stats4
## Loading required package: IRanges
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##     expand.grid
##
## Attaching package: 'IRanges'
## The following object is masked from 'package:grDevices':
##
##     windows
## Loading required package: OrganismDbi
## Loading required package: GenomicFeatures
```

```
## Loading required package: GenomeInfoDb
## Loading required package: GenomicRanges
## Loading required package: GO.db
##
## Loading required package: org.Hs.eg.db
##
## Loading required package: TxDb.Hsapiens.UCSC.hg19.knownGene
```

```
knownGenes <- genes(TxDb.Hsapiens.UCSC.hg19.knownGene) #get all known genes in a genomicranges object

GR_CNV <- makeGRangesFromDataFrame(res_regions_loc_CNV[,1:3]) #get our regions with CNV in a genomicranges objec

genes_CNV <- subsetByOverlaps(knownGenes, GR_CNV) #find overlaps between the two
head(genes_CNV)
```

```
## GRanges object with 6 ranges and 1 metadata column:
##             seqnames              ranges strand |     gene_id
##                <Rle>           <IRanges>  <Rle> | <character>
##          10     chr8   18248755-18258723      + |          10
##   100049076     chr5   69812079-70585523      - |   100049076
##       10010     chr2 161993466-162092683      + |       10010
##   100113402    chr16   70563402-70563502      + |   100113402
##   100113403     chr6 105384169-105388402      - |   100113403
##   100124535     chr2   69747177-69747303      - |   100124535
##   -------
##   seqinfo: 93 sequences (1 circular) from hg19 genome
```

We have this amount of genes that lie in regions with copy number variation:

```
length(genes_CNV$gene_id)
```

```
## [1] 2885
```

Write out results for comparison

We write our the genes which lie in regions for which copy number variation was detected

```
write.table(genes_CNV$gene_id, sep= "\t", file="geneID_CNV_results.txt")
```

APPENDIX E
COPY NUMBER VARIATION ANALYSIS

This appendix shows the R markdown code for the comparison of different results.

Result comparison

In this notebook, the comparison of results of data analysis of four data sources is compared.

Load in data

```
results_files <- list.files(pattern = ".*.txt")
for (i in 1:length(results_files)) { #loop through .txt files, assign them to their respective variables
  assign(sub("\\..*","",results_files[i]), read.table(results_files[i]))
}
```

Comparison of RNA-seq and expression micro-array

First of all, gene IDs must be the same for both results. For our RNA-seq experiment, we have Ensemble gene identifiers, whereas for our microarray experiment, we have gene symbols. We convert our gene symbol to Ensemble gene identifiers.

```
head(RNAseq_results)
```

	logFC	logCPM	LR	PValue	FDR
ENSG00000171401	-5.929711	4.863601	173.9383	0	0
ENSG00000142973	-3.457101	3.670703	159.3757	0	0
ENSG00000242110	4.496585	6.727269	151.0841	0	0
ENSG00000107485	-2.335815	4.442907	124.9034	0	0
ENSG00000204936	-5.836702	7.311235	116.7090	0	0
ENSG00000236699	1.982991	5.385785	114.7369	0	0

```
head(limmaExprsArray_results)
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
21027.ADCK5	0.6070928	6.904212	4.498916	0.0002172	0.9988071	-2.999836
21909.RET	-1.0655598	9.194311	-4.255301	0.0003839	0.9988071	-3.106784
1457.FAM89A	-0.7060515	4.711975	-4.029767	0.0006513	0.9988071	-3.209778
14630.TAF9B	0.7791750	2.882731	3.783337	0.0011595	0.9988071	-3.326386
18567.ADH6	-0.6625130	4.671936	-3.781837	0.0011636	0.9988071	-3.327108
13538.GPCPD1	-0.5177846	5.707743	-3.757274	0.0012323	0.9988071	-3.338950

```
library("org.Hs.eg.db")

ENS <- mapIds(org.Hs.eg.db, gsub(".*\\.","",rownames(limmaExprsArray_results)),'ENSEMBL', 'SYMBOL')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
ENS_uniq <- ENS[!(duplicated(ENS) | is.na(ENS))] #subset for non duplicated and mapped genes

Arr_filtered <- limmaExprsArray_results[!(duplicated(ENS) | is.na(ENS)),]
rownames(Arr_filtered) <- ENS_uniq
```

Look how many gene symbols we have that are present in both datasets:

```
present_in_both <- intersect(ENS_uniq, rownames(RNAseq_results))
length(present_in_both)
```

```
## [1] 11490
```

