Package: smfishHmrf (via r-universe)

September 12, 2024

Title Hidden Markov Random Field for Spatial Transcriptomic Data

Version 0.1

Author Qian Zhu and Guo-Cheng Yuan

Description Discovery of spatial patterns with Hidden Markov Random Field. This package is designed for spatial transcriptomic data and single molecule fluorescent in situ hybridization (FISH) data such as sequential fluorescence in situ hybridization (seqFISH) and multiplexed error-robust fluorescence in situ hybridization (MERFISH). The methods implemented in this package are described in Zhu et al. (2018) <doi:10.1038/nbt.4260>.

Maintainer Qian Zhu <zqian@jimmy.harvard.edu>

Depends R (>= 3.5.0), pracma(>= 1.9.0), fs(>= 1.2)

License GPL

URL https://bitbucket.org/qzhudfci/smfishhmrf-r/src/master/

BugReports https://bitbucket.org/qzhudfci/smfishhmrf-r/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-11-03 12:20:02 UTC

RoxygenNote 7.1.1

Imports Rdpack

RdMacros Rdpack

Contents

findDampFactor 2
seqfishplus 3
smfishHmrf 3
smfishHmrf.generate.centroid 5
smfishHmrf.generate.centroid.it 5

smfishHmrf.generate.centroid.save 7
smfishHmrf.generate.centroid.use.exist 7
smfishHmrf.hmrfem.multi 8
smfishHmrf.hmrfem.multi.it 10
smfishHmrf.hmrfem.multi.it.min 12
smfishHmrf.hmrfem.multi.save 15
smfishHmrf.read.expression 16

Index **17**

`findDampFactor` *findDampFactor*

Description

find dampening factor

Usage

```
findDampFactor(sigma, factor = 1.05, d_cutoff = 1e-60, startValue = 1e-04)
```

Arguments

`sigma`	covariance matrix
`factor`	step factor
`d_cutoff`	determinant cutoff
`startValue`	starting value to initialize the finding

Examples

```
data(seqfishplus)
k<-dim(seqfishplus$mu)[2]
damp<-array(0, c(k))
for(i in 1:k){
   di<-findDampFactor(seqfishplus$sigma[,,i], factor=1.05, d_cutoff=1e-5, startValue=0.0001)
    damp[i]<-ifelse(is.null(di), 0, di)
}
```

`seqfishplus` *SeqFISHplus dataset*

Description

Data from SeqFISH experiment on SS cortex. This is a dataset with 523 cells and the expression of about 500 spatial genes

Usage

```
data(seqfishplus)
```

Format

A list containing the following fields: y, nei, blocks, damp, mu, sigma

y gene expression matrix

nei cell adjacency matrix

blocks vertex (or cell) update order; a list of vertex colors; cells marked with the same color are updated at once

damp dampening constants (length k, the number of clusters)

mu initialization (means). Means is a (i,k) matrix

sigma initialization (sigmas). Sigmas is a (i,j,k) 3D matrix. k is cluster id. (i,j) is covariance matrix

References

Eng CL, Lawson M, Zhu Q, Dries R, Koulena N, Takei Y, Yun J, Cronin C, Karp C, Yuan G, Cai L (2019-Apr-01). "Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+." *Nature*, **568**, 235–239. doi: 10.1038/s415860191049y.

Examples

```
data(seqfishplus)
```

`smfishHmrf` *smfishHmrf: A package for running hidden markov random field on smFISH and other spatial transcriptomic datasets*

Description

A package for running hidden markov random field (Zhu et al. 2018-Dec-01) on smFISH and other spatial transcriptomic datasets.

Input

The inputs of HMRF are the following:

- Gene expression matrix
- Cell neighborhood matrix
- Initial centroids of clusters
- Number of clusters
- beta

smfishHmrf has been tested to work on seqFISH, MERFISH, starMAP, 10X Visium and other datasets. See Giotto (Dries et al. 2020) for examples of such datasets and to learn about the technologies. smfishHmrf is a general algorithm, and should probably work with other data types.

Running

The first step is to calculate initial centroids on the gene expression matrix given k (the number of clusters). The function **smfishHmrf.generate.centroid.it** is used for this purpose.

