(Fast) Significance Analysis Demo

Izzy Grabski

2022-10-14

We introduce a significance analysis approach to clustering single-cell RNA-sequencing (scRNA-seq) data. Our approach is implemented in two ways. First, we present a stand-alone clustering pipeline, called single-cell significance of hierarchical clustering (sc-SHC), that builds hypothesis testing into a hierarchical clustering framework. Second, we offer a way to apply significance analysis to any set of pre-computed clusters.

In this demo, we construct a dataset with two ground-truth populations and show how our approach can be applied.

Single-Cell Significance of Hierarchical Clustering

First, we load in 293T scRNA-seq data from 10X, subset to a small sample of 200 cells for demonstration purposes, and create two distinct populations by permuting the genes of the first 100 cells.

```
library(BiocFileCache)
library(Seurat)

set.seed(73122)

# Load in 10X data
bfc <- BiocFileCache(ask=FALSE)
path <- bfcrpath(bfc,'https://cf.10xgenomics.com/samples/cell-exp/1.1.0/293t/293t_filtered_gene_bc_matri
tmp <- tempfile()
untar(path,exdir=tmp)
data <- Read10X(data.dir = file.path(tmp,'filtered_matrices_mex','hg19'))

# Create two populations
data <- as.matrix(data[,1:200])
top50 <- order(rowSums(data),decreasing=T)[1:50]
bottom50 <- order(rowSums(data),decreasing=F)[1:50]
data[c(top50,bottom50),1:100] <- data[sample(c(top50,bottom50),replace=F),1:100]
labels <- c(rep('population1',100),rep('population2',100))
```

Next, we apply our clustering pipeline to these data. Because this demo dataset is on the small end, we choose to use only six principal components and 100 features; however, by default, the number of principal components is set to 30 and the number of features to 2500, which is what we generally recommend. We also keep the family-wise error rate (alpha) set to 0.05, which is the default setting. Note that this is a fairly conservative setting, and when the risk of false discoveries is more tolerable (for example, in exploratory analysis, when the goal is to detect possibly rare populations), this parameter can be increased.

```
source('significance_analysis_fast.R')

# Apply sc-SHC
clusters <- scSHC(data,num_PCs=6,num_features=100)

table(clusters,labels)
```

```
##         labels
## clusters population1 population2
##        1         100           0
##        2           0         100
```

Our approach finds exactly the right number of clusters (2), with perfect concordance to the ground-truth labels.

Significance Analysis on Pre-Computed Clusters

We now demonstrate our significance analysis approach applied on pre-computed clusters. These clusters can arise from any clustering algorithm, including when manual changes were made to the output. Here, we show results when applying the Louvain algorithm from Seurat.

```
# Cluster with Seurat
data.seurat <- CreateSeuratObject(data)
data.seurat <- NormalizeData(data.seurat)
data.seurat <- FindVariableFeatures(data.seurat,nfeatures=100)
data.seurat <- ScaleData(data.seurat)
data.seurat <- RunPCA(data.seurat,npcs=6)
data.seurat <- RunUMAP(data.seurat,features=VariableFeatures(data.seurat))
data.seurat <- FindNeighbors(data.seurat,dims=1:6)
data.seurat <- FindClusters(data.seurat,resolution=1)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 200
## Number of edges: 6885
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.5013
## Number of communities: 3
## Elapsed time: 0 seconds
```

```
table(Idents(data.seurat),labels)
```

```
##    labels
##     population1 population2
##   0         100           0
##   1           0          77
##   2           0          23
```

At a resolution parameter of 1, using the same number of genes and PCs as our approach, Seurat found 3 clusters, where two clusters subdivide the second population. We now apply our significance analysis approach on top of these clusters. We use the same parameters as before due to the small size of these demo data, but to re-iterate, we generally recommend sticking to the default parameters.

```
# Apply significance analysis on Seurat's clusters
new_seurat <- testClusters(data,as.character(Idents(data.seurat)),num_PCs=6,
                           num_features=100)

table(new_seurat,Idents(data.seurat))
```

```
##
## new_seurat   0   1   2
##       new1 100   0   0
##       new2   0  77  23
```