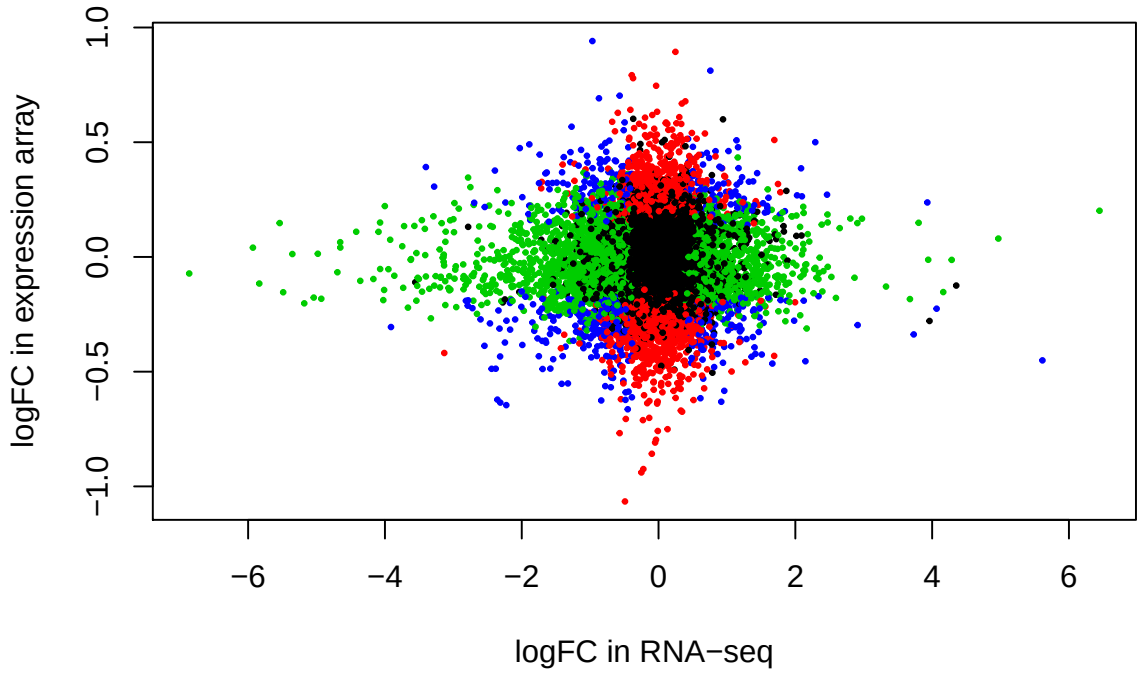
Filter datasets for gene Identifiers that are present in both datasets

```
RNAseq_filtered <- RNAseq_results[which(rownames(RNAseq_results) %in% present_in_both), ]
Arr_filtered <- Arr_filtered[which(gsub(".*\\.", "", rownames(Arr_filtered)) %in% present_in_both), ]

# same ordering
RNAseq_filtered <- RNAseq_filtered[order(rownames(RNAseq_filtered)), ]
Arr_filtered <- Arr_filtered[order(rownames(Arr_filtered)), ]
```

```
sign_RNA <- as.factor(RNAseq_filtered$FDR < 0.05)
sign_arr <- as.factor(Arr_filtered$P.Value < 0.20)

sign_combined <- as.factor(paste(as.double(sign_RNA),as.double(sign_arr), sep="."))
```

Make a scatter plot of logFC of the genes in both datasets. One expects to see a correlation where genes that have a certain logFC in one analysis have a similar logFC in the other.

```
plot(RNAseq_filtered$logFC, Arr_filtered$logFC, pch=20, xlab="logFC in RNA-seq",
     ylab="logFC in expression array", cex=0.50, col=sign_combined)
```



Look at how many genes have the same sign in their logFCs

```
print("Percentage of genes with same sign in their logFCs:")
```

```
paste(substr(as.character(sum(sign(RNAseq_filtered$logFC) == sign(Arr_filtered$logFC))/
                          length(RNAseq_filtered$logFC)*100),1,5), "%", sep="")
```

```
## [1] "Percentage of genes which are significantly DE for both analyses (blue in plot)
       with same sign in their logFCs:"
```

```
## [1] "52.25%"
```

```
print("Percentage of genes which are significantly DE for both analyses (blue in plot)
      with same sign in their logFCs:")
```

```
paste(substr(as.character(sum((sign(RNAseq_filtered$logFC) == sign(Arr_filtered$logFC))
            [as.character(sign_combined)=="2.2"])/sum(as.character(sign_combined)=="2.2")*100),1,5),
            "%", sep="")
```