The next step is to run the HMRF algorithm given the expression matrix, and cell neighborhood matrix. The function **smfishHmrf.hmrfem.multi.it.min** is used for this purpose.

Variations

You might notice several variations of the functions:

- `smfishHmrf.hmrfem.multi.it.min`: supports multiple betas; supports file names as inputs. **This is the recommended function.**
- `smfishHmrf.hmrfem.multi.it`: supports multiple betas; supports R data structures as inputs.
- `smfishHmrf.hmrfem.multi`: supports a single beta; supports R data structures as inputs.
- Note: beta is the smoothness parameter of HMRF

Also:

- `smfishHmrf.generate.centroid.it`: supports file names as inputs. **This is the recommended function**
- `smfishHmrf.generate.centroid`: supports R matrices as inputs. Assumes input files have been read into R matrices.
- `smfishHmrf.generate.centroid.use.exist`: loads existing centroids. Assumes that centroids have been generated previously and saved to disk.

References

Zhu Q, Shah S, Dries R, Cai L, Yuan G (2018-Dec-01). "Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data." *Nature Biotechnology*, **36**, 1183–1190. doi: 10.1038/nbt.4260.

Dries R, Zhu Q, Dong R, Eng CL, Li H, Liu K, Fu Y, Zhao T, Sarkar A, Bao F, George RE, Pierson N, Cai L, Yuan G (2020). "Giotto, a toolbox for integrative analysis and visualization of spatial expression data." *bioRxiv*. doi: 10.1101/701680.

smfishHmrf.generate.centroid

Generate cluster centroids, where input is given as a matrix.

Description

This function assumes that the input gene expression matrix file has been already loaded into a matrix. The function accepts a matrix and applies kmeans clustering to generate cluster centroids.

Usage

```
smfishHmrf.generate.centroid(y, par_k, par_seed = -1, nstart)
```

Arguments

y	expression matrix
par_k	number of clusters
par_seed	random generator seed (to fix it set it to above 0, or -1 if no fixing). Change the par_seed to vary the initialization.
nstart	number of starts (kmeans parameter). It is recommended to set nstart to at least 100 (preferrably 1000).

Value

A kmeans list with centers and cluster fields

Examples

```
data(seqfishplus)
kk<-smfishHmrf.generate.centroid(seqfishplus$y, par_k=9, par_seed=100, nstart=100)
```

smfishHmrf.generate.centroid.it

Generate cluster centroids, where input is a file.

Description

This function generates cluster centroids from applying kmeans. It accepts an expression matrix file as input.

Usage

```
smfishHmrf.generate.centroid.it(
  expr_file,
  par_k,
  par_seed = -1,
  nstart,
  name = "test",
  output_dir = "."
)
```

Arguments

expr_file	expression matrix file. The expression file should be a space-separated file. The rows are genes. The columns are cells. There is no header row. The first column is a gene index (ranges from 1 to the number of genes). Note the first column is not gene name.
par_k	number of clusters
par_seed	random generator seed (-1 if no fixing). Change the par_seed to vary the initialization.
nstart	number of starts (kmeans). It is recommended to set nstart to at least 100 (preferrably 1000).
name	name of this run
output_dir	output directory; where to store the kmeans results

Details

Note that after running kmeans step, the function also automatically saves the kmeans results to the `output_dir` directory. The results will be found in two files:

1. `{output_dir}/k_{par_k}/f{name}.gene.ALL.centroid.txt`
2. `{output_dir}/k_{par_k}/f{name}.gene.ALL.kmeans.txt`

where `{}` refers to the value of parameters.

Value

A kmeans object which is a list with centers and cluster fields

Examples

```
mem_file = system.file("extdata", "ftest.expression.txt", package="smfishHmrf")
kk = smfishHmrf.generate.centroid.it(mem_file, par_k=9, par_seed=100,
    nstart=100, name="test", output_dir=tempdir())
```

smfishHmrf.generate.centroid.save

Save the cluster centroids to file

Description

This function is run after the kmeans step. It takes a kmeans object (containing the kmeans result) as input and save the cluster centroids to file.

Note that the location of saving and the file names are decided by the following rule:

1. {output_dir}/k_{par_k}/f{name}.gene.ALL.centroid.txt
2. {output_dir}/k_{par_k}/f{name}.gene.ALL.kmeans.txt

where {} refers to the value of parameters.

