NGS 2014 Tutorial - Databases Documentation

Release 1.0

Adina Howe

Nov 02, 2020

Contents

Contents:

CHAPTER 1

So you want to start using that big data in NCBI?

CHAPTER 2

Learning objectives

This is a tutorials for working with the data that is available in NCBI. The learning objectives for this tutorial are as follows:

1. To be able to download specific gene sequences or genomes from NCBI (even with a big list of gene sequences).
2. To be able to create use these genes as a database to annotate a sequencing dataset.
3. To estimate the number of genes and their corresponding annotations in multiple sequencing datasets.

You will need to know some things prior to this tutorial:

1. Familiarity with the structure of NCBI website and their nucleotide and genome databases.
2. Ability to navigate in the unix shell.
3. Ability to execute programs in the shell.
4. Access and login to an Amazon EC2 instance or similar ubuntu-based server

The key challenge that we will work through…or your mission, if you choose to accept it, is to identify nitrogen fixation genes found in sequencing DNA from soils.

You have been delivered three dogma-changing metagenomes (sequencing datasets) originating from three different Iowa crop soils (corn, soybean, and prairie). You are interesting in identifying nitrogen fixation genes that are associated with native bacteria in these soils. Nitrogen fixation is a natural process performed by bacteria that converts nitrogen in the atmosphere into a form that is usable for plants. If we can optimize natural nitrogen fixation, our hope is to reduce nitrogen fertilizer inputs that may contribute to the eutrophication of downstream waters (e.g., dead zones in the Gulf of Mexico).

CHAPTER 3

Your tools

Get your EC2 instance going - you need an Ubuntu 14.0 based instance (64-bit) with at least 4 cores. This tutorial was run on 6/26/2018 on ami-a4dc46db.

Then install the software:

```
sudo bash
apt-get update
```

Then. . . :

```
cd /root
apt-get -y install gcc git screen curl make python-pip
```

And finally. . . :

```
pip install screed
curl -O ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/ncbi-blast-2.7.1+-
↪x64-linux.tar.gz
tar xvf ncbi-blast-2.7.1+-x64-linux.tar.gz
cp ncbi-blast-2.7.1+/bin/* /usr/local/bin/
cd /home/ubuntu
```

CHAPTER 4

Getting the Data

CHAPTER 5

Group 1

5.1 Task 1 - Get the data (your sequences)

Get the metagenome datasets and scripts related to this tutorial.

All the tutorial materials are contained on a Github repository. The reason for using Github is that this material can be updated by me and grabbed by you lucky folk seamlessly with just a couple commands. If you are interested in learning more about Git, see these [tutorials]():

```
git clone https://github.com/germs-lab/ncbi-tutorial.git
```

This command will make a directory (or folder for those more Finder/Explorer inclined) named "ncbi-tutorial" in the location where it was run. Within that directory, there will be two directories containing "data" and "scripts". You can see this by navigating (hint: cd) to the "ncbi-tutorial" directory and typing:

```
cd ncbi-tutorial
ls -lah
```

5.2 Task 2 - Explore the data

Navigate to the data/metags directory and identify the number of sequences in each file. Hint: To find specific characters in a file, you can use [grep](http://www.gnu.org/software/grep/manual/html_node/Usage.html). For example, to find all instances of AGTC in the corn.fa file, we could:

```
cd data
cd metags
ls
grep AGTC corn.fa
```

To find sequences, we know that each sequence will start with a special character, ">". This character in the shell, remember, is a bit special. So to find it as a symbol in the text, we're going to put a '^' right before it in quotes:

```
grep ^">" corn.fa
```

Now, to count, you'll remember we can use the command "wc", with a pipe... So you're command will look something like this:

```
grep ^">" corn.fa | wc
```

Or... if you want to do this quicky:

```
for x in *fa; do echo $x; grep ^">" $x | wc; done
```

To identify nitrogen fixation genes, you've been tasked to build a database of all previously observed known nitrogen fixation genes (nifH). To build this database, you have been reading literature for about two weeks and come up with a list of about 50 genes:

gi|985477984|emb|LN997366.1|

gi|985477986|emb|LN997367.1|

gi|985477988|emb|LN997368.1|

gi|985477990|emb|LN997369.1|

...

gi|38679|emb|X51500.1|

gi|470075|emb|Z31716.1|

5.3 Task 3 - Look at your potential reference genes

Check out the file containing these gene IDs, its in the data directory - nifh-ref.fa.

You have a sinking feeling like this isn't really leveraging the big data biology that everyone says sequencing technologies have provided. You've decided to check out NCBI for its contents.

CHAPTER 6

Group 2

6.1 Task 4

Go to the NCBI webpage and identify an estimate of total nifH genes and download a list of their accession numbers.

You'll want to navigate in a web-browser to the http://www.ncbi.nlm.nih.gov/. You'll see in the search query box that you can search a number of databases. Here, we want to look at the nucleotide database and query something along the lines of nifH or nitrogen fixation.

