Differential Expression Analysis of mRNA-seq Data from Single Cell and Circulating Tumor Cells

Ugne Jankauskaite

January 10, 2018

This project aims to reproduce results published in Ramsköld et al. (2012) study Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. The scope of this project is smaller than work done in the original paper. One of the main goals of Ramskold *et al.* study was to provide evidence that (at the time) newly created mRNA sequencing protocol (Smart-Seq) was robust and applicable to single-cell level. For this purpose they also conducted differential gene expression analysis for single cell and circulating tumor cells data. The goal of this project is to reproduce the aforementioned differential expression analysis.

Methods

All work was done using Linux operating system.

Alignment And Reads Count

Alignment and reads counting require a lot of resources and therefore were done outside of RStudio. Simple bash scripts bowtie.sh and rpkm.sh scripts for alignment and reads counting can be found in the same git repository. Before running all the tools mentioned below have to be installed and placed in the same directory. Raw fastq file were obtained following original article's GEO accession number GSE38495 from (European Nucleotide Archive (ENA)](http://www.ebi.ac.uk/ena).

- For sequence alignment Bowtie software was used. It was selected to match the article as well as for clear way to specify running a task on multiple cores. For indexing, pre-built index H. sapiens UCSC hg19 was used (available on bowtie manual page). The -m flag was used to ensure that mapping is unique. This is done to get unique mapping SAM files which were the authors' input to the subsequent pipeline steps. Typical command used:

  bowtie –threads 6 -m 1 -S hg19 -q input.fastq > output.sam

- Raw reads and Reads Per Kilobase per Million mapped reads (RPKM) values were obtained by the python script rpkmforgenes.py which was developed by the authors during previous studies for gene expression quantification in RNA-Seq data. The script allows non-uniquely mapped reads, but this option uses a lot of memory and exceeds my workstation capabilities (8 RAM). For annotation hg19 refGene.txt (RefSeq) file from UCSC database was used. The output is written as a plain text file and contains of four columns: "Gene ID", "Refseq ID", "RPKM" and "Reads". Typical command used:

  python2.7 rpkmforgenes.py -i input.sam -readcount -fulltranscript -mRNAnorm -samse -a refGene.txt -p 20 -sortpos -o output.RPKM.txt

File refGene.txt can be downloaded as follows:

```r
#refGene
print(paste("hg19 reference downloaded on ", Sys.Date()))
refGene.name <- paste0("refGene_",Sys.Date(),".txt")
refGene.name_zip <- paste0(refGene.name, ".zip")
url_hg19 <- "http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/refGene.txt.gz"
utils::download.file(url_hg19, destfile=refGene.name_zip, mode="wb")
R.utils::gunzip(refGene.name_zip, destname=refGene.name, overwrite=TRUE)
```

Differential Expression

- For differential gene expression analysis classic one-way ANOVA test is used. To avoid false positive p-values, they are adjusted with Benjamin-Hochberg (BH) method and reported as q (adjusted p) values. Then Tukey post-hoc test is used to identify pairs of samples which means are significantly different. For all pairs, genes are considered differentially expressed if their BH adjusted ANOVA p-value (q) and Tukey p-value are below 0.05.
- For hierarchical clustering analysis only highly expressed genes were selected (at least 100 RPKM in any sample). Clusters were based on dissimilarity measure obtained from Spearman correlation. For each cluster p-values were obtained by multi-scale bootstrap resampling. For this pvclust R package was used, which provides two types of p values:

  > AU (Approximately Unbiased) p-value and BP (Bootstrap Probability) value. AU p-value, which is computed by multiscale bootstrap resampling, is a better approximation to unbiased p-value than BP value computed by normal bootstrap resampling.

- Principal Component Analysis was performed with function prcomp from stats R package which, rather than SVDMAN tool used by Ramskold *et al.*

Biological setting

Firstly, single-cell transcriptomes from prostate and bladder cancer cells are used. This is to ensure that data obtained with newly proposed Smart-Seq protocol is usable for differential expression analysis and cell lineage identification.

Secondly, differential expression of circulating tumor cells (CTC) versus primary melanocytes (PM), melanoma cancer cells (SKMEL5, UACC257), embrionic stem cells (ESC), white blood cells (WB) and Burkitt's lymphoma cells are studied. The goal is to show that after Smart-Seq application, CTC can be identified by several marker genes with high precision. Possibility of better CTC detection is of special interest because they are associated with tumor metastasis. As Plaks, Koopman, and Werb (2013) explains in their article:

> Because dissemination mostly occurs through the blood, circulating tumor cells (CTCs) that have been shed into the vasculature and may be on their way to potential metastatic sites are of obvious interest.

Immune cells are used for comparison, since CTC are extracted from blood circulation.

Data Analysis

Data Donwload and Preparation

Load all required libraries:

```r
library(tools)
library(ppls)
library(tibble)
library(pvclust)
library(dendextend)
library(gplots)
library(XLConnect)
library(data.table)
library(knitr)
library(kableExtra)
```

Set directory, where all needed files will be stored. By default, this should be the directory of the R markdown source file.

```r
# Setting working directory to source file directory. Works on RStudio.
knitr::opts_knit$set(root.dir = getwd())
workdir <- getwd()
data_dir = paste0(getwd(), "/data")
```

Counts data can also be downloaded here (in this study, using the reproduced data from raw fastq files).

```r
url_main <- "https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE38495&format=file"
utils::download.file(url, destfile="GSE38495_RAW.tar", mode="wb")
```

Extract pre-processed data into the data directory

```r
utils::untar("data_from_raw.tar", exdir = data_dir)
```

From extracted files, identify files for each cell line:

```r
# NG2+ putative melanoma CTC (Circulating melanoma cell)
CTC.list_files = list.files(path = data_dir, pattern = "*CTC_RPKM*")
CTC.num_files = length(CTC.list_files)
# WB - white blood cells
WB.list_files = list.files(path = data_dir, pattern = "*Whitebloodcell_[0-9]_RPKM*")
WB.num_files = length(WB.list_files)
# BL - Burkitt's lymphoma cells
BL.list_files = list.files(path = data_dir, pattern = "*GSM78215[1-9]_BL*")
BL.num_files = length(BL.list_files)
# PM - primary melanocytes
PM.list_files = list.files(path = data_dir, pattern = "*pm_RPKM*")
PM.num_files = length(PM.list_files)
# SKMEL5
SKMEL.list_files = list.files(path = data_dir, pattern = "*SKMEL5_cell[0-9]_RPKM*")
SKMEL.num_files = length(SKMEL.list_files)
#UACC257
UACC.list_files = list.files(path = data_dir, pattern = "*UACC257_cell[0-9]_RPKM*")
UACC.num_files = length(UACC.list_files)
# ESCs
ESC.list_files = list.files(path = data_dir, pattern = "*hESC_RPKM*")
ESC.num_files = length(ESC.list_files)

#T24
T24.list_files = list.files(path = data_dir, pattern = "*[T,t]24-1[0-9]_RPKM*")
T24.num_files = length(T24.list_files)
#Lncap
Lncap.list_files = list.files(path = data_dir, pattern = "*Lncap-1[0-9]_RPKM*")
Lncap.num_files = length(Lncap.list_files)
#PC3
PC3.list_files = list.files(path = data_dir, pattern = "*[P,p][C,c]3-1[0-9]_RPKM*")
PC3.num_files = length(PC3.list_files)

all_files = c(CTC.list_files, PM.list_files, SKMEL.list_files, UACC.list_files,
              ESC.list_files, Lncap.list_files, PC3.list_files,
              T24.list_files, WB.list_files, BL.list_files)
```

