R for cytometry - phenograph and spade

Nello Blaser

Department of Mathematics, University of Bergen

March 16th 2018

Download data from cytobank

```r
# load CRAN package
require(CytobankAPI)
# Authenticate via authentication token
cyto_session <- authenticate(site="cellmass",
                             auth_token="xxx")
# specify experiment id
experiment_id <- 12
# specify data directory
data_dir <- "fcs_from_cytobank"
dir.create(data_dir, recursive = TRUE)
# list fcs files
files <- fcs_files.list(cyto_session,
                        experiment_id = experiment_id)
```

Download data from cytobank

```
# download fcs files
downloaded_zip <- fcs_files.download_zip(
  UserSession = cyto_session,
  experiment_id = experiment_id,
  fcs_files = files[, "id"],
  directory = data_dir,
  timeout = 60*nrow(files))
unzip(downloaded_zip, exdir = data_dir)
```

Read fcs files

```
require(flowCore)
# read fcs file
files <- paste0(data_dir, "/",
                list.files(data_dir,
                           pattern = "\\.fcs"))
fset <- read.flowSet(files)
```

Transform data

```
# decide on lineage markers
pData(fset[[1]]@parameters)
lineage_markers <- c(
  "In(114.903)-Dual", "Cd(109.903)-Dual",
  "Nd(143.910)-Dual", "Nd(144.912)-Dual",
  "Nd(145.913)-Dual", "Nd(147.916)-Dual",
  "Sm(146.914)-Dual", "Er(166.932)-Dual")
# asinh transform
fset_lineage <- fsApply(fset, function(ff){
  ff <- ff[, lineage_markers]
  exprs(ff) <- asinh(exprs(ff)/5)
  ff
})
```

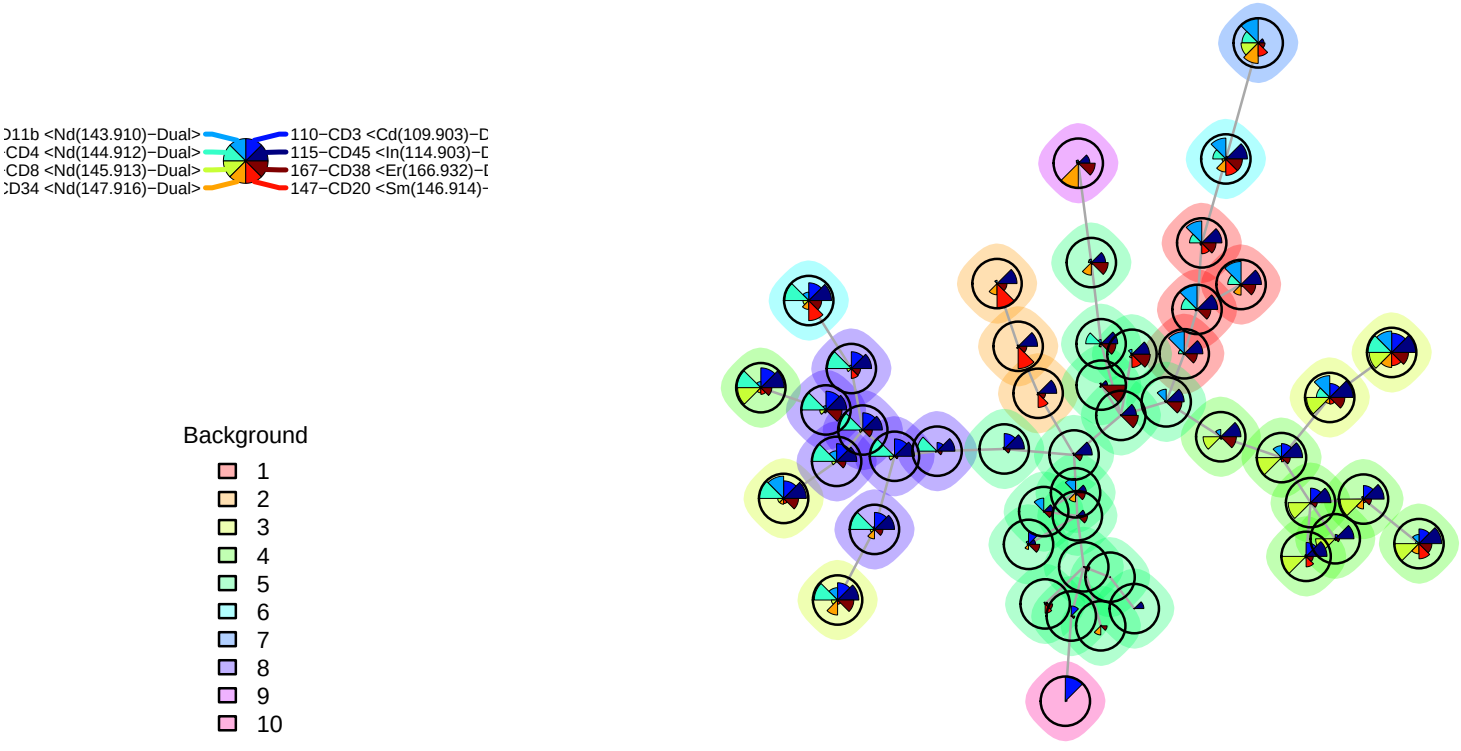
FlowSOM

```
require(FlowSOM)
# Set SOM parameters
flowSOM_metaClusters <- 10
flowSOM_xdim <- 7
flowSOM_ydim <- 7
flowSOM_seed <- 20180309 # for reproducible results

# run self-organizing maps
fsom <- FlowSOM(fset_lineage,
                colsToUse = lineage_markers,
                xdim=flowSOM_xdim,
                ydim=flowSOM_xdim,
                nClus = flowSOM_metaClusters,
                seed = flowSOM_seed)
```

