MetaPathways

BCB2

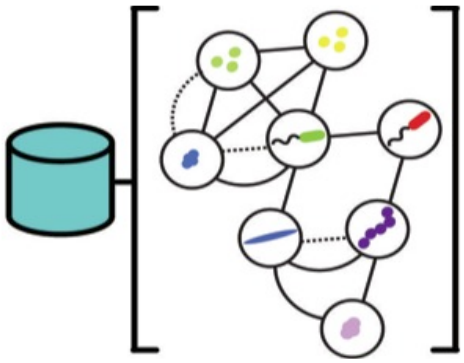
Sep 23, 2023

DOCUMENTATION

CHAPTER

ONE

OVERVIEW



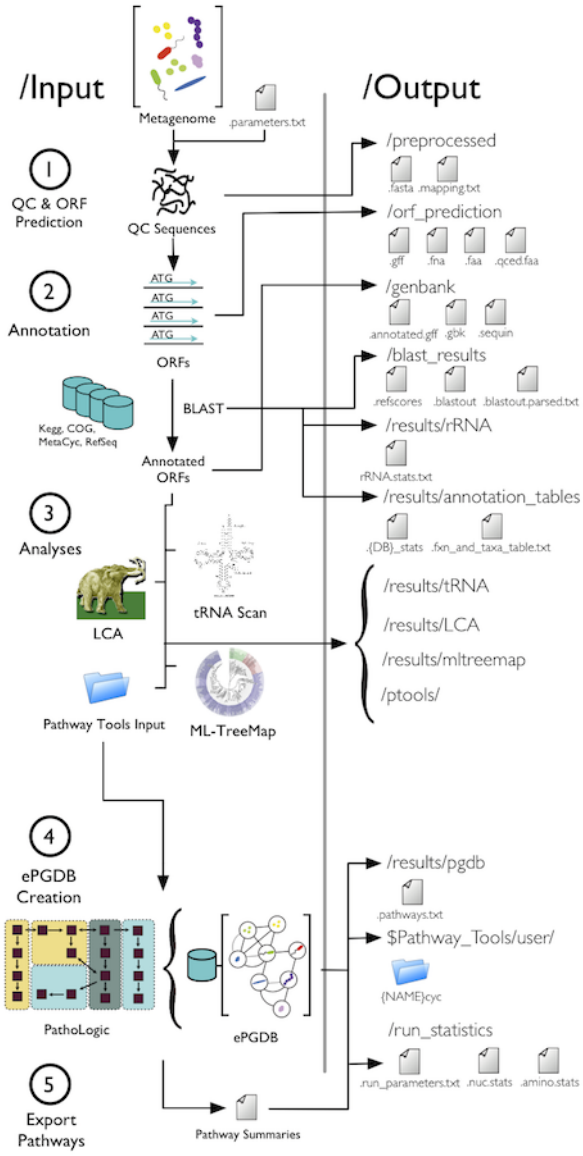
MetaPathways [CIT2002] is a meta'omic analysis pipeline for the annotation and analysis for environmental sequence information. MetaPathways include metagenomic or metatranscriptomic sequence data in one of several file formats (.fasta, .gff, or .gbk). The pipeline consists of five operational stages including

1.1 Pipeline Overview

MetaPathways is composed of five general stages, encompassing a number of analytical or data handling steps **(Figure 1)**:

1. **QC and ORF Prediction**: Here MetaPathways performs basic quality control (QC) including removing duplicate sequences and sequence trimming. Open Reading Frame (ORF) prediction is then performed on the QC'ed sequences using Prodigal [PRODIGAL2010] or GeneMark [GeneMark12]. The final translated ORFs are now also trimmed according to a user-defined setting.
    - MetaPathways steps: *PREPROCESS INPUT*, *ORF PREDICTION*, and *FILTER AMINOS*
2. **Functional and Taxonomic Annotation**: Using seed-and-extend homology search algorithms (B)LAST [BLAST90], [LAST11], MetaPathways can be used to conduct searches against functional and taxonomic databases.
    - MetaPathways steps: *FUNC SEARCH*, *PARSE FUNC SEARCH*, *SCAN rRNA*, and *ANNOTATE ORFS*
3. **Analyses**: After sequence annotation, MetaPathways performs further taxonomic analyses including the Lowest Common Ancestor (LCA) algorithm [MEGAN07] and tRNA Scan [TRNASCAN97], and prepares detected annotations for environmental Pathway/Genome database (ePGDB) creation via Pathway Tools.
    - MetaPathways Steps: *PATHOLOGIC INPUT*, *CREATE ANNOT REPORTS*, and *COMPUTE RPKM*.

