Machine learning methods for gene/protein function prediction

Giorgio Valentini
valentini@dsi.unimi.it

DSI - Dipartimento di Scienze dell'Informazione

Università degli Studi di Milano

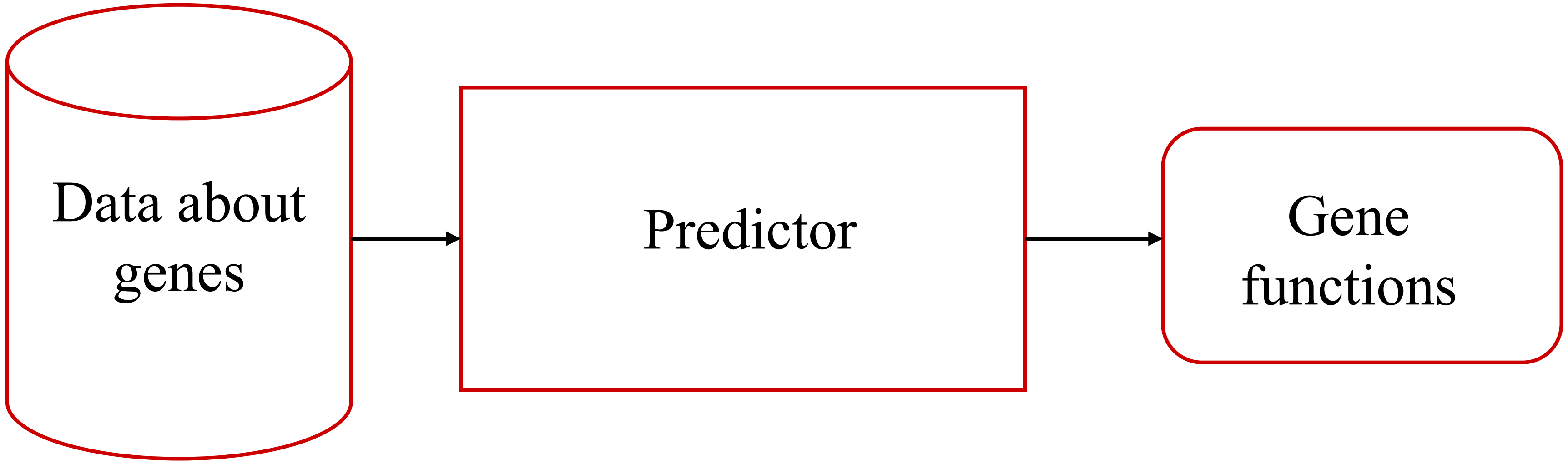
Outline

- Gene Function Prediction (GFP)
- The Gene Ontology
- Computational approaches to GFP
- Hierarchical Ensemble methods for GFP
- Two examples of Hierarchical ensembles:

  - A Bayesian approach (Barutcouglu et al, 2006)

  - True Path Rule ensembles (Valentini, 2011)

Gene function prediction



Gene function prediction can be formalized as a supervised machine learning problem

Motivation

- Novel high-throughput biotechnologies accumulated a wealth of data about genes and gene products
- Manual annotation of gene function is time consuming and expensive and becomes infeasible for growing amount of data.
- For most species the functions of several genes are unknown or only partially known: “in silico” methodsrepresent a fundamental tool for gene function prediction at genome-wide and ontology-wide level (*Friedberg*, 2006).
- Computational analysis provide predictions that can be considered hypotheses to drive the biological validation of gene function (*Pena-Castillo et al*. 2008).

Computational prediction supports biological gene function prediction

$$\begin{tikzcd}
\boxed{\text{Biological genome-wide gene function prediction through direct experimental assays is costly and time-consuming}} \arrow[r] & \boxed{\text{Computational prediction methods}}
\end{tikzcd}$$

Computational prediction methods assist the biologist to:

- •Suggest a restricted set of candidate functions that can be experimentally verified
- •Directly generate new hypotheses
- •Guide the exploration of promising hypotheses

Characteristics of the Gene Function Prediction (GFP) problem

- Large number of functional classes: hundreds (FunCat) or thousands (Gene Ontology (GO)) : large multi-class classification
- Multiple annotations for each gene: multilabel classification
- Different level of evidence for functional annotations: labels at different level of reliability
- Hierarchical relationships between functional classes (tree forest for FunCat, direct acyclic graph for GO): hierarchical relationships between classes (structured output)
- Class frequencies are unbalanced, with positive examples usually largely lower than negatives: unbalanced classification
- The notion of "negative example" is not univocally determined: different strategies to choose negative examples
- Multiple sources of data available: each type captures specific functional characteristics of genes/gene products: multi-source classification
- Data are usually complex (e.g. high-dimensional) and noisy: classification with complex and noisy data

Taxonomies of gene function

1. *Gene Ontology* (GO)

   http://www.geneontology.org/

   Fine grained: classes structured according to a directed acyclic graph

2. *Functional Catalogue (FunCat)*

   http://www.helmholtz-muenchen.de/en/mips/projects/funcat/

   Coarse grained: classes structured according to a tree

The Gene Ontology

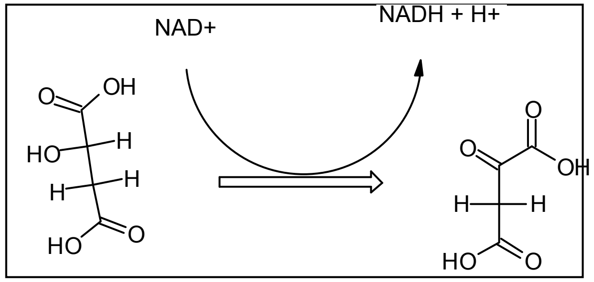
The Gene Ontology (GO) project began as a collaboration between three model organism databases, FlyBase (*Drosophila*), the *Saccharomyces* Genome Database (SGD) and the Mouse Genome Database (MGD), in 1998. Now it includes several of the world's major repositories for plant, animal and microbial genomes.

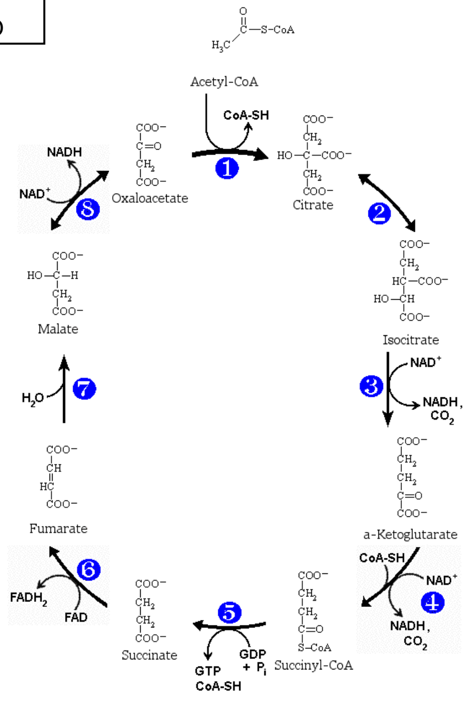
The GO project has developed three structured controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner

The Gene Ontology (GO) is actually three Ontologies

1) Molecular Function

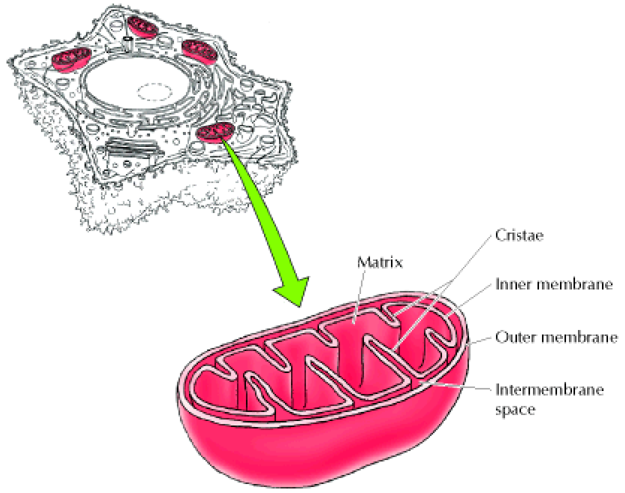
GO term: Malate dehydrogenase activity
GO id: GO:0030060
(S)-malate + <u>NAD(+)</u> = <u>oxaloacetate</u> + <u>NADH</u>.





2) Biological Process

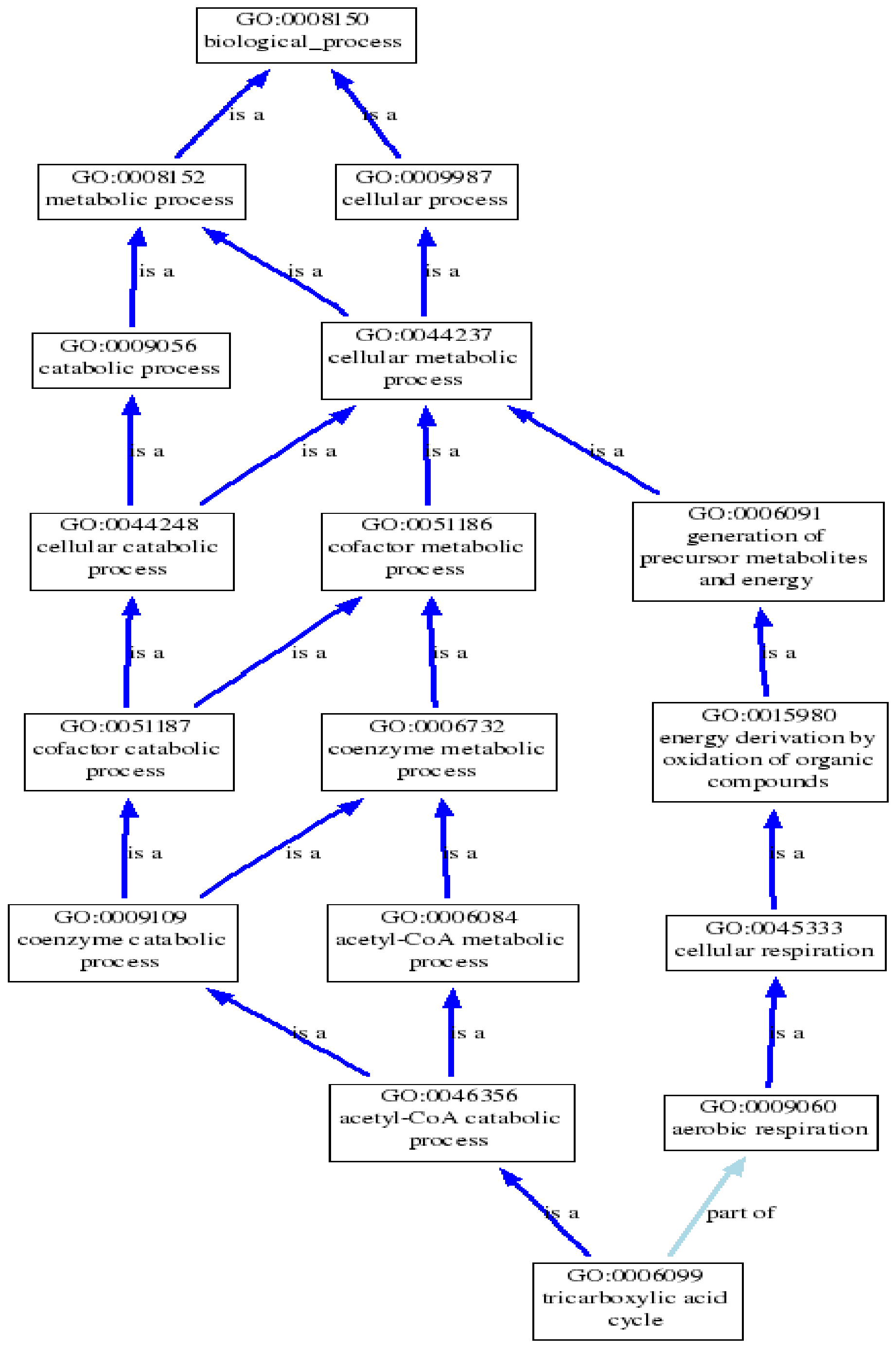
GO term: tricarboxylic acid cycle
Synonym: Krebs cycle
Synonym: citric acid cycle
GO id: GO:0006099



