

The MMG Package

November 18, 2008

Type Package

Title Mixture Model on Graphs

Version 1.4.0

Date 2008-11-18

Author Josselin Noirel and Guido Sanguinetti

Maintainer Josselin Noirel <j.noirel@sheffield.ac.uk>

Depends R (>= 2.6.0)

Description This package implements the Mixture Model on Graphs developed in Sanguinetti et al., Bioinformatics 2008. The

License GPL-3

R topics documented:

MMG-package	1
MMG.compute	3
Index	7

MMG-package *Mixture Model on Graphs*

Description

MMG is a Mixture Model which can integrate the structure of a network in the statistical analysis of data.

This implementation MMG assumes the existence of three classes of genes/proteins/etc. within a network: down-regulated, up-regulated, or unchanged. The underlying data are log-ratios, hence relative, measurements.

The package aims at identifying clusters of genes/proteins/etc. that behave consistently along the network's pathways. This is done by implementing the Bayesian model described in the Reference Sanguinetti et al. (2008), by running a Gibbs sampler, and by cutting the graph.

The Gibbs sampler is run by `MMG.compute`. To cut the graph is done by `MMG.cut.graph`. Finally, `MMG.make.dot` conveniently produces a DOT file, a file whose format can be used for visualation using various softwares (including GraphViz <http://www.graphviz.org/> - see also the package `Rgraphviz`).

Details

Package: MMG
Type: Package
Version: 1.2.2
Date: 2008-05-20
License: GPL-3+

Author(s)

Josselin Noirel, based on an original implementation by Guido Sanguinetti (<http://www.dcs.shef.ac.uk/~guido/>).

Maintainer: Josselin Noirel <j.noirel@sheffield.ac.uk>

References

Sanguinetti, Noirel, and Wright., MMG: a probabilistic tool to identify submodules of metabolic pathways, *Bioinformatics* (2008)

Examples

```
## Not run:
r <- MMG.compute(file.name = "NostocData/R_net.dat",
                 steps = 100000, burn.in = 1000,
                 sigma = 0.3, alpha = 1)
## End(Not run)
## Not run: n <- r$dat$n.nodes
## Not run:
s <- MMG.cut.graph(r, descriptions = "NostocData/R_descr.dat",
                  method = "THRESHOLD", threshold = 0.15, select = "UP")
## End(Not run)
## Not run: l <- (1:n)[s$components != 0]
## Not run:
MMG.make.dot(r, file.name = "nostoc.dot", selection = 1, type = "UNDIRECTED",
             rem.loops = TRUE)
## End(Not run)
```

Description

This implementation MMG assumes the existence of three classes of genes/proteins/etc. within a network: down-regulated, up-regulated, or unchanged. The underlying data are log-ratios, hence relative, measurements.

The package aims at identifying clusters of genes/proteins/etc. that behave consistently along the network's pathways. This is done by implementing the Bayesian model described in the Reference Sanguinetti et al. (2008), by running a Gibbs sampler, and by cutting the graph.

The Gibbs sampler is run by `MMG.compute`. To cut the graph is done by `MMG.cut.graph`. Finally, `MMG.make.dot` conveniently produces a DOT file, a file whose format can be used for visualisation using various softwares (including GraphViz <http://www.graphviz.org/> - see also the package `Rgraphviz`).

Usage

```
r <- MMG.compute(file.name, data, sigma = 0.3, alpha = 1,
                 burn.in = 1000, steps = 5000)

s <- MMG.cut.graph(r, method = "THRESHOLD", select = "UP",
                  threshold = 0.2, descriptions = NA)

MMG.make.dot(s, file.name, type = "DIRECTED",
             selection, rem.loops = FALSE,
             weight.max = NA, weight.min = NA)
```

Arguments

<code>r</code>	List returned by <code>MMG.compute</code> , cf. <i>infra</i> .
<code>s</code>	List returned by <code>MMG.cut.graph</code> , cf. <i>infra</i> .
<code>file.name</code>	<code>file.name</code> used in <code>MMG.compute</code> is the file containing the network alongside the (i.e., proteomic) data. See details for complementary details. Used in <code>MMG.make.dot</code> , it indicates the file that must be output.
<code>data</code>	Allows the user to override the data values using an additional file. This may spare the user the inconvenience of re-generating network files all the time.
<code>sigma</code>	<code>sigma</code> is the standard deviation of the unchanged category (Gaussian).
<code>alpha</code>	<code>alpha</code> controls how important is the contribution of the network in the Mixture Model. Zero means that the contribution is high whereas the model tends towards a normal mixture model when <code>alpha</code> becomes large.
<code>burn.in</code>	<code>burn.in</code> is the number of steps of burn-in of the Gibbs sampler - if one feels that it can be useful.
<code>steps</code>	<code>steps</code> Number of steps that must be performed by the Biggs sampler.