4. **ePGDB Creation**: MetaPathways then predicts MetaCyc pathways using the Pathway Tools software and its pathway prediction algorithm PathoLogic [KARP11], resulting in the creation of an environmental Pathway/Genome Database (ePGDB), an integrative data structure of sequences, genes, pathways, and literature annotations for integrative interpretation.
    - MetaPathways Steps: *BUILD ePGDB*
5. **Pathway Export**: Here MetaCyc pathways or reactions are exported in a tabular format for downstream analysis. *As of the v2.5 release, MetaPathways will perform this step automatically.*
    - MetaPathways Steps: *BUILD ePGDB*



1.2 Output Format

1.3 Visualizing Output

CHAPTER

TWO

INSTALLATION

MetaPathways supports installing the software using Conda and Pip in a 64-bit Linux environment, or from a container image that can be used with Docker or Singularity. If you do not have administrator (i.e., "root") access to your computer, we recommend that users install MiniConda if they do not already have it set up. For users wanting to use MetaPathways in an academic grid computing environment, we recommend using the container image *via* Singularity. Below please find a description of how to install MetaPathways using the two supported options:

2.1 Container Install

Our container images are hosted at Quay.io. The following commands assume that you are already familiar with installing and running Docker containers via the `docker` or `singularity` executables:

Using Docker:

```
sudo docker pull quay.io/hallamlab/metapathways
```

Using Singularity:

```
singularity build metapathways.sif docker://quay.io/hallamlab/metapathways:latest
```

More advanced container-related commands are available as Make targets in the `Makefile`.

2.2 Installing with Pip and Conda

2.2.1 Summary

Assuming that you have all prerequisites satisfied, installng can be as simple as:

```
conda create --name metapathways python=3.10
conda activate metapathways
pip3 install git+https://bitbucket.org/BCB2/metapathways.git@dev#egg=MetaPathways
metapathways-install-deps.sh
```

Read on to learn the details.

2.2.2 Detailed Install

We currently offer a way to use Pip to install the MetaPathways Python package, along with using Conda to install all dependencies. We do not yet have a Conda package for MetaPathways. It is in the works for a future release.

For this to work, we assume that you have the following already set up in your command line environment:

- You have Python 3 (`python3`) and `pip3` installed
- You have already installed Conda, and it is activated
- Development files for `zlib`, `liblzma` and ``libbz2` (required to install `PySAM` via `pip`)
- You have `wget` installed

If you are using a version of Linux that uses `apt`, and you have root access, then you can execute the following to get all of the dependencies except Conda:

```
sudo apt-get update -y
sudo apt-get install -y \
               python3 \
               python3-pip \
               zlib1g-dev \
               liblzma-dev \
               libbz2-dev \
               wget
```

Installing Python Package as Root

If you have root/administrator access, install the MetaPathways Python package using the following command:

```
pip3 install git+https://bitbucket.org/BCB2/metapathways.git@dev#egg=MetaPathways
```

Installing Python Package as an Unpriviledged User

Use this form to install the package to the user's home directory:

```
pip3 install --user git+https://bitbucket.org/BCB2/metapathways.git@dev#egg=MetaPathways
```

Make sure to add `$HOME/.local/bin` to your `$PATH` environment variable. This will allow you to use the programs without having to type the full path each time.

Conda-Based Setup

Once you have installed the Python package, you will have the following executables either in the system Python install path, or in `~/.local/bin`, so be sure to add those paths to your `$PATH` environment variable.

```
MetaPathways
metapathways-install-deps.sh
metapathways-data-install.sh
metacount
fastal
fastdb
```

Execute `metapathways-install-deps.sh` to install pipeline dependencies using Conda.

2.3 Reference Sequences

2.3.1 Summary

Assuming that you have MetaPathways installed, installing the reference DB can be as simple as:

```
metapathways-data-install.sh /media/ref-db-dir stage_fast_full
```

Read on for detailed instructons.

2.3.2 Details

MetaPathways relies on reference databases of sequences to assign functional and taxonomic annotations to the user's sequences. The reference databases, and the index files for each database, take up a significant amount of disk storage. See below for an anecdotal example.

You cannot install these large reference databases within the container, though. You should have a directory on a disk with plenty of capacity, and use Docker's and Singularity's bind options to mount that external directory within the container. Here's an example using Singularity:

```
singularity shell --bind /mnt/sandbox/user:/data docker://quay.io/hallamlab/
↪metapathways:latest
```

The above example binds the host operating system's `/mnt/sandbox/user` directory within the running container as `/data`.

Warning: Circa 2021-10, using a beefy computer with many cores and plenty of RAM, performing the staging of the full Blast databases may take an hour, and staging the full set of FAST databases will take around *24 hours*. The Blast `refseq_protein` databases take up ~90 GB of disk capacity, while the FAST `refseq_protein` database takes up ~375 GB. The combination of other staged databases (including both Blast and FAST versions) consumes an additional ~20 GB. Please make sure you have adequate disk capacity before starting the database staging.