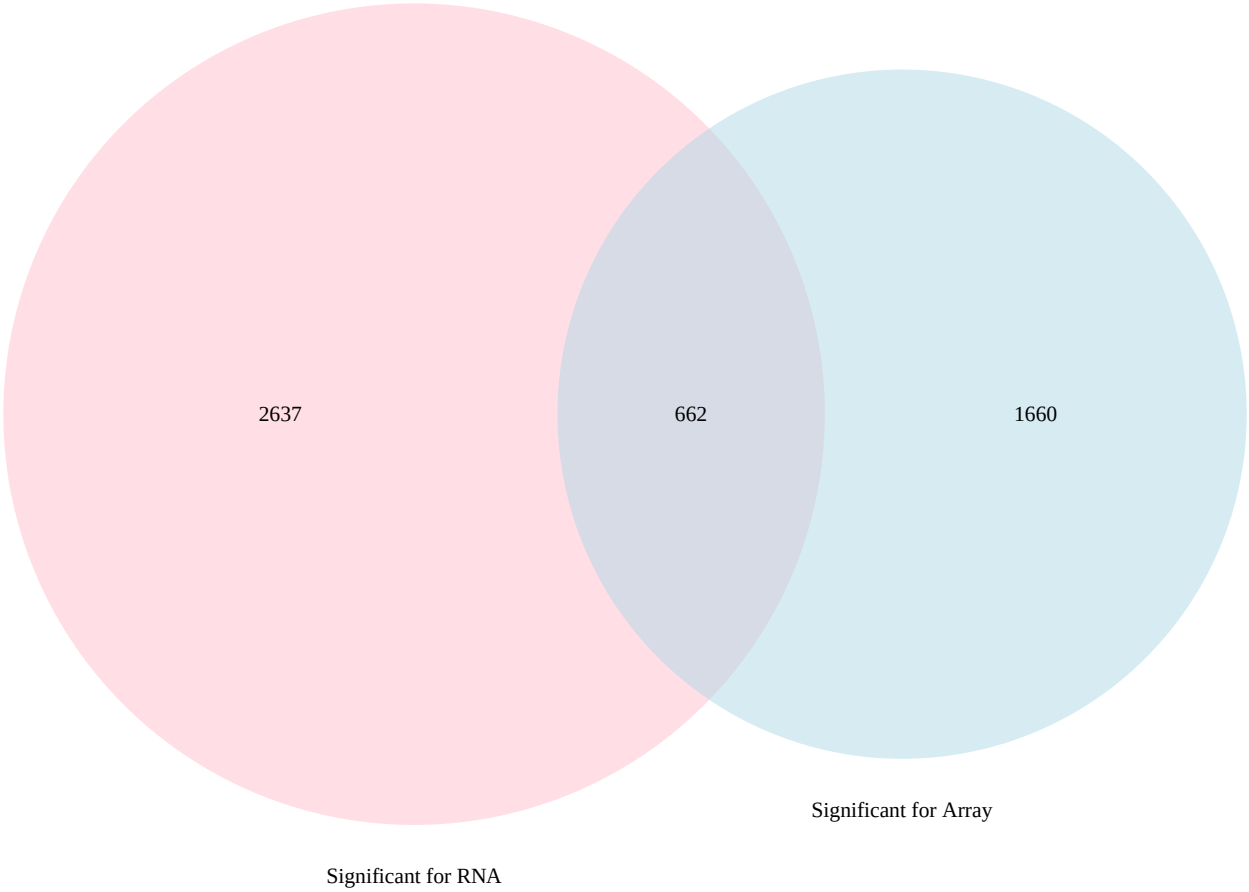
```
## [1] "Percentage of genes which are significantly DE for both analyses (blue in plot)
        with same sign in their logFCs:"
```

```
## [1] "56.49%"
```

Evaluate overlap between significant (being FDR < 0.05 for RNAseq and P-value<0.20 for Array) genes.

```
library("VennDiagram")
```

```
counts <- table(sign_combined)
grid.newpage()
draw.pairwise.venn(counts[2] + counts[4], counts[3] + counts[4], counts[4],
            category = c("Significant for Array", "Significant for RNA"),
            lty = rep("blank", 2), fill = c("light blue", "pink"),
            alpha = rep(0.5, 2), cat.pos = c(0, 0), cat.dist = rep(0.025, 2))
```



Comparison of expression analysis with differential methylation results

We first do the comparison with data coming from differential methylation analysis on regions,after that we do the same but on the differential methylation analysis on individual CpG sites.

Comparison with differentially methylated regions

First, we convert Entrez gene IDs from differential methylation analysis to Ensembl IDs.

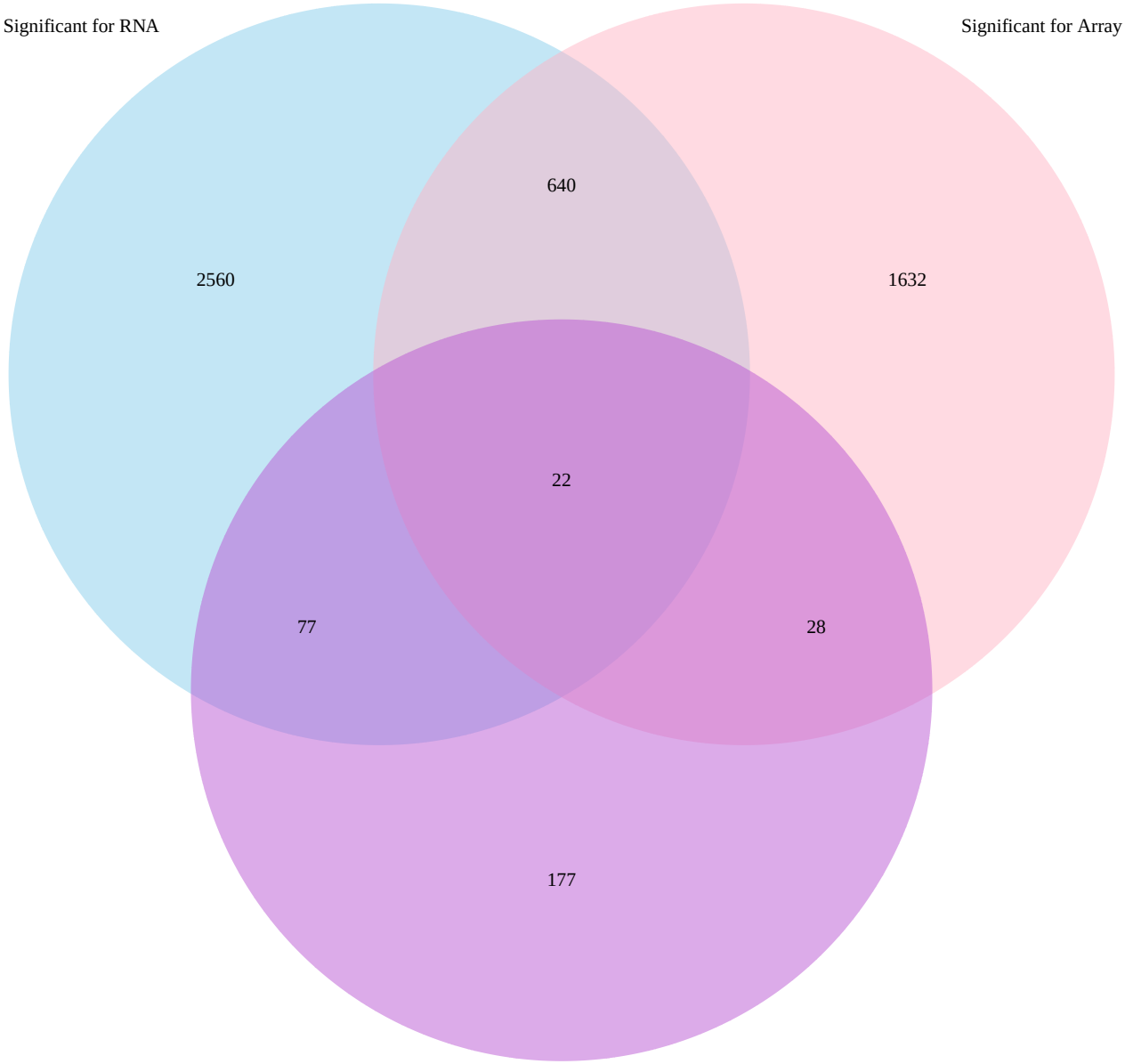
```
ENS_bumps <- mapIds(org.Hs.eg.db, as.character(EntrezIDs_bmp_results$x),'ENSEMBL', 'ENTREZID')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
ENS_bumps_uniq <- ENS_bumps[!(duplicated(ENS_bumps) | is.na(ENS_bumps))]
```

