OXFORD JOURNALS

Bioinformatics

HCGene: a software tool to support the hierarchical classification of genes

Journal:	*Bioinformatics*
Manuscript ID:	BIOINF-2007-1809.R1
Category:	Applications Note
Date Submitted by the Author:	n/a
Complete List of Authors:	Valentini, Giorgio; Universita' degli Studi di Milano, DSI - Dipartimento di Scienze dell' Informazione Cesa-Bianchi, Nicolo'; Universita' degli Studi di Milano, DSI - Dipartimento di Scienze dell' Informazione
Keywords:	hierarchical classification of genes, graph processing, Gene Ontology, data integration, open source software
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
hcgene-bio-revised.tex bioinfo.cls natbib.bst natbib.sty	

BIOINFORMATICS

Vol. 00 no. 00
Pages 1–2

HCGene: a software tool to support the hierarchical classification of genes

Giorgio Valentini[*], Nicolò Cesa-Bianchi

DSI; Dip. di Scienze dell'Informazione, Università degli Studi di Milano, Via Comelico 39, Italy.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Summary: The R package *HCGene* (Hierarchical Classification of Genes) implements methods to process and analyze the Gene Ontology and the FunCat taxonomy in order to support the functional classification of genes. *HCGene* allows the extraction of subgraphs and subtrees related to specific biological problems, the labelling of genes and gene products with multiple and hierarchical functional classes, and the association of different types of bio-molecular data to genes for learning to predict their functions.

Availability: http://homes.dsi.unimi.it/∼valenti/SW/hcgene/download/hcgene_1.0.tar.gz

Contact: valentini@dsi.unimi.it

Supplementary Information:
http://homes.dsi.unimi.it/∼valenti/SW/hcgene

The capability of assigning functions to unannotated gene products using large-scale bio-molecular data is a key issue in functional genomics and bioinformatics [Dopazo, 2006].

Ontologies such as *Gene Ontology* [Harris et al., 2004] and *FunCat* [Ruepp et al., 2004] encode binary relations among functional classes. The graph induced on the class nodes by these relations is a DAG (directed acyclic graph) for the Gene Ontology and a tree for FunCat. Annotations for genes and gene products are provided for both ontologies at different degrees of resolution and reliability, and typically involve multiple classes. Thus gene function prediction can be naturally viewed as a hierarchical classification problem with structured labels involving multiple and partial paths [Barutcuoglu et al., 2006]. Yet, the majority of computational approaches for the prediction of gene functions disregard the hierarchical structure of gene classes and solve the problem using a "flat" multiclass predictor [Brown et al., 2000, Pavlidis et al., 2002].

A full-fledged hierarchical classification approach to gene prediction calls for a new generation of scalable software tools. Graphs with thousands of nodes and edges must be processed in order to extract subgraphs related to the specific biological process under investigation. Multiple functional classes must be properly associated to genes and gene products, according to the specific hierarchy being considered. Finally, gene products must be associated to the different data types (e.g., gene expression data, phylogenetic or protein interaction data) used to infer the function of unknown genes.

Several software tools have been developed for browsing, searching, and processing the Gene Ontology (see, e.g., www.geneontology.org for an updated list). In this work we describe *HCGene* (Hierarchical Classification of Genes), a software library performing data pre-processing in tasks of hierarchical gene classification. The distinctive features of *HCGene* with respect to previous tools are: the integration of data, the processing of multilabels and graphs, and the addition of a library to process and analyze the FunCat taxonomy.

We can divide in three main steps the pre-processing of data and classes in a gene classification task.

1. *Processing of functional classes of genes*: Construction of GO graphs and FunCat trees; extraction of subgraphs related to the problem under investigation.
2. *Labeling gene products with functional classes*: Association of multiple functional classes from the GO or FunCat ontologies to gene products.
3. *Association of gene products to data*: Association of the gene products to their corresponding biological data.

The *HCGene* R package provides methods and functionalities to support all of the above steps. Moreover, it allows to analyze the properties of GO graphs and FunCat trees associated to both human and specific model organisms (such as *S. cerevisiae*, *Mus musculus* and *Arabidopsis thaliana*). Methods for computing various statistics on the structure of GO and FunCat and their associated gene products are also included in the library. Note that *HCGene* does not include any algorithm for hierarchical classification. Its purpose is to rather offer tools that facilitate the use of gene classification algorithms.

The functionalities related to the processing of the GO ontology have been implemented using the *Bioconductor* packages [Gentleman et al., 2004], graph, GO, GOstats, and Rgraphviz. The part of the library related to FunCat has been built from scratch using the hierarchical schemes and the functional annotations obtained from the MIPS website (mips.gsf.de).

HCGene has been primarily designed for the supervised hierarchical classification of genes/gene products. However, it can also be used with unsupervised and semi-supervised methods to incorporate a priori biological knowledge about functional classes of genes [Lottaz et al., 2007, Tai and Pan, 2007].

The main functionalities of the software library can be summarized as follows.

- *Graph processing*: construction of hierarchical structures based on graphs and trees. This part includes methods to analyze the structure and the relationships between functional classes (e.g., distribution of

[*] to whom correspondence should be addressed

node and classes with respect to their depth, in and out-degree, cardinality of classes and distribution of leaves at different levels). It also includes methods to extract biologically meaningful structures from GO DAGs and FunCat trees.

