SamSPECTRAL

April 20, 2011

`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=
```

Arguments

`full`	The matrix containing the coordinates of all data points.
`m`	Determines an upper bound on the final number of sample points which will be in range `m` and 2 `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.

Value

Returns a society which is a list of communities.

Author(s)

Parisa Shooshtari and 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

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.t

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


## 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, conduct
eigenvalues.num =NA, talk=TRUE,stabilizer=1000)
```

Arguments

full	The matrix containing the coordinates of all data points.
maximum.number.of.clusters	This parameter is used for fitting the regression line.
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.
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.

Value

`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.

`number.of.clusters` A list containing the desired cluster numbers.

`eigen.space` The eigen vectors and eigen values of the normalized adjacency matrix computed for spectral clustering.

Author(s)

Parisa Shooshtari and 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

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.t


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

     # Use spectral clustering to cluster the data
```

```
# First example:
    clust_result <- Civilized_Spectral_Clustering(full=full, maximum.number.of.clusters=m
     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 spac
clust_result.not.automatic <-
Civilized_Spectral_Clustering(full=full, society=society, conductance=conductance, number
    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, ta
```

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.

Value

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

Author(s)

Parisa Shooshtari and 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

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.t


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


## 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.clus
```

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)

Parisa Shooshtari and 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

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.t


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

      # Use spectral clustering to cluster the data
     clust_result <- Civilized_Spectral_Clustering(full=full, maximum.number.of.clusters=m
      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.clust
 labels.for_num.of.clusters=labels.for_num.of.clusters, separation.factor=0.39)

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


## End(Not run)
```

`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)
```

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)
```

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_Clust
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

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

## End(Not run)
```

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, separat
talk = TRUE, precision = 6, eigenvalues.num =NA, return_only.labels=TRUE, do.sam
```

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 this number will determine the number of spectral clusters.
`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.

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 for sampling stage, conductance for similarity calculation, and clustering_result.

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_Clust
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, separatio

     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