GOMAP Documentation

Kokulapalan Wimalanathan

Jul 20, 2018

Contents

Danger: This documentation is a work in progress and will be completed soon. Please bear with us untill the documentation catches up with code development

Contents:

Introduction

1 What is GOMAP?

Gene **O**ntology - **M**eta **A**nnotator for **P**lants (**GOMAP**) is a pipeline that annotates GO terms to plant protein sequences. The pipeline uses 3 different approaches to annotate GO terms to plant proteins and uses a mix of custom code and existing software tools to assign GO terms. The pipeline was designed to create a high confidence GO annotation dataset for reference proteomes, and it is recommended that the pipeline in it's current form used to annotate proteomes. The main pipeline is written in Python and R, and other software tools used will be briefly described in the next Section.

2 What annotation methods are used to assign GO terms?

2.1 Sequence-similarity based methods

Sequence-similarity based GO annotations were performed using the reciprocal-best-hit method against two different datasets, namely TAIR and UniProt. The NCBI BLASTP tool will be used reciprocally to search for similar sequences between the protein sequences of target species being annotated and other datasets. The results from BLASTP search will be processed using R script to determine the reciprocal-best-hits and assign GO terms from TAIR and UniProt to the target species.

2.2 Domain-presence based methods

Putative protein domains will be assigned to the protein sequence using InterProScan5 pipeline. InterProScan5 is a java based pipeline that finds protein domain signatures in target sequences and assigns GO terms based on the presence of the protein signatures.

2.3 Mixed-method pipelines or tools

Three top performing pipelines/tools which have competed in the first iteration of the CAFA competition will be used to assign GO terms to proteins. These tools are Argot2.5, and PANNZER. Each of the tools have specific requirements, setup instructions and pre-processing steps. The details of these steps will be explained in the following sections.

Requirements The [DOI release] of the GOMAP pipeline contains code, software, and data files to run the pipeline. Although, there are some basic requirements which need to be installed. The requirements that have to be installed are listed below.

3 What are the requirements that need to be installed to run GOMAP?

- Hardware
  - Storage
    - minimum: 250GB
    - recommended: $\geq$300GB
  - Memory
    - minimum: 16 GB

    * recommended: ≥32 GB

- Software
  - OS
    * linux
  - Programming Languages
    * R v3.4
    * Python v2
    * Java v1.8
    * Perl
  - Software
    * MATLAB ≥v2016a
    * MySQL/MariaDB
  - Python Packages
    * biopython
    * numpy
    * scipy
    * MySQL-python
  - R packages
    * ontologyIndex
    * data.table
    * ggplot2
    * futile.logger
    * jsonlite

4 What are the software tools needed to run specific annotation methods?

- Sequence-similarity
  - BLASTP
- Domain-presence
  - InterProScan5
- Mixed-method Pipelines
  - FANN-GO
    * MATLAB[†]
    * BLASTP
  - PANNZER

    * MySQL/MariaDB[†]
    * BLASTP
  - Argot2
    * Hmmer
    * BLASTP
    * Web browser[†‡]

[†]Part of requirements installed as mentioned in this section
[‡]To submit jobs to batch processing
The pipeline file downloaded from CyVerse contains the data files and software tools to run the process on a given protein sequence fasta file. The disk space required for the pipeline is large (~160GB) and when it runs it will require close to 300GB of disk space.

Setup

5 What are the steps needed to setup the pipeline?

1. Install dependencies
2. Install required packages for R and Python

- A shell script is provided to make the installation of the packages easy.
- Run `bash install/install_packages.sh` from GAMER-pipeline directory
- Users with a python2 virtual environment please activate before running the script

1. Setup MySQL database for Pannzer

- Create a database named pannzer
- Create a user names pannzer and grant all privileges on the database pannzer
- The password should be `pannzer`
- If you decide to change any of this, please update the config.json [mix-meth.PANNZER.database] file accordingly.

Running GOMAP

6 How to run GOMAP?

GOMAP is run in two steps using pipeline1.py and pipleine2.py. First part of the pipeline runs the Sequence-similarity methods and domain-based methods, and FANN-GO and PANNZER. It also runs the pre-processing steps for Argot2.5. Second part of the pipeline processes results from different methods and compiles the final GO annotation dataset from all differnt approaches. The main steps are given below.

