Package ‘SamSPECTRAL’

July 21, 2025

Type Package

Title Identifies cell population in flow cytometry data

Version 1.63.0

Date 2018-05-31

Author Habil Zare and Parisa Shooshtari

Maintainer Habil `<zare@u.washington.edu>`

Depends R (>= 3.3.3)

Imports methods

Description Samples large data such that spectral clustering is possible while preserving density information in edge weights. More specifically, given a matrix of coordinates as input, SamSPECTRAL first builds the communities to sample the data points. Then, it builds a graph and after weighting the edges by conductance computation, the graph is passed to a classic spectral clustering algorithm to find the spectral clusters. The last stage of SamSPECTRAL is to combine the spectral clusters. The resulting ``connected components'' estimate biological cell populations in the data. See the vignette for more details on how to use this package, some illustrations, and simple examples.

License GPL (>= 2)

LazyLoad yes

biocViews FlowCytometry, CellBiology, Clustering, Cancer, FlowCytometry, StemCells, HIV, ImmunoOncology

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

git_branch devel

git_last_commit 83e19f9

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-20

Contents

`SamSPECTRAL-package` *Identifying cell populations in flow cytometry data.*

Description

Using a faithful sampling procedure, SamSPECTRAL reduces the size of data points such that applying spectral clustering algorithm on large data such as flow cytometry is possible. Before running the spectral clustering algorithm, it uses potential theory to define similarity between sampled points.

Details

Package:	SamSPECTRAL
Type:	Package
Version:	1.0
Date:	2009-08-31
License:	GPL-2
LazyLoad:	yes

The main function is SamSPECTRAL. It can be loaded using the command library(SamSPECTRAL) in R. Some parameters should be set properly including: dimensions, normal.sigma and separation.factor. These parameters can be adjusted for a data set by running the algorithm on some samples of that data set. (Normally, 2 or 3 samples are sufficient). Then the function SamSPECTRAL() can be applied to all samples in the data set to identify cell populations in each sample data.

Author(s)

Habil Zare and Parisa Shooshtari

Maintainer: Habil Zare <hzare@bccrc.ca>

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`, `Building_Communities`, `Conductance_Calculation`, `Civilized_Spectral_Clustering`, `Connecting`

Examples

```
 ## Not run:
     library(SamSPECTRAL)

    # Reading data file which has been transformed using log transform
     data(small_data)
  full <- small

   L <- SamSPECTRAL(data.points=full,dimensions=c(1,2,3), normal.sigma = 200, separation.factor = 0.39)

     plot(full, pch='.', col= L)

## End(Not run)
```

`Building_Communities` *Builds the communities from the set of all data points.*

Description

Some sample points are picked up and the points close to each sample point are considered as members of that community.

Usage

```
Building_Communities(full, m=3000, space.length=1, community.weakness.threshold=1, talk=TRUE, do.s
      replace.inf.with.extremum=TRUE)
```

Arguments

`full`	The matrix containing the coordinates of all data points.
`m`	An integer determining upper and lower bounds on the final number of sample points which will be in range `.95*m/2` and 2 `1.1*m`
`space.length`	An estimate for the length of a cube that is assumed to contain all data points.
`community.weakness.threshold`	The communities with number of members less than this threshold will be ignored. Normally, setting it to 1 is reasonable.
`talk`	A boolean flag with default value TRUE. Setting it to FALSE will keep running the procedure quite with no messages.
`do.sampling`	A boolean flag with default value TRUE. If set to FALSE, the sampling stage will be ignored by picking up all the data points.

`replace.inf.with.extremum`
: If TRUE, the Inf and -Inf values will be replaced by maximum and minimum of data in each direction.

Value

Returns a society which is a list of communities.

