Package 'flowVS'

November 13, 2026

Type Package

Title Variance stabilization in flow cytometry (and microarrays)

Version 1.45.0

Date 2021-04-02

Author Ariful Azad

Maintainer Ariful Azad <azad@iu.edu>

Description Per-channel variance stabilization from a collection of flow cytometry samples by Bertlett test for homogeneity of variances. The approach is applicable to microarrays data as well.

License Artistic-2.0

LazyLoad yes

VignetteBuilder knitr

Depends R (>= 3.2), methods, flowCore, flowViz, flowStats

Suggests knitr, vsn,

biocViews ImmunoOncology, FlowCytometry, CellBasedAssays, Microarray

git_url https://git.bioconductor.org/packages/flowVS

git_branch devel

git_last_commit 3a5208d

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Contents

flowVS-package 2
estParamFlowVS 2
HD 3
lymphs 4
microVS 5
plotMeanSd 6
transFlowVS 7

Index **9**

`flowVS-package` *flowVS: Variance stabilization in flow cytometry (and microarrays).*

Description

Please see the vignette.

Author(s)

Ariful Azad <azad@lbl.gov>

References

Ariful Azad, Bartek Rajwa, and Alex Pothen (2015), "flowVS: Channel-Specific Variance Stabilization in Flow Cytometry", BMC Bioinformatics, vol 17, pp 1-14, 2016.

See Also

`transFlowVS`, `microVS`

`estParamFlowVS` *Estimate optimum parameters for per-channel within-population variance stabilization.*

Description

This function estimates the variance stabilizing cofactors, one for each channel for the entire dataset. When a fluorescence channel `z` is transformed by asinh transformation with the optimum cofactor for `z`, the within-population variances of populations from all samples in the channel `z` are approximately stabilized.

Usage

```
estParamFlowVS(fs, channels)
```

Arguments

`fs`	A flowSet containing a collection of flow cyometry samples.
`channels`	A character vector identifying the channels/dimensions to be transformed. If any entry in this vector is not present in the flowSet, the function returns with an error.

Details

Let `z` be a fluorescence channel (column of a flowFrame). We consider transforming `z` by asinh transformation such that after transformation we obtain the tranformed channel asinh(`z/c`), where `c` is a normalizing cofactor.

The `estParamFlowVS` function estimates cofactors, one for each channel for the entire dataset such that the within-population variance is stabilized in each fluorescence channel. When a fluorescence channel `z` is transformed by asinh transformation with the optimum cofactor for `z`, the within-population variances of populations from all samples in the channel `z` are approximately stabilized.

Value

estParamFlowVS returns a numeric vector representing the optimum cofactors for the requested channels. The optimum cofactor for the input channels[i] is stored in the ith entry of the returned vector.

Author(s)

Ariful Azad

References

Ariful Azad, Bartek Rajwa, and Alex Pothen (2015), "flowVS: Channel-Specific Variance Stabilization in Flow Cytometry", BMC Bioinformatics, vol 17, pp 1-14, 2016.

See Also

transFlowVS

Examples

```
data(HD)
## identify optimum cofactor for CD3 and CD4 channels (from five samples)

cofactors = estParamFlowVS(HD[1:5],channels=c('CD3','CD4'))
# See detail examples in the documentation of the transFlowVS function.
```

HD *Sample flow cytometry data from healthy individuals*

Description

A flowSet cotaining 12 flow cytometry samples from three healthy individuals "A", "C", and "D". From each individual, the samples were drawn on two different days and two technical replicates were created from each sample (i.e., 3 x 2 x 2 = 12 samples). Each HD sample was stained using labeled antibodies against CD45, CD3, CD4, CD8, and CD19 protein markers. Here, an HD sample "C_4_2" means that it is collected on day 4 from individual "C" and it is the second replicate on that day. We have identified lymphocytes in each sample of the HD dataset and apply the subsequent analysis on lymphocytes.

Usage

```
data(HD)
```

Value

A flowSet containing 12 flowFrames. There are 3 subject groups with 4 samples each (2 days and 2 technical replicates per day).

References

Ariful Azad, Arif Khan, Bartek Rajwa, Saumyadipta Pyne, and Alex Pothen, "Classifying immunophenotypes with templates from flow cytometry", In Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (ACM BCB), 2013.

