Mitofy was developed to assist in the annotation of genes and tRNAs in seed plant mitochondrial genomes. The program was developed in collaboration with Stacia Wyman and Jeff Boore.

SETTING UP THE PROGRAM (Mac OS X)

There are two main parts to the *Mitofy* program.

I. When you execute the main script (`mitofy.pl`), the program calls NCBI-BLASTX to search your genome against databases of 41 different protein-coding genes known from seed plant mitochondrial genomes. It then calls BLASTN to search your genome against databases for 27 different tRNA and 3 rRNA genes. Finally, it calls tRNAscan-SE to search for organellar tRNAs *de novo*. Each of the raw output files is parsed along the way and converted into a corresponding HTML file.

II. You will open and view the HTML files summarizing the BLAST and tRNAscan output in a web browser (described below). You will then open an annotation form, enter your annotation data, and submit that data to be written to a Sequin-formatted table. All of these actions are controlled by a set of Perl and CGI scripts located in `'/Library/WebServer/CGI-Executables'`. To run these scripts requires that you set up your computer as a local WebServer. Although this can be done on machines running Windows, Linux, and Mac OS, the instructions provided here are specifically tailored for configuring the program to run using the Apache webserver software on a Mac.

1. Download mitofy.tgz from http://dogma.ccbb.utexas.edu/mitofy.tgz

2. Put mitofy.tgz in the directory you want to work from. Most of the following requires that you work in the Unix environment, which you can access on a Mac using either Terminal (Go > Utilities > Terminal) or X11 (Go > Utilities > X11). The following website provides a pretty good introduction to using Terminal: http://smokingapples.com/software/tutorials/mac-terminal-tips/

3. Open a Terminal window, and use the `'cd'` command to navigate to the directory that contains the tar file. Unpack it using the following command (henceforth `'$'` is the Terminal prompt; do not include this in your command):

```
$ tar -xzvf mitofy.tgz
```

4. Type `'ls -al'` and hit `'Enter'` to view the contents of this directory:
   - The `'annotate'` folder can be put anywhere.
   - The `'cgi_config_files'` contains Webserver configuration files, which we'll move later.
   - We will also move files in the `'cgi'` directory (below).

5. Navigate to the `'cgi'` directory, and copy its contents to /Library/WebServer/CGI-Executables with the following commands. You will need to have administrator privileges on your machine (on a Mac, this is set in the Users part of the System Preferences).

```
$ cd cgi/
$ sudo cp -r * /Library/WebServer/CGI-Executables
```

6. Navigate up one directory, and delete the original tape archive (mitofy.tgz):

```
$ cd ..
$ rm mitofy.tgz
```

7. Copy the provided 'htaccess' file to the CGI-Executables directory. This file prohibits outsiders from accessing your webserver directories. Following the copy, this will be a system file that begins with a ".", so it will be visible in Terminal but not in Finder.

```
$ sudo cp cgi_config_files/htaccess.txt /Library/WebServer/CGI-
Executables/.htaccess
```

8. Make a copy of your original Apache configuration file ('httpd.conf') file with the following command:

```
$ sudo cp /private/etc/apache2/httpd.conf
/private/etc/apache2/httpd.conf.original
```

   You've now archived your original Apache Webserver configuration file, and you can revert to it at any time.

9. Now copy over the provided Apache configuration file:

```
$ sudo cp cgi_config_files/httpd.conf /private/etc/apache2/httpd.conf
```

10. Type the following four commands to navigate to the CGI directory and set the appropriate file permissions:

```
$ cd /Library/Webserver/CGI-Executables
$ sudo chmod a+rx *.pl
$ sudo chmod a+rx *.cgi
$ sudo chmod a+rx .htaccess
$ sudo chmod a+rwx cgi_out
```

11. With the 'httpd.conf' and '.htaccess' files in place, you now need to restart Apache so that all of your specified preferences will be enabled. Do this by opening System Preferences, clicking 'Sharing', and checking 'Web Sharing'. If it was already checked, uncheck it and wait until it says "Web Sharing: Off", then check it again. It should now say "Web Sharing: On".