Author(s)

Habil Zare and Parisa Shooshtari

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`,`check.SamSPECTRAL.input`

Examples

```
  ## Not run: 
     library(SamSPECTRAL)

    # Reading data file which has been transformed using log transform
     data(small_data)
  full <- small

  # Parameters:
  m <- 3000; ns <- 200; sl <- 3; cwt <-1

     # Sample the data and build the communities
   society <- Building_Communities(full=full,m=m, space.length=sl, community.weakness.threshold=cwt)

  # Ploting the representatives:
  plot(full[society$representatives,])


## End(Not run)
```

`check.SamSPECTRAL.input`

Checks the input to SamSPECTRAL.

Description

The input to SamSPECTRAL should be a numeric matrix WITHOUT any NA, NaN, and +/- Inf. This function checks the input matrix and prodeuces an error for an inappropriate input.

Usage

```
check.SamSPECTRAL.input(data.points,dimensions=1:ncol(data.points),replace.inf.with.extremum=FAL
```

Arguments

`data.points`	A matrix that contains coordinates of the data points.
`dimensions`	A vector that determines which dimension of the data point matrix are chosen for investigation.
`replace.inf.with.extremum`	If TRUE, the Inf and -Inf values will be replaced by maximum and minimum of data in each direction.

Value

Returns a list with the following entries:

`data.matrix`	The data with infinite elements fixed if `replace.inf.with.extremum=TRUE`
`dimensions`	All the checked dimensions.
`infinite`	Will be TRUE if data contained infinite entries.

Author(s)

Habil Zare

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`, `Building_Communities`, `Conductance_Calculation`, `Connecting`

Examples

```
 ## Not run: 
     library(SamSPECTRAL)

    # Reading data file which has been transformed using log transform
     data(small_data)
  full <- small

    checked <- check.SamSPECTRAL.input(data.points=full,dimensions=c(1,2,3),replace.inf.with.extremum=TRUE)

     plot(checked$data.matrix, pch='.')

## End(Not run)
```

`Civilized_Spectral_Clustering`

Runs the spectral clustering algorithm on the sample points.

Description

The representatives of communities are considered as the vertices of a graph. Assuming the edges have been weighted according to the equivalent conductance between them, this function runs the classic spectral clustering on the graph.

Usage

```
Civilized_Spectral_Clustering(full, maximum.number.of.clusters, society, conductance,
   iterations=200, number.of.clusters="NA",
   k.for_kmeans="NA", minimum.eigenvalue="NA", minimum.degree=0,
   eigenvalues.num =NA, talk=TRUE,stabilizer=1000, one.line=FALSE,
   replace.inf.with.extremum=TRUE)
```

Arguments

`full`	The matrix containing the coordinates of all data points.
`maximum.number.of.clusters`	An integer used to automatically estimate the number of clusters by fitting 2 regression lines on the eigen values curve.
`number.of.clusters`	The default value is "NA" which leads to computating the number of spectral clusters automatically, otherwise this number will determine the number of spectral clusters.
`k.for_kmeans`	The number of clusters for running kmeans algorithm in spectral clustering. The default value of "NA" leads to automatic estimation based on eigen values curve.
`minimum.eigenvalue`	If not "NA", the number of spectral clusters will be determined such that corresponding eigenvalues are larger than this threshold.
`minimum.degree`	If a node in the graph has total edge sum less than this threshold, it will be considered as an isolated community.
`society`	The list of communities.
`conductance`	A matrix in which each entry is the conductance between two communities.
`iterations`	Number of iterations for the k-means algorithm used by the spectral procedure. 200 is an appropriate value.
`talk`	A boolean flag with default value TRUE. Setting it to FALSE will keep running the procedure quite with no messages.
`eigenvalues.num`	An integer with default value NA which prevents ploting the curve of eigenvalues. Otherwise, they will be ploted upto this number.
`stabilizer`	The larger this integer is, the final results will be more stable because the underlying kmeans will restart many more times.
`one.line`	If TRUE, the number of spectral clusters are estimated by fitting 1 line to the eigen values curve. Otherwise 2 lines are fitted.

