Genome-wide RNA profiling of long-lasting stem-cell like memory CD8 T cells induced by Yellow Fever vaccination in humans - Supplementary Material

Silvia A. Fuertes Marraco, Charlotte Soneson, Mauro Delorenzi and Daniel E. Speiser

This document provides the R code that was used to process the gene expression data from five T cell populations in each of 8 human donors vaccinated against yellow fever (YF). The gene expression measurements were obtained using the Agilent Whole Human Genome microarray 8x60K v2 platform (one-color) and pre-processed with the Agilent Feature Extraction software. The code below assumes that the raw data archive from `http://www.dtd.nlm.nih.gov/geo/query/acc.cgi?acc=GSE65804` has been downloaded and extracted in the current working directory (so that there is a folder named "`GSE65804_RAW`" containing the individual data files), and that the text files in this folder have been extracted as well (i.e., that they are ending with the suffix ".txt" rather than ".txt.gz"). Moreover, it assumes that the files "`Targets.txt`" and "`sampleInfo.txt`" (provided as `Supplementary Material 2 and 3`) are available in the current working directory.

```
library(limma)
library(ggplot2)
library(gplots)
library(xtable)

data.dir <- "GSE65804_RAW"
```

1 Read and normalize raw data, quality control

The code below reads the raw data, performs background correction and normalizes the values between samples, using functions from the `limma` package. We set the unique sample identifiers (the column names) to the combination of the number of the microarray and the position of the sample within the slide.

```
targets <- readTargets("Targets.txt")
## Read raw data
raw.data <- read.maimages(targets, source = "agilent",
                          path = data.dir,
                          green.only = TRUE,
                          other.columns = c("gIsPosAndSignif",
                              "gIsWellAboveBG", "chr_coord"))
colnames(raw.data) <- sapply(colnames(raw.data), function(y) {
    v <- strsplit(y, "_")[[1]]
```

```
    paste(v[2], "_", v[7], ".", v[8], sep = "")
})
raw.data$genes$chr_coord <- raw.data$other$chr_coord[, 1]

## Background correction and normalization between arrays.
raw.data_BG <- backgroundCorrect(raw.data, method = "normexp",
                                 offset = 0)
norm.data <- normalizeBetweenArrays(raw.data_BG,
                                    method = "quantile")
```

The Agilent Feature Extraction Software returns, in addition to the intensity values, flags that indicate whether a given feature is detected above the corresponding background level. We use this information to filter out all probes that are not detected well above the background (defined as the estimated background level + 2.6 times the estimated standard deviation of the background) for any of the samples.

```
nbr.pos.sig <- apply(norm.data$other$gIsWellAboveBG, 1, sum)
X <- norm.data[nbr.pos.sig > 0, ]
```

This reduces the number of features from 62976 to 50477. Finally, we average the expression of replicated probes, and filter out control probes.

```
X <- avereps(X, ID = X$genes$ProbeName)
X <- X[X$genes$ControlType == 0, ]
```

After these processing steps, the data set contains 41923 features. This data matrix corresponds to the processed data table deposited in the Gene Expression Omnibus record (`http://www.dtd.nlm.nih.gov/geo/query/acc.cgi?acc=GSE65804`).

```
X$E[1:8, 1:3]

##               253949422880_1.1 253949422880_1.2 253949422880_1.3
## A_23_P117082          6.397027         5.996913         6.749524
## A_33_P3318220         3.451226         3.400895         3.981863
## A_33_P3236322         9.327483         9.106760         7.201390
## A_21_P0000509         4.752888         5.040569         5.024037
## A_21_P0000744         4.096988         6.674370         8.029322
## A_23_P110167          4.068921         3.455562         3.466682
## A_33_P3211513        13.078076        13.652925        14.443181
## A_23_P103349          6.445124         7.685081         5.732062
```

2 Sample and variable annotations

We next create the sample annotation table, containing information about the microarray number, the position within the slide, the original sample name, the donor, the cell population and the sorting day. This information is read from the file `sampleInfo.txt`.

```
sample.annot <- read.delim("sampleInfo.txt", header = TRUE,
                           colClasses = c(rep("character", 7), "numeric"))
rownames(sample.annot) <- paste(sample.annot$microarray,
```

```
                                    sample.annot$position,
                                    sep = "_")
sample.annot <- sample.annot[match(colnames(X),
                                   rownames(sample.annot)), ]
sample.annot$color <-
  c(rgb(242, 232, 34, maxColorValue = 255),
    rgb(72, 167, 72, maxColorValue = 255),
    rgb(199, 24, 33, maxColorValue = 255),
    rgb(54, 153, 202, maxColorValue = 255),
    rgb(118, 67, 130, maxColorValue = 255))[match(sample.annot$celltype,
                                                  c("A2_N24b_naivelike",
                                                    "total_cm",
                                                    "total_effector",
                                                    "total_naive",
                                                    "total_Tscm"))]
```

