

PALMO (Platform for Analyzing Longitudinal Multi-omics data) package

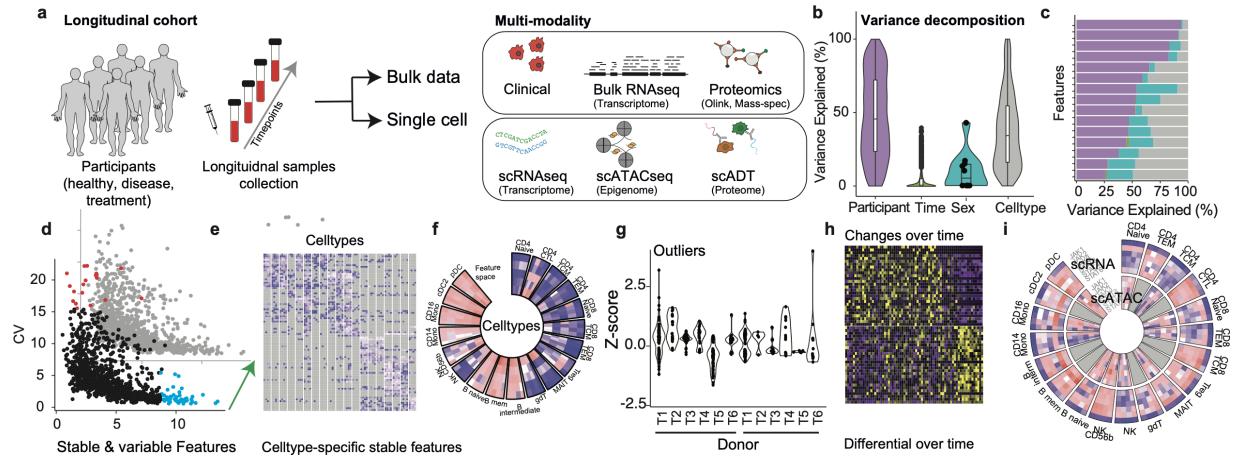
Last compiled on 11 May, 2022

Contents

1	Introduction	1
2	Install package and load library	2
3	Tutorials	3
3.1	Tutorial-1: Plasma proteome (Bulk dataset)	3
3.2	Tutorial-2: scRNA longitudinal data (n=4 and 6 weeks follow-up)	11
3.3	Tutorial-3: scATAC Longitudinal data (n=4 and 6 weeks follow-up)	21
3.4	Tutorial-4: Multi-modal data integration	29
3.5	Tutorial-5: COVID19 longitudinal dataset (CNP0001102)	31
3.6	Tutorial-6: Differential Gene analysis in longitudinal data (CNP0001102)	39
3.7	Tutorial-7: Mouse brain dataset (GSE129788)	41
4	Authors	49
5	License	49
6	Session info	50

1 Introduction

PALMO (Platform for Analyzing Longitudinal Multi-omics data) is a platform for analyzing longitudinal data from bulk as well as single cell. It allows to identify inter-, intra-donor variations in genes over longitudinal time points. The analysis can be done on bulk expression dataset without known celltype information or single cell with celltype/user-defined groups. It allows to infer stable and variable features in given donor and each celltype (or user defined group). The outlier analysis can be performed to identify technical/biological perturbed samples in donor/participant. Further, differential analysis can be performed to decipher time-wise changes in gene expression in a celltype.



General workflow and analysis schema of **PALMO**. It can work with longitudinal data obtained from bulk such as clinical, bulk RNAseq, proteomic or single cell dataset from scRNAseq, and scATACseq.

2 Install package and load library

To install library, simply run

```
library("devtools")
install_github("aifimmunology/PALMO")
library("PALMO")
```

```
library(PALMO)
#> Loading required package: grid
```

3 Tutorials

3.1 Tutorial-1: Plasma proteome (Bulk dataset)

This tutorial allows users to explore bulk plasma proteome measured from 6 healthy donors over 10 time-points. Plasma proteomic data available at github. 1. AIFI-Olink-NPX_log2_Protein.Rda (Normalized protein expression data) 2. AIFI-Metadata.Rda (clinical metadata). Longitudinal dataset includes 6 donors (3 male and 3 females). PBMC samples were collected from 6 donors over 10 weeks. To interrogate longitudinal data, please follow following steps.

3.1.1 Load Library

```
#Load Library and other vizualization packages
library("PALMO")
library("Hmisc")
library("ggpubr")
library("cowplot")
```

3.1.2 Load data and assign paramaters (Time ~ 10sec)

The annotation table `metadata` must consist of column `Sample` (Participant sample name), `PTID` (donor/Participant), `Time` (longitudinal time points) information, but not necessary same column names. Users can assign respective columns in subsequent steps. The datamatrix is an Expression data frame, where rows represents gene/proteins and column represents participant samples (same as annotation table `Sample` column).

```
#Load Plasma proteome data (longitudinal)
load("data/AIFI-Olink-NPX_log2_Protein.Rda")
#Load metadata
load("data/AIFI-Metadata.Rda")
```

3.1.3 Create PALMO object and merge annotation data (Time ~20sec)

The expression dataframe annotations merged with input annotation dataframe. Only overlapping samples kept. Missing annotations with `Sample`, `Donor/participant`, or `Time` columns are removed from downstream analysis.

```
#Create PALMO object
palmo_obj <- createPALMOobject(anndata=ann, data=data)

#Assign Sample, PTID and Time parameters
palmo_obj<- annotateMetadata(data_object=palmo_obj,
                               sample_column= "Sample", donor_column= "PTID",
                               time_column= "Time")

#Sample overlap and final matrix
palmo_obj <- mergePALMOdata(data_object=palmo_obj, datatype="bulk")

#Check for replicates
palmo_obj <- checkReplicates(data_object=palmo_obj, mergeReplicates = T)
```

3.1.4 Remove genes with >40%NAs (optional)

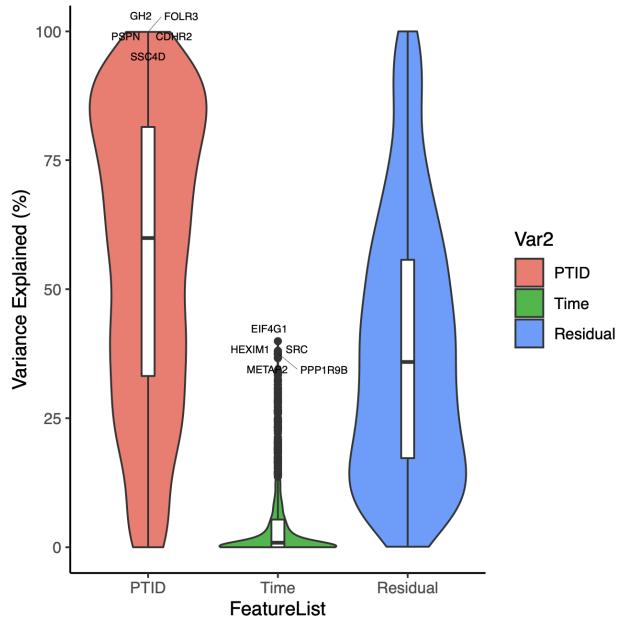
For downstream analysis select genes/proteins with less than 40% of missing values. Users can select cut-off for missing values as necessary.

```
palmo_obj <- naFilter(data_object=palmo_obj, na_cutoff=0.4)
```

3.1.5 Features contributing towards donor variations (Variance decomposition) (Time ~1min)

To perform variance decomposition apply `lmeVariance` function with input metadata, and datamatrix. The `featureSet` is a list of variables to which freaction variance explained by each gene is attributed. `meanThreshold` defines the minimum average expression threshold to be used for ongitudinal dataset. Here we used normalized protein expression 1 based on mean expression profile of each gene across longitudinal samples. Residuals suggest the variance can not be explained by available feature set. The variance explained by each gene towards the featureSet of interest given in percentage.

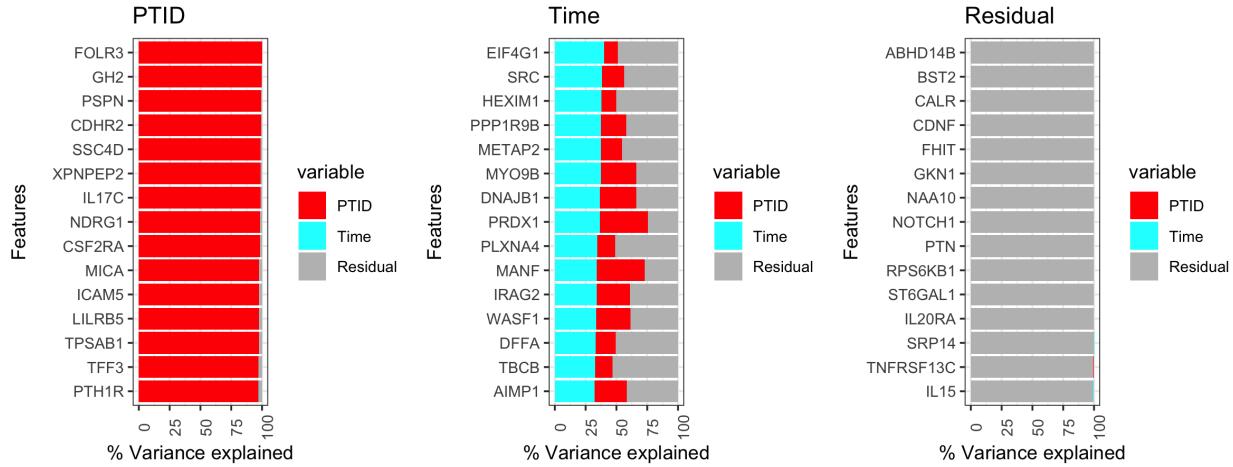
```
featureSet <- c("PTID", "Time")
palmo_obj <- lmeVariance(data_object=palmo_obj, featureSet=featureSet,
                           meanThreshold=1, fileName="olink")
var_decomp <- palmo_obj@result$variance_decomposition
```



```
head(var_decomp[,c(featureSet, "Residual")])
#Features      donor      week  Residuals
#FOLR3    99.90070 0.0000000 0.09930098
#GH2      99.49856 0.0000000 0.50144042
#PSPN     99.26882 0.1021076 0.62906798
#CDHR2     99.07933 0.1157406 0.80493162
#SSC4D     98.82794 0.0000000 1.17206000
#XPNPEP2   98.67628 0.0000000 1.32372323
```

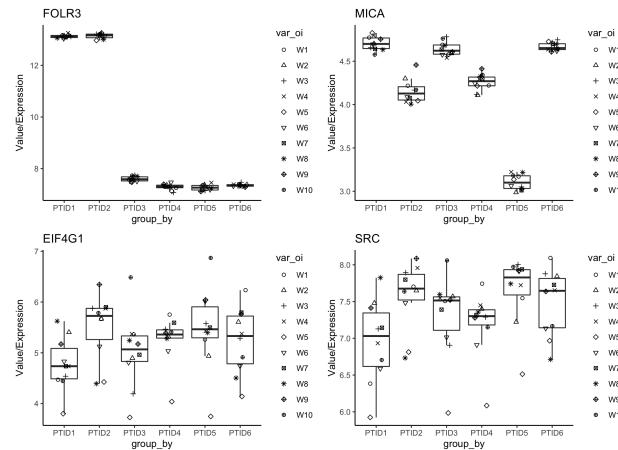
3.1.6 Donor-specific variance contributing features

```
plots <- variancefeaturePlot(vardata=var_decomp, featureSet=featureSet,
                               Residual=T)
plot_grid(plotlist=plots, nrow = 1, align="hv")
```



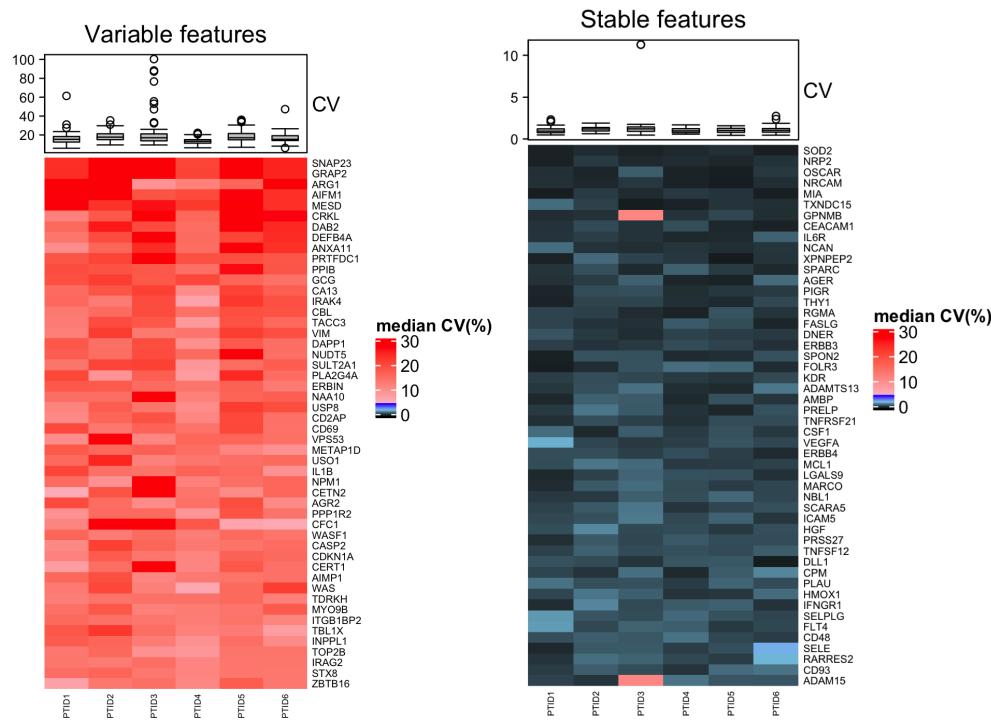
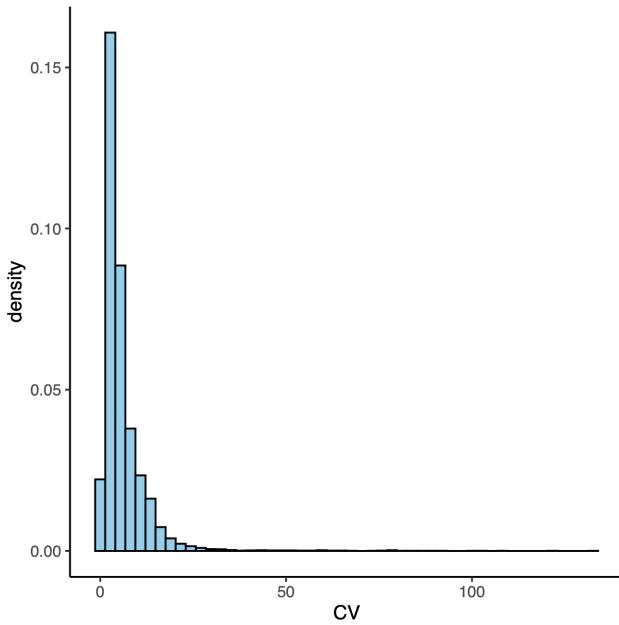
3.1.7 Plot top features

```
plots <- gene_featureplot(data_object=palmo_obj,
                           featureList=c("FOLR3", "MICA", "EIF4G1", "SRC"),
                           x_group_by="PTID", var_oi="Time")
plot_grid(plotlist=plots, ncol= 2, align="hv")
```



3.1.8 Intra-donor variations over time

```
#CV vs Mean
palmo_obj <- cvCalcBulk(data_object=palmo_obj, meanThreshold=1, cvThreshold=5)
```

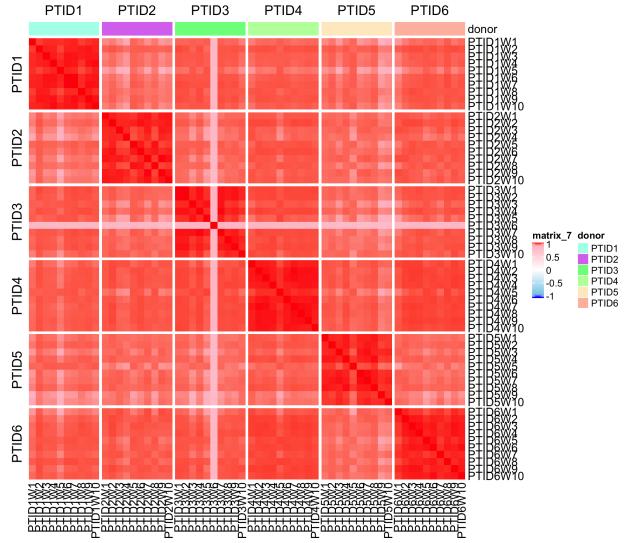


```
#Variable genes
head(palmo_obj@result[["variable_gene"]])
head(palmo_obj@result[["var_mat"]])
#Non-variable genes (stable)
head(palmo_obj@result[["non_variable_gene"]])
head(palmo_obj@result[["stable_mat"]])
```

3.1.9 Outlier analysis (Time ~ 30sec)

Perform the sample correlation to find out overall correlation between longitudinal samples.

```
#Sample variability (Correlation)
palmo_obj <- sample_correlation(data_object=palmo_obj, donor_sep="W")
```