3) Cellular Component

GO term: mitochondrion
GO id: GO:0005739
Definition: A semiautonomous, self replicating organelle that occurs in varying numbers, shapes, and sizes in the cytoplasm of virtually all eukaryotic cells. It is notably the site of tissue respiration.

(Slide downloaded from www.geneontology.org)


GO:0008150 biological_process
is a
is a
GO:0008152 metabolic process
GO:0009987 cellular process
is a
is a
is a
GO:0009056 catabolic process
GO:0044237 cellular metabolic process
is a
is a
is a
is a
GO:0044248 cellular catabolic process
GO:0051186 cofactor metabolic process
GO:0006091 generation of precursor metabolites and energy
is a
is a
is a
is a
GO:0051187 cofactor catabolic process
GO:0006732 coenzyme metabolic process
GO:0015980 energy derivation by oxidation of organic compounds
is a
is a
is a
is a
GO:0009109 coenzyme catabolic process
GO:0006084 acetyl-CoA metabolic process
GO:0045333 cellular respiration
is a
is a
is a
GO:0046356 acetyl-CoA catabolic process
GO:0009060 aerobic respiration
is a
part of
GO:0006099 tricarboxylic acid cycle

Relationships between GO terms are structured according to a DAG

Relationships between terms in the GO

The ontologies of GO are structured as a directed acyclic graph *(DAG)* $G=<V,E>$

$V = \{t \mid \text{terms of the GO}\}$ $E= \{(t, u) \mid t \, \varepsilon \, V \text{ and } t \, \varepsilon \, V\}$

Relations between GO terms are also categorized and defined:

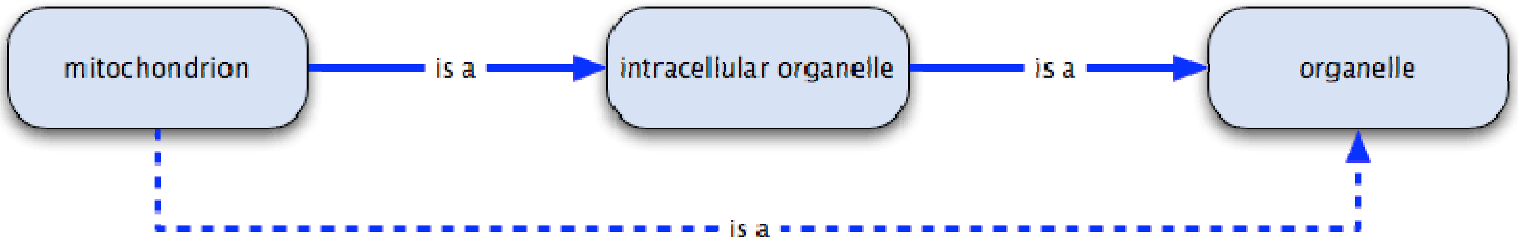
- *is a* (subtype relations)
- *part of* (part-whole relations)
- *regulates* (control relations)

Is a relation

If we say A is a B, we mean that node A is a subtype of node B.

For example, mitotic cell cycle is a cell cycle, or lyase activity is a catalytic activity.

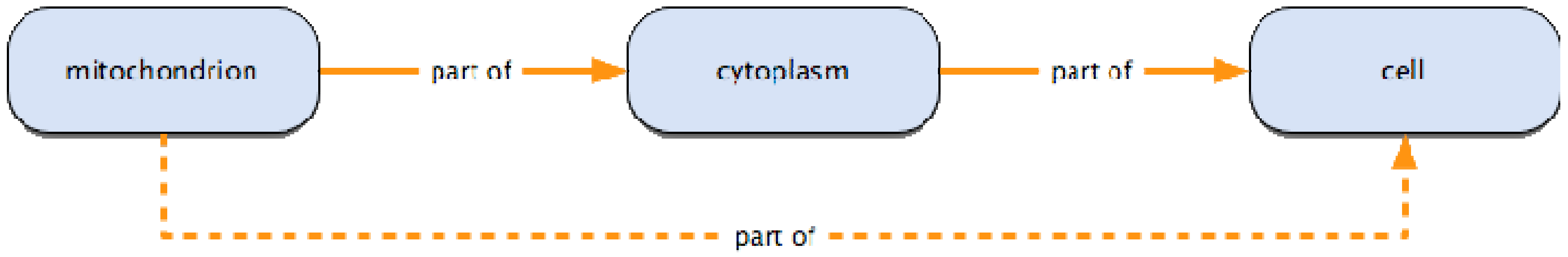
The is a relation is transitive, which means that if A is a B, and B is a C, we can infer that A is a C.
E.g.:



Part of relation

The relation part of represents part-whole relationships in the GO.

The part of relation is transitive:

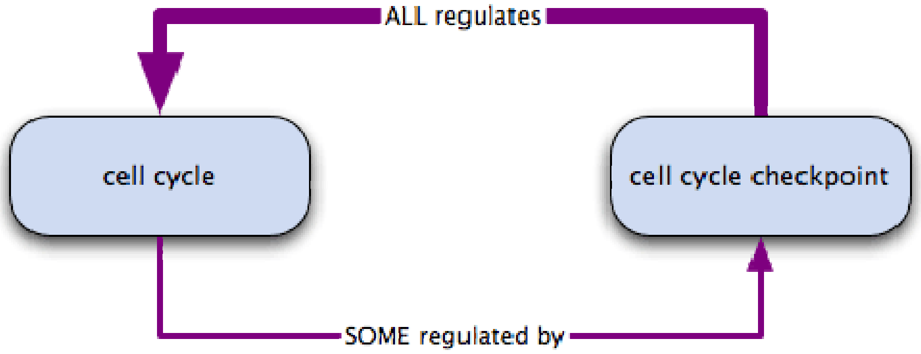


Regulates relation

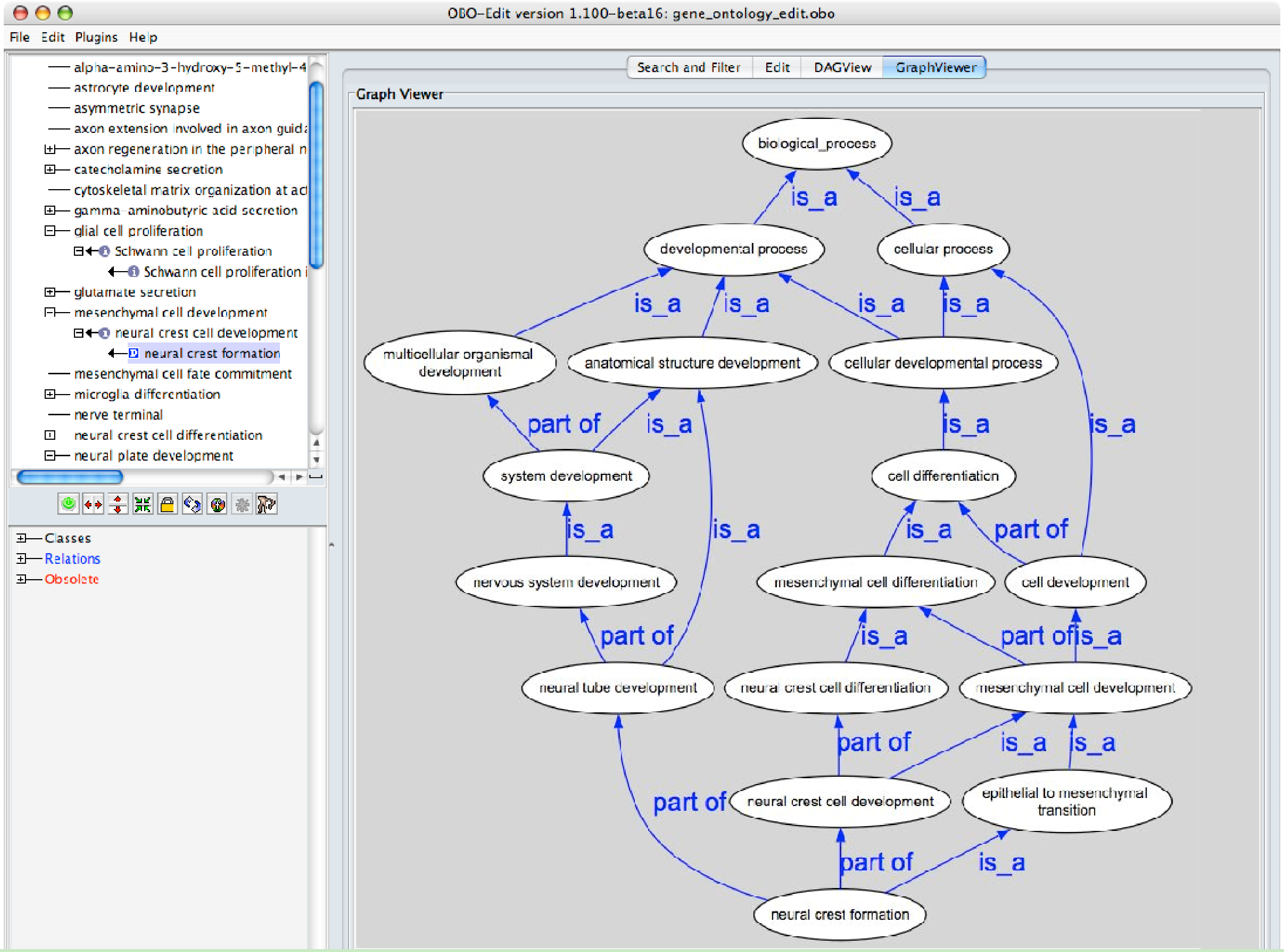
If we say that A regulates B we mean that A directly affects the manifestation of B, i.e. the former regulates the latter.

