Package ‘XMRF’

June 25, 2015

Type Package

Title Markov Random Fields for High-Throughput Genetics Data

Version 1.0

Date 2015-06-12

Author

Ying-Wooi Wan, Genevera I. Allen, Yulia Baker, Eunho Yang, Pradeep Ravikumar, Zhandong Liu

Maintainer Ying-Wooi Wan <yingwoow@bcm.edu>

Depends R (>= 3.0.2)

Imports igraph, glmnet, MASS, snowfall, parallel, Matrix

Description Fit Markov Networks to a wide range of high-throughput genomics data.

License GPL-2

URL http://www.liuzlab.org/XMRF/

LazyLoad true

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-06-25 07:07:40

R topics documented:

XMRF-package *A R Package to Fit Markov Networks to High-throughput Genomics Data*

Description

An R package to learn and visualize the underlying relationships between genes from various types of high-throughput genomics data.

Details

Package:	XMRF
Type:	Package
Version:	1.0
Date:	2015-06-12
License:	GPL-2

Technological advances have produced large amounts of high-throughput "omics" data that allow us to study the complicated interactions between genes, mutations, aberrations, and epi-genetic markers. Markov Random Fields (MRFs), or Markov Networks, enable us to estimate these genomics networks via sparse, high-dimensional undirected graphical models.

Here, we provide the community a convenient and useful tool to learn the complex genomics networks from various types of high-throughput genomics data. This package encodes the recently proposed parametric family of graphical models based on node-conditional univariate exponential family distributions (Yang *et. al*, 2012, 2013a). Specifically, our package has methods for estimating Gaussian graphical model (Meinshausen and Buhlmann, 2006), Ising model (Ravikumar *et. al*, 2010), and the Poisson family graphical models (Allen and Liu, 2012, 2013; Yang *et. al* 2013b). These models can be used to estimate genetic networks from a variety of data types:

Genomics Data	Type	XMRF Family
========================	==========	============
RNA-Seq or miRNA-Seq	Counts	`LPGM`, `SPGM`
Microarray or Methylation array	Continuous	`GGM`
Mutation or CNVs	Binary	`ISM`

To estimate the network structures from different types of genomics data with this package, users simply need to specify the `"method"` in the main function, for example `XMRF(..., method="LPGM")` to fit LPGM to next-generation sequencing data.

In this package, we implement the neighborhood selection graph estimation technique by proximal or projected gradient descent using warm starts over the range of regularization parameters. This technique allows estimation of the neighborhood for each node independently and thus can be done in parallel, thus speeding computation.

This package also implements two data-driven approaches to select the sparsity of a fitted network: a stability-based approach for a single regularization value over many bootstrap resamples (Meinshausen and Buhlmann, 2010), or computed over a range of regularization values with StARS (Liu *et. al.*, 2010).

Author(s)

Ying-Wooi Wan, Genevara Allen, Yulia Baker, Eunho Yang, Pradeep Ravikumar, and Zhandong Liu

Maintainer: Ying-Wooi Wan<yingwoow@bcm.edu>

References

Allen, G.I., and Liu, Z. (2012). A Log-Linear graphical model for inferring genetic networks from high-throughput sequencing data. *The IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2012)*.

Allen, G. I., and Liu, Z. (2013). A Local Poisson Graphical Model for Inferring Genetic Networks from Next Generation Sequencing Data. *IEEE Transactions on NanoBioscience*, **12**(3), pp.1-10

Liu, H., Roeder, K., and Wasserman, L. (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. *NIPS 23*, pp.1432-1440.

Meinshausen, N. and Buhlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, **34**(3), pp.1436-1462.

Meinshausen, N. and Buhlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4), pp.417-473.

Ravikumar, P., Wainwright, M., and Lafferty, J. (2010). High-dimensional ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, **38**(3), pp.1287-1319.

Yang, E., Ravikumar, P.K., Allen, G.I., and Liu, Z. (2012). Graphical models via generalized linear models. *NIPS*, **25**, pp.1367–1375.

Yang, E., Ravikumar, P.K., Allen, G.I., and Liu, Z. (2013a). On graphical models via univariate exponential family distributions. *arXiv preprint arXiv:1301.4183*.

Yang, E., Ravikumar, P.K., Allen, G.I., and Liu, Z. (2013b). On Poisson graphical models. *NIPS*, pp.1718-1726.