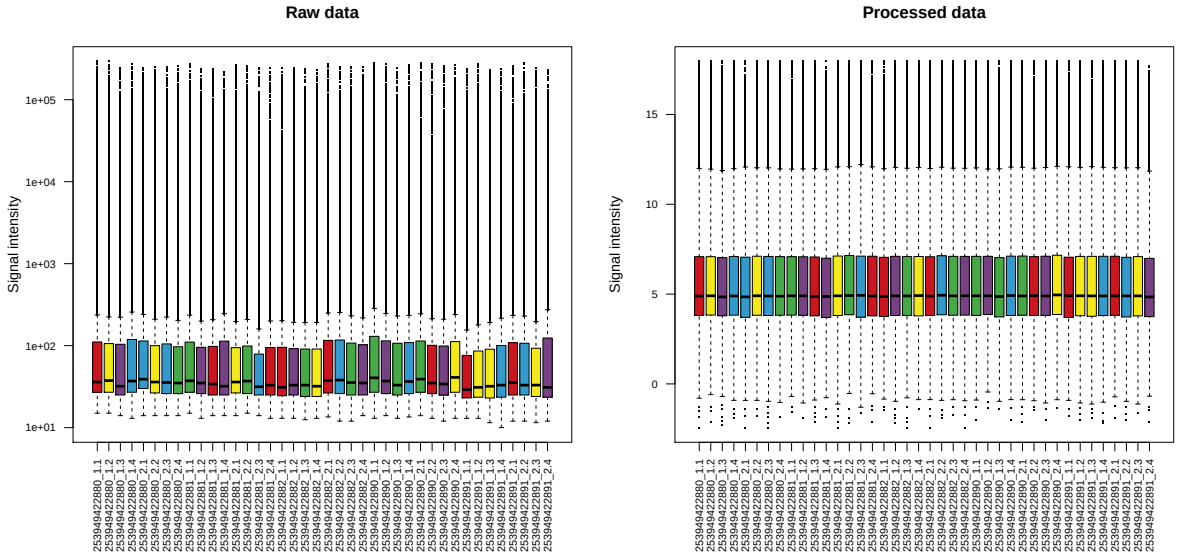
```
sample.annot
```

```
##                     microarray position sampleName          celltype donor sortday gender yrsvacc   color
## 253949422880_1.1 253949422880      1.1        4.5    total_effector    d4       I   male     7.9 #C71821
## 253949422880_1.2 253949422880      1.2        7.1 A2_N24b_naivelike    d7      II   male    12.7 #F2E822
## 253949422880_1.3 253949422880      1.3        5.3        total_Tscm    d5      II female     8.7 #764382
## 253949422880_1.4 253949422880      1.4        6.2       total_naive    d6       I female    12.1 #3699CA
## 253949422880_2.1 253949422880      2.1        8.2       total_naive    d8      II   male    17.8 #3699CA
## 253949422880_2.2 253949422880      2.2        2.1 A2_N24b_naivelike    d2      II female     4.1 #F2E822
## 253949422880_2.3 253949422880      2.3        3.2       total_naive    d3       I female     7.2 #3699CA
## 253949422880_2.4 253949422880      2.4        1.4          total_cm    d1       I female     2.4 #48A748
## 253949422881_1.1 253949422881      1.1        6.4          total_cm    d6       I female    12.1 #48A748
## 253949422881_1.2 253949422881      1.2        2.3        total_Tscm    d2      II female     4.1 #764382
## 253949422881_1.3 253949422881      1.3        1.5    total_effector    d1       I female     2.4 #C71821
## 253949422881_1.4 253949422881      1.4        3.3        total_Tscm    d3       I female     7.2 #764382
## 253949422881_2.1 253949422881      2.1        5.1 A2_N24b_naivelike    d5      II female     8.7 #F2E822
## 253949422881_2.2 253949422881      2.2        4.4          total_cm    d4       I   male     7.9 #48A748
## 253949422881_2.3 253949422881      2.3        7.2       total_naive    d7      II   male    12.7 #3699CA
## 253949422881_2.4 253949422881      2.4        8.5    total_effector    d8      II   male    17.8 #C71821
## 253949422882_1.1 253949422882      1.1        5.5    total_effector    d5      II female     8.7 #C71821
## 253949422882_1.2 253949422882      1.2        1.3        total_Tscm    d1       I female     2.4 #764382
## 253949422882_1.3 253949422882      1.3        2.4          total_cm    d2      II female     4.1 #48A748
## 253949422882_1.4 253949422882      1.4        6.1 A2_N24b_naivelike    d6       I female    12.1 #F2E822
## 253949422882_2.1 253949422882      2.1        3.5    total_effector    d3       I female     7.2 #C71821
## 253949422882_2.2 253949422882      2.2        4.2       total_naive    d4       I   male     7.9 #3699CA
## 253949422882_2.3 253949422882      2.3        8.4          total_cm    d8      II   male    17.8 #48A748
## 253949422882_2.4 253949422882      2.4        7.3        total_Tscm    d7      II   male    12.7 #764382
## 253949422890_1.1 253949422890      1.1        5.4          total_cm    d5      II female     8.7 #48A748
## 253949422890_1.2 253949422890      1.2        8.3        total_Tscm    d8      II   male    17.8 #764382
## 253949422890_1.3 253949422890      1.3        7.4          total_cm    d7      II   male    12.7 #48A748
## 253949422890_1.4 253949422890      1.4        1.2       total_naive    d1       I female     2.4 #3699CA
## 253949422890_2.1 253949422890      2.1        3.4          total_cm    d3       I female     7.2 #48A748
## 253949422890_2.2 253949422890      2.2        2.5    total_effector    d2      II female     4.1 #C71821
## 253949422890_2.3 253949422890      2.3        6.3        total_Tscm    d6       I female    12.1 #764382
## 253949422890_2.4 253949422890      2.4        4.1 A2_N24b_naivelike    d4       I   male     7.9 #F2E822
## 253949422891_1.1 253949422891      1.1        7.5    total_effector    d7      II   male    12.7 #C71821
## 253949422891_1.2 253949422891      1.2        8.1 A2_N24b_naivelike    d8      II   male    17.8 #F2E822
## 253949422891_1.3 253949422891      1.3        1.1 A2_N24b_naivelike    d1       I female     2.4 #F2E822
## 253949422891_1.4 253949422891      1.4        5.2       total_naive    d5      II female     8.7 #3699CA
## 253949422891_2.1 253949422891      2.1        6.5    total_effector    d6       I female    12.1 #C71821
## 253949422891_2.2 253949422891      2.2        2.2       total_naive    d2      II female     4.1 #3699CA
## 253949422891_2.3 253949422891      2.3        3.1 A2_N24b_naivelike    d3       I female     7.2 #F2E822
## 253949422891_2.4 253949422891      2.4        4.3        total_Tscm    d4       I   male     7.9 #764382
```

