Package 'fgga'

November 25, 2021

Type Package

Title Hierarchical ensemble method based on factor graph

Version 1.2.0

Date 2021-05-06

biocViews Software, StatisticalMethod, Classification, Network, NetworkInference, SupportVectorMachine, GraphAndNetwork, GO

Description Package that implements the FGGA algorithm. This package provides a hierarchical ensemble method based ob factor graphs for the consistent GO annotation of protein coding genes. FGGA embodies elements of predicate logic, communication theory, supervised learning and inference in graphical models.

Depends R (>= 4.1), RBGL

Imports graph, stats, e1071, methods, gRbase, jsonlite, BiocFileCache, curl

Suggests knitr, rmarkdown, GOstats, PerfMeas, GO.db, BiocGenerics

License GPL-3

URL https://github.com/fspetale/fgga

Encoding UTF-8

LazyData false

VignetteBuilder knitr

NeedsCompilation no

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

git_branch RELEASE_3_14

git_last_commit f575f34

git_last_commit_date 2021-10-26

Date/Publication 2021-11-25

Author Spetale Flavio [aut, cre], Elizabeth Tapia [aut, ctb]

Maintainer Spetale Flavio <spetale@cifasis-conicet.gov.ar>

R topics documented:

`fgga-package` *FGGA: Factor Graph Gene ontology Annotation.*

Description

FGGA is a graph-based machine learning approach for the automated and consistent GO annotation of protein coding genes. The input is a set of GO-term annotated protein coding genes previously characterized in terms of a fixed number of user-defined features, including the presence/absence of PFAM domains, physical-chemical properties, presence of signal peptides, among others. The set of GO-terms defines the output GO subgraph. A hierarchical ensemble (SVMs) machine learning model is generated. This model can be used to predict the GO subgraph annotations of uncharacterized protein coding genes. Individual GO-term annotations are accompanied by maximum a posteriori probability estimates issued by the native message passing algorithm of factor graphs.

Author(s)

Flavio E. Spetale, *Javier Murillo* and *Elizabeth Tapia*

BioInformatics

Cifasis-Conicet

<spetale@cifasis-conicet.gov.ar>

Maintainer: *Flavio E. Spetale*

References

Spetale F.E., et al. **A Factor Graph Approach to Automated GO Annotation**. *PLoS ONE* (2016). https://doi.org/10.1371/journal.pone.0146986.

Spetale Flavio E., et al. **Consistent prediction of GO protein localization**. *Scientific Report* (2018). https://doi.org/10.1038/s41598-018-26041-z.

See Also

`fgga`, `fgga2bipartite`, `sumProduct`, `svm`

CfData — *A set of characterized protein coding genes from the Cannis familiaris organism annotated to a target GO subgraph considering both experimental and electronic evidence.*

Description

The CfData dataset consists of a list containing the following:

$dxCf: characterizations of *6962* protein coding genes in terms of *72* physico-chemical properties of their amino acid sequences. These sequences, obtained from the Uniprot database, are annotated to *36* GO-terms of the GO Molecular Function (GO-MF) ontology subdomain.

$tableCfGO: a set of *6962* protein coding genes annotated to GO-MF target classes. Genes are identified by their **Uniprot** ID mappings which are obtained with the org.Cf.eg.db annotation package set to work with both experimental and electronic evidence. Additionally, only those GO-MF terms with at least *500* annotated genes were preserved.

$graphCfGO: the target **GO-MF** subgraph obtained with the org.Cf.eg.db annotation package set to work with the set of **GO-MF** target classes.

$indexGO: two arrays of Uniprot ID mappings defining the train-test partition of the set *6962* protein coding genes annotated to **GO-MF** terms.

$nodesGO: labels of the GO-MF subgraph.

$varianceGOs: a vector labeled with the variance of each **GO-MF** term.