12. *Mitofy* uses NCBI-BLAST and tRNAscan-SE, the executables for which are included in the distribution. Please review the corresponding README files, disclaimers, and Terms of Use found in the following directories:

```
annotate/blast/doc
annotate/tRNAscan/doc
```

13. Navigate back to the original annotate directory with the `'cd'` command.

14. Ensure that tRNAscan-SE is working by typing the following command from within the `'annotate'` directory.

```
$ tRNAscan/tRNAscan-SE
```

Verify that you get this (truncated here) output:

```
tRNAscan-SE 1.23 (April 2002)

FATAL: No sequence file(s) specified.

Usage: tRNAscan-SE [-options] <FASTA file(s)>
```

15. Ensure that NCBI-BLAST is working by typing the following commands from within the `'annotate'` directory:

```
$ blast/blastx
$ blast/blastn
```

Verify that you get this output:

```
BLAST query/options error: Either a BLAST database or subject sequence(s)
must be specified
```

16. Ensure that *Mitofy* is working by typing the following command:

```
$ ./mitofy.pl
```

Verify that you get this output:

```
   usage:./mitofy.pl [options] genome.fasta projectID
         projectID = unique project name; all output files with have this prefix

   options
         --prot_emax - maximum BLAST expect value for protein genes (default: 1e-3)
         --prot_pmin - minimum percent identity for protein genes (default: 60)
         --rna_emax  - maximum BLAST expect value for RNA genes (default: 1e-3)
         --rna_pmin  - minimum percent identity for RNA genes (default: 70)
         --rna_mlen  - minimum length for RNA BLASTN matches (default: 30)
```

If it says command not found, be sure you typed the `'./'` at the beginning of the command. If it says permission denied, run the following command, then try running *Mitofy* again.

```
$ chmod u+x *.pl
```

RUNNING THE PROGRAM

1. To run the program, put the FASTA file of your genome in the annotate directory, open a Terminal, navigate to the annotate folder, type `'mitofy.pl'` and hit the ENTER key. Again, if it says "command not found", you will need to type `'./'` (no quotes) at the beginning of the command. This will show the program's usage (step 16 above). All you need to specify, ***in order***, is the FASTA file and a project name (no spaces in the project name). For example, typing,

   ```
   $ ./mitofy.pl legume.fasta Mazzy
   ```

   will annotate the legume.fasta file, and all output files will have the base name "Mazzy".

2. If you want to change the thresholds for which BLAST hits are shown, you can modify them using the options specified by the usage. For example, the following command will increase the stringency of the BLAST e-value cutoff for protein genes by decreasing it to $1e^{-6}$, and it will show tRNA and rRNA hits according to the default e-value ($1e^{-3}$) and percent identity (70%) cutoffs but require a minimum match length of 50 nt:

   ```
   $ ./mitofy.pl --prot_emax=1e-6 --rna_mlen=50 legume.fasta Mazzy
   ```

3. *Mitofy* will now run through the BLAST searches and run tRNAscan (this takes the longest, sometimes an hour or more for large genomes) and write all the output to `'annotate/blast_output/Mazzy'`.

4. It will accept a FASTA file with one or multiple sequences. If the following FASTA file is the input:

   ```
   >Contig1400
   ATCATG…
   >Contig1500
   AAATGG…
   >Contig1600
   AAATTC…
   ```

   then your sequence will be identified as "Contig1400_3" in the annotation windows. "Contig1400" is from the first word of the first FASTA header, and "3" is the number of sequences in the FASTA file. The sequences are concatenated, in order, and the concatenated sequence is then given to the annotation scripts. This might not be ideal in some cases. It might be better to annotate each FASTA file separately. In this case, break up your FASTA file into multiple individual FASTA files and run them in batch using, for example, the following BASH command:

```
for fi in *.fasta;do mitofy.pl $fi ${file%.fasta};done
```

This will annotate all files in the directory that have a ".fasta" suffix, and the project ID will be the filename prefix. For example, legume_contig1.fasta would have the project ID "legume_contig1".

