WORKFLOW FOR GENERATING A QIIME-COMPATIBLE BLAST DATABASE FROM AN ENTREZ SEARCH

CHRIS BAKER*

This workflow allows you to take an input FASTA file generated from the NCBI database (e.g. through an Entrez search) and generate the files that you need to BLAST your data against those sequences through QIIME. These and other output files may also be useful for programs such as MEGAN.

1 NCBI taxonomy database files

If you do not already have a local copy of the NCBI's taxonomy data, you will need to download it. On Odyssey, you can do it with the following commands:

```
wget ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/taxdump.tar.gz
tar -zxvf taxdump.tar.gz

wget ftp://ftp.ncbi.nih.gov/pub/taxonomy/gi_taxid_nucl.dmp.gz
gunzip gi_taxid_nucl.dmp.gz
```

The files you will need are called `nodes.dmp`, `names.dmp`, `merged.dmp`, `delnodes.dmp` and `gi_taxid_nucl.dmp`.

Every sequence in GenBank is identified by a unique 'GenBank ID' or GI. Each GI is associated with a TaxonID number, indicating the organism that the sequence comes from. These associations are listed in `gi_taxid_nucl.dmp` for the nucleotide sequence database. (There are equivalent files for the other sequence databases, and you should download those instead if you intend to blast against sequences from one of those other databases.) Every node in the taxonomy database's tree-of-life has a unique TaxonID, but several sequences may share the same TaxonID e.g. if they come from the same species. To describe the topology of the tree, `nodes.dmp` lists every TaxonID and, for each one, gives the TaxonID for the parent node. Finally, `names.dmp` provides the taxonomic name for each node in the tree.

2 ENTREZ query

Go to `http://www.ncbi.nlm.nih.gov/sites/gquery` and execute a search to obtain the sequences you ultimately want to BLAST against. For example, you might search for

```
aftol AND "internal transcribed spacer 1"
```

which (as at 1 June 2012) finds 710 Nucleotide results. Click on `Nucleotide` to show these results. Then, at the top of the screen, click `Send to:`. Select `File`, then format `FASTA`, then `Create File`. The file should download as something like `sequence.fasta`.

The FASTA file that Entrez creates has deflines of the form

```
>gi|######|...
```

* ccmbaker@fas.harvard.edu. Pierce Lab, Department of Organismic and Evolutionary Biology, Harvard University.

where `######` is the unique GenBank ID (GI) for each sequence. (In fact, any FASTA file that fits this description can be used – it need not be created by Entrez.) We will later modify these deflines to just read `>######` in order to work with QIIME, as described in step 3 below.

3 Run entrez_qiime.py

The Python script `entrez_qiime.py` takes the FASTA file and the three taxonomy files described above as inputs. From these it generates the following output files:

- a copy of the FASTA file with the deflines stripped down to just the GI number;
- a list of GIs in the FASTA file and their corresponding taxonomic names as semicolon delimited lists (this is the taxonomy mapping file that you need for QIIME);
- a list of GIs in the FASTA file with their corresponding TaxonIDs;
- a list of GIs in the FASTA file (may be useful for BLAST+ tools, if required);
- a list of TaxonIDs for sequences in the FASTA file, with their corresponding taxonomic names as semicolon delimited lists (only if requested with the `-t` option).

On Odyssey, you can run the script as follow:

```
module load hpc/python-2.7.1
module load hpc/numpy-1.6.0_python-2.7.1
module load bio/pycogent-1.5.1
python entrez_qiime.py [options]
```

where the options are the following:

`-i`, `--inputfasta` The path, including filename, of the input FASTA file. Filename should have an extension, as it will be used as the basis for output file names. [Required]

`-o`, `--outputdir` The directory where output files should be saved. Will be created if it does not exist. [Default: ./]

`-f`, `--force` If -f or –force option are passed, output files will overwrite any existing files with the same names. Without this option, output filenames will be modified by the addition of a date/time string in the event that a file with the same name already exists. [Default: False]

`-n`, `--nodes` The directory where the files nodes.dmp, names.dmp, merged.dmp and delnodes.dmp are located. The files are typically downloaded from the NCBI in the compressed archive taxdump.tar.gz. [Default: ./]

`-g`, `--gitaxid` The path, including filename, of the (uncompressed) input gi-taxid file e.g. from the nt database dump, typically downloaded from the NCBI as gi_taxid_nucl.dmp.gz and uncompressed to gi_taxid_nucl.dmp. [Default: ./gi_taxid_nucl.dmp]