Data files contain duplicate rows (which might be a bug in the rpkmforgenes.py script). For example, first five repetitive lines and their repetition count can be printed using the following command:

```
cmd = paste0("sort ", data_dir, "/", all_files[1], " | uniq -cd | tail -n +5 |  head -5")
system(cmd)
```

The following code removes them from the list of files.

```
removeDuplicates <- function(pathToFile){
  file = readLines(pathToFile,-1)
  x <- read.table(pathToFile, header=FALSE, sep="\t", comment.char = "#", check.names = FALSE)
  # To check which lines were removed, uncomment line below before running.
  # print(x[duplicated(x[,1]),])
  y <- x[!duplicated(x[,1]),]
  write.table(y, pathToFile, sep="\t",row.names=FALSE, quote = FALSE, col.names = FALSE)
}

for (i in 1:length(all_files)){
  pathToFile = paste0(data_dir, "/", all_files[i])
  removeDuplicates(pathToFile)
}
```

Create a grouping variable, in which sample information is discarded and only cell line name is maintained:

```
group <- as.factor(c(rep("CTC", CTC.num_files), rep("PM", PM.num_files),
                     rep("SKMEL", SKMEL.num_files), rep("UACC", UACC.num_files),
                     rep("ESC", ESC.num_files), rep("Lncap", Lncap.num_files),
                     rep("PC3", PC3.num_files), rep("T24", T24.num_files),
                     rep("WB", WB.num_files), rep("BL", BL.num_files)))
```

Create a data frame of gene RPKM counts for all samples:

```
num_files = length(all_files)
cell_names <- vector(mode="character", length=num_files)
for (i in 1:num_files){
  path=paste0(data_dir, "/", all_files[i])
  cell_names[i] <- substr(all_files[i],11,nchar(all_files[i])-9)
  sample_data <- read.table(path, header=FALSE, sep="\t", stringsAsFactors=FALSE)
  colnames(sample_data) <- c("Gene.symbol","Refseq.ID", "RPKM.FPKM", "reads")
  if (i!=1){
    data.RPKM <- cbind(data.RPKM, sample_data$RPKM.FPKM)
    colnames(data.RPKM)[i+1] = cell_names[i]
    next
  }
  else{data.RPKM <- data.frame(sample_data$Gene.symbol, sample_data$RPKM.FPKM)
  colnames(data.RPKM)[1] <- "Gene"
  colnames(data.RPKM)[2] <- cell_names[1]
  }
}
```

Differenial Expression Analysis of 12 Single Cell Samples

Firstly, 12 single cell cancer samples are separated from the rest data.

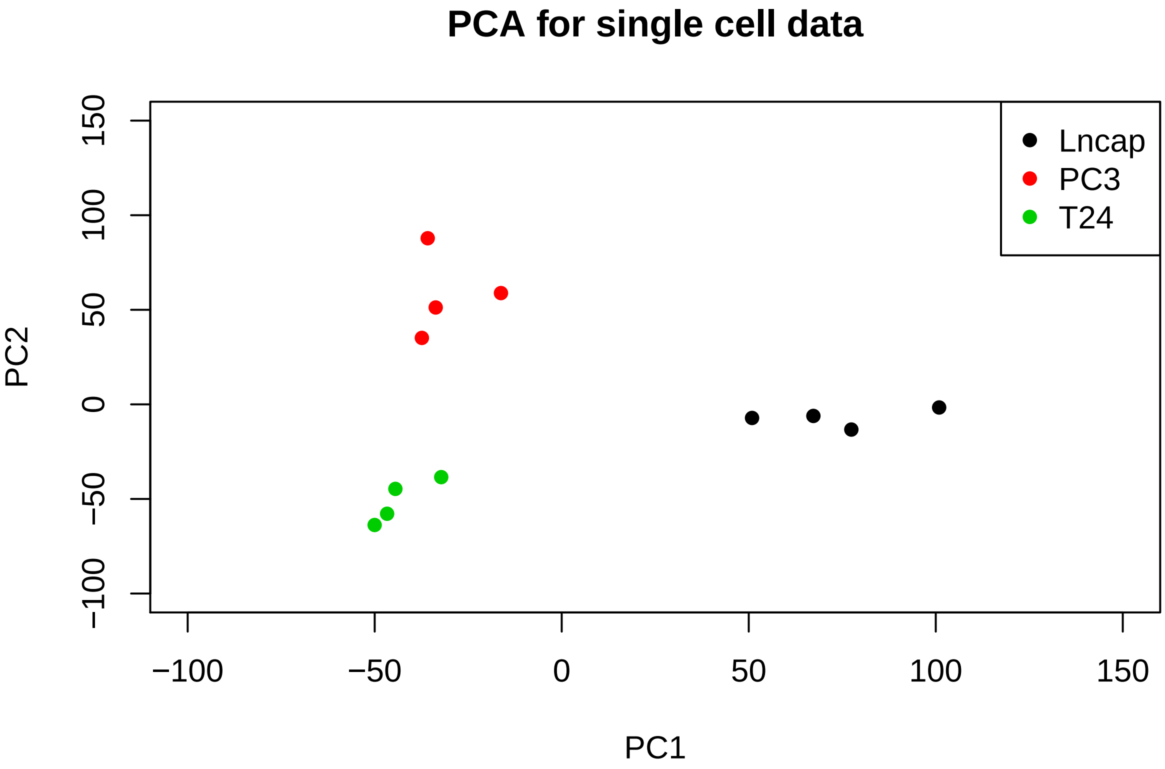
```
data.RPKM_12cancer <- data.frame(data.RPKM$Gene, data.RPKM[,25:36])
dim(data.RPKM_12cancer)
```

```
## [1] 24607    13
```