Look at the overlap with significant genes in RNAseq and expression microarray results:

```
ENS_RNA <- rownames(RNAseq_filtered[RNAseq_filtered$FDR<0.05,])
ENS_Arr <- rownames(Arr_filtered[Arr_filtered$P.Value<0.20,])

grid.newpage()
draw.triple.venn(area1 = length(ENS_RNA), area2 = length(ENS_Arr), area3 = length(ENS_bumps_uniq),
                 n12 = length(intersect(ENS_RNA,ENS_Arr)), n23 = length(intersect(ENS_bumps_uniq,ENS_Arr)),
                 n13 = length(intersect(ENS_RNA,ENS_bumps_uniq)),
                 n123 = length(intersect(intersect(ENS_RNA, ENS_Arr),ENS_bumps_uniq)),
                 category = c("Significant for RNA", "Significant for Array", "Significant differential
                 methylation (bumps)"),lty = "blank", fill = c("skyblue", "pink1", "mediumorchid"))
```


Significant for RNA
Significant for Array
640
2560
1632
22
77
28
177
Significant differential methylation (bumps)

Comparison with differentially methylated CpG sites

First, we convert Entrez gene IDs from differential methylation analysis to Ensembl IDs.

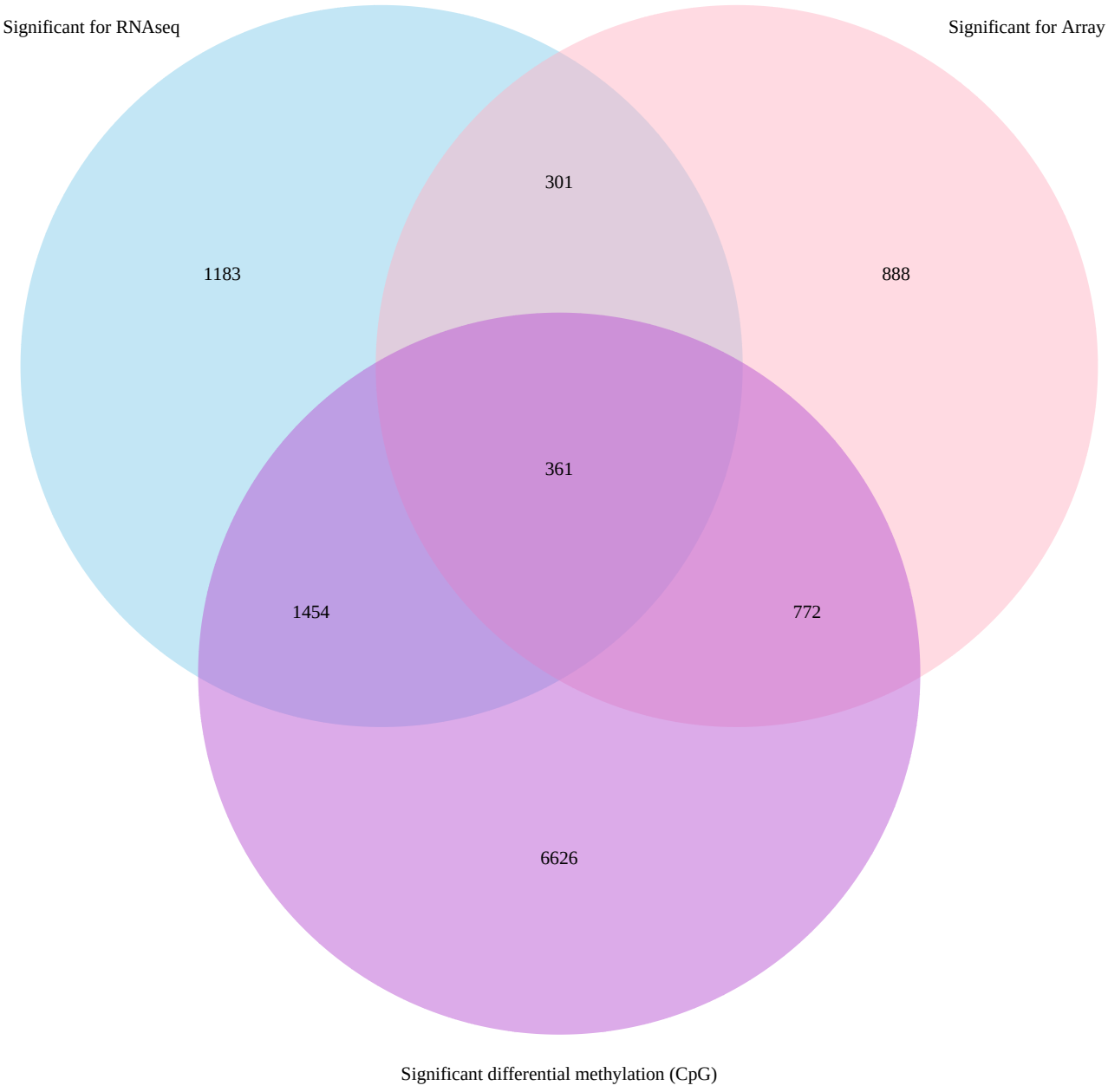
```
ENS_CpG <- mapIds(org.Hs.eg.db, as.character(EntrezIDs_CpG_results$x),'ENSEMBL', 'ENTREZID')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
ENS_CpG_uniq <- ENS_CpG[!(duplicated(ENS_CpG) | is.na(ENS_CpG))]
```