Usage

```
data("CfData")
```

Format

A list with five named entries containg:

dxCf A matrix (6962 rows x 72 columns) containing the characterized proteins.

graphCfGO An adjacency binary matrix (36 rows x 36 columns) corresponding to the GO-MF subgraph.

indexGO A list with two named entries: indexTrain and indexTest each containing a numeric vector.

tableCfGO A binary matrix (6962 rows x 36 columns) containing GOs associated with a protein.

nodesGO A numerical vector containing the nodes of the GO-MF subgraph.

Source

Uniprot Taxonomy: 9615

https://www.uniprot.org/uniprot/?query=taxonomy:9615

Package: org.Cf.eg.db - Version: 3.8.2

https://bioconductor.org/packages/org.Cf.eg.db/

Examples

```
data(CfData)

## list objects included
ls(CfData)
# [1] "dxCf"  "graphCfGO" "indexGO"   "nodesGO"   "tableCfGO"

# Physico-chemical properties of each protein
head(CfData[["dxCf"]])

# GO-MF node labels, GO-terms, of each protein
head(CfData[["tableCfGO"]])
```

`createFolds` *Data splitting function useful for binary classification tasks*

Description

createFolds splits binary classification data into k-folds.

Usage

```
createFolds(target, k_fold = 10)
```

Arguments

`target`	A binary vector of a GO class
`k_fold`	An integer for the number of folds

Details

A random sampling is performed on binary classification data. A set of k data folds reflecting the original class balance is obtained.

Value

list of row position integers corresponding to the training data

Author(s)

Flavio E. Spetale and Pilar Bulacio <spetale@cifasis-conicet.gov.ar>

References

Hyndman and Athanasopoulos (2013), Forecasting: principles and practice. https://www.otexts.org/fpp

Examples

```
data(CfData)

createFolds(CfData[["tableCfGO"]][ ,"GO:0005515"], k_fold = 2)
```

`fgga` *Factor Graph GO Annotation model*

Description

A hierarchical graph-based machine learning model for the consistent GO annotation of protein coding genes.

Usage

```
fgga(graphGO, tableGOs, dxCharacterized, dxTestCharacterized,
    kFold, kernelSVM, tmax, epsilon)
```

Arguments

`graphGO`	A graphNEL graph with 'm' GO node labels.
`tableGOs`	A binary matrix with 'n' proteins (rows) by 'm' GO node labels (columns).
`dxCharacterized`	A data frame with 'n' proteins (rows) by 'f' features (columns).
`dxTestCharacterized`	A data frame with 'k' proteins (rows) by 'f' features (columns).
`kFold`	An integer for the number of folds.
`kernelSVM`	The kernel used to calculate the variance (default: radial).
`tmax`	An integer indicating the maximum number of iterations (default: 200).
`epsilon`	An integer that represents the convergence criteria (default: 0.001).

Details

The **FGGA model** is built in two main steps. In the first step, a core Factor Graph (FG) modeling hidden GO-term predictions and relationships is created. In the second step, the FG is enriched with nodes modeling observable GO-term predictions issued by **binary SVM classifiers**. In addition, probabilistic constraints modeling learning gaps between hidden and observable GO-term predictions are introduced. These gaps are assumed to be independent among GO-terms, locally additive with respect to observed predictions, and zero-mean Gaussian. **FGGA predictions** are issued by the native iterative **message passing algorithm** of factor graphs.

Value

A named matrix with 'k' protein coding genes (rows) by 'm' GO node labels (columns) where each element indicates a probabilistic prediction value.

Author(s)

Flavio E. Spetale and Elizabeth Tapia <spetale@cifasis-conicet.gov.ar>

References

Spetale F.E., Tapia E., Krsticevic F., Roda F. and Bulacio P. "A Factor Graph Approach to Automated GO Annotation". PLoS ONE 11(1): e0146986, 2016.

