Starter cookbook to processing genome w/ annotation pacakges for NCBI

Katherine Amrine

09 February 2015

Contents

Alrighty, here is my step-by-step and explanation for what I did to do a WGS annotated genome submission for NCBI

1 Notify NCBI you will be submitting data

1.1 create a bioproject submission per genome.

Specify that you will be submitting WGS, Annotated, SRA, etc. I believe here, you will use an identifier to identify your genome that will act as the base of your locus tags. For Powdery mildew, it was automatically assigned as "EV44." Pick something meaningful if you don't want one automatically assigned. When you get down to the `newIDs.new.ncbi.txt` file I ask for in Step3, this prefix is what needs to go in this file.

1.2 Next create a biosample submission

include details on your sample

1.3 Next start the WGS submission process

also, they will ask you for the type of evidence that you have for the strings of N's within sequences. First off, N's do NOT occur as singletons, so they are not ambiguous bases with CLC. Ns refer to gaps of the size of the string of Ns if they ask you, but if you do have Ns as ambiguous bases, you need to specify this.

1.3.1 WHAT YOU WILL NEED

-fasta file with scaffolds with concise names this means that you really need to slim down names to as simple of identifiers as possible.

-gff version 2 file (with the scripts I provide here) devoid of sequences you don't need version 2, but I used it and wrote my scripts to use this type of file. The GFF file gene names should match your gene names in your fasta file.

-install of BIOPERL and FASTools

1.3.2 WHAT I GIVE YOU

- a crappy script to do the first conversion of the gff file to the tbl
- file required for the tbl2asn script that will ultimately make your asn submission file
- another crappy script to substitute in the names necessary

2 Pre-process the peptide files

2.1 isolate the peptide sequences with internal stops (check number of seqs)

needs: peptide fasta file

returns: fasta file with internal stop sequences

```
fasgrep -s "[^A-Z].+" <peptide_file.fas> > <file_with_internal_stop_sequences.fas>
```

2.2 remove peptide sequences with internal stops (confirm you are missing seqs)

needs: peptide fasta file

returns: fasta file without the internal stop sequences

```
fasgrep -vs "[^A-Z].+" <peptide_file.fas> > <peptiled_file_minus_internal_stops.fas>
```

2.3 remove annotated genes with internal stops from gff file.

needs: peptide fasta file without the internal stop sequences; gff file

returns: gff file without genes with internal stops

the second command (cut -c 2-) will get everything but the ">" character, and the third command (cut -f1,2 -d".") is specific to the e-necator processing. Our gene names consisted of the species, then gene name separated by periods. There was a third field with ".t1" which specifies only the transcript in our gff files. We remove this because it would only pick up the mRNA line in the gff file. You need to see what is unique about your gene names and modify this. The fourth command (perl code) adds a regex end-of-word wildcard. This only works if your gene names in your fasta peptide file match the gene names in your gff file. You should check this. (confirm correct unique ids are removed after this step)

```
egrep -f <(egrep ">" <peptide_file_minus_internal_stops.fas> | cut -c 2- |cut -f1,2 -d'
```

2.4 make sure there are no sequences in your gff file. It should only be coordinates for the following scripts.

3 Fix the scaffolds files

3.1 remove trailing Ns (no one cares where these are, just truncate them)

needs: original scaffold file

returns: scaffold file without trailing ends

```
fassub -s 'N{1,}$' '' <scaffolds_file_original.fas> > <scaffolds_no_trailing.fas>
```

3.2 find sequences with leading n's and their numbers, put into file

needs: scaffold file from previous step

returns: text file with list of sequences that have trailing N's, and their number

```
fasgrep -s "^N" <scaffolds_no_trailing.fas> | perl -ne 'BEGIN{$start=0;}if(/^\>/){$fini
```

3.3 change coordinates for leading N sequences in gff file

needs: file specifying leading N's

returns: gff file adjusting coordinats for genes on scaffolds where leading Ns were removed

for this script *changecoords.pl*, you need the file `leading_Ns.txt` but it takes the gff file and fixes all the coordinates

```
./change_coords.pl <new_gff_minus_internal_stop_peptides.gff> > <new_gff_with_no_errors
```

3.4 and then remove the leading Ns from the sequences

needs: scaffold file edited from aboce

returns: scaffold file now with removed leading Ns

```
fassub -s '^N{1,}' '' <scaffolds_no_trailing.fas> <scaffolds_no_trailing_leading.fas>
```

3.5 convert files to NCBI-ready with my messy table scripts

needs: processed gff file from previous steps, and a conversion file with the old gene names, and the new gene names that are NCBI-specific

returns: NCBI tbl file with NCBI-specific identifiers for proteins

the first script takes the gff as is, and puts the information into table formt

```
~/bin/katie_gff2tbl.pl <new_gff_with_no_errors.gff> > <genome_annot_from_gff.tbl-sub>
```

the second script takes a file called `gene_name_conversions.txt` and substitutes the correct annotations into the tbl file from the previous command. check *c-strain-annotation.sub.txt* for an example of the format of this file, which is tab-delimited and expects a header line. the subsequent command just changes gff identifiers to NCBI identifiers

```
/bin/katie_gff2tble2.pl <genome_annot_from_gff.tbl-sub> > <genome_annot_from_1gff.tbl>
perl -pe 's/gene_id/protein_id/' <genome_annot_from_1gff.tbl> > <genome_annot_from_2gf
```

3.6 change the gene names to ones that are permissible by NCBI.

needs: .tbl file from previous step; file to annotate the specific identifier assigned by NCBI. `newIDs.new.ncbi.txt` look at *newIDs.new.txt* for an example on how to do it. remember, the prefix that you entered in the WGS

submission will go at the beginning of your new gene IDs. In the example file, I have the number followed by a dash. USE AN UNDERSCORE, NOT A DASH.

returns: ncbi-formatted tbl file

```
~/bin/change_IDs.pl <genome_annot_from_2gff.tbl> > <final.tbl>
```

3.7 miscellaneous help

these were to clean up things listed as problematic in the discrepancy report this example shows deleting weird names that muck up the translator like "AF443189_2" and "homolog" and "af376000_1" in the .tbl annotations

```
perl -pe 's/\"//g;s/AF443189_2//g;s/, putative//g;s/, //g;s/homolog$/-like protein/g;s/
```

4 Prepare files to submit

4.1 make a .sbt file online

Enec.sbt came from the NCBI website creating an sbt file for me at http://www.ncbi.nlm.nih.gov/WebSu

4.2 get tbl2asn script from NCBI

Need to go download mac.tbl2asn or whatever it is called

4.3 run most recent version of tbl2asn. change things that are different, like paired-ends, organism name, isolate name, check your options, -a has to do with gap sizes, etc.

this script looks for a .tbl file in your current directory

```
./mac.tbl2asn -i <> -t Enec.sbt -M n -a r10k -j"[organism=Erysiphe necator][isolate=c]"
```

4.4 check errorsummary.val for fatal errors

5 submit .sqn and .fas file

6 Fix the errors they find

6.1 adapter contamination

You will likely have some adapter contamination they find with your submission. To fix this, Copy the lines from the email NCBI sends out into a file named *adapters_to_remove.txt* (check this file for an example of the input). The next script is still in very raw form. You want to edit the lines that specify your input files and your output files within the script. The standard output of this script is the scaffolds in fasta format. A new gff file will be created within the script. Bioperl is required.

```
./remove_NCBI_adaptors.pl > <new_trimmed_scaffolds.fas>
```