For example, the target of the regulation may be another process—for example, regulation of a pathway or an enzymatic reaction—or it may be a quality, such as cell size or pH.

Analogously to part of, this relation is used specifically to mean necessarily regulates:

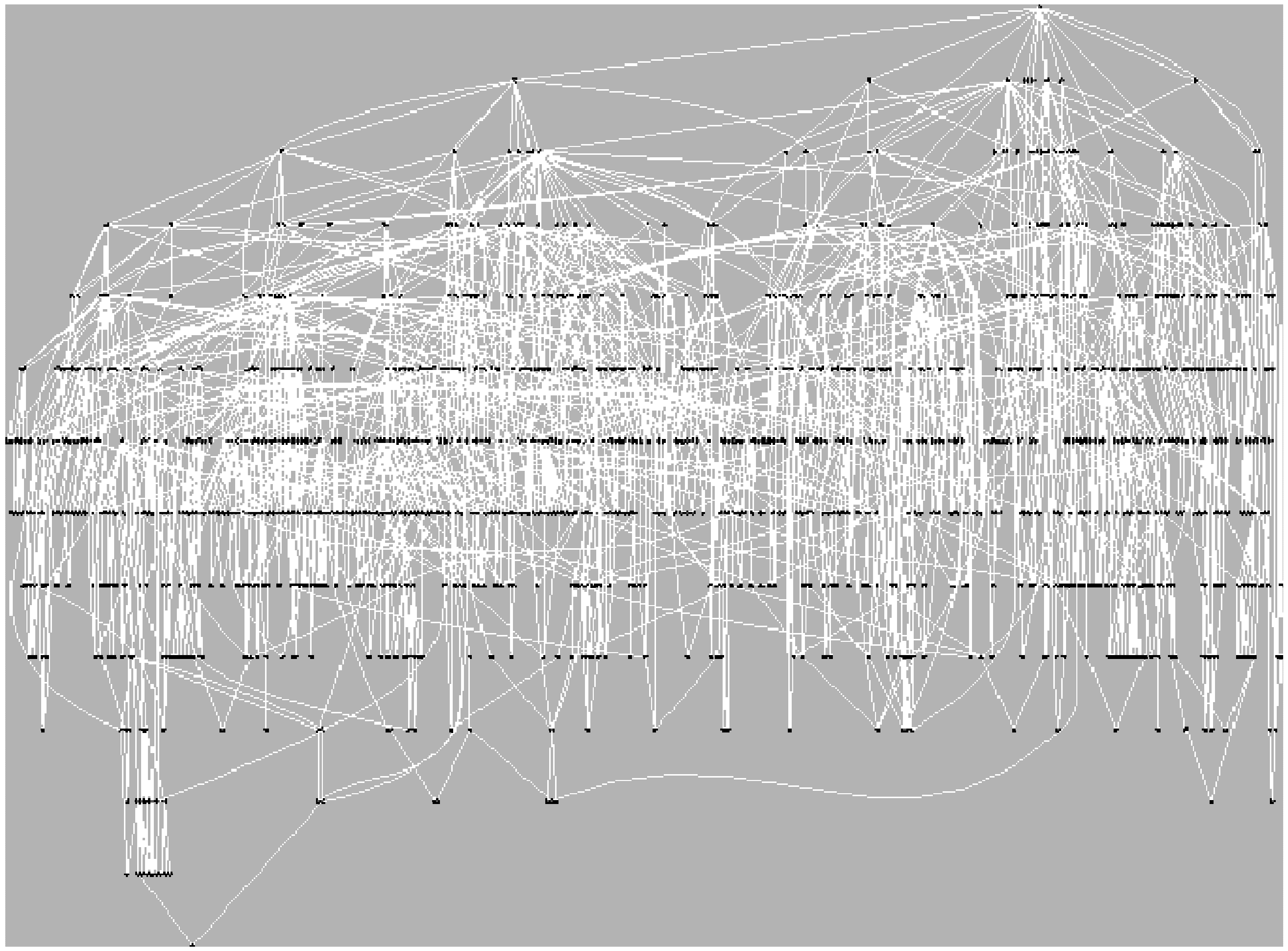


In general regulates is not transitive


OBO-Edit version 1.100-beta16: gene_ontology_edit.obo
File Edit Plugins Help
alpha-amino-3-hydroxy-5-methyl-4
astrocyte development
asymmetric synapse
axon extension involved in axon guid
axon regeneration in the peripheral n
catecholamine secretion
cytoskeletal matrix organization at ac
gamma-aminobutyric acid secretion
glial cell proliferation
Schwann cell proliferation
Schwann cell proliferation i
glutamate secretion
mesenchymal cell development
neural crest cell development
neural crest formation
mesenchymal cell fate commitment
microglia differentiation
nerve terminal
neural crest cell differentiation
neural plate development
Classes
Relations
Obsolete
Search and Filter
Edit
DAGView
GraphViewer
Graph Viewer
biological_process
developmental process
cellular process
multicellular organismal development
anatomical structure development
cellular developmental process
system development
cell differentiation
nervous system development
mesenchymal cell differentiation
cell development
neural tube development
neural crest cell differentiation
mesenchymal cell development
neural crest cell development
epithelial to mesenchymal transition
neural crest formation
is_a
part of

A visualization of the GO DAG trough OBO-Edit

GO DAG of the BP ontology *(S. cerevisiae)*

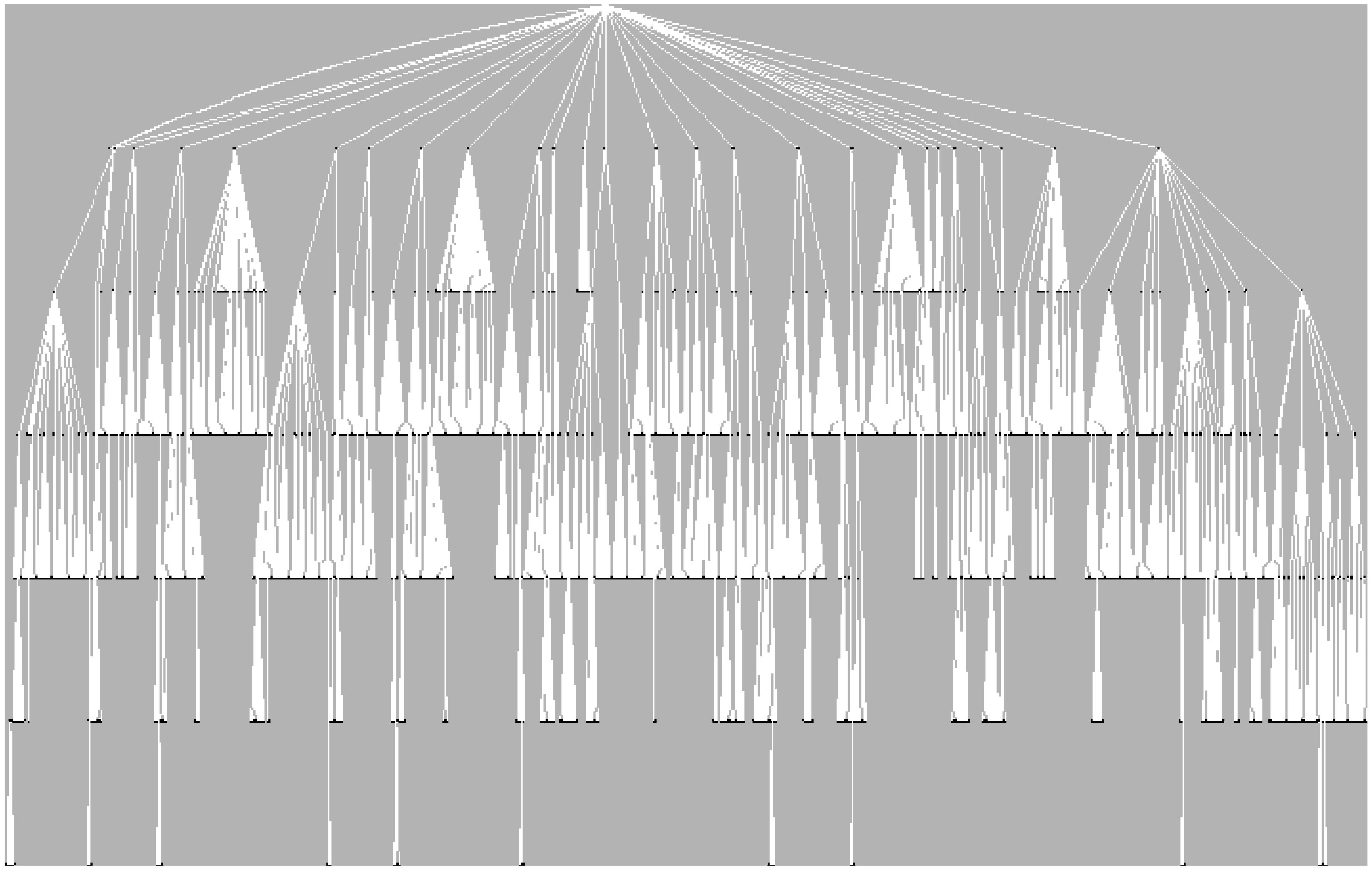


1074 GO classes (nodes) connected by 1804 edges

Graph realized through *HCGene* (Valentini, Cesa-Bianchi, *Bioinformatics* 24(5), 2008)