When I did this, there were hundreds of genes that were hit by this query. Click on the "Genes", and you will want to look for the "Send to" link at the upper right of the page (put on a magnifying glass!), and explore what you can download for this gene query. For this tutorial, let's download the NCBI ID numbers associated with your genes as a list of numbers. [Send –> Complete Record –> File –> Format –> Accession List]. Find the file you downloaded on your computer and give it a peek. Note that you can also download the FASTA sequences of the genes directly with this method – can you imagine a situation where this wouldn't be as efficient?

6.2 Task 5

To make this tutorial not-as-painful to complete in a reasonable amount of time, I've also made a list of 300 nifH genes from NCBI and put them in a file '300-nifh-genes.txt' in the data directory. *I would highly suggest you use this gene to build your database going forward in this tutorial,* although you could build your own and work through this tutorial.

6.3 Task 6

Take a look at this file. Prove to yourself that it contains 300 genes (Hint: wc)

CHAPTER 7

Group 3

Now, we are going to learn how to download these genes (by learning about the NCBI API below)

7.1 Task 7

Think about how you would download the data from your accession list of genes if you didn't have this tutorial.

You may have thought about some of the following:

1. Go to the web portal and look up each FASTA
2. Go to the FTP site, find each genome, and download manually and parse out the genes.
3. Use the NCBI Web Services API to download the data

Among these, I'm going to assume many of you are familiar with the first two. This tutorial then is going to focus on using APIs.

CHAPTER 8

Scaling "Getting the Data" On Up

Here's some answers, among which my favorite is "an interface through which you access someone else's code or through which someone else's code accesses yours – in effect the public methods and properties."

The NCBI has a whole toolkit which they call *Entrez Programming Utilities* or *eutils* for short. You can read all about it in the documentation. There are a lot of things you can do to interface with all things NCBI, including publications, etc., but I am going to focus today on downloading sequencing data.

To do this, you're going to be using one tool in *eutils*, called *efetch*. There is a whole chapter devoted to efetch – when I first started doing this kind of work, this documentation always broke my heart. Its easier for me to just show you how to use it.

8.1 Task 8

Open a web browser, and check out what NCBI knows about a gene. Check it out here.

8.2 Task 9

Download the gene with eutils commands in your web-browser and take a look at the file.

On your web-browser, paste the following URL to download the nucleotide genome for gene X51500.1:

```
https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nuccore&id=X51500.1&
↪rettype=fasta&retmode=text
```

8.3 Task 10

Try downloading the GenBank file instead by pasting this onto your web-browser:

```
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nuccore&id=CP000962&
↪rettype=gb&retmode=text
```

Do you notice the difference in these two commands? Let's breakdown the command here:

1. <http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?> This is command telling your computer program (or your browser) to talk to the NCBI API tool efetch.
2. <db=nuccore> This command tells the NCBI API that you'd like it to look in this particular database for some data. Other databases that the NCBI has available can be found here.
3. <id=X51500.1> This command tells the NCBI API efetch the ID of the gene/genome you want to find.
4. <rettype=gb&retmode=text> These two commands tells the NCBI how the data is returned. You'll note that in the two examples above this command varied slightly. In the first, we asked for only the FASTA sequence, while in the second, we asked for the Genbank file. Here's some elusive documentation on where to find these "return" objects.

Also, a useful command is also <version=1>. There are different versions of sequences and some times that is useful. For reproducibility, I try to specify versions in my queries, see these comments.

Note: Notice the "&" that comes between each of these little commands, it is necessary and important.

Ok, let's think of automating this sort of query. So... we're moving from your lil laptop to your jumbo EC2 instance now.

8.4 Task 11

Download a gene sequence on the command line.

Going back onto your instance, in the shell, you could run the same commands above with the addition of *curl* on your EC2 instance:

```
curl "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nuccore&id=X51500.
↪1&rettype=fasta&retmode=text"
```

You'll see it fly on to your screen. Don't panic - you can save it to a file and make it more useful BUT note the path you are in and where you will save this file (as long as you know... that's fine):

```
cd /home/ubuntu/ncbi-tutorial/data

curl "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nuccore&id=X51500.
↪1&rettype=fasta&retmode=text" > X51500.1.fa
```

You could now imagine writing a program where you made a list of IDs you want to download and put it in a for loop, *curling* each genome and saving it to a file. The following is a script. Thanks to Jordan Fish who gave me the original version of this script before I even knew how and made it easy to use.

To see the documentation for this script in the scripts directory:

```
python ../scripts/fetch-genomes.py
```

You'll see that you need to provide a list of IDs and a directory where you want to save the downloaded files.

8.5 Task 12a

Note: If you are nervous....you may want to run this on just a few of these IDs to begin with. You can create a smaller list using the *head* command with the -n parameter in the shell. For example, head -n 3 300-nifh-genes.txt > 3genes.txt. Please note that you will have to think about WHERE to run this script. This command, for example, is written to be run in the /home/ubuntu/ncbi-tutorial

directory.

Run this script (note that your paths for the script or data may need to be specified) – also see note below:

```
cd /home/ubuntu/ncbi-tutorial/data
python ../scripts/fetch-genomes.py 300-nifh-genes.txt nifh-database-fastas
```

Sit back and think of the glory that is happening on your screen right now...