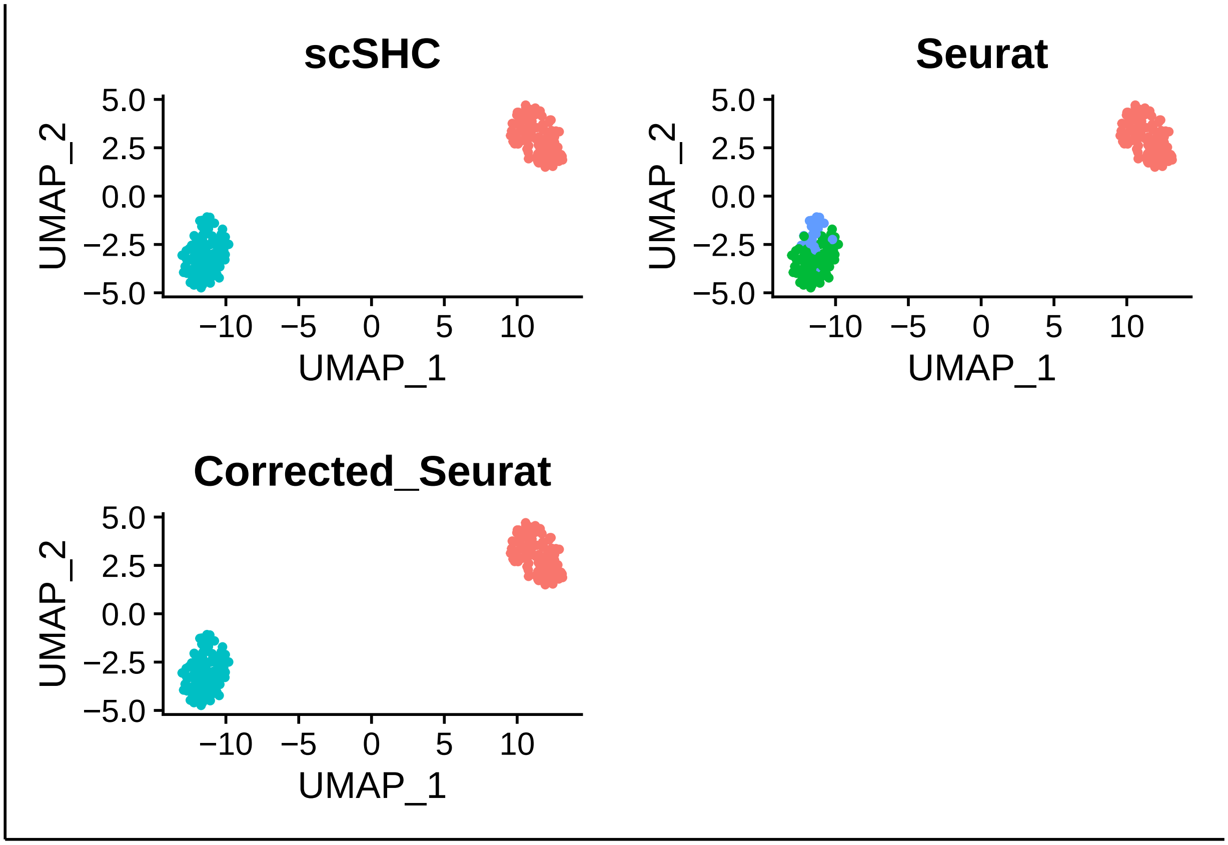
Our approach merged the two extra clusters back into one, finding again the exact right subdivision of the data.

Finally, we visualize all 3 clustering results.

```
library(ggplot2)
library(ggpubr)

# Visualize clustering results
data.seurat$scSHC <- clusters
data.seurat$Seurat <- Idents(data.seurat)
data.seurat$Corrected_Seurat <- new_seurat

ggarrange(DimPlot(data.seurat,group.by='scSHC')+NoLegend(),
          DimPlot(data.seurat,group.by='Seurat')+NoLegend(),
          DimPlot(data.seurat,group.by='Corrected_Seurat')+NoLegend())+
  theme_classic()
```


scCHC
Seurat
Corrected_Seurat
UMAP_1
UMAP_2
5.0
2.5
0.0
−2.5
−5.0
−10
−5
0
5
10