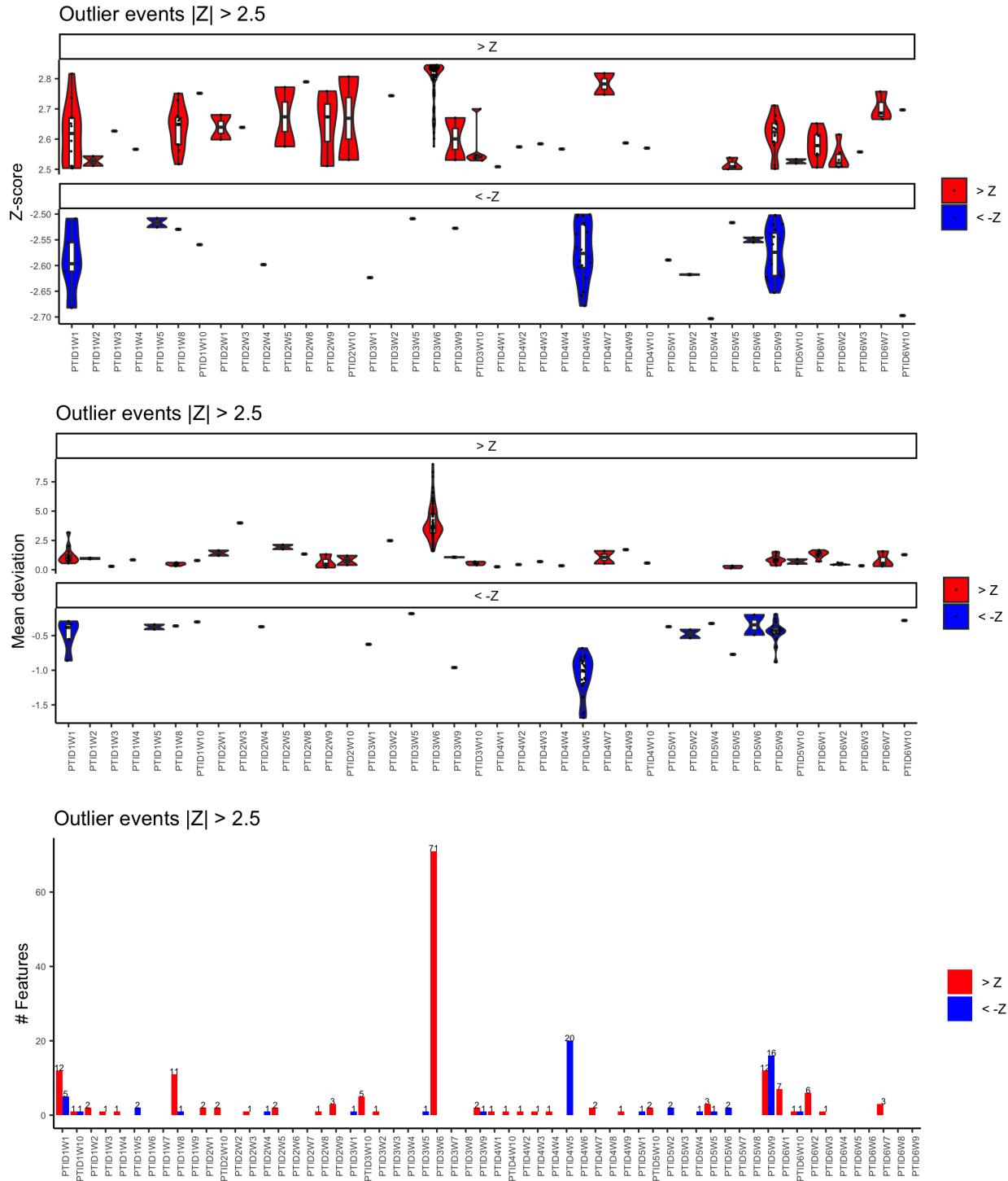


```
#Detect outliers (if any)
palmo_obj <- outlierDetect(data_object=palmo_obj, z_cutoff=2.5)
outlier_res <- palmo_obj@result[["outlier_res"]]
head(outlier_res)

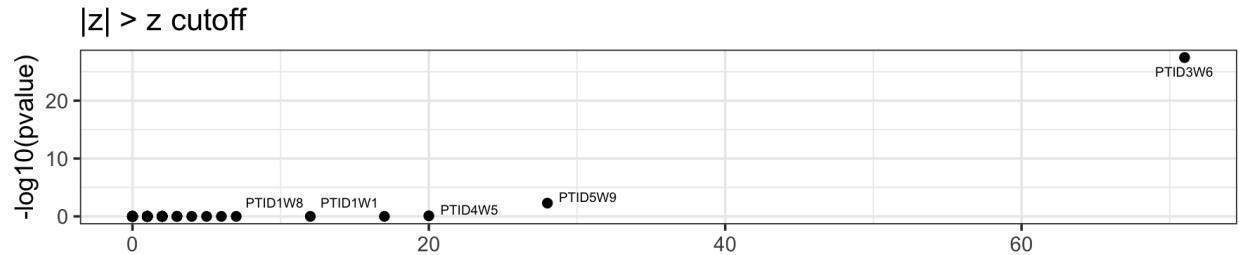
# exp Sample PTID Time Sex gene meanDev z
#PTID3W643 10.729880 PTID3W6 PTID3 W6 Female IFI30 8.340474 2.845471
#PTID3W627 10.133645 PTID3W6 PTID3 W6 Female DPEP2 6.595705 2.844629
#PTID3W631 10.188437 PTID3W6 PTID3 W6 Female FCAR 7.004890 2.844607
#PTID3W626 7.656418 PTID3W6 PTID3 W6 Female DPEP1 4.574449 2.844574
#PTID3W685 9.129149 PTID3W6 PTID3 W6 Female TNFRSF13C 6.605767 2.844410 2.844410
#PTID3W652 8.031215 PTID3W6 PTID3 W6 Female KIR2DL3 5.156982 2.844018

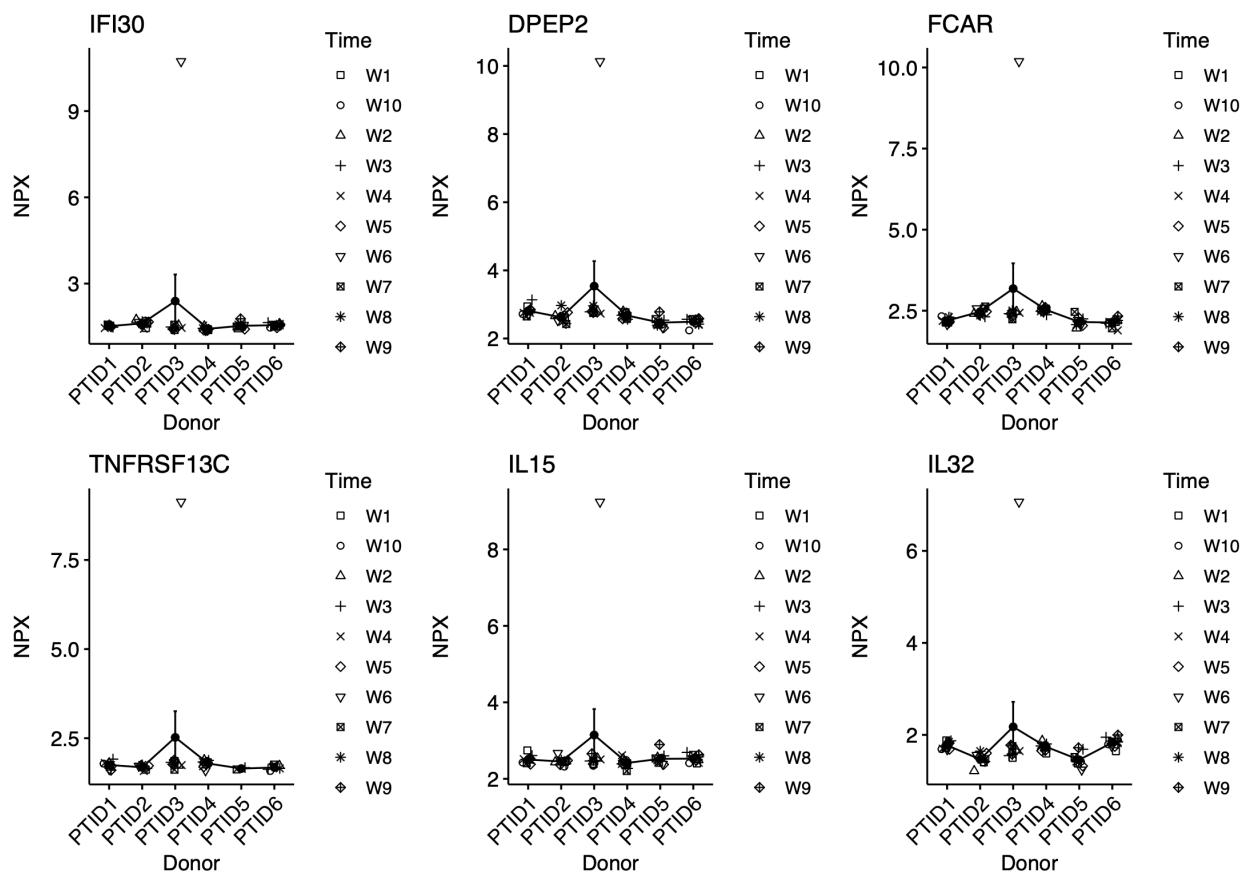
#Number of outlier features
num_outlier <- data.frame(table(outlier_res$Sample))
num_outlier <- num_outlier[order(num_outlier$Freq, decreasing = T),]
head(num_outlier)

#Var1 Freq
#PTID3W6 71
#PTID5W9 28
#PTID4W5 20
#PTID1W1 17
#PTID1W8 12
#PTID6W1 7
```



```
#Calculate p value
outlierDetectP(outlier_events=outlier_res, z_cutoff=2.5, nGenes=1042)
```





3.2 Tutorial-2: scRNA longitudinal data (n=4 and 6 weeks follow-up)

This tutorial allows users to explore single cell RNAseq data measured from 4 healthy donors over 6 time points (week 2-7). Single cell data available at GSE190992. (1) pbmc_longitudinal_data (Normalized scRNA seurat object) (2) AIFI-Metadata.Rda (clinical metadata). Longitudinal data set includes 4 donors and 24 samples. To infer inter-donor, intra-donor variations, and stable features, please follow following steps.

3.2.1 Load Library

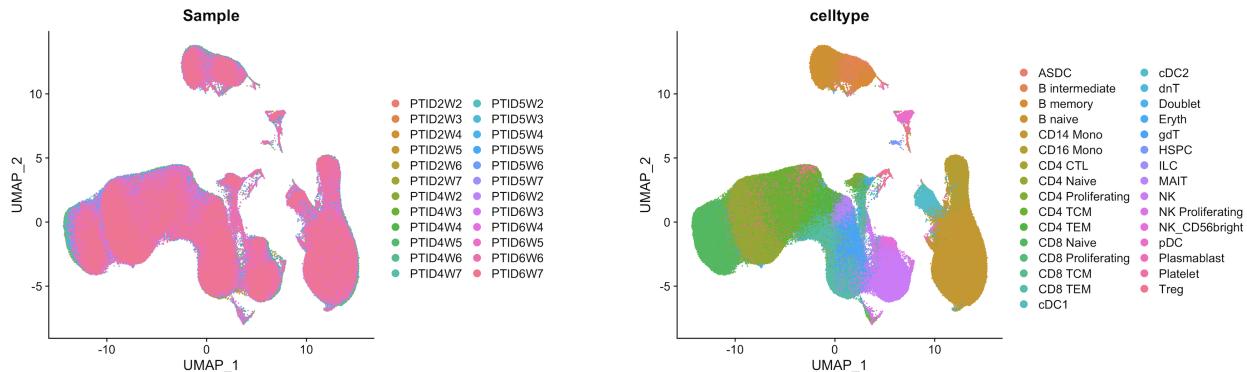
```
#Load Library
library("PALMO")
library("Hmisc")
library("ggpubr")
library("cowplot")
```

3.2.2 Load single cell data and metadata

```
#scRNA seurat object
pbmc <- readRDS("data/AIFI-scRNA-PBMC-FinalData.RDS")
metaData <- pbmc@meta.data
pbmc@meta.data$Sample <- pbmc@meta.data$orig.ident

#Load annotation data
load("data/AIFI-Metadata.Rda")

library("Seurat")
#UMAP plot
p1 <- DimPlot(object = pbmc, reduction = 'umap', group.by = "Sample", label = F)
p2 <- DimPlot(object = pbmc, reduction = 'umap', group.by = "celltype", label = F)
print(plot_grid(p1, p2, align="hv", ncol=2))
```



```
#Celltypes observed in dataset
cell_type <- sort(unique(pbmc@meta.data$celltype))
#Celltypes selected for analysis consisting atleast >5% of cells in each celltype.
celltype_oi <- c("CD4_Naive", "CD4_TEM", "CD4_TCM", "CD4_CTL", "CD8_Naive",
                 "CD8_TEM", "CD8_TCM", "Treg", "MAIT", "gdT",
                 "NK", "NK_CD56bright",
                 "B_naive", "B_memory", "B_intermediate",
                 "CD14_Mono", "CD16_Mono",
                 "cDC2", "pDC")
```

3.2.3 Create PALMO object (Time ~ 2min)

```
#Create PALMO object
palmo_obj <- createPALMOobject(anndata=ann, data=pbmc)

#Assign Sample, PTID and Time parameters
palmo_obj <- annotateMetadata(data_object=palmo_obj,
                                sample_column= "Sample",
                                donor_column= "PTID",
                                time_column= "Time")

#Sample overlap and final matrix
```

For single cell data merge annotation and single cell metadata

```
palmo_obj <- mergePALMOdata(data_object=palmo_obj, datatype="singlecell")

#Aggregate data (Psuedo-bulk)
```

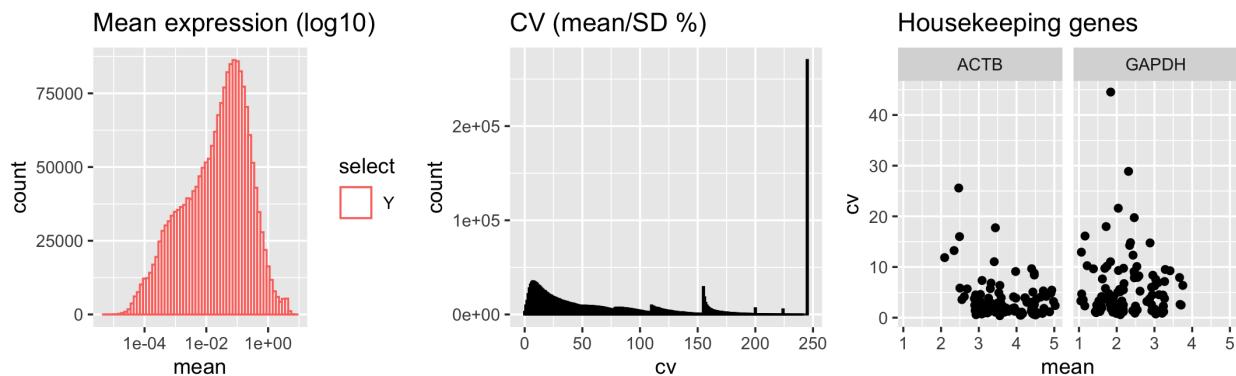
Aggregated samples noted by sample group. Define sample group and Calculate average expression. Keep genes with avgExpression > zero.

```
palmo_obj <- avgExpCalc(data_object=palmo_obj,
                          assay="RNA", group_column="celltype")
head(palmo_obj@curated[["anndata"]]) #merged annotation data
head(palmo_obj@curated[["data"]]) #scRNA average expression data

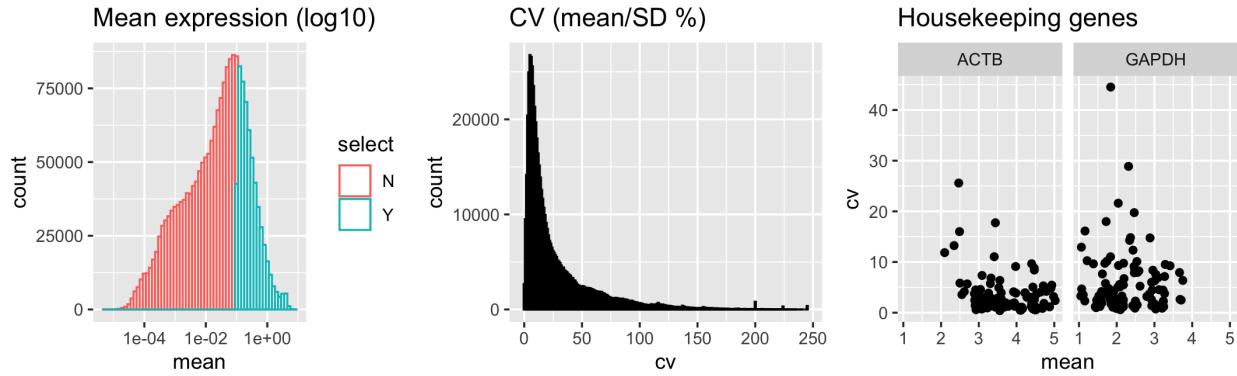
#Check for replicates
palmo_obj <- checkReplicates(data_object=palmo_obj, mergeReplicates = T)
```

3.2.4 CV profile (Time ~ 5min)

```
#Check the mean expression and CV cross groups (celltypes)
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                               housekeeping_genes=c("GAPDH", "ACTB"),
                               fileName="scrna")
```



```
#Lowly expressed genes show abnormal CV, which needs to be filtered.
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj, meanThreshold = 0.1,
                               housekeeping_genes=c("GAPDH", "ACTB"),
                               fileName="scrna")
```



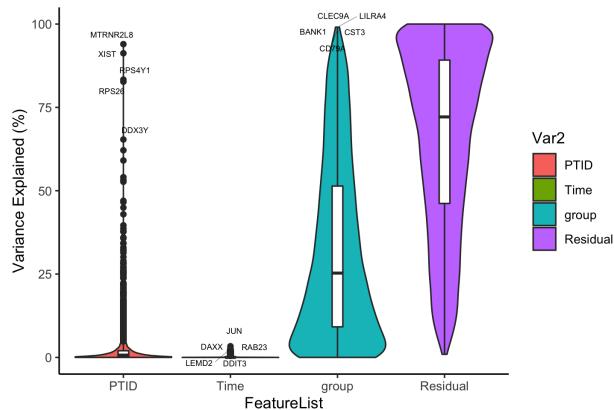
3.2.5 Donorwise CV profile over longitudinal timepoints (Time ~ 4min)

```
library("ggrepel")
#Sample Celltype Mean-CV plot (check output folder)
cvSCsampleprofile(data_object=palmo_obj, meanThreshold = 0.1,
cvThreshold = 10)
```

3.2.6 Features contributing towards donor variations (Time ~ 10min)

```
#Check the group of interest
head(palmo_obj@curated$anndata)

#Variance decomposition
featureSet <- c("PTID", "Time", "celltype")
palmo_obj <- lmeVariance(data_object=palmo_obj,
                           featureSet=featureSet,
                           meanThreshold=0.1, cl=4,
                           fileName="scrna")
var_decomp <- palmo_obj$result$variance_decomposition
```



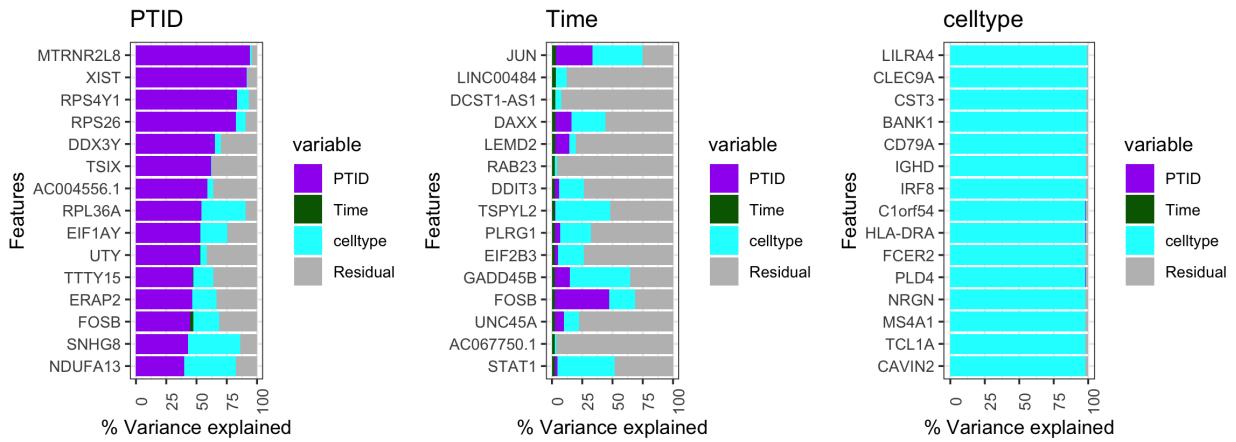
3.2.7 Variance plot

Donor-specific, Time- and celltype-attributed variance contributing features.

```

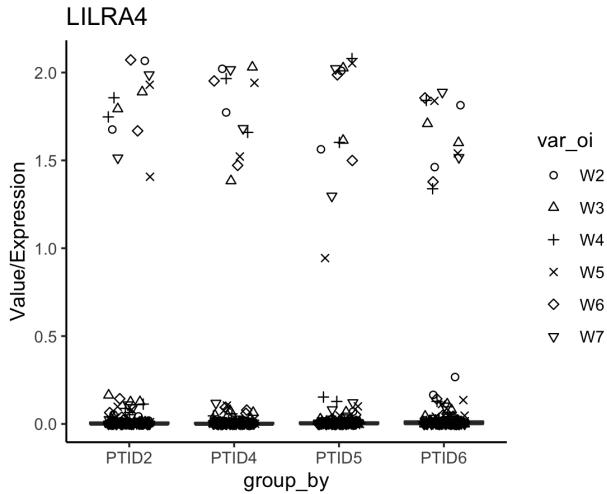
plots <- variancefeaturePlot(vardata=var_decomp, featureSet=featureSet,
                               Residual=T, cols=c("purple", "darkgreen", "cyan"))
plot_grid(plotlist = plots, ncol=3)