Principal Component Analysis

Principal component analysis shows clear separation between three single cell groups. While the separation is as seen in Ramskold *et al.* study (**figure 3a**), the exact values of principal components seem to be slightly different (not possible to compare exact numbers as they are not reported by the authors). In general, the original paper figure shows that PC1 and PC2 values for T24 are smaller, while values for Lncap are bigger and more closely clustered than shown in the plot below.

```r
temp_data<- data.frame(scale(t(data.RPKM_12cancer[c(2:13)])))
# remove genes with NAN values in all samples (no measurements, no variance change)
bad <- sapply(temp_data, function(x) all(is.nan(x)))
temp_data<-temp_data[,!bad]
prc <- prcomp(temp_data, scale. = F, center = F)
plot(-prc$x[,1:2], main="PCA for single cell data", col=factor(group[24:35]),
     pch=16, xlim=c(-100,150), ylim=c(-100,150))
legend("topright", legend=levels(factor(group[24:35])), pch=16,
       col=unique(factor(group[24:35])))
```



One-way ANOVA Test

A shift of one (to avoid log(0)) is applied and log2 of RPKM counts taken for easier calculation and imaging.

```r
data.RPKM_12cancer[,2:13] <- log2(data.RPKM_12cancer[,2:13]+1)
```

Then we can perform one-way ANOVA test with Benjamin-Hochberg p-value adjustment and post-hoc Tukey test.

1. Create some helper variables

```r
# threshold p-value
p_t=0.05
# some helper data frames for data separation and storage
```

```
PC3vsLncap = data.frame(Gene=character(),  p.Tukey=double(), stringsAsFactors=FALSE)
T24vsLncap = data.frame(Gene=character(),  p.Tukey=double(), stringsAsFactors=FALSE)
T24vsPC3 = data.frame(Gene=character(),  p.Tukey=double(), stringsAsFactors=FALSE)
ANOVA=data.frame(Gene=character(), p=double(), q=double())
```

2. Run one-way ANOVA and Tukey post-hoc

```
tmydf = setNames(data.frame(t(data.RPKM_12cancer[,-1])), data.RPKM_12cancer[,1])
s<-data.frame(CellType=factor(c(rep("Lncap", Lncap.num_files),
                     rep("PC3", PC3.num_files), rep("T24", T24.num_files))), tmydf)
num_genes = dim(s)[2]
for (i in 2:num_genes){
  geneCounts <- s[,i]
  geneName <- names(s[i])
  model<-lm(geneCounts~s$CellType)
  an <- anova(model)
  an.p <- an$`Pr(>F)`[1]
  ANOVA = rbind(ANOVA, data.frame(Gene=geneName, p=an.p))

  posthoc <- TukeyHSD(aov(geneCounts~s$CellType))
  for (j in 1:3) {
    tukey.p <- posthoc$`s$CellType`[j,4]
    rowName = rownames(posthoc$`s$CellType`)[j]
    if (identical(rowName,"PC3-Lncap")){
      PC3vsLncap = rbind(PC3vsLncap, data.frame(Gene=geneName, p.Tukey=tukey.p))
    }
    else if (identical(rowName,"T24-Lncap")){
       T24vsLncap = rbind(T24vsLncap, data.frame(Gene=geneName, p.Tukey=tukey.p))
    }
    else if (identical(rowName,"T24-PC3")){
       T24vsPC3 = rbind(T24vsPC3, data.frame(Gene=geneName, p.Tukey=tukey.p))
    }
  }
}
```

Number of genes with unadjusted ANOVA p-values less than 0.05:

```
## [1] 3191
```

3. Benjamin-Hochberg ANOVA p-value adjustment

```
#Use BH to adjust p values
ANOVA$q <- p.adjust(ANOVA$p, method = "BH")
```

Number of genes with BH adjusted ANOVA p-value (q) less than 0.05:

```
## [1] 928
```

Add newly calculated p values to the dataframe:

```
data.RPKM_12cancer$p.ANOVA <- ANOVA$p
data.RPKM_12cancer$q.ANOVA <- ANOVA$q
data.RPKM_12cancer$PC3vsLncap.p.Tukey <- PC3vsLncap$p.Tukey
data.RPKM_12cancer$T24vsLncap.p.Tukey <- T24vsLncap$p.Tukey
data.RPKM_12cancer$T24vsPC3.p.Tukey <- T24vsPC3$p.Tukey
```

Genes are considered differentially expressed between two types of cells if both BH adjusted ANOVA p-value (q) and Tukey p value are smaller than 0.05.

```
id.de_PC3vsLncap <- which(data.RPKM_12cancer$q.ANOVA < p_t &
                            data.RPKM_12cancer$PC3vsLncap.p.Tukey < p_t)
id.de_T24vsLncap <- which(data.RPKM_12cancer$q.ANOVA < p_t &
                            data.RPKM_12cancer$T24vsLncap.p.Tukey < p_t)
id.de_T24vsPC3 <- which(data.RPKM_12cancer$q.ANOVA < p_t &
                          data.RPKM_12cancer$T24vsPC3.p.Tukey < p_t)
```

Number of DE genes between PC3 and Lncap cells:

```
## [1] 676
```

Number of DE genes between T24 and Lncap cells:

```
## [1] 784
```

Number of DE genes between T24 and PC3 cells:

```
## [1] 476
```

The number of pairwise differentially expressed genes are higher than in the original publication. However, proportional relationship is maintained.

Differenial Expression Analysis of Circulating Tumor Transcriptomes

Circulating tumor samples samples are separated from the rest data. Shift of 1 and log2 is applied to RPKM counts.

```
data.RPKM_tumor<-data.frame(Genes=data.RPKM$Gene, log2(data.RPKM[,2:16]+1))
dim(data.RPKM_tumor)
```

```
## [1] 24607    16
```

One-way ANOVA Test

Then we can perform one-way ANOVA test with Benjamin-Hochberg p-value adjustment and post-hoc Tukey test. Since calculations are equivalent to the ones for 12 single cell data, they are excluded from PDF report file. To see the code please refer to the R markdown file.

The number of genes for which ANOVA p-values are less then 0.05 is:

```
## [1] 2266
```

Number of genes with BH adjusted ANOVA p-value (q) less than 0.05:

```
## [1] 567
```

Now, we can obtain number of genes that are differentially expressed between primary melanocytes and circulating tumor cells:

```
id.de_PMvsCTC <- which(data.RPKM_tumor$q.ANOVA < p_t &
                         data.RPKM_tumor$PMvsCTC.p.Tukey < p_t)
length(id.de_PMvsCTC)
```

```
## [1] 302
```

Number of genes that are upregulated in putative CTCs in comparison to primary melanocytes and first rows of these genes expression data:

```
## [1] 138
```