Spetale Flavio E., Arce D., Krsticevic F., Bulacio P. and Tapia E. "Consistent prediction of GO protein localization". Scientific Report 7787(8), 2018

See Also

fgga2bipartite, sumProduct, svmGO

Examples

```
data(CfData)

mygraphGO <- as(CfData[["graphCfGO"]], "graphNEL")

dxCfTestCharacterized <- CfData[["dxCf"]][CfData[["indexGO"]]$indexTest[1:2], ]

myTableGO <- CfData[["tableCfGO"]][
                    CfData[["indexGO"]]$indexTrain[1:300], ]

dataTrain <- CfData[["dxCf"]][
                    CfData[["indexGO"]]$indexTrain[1:300], ]

fggaResults <- fgga(graphGO = mygraphGO,
                tableGOs = myTableGO, dxCharacterized = dataTrain,
                dxTestCharacterized = dxCfTestCharacterized, kFold = 2,
                tmax = 50, epsilon = 0.05)
```

---

fgga2bipartite *Forney Factor Graph model*

Description

fgga2bipartite builds a Forney Factor Graph from a FGGA model.

Usage

```
fgga2bipartite(graphGO)
```

Arguments

graphGO	A graphNEL graph with 'm' GO node labels.

Details

The **Gene Ontology** (GO) is structured as a directed acyclic graph (DAG) with nodes (GO-terms) representing gene functions and edges characterizing relationships between nodes. A variety of relationships are possible (currently 8). To compute GO-term predictions perfectly aware of GO-term relationships, a Forney Factor Graph is required. Hence, GO-terms are mapped to binary variable nodes, and relationships to logical factor nodes.

Value

A binary matrix with *2*m* rows by *2*m-1* columns where *m* is the quantity of GO node labels.

Author(s)

Flavio E. Spetale <spetale@cifasis-conicet.gov.ar>

References

F. Spetale, J. Murillo, E. Tapia, D. Arce, S. Ponce, and P. Bulacio, "Formal modeling of gene ontology annotation predictions based on factor graphs," Journal of Physics: Conference Series, vol. 705, no. 1, p. 012001, 2016.

Spetale F.E., Tapia E., Krsticevic F., Roda F. and Bulacio P. "A Factor Graph Approach to Automated GO Annotation". PLoS ONE 11(1): e0146986, 2016.

Spetale Flavio E., Arce D., Krsticevic F., Bulacio P. and Tapia E. "Consistent prediction of GO protein localization". Scientific Report 7787(8), 2018

Examples

```
data(CfData)

graphGO <- as(CfData$graphCfGO, "graphNEL")
fgga2bipartite(graphGO)
```

`preCoreFG` *Transitive closure processing of a GO DAG*

Description

preCoreFG ensures the transitive closure of inference paths -serial concatenation of relationships- in a GO DAG.

Usage

```
preCoreFG(myGOnames, domains = "goPlus")
```

Arguments

myGOnames	A vector with 'm' GO node labels
domains	A string that indicates which subdomains will be used. Values: "BP-MF", "MF-CC", "BP-CC" or "goPlus" (default, "BP-MF-CC")

Details

Non-transitive relationships in GO DAG's may lead to non-transitive inference paths precluding the free propagation and consistency checking of GO annotations. A transitive closure screening process over GO DAG's relationships is required before the construction of Forney Factor Graphs. Serial concatenation of relationships leading to non-transitive inference paths in a GO DAG are conformed by removing the most specific relationship.

Value

A graphNEL graph with 'm' GO node labels.

Author(s)

Flavio E. Spetale <spetale@cifasis-conicet.gov.ar>

References

Spetale Flavio E., Arce D., Krsticevic F., Bulacio P. and Tapia E. "Consistent prediction of GO protein localization". Scientific Report 7787(8), 2018

See Also

fgga2bipartite

Examples

```
data(CfData)

myGOs <- c(CfData[["nodesGO"]], "GO:1902494", "GO:0032991", "GO:1990234",
           "GO:0005575")


mygraphGO <- preCoreFG(myGOs, domains = "MF-CC")
```

sumProduct *Message passing algorithm between nodes of the Forney Factor Graph*

Description

msgFGGA operates in Forney Factor Graphs and computes approximate maximum a posteriori (MAP) estimates of hidden GO variable nodes (GO-terms).

