

Table of Contents

How to use this guide	4
1 Introduction to DNA Master	6
1.1 DNA Master overview	6
1.2 Installation.....	6
1.3 Quick Start Guide	6
1.4 DNA Master program structure.....	6
1.5 Analysis programs running within DNA Master	7
1.5.1 Glimmer	7
1.5.2 GeneMark.....	8
1.5.3 Aragorn.....	8
1.6 Setting Preferences	9
1.6.1 Set Default Translation Table.....	9
1.6.2 Set color preferences	9
1.6.3 Set start codon choices	10
1.6.4 Set default values for BLAST searches	11
1.6.5 Choose a default location for saving files.....	11
1.6.6 Finishing up your Preference settings	12
1.7 Getting help	12
1.8 Checking for updates.....	13
2 Provisional Cluster assignment of your phage	15
2.1 Overview.....	15
2.2 BLASTing your sequence against the mycobacteriophage database	15
2.3 Cluster assignment.....	18
3 Importing your phage genome sequence into DNA Master	21
3.1 Overview.....	21
3.2 Where do I get my phage genome sequence from?	21
3.3 Importing your DNA sequence into DNA Master	22
3.4 Reverse-complementing your sequence	24
4 Performing and viewing a rapid automated annotation of your genome .	25
4.1 Overview.....	25
4.2 Running Auto-Annotate	25
4.3 Saving your file	27
4.4 Looking at the output of your automated annotation	27
4.4.1 Viewing the documentation.....	28
4.4.2 Viewing features in the Feature Table	29
4.4.3 Viewing the sequence in the Sequence tab	31
4.4.4 Viewing ORFs in the Frames window.....	32
4.5 Running the BLAST function.....	35
4.6 Re-opening an archived (saved) file.....	37

5	Gathering additional information for refining your annotation.....	39
5.1	Generating a six-frame translation	39
5.2	Generating a provisional genome map in DNA Master	42
5.3	Generating a graphical output from GeneMark.....	43
6	Using Phamerator to assist with annotation	47
6.1	Overview.....	47
6.2	Why Phamerator is useful to you at this stage of your annotation.....	47
6.3	How did my genome get into Phamerator already?	47
6.4	Making Phamerator maps.....	48
6.5	Understanding and using the genome maps made by Phamerator.....	50
6.6	Viewing nucleotide sequence similarities in Phamerator	52
6.7	Other Phamerator features	54
6.8	Saving Phamerator maps	55
7	Guiding Principles of Bacteriophage Genome Annotation.....	57
7.1	Overview.....	57
7.2	The Guiding Principles	57
8	Gene by gene: evaluating and improving your draft annotation	61
8.1	Overview.....	61
8.2	Button-pushing mechanics reserved for Section 9	61
8.3	Decision Tree for evaluating the draft annotation	61
8.4	Evaluating protein-coding gene calls	63
8.4.1	Is the designation of this ORF as a gene well-supported?	63
8.4.2	Is the called start site for this gene the best possible choice?	66
8.4.3	Is this gene part of a programmed translational frameshift?	70
8.4.4	Does this gene contain an intron?.....	72
8.4.5	Does this gene wrap around the ends of the genome?	73
8.5	Checking gaps in the draft annotation for uncalled genes.....	75
8.6	Finding and refining tRNA and tmRNA genes	75
8.7	Completing your annotation refinement.....	76
9	The mechanics of making changes to your annotation.....	77
9.1	Overview.....	77
9.2	Making common changes to your annotation	77
9.2.1	Deleting a gene	77
9.2.2	Adding a gene.....	78
9.2.3	Changing the start site for a gene.....	78
9.3	Common steps to take after making changes	79
9.3.1	Posting changes.....	79
9.3.2	Validating your annotation	80
9.3.3	Renumbering annotated features	81
9.3.4	Re-BLASTing a gene	82
9.4	Making less common changes to your annotation	85
9.4.1	Annotating programmed translational frameshifts.....	85
9.4.2	Annotating introns.....	91
9.4.3	Annotating wrap-around genes	91

9.5	Predicting tRNA and tmRNA genes.....	91
9.5.1	Running web-based Aragorn (version 1.2.28).....	92
9.5.2	Running tRNAscan-SE (version 1.21).....	93
9.5.3	tRNA secondary structure and end determination	95
9.5.4	Entering a tRNA in DNA Master	96
9.5.5	Identifying and annotating tmRNA genes.....	97
9.6	Documenting your gene calls	98
10	Assigning gene functions	101
10.1	Overview.....	101
10.2	Using bioinformatic tools to assign gene function	102
10.2.1	BLASTP	102
10.2.2	Conserved Domain Database	103
10.2.3	HHpred.....	105
10.3	Other ways to assign gene function	108
10.3.1	Syteny	108
10.3.2	Prior functional assignments	109
10.3.3	Phamerator	109
11	Merging and checking annotations	111
11.1	Merging overview.....	111
11.2	Merging multiple annotations into a single file.....	111
11.3	Checking an annotation.....	116
12	Submitting final files for review and GenBank submission.....	119
12.1	Details of your final DNA Master (.dnam5) file.....	119
12.2	Details of your author list.....	119

How to use this guide

Once you have a finished phage genome sequence, you are ready to make predictions as to the locations and functions of the tRNA-coding and protein-coding genes. This guide will provide step-by-step instructions as to how to do this.

There are several different ways you can use this guide.

- Begin at **Section 1**, and proceed section by section through the entire guide. This approach will give you a complete understanding of the entire process of annotation and how each of the programs involved works. It's a lot of information, but hopefully you'll emerge from the other side far more knowledgeable about genes and gene calling.
- If you've already used the **DNA Master Quick Start Guide** to create an automated annotation, you can jump in at **Section 5**, and proceed from there. You'll be skipping some basics, but you can always refer back to relevant sections if needed.
- If you're eager to get straight to gene calling, you can perform an automated annotation using the **DNA Master Quick Start Guide** or **Section 4** of this guide, then proceed to **Section 8** which covers how to refine your automated annotation. References back to previous sections are provided so that you'll be able to locate all the information you need.
- If you're already an experienced annotator, and all you want to know is how to push the correct buttons to modify gene calls in DNA Master, **Section 9** is for you. It's an à-la-carte section of "How-To" functions.
- Finally, even if you're accustomed to using a different program to annotate phage genomes, you can use the Guiding Principles described in **Section 7.2** to see how we think about making the best possible gene calls in phage genomes.

A NOTE ON CLASSROOM PRAGMATICS

If you have a group of students annotating a single genome there are several different ways of organizing this activity. Assuming you have a class of around 20 students, there are two main considerations.

1. It works well for students to work in pairs, if possible using two computer stations. One of these can be set up to run DNA Master, while the other is set up to run Phamerator, as well as having other files (such as a six-phase translation) open.
2. You can organize students or groups of students such that:
 - All students annotate all of the genome. Upon completion, student groups (e.g. 5 groups of 4 students each) can each lead a discussion on a segment of the genome (i.e. 20% of it) aimed at resolving any differences found by different groups. The data are then compiled into a single DNA Master file.
 - Groups of students (e.g. 5 groups of 4 students) annotate a different segment of the genome (e.g. ~20%), followed by merging of the five DNA Master files into a single composite file. Instructions are provided in Stage 9 for doing this.

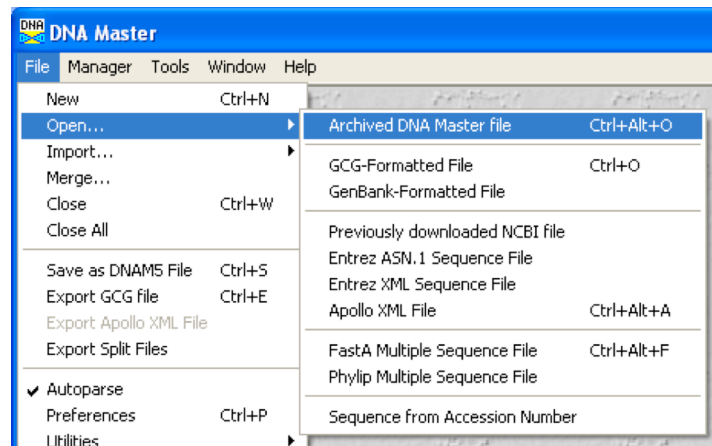
There are of course many other configurations and operational means of accomplishing your annotation. But it is helpful to keep in mind that the goal should be that all participants understand the full genomic context of the phage genome once the annotation is completed.