The variable annotations (mapping between probe IDs and gene symbols) are automatically extracted from the raw data files in the preprocessing above, and included in the `X` object.

The following code creates two boxplots depicting the overall distribution of values in the raw data object and in the final preprocessed (normalized) data object. The boxes are colored according to the cell type.

```
stopifnot(all(colnames(raw.data) == row.names(sample.annot)))
stopifnot(all(colnames(X) == row.names(sample.annot)))
par(mfrow = c(1, 2), pch = ".", mar = c(8, 4, 4, 2))
boxplot(raw.data$E, log = "y", ylab = "Signal intensity",
        main = "Raw data", las = 2, cex.axis = 0.75,
        col = sample.annot$color)
boxplot(X$E, ylab = "Signal intensity",
        main = "Processed data", las = 2, cex.axis = 0.75,
        col = sample.annot$color)
```



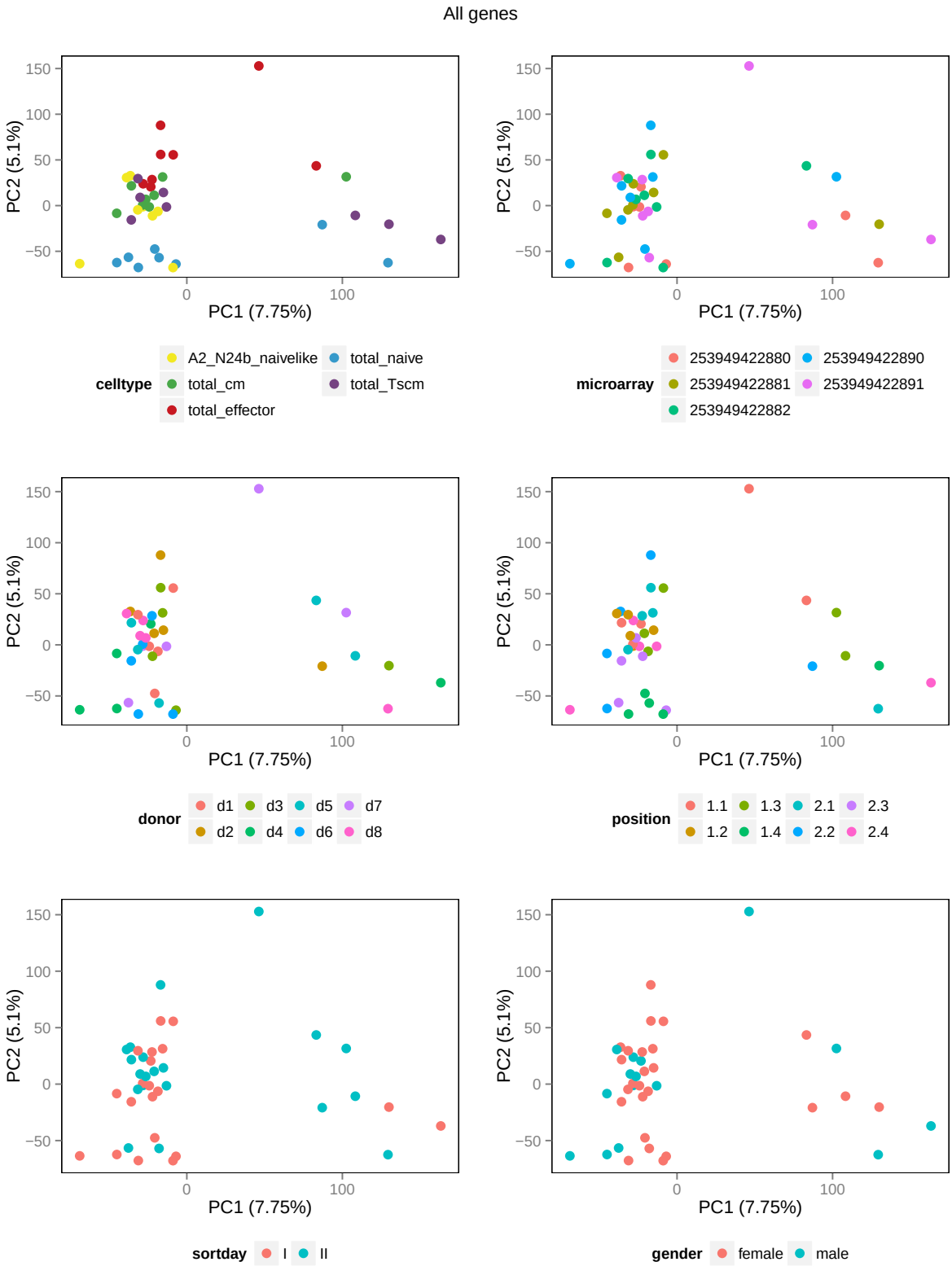
3 Exploratory analysis

Below we perform exploratory analysis using principal component analysis, in order to summarize and visualize the data set and to detect potential artifacts and effects of experimental and technical parameters.