The Functional Catalogue (FunCat)

http://www.helmholtz-muenchen.de/en/mips/projects/funcat



The Functional Catalogue (FunCat)

http://www.helmholtz-muenchen.de/en/mips/projects/funcat

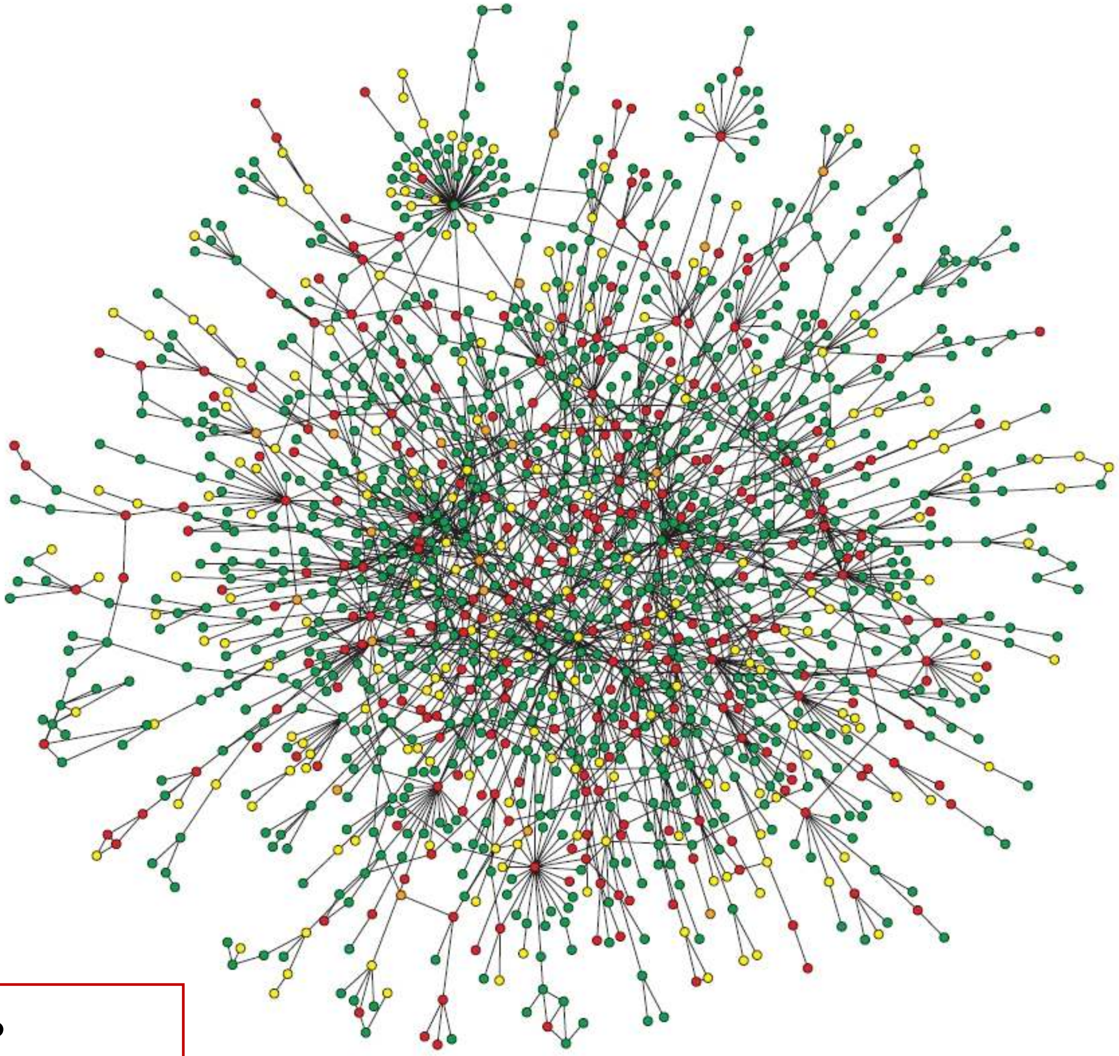
- The *Functional Catalogue* is an annotation scheme for the functional description of proteins of prokaryotic and eukaryotic origin
- Hierarchical tree like structure.
- Up to six levels of increasing specificity. FunCat version 2.1 includes 1362 functional categories.
- FunCat descriptive, but compact: classifies protein functions not down to the most specific level.
- Comparable to parts of the 'Molecular Function' and 'Biological Process' terms of the GO system.
- More compact and stable than GO, focuses on the functional process not describing the molecular function on the atomic level

Computational approaches to GFP

A very schematic taxonomy of computational GFP methods:

- Inference and *annotation transfer through sequence similarity* (BLAST)
- *Network-based* methods
- *Kernel methods* for structured output spaces
- *Hierarchical ensemble methods*