1. Add the protein fasta file to `input/raw/`
2. Make necessary changes to the config.json file

- Update the `work_dir` in the pipeline section

- Update the `input` section
  - Give the correct input FASTA file name
  - If the fasta contains multiple transcripts per gene then put the fasta in the `input/raw` directory and set the `raw_fasta` parameter
  - If the fasta file contains only on transcript per gene put it in the `input/filt` directory, and set the `fasta` parameter
  - Update the species, inbred and version parameters for your species
- [Optional] Update the `seq-sim` section
  - (All the files should be already processed in this section)
- [Optional] Update the `mix-meth` section
  - (All the files and fields should be already set, except changes to database section for PANNZER)
- [Optional] Update `blast` and `hmmer` sections
  - This is to enable the correct number cpu threads for these software
- All other sections should only be updated if things have been drastically changed.

1. execute `python pipeline1.py config.json`

- The pipeline will generate a number of intermidiate output files
- Especially the mixed-method tools will require the input fasta to be split into smaller chunks. the chunks will be numbered serially. (e.g. test.1.fa, test.2.fa)
- Argot 2.5 tool will NOT be executed within the pipeline

1. Submit the files in `mixed-meth/argot2.5/blast` and `mixed-meth/argot2.5/hmmer` using correct pairing
2. Extract the Argot2.5 result files for each job, in the `mixed-meth/argot2.5/results` directory and rename with correct prefix

- Argot2.5 names all results as `argot_results_ts0.tsv` so the file should be renamed correctly (e.g. test.1.tsv, test.2.tsv)
- Please do not leave any other file in the argot2.5 results directory, otherwise it will influence certain metrics.

1. execute `python pipeline2.py config.json`

7 What are the steps needed to setup the pipeline?

1. Install dependencies
2. Install required packages for R and Python

- A shell script is provided to make the installation of the packages easy.
- Run `bash install/install_packages.sh` from GOMAP directory
- Users with a python2 virtual environment please activate before running the script

1. Setup MySQL database for Pannzer

- Create a database named pannzer
- Create a user names pannzer and grant all privileges on the database pannzer

- The password should be `pannzer`
- If you decide to change any of this, please update the config.json [mix-meth.PANNZER.database] file accordingly.

8 How to run the GOMAP?

GOMAP is run in two steps using pipeline1.py and pipleine2.py. First part of the pipeline runs the Sequence-similarity methods and domain-based methods, and FANN-GO and PANNZER. It also runs the pre-processing steps for Argot2.5. Second part of the pipeline processes results from different methods and compiles the final GO annotation dataset from all differnt approaches. The main steps are given below.

1. Add the protein fasta file to `input/raw/`
2. Make necessary changes to the config.json file

- Update the `work_dir` in the pipeline section
- Update the `input` section
  - Give the correct input FASTA file name
  - If the fasta contains multiple transcripts per gene then put the fasta in the `input/raw` directory and set the `raw_fasta` parameter
  - If the fasta file contains only on transcript per gene put it in the `input/filt` directory, and set the `fasta` parameter
  - Update the species, inbred and version parameters for your species
- [Optional] Update the `seq-sim` section
  - (All the files should be already processed in this section)
- [Optional] Update the `mix-meth` section
  - (All the files and fields should be already set, except changes to database section for PANNZER)
- [Optional] Update `blast` and `hmmer` sections
  - This is to enable the correct number cpu threads for these software
- All other sections should only be updated if things have been drastically changed.

1. execute `python pipeline1.py config.json`

- The pipeline will generate a number of intermidiate output files
- Especially the mixed-method tools will require the input fasta to be split into smaller chunks. the chunks will be numbered serially. (e.g. test.1.fa, test.2.fa)
- Argot 2.5 tool will NOT be executed within the pipeline

1. Submit the files in `mixed-meth/argot2.5/blast` and `mixed-meth/argot2.5/hmmer` using correct pairing
2. Extract the Argot2.5 result files for each job, in the `mixed-meth/argot2.5/results` directory and rename with correct prefix

- Argot2.5 names all results as `argot_results_ts0.tsv` so the file should be renamed correctly (e.g. test.1.tsv, test.2.tsv)
- Please do not leave any other file in the argot2.5 results directory, otherwise it will influence certain metrics.

1. execute `python pipeline2.py config.json`