Exercise 'Protein Prediction II'  
Winter Term 2010/11  
Sheet 3

General information

- Our course homepage, containing lecture slides and exercise sheets:  
  http://rostlab.informatik.tu-muenchen.de/cms/pp2/

- Time and place: **Friday, 13:30 – 15:00**, room **MI 00.11.038** ('John von Neumann')

- Contact:  
hampt@rostlab.org,  
schaefer@rostlab.org,  
vicedo@rostlab.org

- Send an email (one per group) to **us (!)** including the paths to your results (**report**, scripts, datasets) until **Friday, January 7, 2011, 9:00 am**. Scripts should be executable for us so that we can reproduce your results. 
  Please do **not** send files as email attachments, the paths to your solutions are totally sufficient. Again, everything has to be readable by us, so please check the permissions of your directories/files! Otherwise we cannot consider your solutions.

Exercise 4: Method implementation and evaluation

As written previously:

"During the course of this exercise we will establish several methods for predicting Gene Ontology (GO) terms on protein sequences. GO is a hierarchical system to annotate the function of proteins. Our methods will work by inferring GO terms by sequence homology from proteins that are already annotated. The great goal for the best performing group is to get the opportunity to participate in the CAFA (Critical Assessment of Function Annotations) challenge in January next year."

We hope that the last two exercises could make you familiar with the tools and databases needed in order to achieve the aspired goal of implementing and evaluating own methods for the CAFA competition. More specifically, in Section 1., you can now find the descriptions of three methods which you are supposed to implement and evaluate. For a complete solution, at least one version of all three is required, so that the hand-in of each group will comprise at least three different methods.
The official deadline for CAFA submissions is January 15, 2011. Since we as the Exercise leaders want to go over and evaluate your methods ourselves before submitting them, your deadline is January 7, 2011.

Due to the extent of the task, it is planned that this will be the last Exercise sheet for the next four weeks. During your work, you will certainly come across questions which are not answered on this sheet and you cannot answer yourselves. In such a case, feel free to email them to us, we will collect and try to solve them in the Exercises on Friday. Please understand that questions sent in after Thursday morning might not be answered on the Friday of the same week, but instead on the Friday of the week after. The same holds true for issues regarding the computational resources of the student machines.

Notice that you actually have more than four weeks to hand in solutions. As we will go on with the Exercises after this period, however, support from our side will be very limited from then on.

In order to successfully evaluate and optimize your methods, it is crucial to have understood the principles of training and testing sets. Please see Protein Prediction II lectures 2 and 3 for more details about this topic. Excellent solutions should optimize free parameters (see method descriptions) and implement so-called individual predictions (see 2.). We will use independent data sets after the hand-in to validate your results.

To enable us to follow your work, please also write a short report containing rough descriptions of the steps you have taken during the implementation and evaluation and hand it in together with the usual paths to your solutions. It should not exceed 5 pages and 1 page can be enough for a simple, yet correct solution. You can use pseudocode to describe more sophisticated issues.

Last, but certainly not least, CAFA wants the methods and results to follow certain formats. You can find the official rules here: http://biofunctionprediction.org/node/20
Please make sure your solutions follow these rules. In case you are not sure about certain details, again do not hesitate to ask us.

DO NOT create CAFA accounts yourself or send in solutions to anyone but the Exercise leaders.

Besides the CAFA rules, please also make your programs comply to the following conditions:

- Only 3 input parameters (Simple wrappers of programs with more parameters are allowed):
  1. a path to a blast database indexed with formatdb (with GO terms as sequence names, as introduced earlier)
  2. a path to a fasta file with target protein sequences
  3. a path to the output folder
- Only consider "is_a" and "part_of" relations of the Gene Ontology. Discard all relations between MFO and BPO terms (use the "namespace" tag to determine which ontology a term belongs to).
- Output in CAFA format, including the accuracy line if identifiers of the target protein sequences are provided as GO terms (a check for ",," is enough; see 3.)
- No hard coded sequences together with their GO annotations. The only source for annotations must be the input blast database.
- Allowed programming languages: Java, Perl, Python, PHP, C(++,#), Basic, Ruby, Bash (exceptions maybe upon request)
- [to be extended]

Happy coding!
1. Descriptions of the methods

a) Best-Hit
Make a PSI-Blast query against the complete set of GO annotated sequences in SwissProt. Use the first hit to derive the predicted GO annotations.

b) k-Nearest-Neighbor
Make a PSI-Blast query against the complete or a redundancy reduced set of GO annotated sequences in SwissProt. Use the first k hits to derive the predicted GO annotations.
Variant: Use an e-Value threshold instead of k to limit the number of annotated sequences to consider.

c) Weighted k-Nearest-Neighbor
Make a PSI-Blast query against the complete or a redundancy reduced set of GO annotated sequences in SwissProt. Use the first k hits to derive the predicted GO annotations. Do not treat every hit equally, but give a higher weight to more homologous sequences. (Example: Use the sequence identity to the power of two as the weight of a hit)

Further notes:
CAFA asks that all the nodes on the path from a predicted GO term to the root are given a reliability score between 0 and 1. For method a), one way to derive such a score could again be the use of the sequence identity of a hit. Obviously, the top hit will not contain all the parent nodes, but you could for example use the highest sequence identity of other annotated sequences which contain a particular parent node.
Ideas to derive scores for the other methods are up to you.