Biological networks

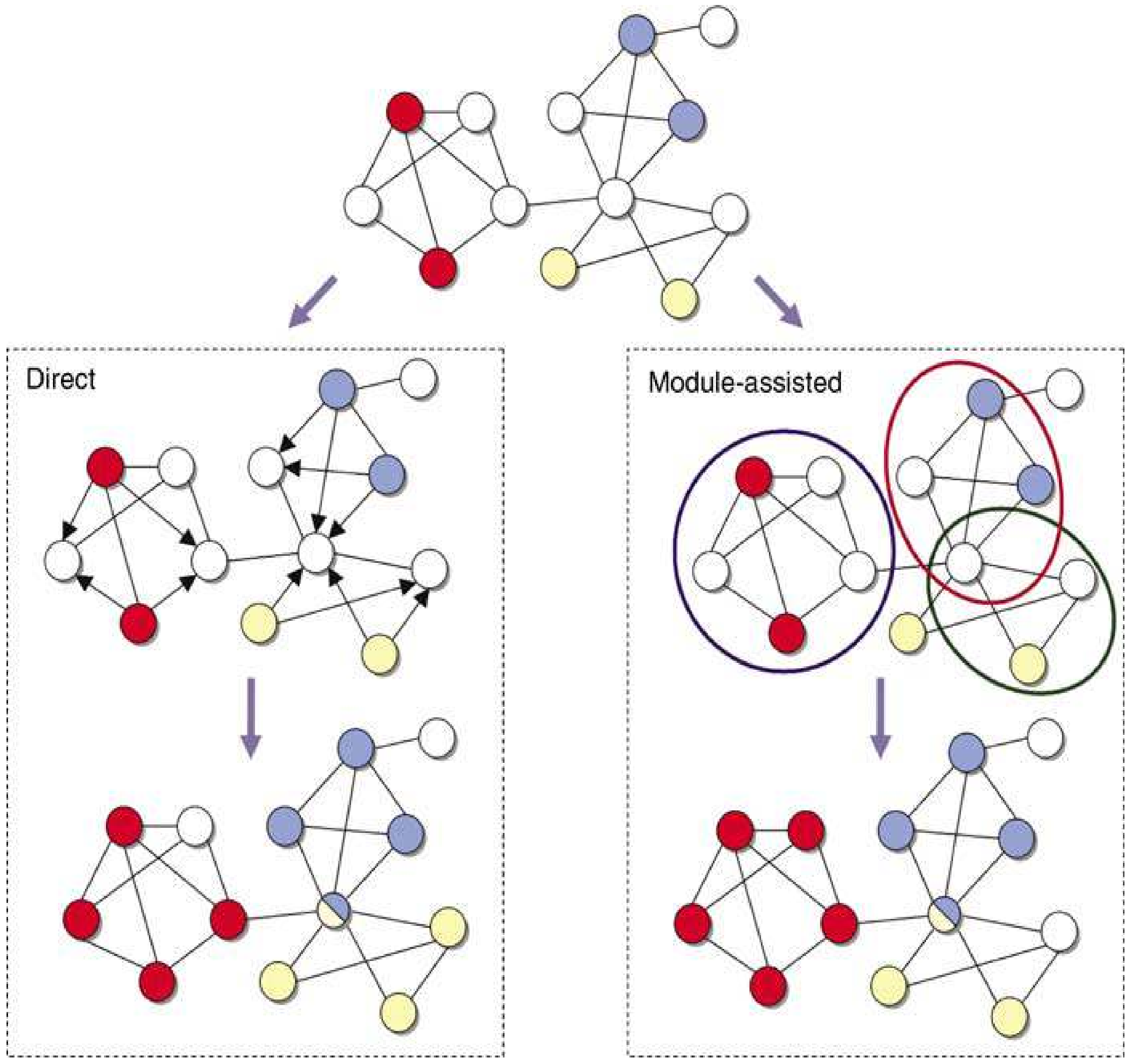


S. Cerevisiae

4389 proteins

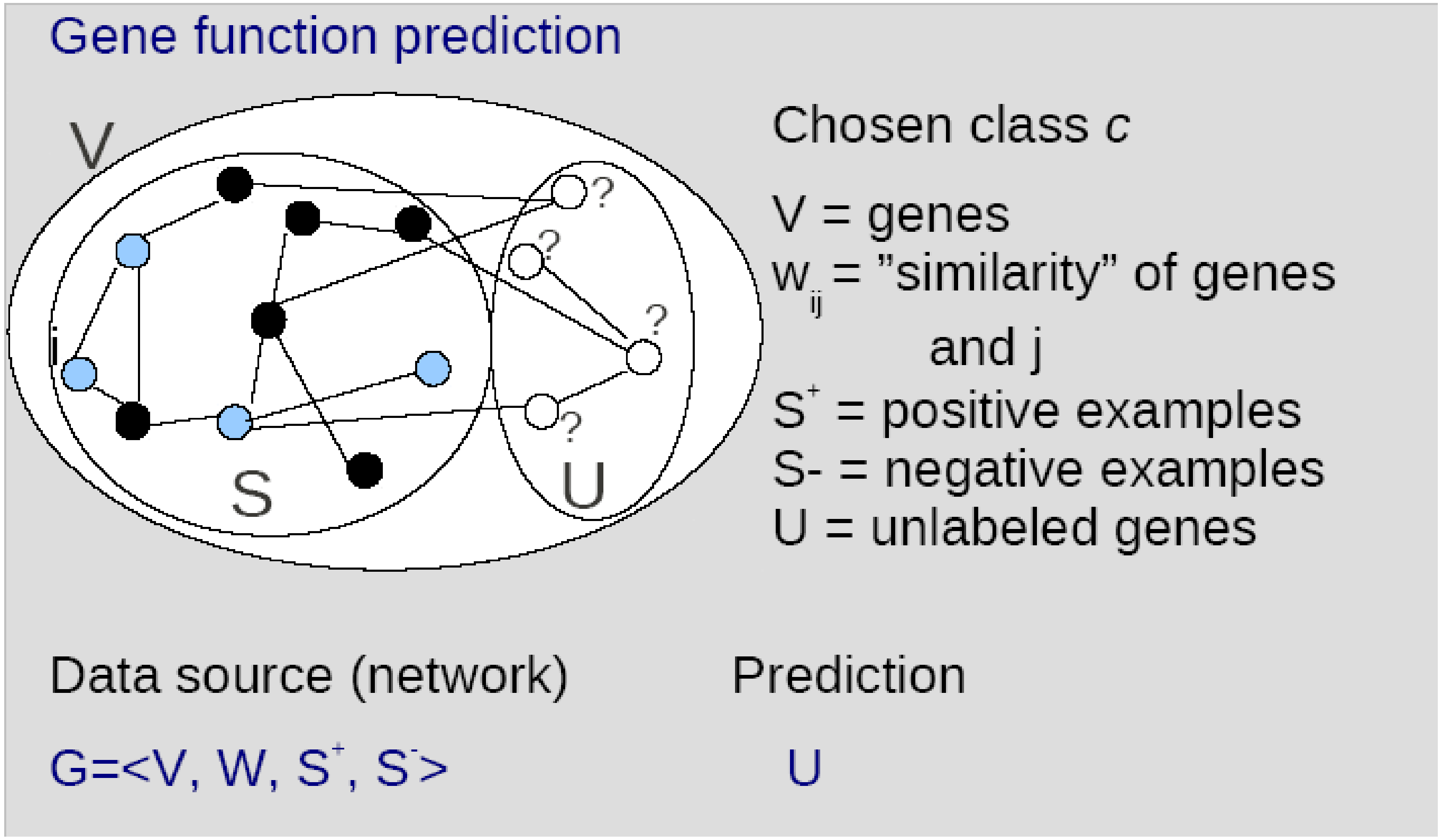
14319 interactions

A network-based approach

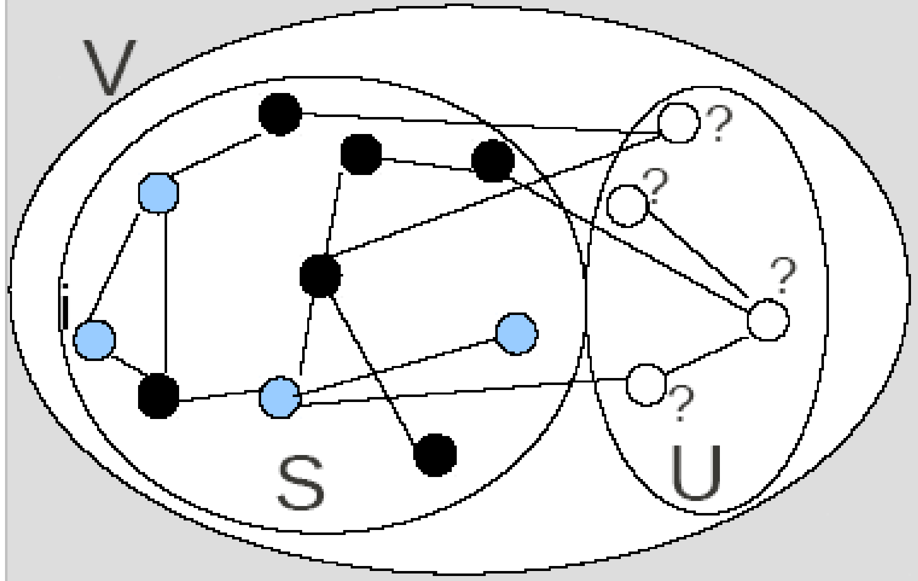


From: Sharan et al. Mol. Sys. Biol. 2007

Network based methods: predicting a specific functional term



Gene function prediction



Chosen class *c*

V = genes

w_{ij} = "similarity" of genes i and j

S^+ = positive examples

S- = negative examples

U = unlabeled genes

Data source (network)

$G=<V, W, S^+, S^->$

Prediction

U

Network-based methods

Several available methods:

- *Guilt by association* (Marcotte *et al.* 1999, Oliver et al. 2000)
- *Label propagation* (Zhu and Ghahramani, 2003, Zhou et al. 2004)
- *Markov random walks* (Szummer and Jaakkola, 2002, Azran et al 2007)
- *Markov random fields* (Deng et al. 2004)
- *Graph regularization techniques* (Belkin et al. 2004, Dellaleu et al 2005)
- *Gaussian random fields* (Tsuda et al. 2005, Mostafavi et al. 2010)
- *Hopfield networks* (Karaoz et al. 2004, Bertoni et al. 2011)

These different approaches *minimize a similar quadratic criterion* to improve:

a) Consistency of the initial labeling

b) Topological consistency of the data

They exploit different types of relational data: physical and genetic interactions, similarities between protein domains or motifs, structural and sequence homologies, correlations between expression profiels, …

-→ need for **network integration algorithms**

Kernel methods

Kernel methods are largely applied to classification problems:

1. Obtaining a non-linear classifier, through a non-linear mapping into the feature space, using an algorithm designed for linear discrimination :

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x})$$

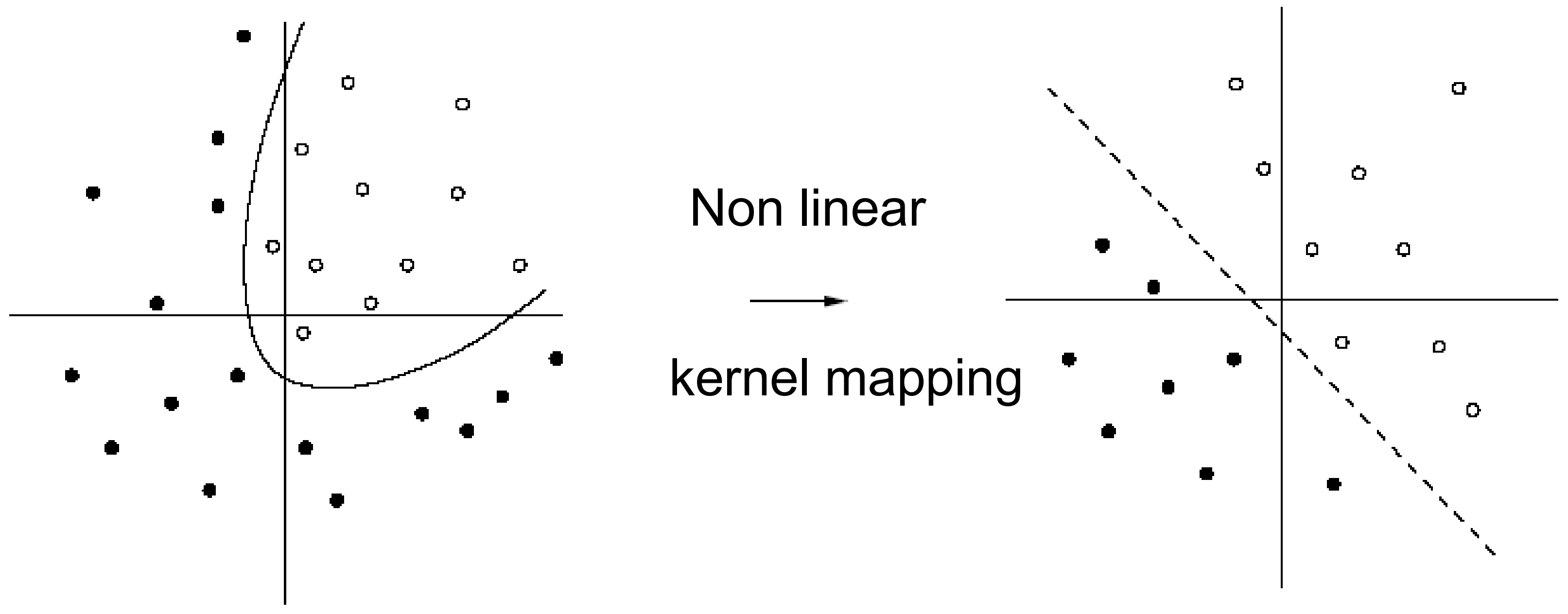
2. Whenever **w** can be expressed as a weighted sum over the images of the input examples:

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i) \Rightarrow f(\mathbf{x}) = \sum_i \alpha_i \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x})$$

3. The discriminant function can be expressed through a suitable kernel function:

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

Kernel metods for binary classification problems



φ

Original input space $\xrightarrow{\hspace{2cm}}$ Transformed feature space

Kernel methods for structured output spaces

A binary classier can predict whether a protein performs a certain function:

$$f : X \rightarrow Y_i \quad Y_i = \{0,1\} \quad 1 \leq i \leq k$$

How to predict the full hierarchical annotation $\mathbf{y} = \{y_1, y_2, \ldots, y_k\}$?

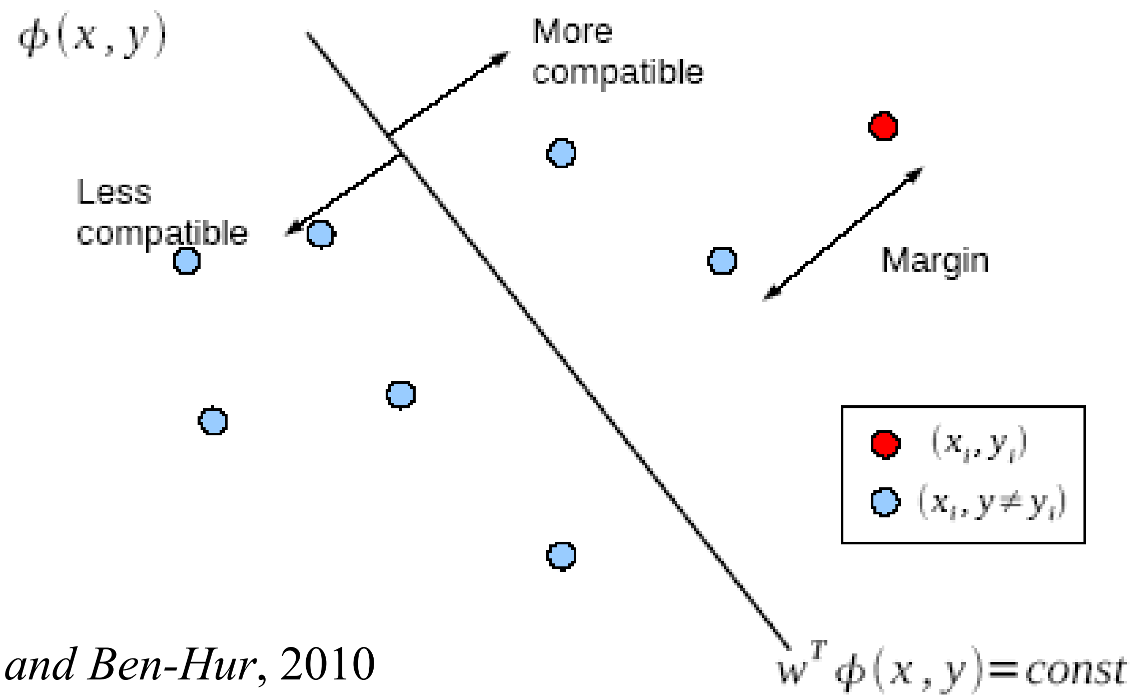
The main idea: using a kernel for structured output, that is a function:

$$f : X \times Y \rightarrow \Re$$

This classification rule chooses the label **y** that is most compatible with an input *x*.

Whereas in two-class classification problems the kernel depends *only on the input* (proteins), in the structured-output setting it is a *joint function of inputs and outputs* (set of the labels)

Kernel methods for structured output spaces: a geometric view



From: *Sokolov and Ben-Hur*, 2010

The classifier is assumed to be linear in the joint input-output feature space:

$$f(\mathbf{x},\mathbf{y} \mid \mathbf{w}) = \mathbf{w}^T \varphi(\mathbf{x},\mathbf{y})$$

Structured output kernel methods for gene function prediction

- *Sokolov and Ben-Hur* (2010): a structured Perceptron,

and a variant of the structured support vector machine (*Tsochantaridis et al.* 2005), applied to the the prediction of GO terms in mouse and other model organisms

- *Astikainen et al. (*2008) and *Rousu et al. (*2006): Structured output maximum-margin algorithms applied to the tree-structured prediction of enzyme functions

Hierarchical ensemble methods

They are in general characterized by a two-step strategy:

1. Flat learning of the protein function on a per-term basis (a set of independent classification problems)
2. Combination of the predictions by exploiting the relationships between terms that govern the hierarchy of the functional classes.