PCA, all genes

First, we perform principal component analysis (PCA) to create a two-dimensional graphical representation of the samples in the data set, based on all the 41923 probes. Two samples are placed close together in the PCA representation if their expression patterns across all these genes are similar. To identify whether the patterns emerging in the PCA correspond to any of the information we have regarding the samples, we color the samples by the various annotations. We also include a table showing the probes that have the largest influence (weight) on each of the two first principal components. Note that the function `plotPCA` is available at the end of this document.

```
pca.res <- plotPCA(X$E, sample.annot, X$genes, title = "All genes")
```




```
xt1 <- xtable(cbind(PC1.pos = pca.res$pc1.pos, PC1.neg = pca.res$pc1.neg),
              caption = "Most influential probes on PC1, PCA with all probes")
print(xt1, tabular.environment = "longtable", type = "latex",
      floating = FALSE)
```

	PC1.pos	PC1.neg
1	A_21_P0014940 (LOC100653236, chr6)	A_33_P3292179 (ABCA9, chr17)
2	A_23_P37441 (B2M, chr15)	A_21_P0013037 (LOC154092, chr6)
3	A_24_P763243 (EEF1A1, chr6)	A_33_P3353496 (GJD4, chr10)
4	A_32_P47701 (EEF1A1, chr6)	A_33_P3410201 (ENST00000529743, chr8)
5	A_23_P2725 (RPL21, chr13)	A_33_P3274080 (A_33_P3274080, chrY)
6	A_21_P0013795 (RPL36A, chrX)	A_33_P3518572 (LOC100287314, chr12)
7	A_32_P115375 (CU677925, chr8)	A_33_P3479999 (LOC494150, chr12)
8	A_24_P179351 (TPT1, chr13)	A_33_P3759592 (FLJ13773, chr17)
9	A_32_P190488 (THC2550570, chr14)	A_33_P3246010 (AK092494, chr16)
10	A_21_P0011796 (XLOC_l2_007271, chr2)	A_33_P3409675 (LOC442132, chr5)

Table 1: Most influential probes on PC1, PCA with all probes

```
xt2 <- xtable(cbind(PC2.pos = pca.res$pc2.pos, PC2.neg = pca.res$pc2.neg),
              caption = "Most influential probes on PC2, PCA with all probes")
print(xt2, tabular.environment = "longtable", type = "latex",
      floating = FALSE)
```

