

PCA, UMAP and Hierarchical Clustering Boeva Neuroblastoma Cells

Gepoliano Chaves

February 25, 2021

Contents

1) Objectives of Analysis	1
2) Creating Analysis folders	2
2. 2) Check File Structure	2
3) UMAP Visualization - RNA-seq	3
Install libraries	3
Import and set up data	4
4) PCA with DESeq	6
4.1) PCA with DESeq	6
Hierarchical Clustering Heatmap	9
4.2) Plot PCA using expression_df object	10
5) References	11
PCA plots	11
PCA GGfortify	11
PCA datacarpentry	11
UMAP plots Alexe's Lemonade Stand Foundation	11
UMAP Plots Tutorial with iris dataset, requires R function	12
UMAP Package	12
UMAPR Package	12

1) Objectives of Analysis

This notebook uses RNA-Seq data from a published paper (Boeva et al. 2017, Nature Genetics, doi:10.1038/ng.3921) to compare Hierarchical Clustering, Principal Component and Uniform Manifold Approximation and Projection (UMAP) analysis for the classification of cells derived from pediatric cancer neuroblastoma, described by Boeva et al., 2017.

The article brings the concept of neuroblastoma as a heterogeneous tumor composed fundamentally of two distinct cell populations: adrenergic and mesenchymal cells. Adrenergic cells represent a group of differentiated cells in the neuroblastoma tumor, which are more easily targeted by chemotherapy than mesenchymal cells.

On the other hand, chemotherapy may leave Minimal Residual Disease in the form of mesenchymal cells which may lead to disease progression after treatment, by allowing tumor to continue its malignant development. One possible therapeutic venue is the use of drugs that break the homeostatic equilibrium between the

adrenergic and mesenchymal states of cells and make cells become more differentiated, or turn mesenchymal cells into adrenergic cells.

We are currently developing a study to quantify ADRN and MES phenotypes using wither RNA-Seq or epigenetic modification 5hmC profiles of cells, tumors and cfDNA. Our hypotheses is that MES scores can be measured from cfDNA 5hmC differentially expressed gene lists.

2) Creating Analysis folders

File organization is essential for clear and neat procedure steps. The chunk bellow creates folders for the data, plots and results in the same directory where the Rmd notebook is saved.

```
# Create the data folder if it doesn't exist
if (!dir.exists("data")) {
  dir.create("data")
}

# Define the file path to the plots directory
plots_dir <- "plots"

# Create the plots folder if it doesn't exist
if (!dir.exists(plots_dir)) {
  dir.create(plots_dir)
}

# Define the file path to the results directory
results_dir <- "results"

# Create the results folder if it doesn't exist
if (!dir.exists(results_dir)) {
  dir.create(results_dir)
}
```

2. 2) Check File Structure

The analysis folder, downloaded from GitHub, where the Rmd notebook is saved, should contain:

- The example analysis .Rmd downloaded
- A folder called “data” which contains:
 - The Boeva folder which contains:
 - * The gene expression file GSE90683_Log2FPKMExpressionSummary.txt
 - * The metadata file metadata_Boeva_Modified.txt
- A folder for **plots** (currently empty)
- A folder for **results** (currently empty)

Your example analysis folder should now look something like this.

In order for the example here to run without a hitch, we need these files to be in these locations. The next chunks are a test to check before we get started with the analysis. These chunks will declare your file paths and double check that your files are in the right place.