The term *ensemble* raises from the fact that a set of learning machines in someway combine their output.

In principle any supervised learning algorithm can be used for step 1.

Step 2 requires a proper combination of the predictions made at step 1.

Hierarchical ensemble methods

- Bayesian network-based ensembles (*Barutcuoglu et al*. 2006, *Guan et al*. 2008)
- Hierarchical renconciliation methods (*Obozinski et al*. 2008)
- Hierarchical decision trees (*Vens et al*. 2008, *Schietgat et al* 2010)
- Hierarchical Bayesian cost-sensitive ensembles (*Cesa-Bianchi and Valentini*, 2010)
- True Path Rule Ensembles (*Valentini*, 2011)

Hierarchical Bayesian network-based prediction of gene function

(Barutcuoglu, Schapire and Troyanskaya, 2006)

Main ideas:

- *Flat prediction* of each term/class (possibly inconsistent)
- *Bayesian hierarchical combination* scheme to allow collaborative error-correction over all nodes

Basic notation:

y_i : binary membership to class *i*

$\hat{y}_i$: classifier output for class *i*, $1 \leq i \leq N$

Bayesian correction of classifier outputs

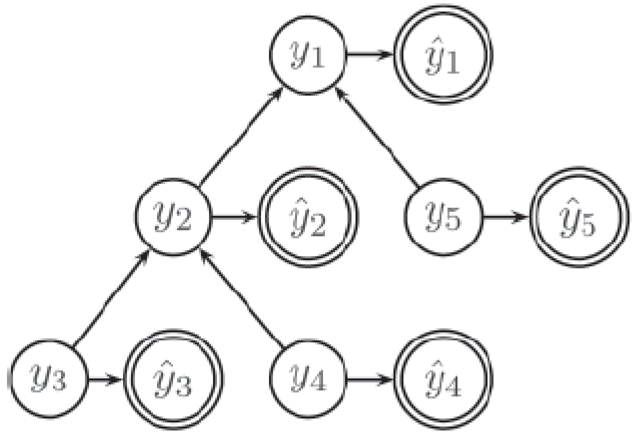
Goal: given a set of (possibly inconsistent) $\hat{y}_i$
find the set of consistent y_i that maximize:

$$P(y_1, \ldots, y_N \mid \hat{y}_1, \ldots, \hat{y}_N) = \frac{P(\hat{y}_1, \ldots, \hat{y}_N \mid y_1, \ldots, y_N) P(y_1, \ldots, y_N)}{Z}$$

Direct solution is too hard … (exponential in time w.r.t to the number of nodes)

Proposed solution: *a Bayesian network structure that exploits the relationships between functional classes*.

The proposed Bayesian network



1. y_i nodes conditioned to their children (structure constraints)
2. $\hat{y}_i$ nodes conditioned on their label y_i (Bayes rule)
3. $\hat{y}_i$ are independent from both $\hat{y}_j, j \neq i$ and $y_j, j \neq i$ given y_i

This allows us to simplify the Bayesian equation:

from 1: $$P(y_1, \ldots, y_N) = \prod_{i=1}^{N} P(y_i \mid ch(y_i))$$

from 2,3: $$P(\hat{y}_1, \ldots, \hat{y}_N, \mid y_1, \ldots, y_N) = \prod_{i=1}^{N} P(\hat{y}_i \mid y_i)$$

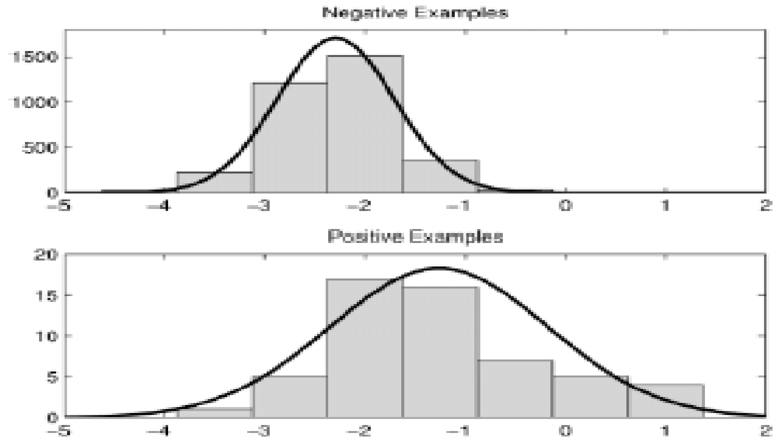
Estimation of the probabilities

Estimation of $P(y_1,\ldots,y_N) = \prod_{i=1}^{N} P(y_i \mid ch(y_i))$

Can be inferred from training labels by counting

Estimation of $P(\hat{y}_1,\ldots,\hat{y}_N, \mid y_1,\ldots,y_N) = \prod_{i=1}^{N} P(\hat{y}_i \mid y_i)$

Can be inferred by validation during training, by modeling the distribution of $\hat{y}_i$ outputs over positive and negative examples.
E.g.: a parametric gaussian model:



Implementation of the method

- *Bagged ensemble of SVMs* (10 SVMs) trained at each node (see next slide …)
- Median values of their outputs on out-of-bag examples have been used to *estimate means and variances for each class*.
- Mean and variances have been used as parameters of the *gaussian models used to estimate the conditional probabilities* $P(\hat{y}_i \mid y_i = 1)$ and $P(\hat{y}_i \mid y_i = 0)$

The prediction of the label for each class i is then computed as follows:

$$y_i^* = \arg\max_{y_i \in \{0,1\}} P(y_i \mid \hat{y}_1, \ldots, \hat{y}_N) = \frac{\prod_{j=1}^{N} P(\hat{y}_j \mid y_i) P(y_i \mid child(y_i))}{Z}$$

Bagging (**B**ootstrap **agg**regat**ing**)

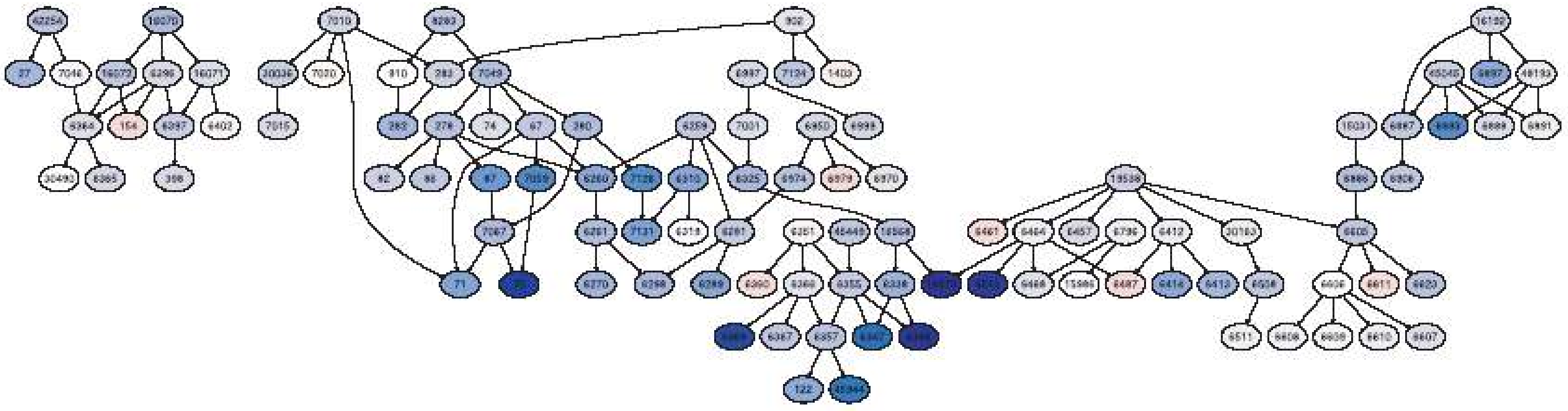
(Breiman, 1996)

```
Input:   $Z = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$   $y_i \in Y = \{1, \ldots, k\}$   $LearnAlg$
Do for t=1 to T:
    1. Bootstrap replicate $Z_t$ from Z
       (random sampling with replacement)
    2. Get back an hypothesis $h_t$:X ->Y
       $h_t$ = LearnAlg($Z_t$ )
end for
Output the final hypothesis by aggregation and majority voting:
```

$$h_{fin}(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} \begin{cases} 1 & if \quad h_t(x) = y \\ 0 & otherwise \end{cases}$$

- Effective with unstable algorithms
- It reduces the variance component of the error

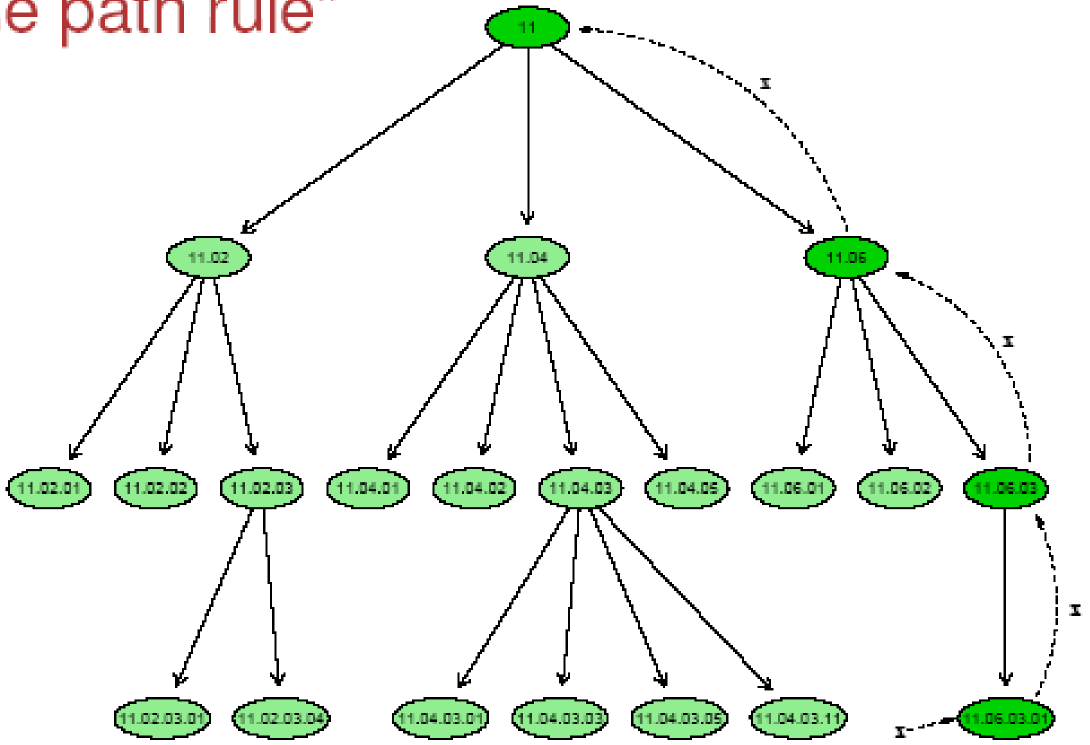
Results on a sub-hierarchy of the BP GO ontology



- 105 terms/nodes of the GO BP (model organism *S.cerevisiae*)
- 4 types of data integrated through Vector Space Integration
- Hierarchical approach improves AUC results on 93 of the 105 GO terms
- Darker blue: improvements; darker red: deterioration; white: no change.