Usage

```
msgFGGA(matrixFGGA, obsValueGOs, graphGO, tmax = 200,
        epsilon = 0.001)
```

Arguments

matrixFGGA	A binary matrix with FGGA model of the class 'fgga.'
obsValueGOs	A named vector with 'm' probabilistic prediction values for a protein coding gene.
graphGO	A graphNEL graph with 'm' GO node labels.
tmax	An integer indicating the maximum number of iterations (defualt: 200).
epsilon	An integer that represents the convergence criteria (default: 0.001)

Details

Starting from GO-term predictions at observable variable nodes, probability distribution functions modelling the learning noise of individual GO-terms, a user-defined number of iterations (maximum 200), a user-defined threshold for the convergence of predictions (maximum 0.001), and the structure of the Forney Factor Graph, the **msgFGGA** delivers approximate maximum a posteriori (MAP) estimates of hidden GO variable nodes (GO-terms).

Value

A named vector with 'm' consistent probabilistic predictions for a protein coding genes.

Author(s)

Flavio E. Spetale and Elizabeth Tapia <spetale@cifasis-conicet.gov.ar>

References

Kschischang FR, Frey BJ, Loeliger H.-A. Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theor. 47, 498–519 (2001).

Yedidia JS. Message-passing algorithms for inference and optimization. Journal of Statistical Physics 145, 860–890 (2011).

Spetale FE, Tapia E, Krsticevic F, Roda F, Bulacio P (2016). A Factor Graph Approach to Automated GO Annotation. PLOS ONE 11(1): e0146986

See Also

tableTPG

Examples

```
data(CfData)

mygraphGO <- as(CfData[["graphCfGO"]], "graphNEL")

myTableGO <- CfData[["tableCfGO"]][
                    CfData[["indexGO"]]$indexTrain[1:500], ]

modelSVMs <- lapply(CfData[["nodesGO"]], FUN = svmTrain, tableGOs = myTableGO,
                    dxCharacterized = CfData[["dxCf"]],
                    graphGO = mygraphGO, kernelSVM = "radial")

rootGO <- leaves(mygraphGO, "in")

varianceGOs <- CfData[["varianceGOs"]]

dxTestCharacterized <- CfData[["dxCf"]][
                                sample(1:dim(CfData[["dxCf"]])[2], 2), ]

matrixGOTest <- svmGO(svmMoldel = modelSVMs,
                      dxCharacterized = dxTestCharacterized,
                      rootNode = rootGO, varianceSVM = varianceGOs)

modelFGGA <- fgga2bipartite(mygraphGO)

matrixFGGATest <- t(apply(matrixGOTest, MARGIN = 1, FUN = msgFGGA,
                    matrixFGGA = modelFGGA, graphGO= mygraphGO,
                    tmax = 50, epsilon = 0.1))
```

---

svmGO *GO-term predictions by binary SVM classifiers*

Description

svmGO delivers soft GO-term predictions based on binary SVM classification models.

Usage

```
svmGO(svmMoldel, dxCharacterized, rootNode, varianceSVM)
```

Arguments

svmMoldel	A list of object of class “svm" created by svm.

dxCharacterized
: A data frame with 'n' protein coding genes (rows) by 'f' features (columns).

rootNode
: A character indicating the root of the graph.

varianceSVM
: A vector named with the variance of GO node labels.

Details

Binary SVM predictions are supplemented with their corresponding margins. These margins are used to model the additive zero-mean Gaussian learning noise that corrupts ideal but hidden GO-term predictions. These ideal predictions are embedded in hidden variable nodes of the Forney Factor Graph.