First we will declare our file paths to our data and metadata files, which should be in our data directory. This is handy to do because if we want to switch the dataset (see next section for more on this) we are using for this analysis, we will only have to change the file path here to get started.

```
# Define the file path to the data directory
# Replace with the path of the folder the files will be in
data_dir <- file.path("data", "Boeva")

# Declare the file path to the gene expression matrix file
# inside directory saved as `data_dir`
# Replace with the path to your dataset file
data_file <- file.path(data_dir, "GSE90683_Log2FPKMEExpressionSummary.txt")

# Declare the file path to the metadata file
# inside the directory saved as `data_dir`
# Replace with the path to your metadata file
metadata_file <- file.path(data_dir, "metadata_SRP133573.tsv")
metadata_file <- file.path(data_dir, "metadata_Boeva_Modified.txt")
```

Now that our file paths are declared, we can use the `file.exists()` function to check that the files are where we specified above.

```
# Check if the gene expression matrix file is at the path stored in `data_file`
file.exists(data_file)
```

```
## [1] TRUE
```

```
# Check if the metadata file is at the file path stored in `metadata_file`
file.exists(metadata_file)
```

```
## [1] TRUE
```

3) UMAP Visualization - RNA-seq

Install libraries

We will use libraries DESeq2, umap, ggplot2 and magrittr. If you do not have the libraries installed, install them. If you have these libraries installed, you can skip this step.

```
if (!("DESeq2" %in% installed.packages())) {
  # Install DESeq2
  BiocManager::install("DESeq2", update = FALSE)
}

if (!("umap" %in% installed.packages())) {
  # Install umap package
  BiocManager::install("umap", update = FALSE)
}

if (!("ggfortify" %in% installed.packages())) {
  # Install ggfortify package
  BiocManager::install("ggfortify", update = FALSE)
}

if (!("pheatmap" %in% installed.packages())) {
```

```
# Install pheatmap package
BiocManager::install("pheatmap", update = FALSE)
}
```

Attach packages used in this analysis:

```
# Attach the `DESeq2` library
library(DESeq2)

# Attach the `umap` library
library(umap)

# Attach the `ggplot2` library for plotting
library(ggplot2)

# We will need this so we can use the pipe: %>%
library(magrittr)

# Attach the `ggfortify` library for PCA using prcomp()
library(ggfortify)

# Attach the `pheatmap` library for PCA using prcomp()
library(pheatmap)

# Set the seed so our results are reproducible:
set.seed(12345)
```

Import and set up data

In the chunk below, the gene expression table needs to be downloaded from the Gene Expression Omnibus or used directly from this repository. It was not possible to import the gene expression data-frame using the exact same code provided in the original ALSF Rmd notebook, because that was giving an error due to duplicated gene names. The

```
# Read in metadata TSV file and gene expression data-frame file.
metadata <- readr::read_tsv(metadata_file)
```

```
##
## -- Column specification -----
## cols(
##   .default = col_character(),
##   refinebio_age = col_logical(),
##   refinebio_cell_line = col_logical(),
##   refinebio_compound = col_logical(),
##   refinebio_disease_stage = col_logical(),
##   refinebio_genetic_information = col_logical(),
##   refinebio_processed = col_logical(),
##   refinebio_sex = col_logical(),
##   refinebio_source_archive_url = col_logical(),
##   refinebio_specimen_part = col_logical(),
##   refinebio_time = col_logical()
## )
## i Use `spec()` for the full column specifications.
```

```

data_file <- file.path(data_dir, "GSE90683_Log2FPKMExpressionSummary.txt") ## repeated from second chunk

# Original code from ALSF:
# Read in data TSV file
# First time, there was this error: Error in `.rowNamesDF<-(x, value = value) : duplicate 'row.names'
# expression_df <- readr::read_tsv(data_file, ) %>%
# Tuck away the gene ID column as row names, leaving only numeric values
# tibble::column_to_rownames("gene")

## Import gene expression data-frame using the entire path provided by downloading our repository.
expression_df <- read.table("data/Boeva/GSE90683_Log2FPKMExpressionSummary.txt", header = T)
## This way of gene expression data-frame upload needs to deal with the duplicated gene names problem.
## Need to remove repeated gene names with line of code below.
## Remove duplicated rows
expression_df <- expression_df[!duplicated(expression_df$gene), ]
## Name rows with gene names
rownames(expression_df) <- expression_df$gene
## Remove Gene Extra-column
expression_df <- subset(expression_df, select = -c(gene))