Plotting with FlowSOM

```
PlotStars(UpdateNodeSize(fsom[[1]], reset=T),
          view = "MST",
          backgroundValues = as.factor(fsom[[2]]))
```



Subsample

```
# set a seed for reproducible results
set.seed(20180308)
# extract data
concatinated_dta <- fsApply(fset_lineage, exprs)
# subsample
subsample <- sample(1:nrow(concatinated_dta), 1000)
# subsampled data
subsampled_dta <- concatinated_dta[subsample, ]
```

Phenograph

```
## devtools::install_github("JinmiaoChenLab/Rphenograph")
require(Rphenograph)
# Set spade parameters
phenograph_neighbors <- 50

# run phenograph
pheno <- Rphenograph(subsampled_dta[, lineage_markers],
                     k = phenograph_neighbors)
```

```
##   Finding nearest neighbors...DONE ~ 0.01 s
##   Compute jaccard coefficient between nearest-neighbor s
##   Build undirected graph from the weighted links...DONE
##   Run louvain clustering on the graph ...DONE ~ 0.08 s
##   Return a community class
##   -Modularity value: 0.7380464
##   -Number of clusters: 8
```

```
pheno_cluster <- factor(membership(pheno[[2]]))
```

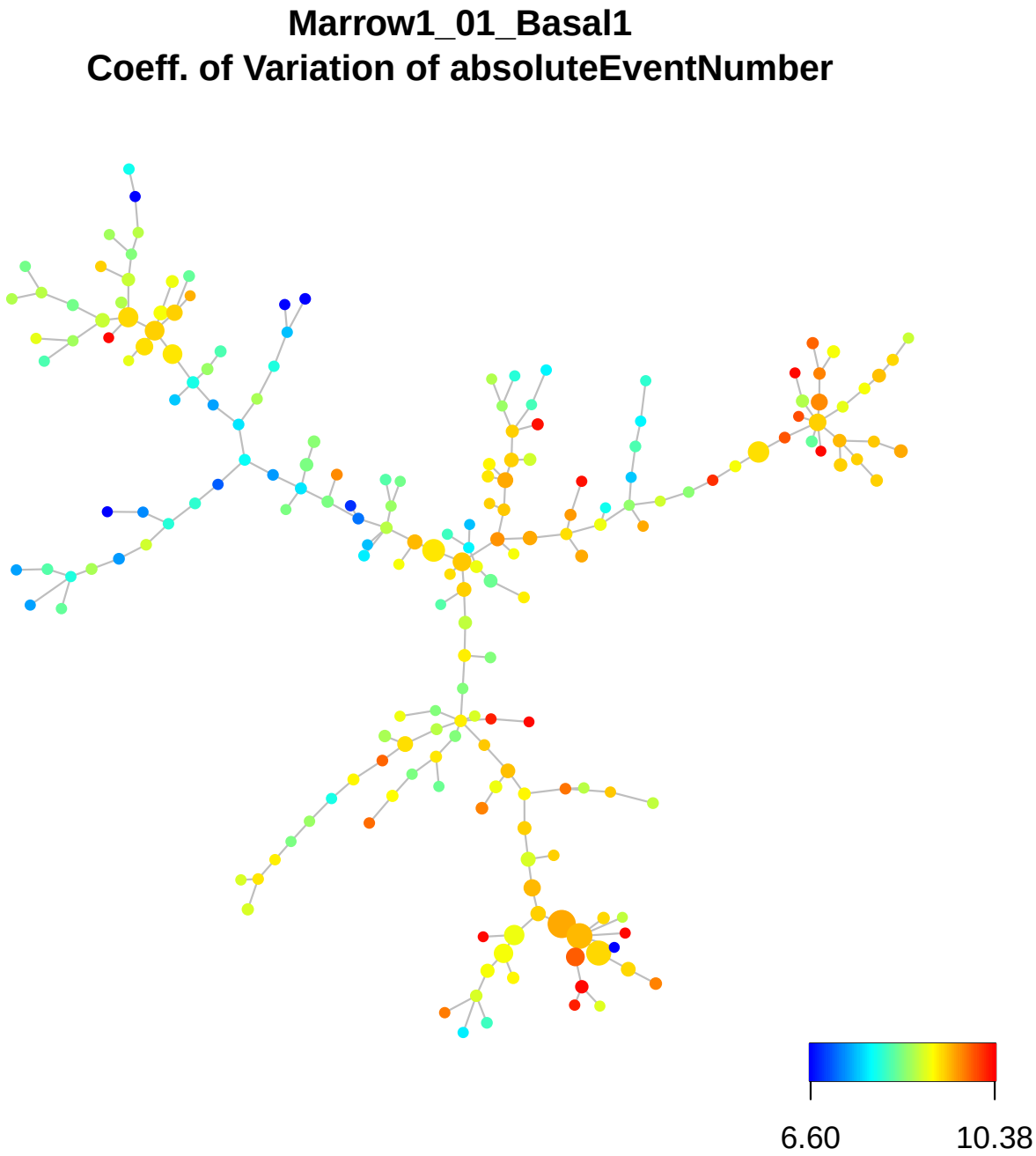
SPADE

```
require(spade)
# prepare to run spade
result_dir <- "spade_results"
dir.create(result_dir)
n_spade_clusters <- 200
# run spade
SPADE.driver(files,
             out_dir = result_dir,
             cluster_cols = lineage_markers,
             k = n_spade_clusters)
```