Free parameters of the methods include:
- Scoring schemes
- Thresholds (k, e-Value, h-Value (PSI-Blast))
- Sequence identity filtering of the training set (cd-hit) (yes/no; if yes, which identity)
- Number of PSI-Blast iterations
- ...

Let us again remind you that a proper testing setup is crucial for reliable optimizations.
As we do not lack samples, a fast and easy way to do this is to split the GO annotated part of SwissProt into three sets: training, testing and hold-out. Use training and testing for parameter optimizations. Once you have found the optimal parameter combination, use it to predict the hold-out set. Keep the hold-out set the same for all three methods. (Extended versions could further include cross-validation and a second hold-out set)

In case you do not want to optimize parameters, ask us for (presumably) reasonable fixed values.
2. Restriction to the scoring schemes

In order to make predictions more consistent with the underlying biology, we introduce the concept of individual predictions.
An individual prediction is defined as a set of nodes which contains exactly one leaf node and all of its parents. A leaf node is a predicted(!) node which is not a parent of any other predicted(!) node.
Make your scoring scheme adhere to individual predictions by implementing the following steps:
1.) Calculate a ranking of all individual predictions (e.g. based on the scores of the terms it contains).
2.) In the second best individual prediction: exclude all nodes which are already contained in the best individual prediction.
   In the third best individual prediction: exclude all nodes which are already contained in the second best individual prediction.
   etc.
3.) Make sure that the score of any term in the best individual prediction is higher than every score in the second best prediction.
   Make sure that the score of any term in the second best individual prediction is higher than every score in the third best prediction.
   etc.

Distributions of scores within one individual prediction is still up to you.

For your own time management, please note that handing in unstable or unexecutable methods will cost you more points than not having implemented individual predictions.
Also, implementing only step 1.) and outputting only the terms of the best individual prediction is better than not implementing a) at all.
In case you have already some evaluation results for methods without individual prediction, please include the resulting performance differences in your report.
3. Evaluation

In the following a way how to calculate the accuracy line for the CAFA output files.
It is supposed to be a more precise reformulation of the respective CAFA rules.

let \( M1 \) be the method to evaluate;
let \( \text{Targets} \) be the set of target input protein sequences with GO terms as their identifiers;
let \( \text{Precision} \) be the precision value in the accuracy line;
let \( \text{Recall} \) be the recall value in the accuracy line;
\( \text{CumPrec} \leq 0.0 \);
\( \text{CumRec} \leq 0.0 \);
for each target \( T_1 \) in \( \text{Targets} \):
  let \( A \) be a set of GO terms;
  for each GO term \( \text{GO}_1 \) provided in the identifier of \( T_1 \):
    add \( \text{GO}_1 \) and all of its parents to \( A \), except for the root;
  let \( B \) be the set of GO terms of \( T_1 \) as predicted by \( M1 \);
  let \( \text{Besthits} \) be the subset of \( B \) which contains the term(s) with the highest score according to \( M1 \). (all terms in \( \text{Besthits} \) must have the same score);
  let \( C \) be the union of the terms in \( \text{Besthits} \) and all of their parents, except for the root;
  let \( \text{TPs} \) be defined as the intersection of \( A \) and \( C \);
  let \( \text{FPs} \) be defined as the difference of \( C \) and \( A \);
  let \( \text{FNs} \) be defined as the difference of \( A \) and \( C \);
\( \text{Precision}_{T1} \leq \frac{|\text{TPs}|}{|\text{TPs}| + |\text{FPs}|} \);
\( \text{Recall}_{T1} \leq \frac{|\text{TPs}|}{|\text{TPs}| + |\text{FNs}|} \);
\( \text{CumPrec} \leq \text{CumPrec} + \text{Precision}_{T1} \);
\( \text{CumRec} \leq \text{CumRec} + \text{Recall}_{T1} \);
\( \text{Precision} \leq \frac{\text{CumPrec}}{|\text{Targets}|} \);
\( \text{Recall} \leq \frac{\text{CumRec}}{|\text{Targets}|} \);