CoReCo

Manual

October 05, 2013

This document explains the design, working, and usage of CoReCo application, a computational method for comparative reconstruction of genome-scale metabolic networks from protein sequence data. This document is divided into the following sections:

1. Software requirements
2. Data requirements
3. Directory structure
4. How to run the application?
5. Software algorithm

Software requirements:

1. Download CoReCo application

2. Install Python and Perl

3. Download InterproScan from ftp://ftp.ebi.ac.uk/pub/databases/interpro/iprscan.

   For installation instruction:

   - ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/index.html

   - ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/4/Installing_InterProScan.txt

4. NCBI Blast+ toolkit: Download the latest NCBI Blast+ toolkit from:

ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

5. Install R - statistical tool package: http://www.r-project.org/

6. Install python phylogenetic tree library from http://users-birc.au.dk/mailund/newick.html and install it as:

   python setup.py install –user

7. Install libsbml

   - download software from http://sourceforge.net/projects/sbml/files/libsbml/5.7.0/stable/
   - unzip libSBML-5.7.0-core-plus-packages-src.zip file
   - cd /path-to-Source-dir/ of libsbml and run the following. (If you have the root right, you don't need to specify --prefix)

     ```
     ./configure --prefix="~/.local" --with-python
     make
     make install
     ```

Data Requirements:

1. Download uniprot_sprot.fasta.gz from ftp://ftp.uniprot.org/pub/databases/uniprot/knowledgebase and extract the uniprot_sprot.fasta.gz to "< path to CoReCo >/data/Uniprot_EC_GO_Data/".

   If a user already has uniprot_sprot BLAST database installed, then change the name of the uniprot_sprot blast db in ProjectDir.py file. Following line should be changed:

   ```
   uniprot_blast_db = NGS_Util.createFilePath(orgBlastDBDir, "uniprot")  # In the line the
   # value uniprot needs to be changed to your local uniprot_sprot blast database.
   ```

2. Download uniprot_sport.dat file from ftp://ftp.uniprot.org/pub/databases/uniprot/knowledgebase and extract it to "< path to CoReCo >/data/Uniprot_EC_GO_Data/".

3. download ec2go.txt file from http//www.geneontology.org/external2go/ec2go and extract it to "< path to CoReCo >/data/Uniprot_EC_GO_Data/".

   The application already comes with a ec2go.txt file. But if a user wants to use a new file then the user can download the ec2go.txt file as described above

4. The users need to license and download the KEGG database via the KEGG website http://www.kegg.jp/kegg/download/. Download the following from KEGG ligand
   - compound (included in compound.tar.gz)
   - reaction (included in reaction.tar.gz)

- enzymes (included in enzymes.tar.gz)
- mol/ (included in compound.tar.gz)

The KEGG database flat files need to be installed in < path to coreco >/data/Kegg/.

5. The application already comes with GTG nrdb40 database. The GTG data is stored in "< path to coreco >/data/GTG_database/".

Application Directory Structure:

The application has the following directory structure:

Dir	Description
< path to coreco >/bin/	This directory contains the scripts needed to run the application
< path to coreco >/data/	This directory contains the input data to the application
< path to coreco >/doc/	This directory contains documents relating to the application
< path to coreco >/results/	This directory contains the results of the application

Following is the description of each directory:

1. < path to coreco >/bin/

Dir / File	Description
Blast_scripts/	BLAST scripts directory
GTG_scripts/	GTG scripts directory
Iprscan_scripts/	Interpro scan scripts directory
model_reconstruction_pipeline/	Model reconstruction pipeline scripts directory
model_training_scripts/	Model training scripts directory
plotting/	Plotting scripts directory
kegg-parsing/	Kegg parsing scripts
reconstruction_scripts/	Network reconstruction scripts directory
MetabolicRecontructionPipeline.py	Main file to run model reconstruction pipeline
ProjectDir.py	This file contains paths for the reconstruction pipeline input and output paths.