`lymphs` *Identify lymphocyte cells from a flow cytometry sample.*

Description

Identify and retain lymphocytes from a flow cytometry sample based on the forward and side scatters.

Usage

```
lymphs(ff, lymph.boundary, fsc, ssc, plot=FALSE)
```

Arguments

`ff`	A flowFrame containing a flow cyometry sample.
`lymph.boundary`	A list denoting an approximate rectangular boundary for lymphocytes. The first element of the list represents the lower and upper limit of forward scatter (FSC), and the 2nd element represents the lower and upper limit of side scatter (SSC). Example: list("FSC"=c(180000, 500000),"SSC"=c(0, 180000)).
`fsc`	name (or numeric index) of the forward scatter channel.
`ssc`	name (or numeric index) of the side scatter channel.
`plot`	true/false. If true then plots the rectangular and elliptical gates for the lymphocytes.

Details

At first a rectangular gate is created based on the `lymph.boundary`. Then the `norm2Filter` function is used to identify lymphocytes.

Value

`lymphs` returns a new flowFrame containing the lymphocytes.

Author(s)

Ariful Azad

References

Ariful Azad, Bartek Rajwa, and Alex Pothen (2015), "flowVS: Channel-Specific Variance Stabilization in Flow Cytometry", BMC Bioinformatics, vol 17, pp 1-14, 2016.

See Also

`estParamFlowVS`

Examples

```
library(flowStats)
data(ITN)
# identify lymphocytes
ITN.lymphs = lymphs(ITN[[1]], list("FS"=c(200, 600),"SS"=c(0, 400)), "FSC", "SSC",TRUE)
```

`microVS` *Variance stabilization for microarray data.*

Description

Variance-stabilizing inverse hyperbolic sine (`asinh`) transformation for microarray data.

Usage

```
microVS(data, cfLow=0, cfHigh=10, frac=1)
```

Arguments

`data`	The microarray data in a Matrix.
`cfLow`	lowest possible value of cofactor (log scale).
`cfHigh`	highest possible value of cofactor (log scale).
`frac`	fraction of differentially expressed genes used in variance stabilization (0< frac <=1).

Details

This function transforms a microarray data matrix `z` by `asinh(z/c)` transformation where `c` is a normalizing cofactor. The cofactor is searched in the range `[cfLow, cfHigh]` and an optimum cofactor is obtained for which the transformed data is variance stabilized. The optimum cofactor is obtained by minimizing Bartlett's test statistics for homogeneity of variance. If the parameter `frac` is less then one, a fraction of differentially expressed genes are used in estimating the cofactor.

Value

`microVS` returns a matrix of the variance-stabilizing microarray data.

Author(s)

Ariful Azad

References

Ariful Azad, Bartek Rajwa, and Alex Pothen (2015), "flowVS: Channel-Specific Variance Stabilization in Flow Cytometry", BMC Bioinformatics, vol 17, pp 1-14, 2016.

Examples

```
# stabilize variance of the Kidney microarray data from the vsn package
library(vsn)
data(kidney)
kidney.t = microVS(exprs(kidney))
plotMeanSd(kidney.t)
```

`plotMeanSd` *Plot row standard deviations versus row means (modified from vsn package)*

Description

Plot row standard deviations versus row means of a data matrix.

Usage

```
plotMeanSd(x,
           ranks = TRUE,
           xlab  = ifelse(ranks, "Rank of means (ascending order)", "mean"),
           ylab  = "Standard deviation",
           pch   = ".",
           plot  = TRUE,
           ...)
```

Arguments

`x`	An object of class `matrix`
`ranks`	Logical, indicating whether the x-axis (means) should be plotted on the original scale (`FALSE`) or on the rank scale (`TRUE`). The latter distributes the data more evenly along the x-axis and allows a better visual assessment of the standard deviation as a function of the mean.
`xlab`	Character, label for the x-axis.
`ylab`	Character, label for the y-axis.
`pch`	Plot symbol.
`plot`	Logical. If `TRUE` (default), a plot is produced. Calling the function with `plot=FALSE` can be useful if only its return value is of interest.
`...`	Further arguments that get passed to plot.default.