	CTC1	CTC2	CTC3	CTC4	CTC5	CTC6	PM1	PM2	p.ANOVA	q.ANOVA	PMvsCTC.p.Tukey
NIPAL3+STPG1	3.862232	6.873917	6.504492	6.164927	3.981111	4.713672	0.1239859	0.0697603	0.0014223	0.0392243	0.0009325
CDC20	5.471837	4.761157	5.642079	3.707087	3.982662	4.651315	0.0000000	0.2893640	0.0000228	0.0021208	0.0002422
PSMB4	11.927822	11.511092	11.207168	10.936159	11.254160	10.694366	9.4784719	9.7895465	0.0000207	0.0019719	0.0054582
IQGAP3	7.313902	3.662955	6.707367	5.291382	1.694006	7.030844	0.0932843	0.0000000	0.0012797	0.0365825	0.0088873
SERTAD4	5.125461	2.997635	6.123106	3.783732	2.033973	3.134272	0.0000000	0.0000000	0.0006068	0.0224578	0.0063098
ENAH	6.039267	3.248495	5.155749	5.767513	3.710193	5.717252	0.0758166	0.0535427	0.0014445	0.0396036	0.0014777

Number of genes that are downregulated in putative CTCs in comparison to primary melanocytes and first rows of these genes expression data:

```
## [1] 164
```

	CTC1	CTC2	CTC3	CTC4	CTC5	CTC6	PM1	PM2	p.ANOVA	q.ANOVA	PMvsCTC.p.Tukey
ISG15	0.000000	0.0000000	0.0000000	3.714270	0.000000	0.000000	9.3696997	8.1844807	0.0000107	0.0011227	0.0000134
MFAP2	0.000000	0.3918379	0.0000000	0.000000	0.000000	0.000000	5.4878813	4.8870950	0.0000000	0.0000000	0.0000000
NBPF3	0.000000	0.1870653	0.0000000	0.000000	0.000000	0.000000	0.6582337	0.3197848	0.0002087	0.0105568	0.0002798
RUNX3	3.321721	0.0000000	0.0406248	2.978488	0.000000	0.000000	5.2672186	5.5126099	0.0004947	0.0194142	0.0058986
TRIM63	0.000000	0.0000000	0.0000000	0.000000	0.000000	0.000000	2.8865314	6.5809734	0.0010607	0.0323831	0.0087888
IFI6	6.793634	5.5332812	5.5186124	6.405115	6.638209	4.455067	10.8136311	10.9901242	0.0000672	0.0046739	0.0000672

Hierarchial Clustering

As in the original paper, hierarchical clustering was performed only for genes for which at least one sample had a high expression value (RPKM > 100).

```r
# Rearange data
row_names <- data.RPKM[,1]
data.RPKM <- data.RPKM[,-1]
rownames(data.RPKM) <- row_names
print(paste0("Original data gene count: ", dim(data.RPKM)[1]))
```

```
## [1] "Original data gene count: 24607"
```

```r
# get highly expressed genes
data.RPKM_high <- data.RPKM[apply(data.RPKM[,-1], 1, function(row) {any(row > 100)}), ]
colnames(data.RPKM_high) <- group
print(paste0("Gene count for genes with RPKM > 100: ", dim(data.RPKM_high)[1]))
```

```
## [1] "Gene count for genes with RPKM > 100: 4587"
```

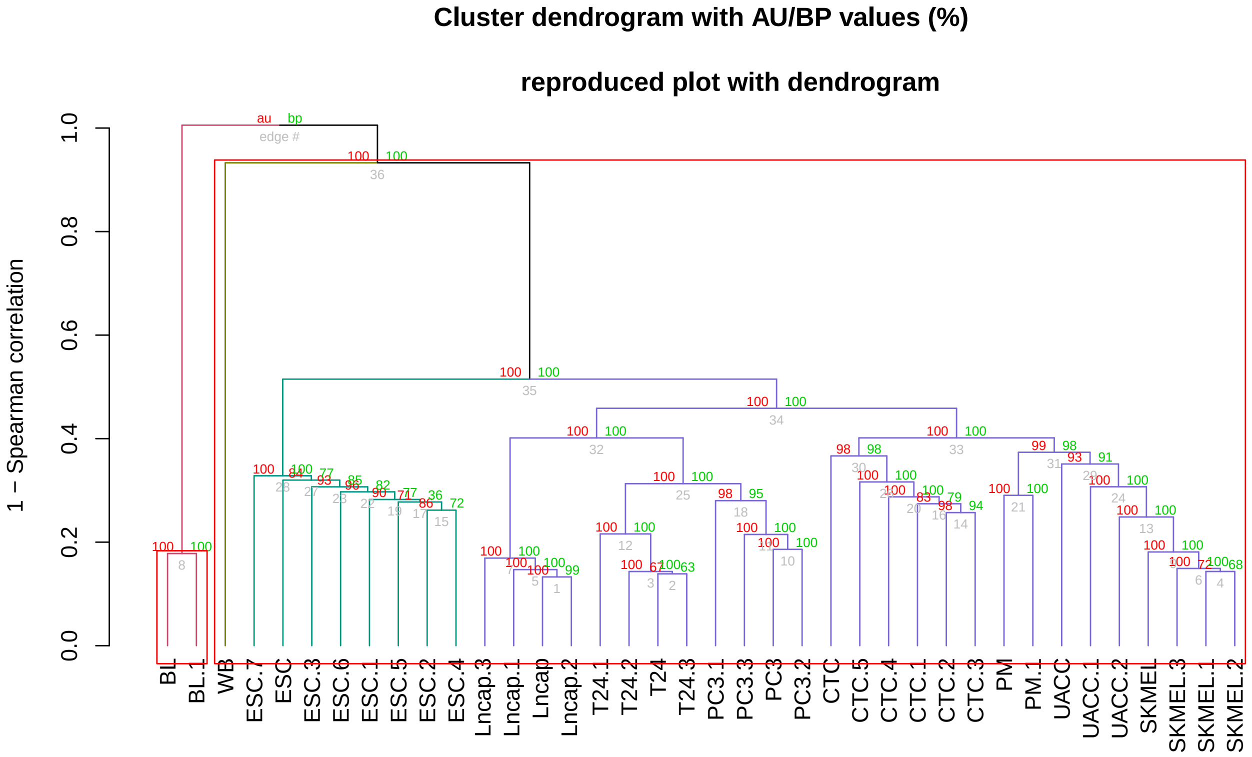
To cluster data, dissimilarity distance is calculated based on Spearman correlation and 1000 bootstrapped samples added.

```r
spearman <- function(x, ...) {
    x <- as.matrix(x)
    res <- as.dist(1 - cor(x, method = "spearman", use = "everything"))
    res <- as.dist(res)
    attr(res, "method") <- "spearman"
    return(res)
}
data.RPKM_high <- data.RPKM_high[, ! apply(data.RPKM_high , 2 ,
                                 function(x) sd(x, na.rm = TRUE)==0 ) ]
cluster.bootstrap <- pvclust(data.RPKM_high, parallel = TRUE, nboot = 1000,
                             method.dist=spearman)
```

```
## Creating a temporary cluster...done:
## socket cluster with 7 nodes on host 'localhost'
## Multiscale bootstrap... Done.
```

In the plot below, AU (Approximately Unbiased) p-value and BP (Bootstrap Probability) value are as described in the Methods section. Red rectangles marks clusters with high AU values (95%).

```
dend <- as.dendrogram(cluster.bootstrap)
dend <- color_branches(dend, 4)

dend %>% as.dendrogram %>%
   plot(main = "Cluster dendrogram with AU/BP values (%)\n
        reproduced plot with dendrogram",
        ylab = "1 - Spearman correlation")
cluster.bootstrap %>% text(cex=0.6)
cluster.bootstrap %>% pvrect(alpha=0.95, pv="au", type="geq")
```



Overall, all cells clustered within their cell lines with high AU p-value. Burkitt's lymphoma samples are clearly separated from prostate and bladder single cell samples, primary melanocytes, melanoma cancer cell line and embryonic stem cell samples. All the rest samples make expected clusters with samples of similar cell lines, importantly CTC are in the same cluster as PM, SKMEL and UACC cells.