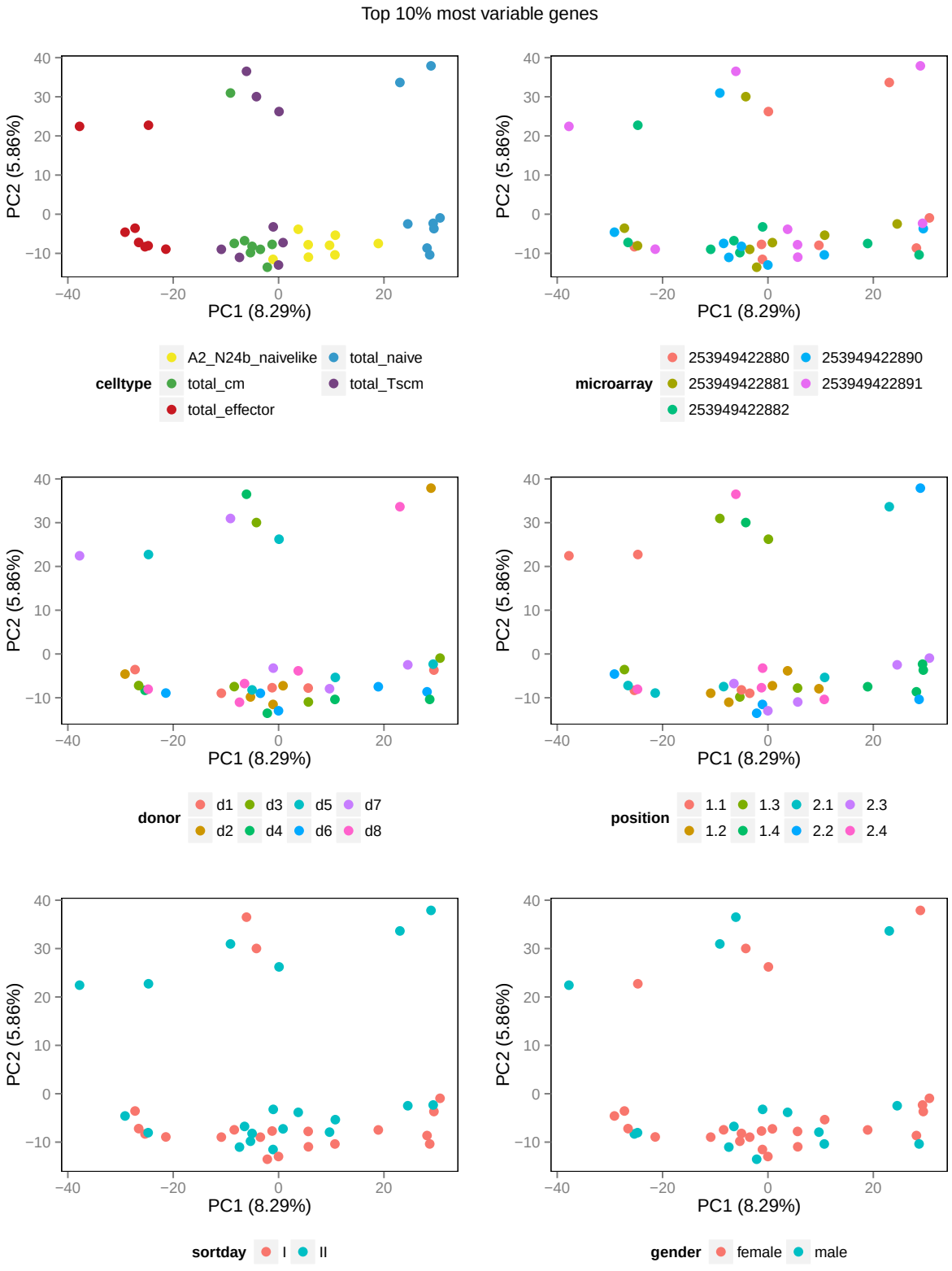
	PC2.pos	PC2.neg
1	A_19_P00322571 (MIAT, chr22)	A_23_P88095 (TBC1D4, chr13)
2	A_23_P122906 (AUTS2, chr7)	A_33_P3250671 (TCF7, chr5)
3	A_23_P397376 (MAF, chr16)	A_23_P105957 (ACTN1, chr14)
4	A_23_P151294 (IFNG, chr12)	A_19_P00802168 (AK123124, chr5)
5	A_21_P0014596 (LOC100506291, chr10)	A_32_P84373 (FAM153A, chr5)
6	A_24_P129632 (DLG5, chr10)	A_33_P3307253 (AK5, chr1)
7	A_19_P00321399 (XLOC_004598, chr5)	A_33_P3209962 (RASGRP2, chr11)
8	A_23_P209954 (GNLY, chr2)	A_23_P343398 (CCR7, chr17)
9	A_23_P102582 (C20orf24, chr20)	A_33_P3334398 (CA6, chr1)
10	A_21_P0013940 (LOC100287223, chr11)	A_23_P55127 (C17orf48, chr17)

Table 2: Most influential probes on PC2, PCA with all probes

PCA, 10% most variable genes

The PCA above was performed taking all the probes in the data set into account. Due to the large number of probes (in total, 41923), patterns characterized by small gene subsets may not be apparent in the PCA. Thus, we also perform PCA restricted to the most variable probes. We consider two cutoffs, first including the most variable 10% of the probes, and then including only the most variable 1%. As above, we color the samples by various annotations.

```
varthr <- quantile(apply(X$E, 1, var), probs = 0.9)
Xtmp <- X[which(apply(X$E, 1, var) > varthr), ]
pca.res <- plotPCA(Xtmp$E, sample.annot, Xtmp$genes,
                   title = "Top 10% most variable genes")
```




```
xt1 <- xtable(cbind(PC1.pos = pca.res$pc1.pos, PC1.neg = pca.res$pc1.neg),
              caption = "Most influential probes on PC1,
                         PCA with the 10\\% most variable probes")
print(xt1, tabular.environment = "longtable", type = "latex",
      floating = FALSE)
```

	PC1.pos	PC1.neg
1	A_23_P105957 (ACTN1, chr14)	A_33_P3243832 (ZEB2, chr2)
2	A_23_P343398 (CCR7, chr17)	A_24_P53976 (GLUL, chr1)
3	A_33_P3334398 (CA6, chr1)	A_23_P207564 (CCL4, chr17)
4	A_19_P00802168 (AK123124, chr5)	A_33_P3363933 (FCRL6, chr1)
5	A_21_P0012873 (LOC100507387, chr5)	A_33_P3354607 (CCL4, chr17)
6	A_32_P84373 (FAM153A, chr5)	A_23_P151294 (IFNG, chr12)
7	A_33_P3307253 (AK5, chr1)	A_23_P99275 (KLRB1, chr12)
8	A_24_P213788 (LOC641518, chr4)	A_24_P261760 (KLRG1, chr12)
9	A_21_P0007821 (XLOC_009661, chr12)	A_23_P142560 (ZEB2, chr2)
10	A_23_P30634 (BACH2, chr6)	A_23_P107744 (S1PR5, chr19)