```

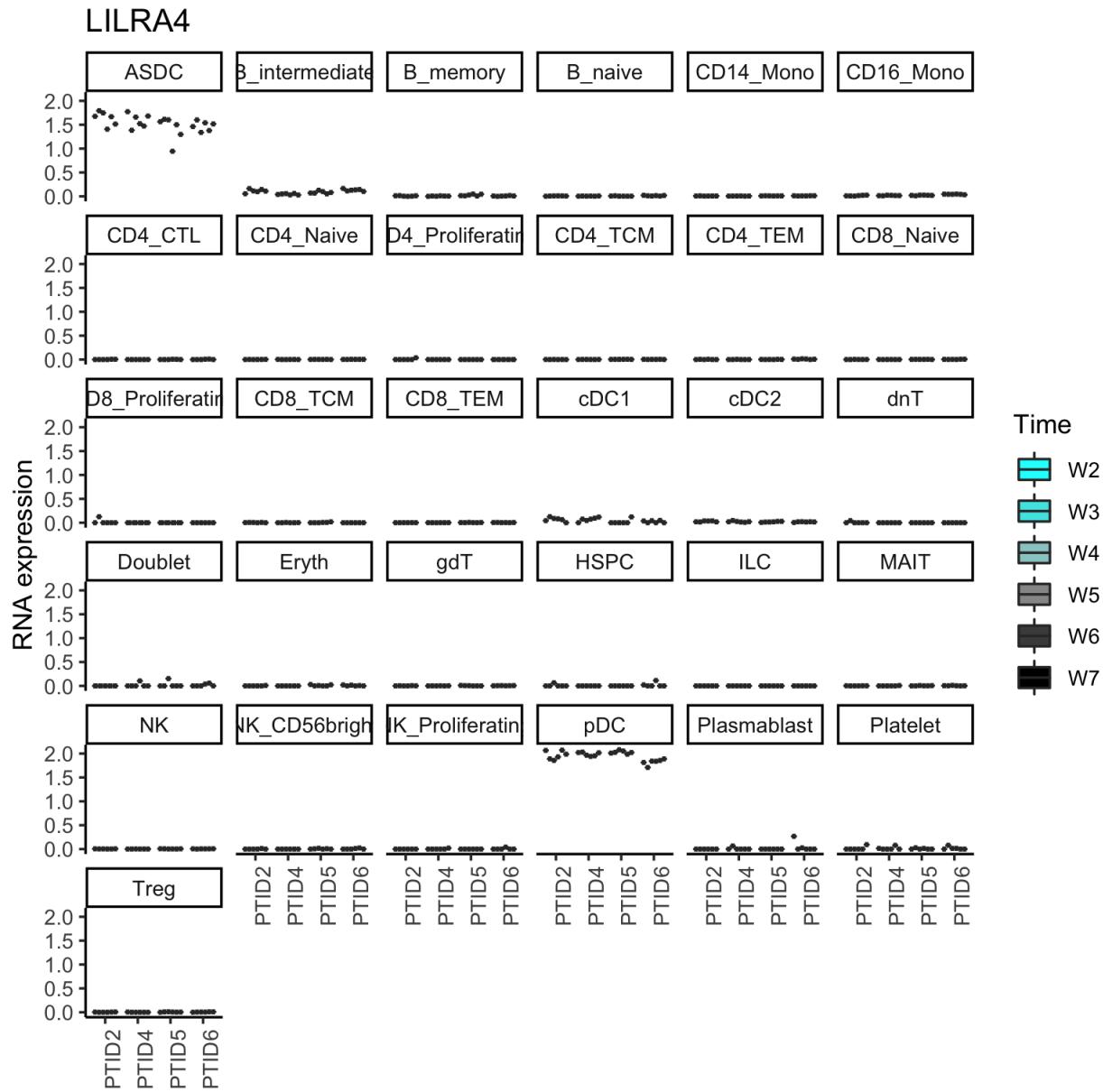


3.2.8 Plot the top features

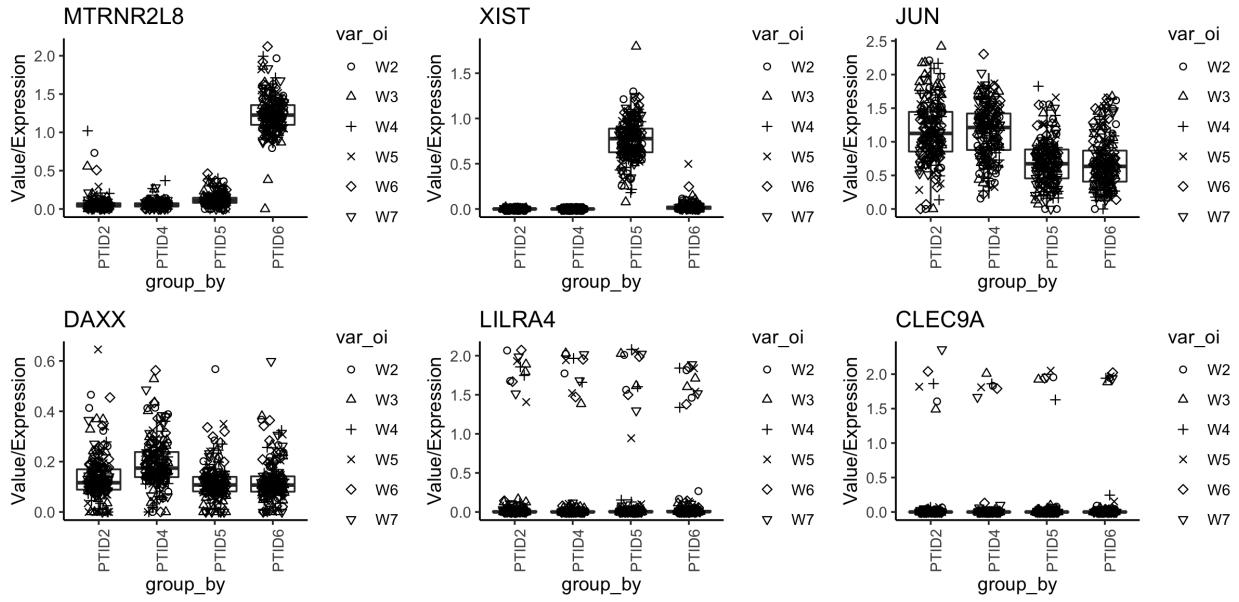
```
gene_featureplot(data_object=palmo_obj, featureList="LILRA4")
```



```
gene_featureplot(data_object=palmo_obj, featureList="LILRA4", facet_by="group")
```



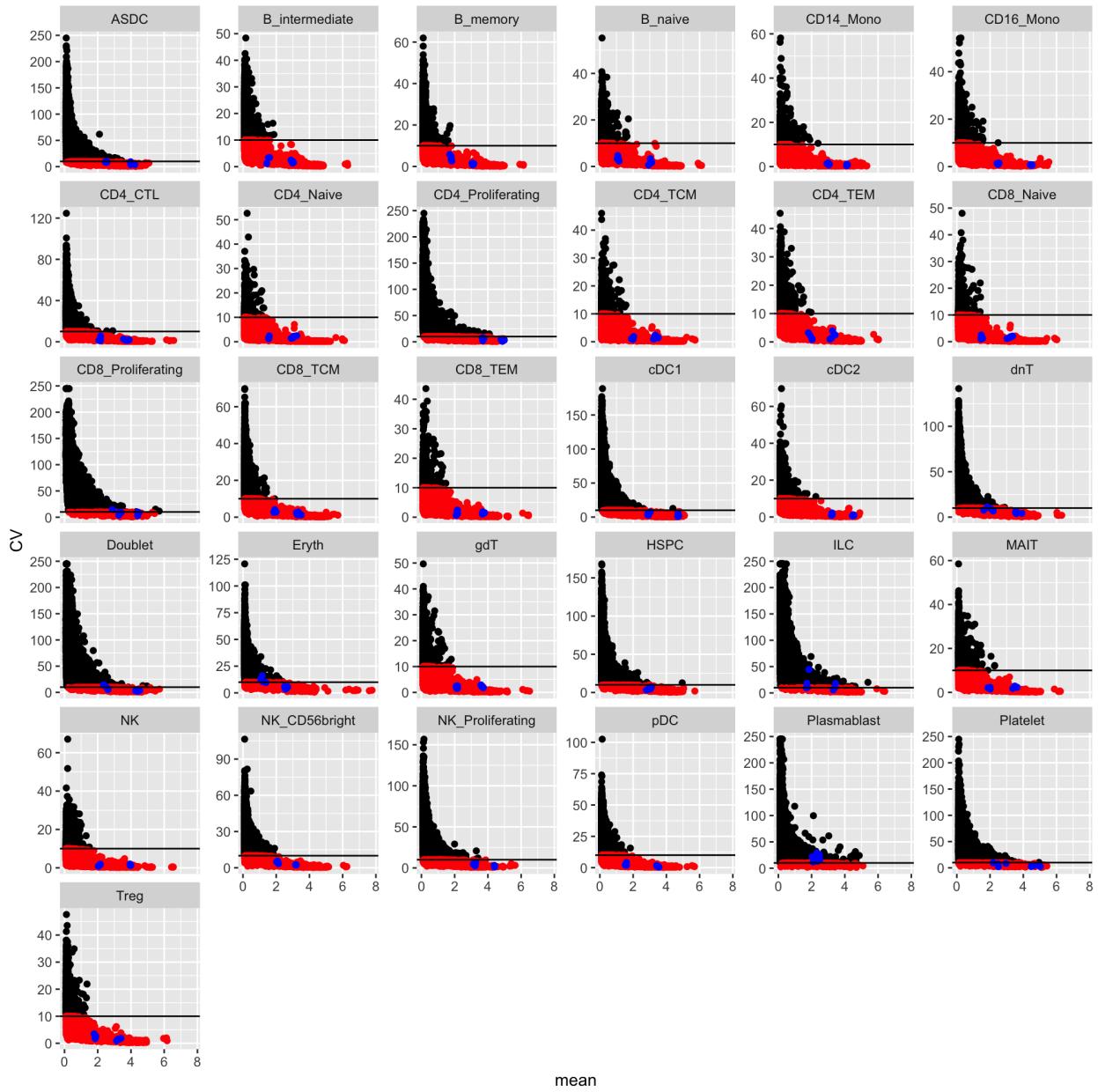
```
#Multiple-gene visualization
plots <- gene_featureplot(data_object=palmo_obj,
                            featureList=c("MTRNR2L8", "XIST",
                                         "JUN", "DAXX",
                                         "LILRA4", "CLEC9A"),
                            x_group_by="PTID", var_oi="Time", x_text_angle=90)
plot_grid(plotlist=plots, ncol= 3, align="hv")
```



3.2.9 Intra-donor variations over time (Time ~10min)

```
#Calculate CV
palmo_obj <- cvCalcSC(data_object=palmo_obj,
meanThreshold=0.1, cvThreshold=10,
housekeeping_genes=c("GAPDH", "ACTB"),
fileName="scrna")
```

Plots saved in user-defined output directory



CV profile in celltypes (black) as well as CV for house-keeping genes (blue). Based on CV of house-keeping genes 10% CV cut-off used and genes considered stable below 10% CV.

3.2.10 Find stable and variable features in longitudinal data (Time <1 min)

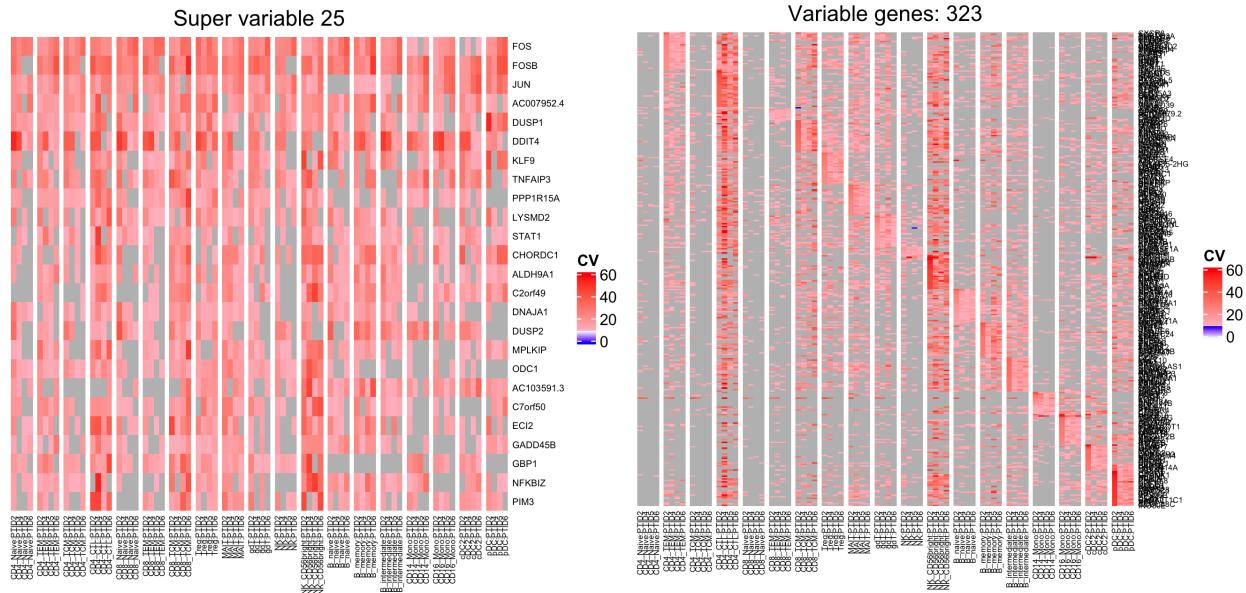
```

donorThreshold <- 4
groupThreshold <- 40 #number of donors * number of celltypes/2 (4x19/2)

palmo_obj <- VarFeatures(data_object=palmo_obj, group_oi=celltype_oi,
                           cvThreshold=10,
                           donorThreshold=4, groupThreshold=40,
                           topFeatures=25,
                           fileName="scrna")
var_genes <- palmo_obj@result$var_genes

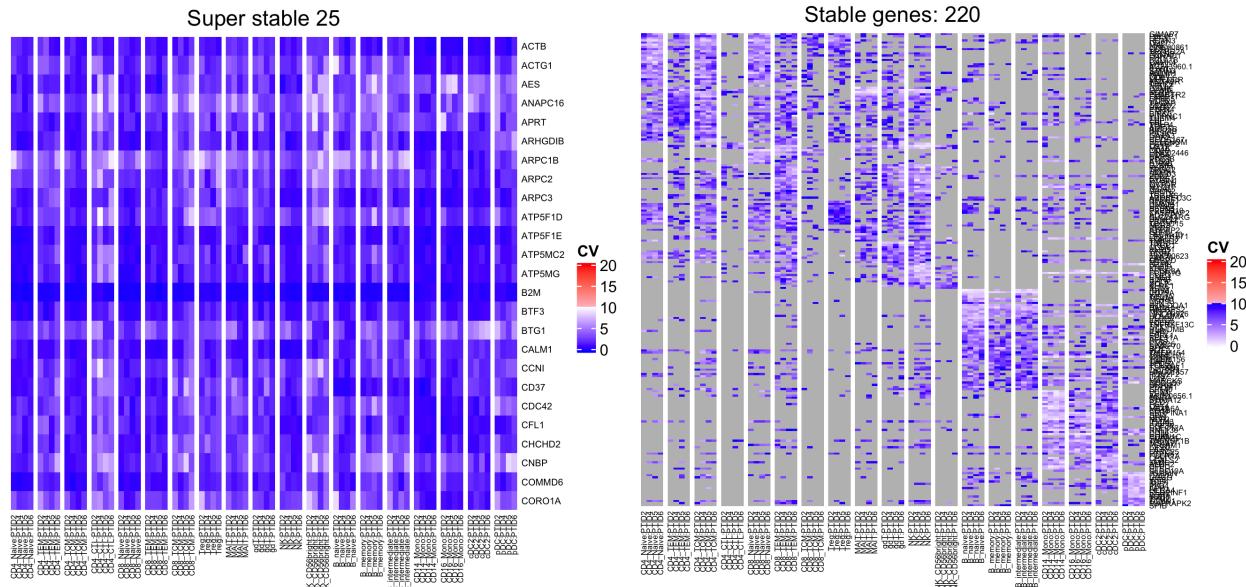
```

Variable genes observed in longitudinal data (CV>10%)



```
palmo_obj <- StableFeatures(data_object=palmo_obj, group_oi=celltype_oi,
                               cvThreshold=10,
                               donorThreshold=4, groupThreshold=40,
                               topFeatures=25,
                               fileName="scRNA")
stable_genes <- palmo_obj@result$stable_genes
```

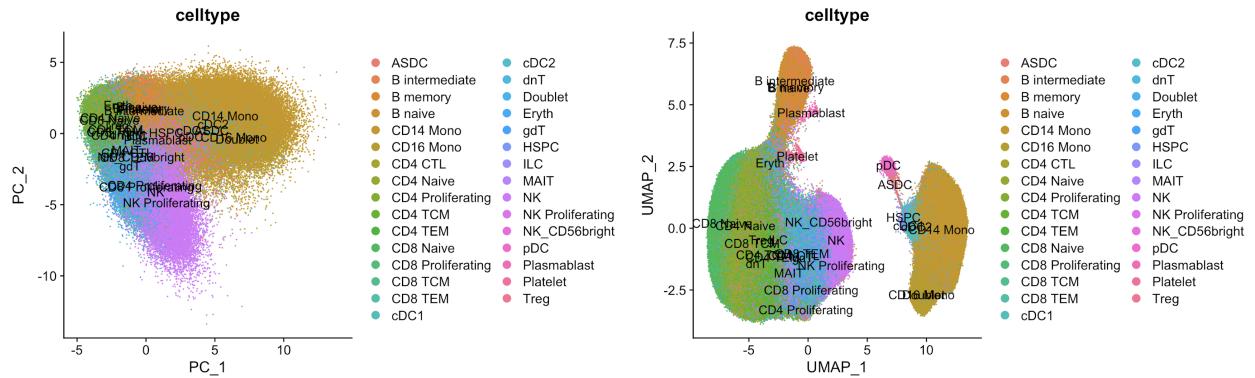
Stable genes observed in longitudinal data (CV<10%)



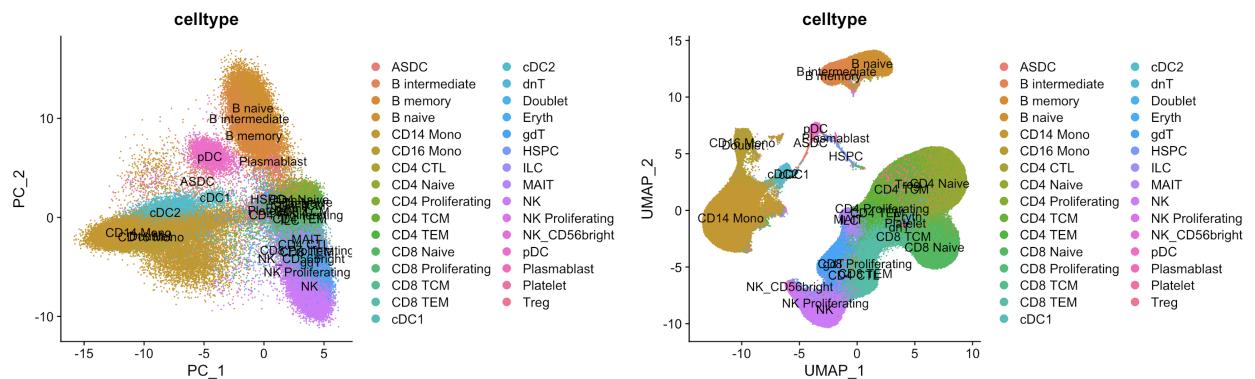
3.2.11 UMAP Plot (Stable/variable) (Time ~ 5min)

#Top variable and stable features used for UMAP

```
dimUMAPPPlot(data_object=palmo_obj, nPC=15,
             gene_oi=unique(var_genes$gene),
             group_column=avgGroup, plotname="variable",
             fileName="scrna")
```

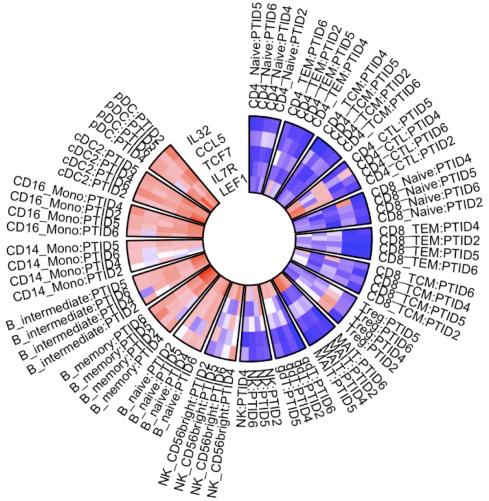


```
dimUMAPPPlot(data_object=palmo_obj, nPC=15,
             gene_oi=unique(stable_genes$gene),
             group_column=avgGroup, plotname="stable",
             fileName="scrna")
```



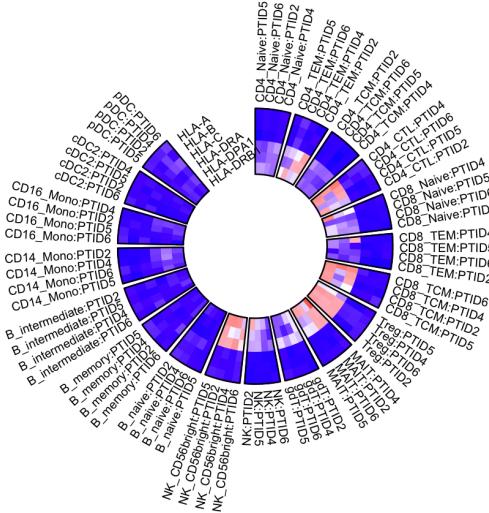
3.2.12 Circular gene expression plot

```
geneList <- c("IL32", "CCL5", "TCF7", "IL7R", "LEF1") #T-cell
plotres <- genecircosPlot(data_object=palmo_obj, geneList=geneList,
                           group_oi=celltype_oi, colorThreshold=10)
```



#Users can also load PALMO output result

```
cv_res <- palmo_obj@result[["cv_all"]]
geneList <- c("HLA-A", "HLA-B", "HLA-C", "HLA-DRA", "HLA-DPA1", "HLA-DRB1")
plotres <- genecircosPlot(data=cv_res, geneList=toList,
                           titleName="HLA", group_oi=celltype_oi,
                           colorThreshold=10)
```



3.3 Tutorial-3: scATAC Longitudinal data (n=4 and 6 weeks follow-up)

This tutorial allows users to explore single cell ATACseq genscore data measured from 4 healthy donors over 6 timepoints (week 2-7). Single cell ATAC data available at GSE190992. (1) pbmc_scatac_archr_genescore_longitudinal_data (2) AIFI-Metadata.Rda (clinical metadata). Longitudinal dataset have 4 donors and 18 samples. To infer the variations at single cell ATAC please follow following steps.