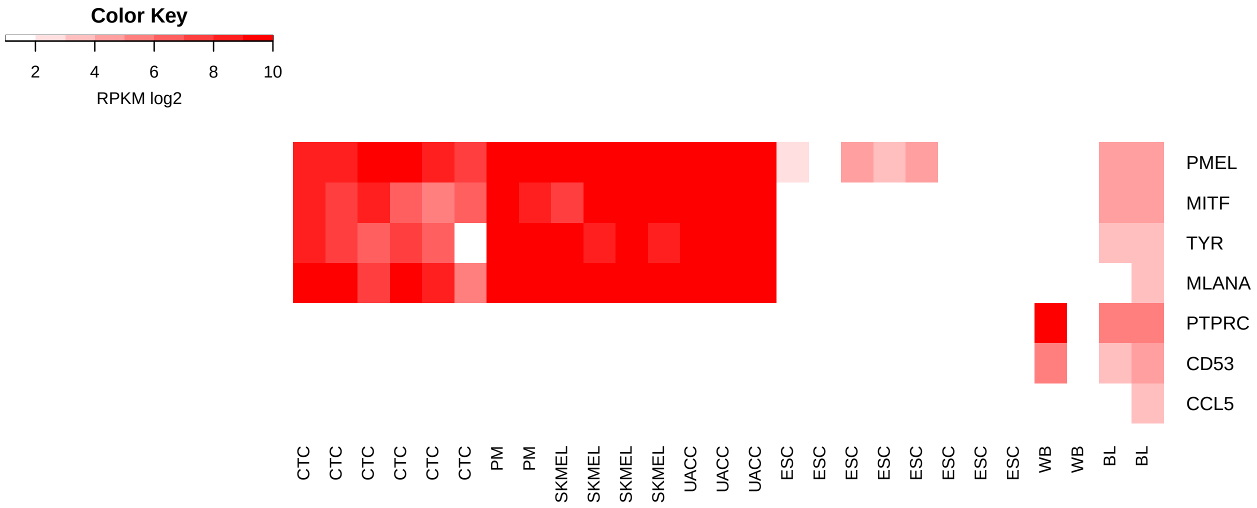
Heatmap Comparison of Gene Expression

For the heatmaps, the data shift value before taking a logarithm of the data was not stated in the original paper. However, since heatmaps in **figures 4b-4f** of the paper include negative logarithm values, here the applied shift is chosen so be smaller then one:

```
data.RPKM_log <- log2(data.RPKM+0.1)
```

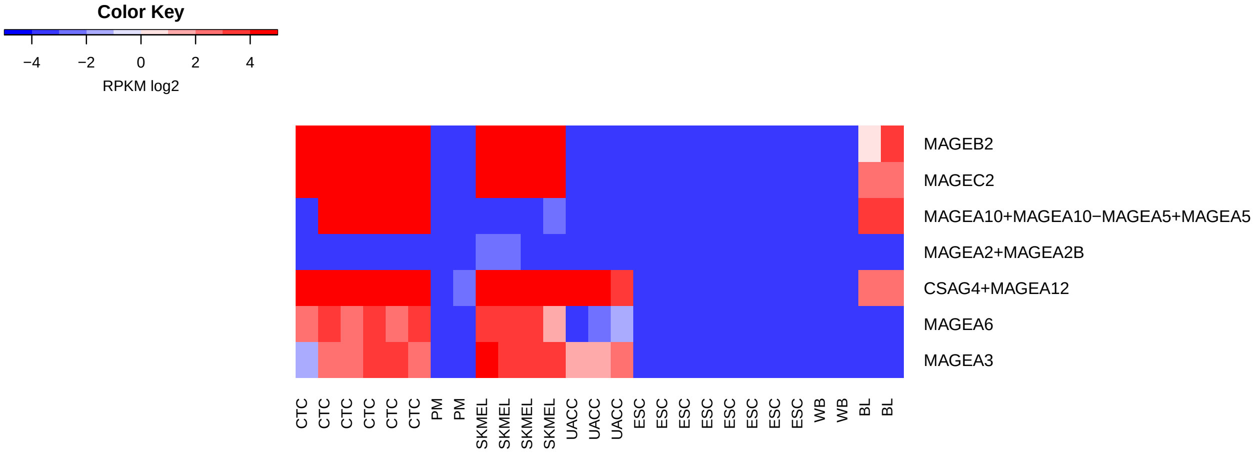
In **figure 4b** and **supplementary figure 9** Ramskold *et al.* used known NG2+ CTC marker genes PMEL, MITF, TYR, MLANA and known immune marker genes PTPRC, CD53, CCL5 to show that CTCs are of melanocytic origin and not immune origin. These results are accurately reproduced in the figure below:

```
cols <- c(1:23, 36:39)
data.RPKM_h1 <- rbind(data.RPKM_log["PMEL", cols], data.RPKM_log["MITF", cols],
                        data.RPKM_log["TYR", cols], data.RPKM_log["MLANA", cols],
                        data.RPKM_log["PTPRC", cols], data.RPKM_log["CD53", cols],
                        data.RPKM_log["CCL5", cols])
colnames(data.RPKM_h1) <- group[cols]

heatmap.2(as.matrix(data.RPKM_h1), dendrogram = "none", Colv=FALSE, Rowv = FALSE,
          tracecol = NA, col=colorRampPalette(c("white", "red")),
          breaks=c(1:0.5:10), density.info="none", key.xlab = "RPKM log2", cexRow = 1)
```