Look at the overlap with significant genes in RNAseq and expression microarray results:

```
ENS_RNA <- rownames(RNAseq_filtered[RNAseq_filtered$FDR<0.05,])
ENS_Arr <- rownames(Arr_filtered[Arr_filtered$P.Value<0.20,])

grid.newpage()
draw.triple.venn(area1 = length(ENS_RNA), area2 = length(ENS_Arr), area3 = length(ENS_CpG_uniq),
                 n12 = length(intersect(ENS_RNA,ENS_Arr)), n23 = length(intersect(ENS_CpG_uniq,ENS_Arr)),
                 n13 = length(intersect(ENS_RNA,ENS_CpG_uniq)),
                 n123 = length(intersect(intersect(ENS_RNA, ENS_Arr),ENS_CpG_uniq)),
                 category = c("Significant for RNAseq", "Significant for Array", "Significant differential
                 methylation (CpG)"), lty = "blank", fill = c("skyblue", "pink1", "mediumorchid"))
```



Comparison with genes where CNV was detected

Since in previous comparison differential methylated overlapped better with results coming from the RNAseq based analysis, we work further with this one.

First, we convert Entrez gene IDs from CNV analysis to Ensembl IDs.

```
ENS_CNV <- mapIds(org.Hs.eg.db, as.character(geneID_CNV_results$x),'ENSEMBL', 'ENTREZID')
```

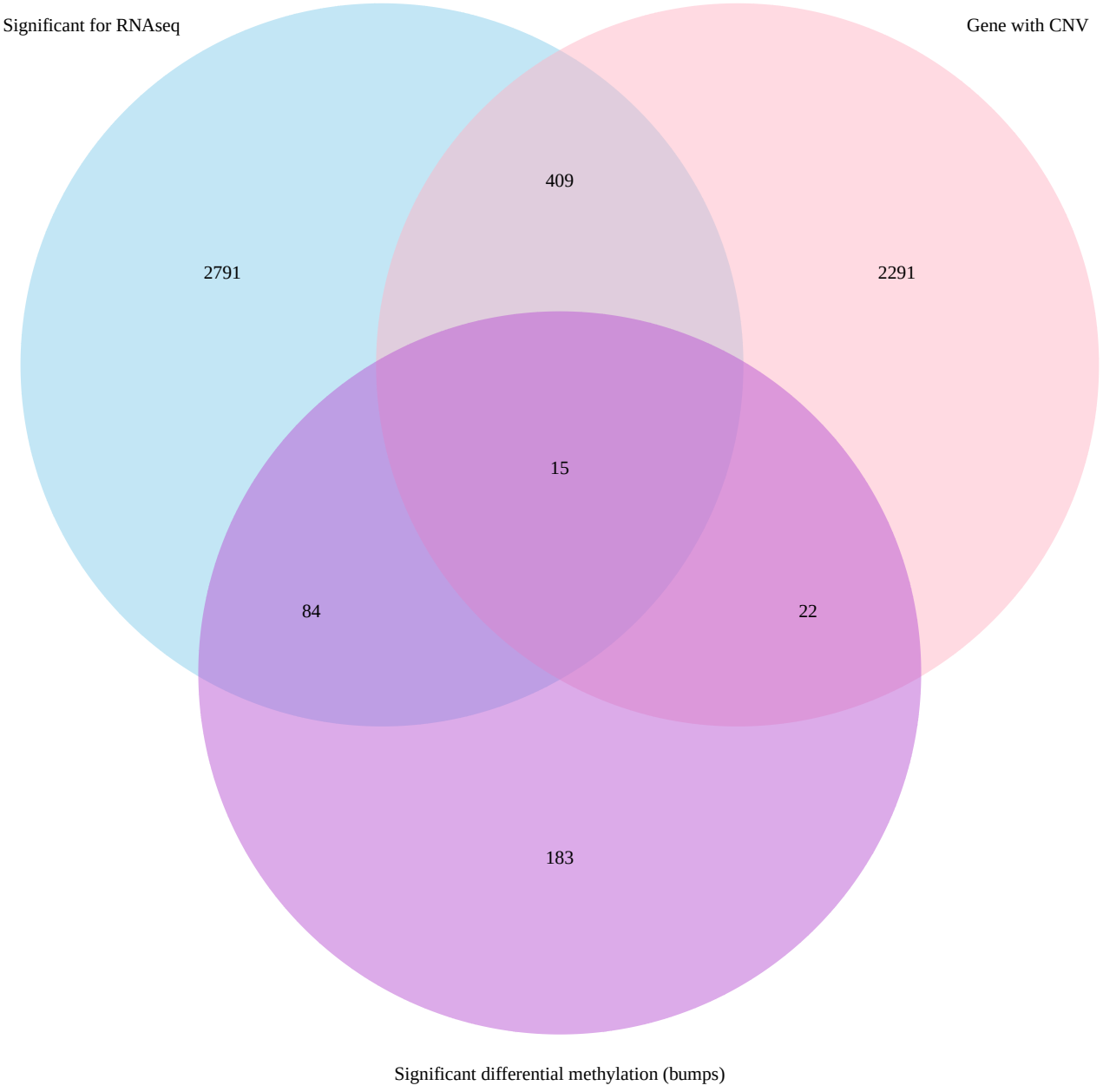
```
## 'select()' returned 1:many mapping between keys and columns
```

```
#subset for non duplicated and mapped genes
ENS_CNV_uniq <- ENS_CNV[!(duplicated(ENS_CNV) | is.na(ENS_CNV))]
```