AN IMPORTANT NOTE ABOUT THIS GUIDE'S SYNTAX

In this guide, we will refer to menus and submenus as follows. If the command is:

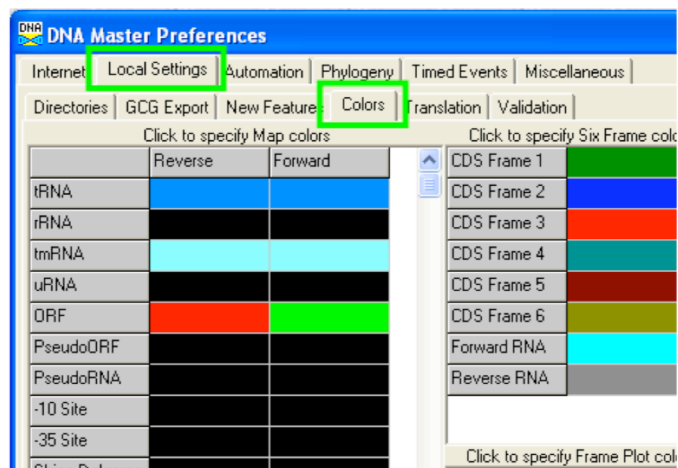
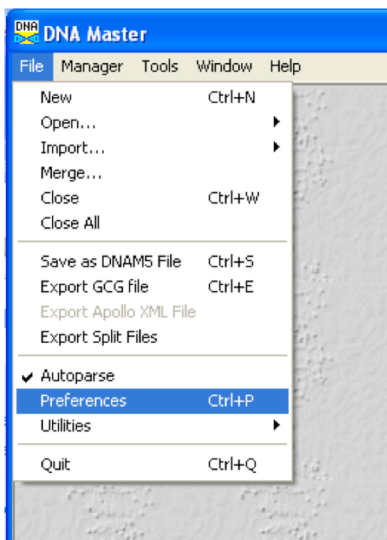
File → Open → Archived DNA Master file

this means that you should click on the **File** menu at the top, scroll down to the sub-menu (**Open**), and select the sub-sub-menu (**Archived DNA Master file**) that appears.



Tabs will be indicated by brackets, and sub-tabs will be shown by double brackets.

File → Preferences [Local Settings] [[Colors]]



1 Introduction to DNA Master

1.1 DNA Master overview

The key program you will use in your genome annotations is **DNA Master**. DNA Master is a DNA sequence editor and analysis package that combines, analyzes, and displays data from a variety of DNA analysis programs, including GeneMark, Glimmer, Aragorn, and BLAST. It organizes and collates all of these data into various tables and forms and saves it a single file with a **.dnam5** extension.

1.2 Installation

This guide assumes that you have installed DNA Master and can open the program successfully. If this is not the case, please install DNA Master before continuing with this guide. System requirements and installation instructions are provided in **Appendix I**, and are also available at <http://phagesdb.org/DNAMaster/>.

1.3 Quick Start Guide

Appendix II is the **DNA Master Quick Start Guide**, which you may find useful if you are using DNA Master for the very first time and just want a quick look at basic functions. However, all parts of the Quick Start Guide are covered in more detail in this guide, so you may choose to use the Quick Start Guide as a future reference or a teaching tool.

1.4 DNA Master program structure

The various files, tables, and databases that DNA Master uses are a little complex, but a general understanding of the structure is important and will help prevent lost work.

The Feature Table

There are two important places DNA Master stores information about a genome annotation. The first, called the **Feature Table**, contains information about each feature (usually a gene) in a genome, including name, position, length, protein sequence, BLAST results, function, notes, etc. Within DNA Master, the data in the Feature Table for a particular genome can be viewed by going to the “**Features**” tab. When you **Post*** changes to your annotation, like changing a start position or adding a gene, you’re altering the Feature Table.

* See **Section 9.3.1** for more on the importance of **Posting** changes.

The Documentation

The second place DNA Master stores information is the **Documentation**, accessible via the Documentation tab. This text contains much of the same information as is present in the Feature Table, but in a less human-friendly and more computer-readable format. Note that not all of the information from the Feature Table is contained in the Documentation Tab (e.g., amino acid sequence and BLAST hits are not present).

Interaction between the two

The Feature Table interacts with the Documentation as shown in **Figure 1.1**.

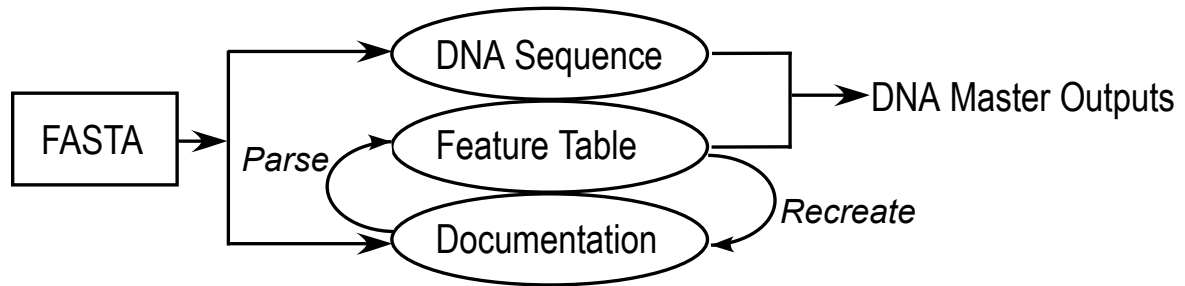


Figure 1.1

There are two functions—accessible through the Documentation tab—that control the interaction between the Feature Table and the Documentation:

Parse takes the contents of the Documentation and uses them to **OVERWRITE** the **Feature Table**. Parsing is done automatically by DNA Master when a genome is auto-annotated, but thereafter should be used rarely if ever. The danger is that you'll have posted data to the Feature Table that are not included in the documentation, and then when you Parse, those data will be lost.

Recreate takes the contents of the Feature Table, and uses them to **OVERWRITE** the **Documentation**. This will update the Documentation with changes you've posted, and thus serves as a helpful backup of some of your data.

IMPORTANT TO REMEMBER:

Using **Parse** may overwrite user-inputted data, and thus Parsing may be **harmful**.

Using **Recreate** will store some user-inputted data in a new location, and thus it's **helpful**.

1.5 Analysis programs running within DNA Master

As noted above, DNA Master runs a collection of programs that can assist in annotation and analysis of your phage genome. The following is a brief explanation of some of the key programs that DNA Master will be running for you, and some of their stand-alone versions that you will be using.

1.5.1 Glimmer

Glimmer (version 3.02) is a program that predicts the coding potential of open reading frames (ORFs). DNA Master is set by default to use Glimmer in a heuristic way, meaning that it searches for potential coding regions (such as in long open reading frames) and then applies the nucleotide codon biases in those ORFs to search for other potential ORFs with similar biases. As such, it is not dependent on the use of externally defined parameters to determine coding potential. Glimmer also recognizes the use of TTG in addition to ATG and GTG as translation initiation (i.e. start) codons. It has very good predictive power for genes.

You will typically use Glimmer as a program that will run when you request DNA Master to

perform an auto-annotation of your phage genome sequence and you will not be required to run it directly.

If you'd like to run Glimmer directly, it is available as a stand-alone program and is web-accessible at:

http://www.ncbi.nlm.nih.gov/genomes/MICROBES/glimmer_3.cgi

1.5.2 GeneMark

GeneMark (version 2.5) provides a similar functionality to Glimmer and is used to predict genes. Its algorithms are different, however, and the joint use of Glimmer and GeneMark is a powerful combination for gene prediction. As with Glimmer, DNA Master runs GeneMark automatically within the Auto-Annotation function. Within DNA Master, GeneMark is heuristic, in that it learns from the genome what the codon usage preferences are in the longest ORFs and then applies this model to predict the remainder of the genes. GeneMark also takes into account potential ribosome binding sites when predicting gene start positions.

In addition, a second GeneMark prediction is helpful for accurately identifying the genes in your phage genome. In this internet browser-accessible version, the gene predictions are made using a codon usage model built from a previously annotated organism. GeneMark has many bacterial models available, and so for bacteriophage we pick a model based on the host organism. For the mycobacteriophage, we use *Mycobacterium smegmatis*.

GeneMark online is available at:

http://opal.biology.gatech.edu/GeneMark/genemark_prok_gms_plus.cgi

The web version contains two key features that are useful for phage genome annotation:

- It allows you to specify the codon usage model from a bacterial host to use for gene prediction, rather than generating a new model heuristically. A codon usage model for *Mycobacterium smegmatis* is available and can be selected to generate gene predictions in the phage genome based on the host's codon preferences. This sometimes allows you to find smaller genes that are not called during heuristic scans, but are likely to be reliable gene calls because they share codon preferences with the host. We refer to this output as the "**GeneMark-Smeg**" output.
- It provides a graphical output (as .pdf) of the gene predictions and coding potential. This is especially useful when you are determining gene starts.

1.5.3 Aragorn

Aragorn is a program for finding tRNAs and tmRNAs. Aragorn (version 1.1) can be run directly within DNA Master, although it is also accessible as a stand-alone program at:

<http://130.235.46.10/ARAGORN/>

The version of Aragorn available online is newer than the version embedded within DNA Master. It is **important to run the updated web-based version of Aragorn** (version 1.2.33.c.) in addition to the DNA Master version because it is better at determining the correct ends of tRNAs and because the version within DNA Master has a specific set of parameters that differ from the default. In addition, another tRNA predictor, tRNAscan-SE, is utilized to fine-tune the tRNA calls. Please refer to **Section 9.5** when you evaluate your tRNAs in your genome.

1.6 Setting Preferences

In general, setting preferences in DNA Master is a matter of opening the Preferences Window, making changes, and applying these changes. There are **five important preferences that you MUST set** before continuing with this guide. They are described in the next five subsections.

To get to the Preferences Window, select:

File → Preferences

You will see a dialog box with a series of tabs (Internet, Local Settings, ...) each of which has another set of sub-tabs associated with it.

1.6.1 Set Default Translation Table

Changing this setting ensures you are using the correct translation tables for phages. Select:

File → Preferences [Local Settings] [[New Features]]

- From the Default Translation Table dropdown menu, select '**Bacteria and Plant Plastid Code**'.
- Make sure that the boxes marked '**Add New Features to Documentation**', and '**Add New Features to Feature Table**' are both checked.
- Click '**Apply**'. Note that the dialog box remains open.