Table 3: Most influential probes on PC1, PCA with the 10% most variable probes

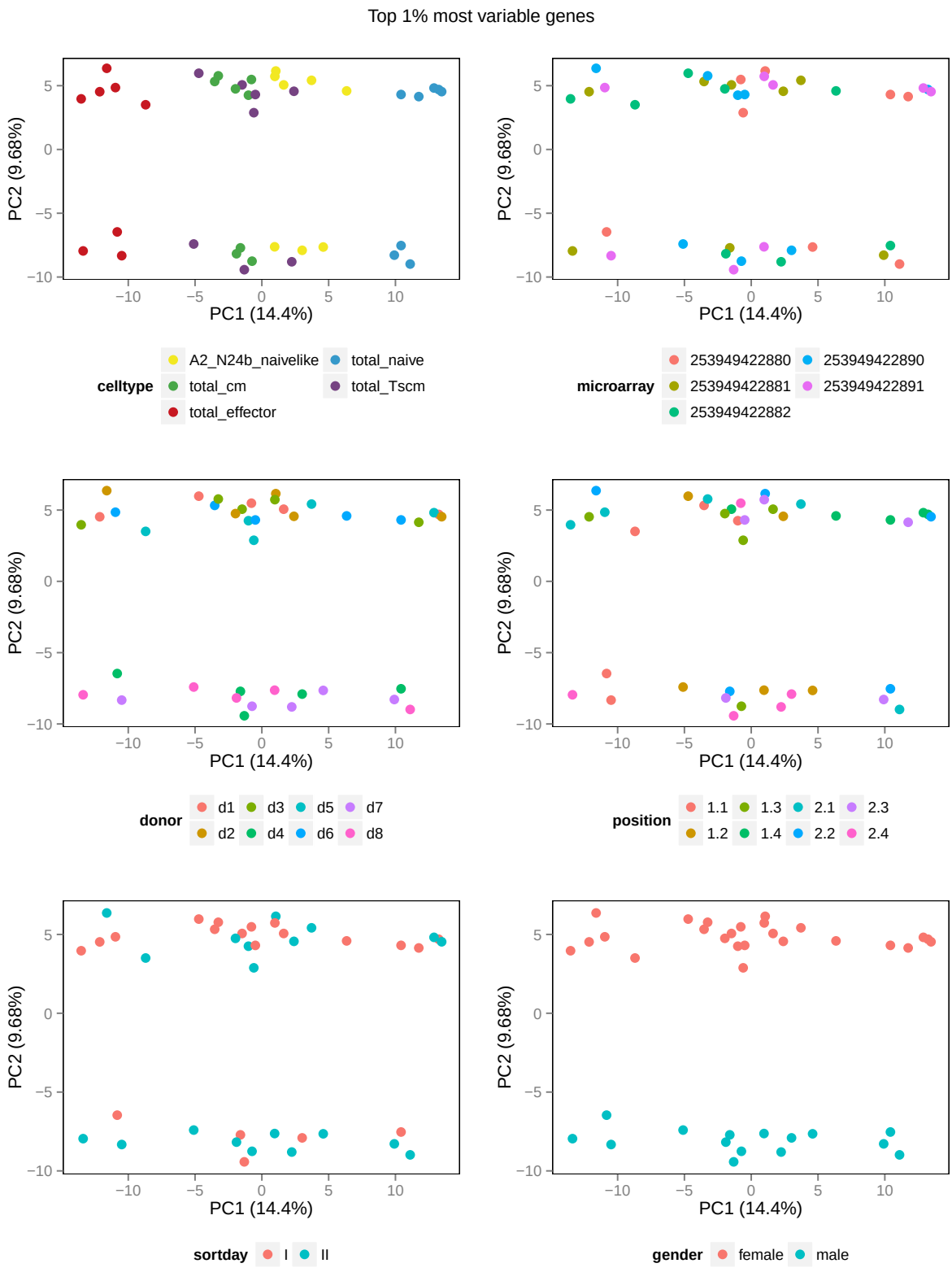
```
xt2 <- xtable(cbind(PC2.pos = pca.res$pc2.pos, PC2.neg = pca.res$pc2.neg),
              caption = "Most influential probes on PC2,
                         PCA with the 10\\% most variable probes")
print(xt2, tabular.environment = "longtable", type = "latex",
      floating = FALSE)
```

	PC2.pos	PC2.neg
1	A_23_P313512 (DCP1B, chr12)	A_33_P3217609 (TBC1D22A, chr22)
2	A_32_P430359 (DDX54, chr12)	A_33_P3266993 (AK130024, chr17)
3	A_23_P112103 (GSDMD, chr8)	A_21_P0014638 (LOC100507030, chr1)
4	A_23_P110879 (TRAF3IP2, chr6)	A_33_P3419360 (LOC731789, chr10)
5	A_33_P3281785 (NKX3-1, chr8)	A_33_P3353630 (A_33_P3353630, chr5)
6	A_21_P0011087 (LOC728715, chr12)	A_33_P3301752 (FLJ21369, chr19)
7	A_33_P3393986 (A_33_P3393986, chrX)	A_21_P0014002 (LOC100653063, chr5)
8	A_23_P211007 (NRIP1, chr21)	A_21_P0004587 (XLOC_005102, chr5)
9	A_23_P100764 (SLC25A39, chr17)	A_21_P0010295 (XLOC_014107, chr21)
10	A_24_P162293 (ERC1, chr12)	A_32_P56397 (LOC731789, chr10)

Table 4: Most influential probes on PC2, PCA with the 10% most variable probes

PCA, 1% most variable genes

```
varthr <- quantile(apply(X$E, 1, var), probs = 0.99)
Xtmp <- X[which(apply(X$E, 1, var) > varthr), ]
pca.res <- plotPCA(Xtmp$E, sample.annot, Xtmp$genes,
                   title = "Top 1% most variable genes")
```




```
xt1 <- xtable(cbind(PC1.pos = pca.res$pc1.pos, PC1.neg = pca.res$pc1.neg),
              caption = "Most influential probes on PC1,
                         PCA with the 1\\% most variable probes")
print(xt1, tabular.environment = "longtable", type = "latex",
      floating = FALSE)
```