3.3.1 Load Library

```
#Load Library
library("PALMO")
library("Hmisc")
library("ggpubr")
library("cowplot")
```

3.3.2 Load data and assign paramaters (Time <30sec)

```
#scATAC object
```

Load genescorematrix from archR or relevant tools (Aggregate data at celltypes (pseudo-bulk))

```
load("data/AIFI-scATAC-PBMC-FinalData.Rda")
datamatrix <- log2(scatac_gm+1)

#Load annotation data
load("data/AIFI-Metadata.Rda")
```

3.3.3 Create PALMO object (Time <30sec)

```
#Create PALMO object
palmo_obj <- createPALMOobject(anndata=ann, data=datamatrix)

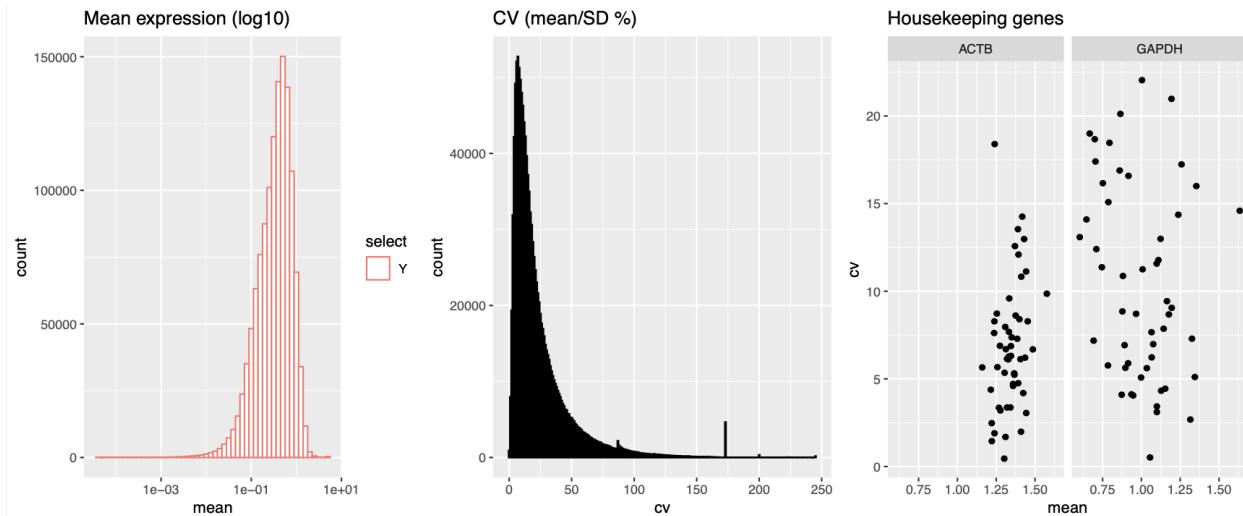
#Assign Sample, PTID and Time parameters
palmo_obj <- annotateMetadata(data_object=palmo_obj,
                                sample_column= "Sample", donor_column= "PTID",
                                time_column= "Time")

#Sample overlap and final matrix
palmo_obj <- mergePALMOdata(data_object=palmo_obj, datatype="singlecell")

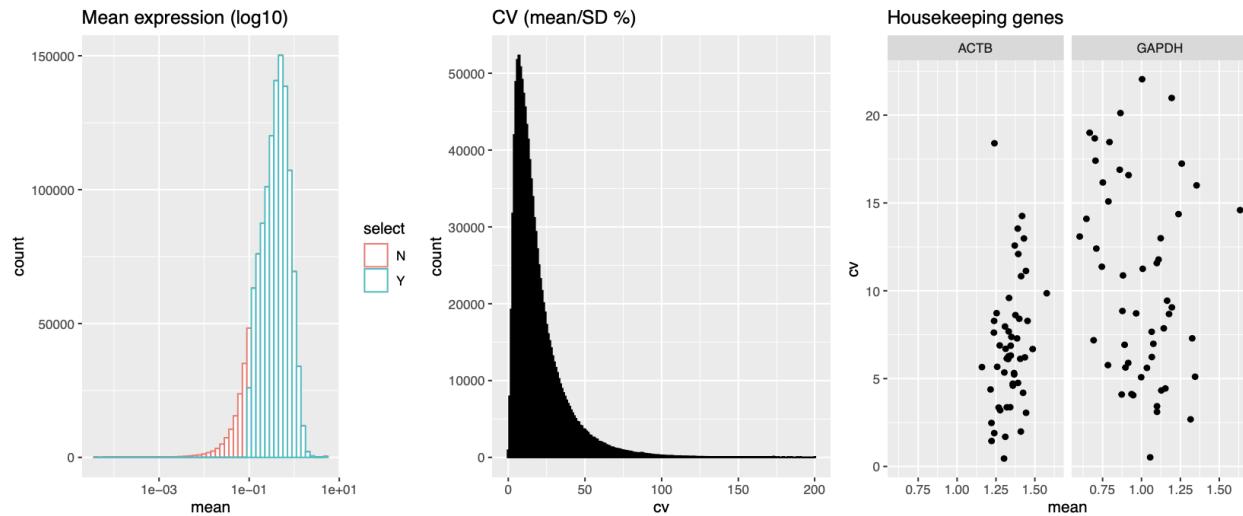
#Check for replicates
palmo_obj <- checkReplicates(data_object=palmo_obj, mergeReplicates = T)
```

3.3.4 CV profile (Time ~2min)

```
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                                 housekeeping_genes=c("GAPDH", "ACTB"),
                                 fileName="scatac")
```



```
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                                housekeeping_genes=c("GAPDH", "ACTB"),
                                meanThreshold = 0.1,
                                fileName="scatac")
```



```
#Sample Celltype Mean-CV plot
cvSCsampleprofile(data_object=palmo_obj, meanThreshold = 0.1,
                   cvThreshold = 10, fileName="scatac")
#plots saved in output directory
```

3.3.5 Features contributing towards donor variations (Time ~ 8min)

```
#Check the group of interest
head(palmo_obj@curated$anndata)

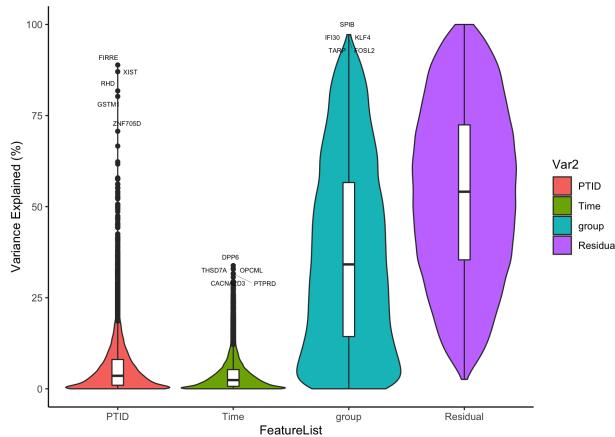
#Variance decomposition
featureSet <- c("PTID", "Time", "group")
palmo_obj <- lmeVariance(data_object=palmo_obj,
```

```

        featureSet=featureSet,
        meanThreshold=0.1, cl=4,
        fileName="scatac")

var_decomp <- palmo_obj@result$variance_decomposition
head(var_decomp[,featureSet])
#          PTID      Time      group
#FIRRE     88.88239 0.2606612 2.8339971
#XIST      87.05064 0.3986387 0.5178851
#RHD       81.77230 0.3641998 1.4574140
#gSTM1    80.29766 0.1786628 6.2277645
#ZNF705D   70.69163 0.4342325 0.4752960
#LOC105376805 66.62977 1.2741294 9.8918653

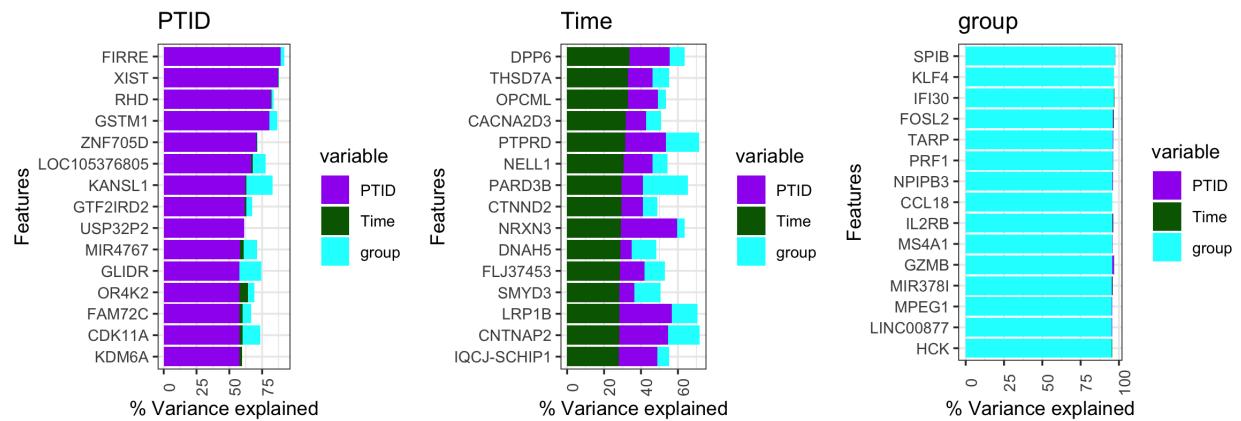
```



```

#Variance contributing features
plots <- variancefeaturePlot(vardata=var_decomp, featureSet=featureSet,
                               cols=c("purple", "darkgreen", "cyan"))
plot_grid(plotlist = plots, ncol=3)

```



```

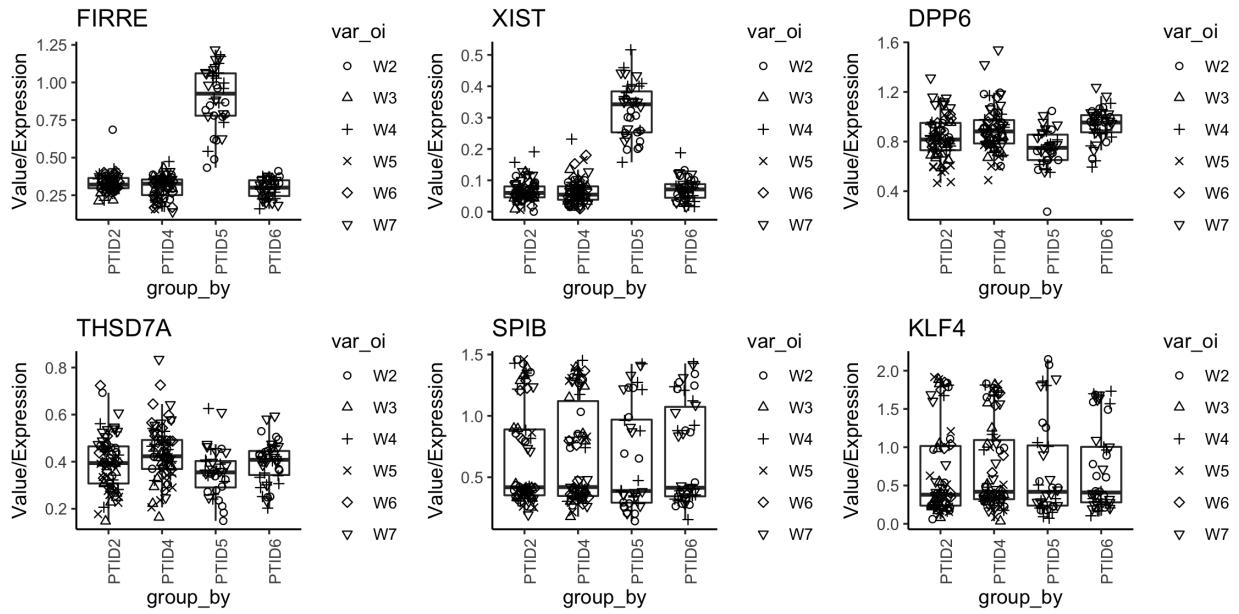
#Top genes
plots <- gene_featureplot(data_object=palmo_obj,
                            featureList=c("FIRRE", "XIST",
                                         "DPP6", "THSD7A",
                                         "SPIB", "KLF4"),
                            x_group_by="PTID", var_oi="Time",

```

```

x_text_angle=90)
plot_grid(plotlist=plots, ncol= 3, align="hv")

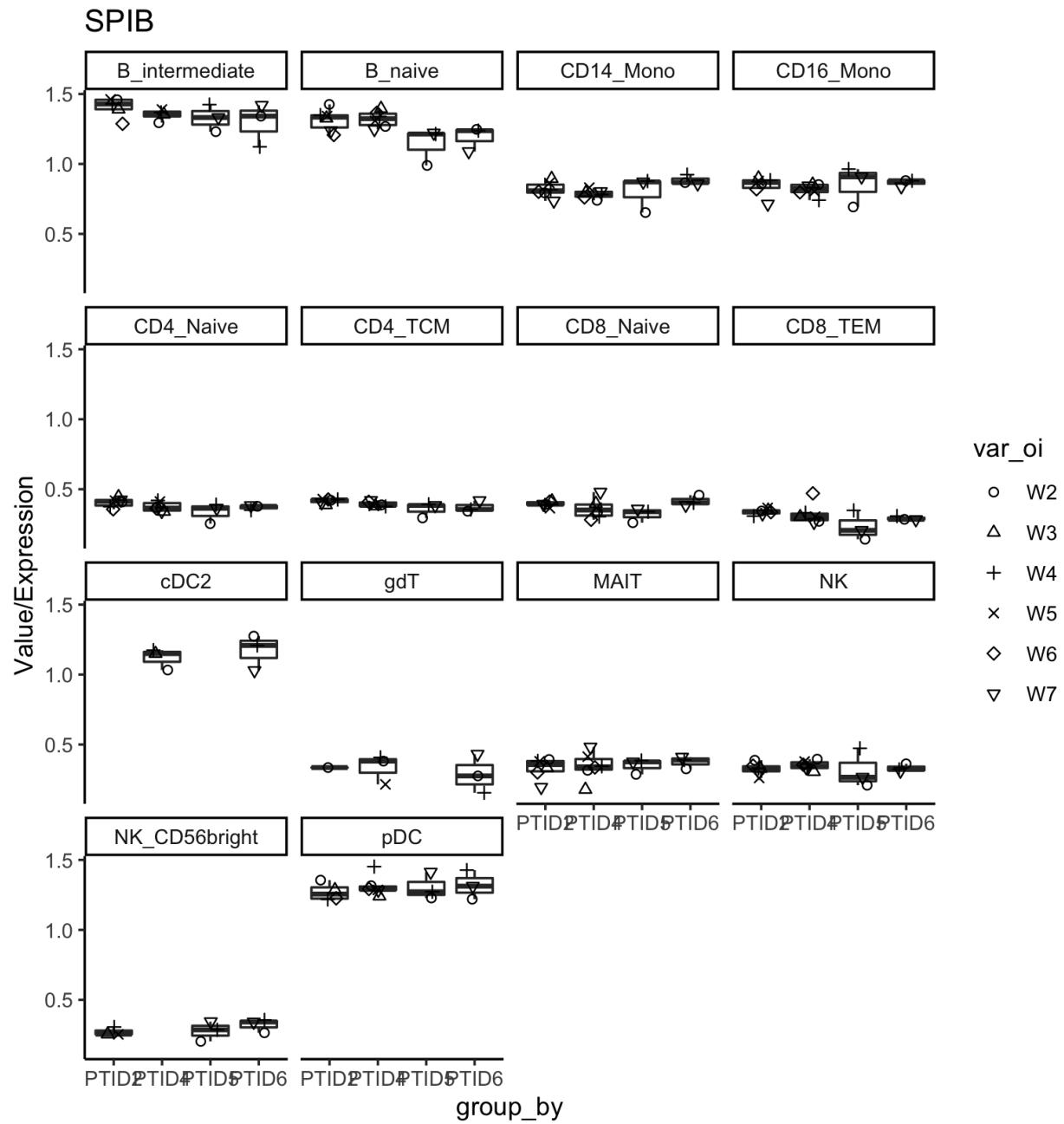
```



```

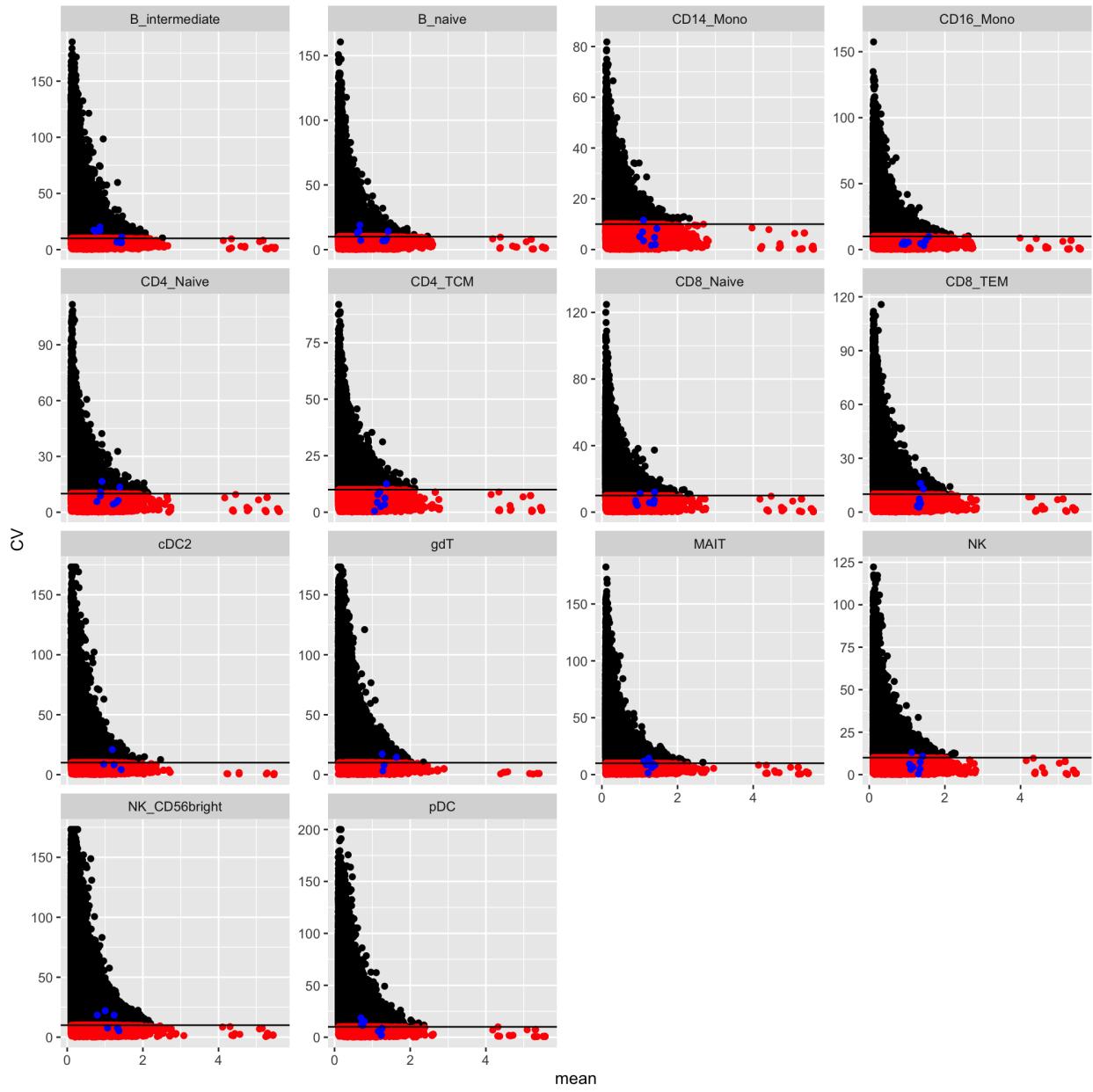
gene_featureplot(data_object=palmo_obj, featureList="SPIB", facet_by="group")

