Application of RESET to 10x PBMC 3k scRNA-seq data using Seurat log normalization for the MSigDB BioCarta collection.

H. Robert Frost

1 Load the RESET package

Loading RESET will also load the required package Matrix. Seurat is referenced via suggests so must be directly loaded to enable access to Seurat functions.

```
> library(RESET)
> library(Seurat)
```

2 Load and process the 10x PBMC scRNA-seq data

This example uses the same 10x PBMC scRNA-seq data set that is used in the Seurat Guided Clustering vignette
(https://satijalab.org/seurat/v3.1/pbmc3k_tutorial.html). The Cell Ranger files for this data set can be downloaded from
https://s3-us-west-2.amazonaws.com/10x.files/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz. This data is loaded and processed using the same Seurat logic found in the Guided Clustering vignette. In particular, the Seurat log normalization method implemented by NormalizeData() is used with variable genes determined by FindVariableFeatures().

```
> # update the data.dir argument to reflect the local location of the PBMC data
> pbmc.data = Read10X(data.dir = "./filtered_gene_bc_matrices/hg19/")
> pbmc = CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3,
+         min.features = 200)
> pbmc[["percent.mt"]] = PercentageFeatureSet(pbmc, pattern = "^MT-")
> pbmc = subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)
> pbmc = NormalizeData(pbmc)
> pbmc = FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
> pbmc = ScaleData(pbmc, features = rownames(pbmc))
> pbmc = RunPCA(pbmc, features = VariableFeatures(object = pbmc))
> pbmc = RunUMAP(pbmc, dims = 1:10)
> pbmc = FindNeighbors(pbmc, dims = 1:10)
> pbmc = FindClusters(pbmc, resolution = 0.5)

Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 2638
Number of edges: 95965

Running Louvain algorithm...
Maximum modularity in 10 random starts: 0.8723
Number of communities: 9
Elapsed time: 0 seconds
```

```
> # Assign the known cell type labels to the clusters (this follows the Seurat vingette)
> cell.types = c("Memory CD4 T",  "CD14+ Mono", "Naive CD4 T", "B", "CD8 T", "FCGR3A+ Mono",
+                        "NK", "DC", "Platelet")
> names(cell.types) = levels(pbmc)
> pbmc = RenameIdents(pbmc, cell.types)
> pbmc

An object of class Seurat
13714 features across 2638 samples within 1 assay
Active assay: RNA (13714 features, 2000 variable features)
 2 dimensional reductions calculated: pca, umap
```

3 Load Ensembl IDs

The Ensembl IDs and gene names must be read in from the genes.tsv file and filtered to match genes left after the quality control steps performed in the prior section.

```
> feature.data = read.delim("./filtered_gene_bc_matrices/hg19/genes.tsv",
+         header = FALSE, stringsAsFactors = FALSE)
> ensembl.ids = feature.data[,1]
> gene.names = feature.data[,2]
> genes.after.QC = rownames(pbmc@assays$RNA@counts)
> indices.to.keep = unlist(sapply(genes.after.QC, function(x) {which(gene.names == x)[1]}))
> ensembl.ids = ensembl.ids[indices.to.keep]
> gene.names = gene.names[indices.to.keep]
```