Details

Standard deviation and mean are calculated row-wise from the expression matrix (in) `x`. The scatterplot of these versus each other allows to visually verify whether there is a dependence of the standard deviation (or variance) on the mean. The red dots depict the running median estimator (window-width 10%). If there is no variance-mean dependence, then the line formed by the red dots should be approximately horizontal.

Value

A named list with four components: its elements px and py are the x- and y-coordinates of the individual data points in the plot; its first and second element are the x-coordinates and values of the running median estimator (the red dots in the plot). Depending on the value of plot, the method can also have a side effect, which is to create a plot on the active graphics device.

Examples

```
library(vsn)
data(kidney)
kidney.t = microVS(exprs(kidney))
plotMeanSd(kidney.t)
```

---

transFlowVS *Transform a flowSet by asinh transformation.*

Description

This function transforms a flowSet by asinh transformation with the cofactors passed on to the function. The optimum cofactors that stabilize within-population variances in different fluorescence channels are estimated beforehand and passed to this function for data transformation.

Usage

```
transFlowVS(fs, channels, cofactors)
```

Arguments

fs	A flowSet containing a collection of flow cyometry samples.
channels	A character vector identifying the channels/dimensions to be transformed. If any entry in this vector is not present in the flowSet, the function returns with an error.
cofactors	A numeric vector. cofactors[i] is used with asinh function to transform the column with name specified by channels[i].

Details

This function transforms a flowSet by asinh transformation with selected cofactors. The column with name channels[i] of every flowFrame of the input flowSet is transformed by asinh transformation with cofactors[i]. For example, let z_ij be the ith column of jth flowFrame in the input flowSet fs. Then after transformation z_ij would be converted to asinh(z_ij/cofactors[i]).

For variance stabiilzation, the optimum cofactors that stabilize within-population variances in different fluorescence channels are estimated beforehand and passed to this function for data transformation. Variance stabilizing cofactors can be estimatd by the estParamFlowVS function.

Value

transFlowVS returns a new flowSet with the transformed channels.

Author(s)

Ariful Azad

References

Ariful Azad, Bartek Rajwa, and Alex Pothen (2015), "flowVS: Channel-Specific Variance Stabilization in Flow Cytometry", BMC Bioinformatics, vol 17, pp 1-14, 2016.

See Also

estParamFlowVS

Examples

```
## ------------------------------------------------
## Example 1: Healthy data from flowVS package
## ------------------------------------------------
data(HD)

## identify optimum cofactor for CD3 and CD4 channels (from five samples)
cofactors = estParamFlowVS(HD[1:5],channels=c('CD3','CD4'))
## transform CD3 and CD4 channels in all samples
HD.VS = transFlowVS(HD, c('CD3','CD4'), cofactors)
densityplot(~CD3+CD4, HD.VS, main="Transfromed CD3 and CD4 channels in HD data")

## ------------------------------------------------
## Example 2: ITN data from flowStats package
## ------------------------------------------------

library(flowStats)
data(ITN)
# identify lymphocytes
ITN.lymphs = fsApply(ITN,lymphs, list("FS"=c(200, 600),"SS"=c(0, 400)), "FSC", "SSC",FALSE)
## identify optimum cofactor for CD3 and CD4 channels (from five samples)
cofactors = estParamFlowVS(ITN.lymphs[1:5],channels=c('CD3', 'CD4'))
## transform CD3 and CD4 channels in all samples
ITN.VS = transFlowVS(ITN.lymphs, c('CD3','CD4'), cofactors)
densityplot(~CD3+CD4, ITN.VS, main="Transfromed CD3 and CD4 channels in ITN data")
```

Index

∗ **datasets**
HD, 3
∗ **hplot**
plotMeanSd, 6
∗ **methods**
plotMeanSd, 6
∗ **transformation**
estParamFlowVS, 2
flowVS-package, 2
lymphs, 4
microVS, 5
transFlowVS, 7
∗ **variance stabilization**
estParamFlowVS, 2
flowVS-package, 2
lymphs, 4
microVS, 5
transFlowVS, 7

estParamFlowVS, 2, *4*, *7*, *8*

flowFrames, *3*
flowSet, *3*
flowVS *(*flowVS-package*)*, 2
flowVS-package, 2

HD, 3

lymphs, 4

matrix, *6*
microVS, *2*, 5

plotMeanSd, 6

transFlowVS, *2*, *3*, 7