```



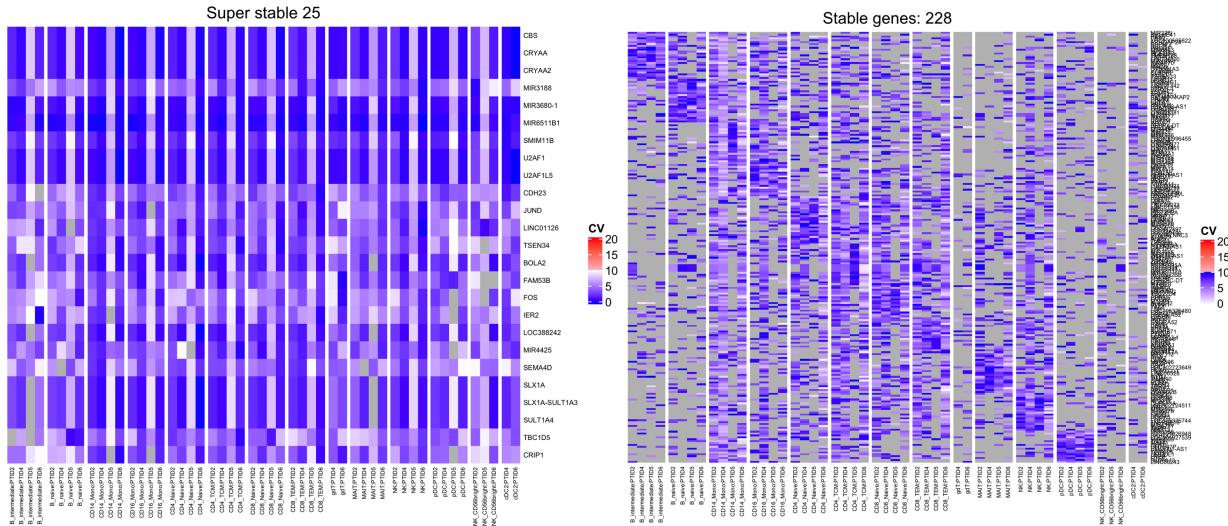
3.3.6 Intra-donor variations over time (Time ~ 5min)

```
palmo_obj <- cvCalcSC(data_object=palmo_obj,
                        meanThreshold=0.1, cvThreshold=10,
                        housekeeping_genes=c("GAPDH", "ACTB"),
                        fileName="scatac")
```



3.3.7 Find stable and variable features in longitudinal data (Time 30sec)

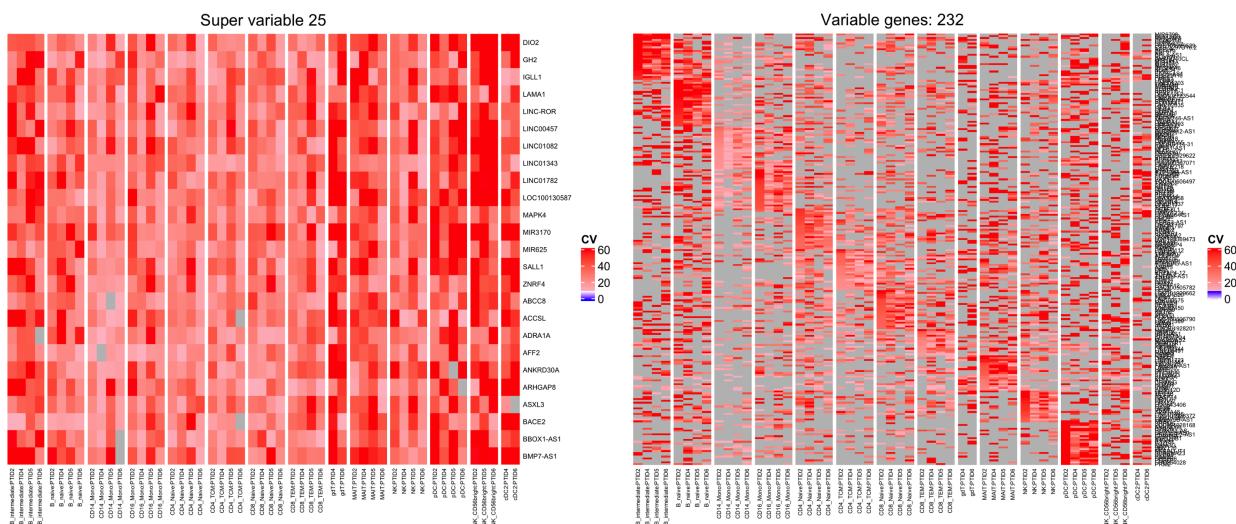
```
#Find stable and variable features in longitudinal data
donorThreshold <- 4
groupThreshold <- 28 #number of donors * number of celltypes/2 (4x14/2)
topFeatures <- 25
palmo_obj <- StableFeatures(data_object=palmo_obj,
                               cvThreshold=10,
                               donorThreshold=4, groupThreshold=28,
                               topFeatures=25,
                               fileName="scatac")
stable_gene <- palmo_obj@result$stable_genes
```



```

palmo_obj <- VarFeatures(data_object=palmo_obj,
                           cvThreshold=10,
                           donorThreshold=4, groupThreshold=28,
                           topFeatures=25,
                           fileName="scatac")
var_gene <- palmo_obj@result$var_genes

```



3.3.8 Circos CV plot (Time ~ 10sec)

```

geneList <- c("HLA-A", "HLA-B", "HLA-C", "HLA-DRA", "HLA-DPA1", "HLA-DRB1",
            "ACTB", "GAPDH")
plotmatrix <- genecircosPlot(data_object=palmo_obj, geneList=geneList,
                               colorThreshold=15)

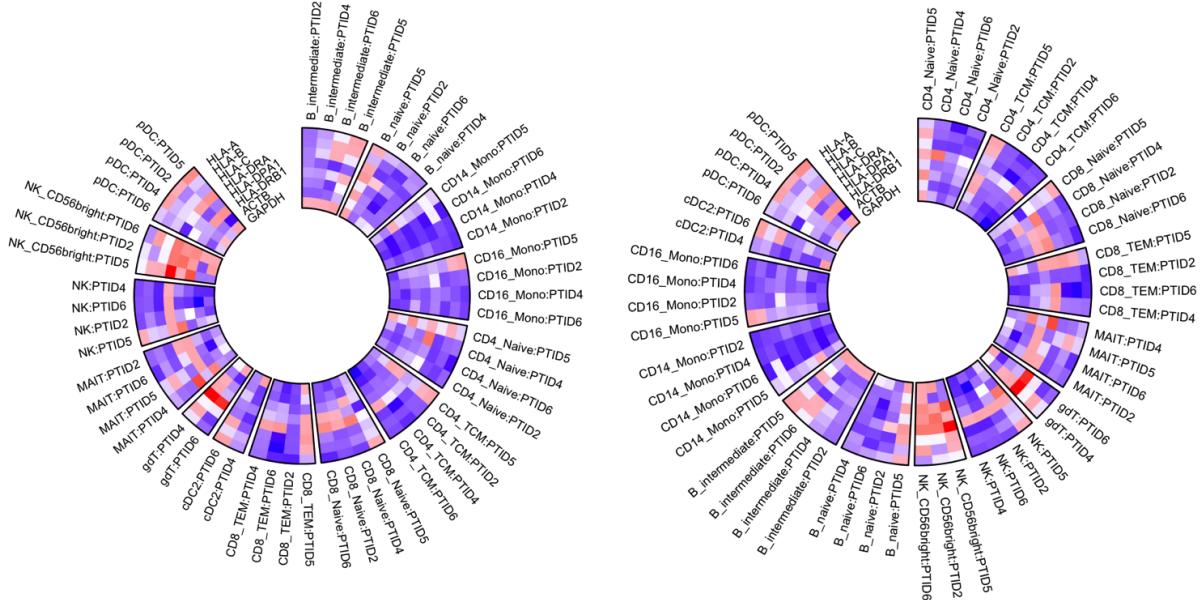
#order by user-defined group order
celltype_oi <- c("CD4_Naive", "CD4_TEM", "CD4_TCM", "CD4_CTL",
                  "CD8_Naive", "CD8_TEM", "CD8_TCM", "Treg", "MAIT", "gdT",

```

```

    "NK", "NK_CD56bright",
    "B_naive", "B_memory", "B_intermediate",
    "CD14_Mono", "CD16_Mono",
    "cDC2", "pDC")
plotmatrix <- genecircosPlot(data_object=palmo_obj, geneList=geneList,
                               group_oi=celltype_oi, colorThreshold=15)

```



3.4 Tutorial-4: Multi-modal data integration

This tutorial allows users to combine intra-donor variation value between different modalities like scRNA and scATAC data here. Load CV result from scRNA and scATAC as described above (check output directory for the files). To integrate variability across modalities, please follow following steps.

3.4.1 Load Library

```
#Load Library
library("PALMO")
library("Hmisc")
library("ComplexHeatmap")
library("circlize")
```

3.4.2 Load data

```
#From scRNA analysis obtain the CV data
load("data/result/scrna-CV-allgenes-raw.Rda")
scrna_cv_res <- cv_res

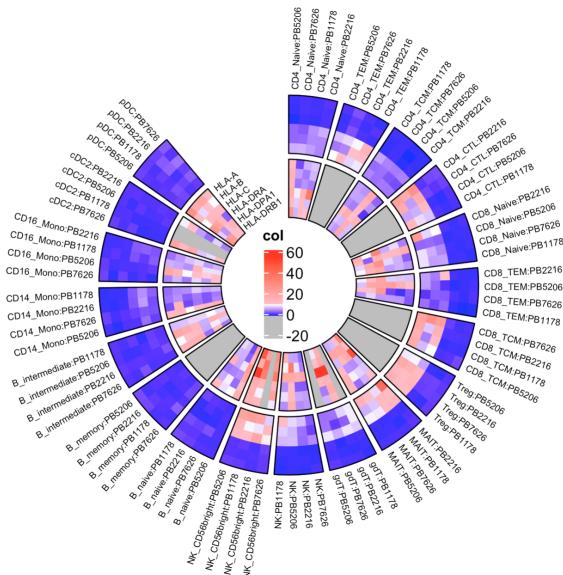
#From scATAC analysis obtain the CV data
load("data/result/scatac-CV-allgenes-raw.Rda")
scatac_cv_res <- cv_res

#Cell type of interest
celltype_oi <- c("CD4_Naive", "CD4_TEM", "CD4_TCM", "CD4_CTL",
                 "CD8_Naive", "CD8_TEM", "CD8_TCM", "Treg",
                 "MAIT", "gdT", "NK", "NK_CD56bright",
                 "B_naive", "B_memory", "B_intermediate",
                 "CD14_Mono", "CD16_Mono",
                 "cDC2", "pDC")

#HLAs
geneList <- c("HLA-A", "HLA-B", "HLA-C", "HLA-DRA", "HLA-DPA1", "HLA-DRB1")
```

3.4.3 Run (Time ~ 10sec)

```
plot <- multimodalView(modality1=scrna_cv_res,
                         modality2=scatac_cv_res,
                         geneList=geneList, group_oi=celltype_oi)
```



3.5 Tutorial-5: COVID19 longitudinal dataset (CNP0001102)

This tutorial allows users to explore single cell RNAseq data variability across COVID and FLU donors. PBMC from the patients were collected longitudinally. Single cell data from Zhu et al. 2020 downloaded from here. Metadata is downloaded from table and can be found in the data. To infer variability (inter- and Intra-) and identify stable genes, please follow following steps.

3.5.1 Load Library

```
#Load Library
library("PALMO")
library("Hmisc")
library("ggpubr")
library("cowplot")
```

3.5.2 Load data and assign paramaters (Time < 1min)

```
#Load scRNA data
pbmc <- readRDS("data/CNP0001102_Final_nCoV_0716_upload.RDS")
#Add column Sample
pbmc@meta.data$Sample <- pbmc@meta.data$batch
#check celltypes
sort(unique(pbmc@meta.data$cell_type))
#[1] Cytotoxic CD8 T cells Naive T cells           NKs
#[4] MAIT          Activated CD4 T cells Naive B cells
#[7] Plasma        Memory B cells                XCL+ NKs
#[10] Cycling T cells Monocytes                 DCs
#[13] Cycling Plasma Stem cells                 Megakaryocytes

#Clinical annotations Table S1. Clinical data of the enrolled subjects
metadata <- read.csv("data/CNP0001102-annotation.csv", stringsAsFactors = F)

#Exploring only COVID samples
metadata <- metadata[metadata$Participant %in% c("COV-1", "COV-2", "COV-3", "COV-4", "COV-5"),]
#Exploring only FLU samples
#metadata <- metadata[metadata$Participant %in% c("IAV-1", "IAV-2"),]
```

3.5.3 Create PALMO object (Time < 1min)

```
#Create PALMO object
palmo_obj <- createPALMOobject(anndata=metadata, data=pbmc)

#Assign Sample, PTID and Time parameters
palmo_obj <- annotateMetadata(data_object=palmo_obj,
                                sample_column= "Sample",
                                donor_column= "Participant",
                                time_column= "Day")

#Sample overlap and final matrix
palmo_obj <- mergePALMOdata(data_object=palmo_obj, datatype="singlecell")

#Aggregate data (Psuedo-bulk)
```

```

palmo_obj <- avgExpCalc(data_object=palmo_obj, assay="RNA",
                           group_column="cell_type")
head(palmo_obj@curated[["anndata"]]) #merged annotation data
head(palmo_obj@curated[["data"]]) #scRNA average expression data

#Check for replicates
palmo_obj <- checkReplicates(data_object=palmo_obj, mergeReplicates = T)

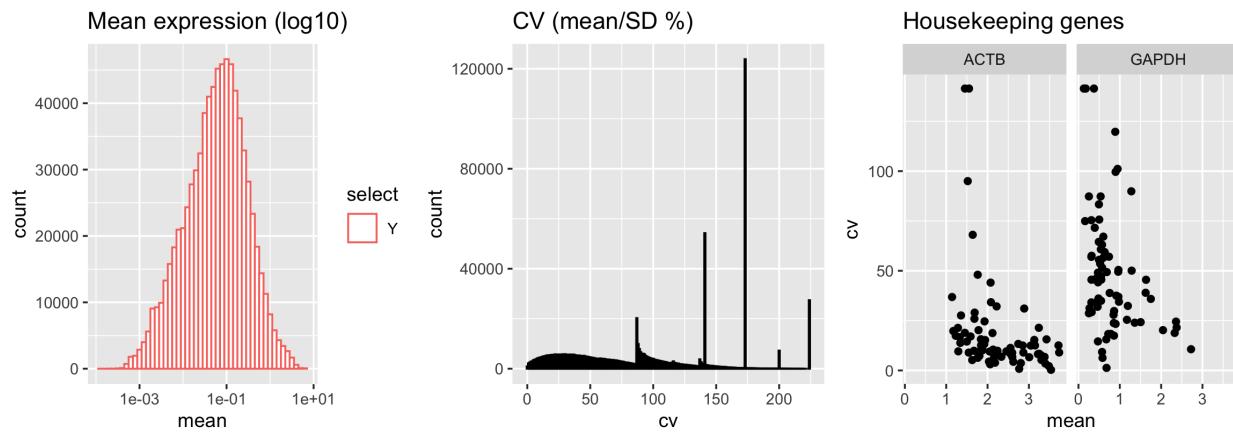
```

3.5.4 CV profile (Time ~ 1min)

```

palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                                housekeeping_genes=c("GAPDH", "ACTB"),
                                fileName="CNP0001102")

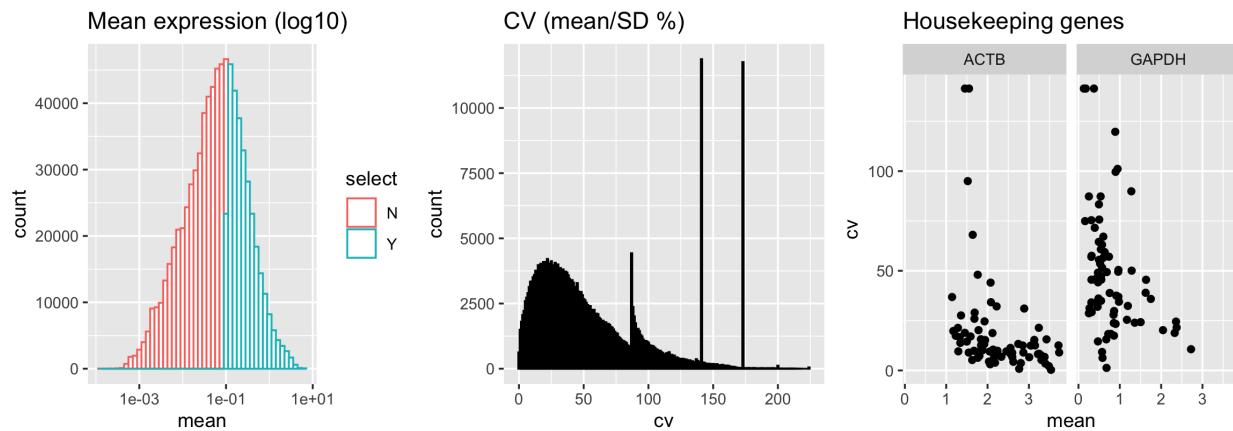
```



```

#Sample Celltype Mean-CV plot (output directory)
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                                housekeeping_genes=c("GAPDH", "ACTB"),
                                meanThreshold = 0.1,
                                fileName="CNP0001102")