See Also

XMRF

Examples

```
library(XMRF)

n = 100
p = 20
```

```
sim <- XMRF.Sim(n=n, p=p, model="LPGM", graph.type="scale-free")
simDat <- sim$X

# Compute the optimal lambda
lmax = lambdaMax(t(simDat))
lambda = 0.01* sqrt(log(p)/n) * lmax
# Run LPGM
lpgm.fit <- XMRF(simDat, method="LPGM", N=10, lambda.path=lambda, parallel=FALSE)

# Print the fitted Markov networks
lpgm.fit

ml = plotNet(sim$B)
ml = plot(lpgm.fit, mylayout=ml)
```

brcadat *RNA-Seq Data of BRCA Patients from TCGA*

Description

A matrix of RNA-Seq read counts from BRCA patients.

Usage

```
data("brcadat")
```

Format

A matrix of level 3 RNA-Seq (UNC Illumina HiSeq RNASeqV2) data with 445 breast invasive carcinoma (BRCA) patients from the Cancer Genome Atlas (TCGA) project on 353 genes with somatic mutations listed in the Catalogue of Somatic Mutations in Cancer (COSMIC). The matrix is organized in dimension of *gene* x *sample*.

ggm.fit *Fitted Gaussian Graphical Models*

Description

An example fitted Gaussian graphical model

Usage

```
data("ggm.fit")
```

Format

A `GMS` object.

Details

This fitted Gaussian graphical model is included as the example model for demonstrating the usage of other functions. The model was fitted with a simulated multivariate Gaussian data of scale-free graph structure, 30 nodes, and 200 observations. StARS stability selection of 100 iterations was used to identify the optimal network over the regularization path of 10 parameters.

See Also

`GMS`, `plot.GMS`

`GMS` *Graphical Models Object*

Description

This class of objects is returned by the `XMRF` function in this package, to represent the fitted Markov Network(s). Objects of this class have the `print` method to display the core information of the fitted models and `plot` method to plot the optimal Markov Network.

Arguments

`v`	vector of (`nlams` length) network variability measured for each regularization level.
`D`	a list of *pxp* matrices of stability scores of inferred edges of each network along the regularization path.
`lambda.path`	numeric vector used for regularization path.
`opt.lambda`	lambda value that gives the optimal network (network with maximum variability).
`network`	a list of *pxp* matrices of fitted networks along the regularization path.
`opt.index`	index of the regularization value that gives the optimal network.

See Also

`XMRF`, `plot.GMS`

lambdaMax *Maximum lambda*

Description

Compute the maximum `lambda`

Usage

```
lambdaMax(X)
```

Arguments

`X`	a *n*x*p* data matrix.

Details

Compute the largest value for regularization (maximum `lambda`) that gives the null model. The maximum `lambda` is computed based on the input data matrix, and is the maximum element from column-wise multiplication of data matrix normalized by the number of observations.

Value

a numeric value

plot.GMS *Plot GMS Object*

Description

Default function to plot the network of a GMS object.

Usage

```
## S3 method for class 'GMS'
plot(x, fn = "", th = 1e-06, i = NULL, mylayout = NULL, vars = NULL, ...)
```

Arguments

`x`	a `GMS` object.
`fn`	file name to save the network plot; default to be an empty string, so the network is plotted to the standard output (screen). NOTE: if a file name is specified, it should be file name for PDF file.
`th`	numeric value, default to 1e-06. To specify the threshold if the estimated coefficient between two variables is to be considered connected.

`i` index of the network (along the regularization path) to be plotted. Default to `NULL` to plot the optimal network.

`mylayout` graph layout to draw the network, default to `NULL`.

`vars` vector of variable names, default to `NULL`.

`...` other generic arguments for `plot` method.

Details

This is the default plotting function for `GMS` objects (Markov Networks inferred over a regularization path). Refer to `GMS` for details on `GMS` object. This function will plot the optimal network on the screen by default. However, given a file name, the plot will be saved to a PDF file. Also, given a specific index corresponding to the index of `lambda.path`, the associated network will be plotted.

The network will be plotted in force-directed layout (`layout.fruchterman.reingold` with default parameters implemented in `igraph` package).

Value

Returns the layout object from `igraph` package - numeric matrix of two columns and the rows with the same number as the number of vertices.

See Also

`GMS`

Examples

```
library(XMRF)
data('ggm.fit')
plot(ggm.fit, fn="ggm.fit.net.pdf")
```

`plotGML` *Plot Network in GML*

Description

Plot the network in graph modeling language (GML).

Usage

```
plotGML(x, fn = "", th = 1e-06, i = NULL, weight = FALSE, vars = NULL)
```

Arguments

`x` a `GMS` object.

`fn` file name to save the GML file.