Usage

```
smfishHmrf.generate.centroid.save(kk, name = "test", output_dir = ".")
```

Arguments

kk	kmeans object
name	name of the run
output_dir	output directory; where to save the results

Examples

```
expr_file = system.file("extdata", "ftest.expression.txt", package="smfishHmrf")
y<-smfishHmrf.read.expression(expr_file)
kk = smfishHmrf.generate.centroid(y, par_k=9, par_seed=100, nstart=100)
smfishHmrf.generate.centroid.save(kk, name="test", output_dir=tempdir())
```

smfishHmrf.generate.centroid.use.exist

Use existing cluster centroids

Description

This function assumes that cluster centroids have already been generated from previously applying kmeans on the dataset. The results should have been saved. It will load cluster centroids from existing clustering result files. The results should be found in input_dir directory. The function looks for the following two kmeans result files:

1. {input_dir}/k_{par_k}/f{name}.gene.ALL.centroid.txt
2. {input_dir}/k_{par_k}/f{name}.gene.ALL.kmeans.txt

where {} refers to the value of parameters down below

Usage

```
smfishHmrf.generate.centroid.use.exist(name = "test", input_dir = ".", par_k)
```

Arguments

name	name of this run
input_dir	input directory
par_k	number of clusters

Value

A kmeans object which is a list with centers and cluster fields

Examples

```
kmeans_results = system.file("extdata", package="smfishHmrf")
kk = smfishHmrf.generate.centroid.use.exist(name="test", input_dir=kmeans_results, par_k=9)
```

---

smfishHmrf.hmrfem.multi

Performs HMRF for multivariate normal distribution. Accepts R data structures as inputs. Accepts a single beta.

Description

This function performs HMRF (Zhu et al. 2018-Dec-01) on multivariate normal distributions. Different from other variations, this function accepts R data structures directly as inputs, and only accepts a single value of beta.

This function exists for legacy and compatibility reason. User should use **smfishHmrf.hmrfem.multi.it.min** function.

Usage

```
smfishHmrf.hmrfem.multi(
  y,
  neighbors,
  numnei,
  blocks,
  beta = 0.5,
  mu,
  sigma,
  err = 1e-07,
  maxit = 50,
  verbose,
  dampFactor = NULL,
  forceDetectDamp = FALSE,
  tolerance = 1e-60
)
```

Arguments

y	gene expression matrix
neighbors	adjacency matrix between cells
numnei	a vector containing number of neighbors per cell
blocks	a list of cell colors for deciding the order of cell update
beta	the beta to try (smoothness parameter)
mu	a 2D matrix (i,j) of cluster mean (initialization)
sigma	a 3D matrix (i,j,k) where (i,j) is the covariance of cluster k (initialization)
err	the error that is allowed between successive iterations
maxit	maximum number of iterations
verbose	TRUE or FALSE
dampFactor	the dampening factor
forceDetectDamp	will auto detect a dampening factor instead of using the specified one
tolerance	applicable when forceDetectDamp is set to TRUE

Value

A list of prob, new mu, new sigma, unnormalized prob after iterations finish

More information

Arguments mu and sigma refer to the cluster centroids from running kmeans algorithm. They serve as initialization of HMRF. Users should refer to **smfishHmrf.hmrfem.multi.it.min** for more information about function parameters and the requirements.

References

Zhu Q, Shah S, Dries R, Cai L, Yuan G (2018-Dec-01). "Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data." *Nature Biotechnology*, **36**, 1183–1190. doi: 10.1038/nbt.4260.

Examples

```
data(seqfishplus)
s <- seqfishplus
res<-smfishHmrf.hmrfem.multi(s$y, s$nei, s$numnei, s$blocks, beta=28,
mu=s$mu, sigma=s$sigma, err=1e-7, maxit=50, verbose=TRUE, dampFactor=s$damp,
tolerance=1e-5)
```

`smfishHmrf.hmrfem.multi.it` *Perform HMRF for multivariate normal distribution. Accepts R data structures as inputs. Accepts multiple betas.*

Description

This function performs HMRF model (Zhu et al. 2018-Dec-01) on inputs which are directly R data structures. Different from smfishHmrf.hmrfem.multi, this function iterates over multiple betas rather than a single beta. Different from smfishHmrf.hmrfem.multi.it.min, this function accepts R data structures (i.e. parameters y, nei, blocks) as inputs to the function rather than accepting file names. This function will save the results of HMRF to the output directory. It will return void.