`replace.inf.with.extremum`
: If TRUE, the Inf and -Inf values will be replaced by maximum and minimum of data in each direction.

Value

A `ClusteringResult` class object with the following slots,

The k'th element of this list is a vector containing the labels as result of clustering to k parts.

labels.for_num.of.clusters **number.of.clusters** A list containing the desired cluster numbers.

eigen.space The eigen vectors and eigen values of the normalized adjacency matrix computed by the eigen() function for spectral clustering.

Author(s)

Habil Zare, Nima Aghaeepour and Parisa Shooshtari

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`,`check.SamSPECTRAL.input`

Examples

```
## Not run:
   library(SamSPECTRAL)

  # Reading data file which has been transformed using log transform
   data(small_data)
full <- small

# Parameters:
m <- 3000; ns <- 200; sl <- 3; cwt <-1; precision <- 6; mnc <-30

   # Sample the data and build the communities
 society <- Building_Communities(full=full,m=m, space.length=sl, community.weakness.threshold=cwt)


   # Compute conductance between communities
 conductance <- Conductance_Calculation(full=full, normal.sigma=ns, space.length=sl, society=society, prec

   # Use spectral clustering to cluster the data
# First example:
 clust_result <- Civilized_Spectral_Clustering(full=full, maximum.number.of.clusters=mnc, society=society
   number.of.clusters <- clust_result@number.of.clusters
   labels.for_num.of.clusters <- clust_result@labels.for_num.of.clusters
L <- labels.for_num.of.clusters[[number.of.clusters]]
   # plot(full, pch='.', col= L)


# Second example:
number.of.clusters <- c(35,20)
```

```
  # This is faster than runnig Civilized_Spectral_Clustering() twice because the eigen space is not needed to be
  clust_result.not.automatic <-
  Civilized_Spectral_Clustering(full=full, society=society, conductance=conductance, number.of.clusters =num
     labels.for_num.of.clusters <- clust_result.not.automatic@labels.for_num.of.clusters
  L35 <- labels.for_num.of.clusters[[35]]
  L20 <- labels.for_num.of.clusters[[20]]
     # plot(full, pch='.', col= L35)

## End(Not run)
```

Conductance_Calculation

Computes the conductance between communities.

Description

For each two communities, the conductance between their members is summed up and the result is returned as the conductance between the two communities.

Usage

```
Conductance_Calculation(full, normal.sigma, space.length, society, precision, talk=TRUE, beta=4,
          replace.inf.with.extremum=TRUE)
```

Arguments

`full`	The matrix containing the coordinates of all data points.
`normal.sigma`	The scaling parameter, the larger it is the algorithm will find smaller clusters.
`space.length`	An estimate for the length of a cube that is assumed to contain all data points.
`society`	The list of communities.
`precision`	Determines the precision of computations. Setting it to 6 will work and increasing it does not improve results.
`talk`	A boolean flag with default value TRUE. Setting it to FALSE will keep running the procedure quite with no messages.
`beta`	A parameter with default value 4 which must NOT be changed except for huge samples with more than 100,000 data points or for developmental purposes. Setting beta to zero will reduce computational time by applying the following approximation to the conductance calculation step. For each two community, the conductance will be the conductance between their representatives times their sizes.
`replace.inf.with.extremum`	If TRUE, the Inf and -Inf values will be replaced by maximum and minimum of data in each direction.

Value

Returns a matrix in which each entry is the conductance between two communities.

Author(s)

Habil Zare and Parisa Shooshtari

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`,`check.SamSPECTRAL.input`

Examples

```
 ## Not run: 
     library(SamSPECTRAL)

    # Reading data file which has been transformed using log transform
     data(small_data)
  full <- small

  # Parameters:
  m <- 3000; ns <- 200; sl <- 3; cwt <-1; precision <- 6

     # Sample the data and build the communities
   society <- Building_Communities(full=full,m=m, space.length=sl, community.weakness.threshold=cwt)


     # Compute conductance between communities
   conductance <- Conductance_Calculation(full=full, normal.sigma=ns, space.length=sl, society=society, prec


## End(Not run)
```

`Connecting` *Combines the spectral clusters to build the connected components.*

Description

Considering some biological criterion based on density, the clusters which are identified by spectral clustering are combined to estimate biological populations.