```



```

cvSCsampleprofile(data_object=palmo_obj,
                   meanThreshold = 0.1, plot_log10=T,
                   cvThreshold = 25)

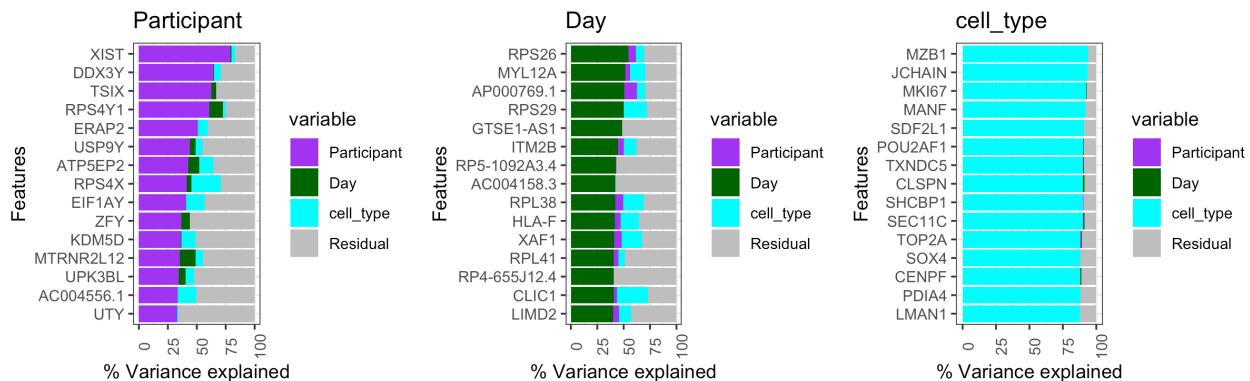
```

3.5.5 Features contributing towards donor variations (Time ~ 5min)

```
#Check the group of interest
head(palmo_obj@curated$anndata)

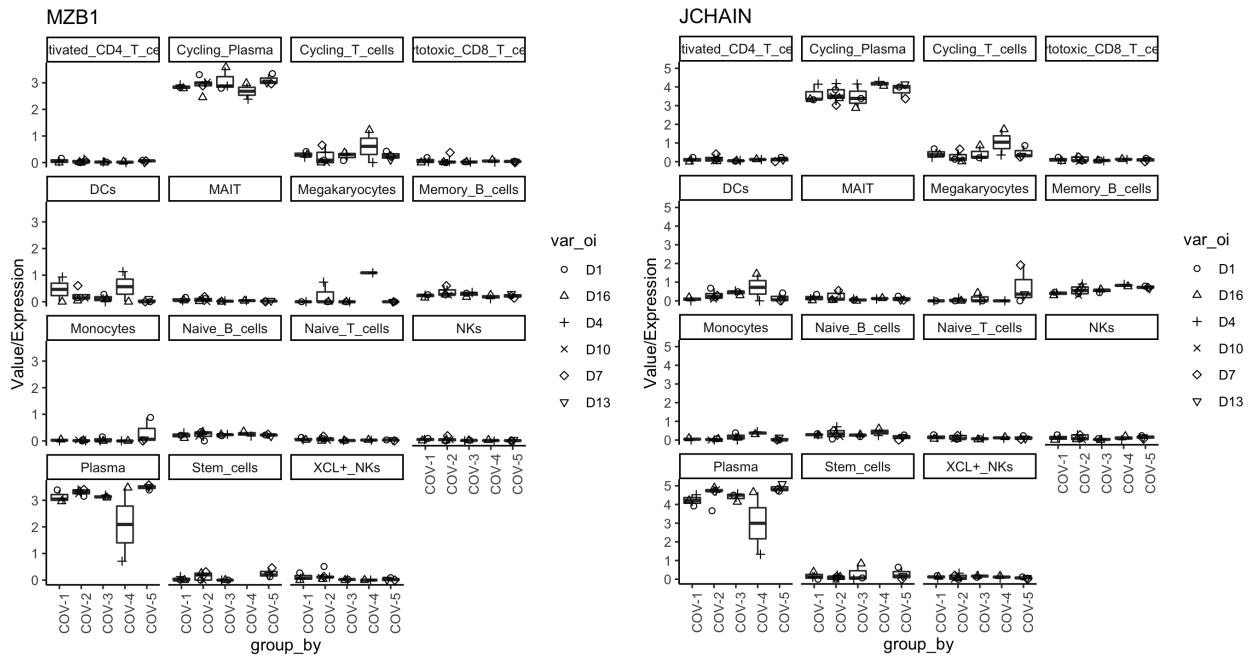
#Variance decomposition
featureSet <- c("Participant", "Day", "cell_type")
palmo_obj <- lmeVariance(data_object=palmo_obj,
                           featureSet=featureSet,
                           meanThreshold=0.1, cl=4,
                           fileName="CNP0001102")
var_decomp <- palmo_obj@result$variance_decomposition
head(var_decomp[,featureSet])
  Participant      Day cell_type
#XIST      78.82243  0.936695  3.510383
#DDX3Y     64.44897  0.637655  5.824030
#TSIX      62.57711  3.997646  0.339527
#RPS4Y1    60.73146  11.768212 2.346928
#ERAP2      50.92503  0.000000  8.745452
#USP9Y     44.21867  4.657177  5.997529

#Variance explained (Donor, Time, and celltype)
plots <- variancefeaturePlot(vardata=var_decomp, featureSet=featureSet, Residual=T, cols=c("purple", "darkgreen", "cyan", "gray"))
plot_grid(plotlist = plots, ncol=3)
```



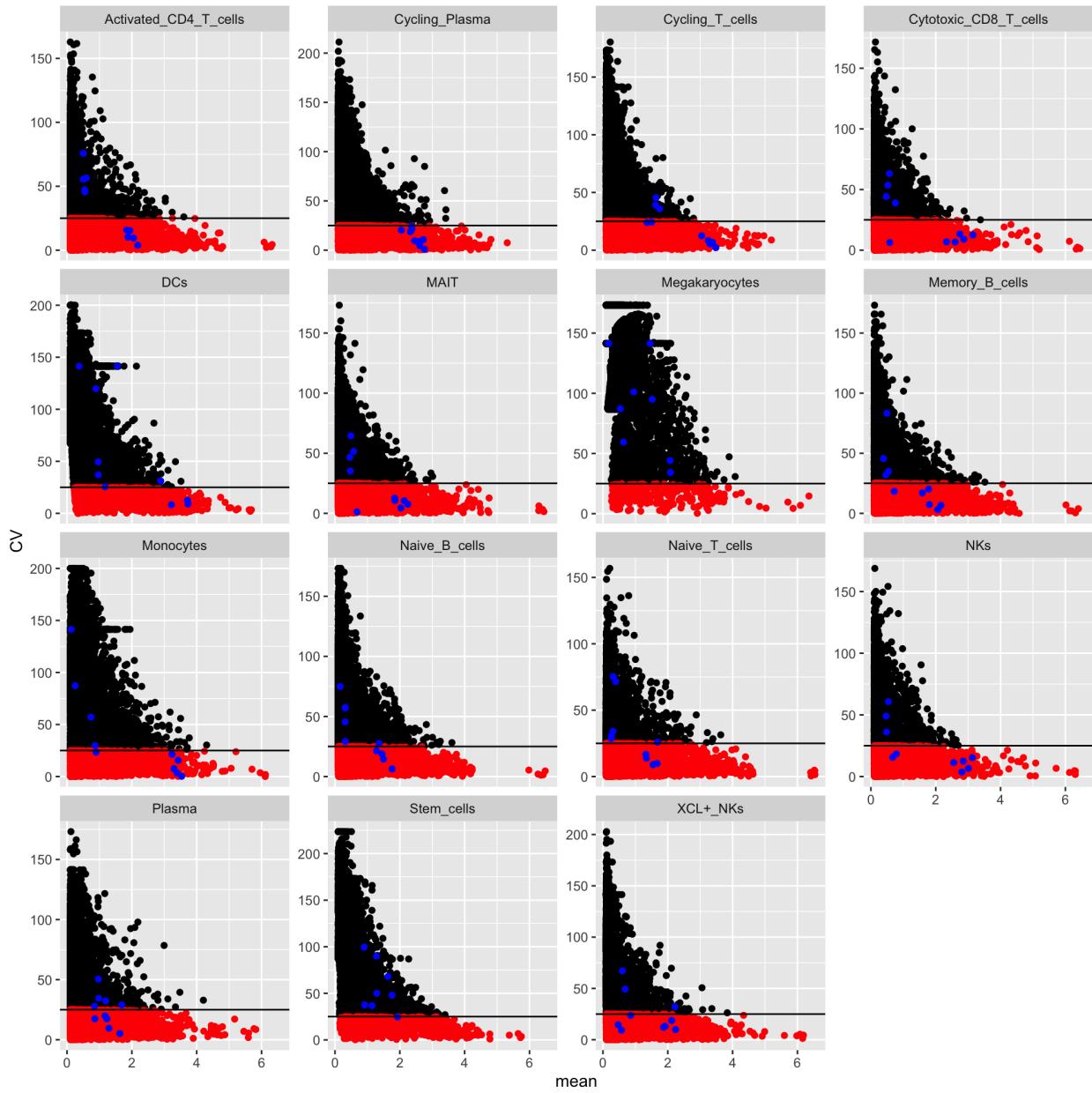
3.5.6 Plot the variables (Time ~ 10sec)

```
plots <- gene_featureplot(data_object=palmo_obj,
                           featureList=c("MZB1", "JCHAIN"),
                           facet_by="cell_type", x_text_angle=90)
```



3.5.7 Intra-donor variations over time (Time ~ 4min)

```
#Calculate CV
palmo_obj <- cvCalcSC(data_object=palmo_obj,
                        meanThreshold=0.1, cvThreshold=25,
                        housekeeping_genes=c("GAPDH", "ACTB"),
                        fileName="CNP0001102")
```

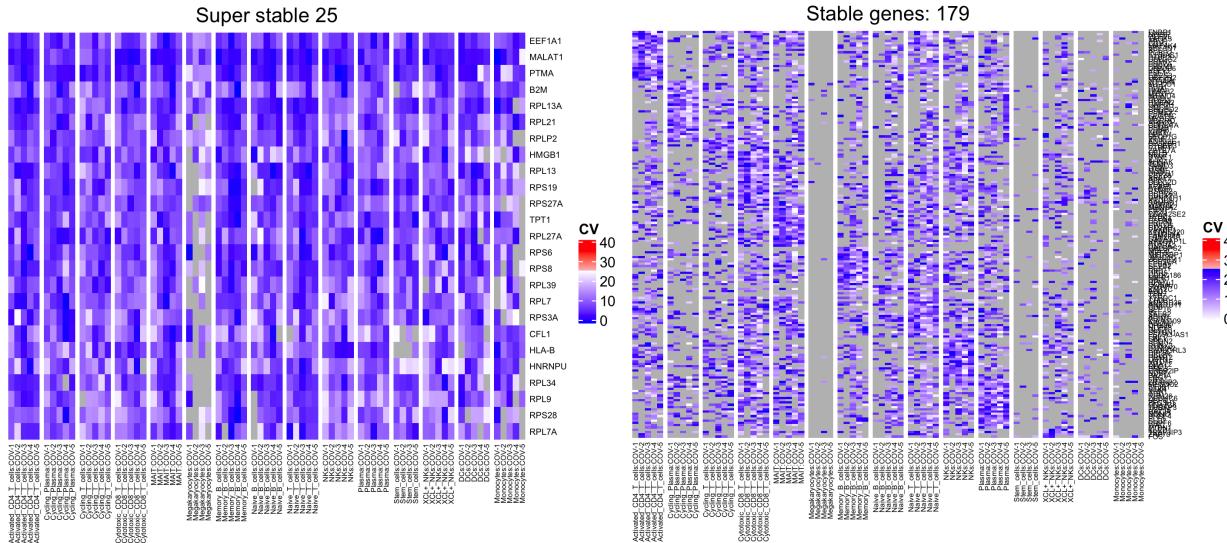


3.5.8 Find stable and variable features in longitudinal data (Time ~ 10sec)

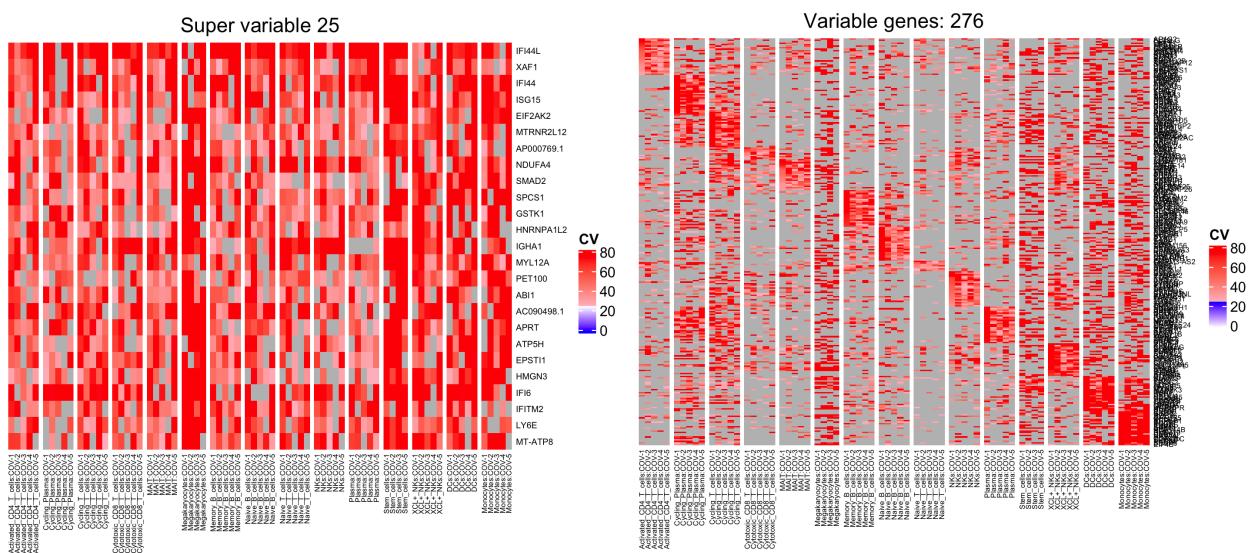
```

donorThreshold <- 5 #number of donors
groupThreshold <- 38 #number of donors * number of celltypes/2 (5x15/2)
topFeatures <- 25
palmo_obj <- StableFeatures(data_object=palmo_obj,
                               cvThreshold=25,
                               donorThreshold=5, groupThreshold=38,
                               topFeatures=25,
                               fileName="CNP0001102")
stable_genes <- palmo_obj@result$stable_genes

```

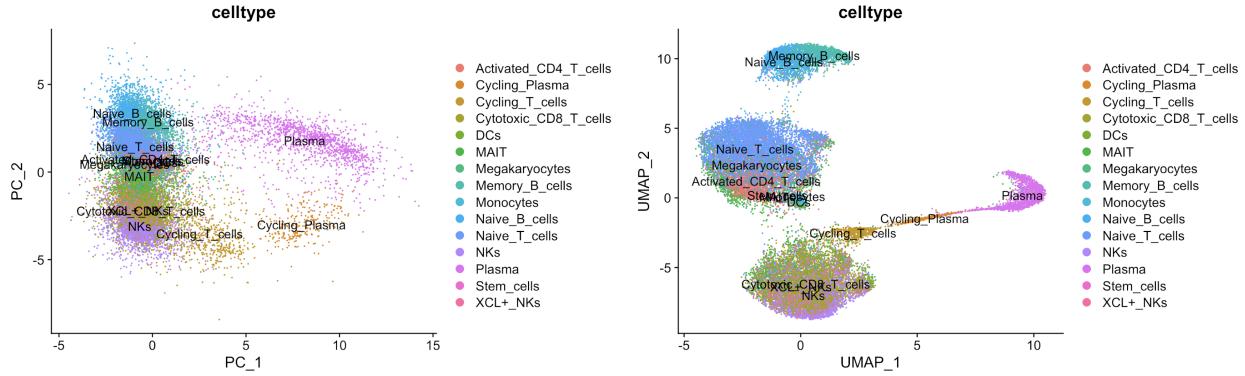


```
palmo_obj <- VarFeatures(data_object=palmo_obj,
                           cvThreshold=25,
                           donorThreshold=5, groupThreshold=38,
                           topFeatures=25,
                           fileName="CNP0001102")
var_genes <- palmo_obj$result$var_genes
```

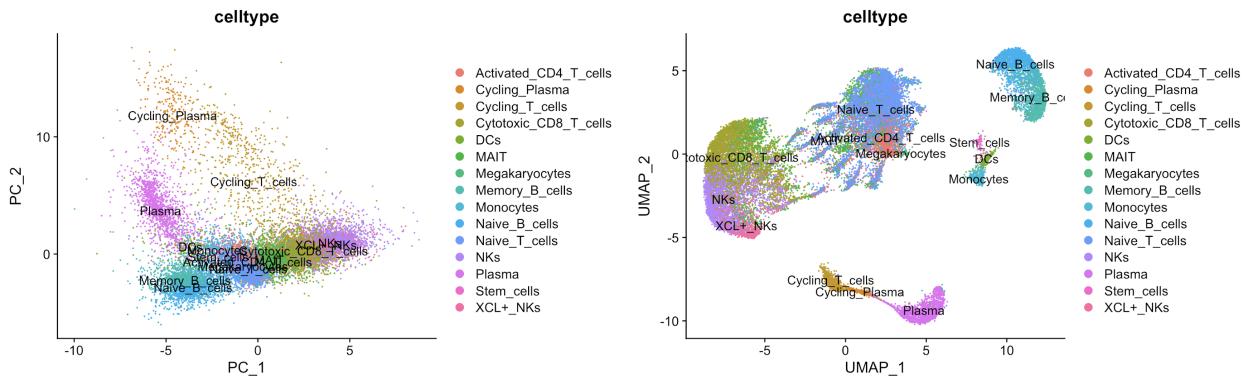


3.5.9 UMAP Plot (Time ~ 2min)

```
#Stable genes UMAP
dimUMAPPlot(data_object=palmo_obj, nPC=15,
             gene_oi=unique(stable_genes$gene),
             group_column="cell_type", plotname="stable",
             fileName="CNP0001102")
```



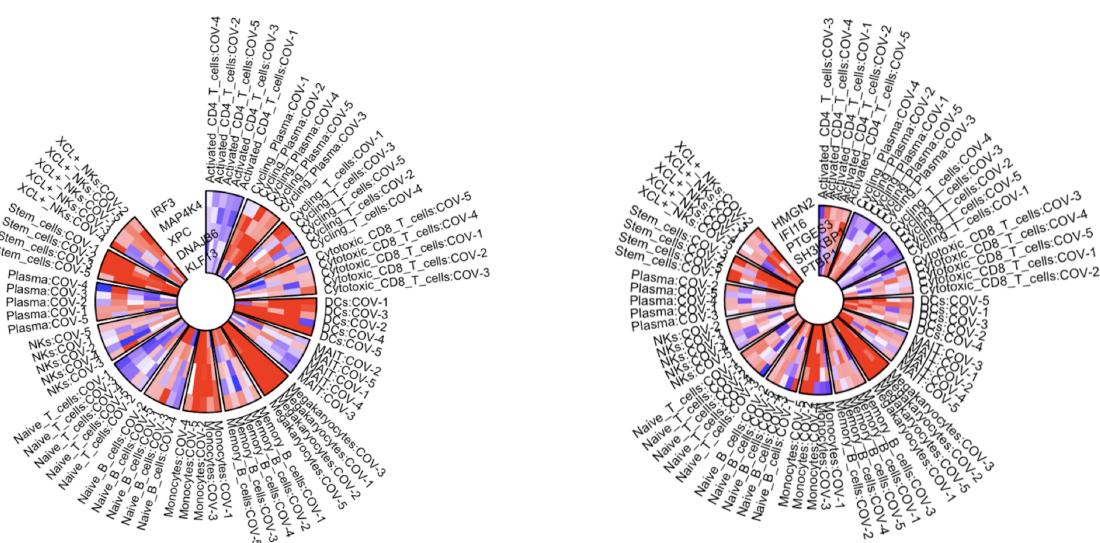
```
#Variable genes UMAP
dimUMAPPlot(data_object=palmo_obj, nPC=15,
             gene_oi=unique(var_genes$gene),
             group_column="cell_type", plotname="variable",
             fileName="CNP0001102")
```



3.5.10 Celltype-specific Circos CV Plot (Time ~ 30sec)

```
#Activated CD4 T-cells
geneList <- c("IRF3", "MAP4K4", "XPC", "DNAJB6", "KLF13")
plotres <- genecircosPlot(data_object=palmo_obj,
                           geneList=toList(geneList), colorThreshold=25)

#Cycling T-cells
geneList <- c("HMGN2", "IFI16", "PTGES3", "SH3KBP1", "PTBP1")
plotres <- genecircosPlot(data_object=palmo_obj,
                           geneList=toList(geneList), colorThreshold=25)
```



3.6 Tutorial-6: Differential Gene analysis in longitudinal data (CNP0001102)

This tutorial allows users to identify differential expressed genes in direction of time-points. As an example single cell data from Zhu et al. 2020 downloaded from here. Metadata is downloaded from table and can be found in the data. The dataset consists of 5 Covid-19 donors, 2 Flu donors with longitudinal data and 3 controls. To explore differential expressed genes in each celltype of each donor we used hurdle model based modeling on input data to retrieve the DEGs. To infer DEGs in each celltype towards time progression (timepoints considered as continuous if more than 2), please follow following steps.

