GenBank Database Pipeline Guideline

Shanti Mechery

November 2019

Contents

1 Introduction

The GenBank DataBase Pipeline has two purposes: to map source site data from the National Center for Biotechnology Information's GenBank Database to their preferred name according to the the National Institute of Health's Unified Medical Language System® and to create a matrix of bacteria species and host species data, both also from the National Center for Biotechnology Information's GenBank Database, pertaining to a specific source site. The first part is the Upstream Process and the second part is the Downstream Process. This matrix created in the Downstream Process will be a part of a NEXUS file, which can be used as input for PAUP*, a phylogenetic analysis software that creates phlyogentic trees. Each section of this guide explains the processes in further detail. If the most up-to-date input data from the GenBank Database and the Unified Medical Language System® for the pipeline is needed, start with the Upstream Process. Note this process only needs to be run once. If all data is up-to-date, only the Downstream process needs to be run. It is advised that all work is put in a root directory named GenBank.

The data from the GenBank Database can be downloaded using a GenBank Loader to a MySQL database (process is detailed in the Upstream Process section). In the GenBank Database, an entry has the following information: Locus, Organism, Source Site (isolation_source or tissue_type), and Host. The GenBank Loader imports this data into a MySQL database named 'genbank' with the following tables: annotations, authors, basic, dbxrefs, journals, keywords. Only the annotations table will be needed, which has the following columns: partitionKey, locus, name, indexedValue, and value. Each row of the table contains a single piece of type of information (name column) with a value (value column) pertaining to a locus number (locus column). The types of information included in the name column are organism, source site (isolation_source or tissue_type), and host. For the pipeline, use cleaned versions of the host data, which can be found in the table cleaned_host_annotations.

2 Upstream Process

The following are the steps needed to run the Upstream Process. In order to run these steps, MySQL, Julia (the progrming language), and MetaMap (https://metamap.nlm.nih.gov/Installation.shtml) must be installed on the computer where all work will be done. Make sure the MetaMap files (in the folder public_mm) is a sub-directory of the GenBank folder. The last argument of all julia commands in this guide is the name of the output files. It is suggested to use the same names of these output files, although other file names may be used.

1. If the GenBank Database is not loaded to MySQL yet, do so using this guide: https://bitbucket.org/UVM-BIRD/genbank-loader/src/master.

2. Query a table for all source site and for bacteria species data from the GenBank database. For testing, the source sites that were used were tissue_type and isolation_source. Below is how each table was queried with their associated CSV file name. It is critical that all of the column names are the same as what is stated below, or else the pipeline will not work.

   - `isolationLocus.csv`
     - query: `select locus,value from annotations where name= "isolation_source";`
     - column names: locus, value
     - This CSV contains the names of all isolation_source data (value) and their associated locus number (locus)

   - `tissueLocus.csv`
     - query: `select locus,value from annotations where name= "tissue_type";`
     - column names: locus, value
     - This CSV contains the names of all tissue_type data (value) and their associated locus number (locus).

     If you would like to use a difference source site, use the same query but change the `where` condition with the name of the source site.

   - `allBacteriaSpecies.csv`
     - query: `select locus,value from annotations where name= "organism";`
     - column names: locus, value
     - This CSV contains the names of all bacteria species data (value) and their associated locus number (locus).

   - `cleaned_host_annotations.csv`
     - query: `select locus,cleanScientificName from cleaned_host_annotation;`
     - column names: locus, cleanScientificName
     - This CSV contains all cleaned host species data (cleanScientificName) and their associated locus number (locus).

3. MetaMap can take a file of inputs comma separated (the file must be a txt file). The `upstream_1_valueCSV_parser.jl` script takes in a source site CSV created from the previous step (`tissueLocus.csv`, `isolationLocus.csv` or `sourceSiteLocus.csv`) and outputs the appropriate MetaMap input format to a TXT file (`sourceSite_MetaMap_input.txt`). Use the following command to run the script:

```
$ julia upstream_1_valueCSV_parser.jl sourceSiteLocus.csv sourceSite_
MetaMap_input.txt
```

4. Run MetaMap with the output TXT file created as the input with the following commands at the GenBank directory:

   ```
   $ ./public_mm/bin/wsdserverctl start
   $ ./public_mm/bin/skrmedpostctl start
   $ ./public_mm/bin/metamap -I sourceSite_MetaMap_input.txt sourceSite_
   MetaMap_output.txt
   ```

   All of MetaMap's output will be printed to `sourceSite_MetaMap_output.txt`. Make sure to use the `-I` flag, which includes the Concept Unique Identifiers (CUI) for each mapping. When the program is done, run the following commands:

   ```
   $ ./public_mm/bin/wsdserverctl stop
   $ ./public_mm/bin/skrmedpostctl stop
   ```