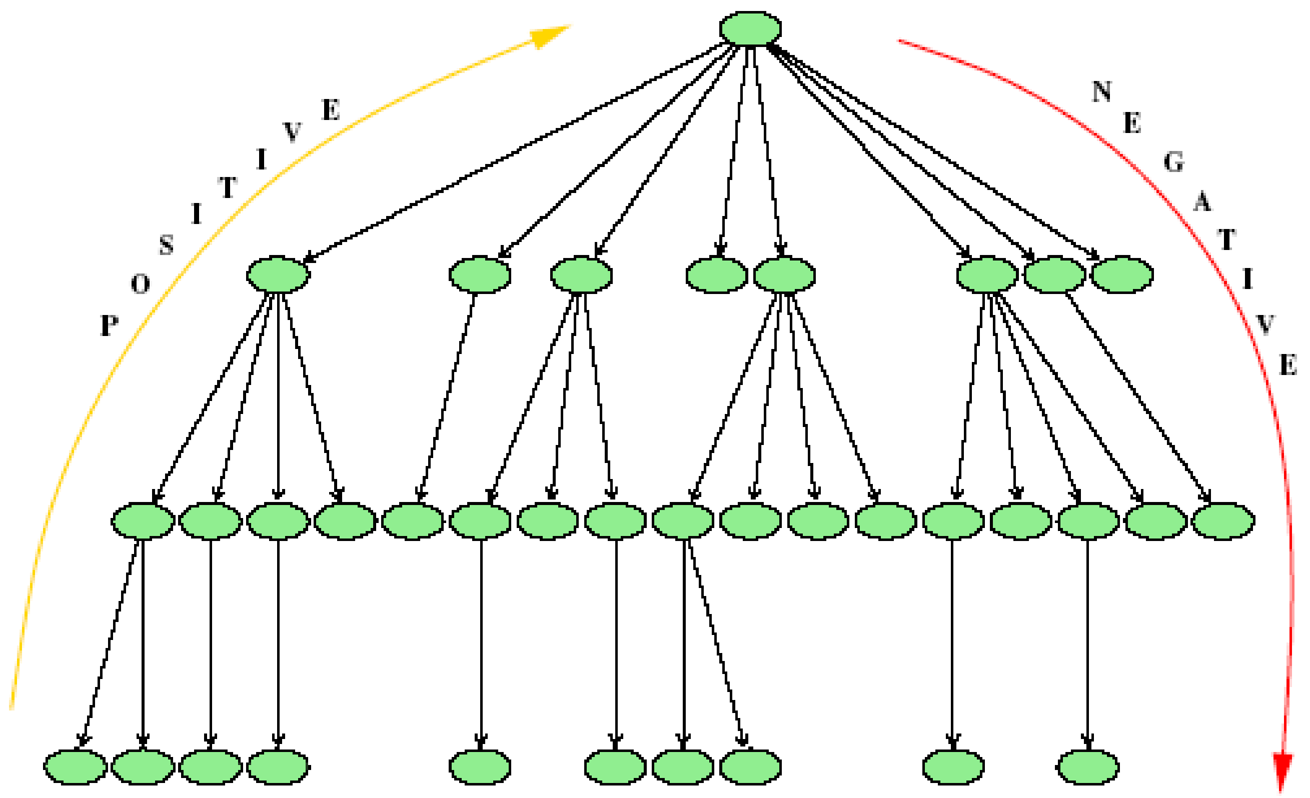
An approach based on the "true path rule"

The "true path rule"



"An annotation for a class in the hierarchy is automatically transferred to its ancestors, while genes unannotated for a class cannot be annotated for its descendants".

True Path Rule ensembles (*Valentini*, 2011): an asymmetric flow of information



From bottom to top : positive predictions influence ancestor nodes/classifiers

From top to bottom : negative predictions influence descendant nodes/classifiers

TPR ensemble: basic notation

- A gene/gene product x can be assigned to one or more functional classes:
  $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$
- Assignments can be coded through a vector of multilabels $\mathbf{y} =< y_1, y_2, \ldots, y_m >\in \{0,1\}^m$.
  If x belongs to class c_i, then $y_i = 1$, otherwise $y_i = 0$.
- Nodes corresponding to the class c_i can be simply denoted by i.
- $\mathrm{child}(i)$ represents the set of children nodes of i; $\mathrm{par}(i)$ the set of its parents.
- The TPR ensemble classifier $D : X \to \{0,1\}^m$ computes the multilabel associated to each gene $x \in X$
- $d_i(x) \in \{0,1\}$ is the label predicted by the TPR classifier at node i. If there is no ambiguity we represent $d_i(x)$ simply by d_i.

The rules a TPR ensemble must obey

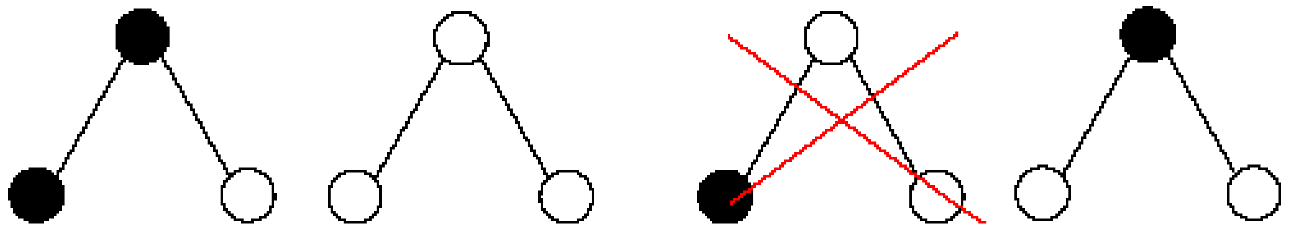
Considering the parents of a given node i:

$$\begin{cases} d_i = 1 & \Rightarrow & d_{par(i)} = 1 \\ d_i = 0 & \not\Rightarrow & d_{par(i)} = 0 \end{cases}$$

Considering the children of a given node i:

$$\begin{cases} d_i = 1 & \not\Rightarrow & d_{child(i)} = 1 \\ d_i = 0 & \Rightarrow & d_{child(i)} = 0 \end{cases}$$

Example: (black $d_i = 1$, white $d_i = 0$)



The basic TPR ensemble (1)

- Base classifiers estimate local probabilities $\hat{p}_i(x)$ that a given example x belongs to class c_i
- The ensemble provides an estimate of the "consensus" global probability $p_i(x)$

The "consensus" global probability $p_i(x)$ is estimated in two steps:

Bottom-up step $p_i(x)$ is computed by averaging the local probabilities of the "positive" predictions of computed at node i and $\mathrm{child}(i)$

Top-down step If $p_i(x) < 1/2$ then the subtree "is set to 0":
$\forall j \in \mathrm{desc}(i), d_j(x) = 0$

The basic TPR ensemble (2)

Given the set $\phi_i(x)$ of all the children nodes of node i for which we have a positive prediction for the gene x:

$$\phi_i(x) = \{j | j \in \mathrm{child}(i), d_j(x) = 1\}$$

The *consensus probability* $p_i(x)$ that a gene x belongs to the node/class i is:

$$p_i(x) = \frac{1}{1 + |\phi_i(x)|}\left(\hat{p}_i(x) + \sum_{j \in \phi(x)} p_j(x)\right)$$

- The $p_i(x)$ are computed in a bottom-up fashion, visiting "per-level" the tree from bottom to top, starting from the leaves, and continuing up to the root.
- At each level and for each node we have an asymmetric flow of information: bottom-up “positive” information, and top-down “negative” information.

The weighted hierarchical True Path Rule (TPR-w) ensemble

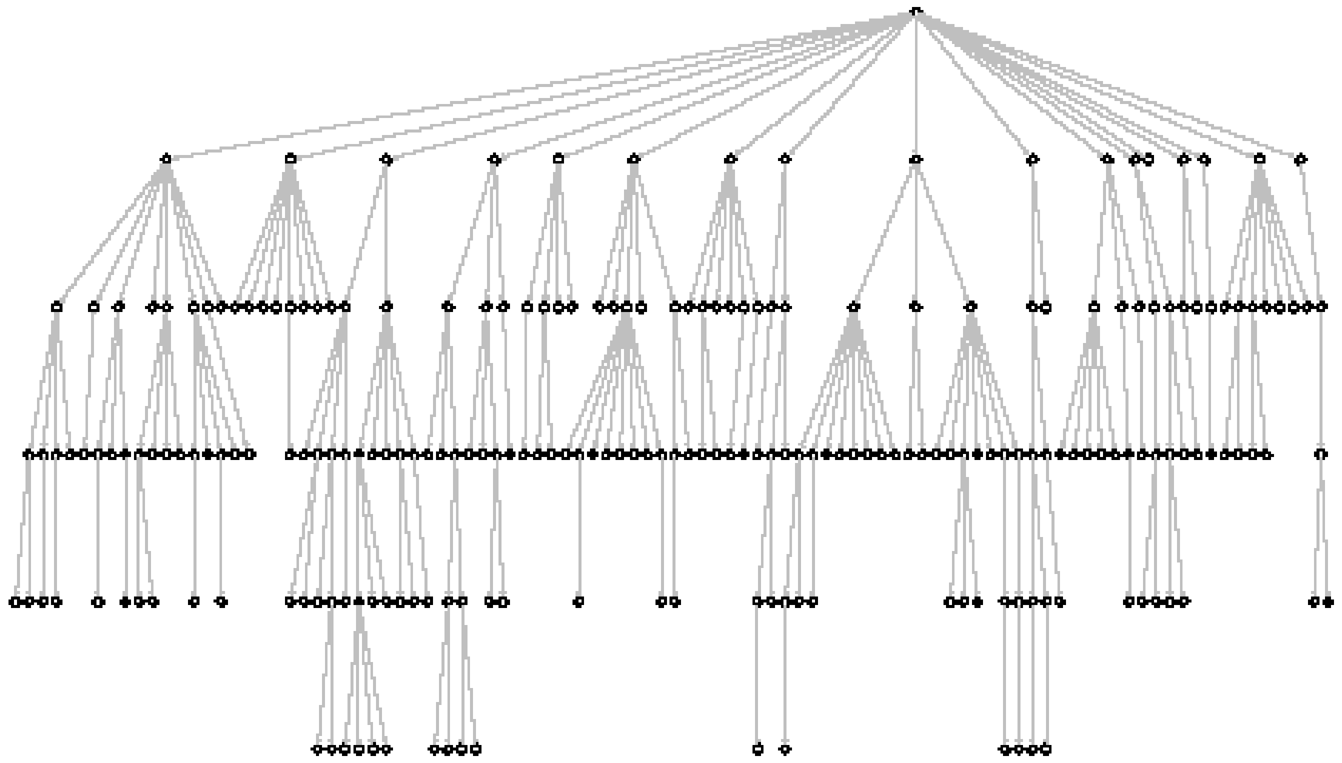
TPR-w is a variant of the basic *TPR*: we can modulate the role of the local predictor w.r.t. the predictions of its children and descendants.