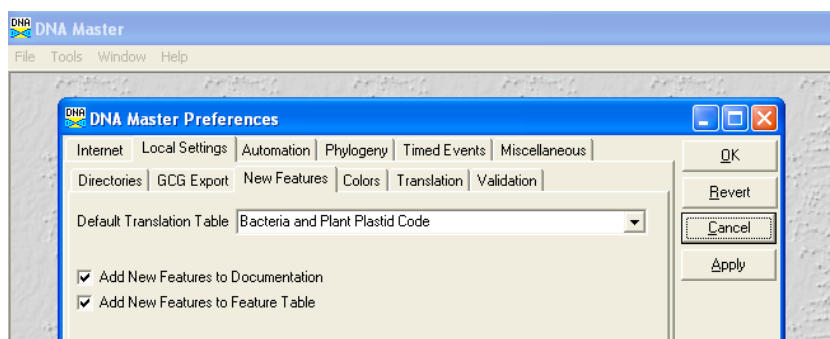


Figure 1.2

1.6.2 Set color preferences

You can select display colors for genes and tRNAs in various visual representations of your genome. The colors we recommend below are our preferences, and are used in most of the screenshots in this guide. You can select any colors you like, but note that if you use different colors, exported six-frame translations may not be properly viewable in Microsoft Word.

To set your colors to our recommended values, go to:

File → Preferences [Local Settings] [[Colors]]

Then set the values as shown below.

- Click on the colored box you want to change.

- A dialog box pops up with the color options.
- Click on the **color** of choice and then click **OK**.
- Continue to the next color.
- Don't forget to click '**Apply**' to save changes.

CDS Frame 1	Yellow	CDS Frame 4	Gray
CDS Frame 2	Pink	CDS Frame 5	Light Green
CDS Frame 3	Light Blue	CDS Frame 6	Light Red

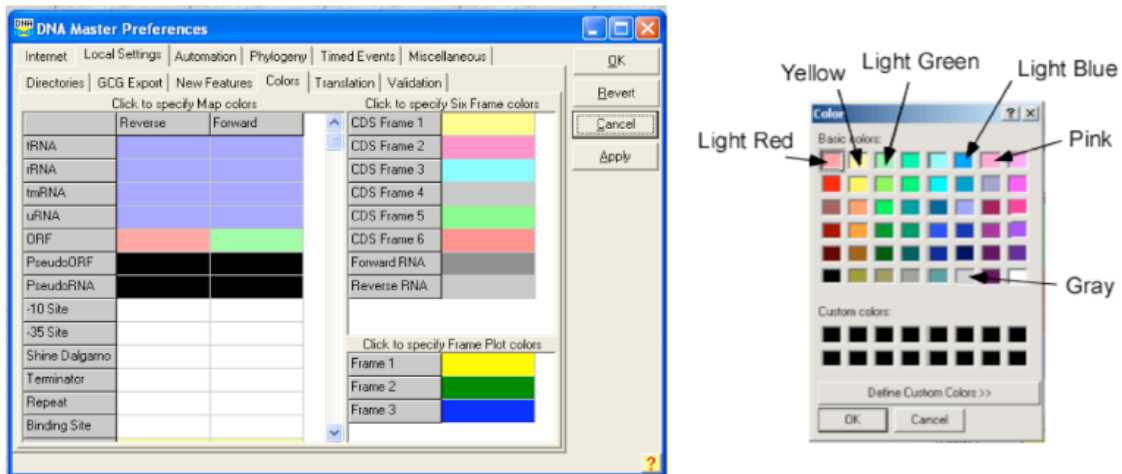


Figure 1.3

1.6.3 Set start codon choices

Because TTG is used as a translation initiation (start) codon in mycobacteriophage genomes – albeit rarely – you must make sure DNA Master recognizes it. To do so, go to:

File → Preferences [Local Settings] [[Translation]]

- All boxes must be checked, as shown in **Figure 1.4** below.
- Click '**Apply**'

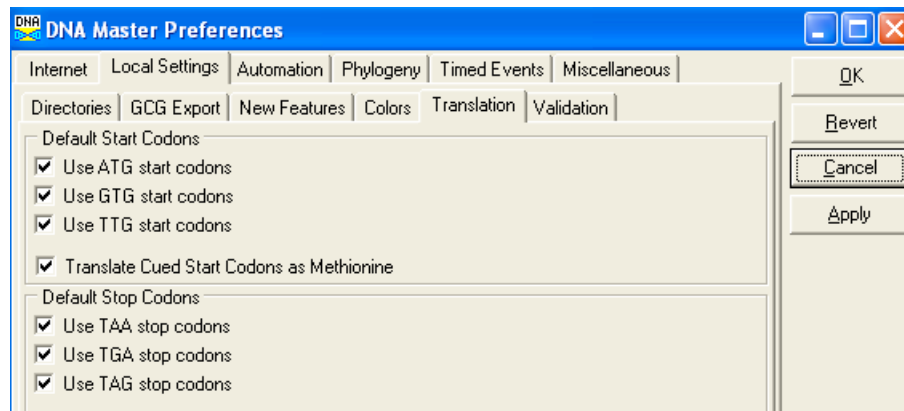


Figure 1.4

1.6.4 Set default values for BLAST searches

DNA Master can run batch BLAST searches and store the results for subsequent viewing. There are several settings relating to BLASTing inside DNA Master that may be helpful. Our suggestions are shown in Figure 1.5. Get to the BLAST menu by going to:

File → Preferences [Internet] [[Blast]]

- Set your preferences.
- Click 'Apply' to save changes.

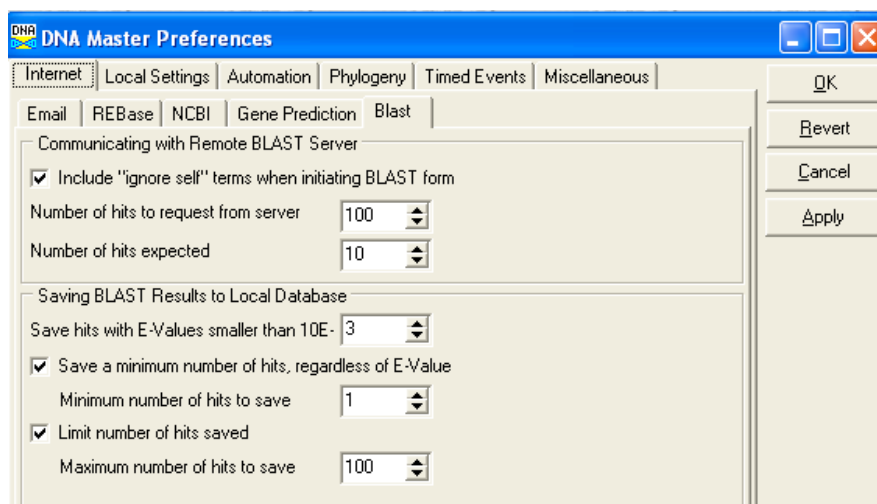


Figure 1.5

1.6.5 Choose a default location for saving files

DNA Master generates a number of files when it runs. It's good practice to create a dedicated DNA Master archiving folder, then direct DNA Master to use it. To do so, go to:

File → Preferences [Local Settings] [[Directories]]

- Click the 'Browse' button next to the 'Archive to...' field.

- Select your archiving folder, or create a new one.
- Click 'Apply' to save.

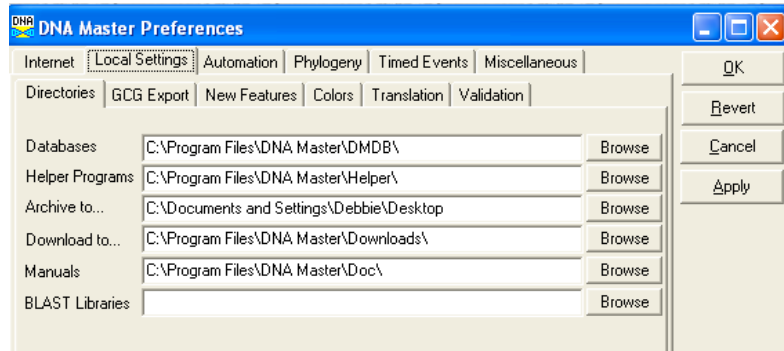


Figure 1.6


1.6.6 Finishing up your Preference settings

Once you have finished setting your DNA Master preferences:

- Click the 'OK' button.
- Click 'Yes' in the dialog box that asks if you want to save changes.

The Preferences Window will close.

1.7 Getting help

Help files and tutorials are available within DNA Master for many of its functions. Help is always available by clicking on the yellow  button at the lower right corner of every window, or through the 'Help' menu.