This function exists for legacy and compatibility reason. User should use **smfishHmrf.hmrfem.multi.it.min** function.

Usage

```
smfishHmrf.hmrfem.multi.it(
  name,
  outdir,
  k,
  y,
  nei,
  beta = 0,
  beta_increment = 1,
  beta_num_iter = 10,
  numnei,
  blocks,
  mu,
  sigma,
  damp
)
```

Arguments

`name`	name for this run (eg test)
`outdir`	output directory
`k`	number of clusters
`y`	gene expression matrix
`nei`	adjacency matrix between cells
`beta`	initial beta
`beta_increment`	beta increment
`beta_num_iter`	number of betas to try
`numnei`	a vector containing number of neighbors per cell

blocks	a list of cell colors for deciding the order of cell update
mu	a 2D matrix (i,j) of cluster mean (initialization)
sigma	a 3D matrix (i,j,k) where (i,j) is the covariance of cluster k (initialization)
damp	a list of dampening factors (length = k)

More information

Arguments mu and sigma refer to the cluster centroids from running kmeans algorithm. They serve as initialization of HMRF. Users should refer to **smfishHmrf.hmrfem.multi.it.min** for more information about function parameters and the requirements.

References

Zhu Q, Shah S, Dries R, Cai L, Yuan G (2018-Dec-01). "Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data." *Nature Biotechnology*, **36**, 1183–1190. doi: 10.1038/nbt.4260.

Examples

```
y<-as.matrix(read.table(system.file("extdata", "ftest.expression.txt",
    package="smfishHmrf"), header=FALSE, row.names=1))
nei<-as.matrix(read.table(system.file("extdata", "ftest.adjacency.txt",
    package="smfishHmrf"), header=FALSE, row.names=1))
colnames(nei)<-NULL; rownames(nei)<-NULL
blocks<-c(t(read.table(system.file("extdata", "ftest.blocks.txt",
    package="smfishHmrf"), header=FALSE, row.names=1)))
blocks<-lapply(1:max(blocks), function(x) which(blocks == x))
numnei<-apply(nei, 1, function(x) sum(x!=-1))
k<-9
kmeans_results = system.file("extdata", package="smfishHmrf")
kk = smfishHmrf.generate.centroid.use.exist(name="test", input_dir=kmeans_results, k)
numcell<-dim(y)[1]; m<-dim(y)[2]
mu<-t(kk$centers) #should be dimension (m,k)
lclust<-lapply(1:k, function(x) which(kk$cluster == x))
damp<-array(0, c(k)); sigma<-array(0, c(m,m,k))
for(i in 1:k){
    sigma[, , i] <- cov(y[lclust[[i]], ])
    di<-findDampFactor(sigma[,,i], factor=1.05, d_cutoff=1e-5, startValue=0.0001)
    damp[i]<-ifelse(is.null(di), 0, di)
}
smfishHmrf.hmrfem.multi.it(name="test", outdir=tempdir(), k=k, y=y, nei=nei,
    beta=28, beta_increment=2, beta_num_iter=1, numnei=numnei, blocks=blocks,
    mu=mu, sigma=sigma, damp=damp)

## Not run:
# alternatively, to test a larger set of betas:
smfishHmrf.hmrfem.multi.it(name="test", outdir=tempdir(), k=k, y=y, nei=nei,
    beta=0, beta_increment=2, beta_num_iter=20, numnei=numnei, blocks=blocks,
    mu=mu, sigma=sigma, damp=damp)

## End(Not run)
```

`smfishHmrf.hmrfem.multi.it.min` *Perform HMRF on multivariate normal distributions. Accepts file names as inputs. Accepts multiple betas.*

Description

This function performs HMRF (Zhu et al. 2018-Dec-01) for multi variate normal distributions. It takes minimum required inputs (inputs being file names). There are a couple of files required:

1. a file containing expression matrix
2. a file containing cell neighborhood matrix
3. a file containing node (or cell) color. This is used for updating cells during HMRF iterations.