4 Load the MSigDB BioCarta collection

The following logic loads the MSigDB BioCarta collection using the msigdbr R package. Because the MSigDB gene sets are defined in terms of Entrez gene IDs, these must be mapped to Ensembl IDs using the org.HS.eg.db R package (from Bioconductor). The data frame returned by msigdbr is then converted into a list of gene ID vectors (each list element corresponds to a gene set and is a vector of Ensembl IDs). This list is transformed into the structure required by resetForSeurat() using the createVarSetCollection() helper function. This helper function filters out genes not also contained in the PBMC scRNA-seq data and generates a list whose elements are vectors of gene indices in the scRNA-seq data.

```
> library(msigdbr)
> library(org.Hs.eg.db)
> # Load the MSigDB BioCarta collection using the msigdbr package
> biocarta.collection = msigdbr(category="C2", subcategory="BIOCARTA")
> # Get the entrez gene IDs that are mapped to an Ensembl ID
> entrez2ensembl = mappedkeys(org.Hs.egENSEMBL)
> # Convert to a list
> entrez2ensembl = as.list(org.Hs.egENSEMBL[entrez2ensembl])
> # Convert Entrez IDs to Ensembl IDs using the org.Hs.eg.db package
> # Doing this a rather simplistic (and inefficient) way for clarity
> msigdb.entrez.ids = biocarta.collection$entrez_gene
> num.ids = length(msigdb.entrez.ids)
> msigdb.ensembl.ids = rep(NA, num.ids)
> for (i in 1:num.ids) {
```

```
+         entrez.id = msigdb.entrez.ids[i]
+         id.index = which(names(entrez2ensembl) == entrez.id)
+         if (length(id.index > 0)) {
+           # only use the first mapped ensembl id
+           msigdb.ensembl.ids[i] = entrez2ensembl[[id.index]][1]
+         }
+ }
> # Save the ensembl IDs in the data frame
> biocarta.collection$ensembl_gene = msigdb.ensembl.ids
> # Create a gene.set.collection list of Ensembl IDs
> gene.set.names = unique(biocarta.collection$gs_name)
> num.sets = length(gene.set.names)
> gene.set.collection = list()
> for (i in 1:num.sets) {
+         gene.set.name = gene.set.names[i]
+         gene.set.rows = which(biocarta.collection$gs_name == gene.set.name)
+         gene.set.ensembl.ids = biocarta.collection$ensembl_gene[gene.set.rows]
+         gene.set.collection[[i]] = unique(gene.set.ensembl.ids)
+ }
> names(gene.set.collection) = gene.set.names
> # Create the collection list required by resetForSeurat(); enforce minimum size of 10
> gene.set.collection.indices = createVarSetCollection(var.names=ensembl.ids,
+         var.sets=gene.set.collection, min.size=10)
> length(gene.set.collection.indices)

[1] 183
```

5 Execute RESET method

Since the scRNA-seq data has been processed using Seurat, we execute RESET using the resetForSeurat() function. First, execute without weights using reconstruction of the top 10 PCs, a reconstruction rank of 4, 2 oversampling dimensions, no power iterations, 5 total reconstruction iterations, standard normal RVs in the random test matrix, and L2 norm for computing scores from the reconstruction error.

```
> pbmc.reset = resetForSeurat(seurat.data=pbmc, num.pcs=15,
+     gene.set.collection=gene.set.collection.indices,
+     k=5,k.buff=0,q=0,random.threshold=30,
+     test.dist="normal", norm.type="2")
```

Look at the first few entries in the "RESET" assay that holds the cell-level gene set scores.

```
> pbmc.reset@assays$RESET[1:5,1:5]

5 x 5 sparse Matrix of class "dgCMatrix"
                       AAACATACAACCAC-1 AAACATTGAGCTAC-1 AAACATTGATCAGC-1
BIOCARTA-41BB-PATHWAY       -0.268239118     0.0050628822      0.028377406
BIOCARTA-ACH-PATHWAY         0.014869926     0.0001397117     -0.005719567
BIOCARTA-AGPCR-PATHWAY      -0.016702710     0.0158485920     -0.022033108
BIOCARTA-AGR-PATHWAY         0.043290455     0.0056903195     -0.008697033
BIOCARTA-AKAP95-PATHWAY      0.008133346    -0.0098122659     -0.023197334
                       AAACCGTGCTTCCG-1 AAACCGTGTATGCG-1
```

```
BIOCARTA-41BB-PATHWAY       0.2218418446      0.059534585
BIOCARTA-ACH-PATHWAY       -0.0081653004      0.437333498
BIOCARTA-AGPCR-PATHWAY      0.0140582038     -0.001465873
BIOCARTA-AGR-PATHWAY        0.0005412717      0.003161598
BIOCARTA-AKAP95-PATHWAY     0.0013006309     -0.002213141
```

