# Package 'VAM'

October 12, 2022

**Type** Package

**Title** Variance-Adjusted Mahalanobis

**Version** 1.0.0

**Author** H. Robert Frost

**Maintainer** H. Robert Frost <rob.frost@dartmouth.edu>

**Description** Contains logic for cell-specific gene set scoring of single cell RNA sequencing data.

**Depends** R (>= 3.6.0), MASS, Matrix

**Imports** methods (>= 3.6.0)

**Suggests** Seurat (>= 4.0.0), SeuratObject (>= 4.0.0), sctransform (>= 0.3.2)

**License** GPL (>= 2)

**Copyright** Dartmouth College

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-09-07 16:50:02 UTC

## R topics documented:


VAM-package . . . . . 2
createGeneSetCollection . . . . . 2
vam . . . . . 3
vamForCollection . . . . . 5
vamForSeurat . . . . . 6

**Index** **8**

VAM-package *Variance-Adjusted Mahalanobis*

## Description

Implementation of Variance-adjusted Mahalanobis (VAM), a method for cell-specific gene set scoring of scRNA-seq data.

## Details

| | |
|---|---|
| Package: | VAM |
| Type: | Package |
| Version: | 1.0.0 |
| Date: | 2021 |
| License: | GPL-2 |

## Note

This work was supported by the National Institutes of Health grants K01LM012426, R21CA253408, P20GM130454 and P30CA023108.

## Author(s)

H. Robert Frost

## References

- Frost, H. R. (2020). Variance-adjusted Mahalanobis (VAM): a fast and accurate method for cell-specific gene set scoring. biorXiv e-prints. doi: https://doi.org/10.1101/2020.02.18.954321

createGeneSetCollection

*Utility function to help create gene set collection list object*

## Description

Utility function that creates a gene set collection list in the format required by vamForCollection() given the gene IDs measured in the expression matrix and a list of gene sets as defined by the IDs of the member genes.

## Usage

```
createGeneSetCollection(gene.ids, gene.set.collection, min.size=1, max.size)
```

## Arguments

| | |
|---|---|
| gene.ids | Vector of gene IDs. This should correspond to the genes measured in the gene expression data. |
| gene.set.collection | List of gene sets where each element in the list corresponds to a gene set and the list element is a vector of gene IDs. List names are gene set names. Must contain at least one gene set. |
| min.size | Minimum gene set size after filtering out genes not in the gene.ids vector. Gene sets whose post-filtering size is below this are removed from the final collection list. Default is 1 and cannot be set to less than 1. |
| max.size | Maximum gene set size after filtering out genes not in the gene.ids vector. Gene sets whose post-filtering size is above this are removed from the final collection list. If not specified, no filtering is performed. |

## Value

Version of the input gene.set.collection list where gene IDs have been replaced by position indices, genes not present in the gene.ids vector have been removed and gene sets failing the min/max size constraints have been removed.

## See Also

vam

## Examples

```
# Create a collection with two sets defined over 3 genes
createGeneSetCollection(gene.ids=c("A", "B", "C"),
    gene.set.collection = list(set1=c("A", "B"), set2=c("B", "C")),
    min.size=2, max.size=3)
```

---

# vam *Variance-adjusted Mahalanobis (VAM) algorithm*

---

## Description

Implementation of the Variance-adjusted Mahalanobis (VAM) method, which computes distance statistics and one-sided p-values for all cells in the specified single cell gene expression matrix. This matrix should reflect the subset of the full expression profile that corresponds to a single gene set. The p-values will be computed using either a chi-square distribution, a non-central chi-square distribution or gamma distribution as controlled by the center and gamma arguments for the one-sided alternative hypothesis that the expression values in the cell are further from the mean (center=T) or origin (center=F) than expected under the null of uncorrelated technical noise, i.e., gene expression variance is purely technical and all genes are uncorrelated.

## Usage

```
vam(gene.expr, tech.var.prop, gene.weights, center=FALSE, gamma=TRUE)
```

## Arguments

| | |
|---|---|
| gene.expr | An n x p matrix of gene expression values for n cells and p genes. |
| tech.var.prop | Vector of technical variance proportions for each of the p genes. If specified, the Mahalanobis distance will be computed using a diagonal covariance matrix generated using these proportions. If not specified, the Mahalanobis distances will be computed using a diagonal covariance matrix generated from the sample variances. |
| gene.weights | Optional vector of gene weights. If specified, weights must be > 0. The weights are used to adjust the gene variance values included in the computation of the modified Mahalanobis distances. Specifically, the gene variance is divided by the gene weight. This adjustment means that large weights will increase the influence of a given gene in the computation of the modified Mahalanobis distance. |
| center | If true, will mean center the values in the computation of the Mahalanobis statistic. If false, will compute the Mahalanobis distance from the origin. Default is F. |
| gamma | If true, will fit a gamma distribution to the non-zero squared Mahalanobis distances computed from a row-permuted version of gene.expr. The estimated gamma distribution will be used to compute a one-sided p-value for each cell. If false, will compute the p-value using the standard chi-square approximation for the squared Mahalanobis distance (or non-central if center=F). Default is T. |

## Value

A data.frame with the following elements (row names will match row names from gene.expr):

- "cdf.value": 1 minus the one-sided p-values computed from the squared adjusted Mahalanobis distances.
- "distance.sq": The squared adjusted Mahalanobis distances for the n cells.