Then, in order to further compare primary melanocytes and CTCs, the authors showed that melanoma-associated tumor antigens are upregulated in CTC samples, compared to PM samples. These results were reproduced in the figure below:

```
cols <- c(1:23, 36:39)
data.RPKM_h2 <- rbind(data.RPKM_log["MAGEB2", cols], data.RPKM_log["MAGEC2", cols],
                      data.RPKM_log["MAGEA10", cols], data.RPKM_log["MAGEA2", cols],
                      data.RPKM_log["CSAG4+MAGEA12", cols], data.RPKM_log["MAGEA6", cols],
                      data.RPKM_log["MAGEA3", cols])
colnames(data.RPKM_h2) <- group[cols]

heatmap.2(as.matrix(data.RPKM_h2), dendrogram = "none", Colv=FALSE, Rowv = FALSE,
          tracecol = NA, col=colorRampPalette(c("blue", "white", "red")),
          breaks=c(-5:5), key=TRUE, symkey=FALSE, density.info="none",
          key.xlab = "RPKM log2", cexRow = 1,margins=c(5,18))
```

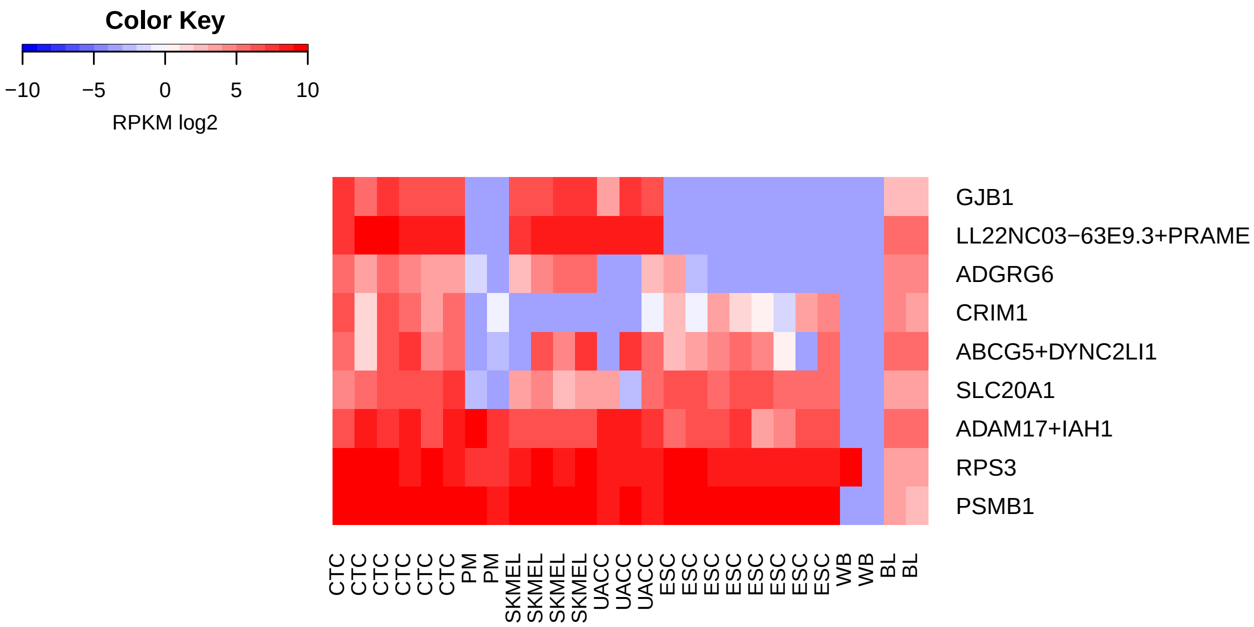


Data table of these genes expression for CTC and PM samples shows clear differential expression with significant one-way ANOVA p-values, BH adjusted q-values, and Tukey post-hoc test p-values for all antigens, except MAGEA2+MAGEA2B:

	CTC1	CTC2	CTC3	CTC4	CTC5	CTC6	PM1	PM2	p.ANOVA	q.ANOVA	PMvsCTC.p.Tukey
MAGEB2	7.2237602	10.033935	7.598047	8.412136	9.092403	8.318488	0	0.0000000	0.0000000	0.0000095	0.0000004
MAGEC2	7.2164381	9.523607	9.352021	9.109006	8.911451	8.806244	0	0.0000000	0.0000000	0.0000022	0.0000001
MAGEA10+MAGEA10-MAGEA5+MAGEA5	0.0000000	7.909798	6.463821	6.299014	6.464823	5.917956	0	0.0000000	0.0016102	0.0422522	0.0187916
MAGEA2+MAGEA2B	0.0000000	0.000000	0.000000	0.000000	0.000000	0.000000	0	0.0000000	0.1084233	0.4206824	1.0000000
CSAG4+MAGEA12	4.6219293	6.032660	6.858064	6.966196	6.478108	6.800355	0	0.1084925	0.0000487	0.0036797	0.0000567
MAGEA6	2.9859013	3.908490	3.075102	3.210015	3.090047	3.798007	0	0.0000000	0.0000031	0.0004065	0.0000551
MAGEA3	0.3481506	2.707171	2.453360	3.258807	3.858179	2.733718	0	0.0000000	0.0037247	0.0737867	0.0184342

Further nine plasma-membrane associated transcripts were identified in CTC compared to PM in the **figure 4e** of the original paper. However, here the authors results are not reproduce completely. The heatmap below shows that two of the nine transcripts - RPS3 and PSMB1 - were also highly expressed in PM cells.

```
cols <- c(1:23, 36:39)
data.RPKM_h3 <- rbind(data.RPKM_log["GJB1", cols], data.RPKM_log["LL22NC03-63E9.3+PRAME", cols],
                      data.RPKM_log["ADGRG6", cols], data.RPKM_log["CRIM1", cols],
                      data.RPKM_log["ABCG5", cols], data.RPKM_log["SLC20A1", cols],
                      data.RPKM_log["ADAM17", cols], data.RPKM_log["RPS3", cols],
                      data.RPKM_log["PSMB1", cols])
colnames(data.RPKM_h3) <- group[cols]

library(gplots)
heatmap.2(as.matrix(data.RPKM_h3), dendrogram = "none", Colv=FALSE, Rowv=FALSE,
          tracecol = NA, col=colorRampPalette(c("blue", "white", "red")),
          breaks=c(-10:10), key=TRUE, symkey=FALSE, density.info="none",
          key.xlab = "RPKM log2", cexRow = 1, margins=c(5,18))
```



Furthermore, comparing the authors data from Supplementary Table 4 to data generated during this project, very similar results can be observed for the nine genes (small differences may be due to different shift before applying log2). It can be seen that genes RPS3 and PSMB1 are quite highly expressed in PM according to the authors data, however in the **figure 4e** of the paper the expression is shown as very small.

