Machine learning methods for gene/protein function prediction

Giorgio Valentini
valentini@di.unimi.it

DI - Dipartimento di Informatica

Università degli Studi di Milano
Outline

• Gene Function Prediction (GFP)
• The Gene Ontology and the FunCat
• Characteristics of the GFP problem
• Computational approaches to GFP
• Machine learning methods for GFP
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” methods represent 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

Biological genome-wide gene function prediction through direct experimental assays is costly and time-consuming

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 (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 + NAD(+) = oxaloacetate + NADH.

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.
Relationships between GO terms are structured according to a DAG
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

Network based methods: predicting a specific functional term

Gene function prediction

Chosen class \( c \)

\( V = \text{genes} \)
\( w_{ij} = \) "similarity" of genes and \( j \)
\( S^+ = \text{positive examples} \)
\( S^- = \text{negative examples} \)
\( U = \text{unlabeled genes} \)

Data source (network)

\[ G = \langle V, W, S^+, S^- \rangle \]

Prediction

\[ U \]
Network-based methods

Several available methods:

• Guilt by association (Marcotte et al. 1999, Oliver et al. 2000)
• 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, Delleu 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 profiles, …

→ 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:

   \[ f(x) = w^T \phi(x) \]

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

   \[ w = \sum_i \alpha_i \phi(x_i) \Rightarrow f(x) = \sum_i \alpha_i \phi(x_i)^T \phi(x) \]

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

   \[ f(x) = \sum_i \alpha_i K(x_i, x) \]
Kernel methods for binary classification problems

Original input space $\rightarrow$ Non linear kernel mapping $\rightarrow$ Transformed feature space

$\phi$
Kernel methods for structured output spaces

A binary classifier 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 \( y = \{y_1, y_2, \ldots, y_k\} \)?

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

\[ f : X \times Y \rightarrow \mathbb{R} \]

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).
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 (*Tschantaridis et al.* 2005), applied to 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

• Hierarchical renconciliation methods (*Obozinski et al.* 2008)
• Hierarchical Bayesian cost-sensitive ensembles (*Cesa-Bianchi and Valentini*, 2010)
• True Path Rule Ensembles (*Valentini*, 2011)


References (1)


References (2)


References (3)