## See Also

vamForCollection,vamForSeurat

## Examples

```
# Simulate Poisson expression data for 10 genes and 10 cells
gene.expr=matrix(rpois(100, lambda=2), nrow=10)
# Simulate technical variance proportions
tech.var.prop=runif(10)
# Execute VAM to compute scores for the 10 genes on each cell
vam(gene.expr=gene.expr, tech.var.prop=tech.var.prop)
# Create weights that prioritize the first 5 genes
gene.weights = c(rep(2,5), rep(1,5))
```

```
# Execute VAM using the weights
vam(gene.expr=gene.expr, tech.var.prop=tech.var.prop,
 gene.weights=gene.weights)
```

---

vamForCollection *VAM method for multiple gene sets*

---

## Description

Executes the Variance-adjusted Mahalanobis (VAM) method (vam) on multiple gene sets, i.e., a gene set collection.

## Usage

```
vamForCollection(gene.expr, gene.set.collection, tech.var.prop,
    gene.weights, center=FALSE, gamma=TRUE)
```

## Arguments

| | |
|---|---|
| gene.expr | An n x p matrix of gene expression values for n cells and p genes. |
| gene.set.collection | List of m gene sets for which scores are computed. Each element in the list corresponds to a gene set and the list element is a vector of indices for the genes in the set. The index value is defined relative to the order of genes in the gene.expr matrix. Gene set names should be specified as list names. |
| tech.var.prop | See description in vam |
| gene.weights | See description in vam. If specified as a single vector of weights, weights must be specified for all p genes and the same weights are used for all gene sets. To use different weights for each set, specify as a list of the same length as the gene.set.collection list. In this case, each list element should be a vector of gene weights of the same length as the size of the corresponding gene set. |
| center | See description in vam |
| gamma | See description in vam |

## Value

A list containing two elements:

- "cdf.value": n x m matrix of 1 minus the one-sided p-values for the m gene sets and n cells.
- "distance.sq": n x m matrix of squared adjusted Mahalanobis distances for the m gene sets and n cells.

## See Also

vam,vamForSeurat

## Examples

```
# Simulate Poisson expression data for 10 genes and 10 cells
gene.expr=matrix(rpois(100, lambda=2), nrow=10)
# Simulate technical variance proportions
tech.var.prop=runif(10)
# Define a collection with two disjoint sets that span the 10 genes
collection=list(set1=1:5, set2=6:10)
# Execute VAM on both sets using default values for center and gamma
vamForCollection(gene.expr=gene.expr, gene.set.collection=collection,
    tech.var.prop=tech.var.prop)
# Create weights that prioritize the first 2 genes for the first set
# and the last 2 genes for the second set
gene.weights = list(c(2,2,1,1,1),c(1,1,1,2,2))
# Execute VAM using the weights
vamForCollection(gene.expr=gene.expr, gene.set.collection=collection,
    tech.var.prop=tech.var.prop, gene.weights=gene.weights)
```

---

`vamForSeurat` *VAM wrapper for scRNA-seq data processed using the Seurat framework*

---

## Description

Executes the Variance-adjusted Mahalanobis (VAM) method (`vamForCollection`) on normalized scRNA-seq data stored in a Seurat object. If the Seurat `NormalizeData` method was used for normalization, the technical variance of each gene is computed as the proportion of technical variance (from `FindVariableFeatures`) multiplied by the variance of the normalized counts. If `SCTransform` was used for normalization, the technical variance for each gene is set to 1 (the normalized counts output by `SCTransform` should have variance 1 if there is only technical variation).

## Usage

```
vamForSeurat(seurat.data, gene.weights, gene.set.collection,
 center=FALSE, gamma=TRUE, sample.cov=FALSE, return.dist=FALSE)
```

## Arguments

| | |
|---|---|
| `seurat.data` | The Seurat object that holds the scRNA-seq data. Assumes normalization has already been performed. |
| `gene.weights` | See description in `vamForCollection` |
| `gene.set.collection` | List of m gene sets for which scores are computed. Each element in the list corresponds to a gene set and the list element is a vector of indices for the genes in the set. The index value is defined relative to the order of genes in the relevant `seurat.data` Assay object. Gene set names should be specified as list names. |
| `center` | See description in `vam` |
| `gamma` | See description in `vam` |

`sample.cov` If true, will use the a diagonal covariance matrix generated from the sample variances to compute the squared adjusted Mahalanobis distances (this is equivalent to not specifying `tech.var` for the `vam` method). If false (default), will use the technical variances as determined based on the type of Seurat normalization.

`return.dist` If true, will return the squared adjusted Mahalanobis distances in a new Assay object called "VAM.dist". Default is F.

## Value

Updated Seurat object that hold the VAM results in one or two new Assay objects:

- If `return.dist` is true, the matrix of squared adjusted Mahalanobis distances will be stored in new Assay object called "VAM.dist".
- The matrix of CDF values (1 minus the one-sided p-values) will be stored in new Assay object called "VAM.cdf".

## See Also

`vam`,`vamForCollection`

## Examples

```
 # Only run example code if Seurat package is available
if (requireNamespace("Seurat", quietly=TRUE) & requireNamespace("SeuratObject", quietly=TRUE)) {
     # Define a collection with one gene set for the first 10 genes
     collection=list(set1=1:10)
     # Execute on the pbmc_small scRNA-seq data set included with SeuratObject
     # See vignettes for more detailed Seurat examples
     vamForSeurat(seurat.data=SeuratObject::pbmc_small,
         gene.set.collection=collection)
 }
```

# Index

∗ **file**
createGeneSetCollection, 2
vam, 3
vamForCollection, 5
vamForSeurat, 6
∗ **package**
VAM-package, 2

createGeneSetCollection, 2

vam, *3*, 3, *5–7*
VAM-package, 2
vamForCollection, *4*, 5, *6, 7*
vamForSeurat, *4, 5*, 6