3.6.1 Load data and clinical metadata (Time ~ 30sec)

Single cell object CNP0001102

```
#Single cell object CNP0001102
pbmc <- readRDS("data/CNP0001102_Final_nCoV_0716_upload.RDS")
#Add column Sample
pbmc@meta.data$Sample <- pbmc@meta.data$batch
```

3.6.2 Clinical annotations Table S1. Clinical data of the enrolled subjects (Time ~ 5sec)

```
metadata <- read.csv("data/CNP0001102-annotation.csv", stringsAsFactors = F)
row.names(metadata) <- metadata$Sample
```

3.6.3 Load library and run (Time ~15 min)

```
#Load Library
library("PALMO")
library("tidyverse")

#Create PALMO object
palmo_obj <- createPALMOobject(anndata=metadata, data=pbmc)

#Assign Sample, PTID and Time parameters
palmo_obj <- annotateMetadata(data_object=palmo_obj,
                                sample_column= "Sample",
                                donor_column= "Participant",
                                time_column= "Day")

#Sample overlap and final matrix
palmo_obj <- mergePALMOdata(data_object=palmo_obj, datatype="singlecell")

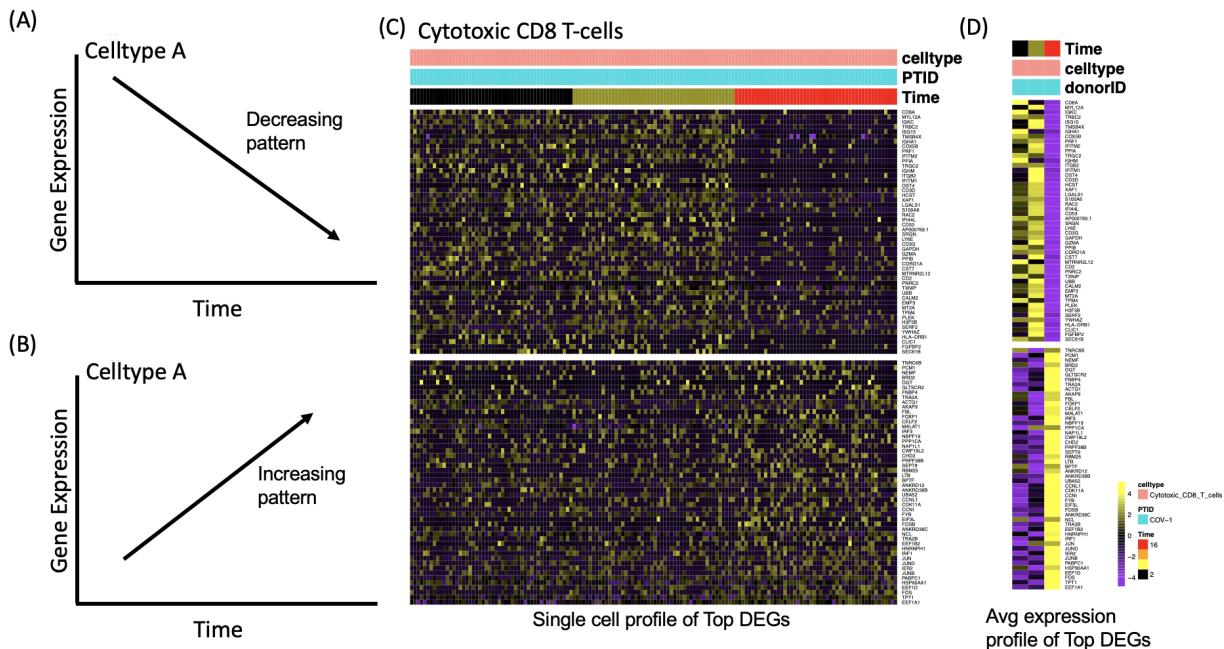
#Perform longitudinal differential analysis
palmo_obj <- sclongitudinalDEG(data_object=palmo_obj, scassay="RNA",
                                  group_column="cell_type")
>Fitting a ZLM model for donorID: COV-1 ...
>Fitting a ZLM model for donorID: COV-2 ...
>Fitting a ZLM model for donorID: COV-3 ...
>Fitting a ZLM model for donorID: COV-4 ...
>Fitting a ZLM model for donorID: COV-5 ...
>Fitting a ZLM model for donorID: IAV-1 ...
>Fitting a ZLM model for donorID: IAV-2 ...
```

Check output folder for tabular results and visualization.

```
#Plots can be seen in output directory output
DEGres <- palmo_obj$result$degs
head(DEGres[order(DEGres$coef, decreasing = T),])
#primerid contrast      nomP      coef      adjP donorID celltype dir
#IGHG4    TimeD9 1.701579e-26 3.056092 1.453999e-23   IAV-2   Plasma upregulated at D9
#JCHAIN   TimeD9 8.759407e-32 2.647757 2.245474e-28   IAV-2   Plasma upregulated at D9
#IGHG3    TimeD9 8.470810e-22 2.485250 2.412769e-19   IAV-2   Plasma upregulated at D9
#IGLC2    TimeD9 1.352490e-16 2.289544 8.890022e-15   IAV-2   Plasma upregulated at D9
#IGHG1    TimeD9 1.292885e-16 2.219146 8.608601e-15   IAV-2   Plasma upregulated at D9
#SYNE2    TimeD4 7.249010e-13 2.209541 1.215659e-09   COV-4 XCL+_NKs upregulated at D4

#Or interested celltypes
celltype_oi <- c("Activated CD4 T cells", "Naive B cells")
palmo_obj <- sclongitudinalDEG(data_object=palmo_obj, scassay="RNA",
                                 group_column="cell_type",
                                 group_oi = celltype_oi)
```

General analysis schema and differential results in each donor over timepoints in celltype Cytotoxic CD8 T-cells using PALM shown below.



3.7 Tutorial-7: Mouse brain dataset (GSE129788)

This tutorial allows users to explore single cell RNAseq data from Mouse brain to show the application of PALMO on tissue samples. Ximerakis et al (2019) study was used out to explore the transcriptomic difference in aging brain. In this case study we used PALMO to identify stable features associated with brain celltypes across aging brain samples. The dataset includes total of 16 mice brains samples (8 young and 8 old) with 37,089 single cells GSE129788. To infer variability (inter- and Intra-) and identify stable genes, please follow following steps.

3.7.1 Load Library

```
#Load Library
library("PALMO")
library("Hmisc")
library("ggpubr")
library("cowplot")
```

3.7.2 Load data and assign paramaters (Time ~ 30sec)

```
#Load scRNA data
mbrain <- readRDS("data/GSE129788_seurat.RDS")
metaDF <- mbrain@meta.data
#check celltypes
sort(unique(mbrain@meta.data$cluster))
#[1] "ABC"      "ARP"       "ASC"       "CPC"       "DC"        "EC"        "EPC"
#[8] "Hb_VC"    "HypEPC"   "ImmN"     "MAC"       "MG"        "MNC"      "mNEUR"
#[15] "NendC"    "NEUT"     "NRP"      "NSC"       "OEG"      "OLG"      "OPC"
#[22] "PC"        "TNC"      "VLMC"     "VSMC"

#Clinical annotations Table S1. Clinical data of the enrolled subjects
metadata <- read.csv("data/GSE129788-annotation.csv", stringsAsFactors = F)
```

3.7.3 Create PALMO object (Time ~ 1min)

```
#Create PALMO object
palmo_obj <- createPALMOobject(anndata=metadata, data=mbrain)

#Assign Sample, PTID and Time parameters
palmo_obj <- annotateMetadata(data_object=palmo_obj,
                                sample_column= "Sample",
                                donor_column= "Age_group",
                                time_column= "Subject_id")

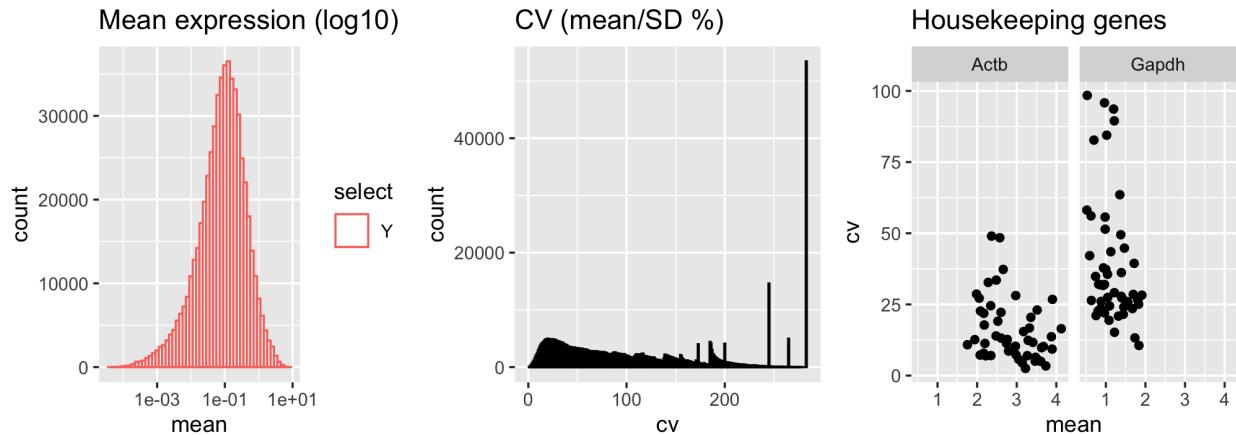
#Sample overlap and final matrix
palmo_obj <- mergePALMOdata(data_object=palmo_obj, datatype="singlecell")

#Aggregate data (Psuedo-bulk)
avgGroup <- "cluster"
palmo_obj <- avgExpCalc(data_object=palmo_obj, assay="RNA",
                         group_column="cluster")
head(palmo_obj@curated[["anndata"]]) #merged annotation data
head(palmo_obj@curated[["data"]]) #scRNA average expression data
```

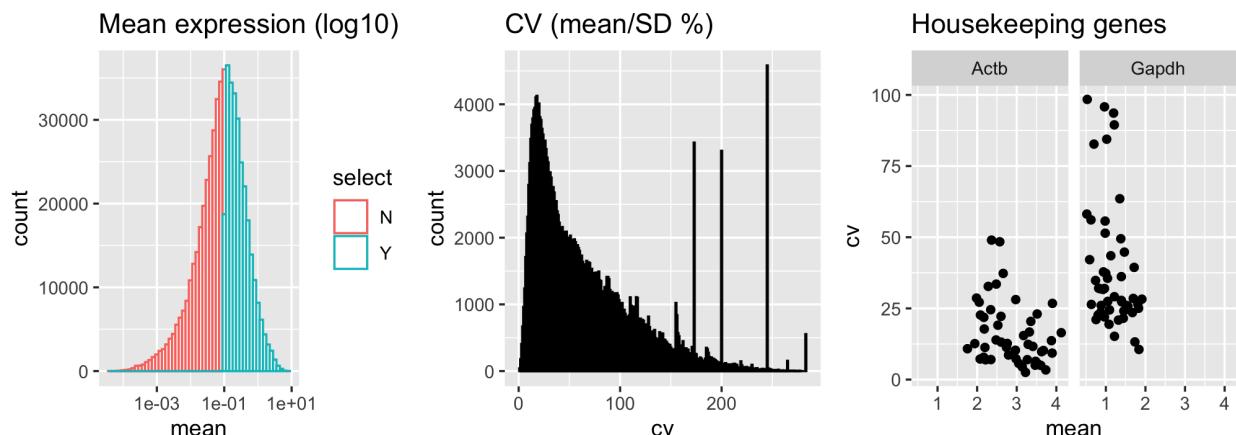
```
#Check for replicates
palmo_obj <- checkReplicates(data_object=palmo_obj, mergeReplicates = T)
```

3.7.4 CV profile (Time ~ 2min)

```
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                                housekeeping_genes=c("Gapdh", "Actb"),
                                fileName="GSE129788")
```



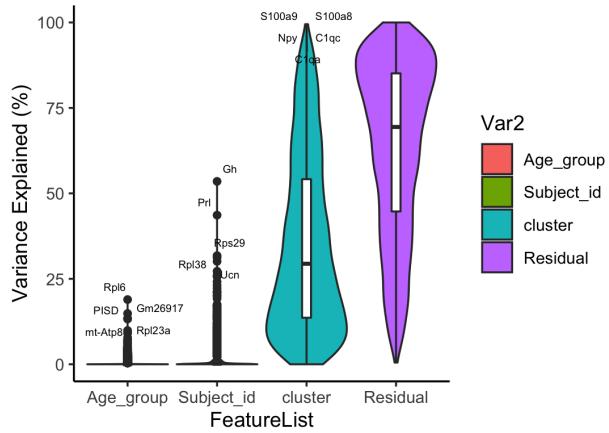
```
#Sample Celltype Mean-CV plot (output directory)
palmo_obj <- cvCalcSCPProfile(data_object=palmo_obj,
                                housekeeping_genes=c("Gapdh", "Actb"),
                                meanThreshold = 0.1,
                                fileName="GSE129788")
```



3.7.5 Features contributing towards donor variations (Time ~ 5min)

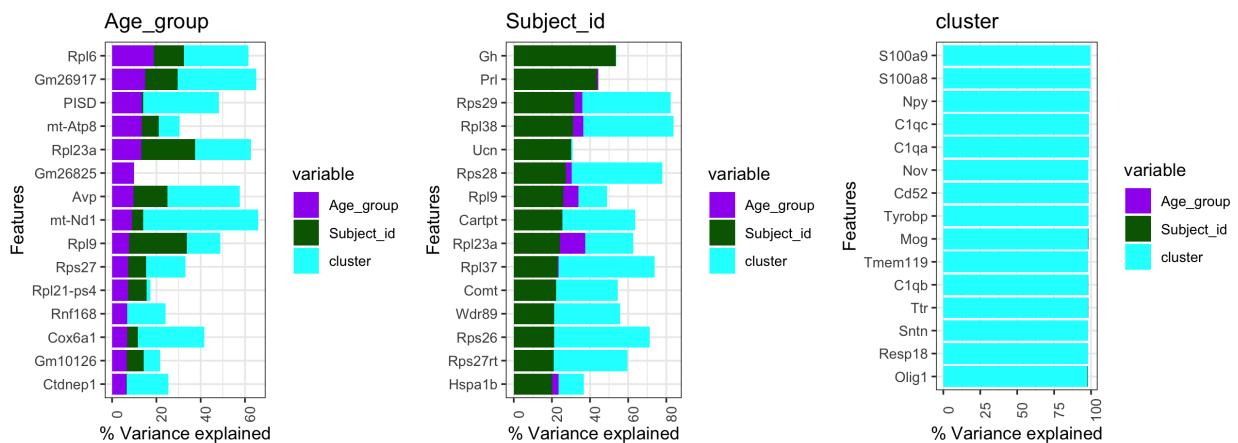
```
#Check the group of interest
head(palmo_obj@curated$anndata)
```

```
#Variance decomposition
featureSet <- c("Age_group", "Subject_id", "cluster")
palmo_obj <- lmeVariance(data_object=palmo_obj,
                           featureSet=featureSet,
                           meanThreshold=0.1, cl=4,
                           fileName="GSE129788")
```



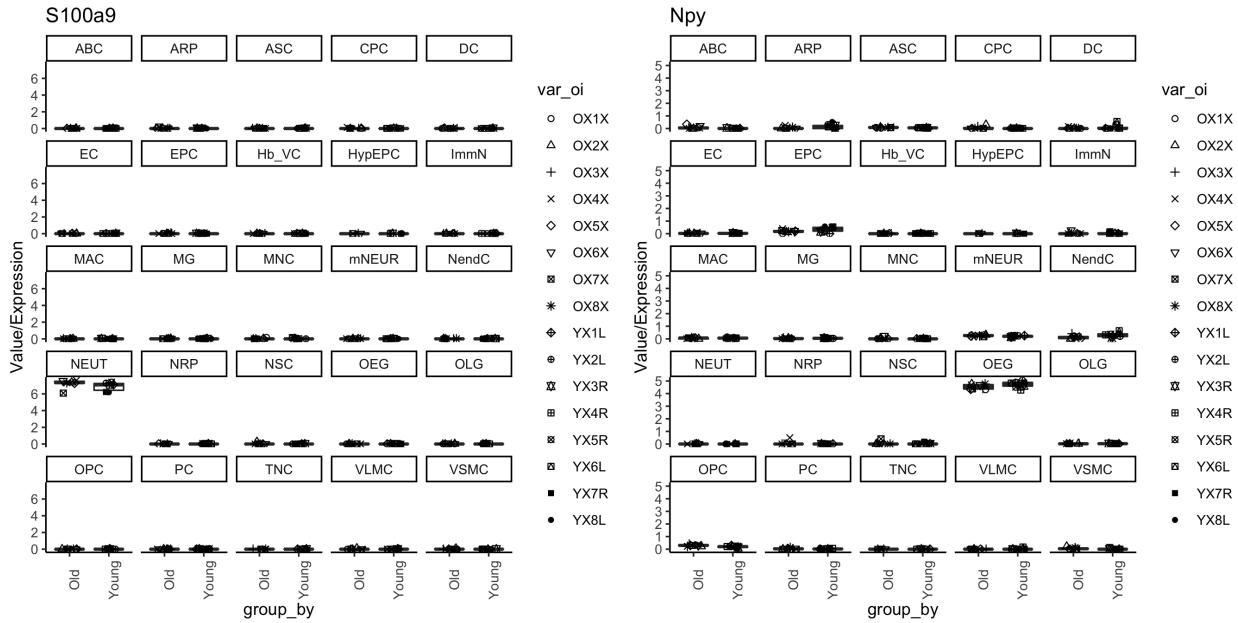
```
var_decomp <- palmo_obj@result$variance_decomposition
head(var_decomp[,featureSet])
#      Age_group Subject_id   cluster
#Rpl6     18.927915 13.6210861 29.057360
#Gm26917 14.853344 14.8230225 35.393884
#PISD    13.413587  0.5785149 34.217585
#mt-Atp8 13.361711  7.6618346 9.437131
#Rpl23a  13.238554 24.1634462 25.245072
#Gm26825 9.885817  0.0000000 0.0000000

#Variance explained (Donor, Time, and celltype)
plots <- variancefeaturePlot(vardata=var_decomp, featureSet=featureSet,
                               cols=c("purple", "darkgreen", "cyan"))
plot_grid(plotlist = plots, ncol=3)
```