<code>method</code>	<code>method</code> determines the method that will be used to identify the clusters of nodes that behave consistently, i.e., that belongs to a same class. Possible methods are "LIKELIEST", "THRESHOLD", and "ENTROPY". See <code>threshold</code> and details.
<code>select</code>	<code>select</code> can be "DOWN", "UNCHANGED", or "UP" depending on the class one desire to inspect.
<code>threshold</code>	<code>threshold</code> controls the stringency that is used to determine whether a node belongs to the class selected by <code>select</code> . Depending on <code>method</code> , the meaning of this parameter changes. See details.
<code>descriptions</code>	A file name. The corresponding file must contain the list of descriptions of the network's nodes, one per line, in the same order as <code>file.name</code> 's. See details.
<code>type</code>	It can be "DIRECTED" or "UNDIRECTED". Because the DOT format understands both types of graph. "UNDIRECTED" makes more sense if the original topology as specified by <code>file.name</code> in <code>MMG.compute</code> was undirected. It can also make sense in the case of directed graphs if it makes them look better in GraphViz. The graph is made undirected if necessary.
<code>selection</code>	This parameter is used to tell <code>MMG.make.dot</code> which subset of the network must be represented. It is generally computed using the value returned by <code>MMG.cut.graph</code> .
<code>rem.loops</code>	A boolean value controlling whether loops should be avoided.
<code>weight.max</code>	This parameter discards any edge that has a weight exceeding it. (Not implemented yet.)
<code>weight.min</code>	This parameter discards any edge that has a weight lower than it. (Not implemented yet.)

Details

`MMG.compute` runs a Gibbs sampler and returns useful information regarding the posterior probabilities of belonging to such-and-such category (down-regulated, unchanged, and up-regulated), assuming the model described in Guido et al., Bioinformatics (2008).

The file containing the data must be made of lines having the following format:

```
n value neighbour1 weight1 ... neighbourN weightN
```

`n` is the number of the line (this is required to make the file easier to read to a human being - it could be, in principle, any number); `value` is the logarithm of experimental measurement (0 meaning no change, and NA the value was not got), the base utilised does not matter very much but `sigma` should be chosen accordingly; the `neighbourIs` are the neighbours of the node `n` alongside the weights `weightIs` of the edges that connect it to them. `N` needs not be the same all throughout the file.

NB: It must be noted that contrarily to the weights used in Croes et al., JMB 2006, here a high weight implies high impact. The weight used by Croes et al. must therefore be inverted before being used.

`MMG.compute` prints some information regarding the parameters

`lambda_down` and `lambda_up` Average and standard deviation

Shannon entropy Average, standard deviation, and stem-and-leaves diagram. This could give a quick flavour of how much uncertainty there is throughout the network.

The file whose name is given by `descriptions` must contain lines, each of which, say the i -th line, describe the content of the node number i of the network. For instance,

```
aldehyde dehydrogenase
acetyl-coenzyme a synthetase
alcohol dehydrogenase
pyruvate dehydrogenase
dihydrolipoamide acetyltransferase
pyruvate kinase
...
```

The parameter `threshold` helps to select the nodes that belong to the desired category (parameter `select`). Once the selection is carried out, `MMG.cut.graph` only has to generate the subgraph that contains the selection. In the following T denotes the parameter `threshold`.

"LIKELIEST" A node belongs to class C if and only if the probability p_C of belonging to the class C is greater than that of belonging to the other classes $D1$ and $D2$ ($p_C > p_{D1}$ and $p_C > p_{D2}$), and if $p_C > T$.

"THRESHOLD" A node belongs to class C if and only if the probability p_C of belonging to the class C is greater than that of belonging to the other classes $D1$ and $D2$ by at least T : $p_C > p_{D1} + T$ and $p_C > p_{D2} + T$.

"ENTROPY" A node belongs to class C if and only if the probability p_C of belonging to the class C is the greatest ($p_C > p_{D1}$ and $p_C > p_{D2}$) and if the Shannon entropy is below T .

Value

`MMG.compute` returns a list

```
data          The data as read from the file file.name. This is a list:
              n.nodes Number of nodes
              adjacency.matrix Contains a matrix each row of which is the list of values
                              n value neighbour1 weight1 ... neighbourN weightN
              lengths Number of fields of the rows
samples       The samples drawn from the Gibbs sampler (n x 3 matrix)
lup           The series of lambda_up
ldown        The series of lambda_down
entropies     List of the Shannon entropies
components   To which component belongs the nodes (0 means not selected)
descriptions  The vector of the descriptions as given in the file passed as descriptions
normal-bracket136bracket-normal
MMG.make.dot does not return anything.
```

Note

Josselin Noirel, Department of Chemical and Process Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD - The United Kingdom,

Based on an original implementation by Guido Sanguinetti <URL: <http://sheffield.ac.uk/guido/>>.

Author(s)

Josselin Noirel, <j.noirel@sheffield.ac.uk>

References

Sanguinetti, Noirel, and Wright., MMG: a probabilistic tool to identify submodules of metabolic pathways, *Bioinformatics* (2008)

Examples

```
## Not run:
r <- MMG.compute(file.name = "NostocData/R_net.dat",
                 steps = 100000, burn.in = 1000,
                 sigma = 0.3, alpha = 1)
## End(Not run)
## Not run: n <- r$dat$n.nodes
## Not run:
s <- MMG.cut.graph(r, descriptions = "NostocData/R_descr.dat",
                  method = "THRESHOLD", threshold = 0.15, select = "UP")
## End(Not run)
## Not run: l <- (1:n)[s$components != 0]
## Not run:
MMG.make.dot(r, file.name = "nostoc.dot", selection = l, type = "UNDIRECTED",
             rem.loops = TRUE)
## End(Not run)
```

Index

*Topic **models**

MMG.compute, 2

*Topic **package**

MMG-package, 1

MMG (*MMG-package*), 1

MMG-package, 1

MMG.compute, 2

MMG.cut.graph (*MMG.compute*), 2

MMG.make.dot (*MMG.compute*), 2