

Application of VAM to Seurat pbmc_small scRNA-seq data using Seurat log normalization.

H. Robert Frost

1 Load the VAM package

```
> library(VAM)
```

2 Summary statistics for the pbmc_small scRNA-seq data

This example uses the pbmc_small data set included in the SeuratObject package and a single contrived gene set. Please see the other vignettes for more realistic examples using larger scRNA-seq data sets and gene set collections based on MSigDB.

```
> if (requireNamespace("Seurat", quietly=TRUE)) {  
+   SeuratObject::pbmc_small  
+   gene.names = rownames(SeuratObject::pbmc_small)  
+   gene.names[1:5]  
+   Seurat::VariableFeatures(SeuratObject::pbmc_small)[1:5]  
+ } else {  
+   message("Seurat package not available! Not executing associated vignette content.")  
+ }
```

```
[1] "PPBP" "IGLL5" "VDAC3" "CD1C" "AKR1C3"
```

3 Define gene set collection

A gene set collection containing just a single contrived set (containing the top 5 variable genes) will be used for this example.

```
> if (requireNamespace("Seurat", quietly=TRUE)) {  
+   gene.set.name = "Test"  
+   gene.ids = c("PPBP", "IGLL5", "VDAC3", "CD1C", "AKR1C3")  
+   # Create a collection list for this gene set  
+   gene.set.id.list = list()  
+   gene.set.id.list[[1]] = gene.ids  
+   names(gene.set.id.list)[1] = gene.set.name  
+   gene.set.id.list  
+   # Create the list of gene indices required by vamForSeurat()  
+   (gene.set.collection = createGeneSetCollection(gene.ids=gene.names,  
+   gene.set.collection=gene.set.id.list))  
+   gene.indices = gene.set.collection[[1]]  
+   (gene.names = gene.names[gene.indices])  
+ } else {
```

```
+      message("Seurat package not available! Not executing associated vignette content.")
+ }

[1] "PPBP"    "IGLL5"   "VDAC3"   "CD1C"    "AKR1C3"
```

4 Execute VAM method

Since the scRNA-seq data has been processed using Seurat, we execute VAM using the `vamForSeurat()` function. We have set `return.dist=T` so that the squared adjusted Mahalanobis distances will be returned in a "VAMdist" Assay.

```
> if (requireNamespace("Seurat", quietly=TRUE)) {
+   pbmc.vam = vamForSeurat(seurat.data=SeuratObject::pbmc_small,
+     gene.set.collection=gene.set.collection,
+     center=F, gamma=T, sample.cov=F, return.dist=T)
+ } else {
+   message("Seurat package not available! Not executing associated vignette content.")
+ }
```

Look at the first few entries in the "VAMdist" and "VAMcdf" Assays.

```
> if (requireNamespace("Seurat", quietly=TRUE)) {
+   pbmc.vam@assays$VAMdist[1,1:10]
+   pbmc.vam@assays$VAMcdf[1,1:10]
+ } else {
+   message("Seurat package not available! Not executing associated vignette content.")
+ }
```

```
1 x 10 sparse Matrix of class "dgCMatrix"
```

```
Test . . 0.1063384 0.2849933 . . 0.3990145 . . 0.3237177
```

5 Visualize VAM scores

Visualize VAM scores using Seurat `FeaturePlot()`. The default Assay must first be changed to "VAMcdf".

```
> if (requireNamespace("Seurat", quietly=TRUE)) {
+   Seurat::DefaultAssay(object = pbmc.vam) = "VAMcdf"
+   Seurat::FeaturePlot(pbmc.vam, reduction="tsne", features=gene.set.name)
+ } else {
+   message("Seurat package not available! Not executing associated vignette content.")
+   par(mar = c(0,0,0,0))
+   plot(c(0, 1), c(0, 1), ann = F, bty = 'n', type = 'n', xaxt = 'n', yaxt = 'n')
+   text(x = 0.5, y = 0.5, paste("Seurat package not available!\n",
+     "FeaturePlot not generated."),
+     cex = 1.6, col = "black")
+ }
```