Then we can make a venn diagram of overlap between RNAseq results, differential methylation results on bumps level and CNV

analysis results

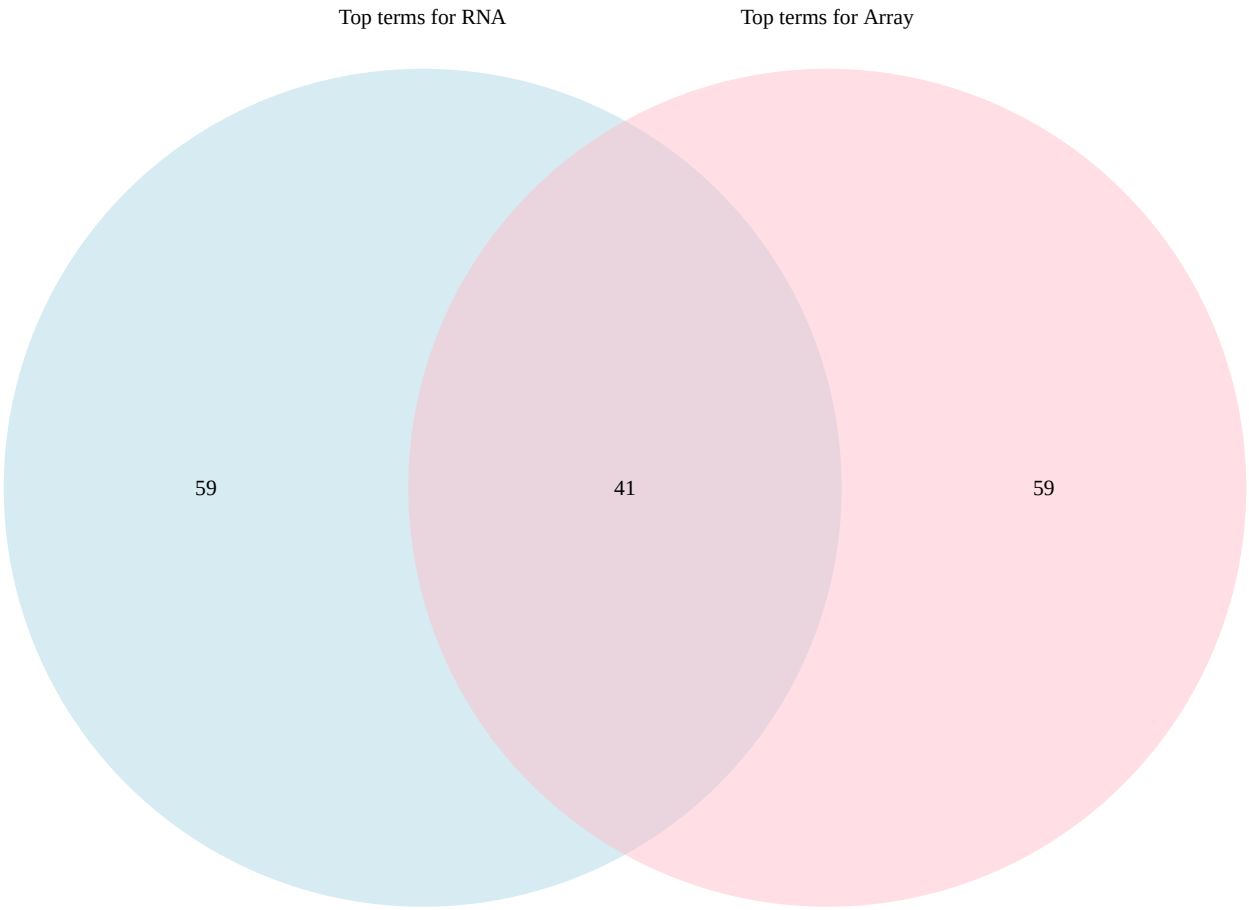
```
grid.newpage()
draw.triple.venn(area1 = length(ENS_RNA), area2 = length(ENS_CNV_uniq), area3 = length(ENS_bumps_uniq),
                 n12 = length(intersect(ENS_RNA,ENS_CNV_uniq)), n23 = length(intersect(ENS_bumps_uniq,ENS_CNV_ur
                 n13 = length(intersect(ENS_RNA,ENS_bumps_uniq)),
                 n123 = length(intersect(intersect(ENS_RNA, ENS_CNV_uniq),ENS_bumps_uniq)),
                 category = c("Significant for RNAseq", "Gene with CNV", "Significant differential
                 methylation (bumps)"), lty = "blank", fill = c("skyblue", "pink1", "mediumorchid"))
```