Value

svmGO
: A named vector of predicted values for a protein sequence.

Author(s)

Flavio E. Spetale, Pilar Bulacio and Javier Murillo <spetale@cifasis-conicet.gov.ar>

References

Chang, Chih-Chung and Lin, Chih-Jen: LIBSVM: a library for Support Vector Machines http://www.csie.ntu.edu.tw/~cjlin/libsvm

Eisner R, Poulin B, Szafron D, Lu P, Greiner R. Improving protein function prediction using the hierarchical structure of the Gene Ontology. In: Proc. IEEE CIBCB; 2005. p. 1–1

Spetale FE, Tapia E, Krsticevic F, Roda F, Bulacio P (2016). A Factor Graph Approach to Automated GO Annotation. PLOS ONE 11(1): e0146986

See Also

svmTrain

Examples

```
data(CfData)

mygraphGO <- as(CfData[["graphCfGO"]], "graphNEL")

modelSVMs <- lapply(CfData[["nodesGO"]][1:4], FUN = svmTrain,
                    tableGOs = CfData[["tableCfGO"]],
                    dxCharacterized = CfData[["dxCf"]],
                    graphGO = mygraphGO, kernelSVM = "radial")

rootGO <- leaves(mygraphGO, "in")

varianceGOs <- CfData[["varianceGOs"]]

# SVM testing in four GO-terms
dxTestCharacterized <- CfData[["dxCf"]][
```

```
                                sample(1:dim(CfData[["dxCf"]])[1], 20), ]

matrixGOTest <- svmGO(svmMoldel = modelSVMs,
                      dxCharacterized = dxTestCharacterized,
                      rootNode = rootGO, varianceSVM = varianceGOs)
```

---

svmTrain *Binary SVM classification models for individual GO-term predictions*

Description

svmTrain delivers a set of binary SVM classifiers for different GO-terms.

Usage

```
svmTrain(nodeGraph, tableGOs, dxCharacterized, graphGO,
        kernelSVM = "radial")
```

Arguments

nodeGraph	A character indicating a GO node label
tableGOs	A binary matrix with 'n' proteins (rows) by 'm' GO node labels (columns).
dxCharacterized	A data frame with 'n' protein coding genes (rows) by 'f' features (columns).
graphGO	A graphNEL graph with 'm' GO node labels.
kernelSVM	The kernel used to calculate the variance (default: radial).

Details

Starting from sets of positively annotated protein sequences to different GO-terms in a GO subgraph, corresponding sets of negatively annotated protein sequences are computed using the inclusive separation policy proposed by Eisner et al. Training datasets for each GO-term are used to train binary Support Vector Machine (SVM) classifiers with a variety of kernel options.

Value

svmTrain	A list of objects of "svm" class containing the fitted model.

Author(s)

Flavio E. Spetale, Pilar Bulacio and Javier Murillo <spetale@cifasis-conicet.gov.ar>

References

Chang, Chih-Chung and Lin, Chih-Jen: LIBSVM: a library for Support Vector Machines http://www.csie.ntu.edu.tw/~cjlin/libsvm

Eisner R, Poulin B, Szafron D, Lu P, Greiner R. Improving protein function prediction using the hierarchical structure of the Gene Ontology. In: Proc. IEEE CIBCB; 2005. p. 1–1

Spetale FE, Tapia E, Krsticevic F, Roda F, Bulacio P (2016). A Factor Graph Approach to Automated GO Annotation. PLOS ONE 11(1): e0146986

See Also

svmGO

Examples

```
data(CfData)

mygraphGO <- as(CfData[["graphCfGO"]], "graphNEL")

# SVM training in four GO-terms
modelSVMs <- lapply(CfData[["nodesGO"]][1:4], FUN = svmTrain,
                    tableGOs = CfData[["tableCfGO"]],
                    dxCharacterized = CfData[["dxCf"]],
                    graphGO = mygraphGO, kernelSVM = "radial")
```

---

tableTPG *Valid configurations for hidden variable nodes in a Forney Factor Graph*

Description

tableTPG provides valid configurations of hidden variable nodes at logical function nodes in a Forney Factor Graph under the True Path Graph (TPG) constraint.