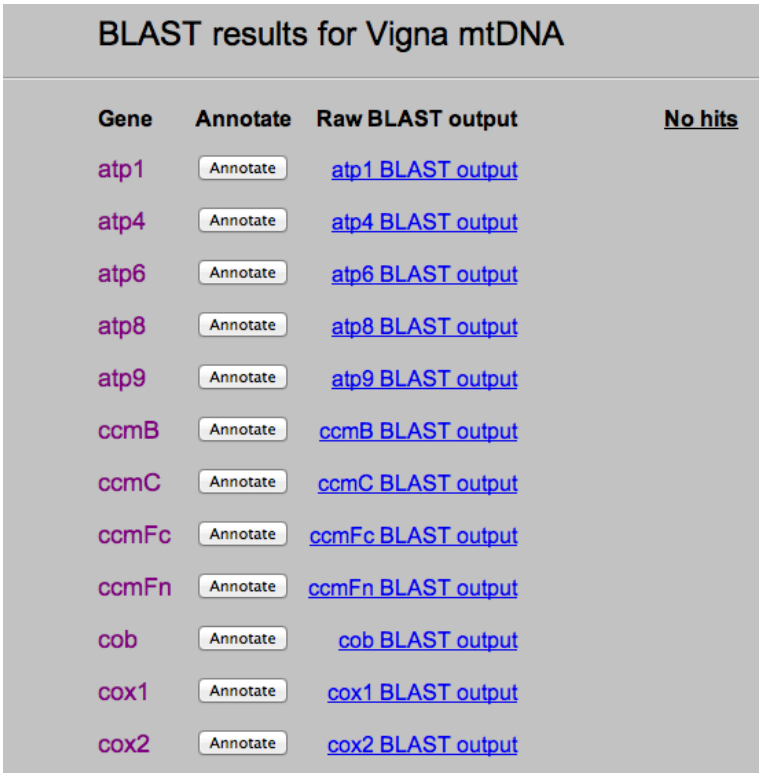
5. Continuing with the example we started at Step 1 of this section, after BLAST and tRNAscan-SE have run to completion, you will see the following message:

```
"Open 'blast_output/Mazzy_out/Mazzy_summary.html' and
'blast_output/Mazzy_out/Mazzy_rna_summary.html' in a web browser (preferably
Safari) to see results."
```

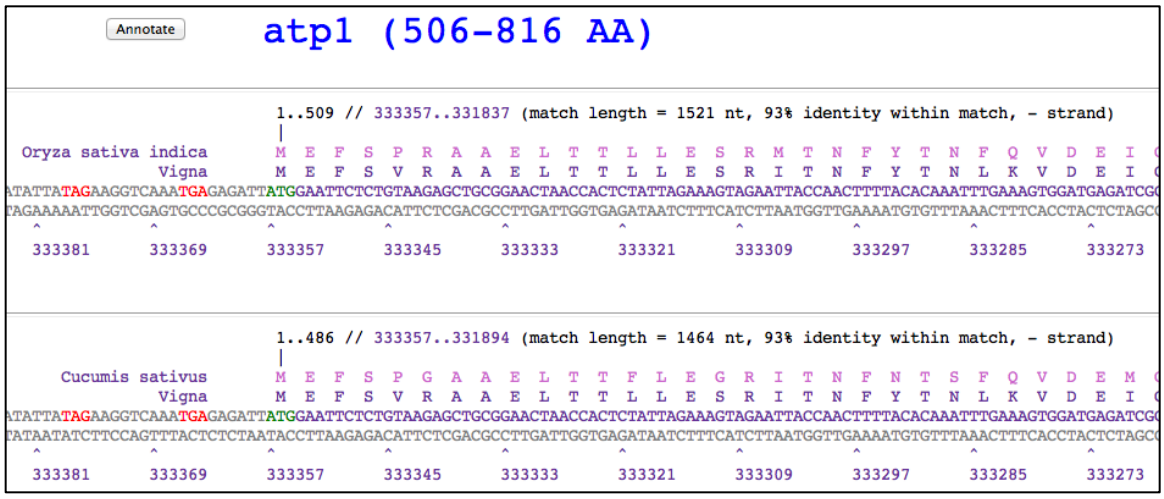
More generally, you should always begin by opening the following two files in a web browser. These files contain links to the raw and HTML-formatted output for all of the genes:

```
annotate/blast_output/project_name/project_name_summary.html
annotate/blast_output/project_name_rna_summary.html
```

Here is a partial screenshot of a gene summary page:



6. Click on the "Annotate" button to view the HTML-formatted BLAST output for that gene. You will get a screen that looks like this:

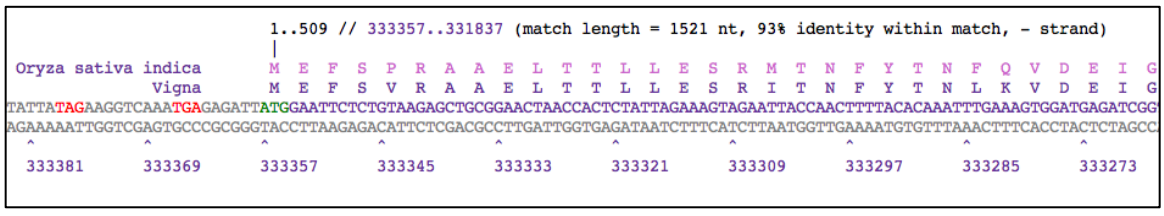


7. The header consists of the gene name and size range of that gene in the database. Each one of the entries, separated by line breaks, is a different BLAST hit. The top line of each hit shows some of the statistics for that hit. The first hit in the output immediately above has the following summary information:

```
1..509 // 333357..331837 (match length = 1521 nt, 93% identity within match,
- strand)
```

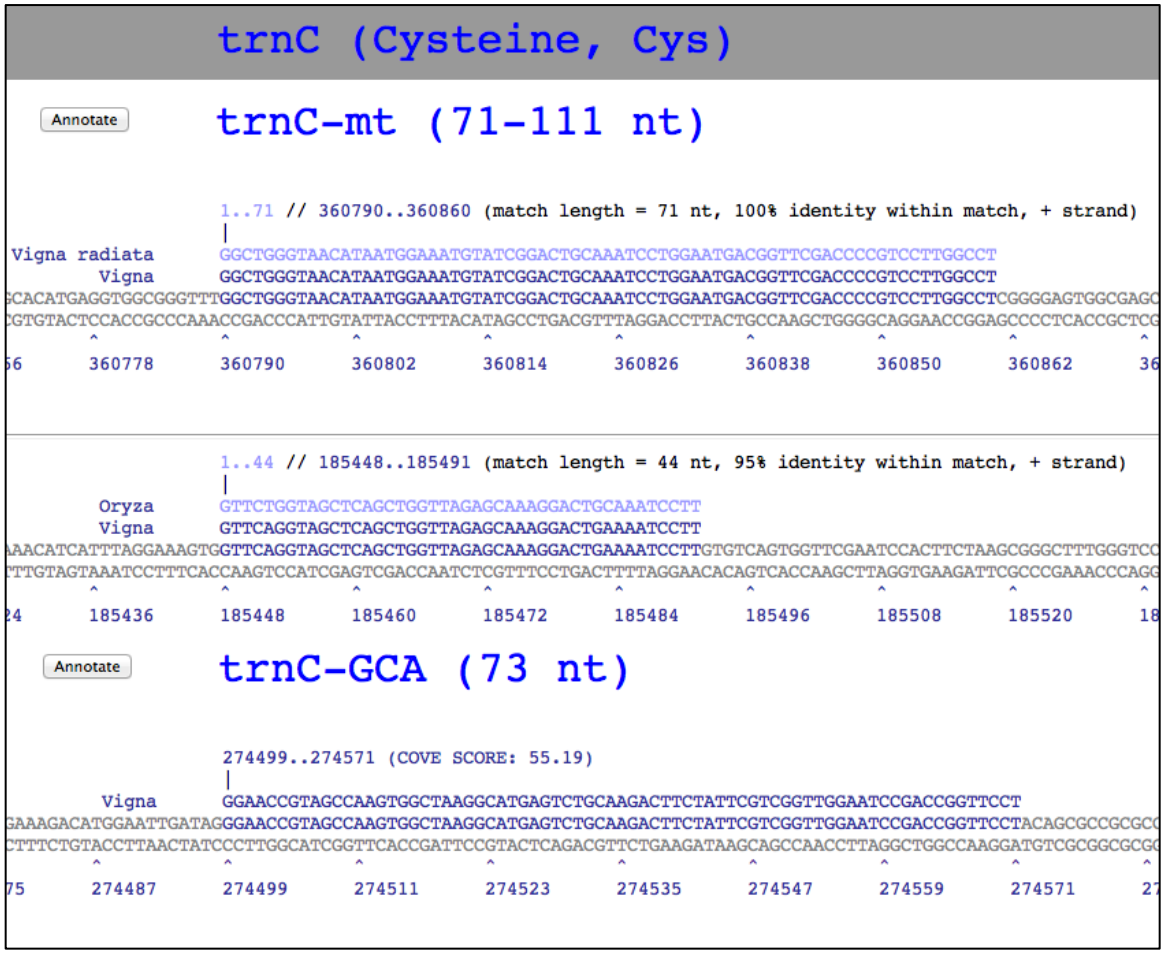
This indicates that amino acid coordinates 1–509 of the subject (database) sequence (from *Oryza* in this case) match coordinates 333357–331837 of the query genome (from *Vigna* in this case). The total match length is 1,521 nt, and 93% of the amino acids are identical within that match. Finally, this gene is located on the reverse ("–") strand, which should also be evident from the descending coordinates along the bottom of the hit.