Usage

```
Connecting(full, society,conductance, number.of.clusters, labels.for_num.of.clusters, separation.
```

Arguments

`full`	The matrix containing the coordinates of all data points.
`society`	The list of communities.
`conductance`	A matrix in which each entry is the conductance between two communities.

`number.of.clusters`	A list containing the desired cluster numbers.
`labels.for_num.of.clusters`	The k'th element of this list, is a vector containing the labels as result of clustering to k parts.
`separation.factor`	This threshold controls to what extend clusters should be combined or kept separate.
`talk`	A boolean flag with default value TRUE. Setting it to FALSE will keep running the procedure quite with no messages.

Details

A hint for setting `separation.factor`: While `separation.factor`=0.7 is normally an appropriate value for many datasets, for others some value in range 0.3 to 1.2 may produce better results depending on what populations are of particular interest.

Value

Returns two objects: 1) label, a vector containing the labels that determines to which component each data point belongs. 2) clusters.graph, the max.conductance matrix that describes the original graph based on clusters.

Author(s)

Habil Zare and Parisa Shooshtari

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`

Examples

```
## Not run:
    library(SamSPECTRAL)

   # Reading data file which has been transformed using log transform
    data(small_data)
 full <- small

 # Parameters:
 m <- 3000; ns <- 200; sl <- 3; cwt <-1; precision <- 6; mnc <-30

    # Sample the data and build the communities
  society <- Building_Communities(full=full,m=m, space.length=sl, community.weakness.threshold=cwt)


    # Compute conductance between communities
  conductance <- Conductance_Calculation(full=full, normal.sigma=ns, space.length=sl, society=society, prec
```

```
        # Use spectral clustering to cluster the data
      clust_result <- Civilized_Spectral_Clustering(full=full, maximum.number.of.clusters=mnc, society=society
        number.of.clusters <- clust_result@number.of.clusters
        labels.for_num.of.clusters <- clust_result@labels.for_num.of.clusters
    L <- labels.for_num.of.clusters[[number.of.clusters]]
        # plot(full, pch='.', col= L)


        # Connect components
      L <- Connecting(full=full, society=society, conductance=conductance, number.of.clusters=number.of.cluster
       labels.for_num.of.clusters=labels.for_num.of.clusters, separation.factor=0.39)

        plot(full, pch='.', col= L)


## End(Not run)
```

`eigen.values.10` *Eigenvalues for building the SamSPECTRAL vignette.*

Description

This file contains a vector that represents the eigenvalues of the small example if normal.sigma=10.

Usage

```
data(eigen.values.10)
```

Format

This RData contains a vector.

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

Examples

```
data(eigen.values.10)

    plot(eigen.values.10)
```

`eigen.values.1000` *Eigenvalues for building the SamSPECTRAL vignette.*

Description

This file contains a vector that represents the eigenvalues of the small example if normal.sigma=1000.

Usage

```
data(eigen.values.1000)
```

Format

This RData contains a vector.

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

Examples

```
data(eigen.values.1000)

    plot(eigen.values.1000)
```

`kneepointDetection` *Fits 2 regression lines to data to estimate the knee (or elbow) point.*

Description

With an appropriate sigma value, the curve of eigenvalues has a knee point shape. The bending point is a good estimate for the number of informative spectral clusters because the eigenvalues above the corresponding threshold can reasonably be assumed to be close to 1. This function estimate the knee point by fitting 2 lines using linear regression.