```

Check that metadata and data are in the same sample order.

```

# Make the data in the order of the metadata
expression_df <- expression_df %>%
  dplyr::select(metadata$refinebio_accession_code)

# Check if this is in the same order
all.equal(colnames(expression_df), metadata$refinebio_accession_code)

```

```
## [1] TRUE
```

Choose metadata annotation columns to be used in analysis.

```

# convert the columns we will be using for annotation into factors
metadata_metadata <- metadata %>%
  dplyr::select( # select only the columns that we will need for plotting
    refinebio_accession_code,
    refinebio_treatment,
    refinebio_disease
  )

```

Set minimum of counts to be used in analysis.

```

filtered_expression_df <- expression_df %>%
  dplyr::filter(rowSums(.) >= 100)

```

Counts need to be rounded before their values are passed to DESeqDataSetFromMatrix() function.

```
filtered_expression_df <- round(filtered_expression_df)
```

The chunk below creates DESeqDataSet object from gene expression data-frame as input. This highlights the DESeqDataSetFromMatrix function.

```

dds <- DESeqDataSetFromMatrix(
  countData = filtered_expression_df, # the counts values for all samples in our dataset
  colData = metadata, # annotation data for the samples in the counts data frame
  design = ~1 # Here we are not specifying a model
  # Replace with an appropriate design variable for your analysis
)

```

```
)
```

```
## converting counts to integer mode
```

4) PCA with DESeq

4.1) PCA with DESeq

```
combinedDNAHTSeq<- DESeq(dds)
```

```
## Warning in DESeq(dds): the design is ~ 1 (just an intercept). is this intended?
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## -- note: fitType='parametric', but the dispersion trend was not well captured by the  
## function: y = a/x + b, and a local regression fit was automatically substituted.  
## specify fitType='local' or 'mean' to avoid this message next time.
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

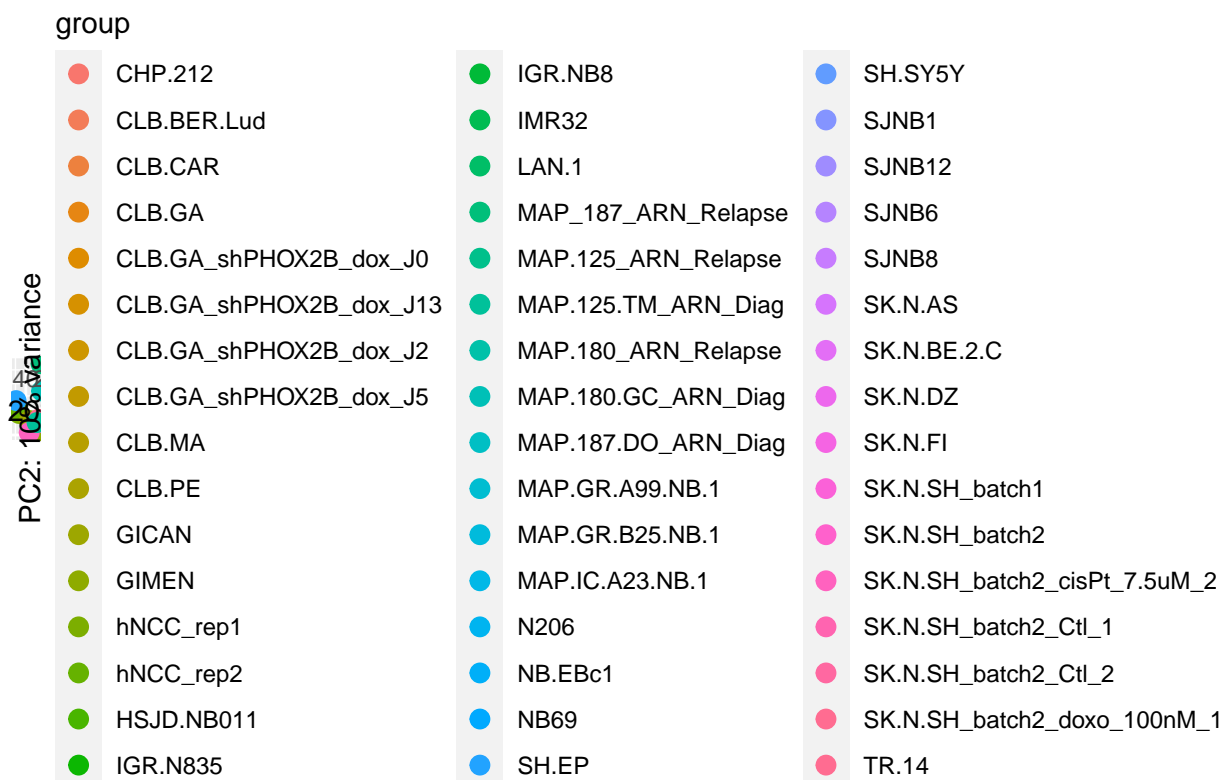
```
summary(combinedDNAHTSeq)
```