`-r`, `--ranks` A comma-separated list (no spaces) of the taxonomic ranks you want to keep in the taxon names output. Ranks can be provided in any order, but names output will be generated in that order, so should typically be in descending order. Any taxonomic names assigned to one of the ranks in the list will be retained; other names will be discarded. If a taxon has no name for one of the ranks in the list, then NA will be inserted in the output. No sequences or taxa will be discarded – this option only affects the names that are output for each taxon or sequence.The following ranks are available in the NCBI taxonomy database as at 2 June 2012 (in alphabetical order): class, family, forma, genus, infraclass, infraorder, kingdom, no_rank, order, parvorder, phylum, species, species_group, species_subgroup, subclass, subfamily, subgenus, subkingdom, suborder, subphylum, subspecies, subtribe, superclass, superfamily, superkingdom, superorder, superphylum, tribe, varietas. Note the use of underscores in three of the names. You should include these on the command line but they will be replaced by spaces when the script runs so that the names match the NCBI ranks. [Default: phylum,class,order,family,genus,species]

`-t`, `--taxid` If -t or –taxid are passed, an additional output file will be generated that lists taxonomy for each TaxonID. [Default: False]

`-h`, `--help` Print this help information.

The script automatically generates output filenames. Given the input FASTA file `somepath/file.extn`, the script will save files `file_stripped.extn`, `file_gi_taxonomy.txt`, `file_gi_taxid.txt` and `file_gi.txt` in the output directory passed to `-o`. It will also generate `file_taxid_taxonomy.txt` in the same directory if the option `-t` is passed. If these filenames conflict with any existing files, then those existing files will be overwritten if `-f` is passed, or the output filenames will be modified by the addition of a date-time string otherwise.

Sometimes the script may encounter small discrepancies between the various input files. For example, your input FASTA file may contain GIs with outdated TaxonIDs in `gi_taxid_nucl.dmp` if two taxa have been merged. Or your FASTA file may contain sequences that have only recently been uploaded to GenBank and have not yet made it into `gi_taxid_nucl.dmp`. `entrez_qiime.py` can deal with some of these discrepancies, and any changes that are made are documented in a log file saved in the output directory alongside the other outputs.

The script will take several minutes to run, even with a small input FASTA file, since it takes some time to load the NCBI's taxonomy data. Running time increases with the size of the input FASTA file.

4 Generate BLAST database

Now we want to generate a BLAST-formatted database from the FASTA file with the stripped-down deflines, i.e. the output from `entrez_qiime.py`, not the version originally downloaded from Entrez. (Actually, you might be able to skip this step if you are just using QIIME – see step 5 below.)

If the FASTA file output by `entrez_qiime.py` is saved as `somepath/sequence_stripped.fasta`, you can generate the BLAST database on Odyssey with the following code:

```
module load bio/ncbi-blast-2.2.25+

makeblastdb \
-in somepath/sequence_stripped.fasta \
-dbtype nucl \
-title 'a title for your database' \
-out yourblastdbpath/yourblastdbname \
-max_file_sz '1GB'
```

This will save a handful of files called `yourblastdbpath/yourblastdbname.???` that together comprise the sequences in your FASTA file, but formatted appropriately for BLAST. You could now BLAST against that database using any of the usual BLAST command line tools, such as `blastall`. You can give the database any title you want, and it may be useful to be fairly descriptive. The name `yourblastdbname` is used in naming the output files, and will also be used later to refer other programs (like QIIME or `blastall`) to your database, just like you might use `nt` or `nr` to refer to the entire nucleotide database, so it is probably useful to keep that name short.

5 Run BLAST search in QIIME

You should now have all the files you need to BLAST against the sequences you obtained through your Entrez search using QIIME. To ensure that QIIME can find your new BLAST database, you can use some code like this on Odyssey:

```
module load bio/qiime-1.5.0

BLASTDB_OLD=$BLASTDB
BLASTDB=yourblastdbpath/

assign_taxonomy.py \
-i your454data.fasta \
-t outputdir/sequence_gi_taxonomy.txt \
-m blast \
-b yourblastdbname

BLASTDB=$BLASTDB_OLD
unset BLASTDB_OLD
```

The file `outputdir/sequence_gi_taxonomy.txt` is one of the output files from `entrez_qiime.py`, and will be located in the output directory that you specified.

You may be able to skip step 4 above if you replace the option `-b yourblastdbname` with the option `-r somepath/sequence_stripped.fasta`, thus providing the output FASTA file directly to QIIME. In this case, you should not need to change the environmental variables `$BLASTDB_OLD` and `$BLASTDB`. QIIME should format the BLAST database for you and go on to execute the BLAST search as usual.