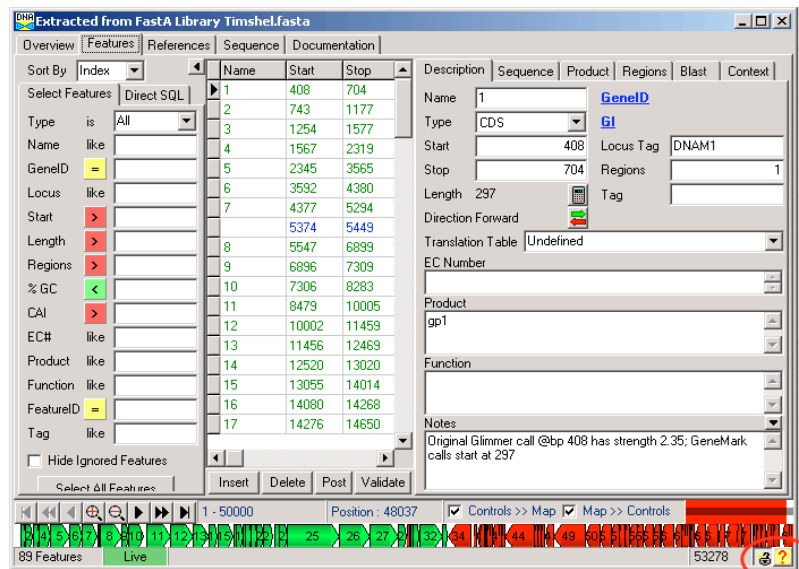


Figure 1.7

To get a sense of how the help files work, go to:

Help → Help

- Read the 'Welcome to DNA Master' and the 'Getting Started Tutorial' sections.

1.8 *Checking for updates*

DNA Master is regularly updated, and with an internet connection it is easy to make sure your copy is up-to-date. Go to:

Help → Update DNA Master

- If a new version is available, it will update the program, and a dialog box will appear when completed. Please note that you must have an active internet connection to do this!
- When the update is complete, close and restart the program.
- As of the time of writing (October 2011), the most up-to-date version of DNA Master is Version 5.22.5 Build 2338, dated 17 Oct 2011. You can find your current version by going to:

Help → About

2 Provisional Cluster assignment of your phage

2.1 Overview

All sequenced mycobacteriophage genomes have been compared to one another, and based on these comparisons they have been grouped into **clusters** of related phages. Some of these clusters are small (Cluster M currently has only two members), whereas others are quite large (Cluster A has over 90 members). Some clusters are further divided into **subclusters**; for example, Cluster B's genomes are currently divided into five subclusters: B1, B2, B3, B4, and B5. There are also some phages (ten currently) who have no close relatives, and therefore are classified as **Singletons**. Up-to-date cluster assignments are available at:

<http://phagesdb.org/clusters/>

Your phage's final cluster designation depends on a variety of analyses, as described in:

Hatfull *et al.*, (2010) Comparative genomic analysis of sixty mycobacteriophage genomes: Genome clustering, gene acquisition and gene size. *J Mol Biol.* **397**, 119-143.

In the meantime, however, it is helpful to make a provisional cluster assignment for your phage. This can be done using just a completed genome sequence, before any annotation has taken place because clustered phages usually share a span of 50% or more recognizable nucleotide similarity across their genomes.

Performing a BLAST search of your phage sequence against a database of mycobacteriophage genomes provides a simple and quick approach to making a provisional cluster assignment.

2.2 BLASTing your sequence against the mycobacteriophage database

To BLAST your genome on phagesdb.org:

- Go to <http://phagesdb.org/phages/>
- Locate your phage in the phage list, then click to open its detail page.
- Click on the green "Locally BLAST this genome" button.
- It will open a page that looks like the one in **Figure 2.1**.

Local Phage BLAST

This tool will run a local BLAST search against our phage database. This will include some genomes that are not yet in GenBank and thus not accessible via NCBI BLAST.

Choose program to use and database to search:

Program Database

Enter sequence below in **FASTA** format

```
>Etude
AGCGACACTTCTCTCTGGAATTCAGGCAAGAACATGAGGGGGTTAGCGCCCTAAA
ACCCCTGGTAGGAGGCTAAATCGTGGGTAGAGGACGTGGTAAGGACCCGTCAGCCCTGG
TGGGCGGTCTCGGGACAGTCGTCGCGCACGCGCTCGGCCTGGGAGGCCAAGTTGCCGC
CAAACCGAAGAACCGCAGGAATACGCGGTGCAGATGGCCGAAAGCCTCGGTTGGGAGGT
TGAGAAGCCGAACGTTGGACCAATCAGGGGATGCACGCCGCTGGTATCGAGACTTGAC
GATGCGCAAGGGCGATGCGTACGTGTATGCGACGTCACCTGGCCTAATGGCCGATTCG
```

Or load it from disk

Set subsequence: From To

The query sequence is filtered for low complexity regions by default.

Filter Low complexity Mask for lookup table only

Expect Matrix Perform ungapped alignment

Query Genetic Codes (blastx only)

Database Genetic Codes (tblast[nx] only)

Frame shift penalty for blastx

Figure 2.1

- The defaults are set so that the program will run **blastn** (i.e. a nucleotide search against a nucleotide database) against a database of previously sequenced mycobacteriophage genomes (e.g., Mycophages as of 6.01.11).
- Click on the '**BLAST!**' button. It is just above the gray dividing bar at the center of **Figure 2.1** above.

A new page will open showing the results of the BLAST search, as shown in **Figure 2.2** below.

Your query is represented by a black bar underneath "Color Key for Alignment Scores". Subject sequences from the database that align well to your query sequence are represented by colored bars beneath the black bar. The length and location of the subject bars indicates the portion(s) of the query sequence the subject sequences match. The quality of each alignment is shown by color, with the best matches colored red.

Distribution of 488 Blast Hits on the Query Sequence

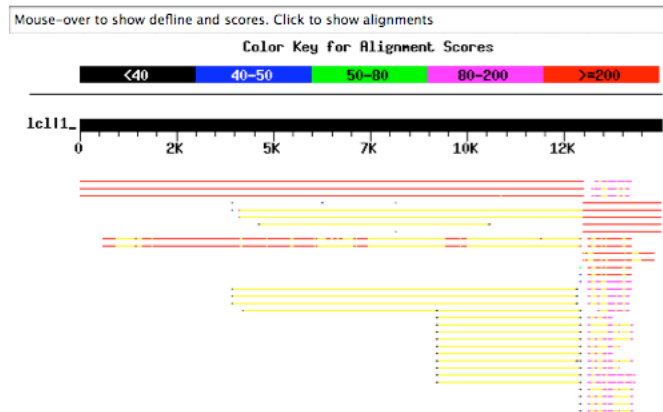


Figure 2.2

To see which subjects your query has aligned to, simply mouseover any of the colored bars, and the subject's name will appear in the box above the "Color Key for Alignment Scores". Then, either scroll down or click on one of the lines to get the names of subject sequences that have the best alignments to your query sequence, listed in order from best match to worst match (see below). After each subject sequence name is the raw score of the alignment to your query sequence (higher is a better alignment), and the E value (lower is a better alignment).

Sequences producing significant alignments:	Score (bits)	E Value
UPIE Complete Sequence, 73784 bp including 10 bp 3' overhang (TC...	1.314e+04	0.0
LeBron	1.178e+04	0.0
JoeDirt Final Sequence, 74914 bp including 10 bp 3' overhang (TC...	1.169e+04	0.0
Microwolf Final Sequence, 50864 bp including 10 bp 3' overhang, ...	4022	0.0
Vix Complete Sequence, 50963 bp including 10 bp 3' overhang (CG...	3998	0.0
Methuselah Complete Sequence, 50891 bp including 10 bp 3' overha...	3998	0.0
JHC117 Final Sequence, 50877 bp including 10 bp 3' overhang, Clu...	3998	0.0
Bx22	3998	0.0
Faith1 Complete Sequence, 75960 bp including 10 bp 3' overhang (...)	1388	0.0
Rumpelstiltskin Complete Sequence, 69279 bp including 10 bp 3' o...	1364	0.0
Heldan Complete Sequence, 50364 bp including 10 bp 3' overhang (...)	353	1e-95
Rockstar Complete Sequence, 47780 bp including 10 bp 3' overhang...	232	3e-59
Peaches	212	2e-53
TiroTheta9 Complete Sequence, 51367 bp including 10 bp 3' overha...	204	6e-51
MeeZee Complete Sequence, 51368 bp including 10 bp 3' overhang (...)	204	6e-51
Eagle	204	6e-51
LHTSCC Complete Sequence (51813bp, including 10bp 3' overhang: C...	196	1e-48
Shaka Complete Sequence, 51369 bp including 10 bp 3' overhang (C...	188	4e-46
Twister Complete Sequence, 51094 bp including 10 bp 3' overhang ...	149	3e-34
George Final Sequence, 51578 bp including 10 bp 3' overhang, Clu...	137	1e-30
Benedict Complete Sequence, 51083 bp including 10 bp 3' overhang...	137	1e-30
Airmid Complete Sequence, 51241 bp including 10 bp 3' overhang (...)	137	1e-30
Theia Complete Sequence, 51543 bp including 10 bp 3' overhang (C...	129	3e-28
Cuco Complete Sequence, 50965 bp including 10 bp 3' overhang (CG...	129	3e-28
Bxh1	123	2e-26
Violet Complete Sequence, 52481 bp including 10 bp 3' overhang (...)	121	7e-26
Switzer	121	7e-26
Pari Complete Sequence, 50614 bp including 10 bp 3' overhang (CG...	121	7e-26
KBG	121	7e-26
Doom Final Sequence, 51421 bp including 10 bp 3' overhang (CGGAT...	121	7e-26
Dreamboat Complete Sequence, 51083 bp including 10 bp 3' overhan...	117	1e-24
Jasper	115	4e-24
Wille Complete Sequence, 51308 bp including 10 bp 3' overhang (CG...	113	2e-23
U2	113	2e-23
Solon	113	2e-23
SkiPole	113	2e-23
RidgeCB Complete Sequence, 50844 bp including 10 bp 3' overhang ...	113	2e-23
Perseus Complete Sequence, 53142 bp including 10 bp 3' overhang ...	113	2e-23
MrGordo Complete Sequence, 50988 bp including 10 bp 3'overhang (...)	113	2e-23
Lockley	113	2e-23
Lesedi	113	2e-23
KSSJEB	113	2e-23
.TC27	113	2e-23

Figure 2.3

Scroll down further (or click on the blue raw score number) to get the nucleotide alignment of your query sequence (top) to each subject sequence (bottom). The numbers on the sides of the sequences indicate the nucleotide coordinates within each sequence. Identical nucleotides are connected with vertical lines and smaller gaps in the alignment are shown by horizontal dashes.