```
## [1] "DESeqDataSet object of length 11298 with 19 metadata columns"
```

```
vst_Boeva <- vst(combinedDNAHTSeq, blind=FALSE)
```

```
## Plot PCA
```

```
plotPCA(vst_Boeva, intgroup = c("refinebio_accession_code"))
```



colData(dds)

```
## DataFrame with 48 rows and 20 columns
##               refinebio_accession_code experiment_accession refinebio_age
##               <character>           <character>           <logical>
## SJNB6                SJNB6                SRP133573                NA
## SJNB8                SJNB8                SRP133573                NA
## SK.N.AS              SK.N.AS                SRP133573                NA
## CLB.CAR              CLB.CAR                SRP133573                NA
## CLB.PE              CLB.PE                SRP133573                NA
## ...                  ...                  ...                  ...
## MAP.180.GC_ARN_Diag  MAP.180.GC_ARN_Diag  SRP133573                NA
## MAP.187.DO_ARN_Diag  MAP.187.DO_ARN_Diag  SRP133573                NA
## MAP.180_ARN_Relapse  MAP.180_ARN_Relapse  SRP133573                NA
## MAP.125_ARN_Relapse  MAP.125_ARN_Relapse  SRP133573                NA
## MAP_187_ARN_Relapse  MAP_187_ARN_Relapse  SRP133573                NA
##               refinebio_cell_line refinebio_compound
##               <logical>           <logical>
## SJNB6                NA                NA
## SJNB8                NA                NA
## SK.N.AS              NA                NA
## CLB.CAR              NA                NA
## CLB.PE              NA                NA
## ...                  ...                  ...
## MAP.180.GC_ARN_Diag  NA                NA
## MAP.187.DO_ARN_Diag  NA                NA
## MAP.180_ARN_Relapse  NA                NA
## MAP.125_ARN_Relapse  NA                NA
## MAP_187_ARN_Relapse  NA                NA
```