Usage

```
tableTPG(att)
```

Arguments

att	An integer indicating the number of GO nodes involved

Details

Valid configurations of hidden variable nodes at logical function nodes enable messaging passing across the Forney Factor Graph. The TPG constraint is defined as: "If the child GO node describes the protein, then all its parent terms must also apply to that protein; and if a GO node does not describe a protein, then all its descendant GO nodes must not describe it". The TPG constraint governs the structure of the GO-DAG and the inference process in the associated Forney Factor Graph.

Value

A binary matrix with *(n-1)^2+1* rows by *n+1* columns where *n = attr*

Author(s)

Flavio E. Spetale, Pilar Bulacio and Javier Murillo <spetale@cifasis-conicet.gov.ar>

References

Tanoue J, Yoshikawa M, Uemura S (2012). The Gene Around GO viewer. Bioinformatics 18(12): 1705–1706.

Spetale FE, Tapia E, Krsticevic F, Roda F, Bulacio P (2016). A Factor Graph Approach to Automated GO Annotation. PLOS ONE 11(1): e0146986

Examples

```
tableTPG(3)
```

---

varianceGO *The variance of the gaussian learning noise at individual GO-terms*

Description

varianceGO estimates the variance of gaussian distributions modeling the additive learning noise that corrupts ideal GO-term predictions.

Usage

```
varianceGO(tableGOs, dxCharacterized, kFold, graphGO, rootNode,
           kernelSVM = "radial")
```

Arguments

`tableGOs`	A binary matrix with 'n' protein coding genes (rows) by 'm' GO node labels (columns).
`dxCharacterized`	A data frame with 'n' protein coding genes (rows) by 'f' features (columns).
`kFold`	An integer for the number of folds.

graphGO	A graphNEL graph with 'm' GO node labels.
rootNode	A character indicating the root of the graph.
kernelSVM	The kernel used to calculate the variance (default: radial).

Details

Under the assumption of symmetrical (Gaussian) conditional probability distributions for observable variable node predictions y_i over a hidden variable node annotations x_i, variances η_i can be estimated using a validation dataset of positively annotated samples. Let *D* be a validation dataset with L^+ positively annotated samples

$$\hat{\eta_i} = 1/(L^+ - 1) * \sum_{i=1}^{L}(x_i - y_i)$$

where $x_i = 1$ is the positive annotation of the l-th data sample to the ith GO-term and y_i is the corresponding real-valued classifier (SVM) prediction.

Value

A vector named with the variance of each GO node.

Author(s)

Flavio E. Spetale and Javier Murillo <spetale@cifasis-conicet.gov.ar>

References

Spetale FE, Tapia E, Krsticevic F, Roda F, Bulacio P (2016). A Factor Graph Approach to Automated GO Annotation. PLOS ONE 11(1): e0146986

Examples

```
data(CfData)

mygraphGO <- as(CfData[["graphCfGO"]], "graphNEL")

rootGO <- leaves(mygraphGO, "in")

mygraphGO <- subGraph(c("GO:0140110", "GO:0098772", "GO:0003674"), mygraphGO)

myTableGO <- CfData[["tableCfGO"]][
                    CfData[["indexGO"]]$indexTrain,
                    c("GO:0140110", "GO:0098772", "GO:0003674")]

varianceGOs <- varianceGO(tableGOs = myTableGO,
                          dxCharacterized = CfData[["dxCf"]],
                          kFold = 2, graphGO = mygraphGO,
                          rootNode = rootGO, kernelSVM = "radial")
```

Index