• *Multilabel generation*: extraction of the most specific annotations and derivation of the full annotation of genes; building of the multilabel for each gene using compact representations; mapping functions to associate gene names or identifiers (e.g., ORF ID or EntrezGene IDs) to functional classes

• *Data processing*: This part includes methods to associate gene names to different types of data, methods to select positive and negative examples for each class according to different strategies (see below for a discussion of selection strategies), and methods to build data related to specific functional classes.

Moreover, the library provides functions to graphically show the results of statistical analyses, and to draw subgraphs of GO and FunCat ontologies.

The mapping of genes to their corresponding classes has been performed according to the *Gene Ontology Annotation (GOA)* consortium [Camon et al., 2006], and according to the MIPS (mips.gsf.de) annotations for FunCat. Usually, genes are annotated with the lowest level (that is, most specific) terms of GO and FunCat. These terms are associated to nodes in the underlying graph (DAG or tree). The multilabel associated with a gene is the set of all terms that can be associated with it. To obtain this multilabel, we start from the set of initial nodes, corresponding to the most specific terms, and add to them all the nodes that belong to any path from the initial nodes to a root (in a DAG a root is any node having no parents). This “transitive closure” operation is based on the notion of consistency for multilabels: if the multilabel of a gene includes a node, then it must also include all of its ancestors.

A typical approach to hierarchical gene classification is to associate a binary classifier to each node of the graph and then use some global criterion to infer a consistent multilabel from the binary classifications performed at each node. Binary classifiers perform better when trained on a mixture of positive and negative examples (we call *example* a gene, or gene product, together with its associated multilabel). Thus, one of the issues with this approach is to determine the set of negative examples each node classifier should be trained on. However, gene sets annotated with GO and FunCat classes almost never include explicit negative examples for specific nodes. As a consequence, any example whose multilabel does not include a given node is a candidate negative example for training the associated node classifier. In practice, different strategies for selecting negative examples are used in order to carefully balance the fraction of negative examples used in training. Our library implements the three following strategies.

1. A negative example for a node is any example whose multilabel does not include that node.
2. A negative example for a node is any example whose multilabel does not include that node and any of its ancestors.
3. A negative example for a node is any example whose multilabel does not include that node and includes at least one of its parents.

The discussion of these strategies is beyond the scope of this paper: we only recall that most of the works on the functional classification of genes adopt the first and the second strategy [Pavlidis et al., 2002, Barutcuoglu et al., 2006, Lewis et al., 2006]. Other strategies could be equally well motivated. For instance, the presence of incomplete annotations justifies requiring that a negative example for a node have a multilabel including at least a sibling of that node. Such alternative strategies will be considered in future implementations.

HCGene provides several methods to extract subgraphs from GO or subtrees from FunCat, and to automatically associate corresponding genes and data. Subgraphs can be selected according to the depth of the nodes (in FunCat trees) or to the minimum distance from roots (in GO DAGs). A node can be also selected based on the number of genes that are associated to it. For example, we might be interested in classifying, in the FunCat tree, genes related to the amino acid metabolism in the yeast while considering only classes with more than ten annotated genes. With a few lines of R code we can extract the corresponding subtree, as well as the yeast genes associated to each FunCat class and the corresponding gene expression or phylogenetic data, or any other data supplied by the user.

Finally, *HCGene* provides methods to analyze the statistical properties of FunCat and GO ontologies. For example, one can compute the distribution of the number of labels in the *GO Biological Process* ontology associated to each gene in *A. thaliana*, considering only genes reliably annotated with TAS (Traceable Author Statement) evidence. One can also extract a subgraph rooted at a specific *GO* node, and select nodes with genes annotated with IGI (Inferred from Genetic Interaction) or IPI (Inferred from Physical Interaction) evidence in *H. sapiens*. The structural characteristics of the resulting graph and associated genes can then be analyzed (e.g., indegree and outdegree node distributions, distribution of the length of the shortest path from the root, distribution of the cardinality of the functional classes and multilabels).

The Supplementary Information available on line offers more details about the methods implemented in the software library, several examples of application, and a detailed reference manual. Considering that some functionalities of *HCGene* are not just specific to the functional classification of genes, future developments of this work will be the adaptation and the extension of the package to other applications characterized by the presence of structured domains.

REFERENCES

Z. Barutcuoglu, R.E. Schapire, and O.G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

M. Brown et al. Knowledge-base analysis of microarray gene expression data by using support vector machines. *PNAS*, 97(1):262–267, 2000.

E. Camon et al. The Gene Ontology Annotation (GOA) database. In *Silico Genomics and Proteomics*. Nova Science, New York, 2006.

J. Dopazo. Functional interpretation of microarray experiments. *OMICS*, 3(10), 2006.

R. Gentleman et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10), 2004.

M.A. Harris et al. The Gene Ontology (GO) database and informatics resource. *Nucleic Acid Res.*, 32:D258–D261, 2004.

D.P. Lewis, T. Jebara, and W.S. Noble. Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics*, 22(22):2753–2760, 2006.

C. Lottaz et al. Annotation-based distance measures for patient subgroup discovery in clinical microarray studies. *Bioinformatics*, 23(17):2256–2264, 2007.

P. Pavlidis et al. Learning gene functional classification from multiple data. *J. Comput. Biol.*, pages 401–411, 2002.

A. Ruepp et al. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.

F. Tai and Pan. W. Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics*, 23(14):1775–1782, 2007.