```

##          refinebio_disease refinebio_disease_stage
##          <character>          <logical>
## SJNB6          radical prostatectomy          NA
## SJNB8          radical prostatectomy          NA
## SK.N.AS        radical prostatectomy          NA
## CLB.CAR        radical prostatectomy          NA
## CLB.PE          biopsy          NA
## ...          ...          ...
## MAP.180.GC_ARN_Diag radical prostatectomy          NA
## MAP.187.DO_ARN_Diag          biopsy          NA
## MAP.180_ARN_Relapse          biopsy          NA
## MAP.125_ARN_Relapse          biopsy          NA
## MAP_187_ARN_Relapse          biopsy          NA
##          refinebio_genetic_information refinebio_organism
##          <logical>          <character>
## SJNB6          NA          HOMO_SAPIENS
## SJNB8          NA          HOMO_SAPIENS
## SK.N.AS        NA          HOMO_SAPIENS
## CLB.CAR        NA          HOMO_SAPIENS
## CLB.PE          NA          HOMO_SAPIENS
## ...          ...          ...
## MAP.180.GC_ARN_Diag          NA          HOMO_SAPIENS
## MAP.187.DO_ARN_Diag          NA          HOMO_SAPIENS
## MAP.180_ARN_Relapse          NA          HOMO_SAPIENS
## MAP.125_ARN_Relapse          NA          HOMO_SAPIENS
## MAP_187_ARN_Relapse          NA          HOMO_SAPIENS
##          refinebio_platform refinebio_processed
##          <character>          <logical>
## SJNB6          Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## SJNB8          Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## SK.N.AS        Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## CLB.CAR        Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## CLB.PE          Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## ...          ...          ...
## MAP.180.GC_ARN_Diag Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## MAP.187.DO_ARN_Diag Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## MAP.180_ARN_Relapse Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## MAP.125_ARN_Relapse Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
## MAP_187_ARN_Relapse Illumina HiSeq 2500 (IlluminaHiSeq2500)          TRUE
##          refinebio_race refinebio_sex refinebio_source_archive_url
##          <character>          <logical>          <logical>
## SJNB6          white          NA          NA
## SJNB8          white          NA          NA
## SK.N.AS        white          NA          NA
## CLB.CAR        white          NA          NA
## CLB.PE          white          NA          NA
## ...          ...          ...          ...
## MAP.180.GC_ARN_Diag          white          NA          NA
## MAP.187.DO_ARN_Diag          white          NA          NA
## MAP.180_ARN_Relapse          white          NA          NA
## MAP.125_ARN_Relapse          white          NA          NA
## MAP_187_ARN_Relapse          white          NA          NA
##          refinebio_source_database refinebio_specimen_part
##          <character>          <logical>

```



```

## SJNB6                                SRA                                NA
## SJNB8                                SRA                                NA
## SK.N.AS                             SRA                                NA
## CLB.CAR                             SRA                                NA
## CLB.PE                              SRA                                NA
## ...                                  ...                                ...
## MAP.180.GC_ARN_Diag                  SRA                                NA
## MAP.187.DO_ARN_Diag                  SRA                                NA
## MAP.180_ARN_Relapse                  SRA                                NA
## MAP.125_ARN_Relapse                  SRA                                NA
## MAP_187_ARN_Relapse                  SRA                                NA
##                                     refinebio_subject refinebio_time refinebio_title
##                                     <character>      <logical>      <character>
## SJNB6                prostate tumor tissue          NA      CHU001.RP
## SJNB8                prostate tumor tissue          NA      CHU001.RP
## SK.N.AS              prostate tumor tissue          NA      CHU001.RP
## CLB.CAR              prostate tumor tissue          NA      CHU001.RP
## CLB.PE              prostate tumor tissue          NA      CHU001.Bx
## ...                  ...                          ...          ...
## MAP.180.GC_ARN_Diag prostate tumor tissue          NA      CHU009.RP
## MAP.187.DO_ARN_Diag prostate tumor tissue          NA      CHU009.Bx
## MAP.180_ARN_Relapse prostate tumor tissue          NA      CHU009.Bx
## MAP.125_ARN_Relapse prostate tumor tissue          NA      CHU009.Bx
## MAP_187_ARN_Relapse prostate tumor tissue          NA      CHU009.Bx
##                                     refinebio_treatment
##                                     <character>
## SJNB6                post.adt
## SJNB8                post.adt
## SK.N.AS              post.adt
## CLB.CAR              post.adt
## CLB.PE              pre.adt
## ...                  ...
## MAP.180.GC_ARN_Diag post.adt
## MAP.187.DO_ARN_Diag pre.adt
## MAP.180_ARN_Relapse pre.adt
## MAP.125_ARN_Relapse pre.adt
## MAP_187_ARN_Relapse pre.adt