8. Looking at the actual BLAST hit (immediately below), the amino acid sequence for the subject (*Oryza*) is shown first, followed by the conceptual translation of the query sequence (*Vigna*). The corresponding nucleotide (nt) sequence from your query is shown next, and it includes 60 nt up- and downstream of the BLAST match to help you locate potential start and stop codons. For protein-coding genes, potential start codons are shown in green, and stop codons are shown in red (genomic stop). C-to-U mRNA editing is common in plant mitochondria, and in some genes editing is required to create canonical start or stop codons Potential stop codons (CAA or CGA) are shown in yellow, and potential ACG start codons are likewise shown in green. Holding the cursor over a green, red, or yellow codon will give the start position of that codon in the genome. Corresponding genome coordinates for the query are listed below the nucleotide sequence.

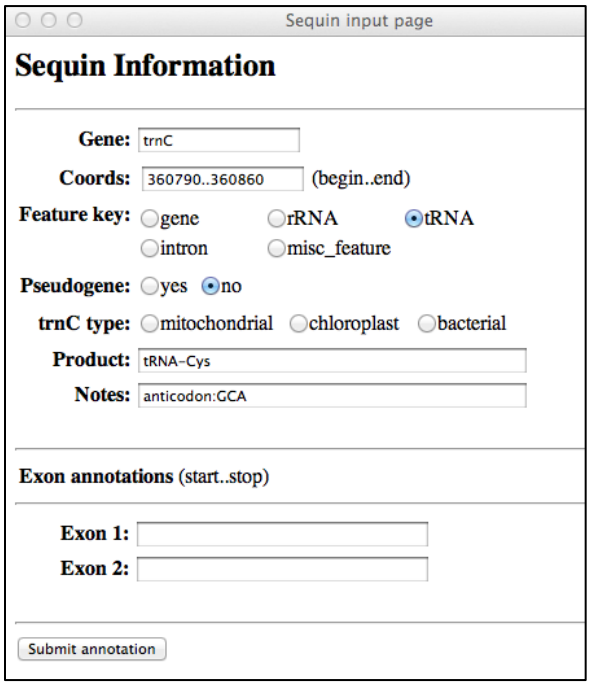


9. The output is slightly different for tRNA genes (immediately below). These pages have two panels, a top panel summarizing the BLASTN output and a bottom panel summarizing the tRNAscan-SE output. The advantage of the BLAST output is that the various flavors (e.g., plastid- vs. mitochondrial-type) of the different tRNAs are discriminated, giving you an indication of the ancestry of a particular tRNA. The advantage of the tRNAscan output is that it more precisely delineates the gene boundaries and identifies the anticodon (see below). In addition, the COVE score provides another measure of how well the match conforms to a canonical tRNA structure.

   I recommend using the BLAST output to discern the ancestry of the tRNA and the tRNAscan output for the final annotation. I also tend to base tRNA annotations on the tRNAscan output because the scripts will automatically fill in the genome coordinates and anticodon information in the annotation form if you click the `'Annotate'` button within the tRNAscan window. You should verify any questionable tRNAscan results by carefully checking the COVE scores. The structure predictions are also available in the raw tRNAscan output files, which are located in the output directory.