	PC1.pos	PC1.neg
1	A_23_P105957 (ACTN1, chr14)	A_33_P3243832 (ZEB2, chr2)
2	A_21_P0000787 (LOC100507387, chr5)	A_23_P107744 (S1PR5, chr19)
3	A_32_P84373 (FAM153A, chr5)	A_23_P99275 (KLRB1, chr12)
4	A_21_P0000788 (LOC100507387, chr5)	A_23_P207564 (CCL4, chr17)
5	A_24_P213788 (LOC641518, chr4)	A_33_P3354607 (CCL4, chr17)
6	A_23_P411723 (PLAG1, chr8)	A_23_P68601 (CST7, chr20)
7	A_23_P34375 (TCEA3, chr1)	A_23_P119042 (NKG7, chr19)
8	A_21_P0008483 (XLOC_011117, chr14)	A_23_P142447 (MYO1F, chr19)
9	A_21_P0008486 (XLOC_011117, chr14)	A_23_P103803 (FCRL3, chr1)
10	A_24_P376760 (CA6, chr1)	A_23_P146644 (ANXA2, chr15)

Table 5: Most influential probes on PC1, PCA with the 1% most variable probes

```
xt2 <- xtable(cbind(PC2.pos = pca.res$pc2.pos, PC2.neg = pca.res$pc2.neg),
              caption = "Most influential probes on PC2,
                         PCA with the 1\\% most variable probes")
print(xt2, tabular.environment = "longtable", type = "latex",
      floating = FALSE)
```

	PC2.pos	PC2.neg
1	A_33_P3405911 (TSIX, chrX)	A_23_P364792 (TXLNG2P, chrY)
2	A_19_P00316565 (TSIX, chrX)	A_21_P0006594 (TTTY15, chrY)
3	A_19_P00327297 (XLOC_008015, chrX)	A_23_P324384 (RPS4Y2, chrY)
4	A_19_P00320438 (TSIX, chrX)	A_21_P0006606 (XLOC_008323, chrY)
5	A_19_P00326132 (TSIX, chrX)	A_23_P259314 (RPS4Y1, chrY)
6	A_19_P00321129 (TSIX, chrX)	A_33_P3725324 (USP9Y, chrY)
7	A_19_P00331623 (XIST, chrX)	A_23_P137238 (KDM5D, chrY)
8	A_19_P00321917 (TSIX, chrX)	A_23_P160004 (UTY, chrY)
9	A_19_P00806762 (TSIX, chrX)	A_33_P3261353 (BCORP1, chrY)
10	A_21_P0006456 (XLOC_008185, chrX)	A_21_P0006651 (XLOC_008386, chr10)