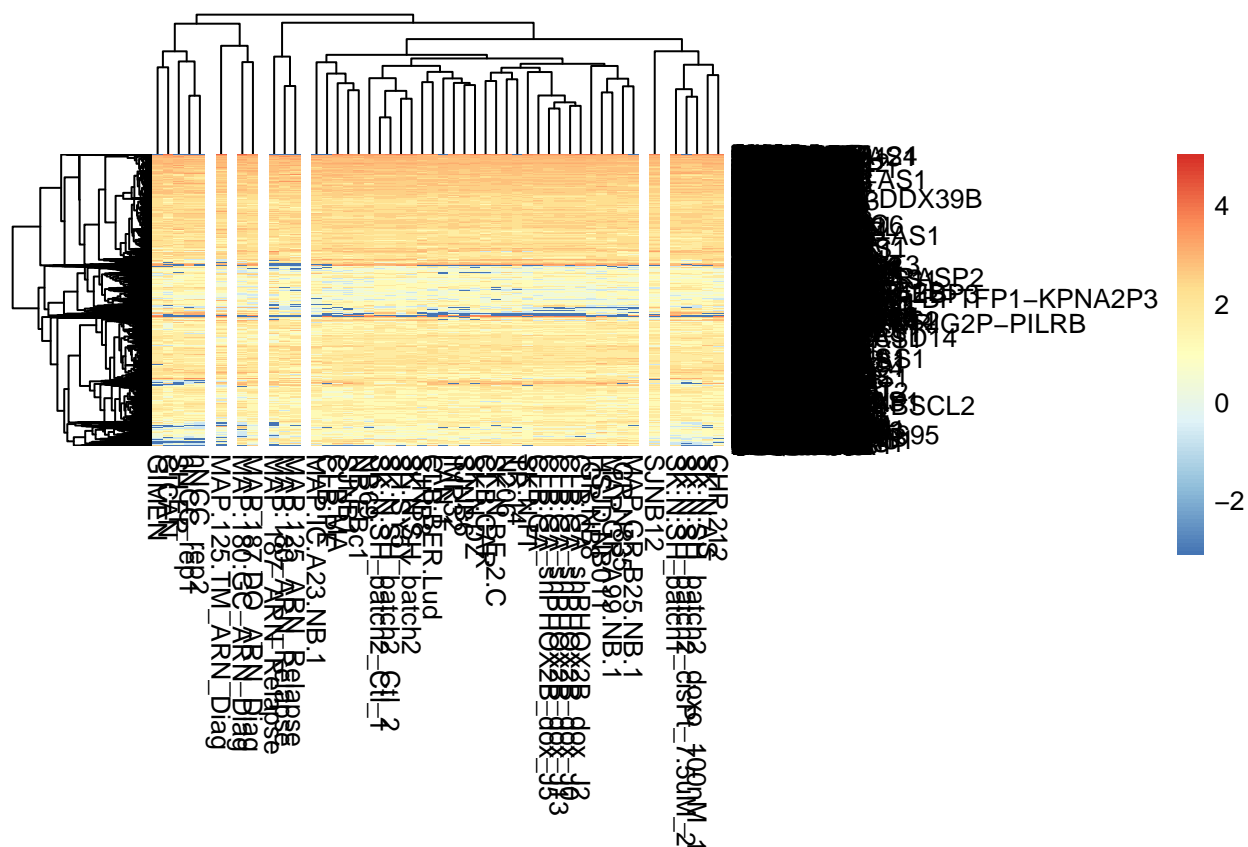
```

Hieracrchical Clustering Heatmap

```

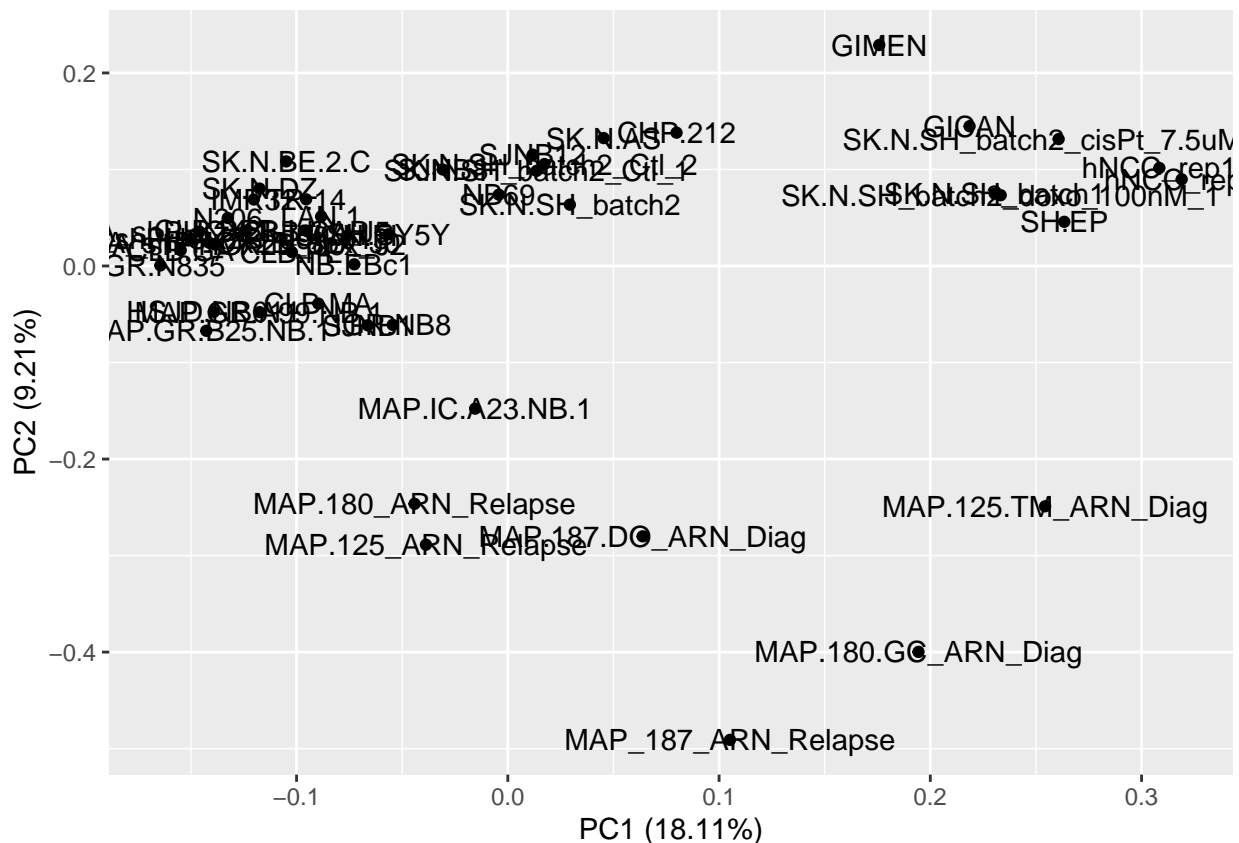
## Set counts threshold to 100 for ease of plotting
library(pheatmap)
pheatmap(assay(vst_Boeva), cutree_cols = 7)

```



4.2) Plot PCA using expression_df object

```
pca_res <- prcomp(t(expression_df[, -1]))
autoplot(pca_res, label = TRUE, label.size = 4)
```



5) References

PCA plots

PCA GGfortify

https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_pca.html

PCA datacarpentry

https://tavareshugo.github.io/data-carpentry-rnaseq/03_rnaseq_pca.html

<https://tavareshugo.github.io/data-carpentry-rnaseq/>

<https://github.com/tavareshugo/data-carpentry-rnaseq/find/master>

<https://datacarpentry.org/R-ecology-lesson/>

UMAP plots Alexe's Lemonade Stand Foundation

<https://github.com/AlexsLemonade/refinebio-examples>

https://alexslemonade.github.io/refinebio-examples/03-rnaseq/dimension-reduction_rnaseq_02_unap.html#analysis

UMAP Plots Tutorial with iris dataset, requires R function

<https://cran.r-project.org/web/packages/umap/vignettes/umap.html>

UMAP Package

<https://github.com/lmcinnes/umap#installing>

UMAPR Package

<https://github.com/ropenscilabs/umapr>