8.6 Task 12b

After all the 300 genes are downloaded, you will want to concatenate them into one file, named "all-nifH.fa":

```
cd nifh-database-fastas
cat *fa > all-nifH.fa
```

8.7 Task 13

Look at the script/program content in "fetch-genomes.py".

The meat of this script uses the following code:

```
url_template = "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?
↪db=nucleotide&id=%s&rettype=fasta&retmode=text"
```

You'll see that the *id* here is a string character which is obtained from list of IDs contained in a separate file. The rest of the script manages where the files are being placed and what they are named. It also prints some output to the screen so you know its running.

8.8 Task 14

Take a break. Put up your pink stickie if you need help with this.

CHAPTER 9

Group 4

So there are lots of ways to do this and arguably "blasting" is one of the most common. If you're already familiar with how to run BLAST, you can just tackle this next task. Else, instructions from Titus Brown's NGS course are below.

9.1 Task 15

Format your nifH database for BLAST and then perform an alignment (with BLAST) of your metagenomes against your new database:

```
makeblastdb -in <nifh-db> -dbtype nucl -out <nifh-db>

cd /home/ubuntu/ncbi-tutorial/data
makeblastdb -in nifh-database-fastas/all-nifH.fa -dbtype nucl -out nifh-database-
↪fastas/all-nifH-db.fa
```

You should also add your lit search genes (those 30 genes) into your NCBI downloaded genes if you're up to the challenge – it takes a couple steps. Now, run your blast.

I would suggest something like the following variables, note that the outfmt flag identifies a tabular output:

```
blastn -query <metag file> -db <db file> -out <name of output> -outfmt 6

cd /home/ubuntu/ncbi-tutorial/data

blastn -query metags/corn.fa -db nifh-database-fastas/all-nifH-db.fa -out corn-blast-
↪output.txt -outfmt 6
```

This takes a bit of time... so stretch a bit. Like all good bioformaticians, you should go do something fun while the computer does all the work. If you want to see it running, try using the "top" command.

Note: BLAST initiation or refresher tutorial.

Make sure you've created a blast output for each of the metagenomes in the tutorial.

> for x in metags/*fa; do blastn -query $x -db nifh-database-fastas/all-nifH-db.fa -out $x-blast-output.txt -outfmt 6; done

9.2 Task 16

Examine your blast outputs. What do the first few lines contain? How many hits do you have per metagenome to your database?:

```
cd metags
ls
```

CHAPTER 10

Next steps, what's next

Let's step back. What is the reason for annotating your genes? You want to get an idea of what genes are in each soil metagenome as well as a quantitative estimate of each gene. Eventually, you would likely use this information to do some statistical analyses – maybe in sophisticated packages like Qiime, Mothur, or Phyloseq. All these programs take similar inputs, and the big secret is knowing what they are and how to parse/move/shift/wrangel this information into the specific format needed for each program:

1. Metadata/Environmental data - What treatment does corn.fa represent – duh, corn! When was it sampled? What kind of other data have you collected on this sample (amount of fertilizer, etc).
2. Annotation information - What are the corresponding annotations to whatever shortcut ID associated with your genes (e.g., GI number). This file can also contain some ontology / hierarchical information (e.g., Taxonomy domains / phyla / species).
3. Abundance estimates - What are the estimates of each gene in each metagenome sample.

Of these, you should be able to make a metadata file for your experiment. The annotation file can be provided by your database.

You have a list of genes GI IDs and can pull the annotation of the gene from the header in the database nucleotide file OR better yet from the Genbank file. For example, you might pull out the lineage of each gene from the Genbank file and have taxonomy to provide for each gene.

Perhaps the most difficult is to get the abundance estimates. Intuitively, you can think about how to get this out of the BLAST outputs. I observed Gene X, Y times in dataset I, Z times in dataset II. Fortunately, I have a script to do this computationally.

10.1 Task 17

Produce an abundance table of all genes in your 3 metagenomes. You should call this command with "paths" in mind... where are your blast output files:

```
cd /home/ubuntu/ncbi-tutorial/data/metags
python ../../scripts/count-up.py *txt
```

This script produces a file, 'summary-count.tsv'. Can you read it?

10.2 Task 18

Take a look at 'summary-count.tsv'. What's in it?

10.3 Task 19

Often, I get really excited and need to know what gene is associated with each Gene ID. I like to pull in my annotations. Let's add the annotations by giving this a try:

```
python import-ann.py <blast database fasta file> summary-count.tsv > <output file>

python ../../scripts/import-ann.py ../nifh-database-fastas/all-nifH.fa summary-count.
↪tsv > summary-count-ann.tsv
```

10.4 Task 20

Transfer the annotated file over to your laptop and open it in Excel.:

```
mv summary-count-ann.tsv ~/.
```

CHAPTER 11

Conclusion

You now have the foundation for having some sequencing data that you need to compare to any database. You should be able to generate the information needed to perform statistical analyses. Note, that you can do this for specific genes and also genomes. . . ! Now, go forth and conquer!

CHAPTER 12

Indices and tables

- genindex
- modindex
- search