Look at the top entries in the "RESET" assay feature metadata that holds the overall gene set scores

```
> overall.scores = pbmc.reset@assays$RESET@meta.features$RESET
> names(overall.scores) = rownames(pbmc.reset@assays$RESET@meta.features)
> sort(overall.scores, decreasing=TRUE)[1:10]
```

```
       BIOCARTA-CSK-PATHWAY      BIOCARTA-FCER1-PATHWAY
                  0.13817460                  0.10769021
       BIOCARTA-MHC-PATHWAY        BIOCARTA-CTL-PATHWAY
                  0.10699497                  0.10073764
  BIOCARTA-UCALPAIN-PATHWAY        BIOCARTA-TCR-PATHWAY
                  0.09449048                  0.09226303
   BIOCARTA-THELPER-PATHWAY BIOCARTA-TCYTOTOXIC-PATHWAY
                  0.08558502                  0.06708214
      BIOCARTA-IL17-PATHWAY   BIOCARTA-CHEMICAL-PATHWAY
                  0.06576168                  0.06203267
```

6 Compute DE pathways

Use the Seurat FindAllMarkers() function to identify BioCarta pathways whose weighted RESET scores are enriched within each cell type cluster according to a Wilcoxon test.

```
> library(dplyr)
> pbmc.markers = FindAllMarkers(pbmc.reset, assay="RESET", only.pos = TRUE, logfc.threshold = 0.1)
> pbmc.markers %>% group_by(cluster) %>% top_n(n = 5, wt = avg_log2FC)
```

```
# A tibble: 24 × 7
# Groups:   cluster [7]
       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster      gene
       <dbl>      <dbl> <dbl> <dbl>     <dbl> <fct>        <chr>
 1 4.71e- 31      0.140 0.799 0.763 8.63e- 29 Memory CD4 T BIOCARTA-MHC-PATHWAY
 2 3.17e- 30      0.116 0.875 0.79  5.80e- 28 Memory CD4 T BIOCARTA-CSK-PATHWAY
 3 2.74e- 17      0.110 0.608 0.579 5.01e- 15 Memory CD4 T BIOCARTA-CCR3-PATHWAY
 4 6.57e- 72      0.291 0.91  0.681 1.20e- 69 CD14+ Mono   BIOCARTA-TCR-PATHWAY
 5 8.40e- 69      0.317 0.842 0.693 1.54e- 66 CD14+ Mono   BIOCARTA-UCALPAIN-PA...
 6 1.18e- 45      0.198 0.273 0.856 2.16e- 43 CD14+ Mono   BIOCARTA-INFLAM-PATH...
 7 8.36e- 35      0.188 0.715 0.648 1.53e- 32 CD14+ Mono   BIOCARTA-RACCYCD-PAT...
 8 3.76e- 34      0.183 0.715 0.633 6.87e- 32 CD14+ Mono   BIOCARTA-AKT-PATHWAY
 9 7.34e-128      0.457 0.939 0.534 1.34e-125 B            BIOCARTA-BCR-PATHWAY
10 3.40e- 38      0.124 0.122 0.455 6.22e- 36 B            BIOCARTA-TALL1-PATHW...
# ... with 14 more rows
```

7 Visualize RESET scores

Visualize RESET scores using Seurat DoHeatmap(). The default Assay must first be changed to "RESET" and the slot parameter must be set to "data" in the call to DoHeatmap().

```
> library(ggplot2)
> DefaultAssay(object = pbmc.reset) = "RESET"
> top.pathways <- pbmc.markers %>% group_by(cluster) %>% top_n(n = 5, wt = avg_log2FC)
> DoHeatmap(pbmc.reset, slot="data", features = top.pathways$gene,  size=5, label=T) +
+   scale_fill_gradient2(low="blue", mid="white", high="orange", midpoint=0)
```