Distribution of 1211 Blast Hits on the Query Sequence

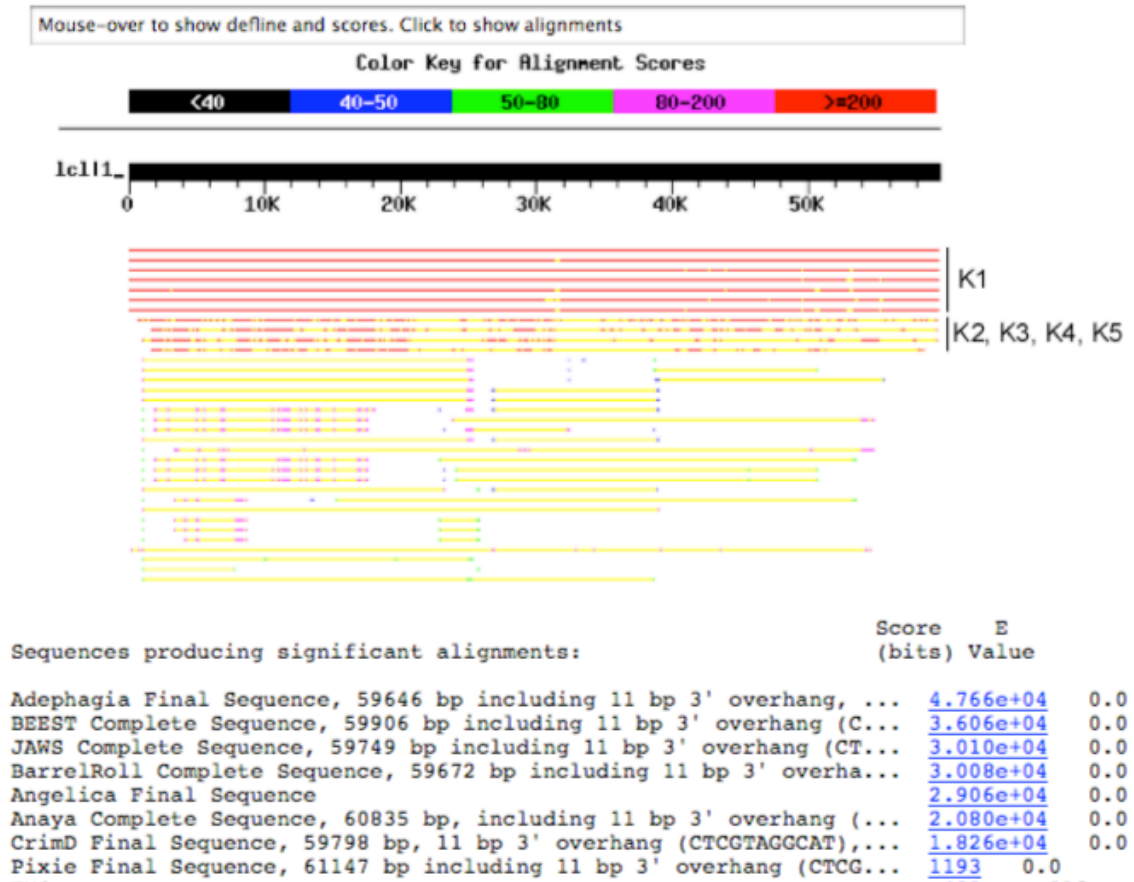


Figure 2.5

Adephagia's best hit is to itself. After that, there are six heavy red lines that indicate very similar genomes to Adephagia's. Scrolling down to the "Sequences producing significant alignments" section, we can see that these red lines correspond to the genomes of BEEST, JAWS, BarrelRoll, Angelica, Anaya, and CrimD. Using phagesdb.org, we can then look up the Cluster assignments of these six phages. All six, it turns out, are members of Cluster K, and Subcluster K1.

There are four more genomes that appear to have significant similarity to Adephagia, though the matches are less solid and cover less of the query sequence. These more tattered-looking red lines correspond to Pixie, TM4, Larva, and Fionnbharth. Using phagesdb.org, we can see that these are all member of Cluster K, though they belong to Subclusters K2-K5, not K1.

Therefore, we can provisionally determine that Adephagia is a member of **Cluster K** and **Subcluster K1**.

NOTE: Though the example above may seem clear-cut, Cluster assignment will not always be so simple. If it's not, don't be concerned. You may have found a new Singleton phage, or a phage that will lead to a new Subcluster being created. The main point of doing this now is so that you have an idea of which phages are most closely related to the one you are annotating. These closely related phages can be very useful guides as you go through the annotation process.

3 Importing your phage genome sequence into DNA Master

3.1 Overview

Now that you have a sense of your software and an overview of your phage genome, you are ready to move onto the really exciting stuff! The first thing you need to do is to download your phage's genome sequence, then import it into DNA Master.

3.2 Where do I get my phage genome sequence from?

Sequencing a phage genome involves two parts: Shotgun Sequencing and Finishing (aka Polishing). The second part, **Finishing**, involves generating targeted reads to fix weak areas, determining the type and/or sequence of genome ends, and orienting a genome to match convention. When performing annotations, you **must always use a Finished sequence file**, or your annotation work may have to be redone.

Fortunately, **phagesdb.org** only posts Finished sequence files, so be sure to get your sequence from phagesdb.org. Though you may have access to preliminary, un-Finished files from other sources, **the phagesdb.org site should be the only source for sequence when beginning annotation.**

A NOTE ON FILE TYPES

DNA, RNA, and protein sequence files are often saved in **fasta** format. This is the standard format required by many bioinformatics programs, including BLAST. Fasta files are simply text files where:

1. The first line begins with ">" and contains information about the sequence
2. Subsequent lines contain the sequence itself

For example, the first few lines of a phage genome sequence fasta file may look like:

```
> Giles Complete Genome Sequence, 53746 bp
GGCAGACTTTTTTTTGC GCGGGCGGCCCTGCGCGCGCGGCCCGCCCGCCCC
GCCGGGTCGGAGGCGGCCGAATGACGCCACCTCGGGCCGCGGTGGCCGAC
ACGCCGGATACGCCCGCAGAGGGCAAATCAGGGGCCAAAACGCGGGCCAA
```

A few things to keep in mind:

- Fasta files can be opened with any text editor.
- A file does not need to have the extension **.fasta** to be in fasta format. For example, if you rename Giles.fasta to Giles.txt, the file will still be fasta-formatted.
- Sequence files from phagesdb.org are in fasta format and have a **.fasta** extension.

To download your genome sequence as a fasta file, go to:

- <http://phagesdb.org/phages/>
- Scroll down to find your phage and click its name to open its detail page.
- Scroll down to the section titled ‘Sequencing Information’.
- Click on the ‘**Download fasta file**’ link, and save the file to a known location

IMPORTANT NOTES:

- If you can’t find the downloaded file, simply search your computer for a file named YourPhageName.fasta.
- If you are using a Windows emulator on a Mac (and use your internet browser on the Mac side to get the fasta file), then you should either copy the fasta file from the Mac side to the Windows side, or alternatively set up your emulator preferences so that it can directly read files from the Mac side from a shared folder.
- If for some reason you’re using a sequence file from a location other than phagesdb.org, be mindful that there are two possible orientations for a genome, and that yours needs to conform to the standard convention (the virion structural genes on the left, transcribed rightwards). If you determine that a sequence needs to be reverse-complemented, instructions are provided at the end of this section for doing so.

3.3 Importing your DNA sequence into DNA Master

You are now ready to import your fasta file into DNA Master. Open DNA Master, then go to:

File → Open → FastA Multiple Sequence File

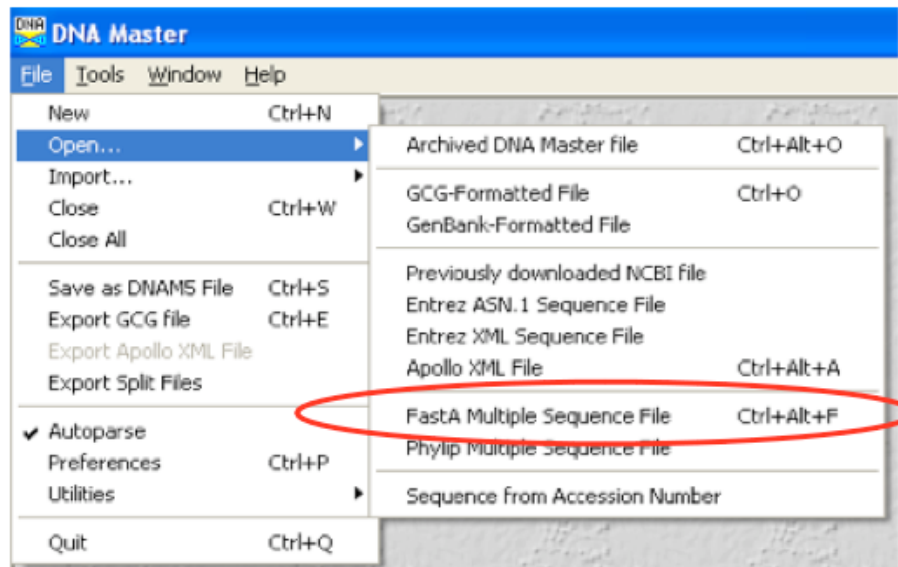


Figure 3.1

- Browse to the correct folder and select your fasta file.
- A window like the one shown in **Figure 3.2** appears.