ScriptsDir.py	This file contains paths to the scripts used in the pipeline and tools installed. **A user needs to set the following paths in this file**: a. projectDir: <path to coreco> b. BlastDir:<ncbi-blast-2.2.28+ dir>/bin/ c. BlastDBDir : <blast database directory> d. BlastDustDir: <blast database directory> e. IprscanDir:<iprscan dir>/bin/

2. < path to coreco >/data/

Dir / Files	**Description**
GTG_database/	GTG data directory
Kegg/	Kegg data directory
Kegg/kegg-no-general/	The modified version of the KEGG database used by the reconstruction pipeline. This folders and all files in it are created during the preprocessing.
Kegg/aux/	Contains auxiliary files related to KEGG. The files cofactors, empty and sources-augmented are provided as part of the distribution. The files kegg-compounds, pathway-names, reaction-pathways and ec-to-pathways are created from the KEGG flat files at the same time as t kegg-no-general/ folder. The ec-list-augmented.txt file is manually modified version of kegg-no-general/ec-list.txt. The reconstruction algorithm is reading Kegg/aux/ec-list-augmented.txt to map E.C. numbers to reactions.
org_sequence_db/	Fasta protein sequences data directory. The name of the fasta files in the directory should be “<organism name>.faa”, where, organism name should be the same a in org_list.backup file. The description of org_list.backup file is given below.
Uniprot_EC_GO_Data/	This directory contains Uniprot, EC, and Go data
org_list.backup	This file contains the organism four letter name abbrevia and organism name separated by tab. E.g. Anig Aspergillus_niger

	Nfis Neosartorya_fischeri Pgra Puccinia_graminis **The contents of this file needs to be provided by the user.**
seq_org_list.txt	This file contains the description about organism and its fasta sequence ids
taxonomy	This file contains the organism four letter name abbreviat and its NCBI taxonomy number. The file is space separated with the header line “#NCBITaxonomy SpeciesAbbr”. Following are the example values of the fi NCBITaxonomy SpeciesAbbr 5061 Anig 36630 Nfis 5297 Pgra **The contents of this file needs to be provided by the user.**
network_reconstruction_org_list.txt	This file contains the list of the species for which you wa to run the network reconstruction. Presumably all, but as the reconstruction step is quite time consuming, this file c be used to control which species are reconstructed E.g. Anig Nfis Pgra **The contents of this file needs to be provided by the user.**
tree.txt	The phylogenetic tree file. The tree could be either in .ne or in .newick format. **The contents of this file needs to be provided by the user.**

3. < path to coreco >/results/

Dir	Description
blast_joint_results/	This directory contains the final blast results
blast_results/	This directory contains the intermediate blast results

GTG_best_hits/	This directory contains the intermediate GTG results
GTG_blast_results/	This directory contains the intermediate GTG results
GTG_knn/	This directory contains the final GTG results
iprscan_ec_raw_results/	This directory contains the final InterproScan results
iprscan_results/	This directory contains the intermediate InterproScan resu
ModelTraining/	This directory contains the model training and network reconstruction results
ModelTraining/Model/reco/	This directory contains the network reconstruction results species specified in network_reconstruction_org_list.txt

How to execute the software?

1. Set the following paths in "< path to coreco >/bin/Scripts.py"

    - projectDir : <path to coreco>
    - BlastDir : <ncbi-blast-2.2.28+ dir>/bin/
    - BlastDBDir : <blast database directory>
    - BlastDustDir : <blast database directory>
    - IprscanDir : <iprscan dir>/bin/

2. Copy your fasta sequences to "< path to coreco >/data/org_sequence_db/". The name of the fasta files should be "<organism name>.faa", where, the organism name should be the same as its long name in org_list.backup file

3. Change the entries of the file org_list.backup file in "< path to coreco >/data/" directory. Append organism four letter name abbreviation and organism name separated by tab in the file as below:

    ```
    Anig   Aspergillus_niger
    Nfis   Neosartorya_fischeri
    Pgra   Puccinia_graminis
    ```

4. Set taxonomy ids of the organisms in taxonomy file in "< path to coreco >/data/" directory. The taxonomy file contains the organism four letter name abbreviation and its taxonomy value separated by space with the header line "#NCBITaxonomy SpeciesAbbr". Following are the example values of the file:

```
5061 Anig
36630 Nfis
5297 Pgra
```

5. List the organism in "< path to coreco >/data/network_reconstruction_org_list.txt" file for which the application should generate the gapless metabolic model. network_reconstruction_org_list.txt file contains organisms four letter name abbreviation list as below:

```
Anig
Nfis
Pgra
```

6. Run the software as: python < path to coreco >/bin/MetabolicRecontructionPipeline.py

Software workflow

1. Preprocessing:

    a. Create uniprot_sprot blast database
    b. Extract EC ids from uniprot_sprot.dat file
    c. Create GTG nrdb40 blast database
    d. Extract sequence ids from the fasta file for each organism and append it to the seq_org_list.txt.
    e. Generate a modified version of KEGG data: remove "general" reactions, try to balance the remaining reactions and compute atom mappings for balanced reactions. The preprocessing script uses a version of the python GLPK library that (for us) only worked on a 32-bit computer.

2. Compute scores for the probability of observing enzymes (by E.C. number).

    For every organism in org_list.backup file do:

    a. Two way blast between organism sequences and uniprot_sprot BLAST database if "blast_joint_results/" directory contains no results for such organism.

        i. Create protein BLAST database for the organism sequences. The BLAST database will be created with the organism name. The database will not be created if there already is a database with the organism name.
        ii. Do two way blast between organism sequences and uniprot_sprot
        iii. Map the best hit UniprotIDs to E.C. numbers using "< path to coreco >/data/Uniprot_EC_GO_Data/ec_files.txt" file.

    b. To way blast between organism sequences and GTG nrdb40 BLAST database if "GTG_knn/" directory contains no results for such organism.

        i. Map the best hit UniprotIDs to E.C. numbers using "< path to coreco >/data/Uniprot_EC_GO_Data/ec_files.txt" file.

c. Run interproscan for organism’s sequences if “iprscan_ec_raw_results/” directory contains no results for such organism.

   i. Map the predicted GO categories to E.C. numbers using “< path to coreco >/data/Uniprot_EC_GO_Data/ec2go.txt” file.

The outputs of step 2, separately from each information source, are scores for E.C. numbers in each species.

3. Generate probabilistic model with the outputs in steps 2. This step is taking into account the phylogeny of the species and requires the “< path to coreco >/data/tree.txt” file.

   The output of step 3 is combined probability for ECs in each species and in each ancestral species.

4. Reconstruct a gapless metabolic network for the specified organisms using the outputs from the previous step. The reconstruction is done for each species separately.

   This step requires as input the KEGG reactions database as we all the atom mappings for each reaction in the database: <path to coreco>/data/Kegg/kegg-no-general/reaction” and <path to coreco>/data/Kegg/kegg-no-general/atommaps/*”

   To link the EC based scores from previous steps to KEGG reactions and metabolic pathways this step needs the EC number to KEGG reactionID mapping file “< path to coreco >/data/Kegg/aux/ec_list_augmented.txt”

   In addition, the step reads files from “< path to coreco >/data/Kegg/aux/”. The manually created files “cofactors”, “empty” and “sources-augmented” are provided as part of the distribution. The file “kegg-compounds” is created during the preprocessing.

Tips:

1. A user can run the application for as many species he/she wants providing he updates the tree.txt file, organism_list.backup file, network_reconstruction_org_list.txt file, taxonomy file, and org_sequence_db/ directory.

2. A user can update network_reconstruction_org_list.txt and generate the gapless metabolic network for the updated list