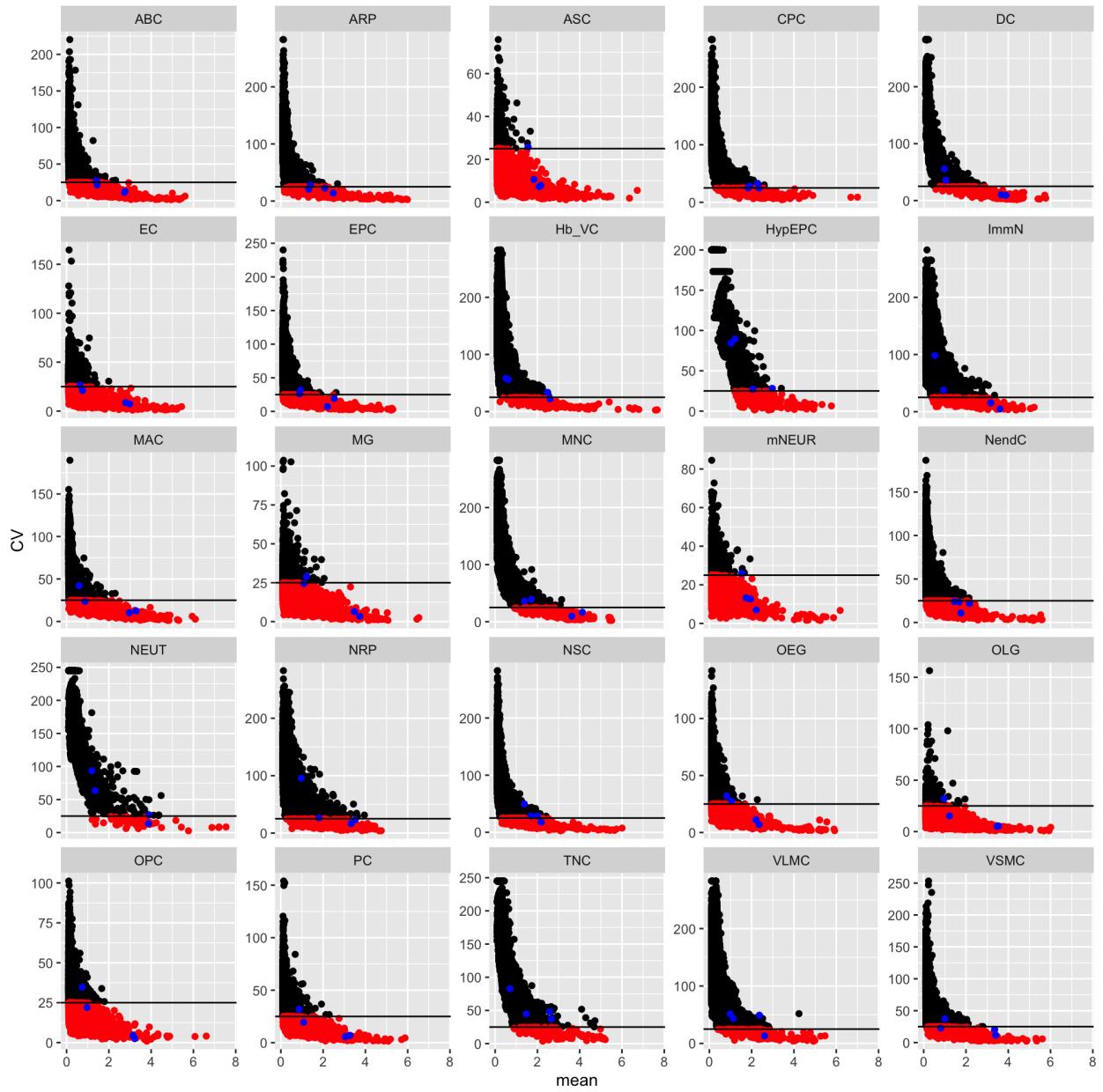
3.7.6 Plot the variables (Time ~ 10sec)

```
plots <- gene_featureplot(data_object=palmo_obj,
                            featureList=c("S100a9", "Npy"),
                            x_group_by="Age_group", var_oi="Subject_id",
                            facet_by = "cluster",
                            x_text_angle=90)
plot_grid(plotlist=plots, ncol= 2, align="hv")
```



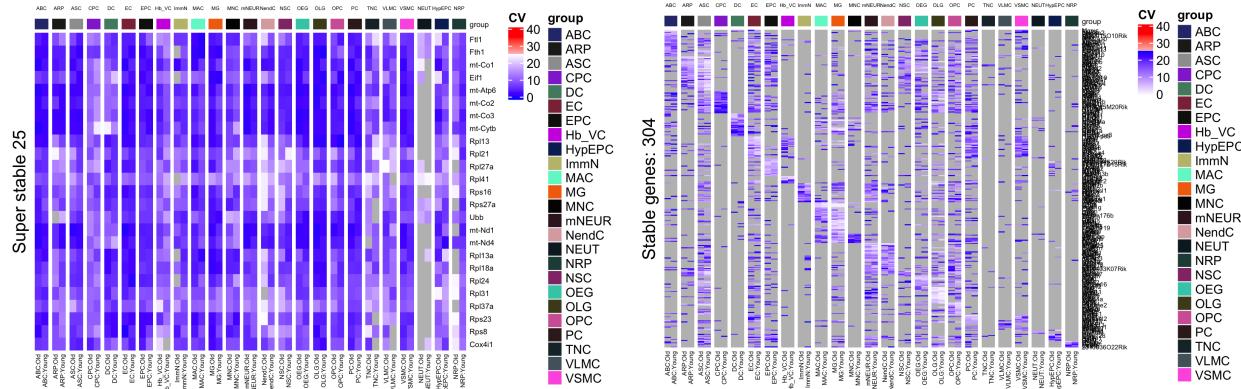
3.7.7 Intra-donor variations over time (Time ~ 4min)

```
#Calculate CV
palmo_obj <- cvCalcSC(data_object=palmo_obj,
                        meanThreshold=0.1, cvThreshold=25,
                        housekeeping_genes=c("Gapdh", "Actb"),
                        fileName="GSE129788")
```



3.7.8 Find stable and variable features in longitudinal data (Time ~ 30sec)

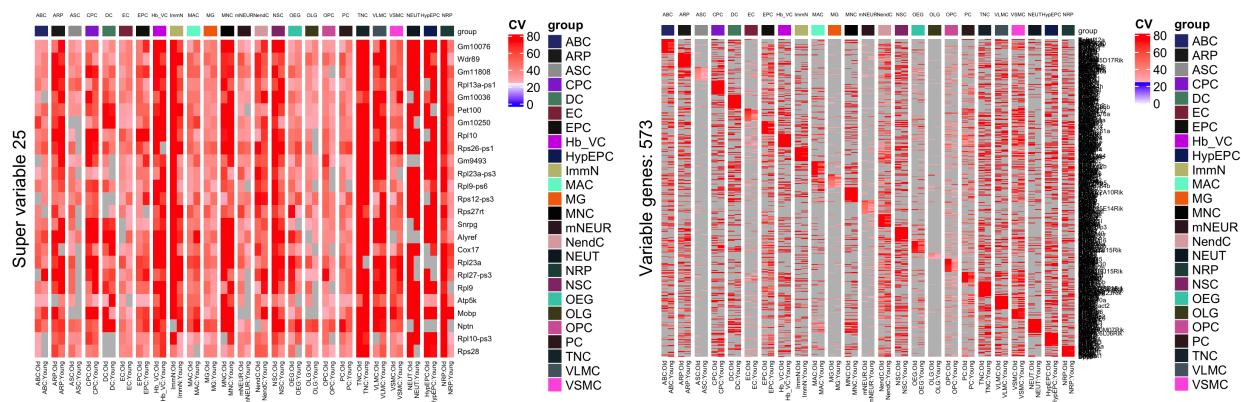
```
palmo_obj <- StableFeatures(data_object=palmo_obj,
                               cvThreshold=25,
                               topFeatures=25,
                               fileName="GSE129788")
stable_genes <- palmo_obj@result$stable_genes
```



```

palmo_obj <- VarFeatures(data_object=palmo_obj,
                           cvThreshold=25,
                           topFeatures=25,
                           fileName="GSE129788")
var_genes <- palmo_obj@result$var_genes

```

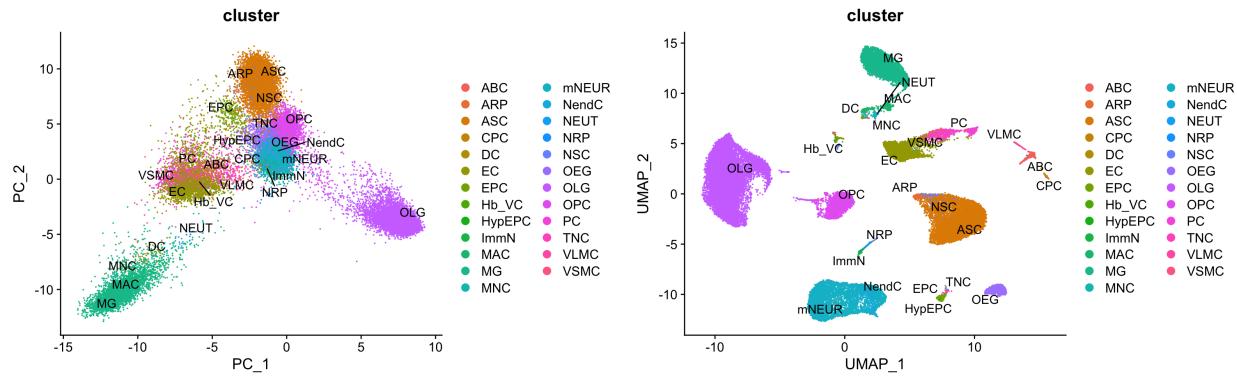


3.7.9 UMAP Plot (Time ~ 2min)

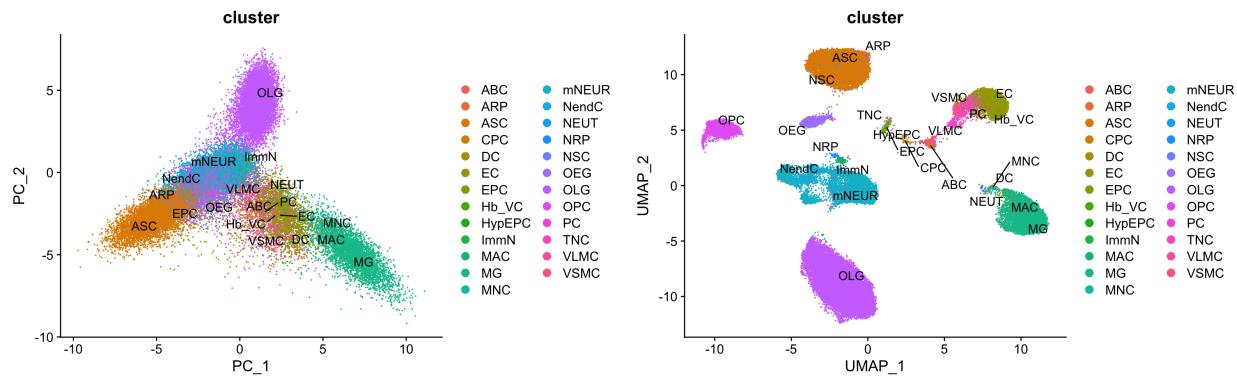
```

#Stable genes UMAP
dimUMAPPlot(data_object=palmo_obj, nPC=15,
             gene_oi=unique(stable_genes$gene),
             group_column="cluster", plotname="stable",
             fileName="GSE129788")

```



```
#Variable genes UMAP
dimUMAPPlot(data_object=palmo_obj, nPC=15,
             gene_oi=unique(var_genes$gene),
             group_column="cluster", plotname="variable",
             fileName="GSE129788")
```

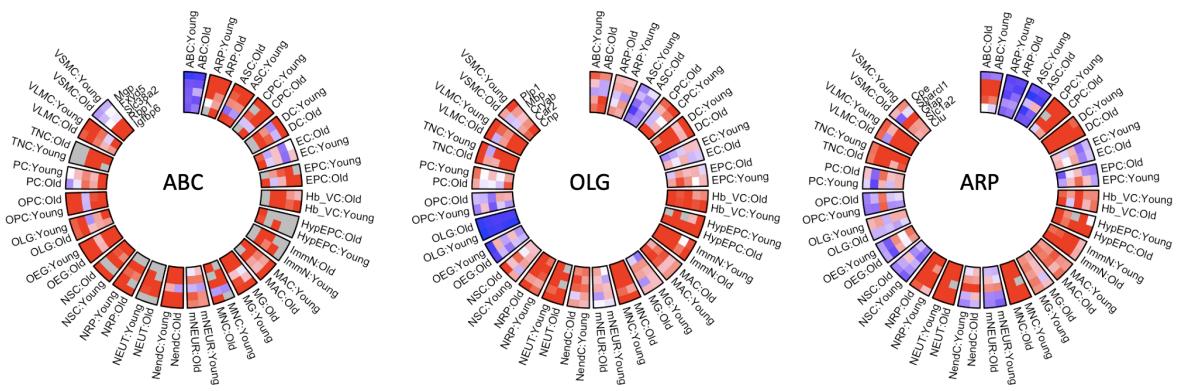


3.7.10 Celltype-specific Circos CV Plot (Time ~ 30sec)

```
#ABC: Arachnoid barrier cells
geneList <- c("Mgp", "Fxyd5", "Slc38a2", "Rbp1", "Igfbp6")
plotres <- genecircosPlot(data_object=palmo_obj,
                           geneList=geneList, colorThreshold=25)

#Olg: Oligodendrocytes
geneList <- c("Plp1", "Mbp", "Cryab", "Car2", "Cnp")
plotres <- genecircosPlot(data_object=palmo_obj,
                           geneList=geneList, colorThreshold=25)

#ARP: Astrocyte-restricted precursors
geneList <- c("Cpe", "Sparcl1", "Gfap", "Slc1a2", "Clu")
plotres <- genecircosPlot(data_object=palmo_obj,
                           geneList=geneList, colorThreshold=25)
```



4 Authors

Suhas Vasaikar, Aarthi talla and Xiaojun Li designed the PALMO algorithm. Suhas Vasaikar implemented the PALMO package.

5 License

PALMO is licensed under the MIT-License.

6 Session info

```
sessionInfo()
#> R version 4.0.3 (2020-10-10)
#> Platform: x86_64-apple-darwin17.0 (64-bit)
#> Running under: macOS Catalina 10.15.7
#>
#> Matrix products: default
#> BLAS:    /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
#> LAPACK:  /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> attached base packages:
#> [1] grid      stats     graphics  grDevices  utils      datasets  methods
#> [8] base
#>
#> other attached packages:
#> [1] PALMO_0.1.0
#>
#> loaded via a namespace (and not attached):
#> [1] readxl_1.3.1           backports_1.2.0
#> [3] circlize_0.4.11        plyr_1.8.6
#> [5] igraph_1.2.8           lazyeval_0.2.2
#> [7] splines_4.0.3          listenr_0.8.0
#> [9] scattermore_0.7         GenomeInfoDb_1.24.2
#> [11] ggplot2_3.3.5          digest_0.6.28
#> [13] htmltools_0.5.2        fansi_0.5.0
#> [15] magrittr_2.0.1          tensor_1.5
#> [17] cluster_2.1.0          ROCR_1.0-11
#> [19] ComplexHeatmap_2.4.3   globals_0.14.0
#> [21] readr_1.4.0             modelr_0.1.8
#> [23] matrixStats_0.61.0     colorspace_2.0-2
#> [25] rvest_0.3.6            blob_1.2.1
#> [27] ggrepel_0.9.1          haven_2.3.1
#> [29] xfun_0.25              dplyr_1.0.7
#> [31] RCurl_1.98-1.2          crayon_1.4.2
#> [33] jsonlite_1.7.2          lme4_1.1-25
#> [35] spatstat_1.64-1         spatstat.data_2.1-0
#> [37] survival_3.2-7          zoo_1.8-9
#> [39] glue_1.5.0              polyclip_1.10-0
#> [41] gtable_0.3.0             zlibbioc_1.34.0
#> [43] XVector_0.28.0          leiden_0.3.9
#> [45] DelayedArray_0.14.1     GetoptLong_1.0.4
#> [47] SingleCellExperiment_1.10.1 future.apply_1.8.1
#> [49] shape_1.4.5              BiocGenerics_0.34.0
#> [51] abind_1.4-5              scales_1.1.1
#> [53] pheatmap_1.0.12          DBI_1.1.0
#> [55] miniUI_0.1.1.1          Rcpp_1.0.7
#> [57] viridisLite_0.4.0         xtable_1.8-4
#> [59] clue_0.3-57              reticulate_1.22
#> [61] stats4_4.0.3              htmlwidgets_1.5.4
```

```

#> [63] httr_1.4.2
#> [65] ellipsis_0.3.2
#> [67] factoextra_1.0.7.999
#> [69] farver_2.1.0
#> [71] uwot_0.1.10
#> [73] deldir_1.0-6
#> [75] tidyselect_1.1.1
#> [77] reshape2_1.4.4
#> [79] munsell_0.5.0
#> [81] tools_4.0.3
#> [83] broom_0.7.2
#> [85] evaluate_0.14
#> [87] fastmap_1.1.0
#> [89] goftest_1.2-3
#> [91] fs_1.5.0
#> [93] purrrr_0.3.4
#> [95] pbapply_1.5-0
#> [97] nlme_3.1-149
#> [99] xml2_1.3.2
#> [101] plotly_4.10.0
#> [103] spatstat.utils_2.2-0
#> [105] tweenr_1.0.1
#> [107] statmod_1.4.35
#> [109]forcats_0.5.0
#> [111] Matrix_1.3-4
#> [113] vctrs_0.3.8
#> [115] lifecycle_1.0.1
#> [117] GlobalOptions_0.1.2
#> [119] bitops_1.0-7
#> [121] cowplot_1.1.1
#> [123] GenomicRanges_1.40.0
#> [125] patchwork_1.1.1
#> [127] promises_1.2.0.1
#> [129] gridExtra_2.3
#> [131] parallelly_1.28.1
#> [133] boot_1.3-25
#> [135] assertthat_0.2.1
#> [137] MAST_1.14.0
#> [139] SeuratObject_4.0.2
#> [141] GenomeInfoDbData_1.2.3
#> [143] mgcv_1.8-33
#> [145] hms_0.5.3
#> [147] tidyverse_1.3.0
#> [149] minqa_1.2.4
#> [151] Rtsne_0.15
#> [153] Biobase_2.48.0
#> [155] lubridate_1.7.9

RColorBrewer_1.1-2
Seurat_4.0.0
ica_1.0-2
pkgconfig_2.0.3
dbplyr_1.4.4
utf8_1.2.2
rlang_0.4.12
later_1.3.0
cellranger_1.1.0
generics_0.1.1
ggridges_0.5.3
stringr_1.4.0
yaml_2.2.1
knitr_1.30
fitdistrplus_1.1-6
RANN_2.6.1
future_1.23.0
mime_0.12
compiler_4.0.3
png_0.1-7
reprex_0.3.0
tibble_3.1.6
stringi_1.7.5
lattice_0.20-41
nloptr_1.2.2.2
pillar_1.6.4
lmtest_0.9-39
RcppAnnoy_0.0.19
data.table_1.14.2
irlba_2.3.3
httpuv_1.6.3
R6_2.5.1
KernSmooth_2.23-17
IRanges_2.22.2
codetools_0.2-16
MASS_7.3-53
SummarizedExperiment_1.18.2
rjson_0.2.20
sctransform_0.3.2
S4Vectors_0.26.1
parallel_4.0.3
rpart_4.1-15
tidyR_1.1.4
rmarkdown_2.5
ggforce_0.3.2
shiny_1.7.1

```