Plot SPADE

```
# get necessary information
layout <- read.table(paste0(result_dir, "/layout.table"))
mst <- read.graph(paste0(result_dir, "/mst.gml"),
                  format = "gml")
# plotting
SPADE.plot.trees(mst, result_dir,
                 file_pattern = "*fcs*Rsave",
                 layout = as.matrix(layout),
                 out_dir = result_dir,
                 size_scale_factor = 1.2)
```

Plot SPADE


Marrow1_01_Basal1
Coeff. of Variation of absoluteEventNumber
6.60
10.38

t-SNE

```
# load tSNE package
require(Rtsne)
# Run tSNE
tsne_out <- Rtsne(subsampled_dta[, lineage_markers])
```

Combine results

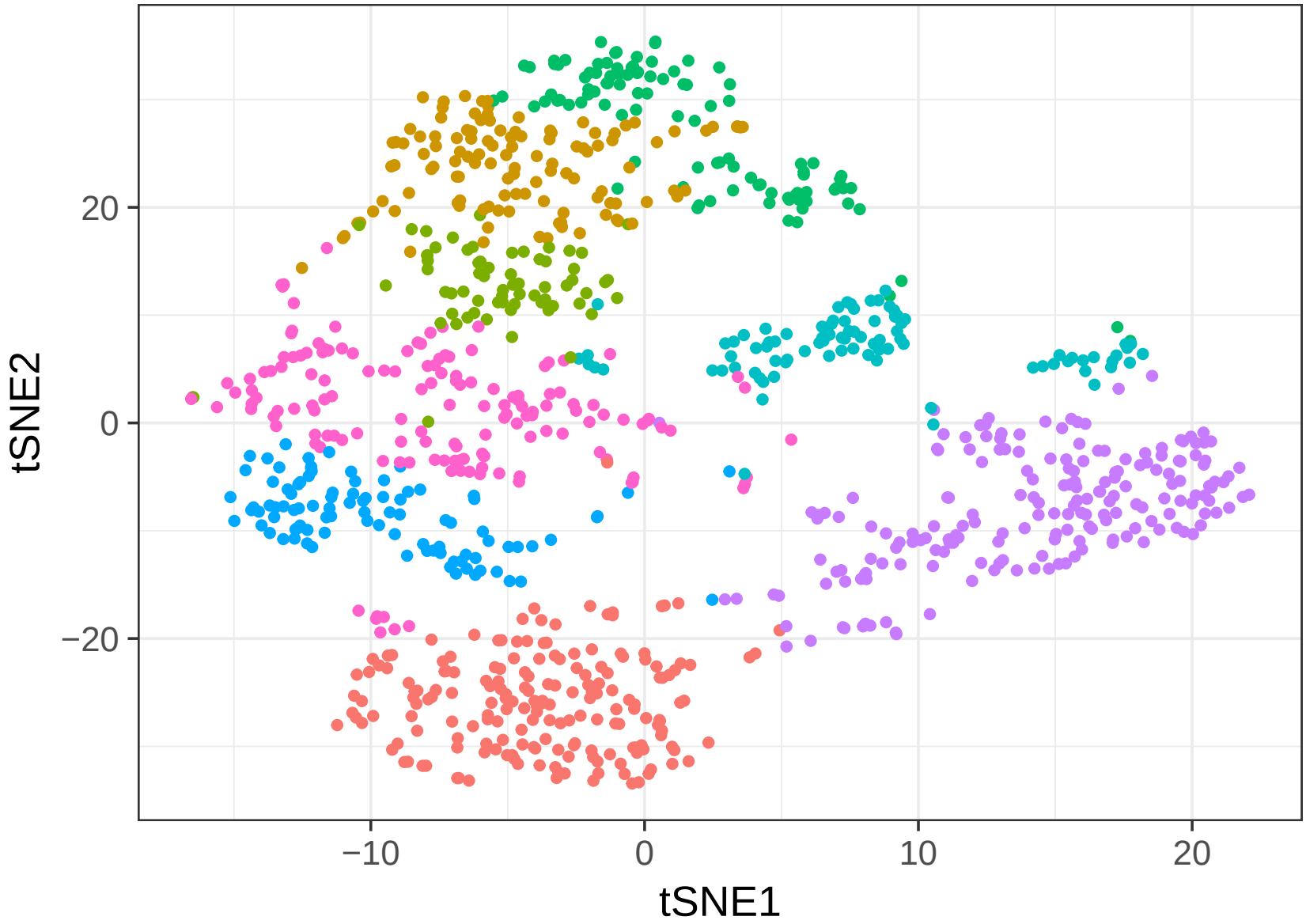
```r
# som results
fsom_cluster <- fsom$FlowSOM$map$mapping[subsample, 1]
fsom_meta_cluster <- fsom$metaclustering[fsom_cluster]
# combined results
result <- data.frame(
  tSNE1 = tsne_out$Y[, 1],
  tSNE2 = tsne_out$Y[, 2],
  fsom_cluster = factor(fsom_cluster),
  fsom_meta_cluster = fsom_meta_cluster,
  pheno_cluster = factor(pheno_cluster))
```

