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

H. Robert Frost

1 Load the VAM package

```
> library(VAM)
> if (!requireNamespace("Seurat", quietly=TRUE)) {
+    stop("Seurat package not available!")
+ }
```

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.

```
> SeuratObject::pbmc_small

An object of class Seurat
230 features across 80 samples within 1 assay
Active assay: RNA (230 features, 20 variable features)
 2 dimensional reductions calculated: pca, tsne

> gene.names = rownames(SeuratObject::pbmc_small)
> gene.names[1:5]

[1] "MS4A1"   "CD79B"   "CD79A"   "HLA-DRA" "TCL1A"
```

3 Apply SCTransform normalization to the data

```
> pbmc_sctransform = Seurat::SCTransform(SeuratObject::pbmc_small, verbose=F)
> # Compute PCA and UMAP on the normalized values
> pbmc_sctransform = Seurat::RunPCA(pbmc_sctransform, npcs=10)
> pbmc_sctransform = Seurat::RunUMAP(pbmc_sctransform, dims = 1:10)
> Seurat::VariableFeatures(pbmc_sctransform)[1:5]

[1] "NKG7"  "PPBP"  "GNLY"  "PF4"   "GNG11"
```

4 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.

```
> gene.set.name = "Test"
> gene.ids = c("NKG7", "PPBP", "GNLY", "PF4", "GNG11")
> # 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

$Test
[1] "NKG7"  "PPBP"  "GNLY"  "PF4"   "GNG11"

> # Create the list of gene indices required by vamForSeurat()
> (gene.set.collection = createGeneSetCollection(gene.ids=gene.names,
+         gene.set.collection=gene.set.id.list))

$Test
 NKG7  PPBP  GNLY   PF4 GNG11
   63   174   206   177   181

> gene.indices = gene.set.collection[[1]]
> (gene.names = gene.names[gene.indices])

[1] "NKG7"  "PPBP"  "GNLY"  "PF4"   "GNG11"
```

5 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.

```
> pbmc.vam = vamForSeurat(seurat.data=pbmc_sctransform,
+     gene.set.collection=gene.set.collection,
+     center=F, gamma=T, sample.cov=F, return.dist=T)
```

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

```
> pbmc.vam@assays$VAMdist[1,1:10]

1 x 10 sparse Matrix of class "dgCMatrix"

Test . 13.84296 5.929805 9.905785 1.482451 3.724075 . . . 16.35915

> pbmc.vam@assays$VAMcdf[1,1:10]

1 x 10 sparse Matrix of class "dgCMatrix"

Test . 0.5923161 0.3445219 0.4857771 0.11921 0.2451155 . . . 0.6475941
```

6 Visualize VAM scores

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

```
> Seurat::DefaultAssay(object = pbmc.vam) = "VAMcdf"
> Seurat::FeaturePlot(pbmc.vam, reduction="tsne", features=gene.set.name)
```