Usage

```
  kneepointDetection(vect, PlotFlag=FALSE)
```

Arguments

`vect`	The vector of values on which the 2 regression lines will be fitted.
`PlotFlag`	If TRUE and in unix, an animation will be produced in tmpfigs folder that shows how the best selected model in gif format.

Details

The running time is in order of minutes for 100 points. This function was borrowed from flowMeans package and for application in SamSPECTRAL package, it was customized such that the first line is always horizontal.

Value

Returns a list where MinIndex is the index of the knee point and l1 and l2 the fitted lines.

Author(s)

Nima Aghaeepour

References

Aghaeepour N., Nikolic R., Hoos HH., Brinkman RR.: Rapid cell population identification in flow cytometry data. Cytometry A, 2011, 79:6.

See Also

`changepointDetection`

Examples

```
## Data
  values <- rep(1,times=10)
  values <- c(values,(10:0)/10)

## Looks like knee point:
plot(values)

## Find the knee point:
detected <- kneepointDetection(vect=values, PlotFlag=FALSE)
print(detected$MinIndex)
##  Also, under unix, set PlotFlag=TRUE and look at animation.gif.
```

`SamSPECTRAL` *Identifies the cell populations in flow cytometry data.*

Description

Given an FCS file as input, SamSPECTRAL first builds the communities to sample the data points. Then, it builds a graph and after weighting the edges of the graph by conductance computation, it is passed to a classic spectral clustering algorithm to find the spectral clusters. The last stage of SamSPECTRAL is to combine the spectral clusters. The resulting "connected components" estimate biological cell populations in the data sample.

Usage

```
SamSPECTRAL(data.points, dimensions=1:dim(data.points)[2], normal.sigma, separation.factor,number
    talk = TRUE, precision = 6, eigenvalues.num =NA, return_only.labels=TRUE, do.sampling=TRUE, beta
      k.for_kmeans = "NA", maximum.number.of.clusters=30, m=3000,
      minimum.eigenvalue = "NA", previous.result = NULL,
    replace.inf.with.extremum=TRUE, minimum.degree=0, one.line=FALSE, doOrderLabels=TRUE)
```

Arguments

`data.points`	A matrix that contains coordinates of the data points.
`dimensions`	A vector that determines which dimension of the data point matrix are chosen for investigation.
`normal.sigma`	A scaling parameter that determines the "resolution" in the spectral clustering stage. By increasing it, more spectral clusters are identified. This can be useful when "small" population are aimed. See the user manual for a suggestion on how to set this parameter using the eigenvalue curve.
`separation.factor`	This threshold controls to what extend clusters should be combined or kept separate.Normally, an appropriate value will fall in range 0.3-2.
`number.of.clusters`	The default value is "NA" which leads to computing the number of spectral clusters automatically, otherwise it can be a vector of integers each of which determines the number of spectral clusters. The output will contain a clustering resulting from each value.
`talk`	A boolean flag with default value TRUE. Setting it to FALSE will keep running the procedure quite with no messages.
`precision`	Determines the precision of computations. Setting it to 6 will work and increasing it does not improve results.
`eigenvalues.num`	An integer with default value NA which prevents ploting the curve of eigenvalues. Otherwise, they will be ploted upto this number.
`return_only.labels`	A boolean flag with default value TRUE. If the user set it to FALSE, SamSPECTRAL function will return all the intermediate objects that are computed during the sampling, similarity calculation, spectral clustering and combining stages.
`do.sampling`	A boolean flag with default value TRUE. If set to FALSE, the sampling stage will be ignored by picking up all the data points.
`beta`	A parameter with default value 4 which must NOT be changed except for huge samples with more than 100,000 data points or for developmental purposes. Setting beta to zero will reduce computational time by applying the following approximation to the conductance calculation step. For each two community, the conductance will be the conductance between their representatives times their sizes.
`scale`	A vector the length of which is equal to the number of dimensions. The coordinates in each dimension are multiplied by the corresponding scaling factor. So, the bigger this factor is for a dimension, SamSPECTRAL will consider that dimension to be "more significant" and consequently, that dimension will be more effective in clustering.
`stabilizer`	The larger this integer is, the final results will be more stable because the underlying kmeans will restart many more times.
`k.for_kmeans`	The number of clusters for running kmeans algorithm in spectral clustering. The default value of "NA" leads to automatic estimation based on eigen values curve.
`maximum.number.of.clusters`	An integer used to automatically estimate the number of clusters by fitting 2 regression lines on the eigen values curve.