Comparison of Gene set analyses of different data sources

First of all, overlap between over-represented gene sets of differentially expressed genes for RNAseq & microarray based analysis is evaluated.

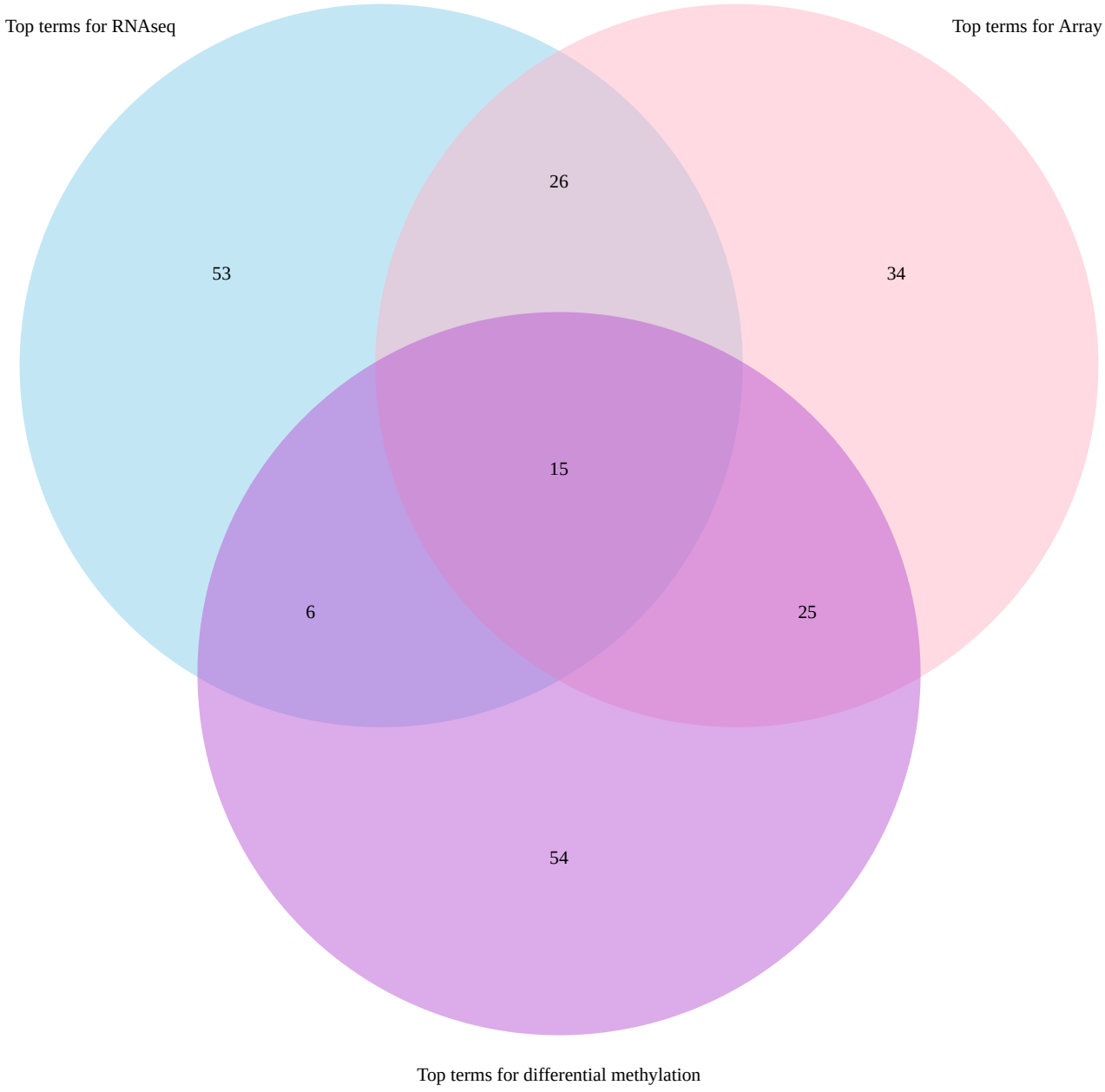
```
RNAterms <- array_GSA_results$Term
Arrterms <- RNAseq_GSA_results$Term

grid.newpage()

draw.pairwise.venn(length(RNAterms), length(Arrterms), length(intersect(RNAterms,Arrterms)),
                   category = c("Top terms for RNA", "Top terms for Array"),
                   lty = rep("blank", 2), fill = c("light blue", "pink"), alpha = rep(0.5, 2),
                   cat.pos = c(0, 0), cat.dist = rep(0.025, 2))
```



This has a considerably larger overlap than on gene level. A critical evaluation is in order: not necessarily the same genes are differentially expressed, but the biological functions of the differentially expressed genes remain. But: Top GO terms are more general ones, and these are bound to be overrepresented in a lot of cases, so overlap in those GO terms doesn't tell a lot.