Table 6: Most influential probes on PC2, PCA with the 1% most variable probes

4 Help function

The following function is used to perform the principal component analysis and generate the plots shown above.

```
plotPCA <- function(data.matrix, sample.annotation,
                    variable.annotation, title = "") {

  ## Check input
  stopifnot(all(rownames(sample.annotation) == colnames(data.matrix)))
  stopifnot(all(rownames(data.matrix) == variable.annotation$ProbeName))

  ## Load necessary packages
  library(ggplot2)
  library(grid)
  library(gplots)
  library(gridExtra)

  ## Define viewport layout
  vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)

  ## Perfom PCA and collect annotation information
  pca <- prcomp(t(data.matrix), center = TRUE, scale. = TRUE)
  df <- data.frame(pc1 = pca$x[, 1],
                   pc2 = pca$x[, 2],
                   donor = sample.annot$donor,
                   celltype = sample.annot$celltype,
                   microarray = sample.annot$microarray,
                   position = sample.annot$position,
                   sortday = sample.annot$sortday,
                   gender = sample.annot$gender)

  ## Generate plots
  p1 <- ggplot(df, aes(x = pc1, y = pc2, color = donor)) +
    geom_point(size = 3) +
    scale_colour_discrete(guide = guide_legend(nrow = 2)) +
    xlab(paste0("PC1 (", signif(pca$sdev[1]^2/sum(pca$sdev^2)*100, 3), "%)")) +
    ylab(paste0("PC2 (", signif(pca$sdev[2]^2/sum(pca$sdev^2)*100, 3), "%)")) +
    theme(panel.grid.minor.y = element_blank(),
          panel.grid.major.x = element_blank(),
          panel.grid.major.y = element_blank(),
          panel.grid.minor.x = element_blank(),
          panel.background = element_rect(fill = "white", colour = "black"),
          legend.position = "bottom")

  p2 <- ggplot(df, aes(x = pc1, y = pc2, colour = factor(celltype))) +
    geom_point(size = 3) +
    scale_colour_manual(values = c(rgb(242, 232, 34, maxColorValue = 255),
                                   rgb(72, 167, 72, maxColorValue = 255),
                                   rgb(199, 24, 33, maxColorValue = 255),
                                   rgb(54, 153, 202, maxColorValue = 255),
```

```
                                  rgb(118, 67, 130, maxColorValue = 255)),
                       name = "celltype",
                       guide = guide_legend(nrow = 3)) +
  xlab(paste0("PC1 (", signif(pca$sdev[1]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  ylab(paste0("PC2 (", signif(pca$sdev[2]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  theme(panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        legend.position = "bottom")

p3 <- ggplot(df, aes(x = pc1, y = pc2, color = microarray)) +
  geom_point(size = 3) +
  scale_colour_discrete(guide = guide_legend(nrow = 3)) +
  xlab(paste0("PC1 (", signif(pca$sdev[1]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  ylab(paste0("PC2 (", signif(pca$sdev[2]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  theme(panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        legend.position = "bottom")

p4 <- ggplot(df, aes(x = pc1, y = pc2, color = position)) +
  geom_point(size = 3) +
  scale_colour_discrete(guide = guide_legend(nrow = 2)) +
  xlab(paste0("PC1 (", signif(pca$sdev[1]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  ylab(paste0("PC2 (", signif(pca$sdev[2]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  theme(panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        legend.position = "bottom")

p5 <- ggplot(df, aes(x = pc1, y = pc2, color = sortday)) +
  geom_point(size = 3) +
  scale_colour_discrete(guide = guide_legend(nrow = 1)) +
  xlab(paste0("PC1 (", signif(pca$sdev[1]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  ylab(paste0("PC2 (", signif(pca$sdev[2]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  theme(panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        legend.position = "bottom")

p6 <- ggplot(df, aes(x = pc1, y = pc2, color = gender)) +
  geom_point(size = 3) +
  scale_colour_discrete(guide = guide_legend(nrow = 1)) +
```

```
  xlab(paste0("PC1 (", signif(pca$sdev[1]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  ylab(paste0("PC2 (", signif(pca$sdev[2]^2/sum(pca$sdev^2)*100, 3), "%)")) +
  theme(panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.background = element_rect(fill = "white", colour = "black"),
        legend.position = "bottom")

## Place plots in grid
grid.newpage()
pushViewport(viewport(layout = grid.layout(4, 2, heights =
                                             unit(c(0.5, 5, 5, 5), "null"))))
grid.text(title, vp = vplayout(1, 1:2))
print(p2, vp = vplayout(2, 1))
print(p3, vp = vplayout(2, 2))
print(p1, vp = vplayout(3, 1))
print(p4, vp = vplayout(3, 2))
print(p5, vp = vplayout(4, 1))
print(p6, vp = vplayout(4, 2))

## Create tables with most influential probes
pc1.pos <- rownames(pca$rotation)[order(pca$rotation[, 1],
                                        decreasing = TRUE)[1:10]]
pc1.pos <-
  paste(pc1.pos, " (",
        variable.annotation$GeneName[match(pc1.pos,
                                           variable.annotation$ProbeName)],
        ", ", gsub(":.*", "",
                   gsub("hs\\|", "",
                        variable.annotation$chr_coord[match(pc1.pos,
                                                            variable.annotation$ProbeName)])),
        ")", sep = "")

pc1.neg <- rownames(pca$rotation)[order(pca$rotation[, 1],
                                        decreasing = FALSE)[1:10]]
pc1.neg <-
  paste(pc1.neg, " (",
        variable.annotation$GeneName[match(pc1.neg,
                                           variable.annotation$ProbeName)],
        ", ", gsub(":.*", "",
                   gsub("hs\\|", "",
                        variable.annotation$chr_coord[match(pc1.neg,
                                                            variable.annotation$ProbeName)])),
        ")", sep = "")

pc2.pos <- rownames(pca$rotation)[order(pca$rotation[, 2],
                                        decreasing = TRUE)[1:10]]
pc2.pos <-
  paste(pc2.pos, " (",
        variable.annotation$GeneName[match(pc2.pos,
```

```
                                              variable.annotation$ProbeName)],
          ", ", gsub(":.*", "",
                     gsub("hs\\|", "",
                          variable.annotation$chr_coord[match(pc2.pos,
                                                variable.annotation$ProbeName)])),
          ")", sep = "")

  pc2.neg <- rownames(pca$rotation)[order(pca$rotation[, 2],
                                          decreasing = FALSE)[1:10]]
  pc2.neg <-
    paste(pc2.neg, " (",
          variable.annotation$GeneName[match(pc2.neg,
                                             variable.annotation$ProbeName)],
          ", ", gsub(":.*", "",
                     gsub("hs\\|", "",
                          variable.annotation$chr_coord[match(pc2.neg,
                                                variable.annotation$ProbeName)])),
          ")", sep = "")

  return(list(pca = pca, pc1.pos = pc1.pos, pc1.neg = pc1.neg,
              pc2.pos = pc2.pos, pc2.neg = pc2.neg))
}
```

5 R session

- R version 3.2.0 (2015-04-16), `x86_64-apple-darwin13.4.0`
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: ggplot2 1.0.1, gplots 2.17.0, gridExtra 0.9.1, knitr 1.10.5, limma 3.24.3, xtable 1.7-4
- Loaded via a namespace (and not attached): bitops 1.0-6, caTools 1.17.1, colorspace 1.2-6, digest 0.6.8, evaluate 0.7, formatR 1.2, gdata 2.16.1, gtable 0.1.2, gtools 3.5.0, highr 0.5, KernSmooth 2.23-14, labeling 0.3, magrittr 1.5, MASS 7.3-40, munsell 0.4.2, plyr 1.8.2, proto 0.3-10, Rcpp 0.11.6, reshape2 1.4.1, scales 0.2.4, stringi 0.4-1, stringr 1.0.0, tools 3.2.0