HMRF needs users to specify the initializations of parameters (mu and sigma). It is recommended to use the kmeans centroids as initializations (specified by `kk` parameter). Note: kmeans should be run prior to this function.

Usage

```
smfishHmrf.hmrfem.multi.it.min(
  mem_file,
  nei_file,
  block_file,
  kk,
  par_k,
  name = "test",
  output_dir = ".",
  tolerance = 1e-05,
  beta = 0,
  beta_increment = 1,
  beta_num_iter = 10
)
```

Arguments

`mem_file`	expression file. The expression file should be a space-separated file. The rows are genes. The columns are cells. There is no header row. The first column is a gene index (ranges from 1 to the number of genes). Note the first column is not gene name. See section **Data preprocessing** for which form of expression works best.

`nei_file` file containing cell neighborhood matrix. This should be a space-separated file. The rows are cells. The columns are neighbors. There is no header row. The first column is the cell index (1 to number of cells). Each row lists the indices of neighbor cells. The dimension of the cell neighborhood matrix is (num_cell, max_num_neighbors). If a cell does not have enough neighbors, the remaining entries of that row is padded with -1. The R package Giotto http://spatialgiotto.com (Dries et al. 2020) contains a number of functions for generating the cell neighborhood network.

`block_file` file containing cell colors (which determines cell update order). The order of updating the state probabilities of each cell can matter the result. Cells (or nodes) and their immediate neighbors are not updated at the same time. This is akin to the vertex coloring problem. This file contains the color of each cell such that no two neighbor cells have the same color. The file is 2-column, space-separated. Column 1 is cell ID, and column 2 is the cell color (integer starting at 1). The python utility get_vertex_color.py https://bitbucket.org/qzhudfci/smfishhmrf-py/src/master/get_vertex_color.py (requires smfishHmrf-py package https://pypi.org/project/smfishHmrf/) can generate this file.

`kk` kmeans results (object returned by kmeans). Kmeans (one of functions smfishHmrf.generate.centroid.it or smfishHmrf.generate.centroid) should be run before this function.

`par_k` number of clusters

`name` name for this run (eg test)

`output_dir` output directory

`tolerance` tolerance

`beta`, `beta_increment`, `beta_num_iter`
3 values specifying the range of betas to try: the initial beta, the beta increment, and the number of betas. Beta is the smoothness parameter. Example: `beta`=0, `beta_increment`=2, `beta_num_iter`=6 means to try betas: 0, 2, 4, 6, 8, 10. See section **Betas** for more information.

Data preprocessing

It assumes that the expression values follow a multivariate gaussian distribution. We generally recommend using **log2 transformed counts further normalized by z-scores (in both x- and y-dimensions)**. Double z-scoring this way helps to remove the inherent bias of zscoring just one dimension (as the results might present a bias towards cell counts).

Betas

Beta is the smoothness parameter in HMRF. The higher the beta, the more the HMRF borrows information from the neighbors. This function runs HMRF across a range of betas. To decide which beta range, here are some guideline:

- if the number of genes is from 10 to 50, the recommended range is 0 to 10 at beta increment of 0.5.
- if the number of genes is below 50, the recommended range is 0 to 15 at beta increment of 1.
- if the number of genes is between 50 to 100, the range is 0 to 50 at beta increment of 2.

- if the number of genes is between 100 and 500, the range is 0 to 100 at beta increment of 5.

Within the range of betas, we recommend selecting the best beta by the Bayes information criterion. This requires first performing randomization of spatial positions to generate the null distribution of log-likelihood scores for randomly distributed cells for the same range of betas. Then find the beta where the difference between the observed and the null log-likelihood is maximized.

Variations

- `smfishHmrf.hmrfem.multi.it.min` (this function): supports multiple betas; supports file names as inputs. Recommended.
- `smfishHmrf.hmrfem.multi.it`: supports multiple betas; supports R data structures as inputs.
- `smfishHmrf.hmrfem.multi`: supports a single beta; supports R data structures as inputs.

References

Zhu Q, Shah S, Dries R, Cai L, Yuan G (2018-Dec-01). "Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data." *Nature Biotechnology*, **36**, 1183–1190. doi: 10.1038/nbt.4260.