- Data for 9 genes by Ramskold *et al.*:

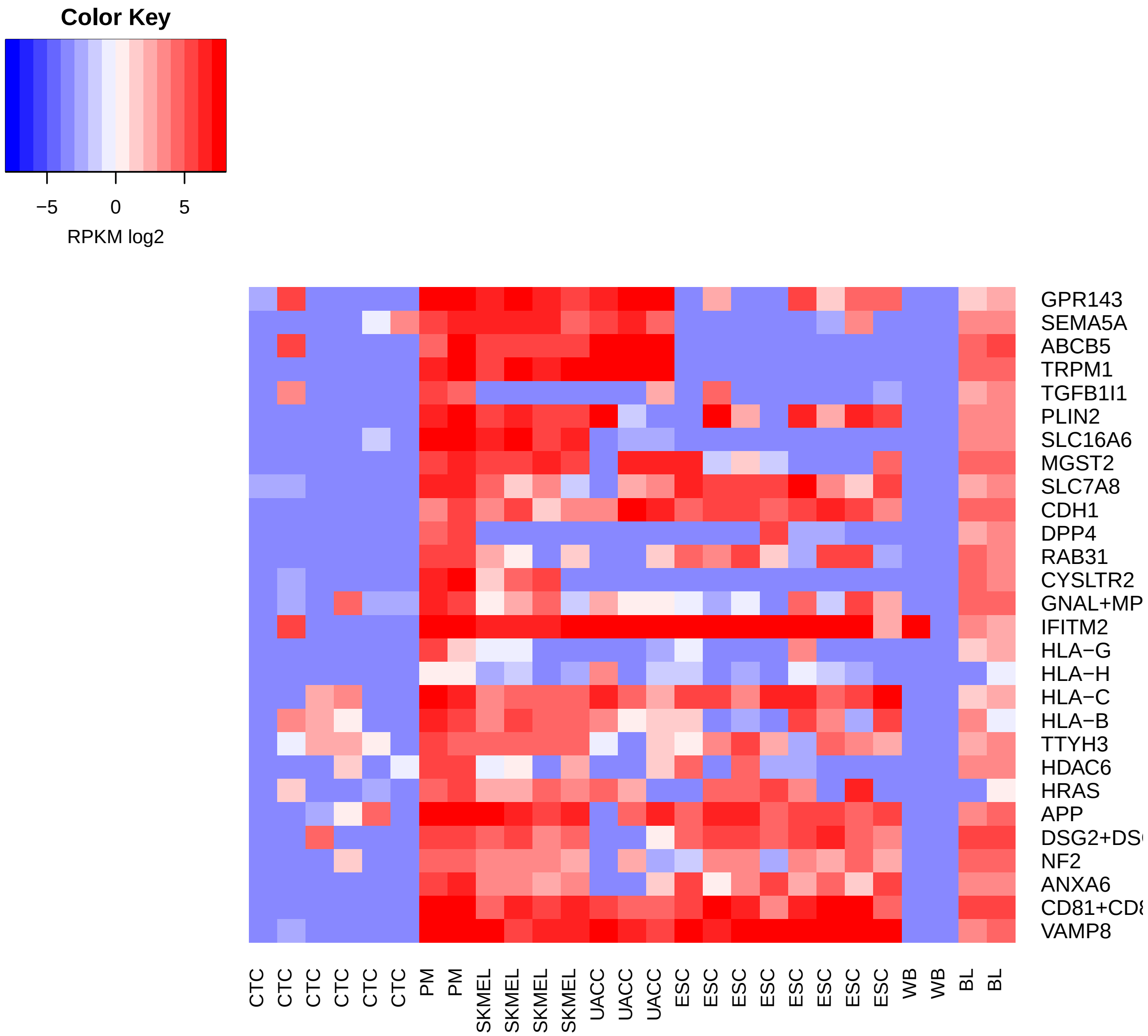
	ANOVA p-value	ANOVA q-value	skmel_1	skmel_2	skmel_3	skmel_4	uacc_1	uacc_2	uacc_3	pm_1	pm_2	ctc_1	ctc_2	ctc_3	ctc_4	ctc_5	ctc_6	pm-ctc	skmel-ctc	uacc-ctc	skmel-pm	uacc-pm	uacc-skmel
GJB1	0.0000970759	0.0049404852	6.12	6.11	6.57	6.98	3.08	6.52	6.04	0.00	0.00	7.05	4.43	6.68	6.06	5.75	5.56	0.000110648	0.8606121831	0.7688454329	0.0000876726	0.0008702258	0.4380953055
PRAME	0.0000000169	0.0000040282	7.18	7.82	7.91	7.72	7.62	8.09	8.18	0.00	0.00	6.90	9.14	8.63	7.95	8.34	7.61	0.0000000144	0.6515635278	0.9880147259	0.0000000454	0.000000052	0.8950328063
GPR126	0.0012470226	0.0304781205	2.37	3.53	5.38	4.39	0.00	0.00	2.24	0.00	0.00	4.51	3.07	4.73	4.05	3.11	2.41	0.0070193545	0.9782308143	0.0123044228	0.0064279949	0.86792823	0.0113745639
CRIM1	0.0007766786	0.0229789091	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.40	0.60	5.55	5.01	3.26	4.66	0.0109389433	0.0020642013	0.004074811	1	1	1
ABCG5	0.0011388686	0.0289473735	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.31	0.65	2.91	2.79	3.71	4.31	0.0147242625	0.0029404034	0.0056836826	1	1	1
SLC20A1	0.0015717156	0.0359951974	2.62	3.66	1.64	2.62	3.18	0.00	5.16	0.00	0.00	4.19	4.91	6.33	5.84	5.78	6.79	0.0016237513	0.0240812314	0.0511150898	0.1688317914	0.1686737375	0.9989822561
ADAM17	0.0011468641	0.0289473735	4.25	2.81	3.44	3.03	6.62	2.58	4.29	0.00	0.00	5.71	3.48	5.77	6.35	5.02	5.24	0.0007327487	0.1065602584	0.780758385	0.0249646941	0.0057130703	0.5954955901
RPS3	0.0010898185	0.0283146755	12.18	12.70	12.59	12.75	11.85	12.19	11.82	11.23	11.03	12.68	13.35	13.14	12.43	13.38	12.12	0.0010110338	0.6519568294	0.0325484915	0.0064724161	0.1512022487	0.2355870606
PSMB1	0.0018572522	0.0398377849	10.01	9.69	9.80	10.08	8.48	9.32	8.06	9.86	8.45	10.13	9.86	10.33	10.51	10.85	10.37	0.0452155972	0.4934507157	0.0016155834	0.3196802873	0.6286366605	0.0217423522

- Data for 9 genes generated during this project:

	CTC1	CTC2	CTC3	CTC4	CTC5	CTC6	PM1	PM2	SKMEL1	SKMEL2	SKMEL3	SKMEL4	UACC1	UACC2	UACC3	p.ANOVA	q.ANOVA	PMvsCTC.p.Tukey	SKMELvsCTC.p.Tukey	UACCvsCTC.p.Tukey	SKMELvsPM.p.Tukey	UACCvsPM.p.Tukey	UACCvsSKMEL.p.Tukey
GJB1	7.824396	5.186566	7.381308	6.646872	6.454946	6.210213	0.0000000	0.0000000	6.774579	6.761906	7.219762	7.6432863	3.849982	7.2062825	6.6826237	0.0000312	0.0026269	0.0000337	0.8804247	0.7625852	0.0000302	0.0002590	0.4530095
LL22NC03-63E9.3+PRAME	7.671798	9.907838	9.298326	8.538039	8.995661	8.034773	0.0000000	0.0000000	7.788570	8.483391	8.646811	8.3206468	8.334409	8.3367885	8.6932409	0.0000000	0.0000035	0.0000000	0.6860258	0.9032503	0.0000000	0.0000000	0.9882687
ADGRG6	5.161283	3.872159	5.167736	4.623820	3.887788	3.175038	0.4383693	0.0000000	3.131634	4.123306	5.933340	5.0954750	0.000000	0.0235495	3.0698949	0.0007718	0.0264125	0.0045787	0.9841691	0.0078299	0.0045197	0.8575949	0.0079317
CRIM1	6.036823	1.799073	6.232676	5.633584	4.031314	5.305246	0.0000000	0.6343014	0.000000	0.000000	0.000000	0.0172355	0.000000	0.0000000	0.9169042	0.0001073	0.0065024	0.0028044	0.0002391	0.0008918	0.9890369	0.9999995	0.9858419
ABCG5+DYNC2LI1	5.958574	2.231251	6.331449	7.526522	4.567064	5.356500	0.0000000	0.0638216	0.000000	6.647444	4.739803	7.2886904	0.000000	7.5157641	5.3263829	0.1708232	0.5149226	0.1298621	0.9801277	0.9436853	0.2456213	0.3509883	0.9974237
SLC20A1	4.869780	5.497657	6.930370	6.337518	6.358833	7.386363	0.2011737	0.0000000	3.417988	4.330383	2.544618	3.4066786	4.020708	0.1373316	5.6608377	0.0014532	0.0396871	0.0011996	0.0447388	0.0542499	0.0809413	0.1224083	0.9989399
ADAM17+IAH1	6.330511	8.811522	7.895789	8.282476	6.022133	8.217339	9.0746712	7.5419164	6.535850	6.986341	6.977785	6.8434814	8.052411	8.6610982	7.2474001	0.2526594	0.6238995	0.7644026	0.5753344	0.9229178	0.2828689	0.9783528	0.3764294
RPS3	9.057396	9.728213	9.520673	8.744502	9.811728	8.425373	7.5235992	7.3320722	8.614279	9.081603	8.946302	9.1815132	8.648596	8.7324657	8.1259749	0.0022480	0.0537808	0.0015821	0.7846640	0.1427282	0.0076482	0.0754732	0.5287168
PSMB1	10.295304	10.120670	10.558845	10.702183	11.099917	10.539397	10.0833632	8.6621295	10.256641	9.944024	10.036955	10.3009553	8.944355	9.5981673	8.2963848	0.0030056	0.0640519	0.0492212	0.5511161	0.0028776	0.3075790	0.7656162	0.0334537

Finally, loss of expression in plasma-membrane proteins in CTCs was investigated by the authors. The loss of membrane proteins makes cells less visible for the immune system, thus can escape control, drift to different regions of the body and cause metastasis. Figure shows 37 plasma membrane proteins which are , as expected, highly downregulated in CTCs:

```r
cols <- c(1:23, 36:39)
data.RPKM_h4 <- rbind(data.RPKM_log["GPR143", cols], data.RPKM_log["SEMA5A", cols],
                      data.RPKM_log["ABCB5", cols], data.RPKM_log["TRPM1", cols],
                      data.RPKM_log["TGFB1I1", cols], data.RPKM_log["PLIN2", cols],
                      data.RPKM_log["SLC16A6", cols], data.RPKM_log["MGST2", cols],
                      data.RPKM_log["SLC7A8", cols], data.RPKM_log["CDH1", cols],
                      data.RPKM_log["DPP4", cols], data.RPKM_log["RAB31", cols],
                      data.RPKM_log["CYSLTR2", cols], data.RPKM_log["GNAL", cols],
                      data.RPKM_log["IFITM2", cols], data.RPKM_log["HLA-G", cols],
                      data.RPKM_log["HLA-H", cols], data.RPKM_log["HLA-C", cols],
                      data.RPKM_log["HLA-B", cols], data.RPKM_log["TTYH3", cols],
                      data.RPKM_log["HDAC6", cols], data.RPKM_log["HRAS", cols],
                      data.RPKM_log["APP", cols], data.RPKM_log["DSG2", cols],
                      data.RPKM_log["NF2", cols], data.RPKM_log["ANXA6", cols],
                      data.RPKM_log["CD81", cols], data.RPKM_log["VAMP8", cols])
colnames(data.RPKM_h4) <- group[cols]

heatmap.2(as.matrix(data.RPKM_h4), dendrogram = "none", Colv=FALSE, Rowv=FALSE,
          tracecol = NA, col=colorRampPalette(c("blue", "white", "red")),
          breaks=c(-8:8), key=TRUE, symkey=FALSE, density.info="none",
          key.xlab = "RPKM log2", cexRow = 1)
```



Discussion

The main goal of this project was to apply methods learned during the course and reproduce real research results. Quantitatively, results were not always exactly matching those of the selected paper (PCA analysis for 12 cancer single sell samples, exact number of differentially expressed genes between pairs of samples).

This, among other reasons, might happen due to updated reference files I used for alignment and counting reads. Some small differences might also happen because of different shift value when taking a logarithm. However, with one exception the qualitative results of the original paper and this study were the same. The reason of different qualitative results in expression data heatmap of nine plasma membrane associated genes is unclear as the exact numbers provided by authors in **supplementary table 4** agree with the expression data obtained by this study.
Furthermore, in the future I would prefer using a different read counts expression than RPKM. Doing the research for this project, I found that RPKM were criticized by many researchers, see, for example Wagner, Kin, and Lynch (2012). Also, it would be interesting to apply negative binomial based models to the same raw counts data.

References

Plaks, Vicki, Charlotte D Koopman, and Zena Werb. 2013. "Circulating Tumor Cells." *Science* 341 (6151). American Association for the Advancement of Science: 1186–8.

Ramsköld, Daniel, Shujun Luo, Yu-Chieh Wang, Robin Li, Qiaolin Deng, Omid R Faridani, Gregory A Daniels, et al. 2012. "Full-Length mRNA-Seq from Single-Cell Levels of Rna and Individual Circulating Tumor Cells." *Nature Biotechnology* 30 (8). Nature Research: 777–82.

Wagner, Günter P, Koryu Kin, and Vincent J Lynch. 2012. "Measurement of mRNA Abundance Using Rna-Seq Data: RPKM Measure Is Inconsistent Among Samples." *Theory in Biosciences* 131 (4). Springer: 281–85.