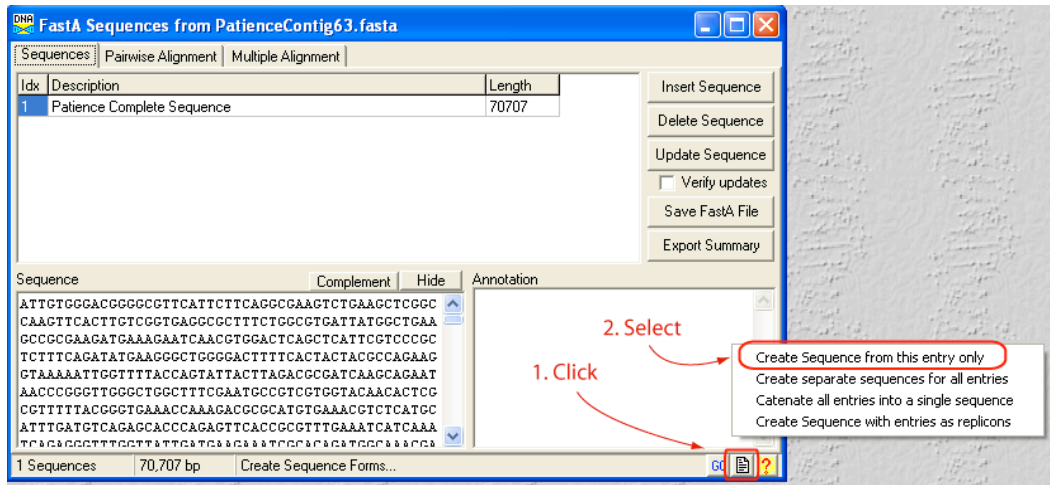


Figure 3.2

- Click on the Export button in the lower right hand corner (1).
- From the menu that opens, select 'Create Sequence from this entry only' (2).
- A new window titled 'Extracted from FastA library YourPhage.fasta' will open within DNA Master.

Let's take a moment to look at some of the new views that are available.

- There are five tabs in the new window: [Overview], [Features], [References], [Sequence], and [Documentation].
- Select the [Overview] tab if it's not already selected. Your window should look similar to the one in Figure 3.3.

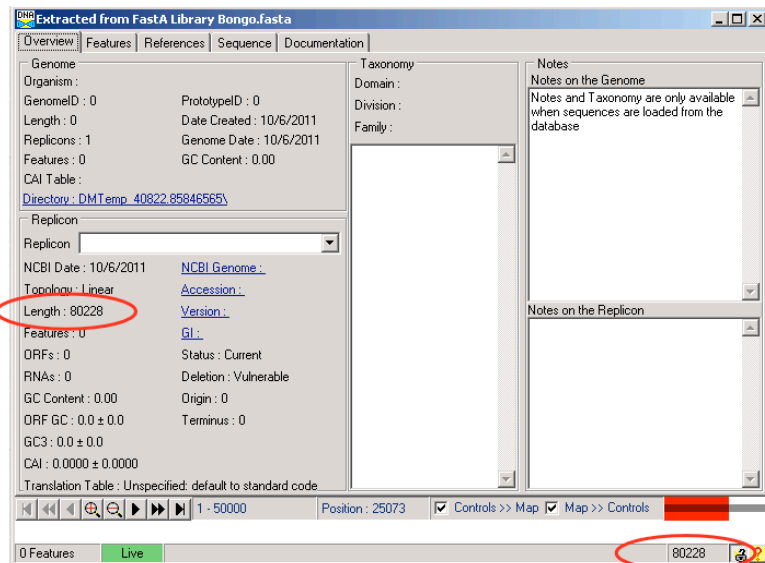


Figure 3.3

- Check the sequence length (shown in the red circles in Figure 3.3) and verify that it matches the published sequence length on your phage's detail page on phagesdb.org. If there is a discrepancy, restart the program and try importing

again, or re-download your sequence file from phagesdb.org.

- Select the [Sequence] tab. This tab displays the DNA sequence of your phage. You can click and drag to select part of the sequence, whereupon DNA Master will display the coordinates and length of the selected portion near the top of the window, as in **Figure 3.4**.

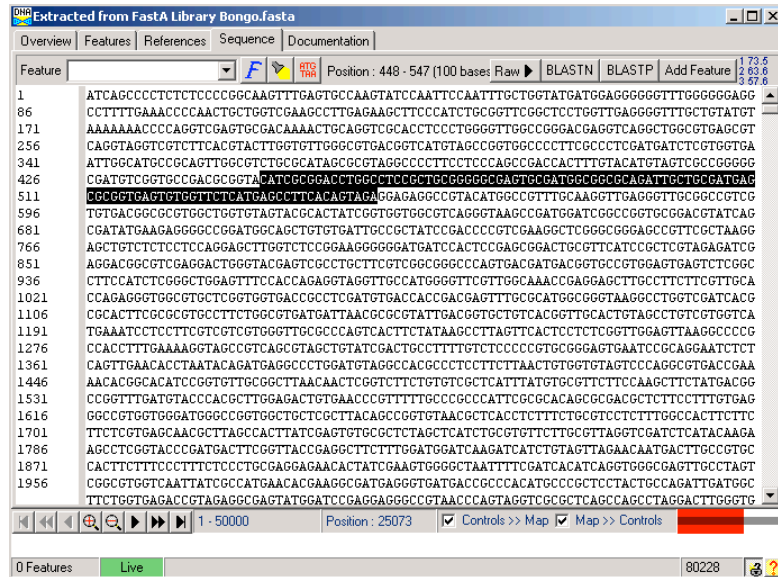


Figure 3.4

- Until you run an automated annotation in the next section, the tabs for [Features], [References], and [Documentation] are largely empty. We'll revisit these later.

Congratulations! You have now imported your phage sequence into DNA Master and are ready to run an Auto-Annotation.

3.4 Reverse-complementing your sequence

To re-emphasize, if you download your genome sequence from phagesdb.org, it will **NOT** need to be reverse-complemented. If you need to reverse-complement a sequence from a different source to match conventions, you can do so easily within DNA Master.

To reverse-complement a sequence:

- Go to the [Sequence] tab.
- **Make sure that no segment of the sequence is selected** (otherwise you will only flip that part—a big mess). If in doubt, just click somewhere within the sequence, but without selecting anything.
- Select: **DNA → Convert → Complement**
- A dialog box will open that asks if you want to convert XXXXX bp to 5' → 3'. Click 'Yes'.
- Select: **File → Save as**, then save your reverse-complemented file with a new name.

4 Performing and viewing a rapid automated annotation of your genome

4.1 Overview

DNA Master has an **Auto-Annotate** function that provides quick and simple identification of genes within your phage genome. It works by running Glimmer, GeneMark, and Aragorn, then combining the outputs from these programs to arrive at consensus gene calls. The consensus output is used to populate DNA Master's Documentation and Feature Table sections.

Generally, this auto-annotation will identify 80% or more of the genes accurately, but the careful refinement that you will perform in **Section 8** will be essential for obtaining the best possible annotation that will be ready for GenBank submission.

4.2 Running Auto-Annotate

- As shown in **Figure 4.1**, go to:

Genome → Annotation → Auto-Annotate

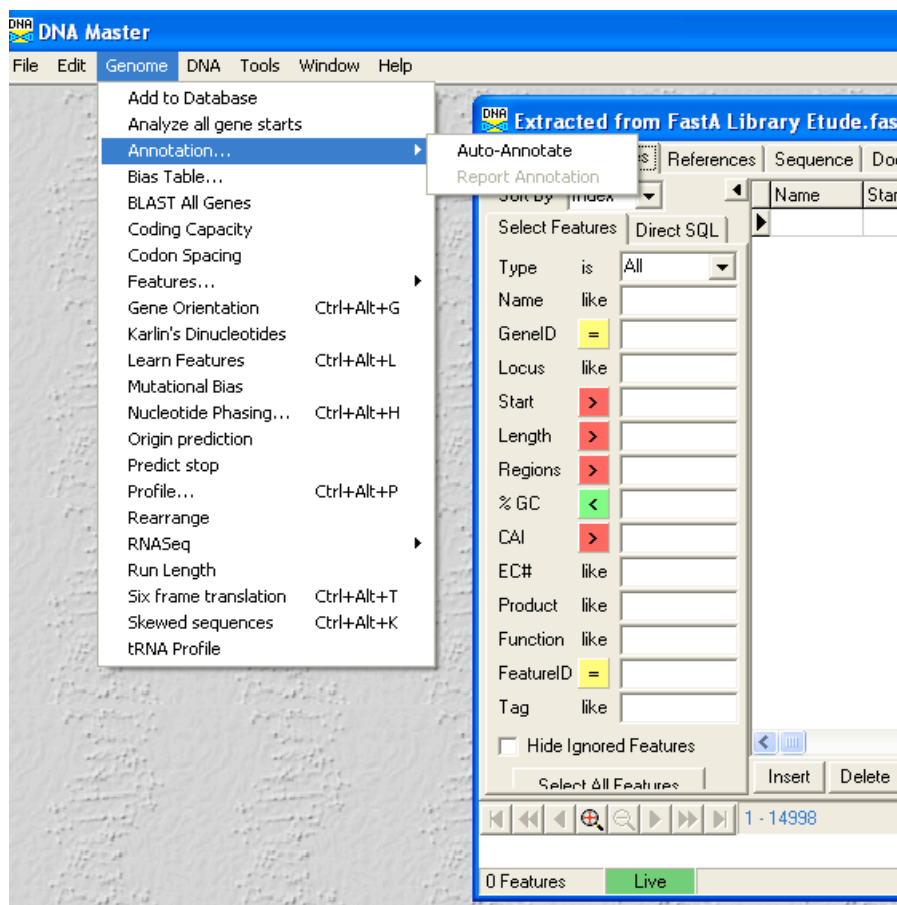


Figure 4.1

- An Auto-Annotate dialog box will open. We recommend that you use the settings shown in **Figure 4.2**.