10. When you open a gene window, an annotation form should automatically pop up. If you've configured your web browser to block pop-up windows, you may need to re-enable them. Here is an example of an annotation form:



11. Enter the genome coordinates in the `'Coords'` box. The begin and end coordinates can be separated any character or characters except a number. I've followed the NCBI Sequin conventions, so if the gene is on the reverse ("–") strand, the begin coordinate will be a larger number than the end coordinate. See links below for more information on Sequin format. Click the appropriate radio buttons, and hit 'Submit annotation' to record your annotation.

12. The output is located here:

```
/Library/WebServer/CGI-Executables/cgi_out/project_name.tbl
```

13. Check the output file to make sure the CGI scripts are working and your annotation has been written out to this file. If the file wasn't created, or the annotation was not appended to the file, it will most likely reflect either permissions problems or problems in your Apache configuration. The Apache access and error logs will help you diagnose the problem. They are located here:

```
/var/log/apache2/access_log
/var/log/apache2/error_log
```

You can view the access log by typing the following command:

```
tail /var/log/apache2/access_log
```

You can view the error log by typing the following command:

```
tail /var/log/apache2/error_log
```

OTHER COMMENTS

- Familiarize yourself with the NCBI Sequin table format before getting started because many of the conventions in these programs and their output follow the Sequin format. See the following websites:
    - http://www.ncbi.nlm.nih.gov/Sequin/table.html
    - http://www.ncbi.nlm.nih.gov/Sequin/modifiers.html

- These scripts assume that the input sequence has a circular topology, i.e., that the last and first nucleotides in the sequence are adjacent, so features can actually overlap this boundary. This only shows up in the display of sequences up- or downstream of a BLAST hit that happens to fall at the beginning or end of the input genome. If you have an intact feature that is split in this way, you should consider re-orienting the input sequence so that important features are not split.

- You may want to refer to the GenBank files for the cucurbit genomes I've annotated (GQ856147, GQ856148, and NC_016005) if you have questions about exon/intron boundaries because all of them were verified with cDNA sequencing for those genomes. There are also notes about the handful of unusual tRNAs, noncanonical start codons, etc. Note that it is normal for intron boundaries to be out of phase, i.e., not located after a third codon position.

- You can annotate partial genes, misc_features, and introns by bringing up any annotation window, entering whatever data you need/want, and checking the appropriate radio button.

- Gene and exon duplications are common in plant mitochondrial genomes. Keep this in mind as you're inspecting the output, so that you don't overlook a duplication.

- To compile the results, run the *.tbl file through the `'tbl2asn'` script provided with the NCBI Sequin download. Check the error messages for warnings and errors that would indicate mistakes with your annotations. This is one of the best ways to find annotation errors. The *.tbl file can be submitted directly to GenBank, and the GenBank formatted file (output by tbl2asn) can be submitted to any number of programs (e.g., OGDRAW) to create a circular genome map.

PARSING GENBANK-FORMATTED FILES

There are numerous parsers available for GenBank formatted files, but many of them cannot deal with *trans*-spliced introns. I've included a Perl script for parsing GenBank files. It should parse any GenBank-formatted file, but it's mainly been tested on plant mitochondrial genome files. This script requires the Getopt::Long module (http://perldoc.perl.org/Getopt/Long.html). Follow these instructions to use this script:

1. Put `'parseGenbank.pl'` where you'd like to run it.
2. Make sure it's executable by typing the following command:

```
'chmod u+x parseGenbank.pl'
```

3. Type `'./parseGenbank.pl'` to see the usage. The only required input is a GenBank file (e.g., 'legume.gbf'). Calling the script with no options will just list the size and GC content of the genome:

```
$ ./parseGenbank.pl legume.gbf
```

4. Use the following command to extract nucleotide sequences for the genes. These will be written out to a directory called `'legume_nt'`.

```
$ parseGenbank.pl legume.gbf --nt
```

These scripts are fully readable and writable, and they can be easily modified to suit your specific needs.

TERMS AND CONDITIONS

Contact me at aja@uark.edu if you encounter any bugs or unanticipated output.