Plot

```
require(ggplot2)
p_phenograph <- ggplot(result, aes(x = tSNE1, y = tSNE2,
                                   color = pheno_cluster)) +
  geom_point() +
  scale_color_discrete(guide = FALSE) +
  theme_bw(base_size = 16)
```

Plot

```
p_phenograph
```



Prepare data update

```
# numeric results
require(dplyr)
cols <- c("tSNE1", "tSNE2", "pheno_cluster",
            "fsom_cluster", "fsom_meta_cluster")
result <- dplyr::mutate_at(result, cols, as.numeric)
# files and sample ids
result$filenames <-
  rep(sampleNames(fset),
      fsApply(fset, nrow, use.exprs = TRUE))[subsample]
result$sample_id <- subsample
result$sample_fid <-
  apply(sapply(c(0, cumsum(fsApply(fset, nrow,
                                   use.exprs = TRUE))),
               function(x) subsample - x), 1,
      function(y) min(y[y>0]))
```

Update data

```
update_ff <- function(ff, res_data, cols){
  rd <- res_data[res_data$filenames == identifier(ff), ]
  ff@exprs <- cbind(ff@exprs[rd$sample_fid, ],
                    as.matrix(rd[, cols]))
  ff@parameters@data <- rbind(
    ff@parameters@data,
    data.frame(
      name = cols,
      desc = "R_cols",
      range = apply(rd[, cols], 2, function(x)
        diff(range(x))),
      minRange = apply(rd[, cols], 2, min),
      maxRange = apply(rd[, cols], 2, max),
      row.names =
        paste0("$P", nrow(ff@parameters@data) +
                 1:length(cols))))
  ff
}
```

Update data

```
# change data
fset_new <- fsApply(fset, update_ff,
                    res_data = result,
                    cols = cols)
fset_new
```

```
## A flowSet with 3 experiments.
##
##   column names:
##   Time Cell_length Ir(190.960)-Dual Ir(192.962)-Dual Rh
```

Writing FCS file

```
# write file
new_files <- gsub(".fcs", "_clustered.fcs", files)
flowCore::write.flowSet(fset_new,
                        outdir = ".",
                        filename = new_files)
```

```
## [1] "."
```

```
# zip output files
zip_filename <- paste0(data_dir, "/clustered_som.zip")
zip(zip_filename, new_files)
```

Uploading files to cytobank

```
# create new experiment
new_experiment_name <- "test_clustered"
new_experiment <- experiments.new(
  cyto_session,
  experiment_name = new_experiment_name,
  purpose = "testing")

# upload files
fcs_files.upload_zip(cyto_session,
                     experiment_id = new_experiment$id,
                     file_path = zip_filename,
                     timeout = 720)
```