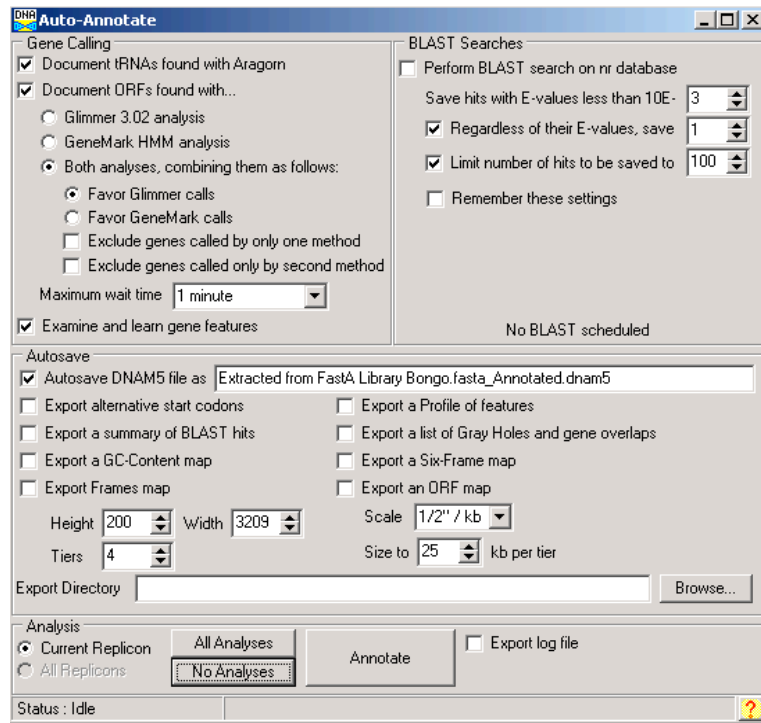


Figure 4.2

Click the '**Annotate**' button to launch the automated annotation. (Click '**Yes**' when prompted to "Erase features?")

SOME NOTES ON AUTO-ANNOTATE OPTIONS

- One key Auto-Annotate option is the ‘**Perform BLAST searches on nr database**’ checkbox. When checked, this option will BLASTP the protein product of each gene Auto-Annotate finds, then save the results for viewing later—a powerful tool, and recommended if you have the time. However, performing that many BLAST searches often takes more than 45 minutes, during which DNA Master will be inaccessible. If you’d like to move on to further steps quickly, uncheck this box and Auto-Annotate will run in fewer than five minutes.

See **Section 4.5** for how to BLAST genes at a later time.

- In the Gene Calling pane, we prefer to use the default option of using ‘**Both analyses**’ (Glimmer and GeneMark), with ‘**Favor Glimmer Calls**’ selected. Often, the two programs’ gene calls differ only in the location of the start codon, and since Glimmer recognizes TTG as a start codon, we prefer to favor its calls. If desired, you can try modifying options to see their effects on the resulting gene calls. Auto-Annotate runs quickly enough to experiment!

When there is a conflict between Glimmer and GeneMark calls, both calls will be reported in the gene’s Notes. If the two programs agree, the Notes will contain only one program’s call.

- The checkbox to ‘**Export a Six-Frame map**’ produces a translation of the sequence in all six frames, a useful asset for annotation. See Section 5 for generating maps and translations at a later time.

4.3 *Saving your file*

As with any program, it is important to **save your file often** to protect changes you’ve made from being lost. This can be done by going to:

File → Save as DNAM5 file

Choose a new file name if you wish to keep both previous and current versions. This is a way to keep backups of work you’ve done. To avoid confusion about which file is the current version, it is helpful to establish systematic naming conventions when saving files.

4.4 *Looking at the output of your automated annotation*

Once the Auto-Annotate function has run, it will return you to your main phage window. Under the [**Overview**] tab, however, you will see some immediate differences.

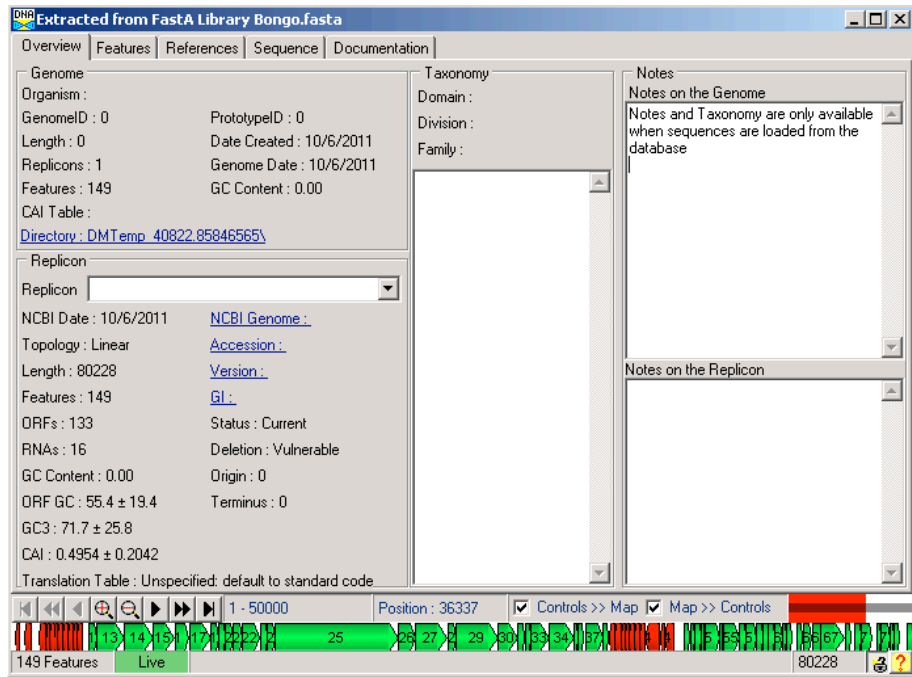







Figure 4.3

For example, note that there is a map showing the predicted genes at the bottom of the window. Genes transcribed leftwards and rightwards are shown in different colors depending on how you have set your DNA Master preferences (**Section 1.6.2**; green and red in **Figure 4.3**).

This map is dynamic and can be manipulated as follows:

- Roll your mouse over the map. You will see the number changing in the box above it labeled '**Position**'. This reports the coordinate in the genome where your mouse is pointing.
- Click on the  button to zoom in and the  button to zoom out.
- Click on the left and right arrows to move  a little each way,  a lot each way, or  to the extreme left or right ends.

4.4.1 Viewing the documentation

Auto-Annotate writes its output to the **Documentation**. Though you will generally work in the **[Features]** tab, it is useful to be familiar with this underlying Documentation. Click on the **[Documentation]** tab to take a look.

You will see that DNA Master has populated the Documentation with the consensus outputs from Glimmer, GeneMark, and Aragorn. In the example shown in **Figure 4.4**, the first line says "CDS complement (238-450)". This means the first feature is a protein-coding sequence (CDS) transcribed right to left and located at coordinates 238 – 450.