`th` numeric value, default to 1e-06. To specify the threshold if the estimated coefficient between two variables is to be considered connected.

`i` index of the network (along the regularization path) to be plotted. Default to `NULL` for optimal network.

`weight` boolean value to indicate if writing the stability on the inferred edges, default to `FALSE`.

`vars` vector of variable names, default to `NULL`.

`plotNet` *Plot Network*

Description

Plot a network with specific layout.

Usage

```
plotNet(net, fn = "", th = 1e-06, mylayout = NULL)
```

Arguments

`net` a square adjacency matrix of the network to be plotted.

`fn` file name to save the network plot. Default to be an empty string, so the network is plotted to the standard output (screen). NOTE: if a file name is specified, it should be file name for PDF file.

`th` numeric value, default to 1e-06. To specify the threshold if the estimated coefficient between two variables is to be considered connected.

`mylayout` graph layout to draw the network, default to `NULL`.

Details

This function serves as the alternative plotting function to allow users to plot a specific network with specific layout, such as plotting the simulated network.

Value

Returns the layout object from `igraph` package - numeric matrix of two columns and the rows with the same number as the number of vertices.

Examples

```
library(XMRF)
n = 200
p = 30
sim <- XMRF.Sim(n=n, p=p, model="LPGM", graph.type="scale-free")
ml = plotNet(sim$B)
```

---

processSeq *Process Sequencing Data for Poisson-based MRFs*

Description

Process and normalize RNA-Sequencing count data into a distribution appropriate for Poisson MRFs.

Usage

```
processSeq(X, quanNorm = 0.75, nLowCount = 20, percentLowCount = 0.95, NumGenes = 500,
PercentGenes = 0.1)
```

Arguments

`X`	*nxp* data matrix.
`quanNorm`	an optional parameter controlling the quantile for sample normalization, default to 0.75.
`nLowCount`	minimum read count to decide if to filter a gene, default to 20.
`percentLowCount`	filter out a gene if it has this percentage of samples less than `nLowCount`, default to 0.95.
`NumGenes`	number of genes to retain in the final data set, default to 500.
`PercentGenes`	percentage of genes to retain, default to 0.1.

Details

To process the next-generation sequencing count data into proper distribution (with dispersion removed), the following steps are taken in this function:

1. Quantile normalization for the samples.
2. Filter out genes with all low counts.
3. Filter genes by maximal variance (if specified).
4. Transform the data to be closer to the Poisson distribution. A log or power transform is considered and selected based upon the Kolmogorov-Smirnov goodness of fit test.

Value

a *n* x `NumGenes` or `PercentGenes` processed data matrix.

Examples

```
library(XMRF)
data('brcadat')
brca = t(processSeq(t(brcadat), PercentGenes=1))
```

XMRF *Markov Random Fields for Exponential Family Distributions*

Description

Infer networks from genomics data using Markov Random Fields specified by node-conditional univariate exponential family distributions.

Usage

```
XMRF(X, method = "LPGM", stability = "bootstrap", N = 100, beta = 0.01, lmin = 0.01,
  nlams = 20, lambda.path = NULL, parallel = TRUE, nCpus = 4, sym = TRUE, th = 0.01,
   sth = 0.95, R = max(X), R0 = 0)
```

Arguments

`X` a *p*x*n* data matrix.

`method` specification of the type of MRF model, default to "`LPGM`" for log-linear Poisson-based graphical model. Other allowed methods are "`PGM`" for regular Poisson, "`TPGM`" for truncated Poisson, "`SPGM`" for sublinear Poisson, "`GGM`" for Gaussian graphical models, and "`ISM`" for Ising model.

`stability` specification of the stability method, default to "`bootstrap`". Another accepted value is "`star`" for Stability Approach to Regularization Selection (StARS).

`N` number of iterations for stability selection, default to 100.

`beta` threshold value on sparsity of the network, default to 0.01.

`lmin` ratio of minimum lambda value from the maximum lambda value, default to 0.01.

`nlams` number of lambda for regularization, default to 20.

`lambda.path` vector lambda used for regularization, default ot `NULL`.

`parallel` logical value to indicate if the process should be run parallelly in multiple threads, default to `TRUE`.

`nCpus` number of (maximum) cores to use for parallel execution, default to 4.

`sym` logical value to indicate if symmetry is enforced on the inferred edges, default to `TRUE`.

`th` threshold value for the estimated edge coefficient, default to 0.005.

`sth` an inferred edge is retained only if its stability score is greater than `sth`, default to 0.9.