We use `Snakemake` to automate the staging of reference databases needed by MetaPathways. We have installed `Snakemake` via Conda. If you are using the Docker container, then Conda is already initialized. If you are using the container via Singularity, you must first initialize Conda as follows (note the space between the period character, and the first slash character):

```
. /opt/conda/etc/profile.d/conda.sh
```

Now we can run the `metapathways-data-install.sh` script

```
metapathways-data-install.sh /media/ref-db-dir stage_fast_lite
```

... where `/media/ref-db-dir` is the reference database installation directory (make sure this directory has adequate capacity for the data to be installed).

Above we issued the `stage_fast_lite` command to Snakemake, as an example that runs quickly. There are actually four options for staging the data:

- All databases, indexed for use with Blast: `stage_blast_full`
- All databases except RefSeq Proteome, indexed for use with Blast: `stage_blast_lite`
- All databases, indexed for use with FAST: `stage_fast_full`
- All databases except RefSeq Proteome, indexed for use with FAST: `stage_fast_lite`

So, first decide whether you want to use Blast or FAST, and then decide whether you have the disk space and the install time to install the NCBI RefSeq Proteome reference database. FAST runs faster than Blast, with comparable sensitivity. And the RefSeq Proteome is currently required for MetaPathways to accurately annotate contigs taxonomically. Thus, we recommend running `stage_fast_full`, if you have the disk storage and the time to let it run.

CHAPTER

THREE

RUNNING METAPATHWAYS

3.1 Input

MetaPathways inputs are fasta files provided in an input folder. The file names must end with a *.fasta* or *.fas*. These fasta files contains the contigs or DNA sequences from assembling.

3.2 Parameter File

The parameter file must indicate the setting for any MetaPathways run. An example paramter file can be downloaded as

```
$ wget  https://github.com/kishori82/MetaPathways_Python.3.0/raw/kmk-develop/data/text/
↪template_param.txt
```

Below we describe the settings in the parameter file.

3.3 Run

As an illustration we donwload a small input file *testsample1.fasta* in a folder named *mp_input* and we want the output in a folder names *mp_output*

```
$ mkdir mp_input
$ cd mp_input
$ wget https://github.com/kishori82/MetaPathways_Python.3.0/raw/kmk-develop/data/
↪testdata/testsample1.fasta
$ cd ..
```

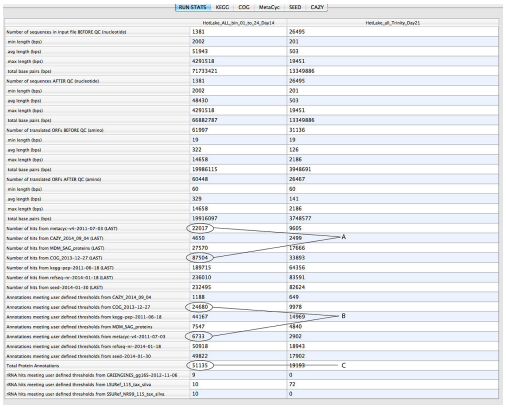
Now we kick off a run as

```
$ MetaPathways --input mp_input --output mp_output -p template_param.txt -d ~/
↪MetaPathways_DBs/
```

CHAPTER

FOUR

PHANDI GUI OVERVIEW



MetaPathways (Phandi) GUI viwers is a stand-along desktop tool for inspecting and exporting the large amount of outputs produced by the MetaPathways pipeline.

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

SIX

CONTACT

contact

BIBLIOGRAPHY

[CIT2002] K. M. Konwar, N. W. Hanson, A. P. Pagé, S. J. Hallam, MetaPathways: a modular pipeline for constructing pathway/genome databases from environmental sequence information. BMC Bioinformatics 14, 202 (2013) http://www.biomedcentral.com/1471-2105/14/202

[PRODIGAL2010] D. Hyatt et al., Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 11, 119 (2010).

[GeneMark12] D. Hyatt, P. F. LoCascio, L. J. Hauser, E. C. Uberbacher, Gene and translation initiation site prediction in metagenomic sequences. Bioinformatics 28, 2223–2230 (2012).

[BLAST90] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic local alignment search tool. J Mol Biol 215, 403–410 (1990).

[LAST11] S. M. Kiełbasa, R. Wan, K. Sato, P. Horton, M. C. Frith, Adaptive seeds tame genomic sequence comparison. Genome Res 21, 487–493 (2011).

[MEGAN07] D. H. Huson, A. F. Auch, J. Qi, S. C. Schuster, MEGAN analysis of metagenomic data. Genome Res 17, 377–386 (2007).

[TRNASCAN97] T. M. Lowe, S. R. Eddy, tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Research 25, 0955–0964 (1997).

[KARP11] P. D. Karp, M. Latendresse, R. Caspi, The pathway tools pathway prediction algorithm. Stand Genomic Sci 5, 424–429 (2011).