We introduce a *parent weight* w_p, $0 \leq w_p \leq 1$, such that the prediction is shared proportionally to w_p and $1 - w_p$ between respectively the local parent predictor and the set of its children:

$$p_i(x) = w_p \cdot \hat{p}_i(x) + \frac{1 - w_p}{|\phi(x)|} \sum_{j \in \phi(x)} p_j(x)$$

This learning behaviour is recursively reproduced from the leaves up to the root of the overall taxonomy.

An application of TPR ensembles to a genome and ontology-wide GFP problem



- Genome-wide gene function prediction in *S. cerevisiae*
- About 200 FunCat classes (5 hierarchical levels)
- About 6000 genes to be classified (1000 unknown)

Average F-score results

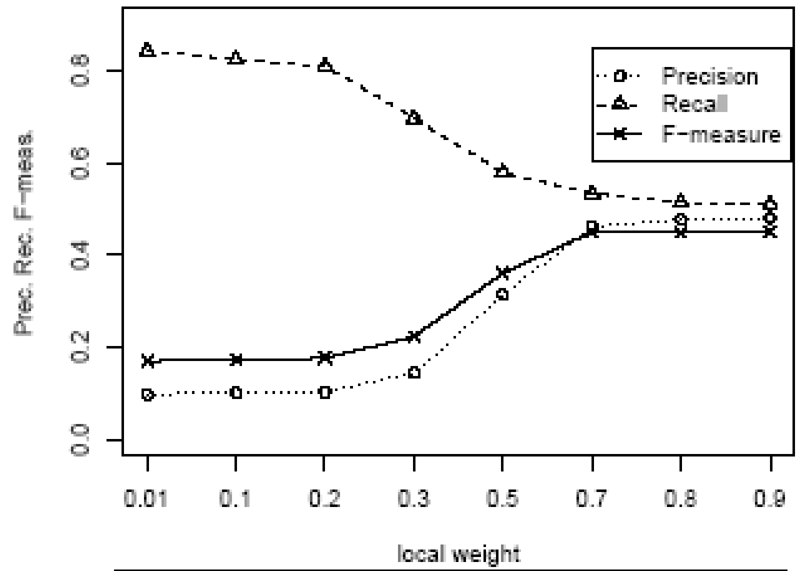
Average per-class F-score results

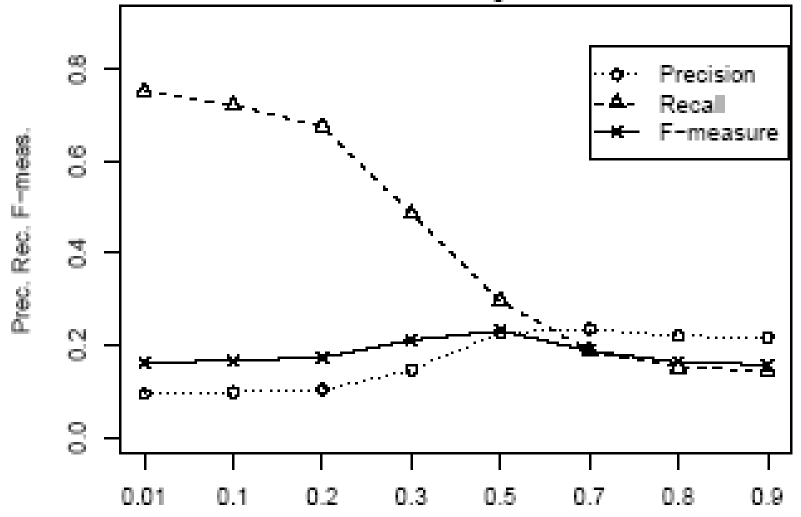
Flat	HTD	TPR	TPR-w
0.1489	0.2222	0.1824	0.2414

Hierarchical F-score results

Data set	HTD	TPR	TPR-w
Pfam-1	0.4123	0.3080	0.4131
Pfam-2	0.3406	0.2684	**0.3700**
Phylo	0.0497	0.2010	**0.2174**
Expr	0.1166	0.1696	**0.1784**
PPI-BG	0.3226	0.2670	**0.3485**
PPI-VM	0.3977	0.2796	0.4000
SP-sim	0.4251	0.2398	**0.4472**
Average	0.2949	0.2468	0.3392

Tuning precision and recall in TPR-w ensembles





- A single global parameter can regulate precision/recall characteristics (useful for different types of gene function prediction tasks)
- Opposite trends of precision and recall w.r.t the parent weight parameter w_p
- Usually better F-scores with $w_p > 0.5$

Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference

Effectiveness of hierarchical ensemble methods *depend on the synergy between different learning strategies* (*Cesa-Bianchi, Re, Valentini*, 2010):

(a) hierarchical strategies to take into account the relationships between classes;
(b) data integration approaches to capture different functional characteristics of genes;
(c) cost-sensitive methods to address the unbalance between positive and negative examples for each functional class.
(d) methods to choose negative examples and to take into account the reliability of the annotations.

Open problems

- Can we design scalable methods for massive biomolecular data integration in the context of GFP?
- Can we introduce active learning techniques in this context?
- Which is the "best" method for GFP? (it is likely that this is a "ill posed" problem …). Experimental work: comparison between different methods in the context of genome-wide gene function prediction (e.g. in the spirit of the *MouseFunc* project).
- Can we develop GFP based on comparative genomics techniques to exploit "cross-species" knowledge?
- Can we actually develop protocols for a joint "in silico" and "wet" GFP?

References (1)

Astikainen, K., Holm, L., Pitkanen, E., Szedmak, S., and Rousu, J. (2008). Towards structured output prediction of enzyme function. *BMC Proceedings*, 2(Suppl 4:S2).

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

Belkin, M, Matveeva, I, Niyogi, P. (2004) Regularization and semi-supervised learning on large graphs. In COLT

Bengio, Y., Delalleau, O., and Le Roux, N. (2006). Label Propagation and Quadratic Criterion. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press.

Bertoni, A., Frasca, M., Valentini G. (2011) COSNet: a Cost Sensitive Neural Network for Semi-supervised Learning in Graphs., *European Conference on Machine Learning 2011, Athens, Lecture Notes in Computer Science, Springer*

Cesa-Bianchi, N. and Valentini, G. (2010). Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, 8, 14–29.

Cesa-Bianchi, N., Re, M., and Valentini, G. (2010). Functional inference in FunCat through the combination of hierarchical ensembles with data fusion methods. In *ICML-MLD 2nd International Workshop on learning from Multi-Label Data*, pages 13–20, Haifa, Israel.

Delalleau, O., Bengio, Y, Le oux, N (2005) Efficient non-parametric function induction in semi-supervised learning. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005.

Deng, M., Chen, T., and Sun, F. (2004). An integrated probabilistic model for functional prediction of proteins. *J. Comput. Biol.*, 11, 463–475.

Friedberg, I. (2006). Automated protein function prediction-the genomic challenge. *Brief. Bioinformatics*, 7, 225–242

Guan, Y., Myers, C., Hess, D., Barutcuoglu, Z., Caudy, A., and Troyanskaya, O. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(S2).

Karaoz, U. *et al.* (2004). Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl Acad. Sci. USA*, 101, 2888–2893.

References (2)

Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T., and Eisenberg, D. (1999). A combined algorithm for genome-wide prediction of protein function. *Nature*, 402, 83–86.

Mostafavi, S. and Morris, Q. (2010). Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, 26(14), 1759–1765.

Mostafavi, S., Ray, D.,Warde-Farley, D., Grouios, C., and Morris, Q. (2008). GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(S4).

Obozinski, G., Lanckriet, G., Grant, C., M., J., and Noble, W. (2008). Consistent probabilistic output for protein function prediction. *Genome Biology*, 9(S6).

Oliver, S. (2000). Guilt-by-association goes global. *Nature*, 403, 601–603.

Pavlidis, P.,Weston, J., Cai, J., and Noble,W. (2002). Learning gene functional classification from multiple data. *J. Comput. Biol.*, 9, 401–411.

Pena-Ca stillo, L., et al. (2008): A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. Genome Biology 9 S1

Re, M. and Valentini, G. (2010). Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, 8, 98–111.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7, 1601–1626.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., and Dzeroski, S. (2010). Predicting gene function using hierarchical multilabel decision tree ensembles. *BMC Bioinformatics*, 11(2).

Sharan, R. Ulitsky, I.Shamir, R. (2007) Network-based prediction of protein function, Molecular Systems Biology 3:88

Sokolov, A. and Ben-Hur, A. (2010). Hierarchical classification of Gene Ontology terms using the GOstruct method. *Journal of Bioinformatics and Computational Biology*, 8(2), 357–376.

References (3)

Szummer, M Jaakkola, T. (2001) Partially labeled classication with markov random walks. In NIPS, volume 14.

Tsochantaridis, I., Joachims, T., Hoffman, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6, 1453–1484.

Tsuda, K., Shin, H., and Scholkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics*, 21(Suppl 2), ii59–ii65.

Valentini, G. and Cesa-Bianchi, N. (2008). Hcgene: a software tool to support the hierarchical classification of genes. *Bioinformatics*, 24(5), 729–731.

Valentini, G. (2011), True Path Rule hierarchical ensembles for genome-wide gene function prediction, *IEEE ACM Transactions on Computational Biology and Bioinformatics*, 8(3), 832-847.

Vazquez, A., Flammini, A., Maritan, A., and Vespignani, A. (2003). Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 21, 697–700.

Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2), 185–214.

Zhou, D., et al. (2004) Learning with local and global consistency. In NIPS, volume 16

Zhu, X. Ghahramani, Z. , Laerty J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In ICML.