`R` truncation level for classes "`TPGM`" and "`SPGM`". The value has to be positive. Default to the maximum value of the input data matrix.

`R0` lower-bound truncation level for "`SPGM`", default to 0.

Details

This is the main function of the package that fits exponential family Markov Networks to genomics data. To estimate the network structures using native distribution of the genomics data, specify the MRF family types in the "`method`" parameter. For genomic networks based on next-generation sequencing data, we recommend using the `LPGM` family. The table at the beginning of the document lists the family type recommended for each of the genomic data platforms.

Value

An object of class `GMS` will be returned representing the inferred Markov networks over the regularization path. See `GMS` for details.

References

Allen, G.I., and Liu, Z. (2012). A Log-Linear graphical model for inferring genetic networks from high-throughput sequencing data. *The IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2012)*.

Allen, G. I., and Liu, Z. (2013). A Local Poisson Graphical Model for Inferring Genetic Networks from Next Generation Sequencing Data. *IEEE Transactions on NanoBioscience*, **12**(3), pp.1-10

Liu, H., Roeder, K., and Wasserman, L. (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. *NIPS 23*, pp.1432?1440.

Meinshausen, N. and Buhlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, **34**(3), pp.1436?1462.

Meinshausen, N. and Buhlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4), pp.417?473.

Ravikumar, P., Wainwright, M., and Lafferty, J. (2010). High-dimensional ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, **38**(3), pp.1287?1319.

Yang, E., Ravikumar, P.K., Allen, G.I., and Liu, Z. (2012). Graphical models via generalized linear models. *NIPS*, **25**, pp.1367–1375.

Yang, E., Ravikumar, P.K., Allen, G.I., and Liu, Z. (2013a). On graphical models via univariate exponential family distributions. *arXiv preprint arXiv:1301.4183*.

Yang, E., Ravikumar, P.K., Allen, G.I., and Liu, Z. (2013b). On Poisson graphical models. *NIPS*, pp.1718-1726.

See Also

`XMRF-package`, `GMS`, `plot.GMS`

Examples

```
# Example for LPGM
# Refer to the package's introduction for identical example
## Not run: n = 100
## Not run: p = 20
## Not run: sim <- XMRF.Sim(n=n, p=p, model=”LPGM”, graph.type=”scale-free”)
## Not run: simDat <- sim$X
## Not run: # Compute the optimal lambda
## Not run: lmax = lambdaMax(t(simDat))
## Not run: lambda = 0.01* sqrt(log(p)/n) * lmax
## Not run: # Run LPGM
## Not run: lpgm.fit <- XMRF(simDat, method="LPGM", N=10, lambda.path=lambda)
## Not run: ml = plotNet(sim$B, fn="simDat.netPlot.pdf")
## Not run: ml = plot(lpgm.fit, fn="lpgm.netPlot_1.pdf", i=1, mylayout=ml)
## Not run: plot(lpgm.fit, fn="lpgm.fit.net.pdf")
```

XMRF.Sim *Generate simulated data from XMRF models*

Description

Generate data from different multivariate distributions with different network structures.

Usage

```
XMRF.Sim(n = 100, p = 50, model = "LPGM", graph.type = "scale-free")
```

Arguments

`n`	number of samples, default to 100.
`p`	number of variables, default to 50.
`model`	Markov Network models to indicate the distribution family of the data to be generated, default to "`LPGM`". Other model options include "`PGM`", "`TPGM`", "`SPGM`", "`GGM`" and "`ISM`".
`graph.type`	graph structure with 3 options:"`scale-free`", "`hub`", and "`lattice`". Default to "`scale-free`".

Details

This function will first generate a graph of the specified graph structure; then based on the generated network, it simulates a multivariate data matrix that follows distribution for the Markov Random Fields model specified.

Value

A list of two elements:

`B`	*p*x*p* adjacency matrix of the generated graph.
`X`	*p*x*n* data matrix.

Examples

```
library(XMRF)

# simulate scale-free network and data of multivariate Poisson for LPGM
sim <- XMRF.Sim(n=100, p=20, model="LPGM", graph.type="scale-free")
hist(sim$X)
plotNet(sim$B)

# simulate hub network and data of multivariate Gaussian for GGM
sim <- XMRF.Sim(n=100, p=20, model="GGM", graph.type="hub")
hist(sim$X)
plotNet(sim$B)

# simulate hub network and data of multivariate bionomial for ISM
sim <- XMRF.Sim(n=100, p=15, model="ISM", graph.type="hub")
hist(sim$X)
plotNet(sim$B)
```

Index