   In order to make this run this step faster, it is advised to split the input file depending on its size and use the Unix screen function to run the MetaMap program multiple times with different inputs. Give all input files an associated output file. For more information on how to run MetMap, visit https://metamap.nlm.nih.gov/Docs/MM_2016_Usage.pdf

5. In order to parse the MetaMap output files, use the `upstream_2_metamapParser.jl` file. Each UMLS concept has a Semantic Type, and the script can filter out the concepts based on a set of Semantic Types. A file of Semantic Types associated with body parts is provided (`semanticTypes.txt`). However, another set of Semantic Types can be used, as long as it is in a file with the same file format as `semanticTypes.txt`.This script creates two CSVs in the following format:

   - `sourceSite_MetaMap_parsed.csv`
     - columns: original, preferred, semanticType
     - original: original source site as found in GenBank
     - preferred: source site's preferred MetaMap name, which is all of the concept names concatenated together
     - semanticType: semantic types associated with each MetaMap concept
   - `sourceSite_CUI.csv`
     - columns: original, concept, CUI, semanticType
     - original: original source site as found in GenBank

- concept: mapping associated with the original source site's name
- CUI: Concept Unique Identifiers
- semanticType: semantic type associated with preferred MetaMap Name

Run the script with the following command:

```
$ julia upstream_2_metamapParser.jl sourceSite_MetaMap_output.
txt semanticTypes.txt sourceSite_MetaMap_parsed.csv sourceSite_
CUI.csv
```

6. When the MetaMap output is parsed, the the appropiate bacteria species data needs to be filtered from the `allBacteriaSpecies.csv` file for each source site. This file will most likely be very large and should be split into smaller files. In order to get the appropriate bacteria species for each source site, run the `upstream_3_createBacteriaSpecieFile.jl` script with the following command:

   ```
   $ julia upstream_3_createBacteriaSpecieFile.jl sourceSiteLocus.
   csv allBacteriaSpecies.csv sourceSite_bacteriaSpecies.csv
   ```

7. Now the bacteria specie's data for each source site (`sourceSite_bacteriaSpecies.csv`) needs to be mapped to its corresponding source site preferred name in `sourceSite_MetaMap_parsed.csv` using the `upstream_4_speciesPreferredMapping.jl` script. Run the script with the following the command:

   ```
   $ julia upstream_4_speciesPreferredMapping.jl sourceSite_MetaMap_
   parsed.csv sourceSite_bacteriaSpecies.csv sourceSite_bacteriaSpecies_
   preferred.csv
   ```

After all of these steps are done, all of the files needed for the Downstream Process have been created.

3 Downstream Process

Before starting, please make sure that the `sourceSite_bacteriaSpecies_preferred.csv` and `cleaned_host_annotations.csv` files are prepared and updated. This process will join source site, bacteria species, and host species data together based on locus number and create a matrix of host and bacteria species data associated with a source site.

1. To map the data in `sourceSite_bacteriaSpecies_preferred.csv` to data in `cleaned_host_annotations.csv`, use the `downstream_1_createHostBacteriaSpecieCSV.jl` script.

Run the script with the following command:

```
$ julia downstream_1_createHostBacteriaSpecieCSV.jl cleaned_
host_annotations.csv sourceSite_bacteriaSpecies_preferred.csv
sourceSite_Host_Bacteria_.csv
```

Please note this step only needs to run once if `sourceSite_Host_Bacteria_.csv` does not already exist.

2. To create the NEXUS file with the matrix of host and bacteria species pertaining to a specific source site, run the following command:

```
$ julia downstream_2_createMatrix.jl [specific source site]
sourceSite_Host_Bacteria_.csc specificSourceSite_sourceSite_
matrix.nex
```

Note that if `[specific source site]` is longer than one word, encapsulate the term with singles quotes.

4 Using PAUP* and Dendroscope

PAUP* can be used to create phylogentic trees for analysis. The software can be downloaded at http://phylosolutions.com/paup-test/. Below are the steps to run PAUP*

1. Launch PAUP* and set the working directory to the file where the NEXUS files are located using the `cd` command.
2. To run a NEXUS file, run the command:

```
execute specificSourceSite_sourceSite_matrix.nex
```

3. PAUP* uses two methods to search for optimal trees: exact and heuristic. This guide will use the heuristic method. For further information on the methods, visit http://phylosolutions.com/paup-documentation/paupmanual.pdf

   To perform a heuristic serach, run the command:

```
hsearch
```

4. To save this optimal tree, run the command:

```
savetree file=specificSourceSite_sourceSite.tre
```

5. The tree can be viewed in PAUP* with the command **showtrees** or the software Dendroscope, which can be downloaded at https://software-ab.informatik.uni-tuebingen.de/download/dendroscope3/welcome.html