Eng CL, Lawson M, Zhu Q, Dries R, Koulena N, Takei Y, Yun J, Cronin C, Karp C, Yuan G, Cai L (2019-Apr-01). "Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+." *Nature*, **568**, 235–239. doi: 10.1038/s415860191049y.

Dries R, Zhu Q, Dong R, Eng CL, Li H, Liu K, Fu Y, Zhao T, Sarkar A, Bao F, George RE, Pierson N, Cai L, Yuan G (2020). "Giotto, a toolbox for integrative analysis and visualization of spatial expression data." *bioRxiv*. doi: 10.1101/701680.

Examples

```
mem_file = system.file("extdata", "ftest.expression.txt", package="smfishHmrf")
nei_file = system.file("extdata", "ftest.adjacency.txt", package="smfishHmrf")
block_file = system.file("extdata", "ftest.blocks.txt", package="smfishHmrf")
par_k = 9
name = "test"
output_dir = tempdir()

## Not run:
kk = smfishHmrf.generate.centroid.it(mem_file, par_k, par_seed=100,
nstart=100, name=name, output_dir=output_dir)

## End(Not run)

# alternatively, if you already have run kmeans before, you can load it directly
kmeans_results = system.file("extdata", package="smfishHmrf")
kk = smfishHmrf.generate.centroid.use.exist(name=name, input_dir=kmeans_results, par_k)

smfishHmrf.hmrfem.multi.it.min(mem_file, nei_file, block_file, kk, par_k,
name=name, output_dir=output_dir, tolerance=1e-5,
beta=28, beta_increment=2, beta_num_iter=1)

## Not run:
```

```
# alternatively, to test a larger set of beta's
smfishHmrf.hmrfem.multi.it.min(mem_file, nei_file, block_file, kk, par_k,
name=name, output_dir=output_dir, tolerance=1e-5,
beta=0, beta_increment=2, beta_num_iter=20)

## End(Not run)
```

`smfishHmrf.hmrfem.multi.save` *Save the HMRF result*

Description

This function assumes that HMRF has been run via smfishHmrf.hmrfem.multi, smfishHmrf.hmrfem.multi.it or smfishHmrf.hmrfem.multi.it.min function. It assumes the results have been generated. This function saves the results of each beta to the output directory. It will return void.

Usage

```
smfishHmrf.hmrfem.multi.save(name, outdir, beta, tc.hmrfem, k)
```

Arguments

`name`	name for this run (eg test)
`outdir`	output directory
`beta`	beta to save
`tc.hmrfem`	the result of running of hmrfem on single beta (from smfishHmrf.hmrfem.multi)
`k`	number of clusters

Examples

```
data(seqfishplus)
s <- seqfishplus
tc.hmrfem<-smfishHmrf.hmrfem.multi(s$y, s$nei, s$numnei, s$blocks, beta=28,
    mu=s$mu, sigma=s$sigma, err=1e-7, maxit=50, verbose=TRUE, dampFactor=s$damp,
    tolerance=1e-5)
smfishHmrf.hmrfem.multi.save(name="test", outdir=tempdir(), beta=28, tc.hmrfem=tc.hmrfem, k=9)
```

smfishHmrf.read.expression *smfishHmrf.read.expression*

Description

Reads expression

Usage

```
smfishHmrf.read.expression(expr_file)
```

Arguments

expr_file	expression matrix file. The expression file should be a space-separated file. The rows are genes. The columns are cells. There is no header row. The first column is a gene index (ranges from 1 to the number of genes). Note the first column is not gene name.

Value

A matrix with gene expression matrix

Examples

```
expr_file = system.file("extdata", "ftest.expression.txt", package="smfishHmrf")
y<-smfishHmrf.read.expression(expr_file)
```

Index

∗ **datasets**
seqfishplus, 3

findDampFactor, 2

seqfishplus, 3
smfishHmrf, 3
smfishHmrf.generate.centroid, 5
smfishHmrf.generate.centroid.it, 5
smfishHmrf.generate.centroid.save, 7
smfishHmrf.generate.centroid.use.exist, 7
smfishHmrf.hmrfem.multi, 8
smfishHmrf.hmrfem.multi.it, 10
smfishHmrf.hmrfem.multi.it.min, 12
smfishHmrf.hmrfem.multi.save, 15
smfishHmrf.read.expression, 16