Now we can add the top GO terms for differential methylation to the evaluation: (Only done on region level as this makes most sense biologically).

```
Methterms <- bmp_GSA_results$Term

grid.newpage()
draw.triple.venn(area1 = length(RNAterms), area2 = length(Arrterms), area3 = length(Methterms),
                 n12 = length(intersect(RNAterms,Arrterms)), n23 = length(intersect(Methterms,Arrterms)),
                 n13 = length(intersect(RNAterms,Methterms)),
                 n123 = length(intersect(intersect(RNAterms, Arrterms),Methterms)),
                 category = c("Top terms for RNAseq", "Top terms for Array", "Top terms for differential
                 methylation"), lty = "blank", fill = c("skyblue", "pink1", "mediumorchid"))
```



Genes and Terms of interest

From the above evaluation, we can take some genes and GO Terms of special interest. Particularly, we are interested in those genes that were both differentially expressed in both DE analyses and differentially methylated. Also, GO Terms that were found to be overlapping in the three analyses are of interest.

```
print("Ensembl gene IDs that are significant for the three (2 DE and 1 DM) analyses:")
```

```
## [1] "Ensembl gene IDs that are significant for the three (2 DE and 1 DM) analyses:"
```

```
intersect(intersect(ENS_RNA, ENS_Arr), ENS_bumps_uniq)
```

```
##  [1] "ENSG00000012779" "ENSG00000105971" "ENSG00000108352"
##  [4] "ENSG00000124839" "ENSG00000131759" "ENSG00000134824"
##  [7] "ENSG00000138356" "ENSG00000141574" "ENSG00000145284"
## [10] "ENSG00000147526" "ENSG00000148908" "ENSG00000152518"
## [13] "ENSG00000154188" "ENSG00000158055" "ENSG00000159674"
## [16] "ENSG00000162551" "ENSG00000163497" "ENSG00000165078"
## [19] "ENSG00000169862" "ENSG00000171056" "ENSG00000176532"
## [22] "ENSG00000184113"
```

```
print("top GO terms overlapping for the three (2 DE and 1 DM) analyses:")
```

```
## [1] "top GO terms overlapping for the three (2 DE and 1 DM) analyses:"
```

```
intersect(intersect(RNAterms, Arrterms), Methterms)
```

```
##  [1] "positive regulation of biological process"
##  [2] "multicellular organism development"
##  [3] "positive regulation of cellular process"
##  [4] "negative regulation of cellular process"
##  [5] "anatomical structure development"
##  [6] "developmental process"
##  [7] "regulation of signaling"
##  [8] "system development"
##  [9] "regulation of cell communication"
## [10] "regulation of cellular process"
## [11] "nervous system development"
## [12] "anatomical structure morphogenesis"
## [13] "cellular developmental process"
## [14] "animal organ morphogenesis"
## [15] "regulation of signal transduction"
```

Extract data on Genes and Terms of interest (GOI & TOI):

```
GOI <- RNAseq_results[which(rownames(RNAseq_results) %in%
                      intersect(intersect(ENS_RNA, ENS_Arr), ENS_bumps_uniq)), ]
rownames(GOI) <- mapIds(org.Hs.eg.db, rownames(GOI), 'SYMBOL', 'ENSEMBL') #convert names to gene symbols

POI <- RNAseq_GSA_results[which(RNAseq_GSA_results$Term %in%
                         intersect(intersect(RNAterms, Arrterms), Methterms)), ]

head(GOI, n=10)
```

	logFC	logCPM	LR	PValue	FDR
SOX7	-2.313468	3.800819	90.83460	0e+00	0.0e+00
AOX1	-2.169489	5.188479	77.81864	0e+00	0.0e+00
CAV2	-1.217255	5.385426	54.75269	0e+00	0.0e+00
CPA6	-2.380225	2.798609	49.90950	0e+00	0.0e+00
ALPL	-1.834373	3.115836	35.87081	0e+00	1.0e-07
ZFP36L2	-1.252946	7.148720	32.71724	0e+00	6.0e-07
RAB17	1.380268	3.747938	31.59111	0e+00	9.0e-07
RAPGEFL1	-1.205473	3.126515	28.86572	1e-07	3.0e-06
SPON2	2.463804	8.761226	28.35629	1e-07	3.7e-06
SECTM1	-1.535684	2.854525	26.76899	2e-07	7.2e-06

```
head(POI, n=10)
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0032502	developmental process	BP	6212	1588	0	0
GO:0048856	anatomical structure development	BP	5810	1498	0	0
GO:0007275	multicellular organism development	BP	5330	1378	0	0
GO:0048731	system development	BP	4783	1250	0	0
GO:0050794	regulation of cellular process	BP	10828	2506	0	0
GO:0009653	anatomical structure morphogenesis	BP	2604	747	0	0
GO:0048869	cellular developmental process	BP	4282	1119	0	0
GO:0048522	positive regulation of cellular process	BP	5330	1343	0	0
GO:0007399	nervous system development	BP	2296	659	0	0
GO:0048523	negative regulation of cellular process	BP	4759	1214	0	0