`m`	An integer determining upper and lower bounds on the final number of sample points which will be in range `.95*m/2` and 2 `1.1*m`
`minimum.eigenvalue`	If not "NA", the number of spectral clusters will be determined such that corresponding eigenvalues are larger than this threshold.
`previous.result`	If provided, the intermediate results from previous run can be passed to save on computing time while setting the parameters.
`replace.inf.with.extremum`	If TRUE, the Inf and -Inf values will be replaced by maximum and minimum of data in each direction.
`minimum.degree`	If a node in the graph has total edge sum less than this threshold, it will be considered as an isolated community.
`one.line`	If TRUE, the number of spectral clusters are estimated by fitting 1 line to the eigen values curve. Otherwise 2 lines are fitted.
`doOrderLabels`	Used for debugging. If TRUE, after connecting components, relabeling will be done such that the largest component gets label 1. If FALSE, the label of each data point will be the index of the component it belongs to (after connecting components).

Details

Hints for setting `separation.factor` and `normal.sigma`: While `separation.factor`=0.7 is normally an appropriate value for many datasets, for others some value in range 0.3 to 1.2 may produce better results depending on what populations are of particular interest. The larger `normal.sigma` is the algorithm will find smaller clusters. It can be adjusted best by considering the plot of eigenvalues as explained in the vignette.

Value

Returns a vector of labels for data points. If the input parameter return_only.labels is set to FALSE, all the objects that are computed during the intermediate will be returned including: society from sampling stage, conductance from similarity calculation, clustering_result, component.of from connecting step (the same as labels if doOrderLabels=FALSE, used for debugging), timeTaken, and sizes which is a table of size of each component.

Author(s)

Habil Zare and Parisa Shooshtari

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

See Also

`SamSPECTRAL`, `Building_Communities`, `Conductance_Calculation`, `Civilized_Spectral_Clustering`, `Connecting`,`check.SamSPECTRAL.input`

Examples

```
 ## Not run:
     library(SamSPECTRAL)

    # Reading data file which has been transformed using log transform
     data(small_data)
  full <- small

   L <- SamSPECTRAL(data.points=full,dimensions=c(1,2,3), normal.sigma = 200, separation.factor = 0.39)

     plot(full, pch='.', col= L)

## End(Not run)
```

`small` *Flow cytometry data to test SamSPECTRAL algorithm.*

Description

This FCS file is a small one used to show how to set SamSPECTRAL parameters.

Usage

```
data(small_data)
```

Format

This is an FCS file.

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

Examples

```
data(small_data)
 full <- small

    plot(full, pch='.')
```

`stmFSC` *Flow cytometry data to test SamSPECTRAL algorithm.*

Description

This FCS file is used as demo data to illustrate SamSPECTRAL capabilities in identifying cell populations.

Usage

```
data(stm)
```

Format

The is an FCS file.

References

Zare, H. and Shooshtari, P. and Gupta, A. and Brinkman R.B: Data Reduction for Spectral Clustering to Analyse High Throughput Flow Cytometry Data. BMC Bioinformatics, 2010, 11:403.

Examples

```
data(stm)
    # Read data files and transform them using log transform
 data.points <- stmFSC@exprs
 dimensions <- c(3,4,7)
 full <- log10(data.points[,dimensions])

    plot(full, pch='.')
```

Index