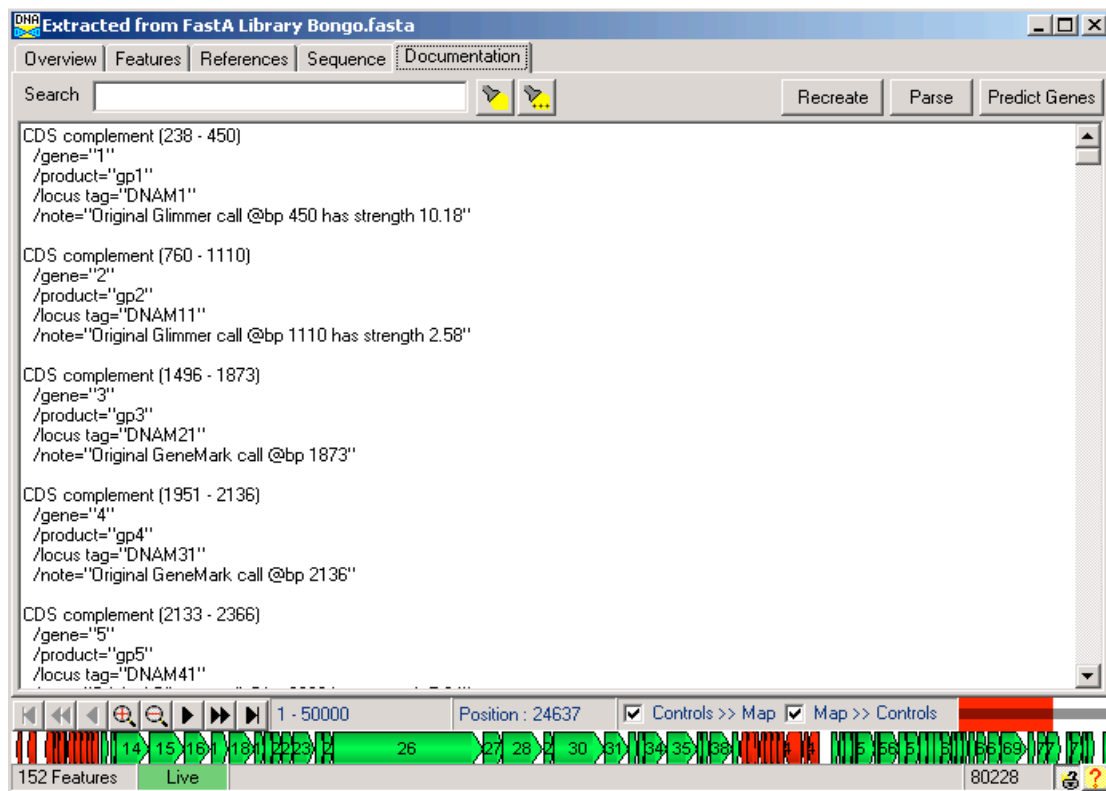


Figure 4.4

This “complement” orientation is worth thinking about for a moment. It means that the first base of the first codon of this predicted gene is at position 450, while the last base of the termination codon is at position 238.

Additional data for each feature are shown in the indented lines that follow. For example, the first feature has a gene name of “1”, a protein product named “gp1”, a locus tag of “DNAM1”, and a note about where Glimmer called the start position.

The data contained in the Documentation are also viewable in the Features Table (see below).

4.4.2 Viewing features in the Feature Table

The Documentation that you viewed above has been automatically Parsed by DNA Master into the **Feature Table**. Click on the [Features] tab to view the Features Table (Figure 4.5).

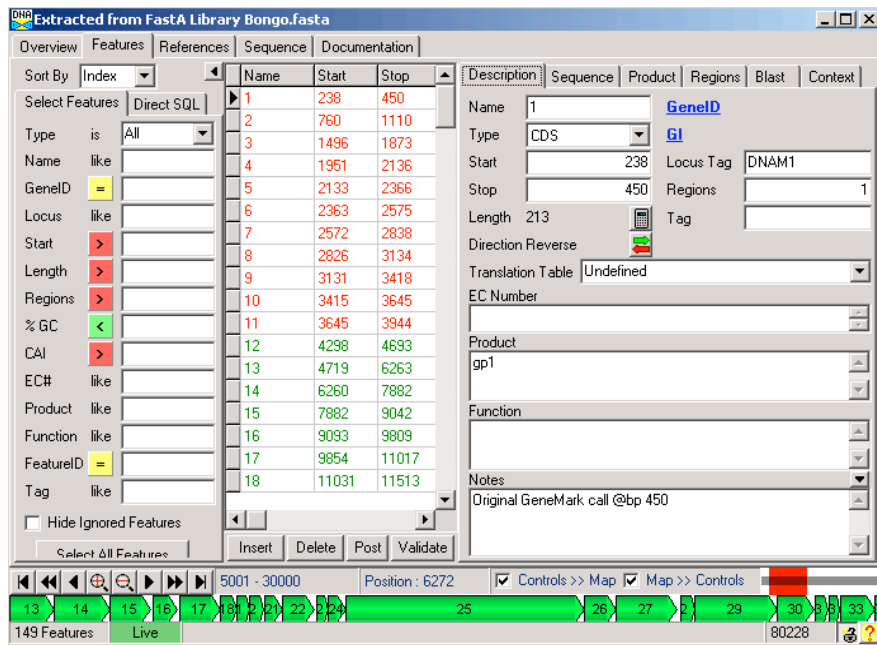


Figure 4.5

The central box shows each gene’s **Name**, **Start** and **Stop** coordinates, and—if you use the scroll bar to move to the right—the gene **Length**. You can select any gene by clicking on it. Gene “1” is selected in the example above, as indicated by the small black triangle next to it.

CRITICAL NOTE ABOUT A POTENTIALLY CONFUSING PROGRAM FEATURE

DNA Master **ALWAYS** lists the leftmost genomic coordinate as the **Start** position and the rightmost as the **Stop** position, regardless of a gene’s direction of transcription. This means that genes that are transcribed from right to left, and thus have start codons at their rightmost coordinate, will still have their rightmost coordinate in the “Stop” column. Don’t let this confuse you!

It is helpful to think of those column headings as “**Left**” and “**Right**” rather than Start and Stop.

If you look to the right, you will see six sub-tabs named **[Description]**, **[Sequence]**, **[Product]**, **[Regions]**, **[Blast]**, and **[Context]**.

The **[Description]** sub-tab is shown by default and contains basic information about the gene that you’ll recognize from the documentation, including gene name, coordinates, product name, and notes.

The **Notes** for gene 1, shown above, indicate that Glimmer called the start at position 450. There is no mention of GeneMark in these notes, which means that GeneMark’s gene call agreed with Glimmer’s gene call. If the two programs do not agree, this will be mentioned in the Notes as shown below.

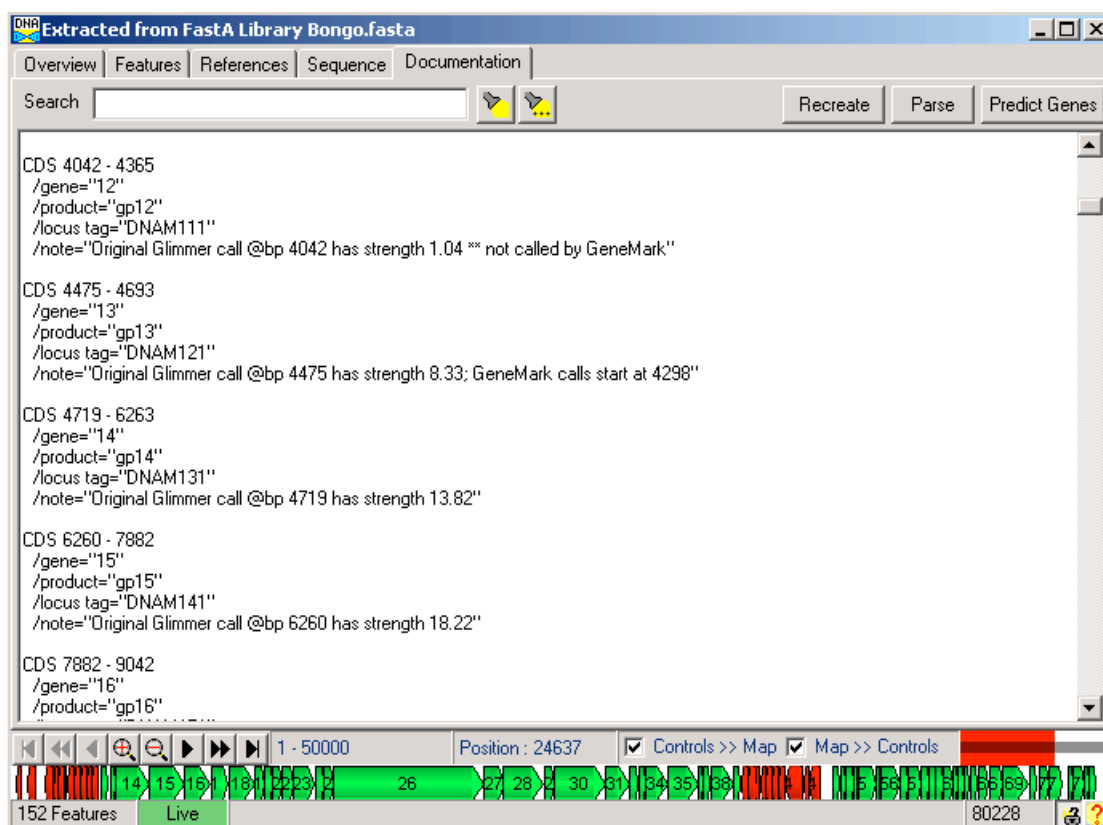


Figure 4.6

In the next example, gene 12 was predicted by Glimmer, but was “**not called by GeneMark**”.

For gene 13, the assigned start is 4475 as called by Glimmer, but there is a note that “**GeneMark calls start at 4298**”.

Your refinement of your annotation in **Section 8** will focus substantially on evaluating the predictions made by Glimmer and GeneMark. You will be resolving any ambiguities that have arisen and adding or deleting genes that were missed or errantly called by these programs.

You don’t need them just yet, but you can see that there are also buttons (at the bottom of the central box middle) that will let you either ‘**Insert**’ or ‘**Delete**’ features. And eventually the ‘**Validate**’ button will help you assess whether all your gene calls make sense.

4.4.3 Viewing the sequence in the Sequence tab

Click on the [Sequence] tab.

You will see the sequence appear as before, but now you can use the ‘**Feature**’ dropdown menu at the top left. When you click on this menu, a list appears that shows each gene and whether it is transcribed leftwards (R, for reverse) or rightwards (F for forward).

You can scroll down and select any of these and it will then select and highlight the corresponding part of the DNA sequence. This can be a very useful feature for examining specific parts of the genome.

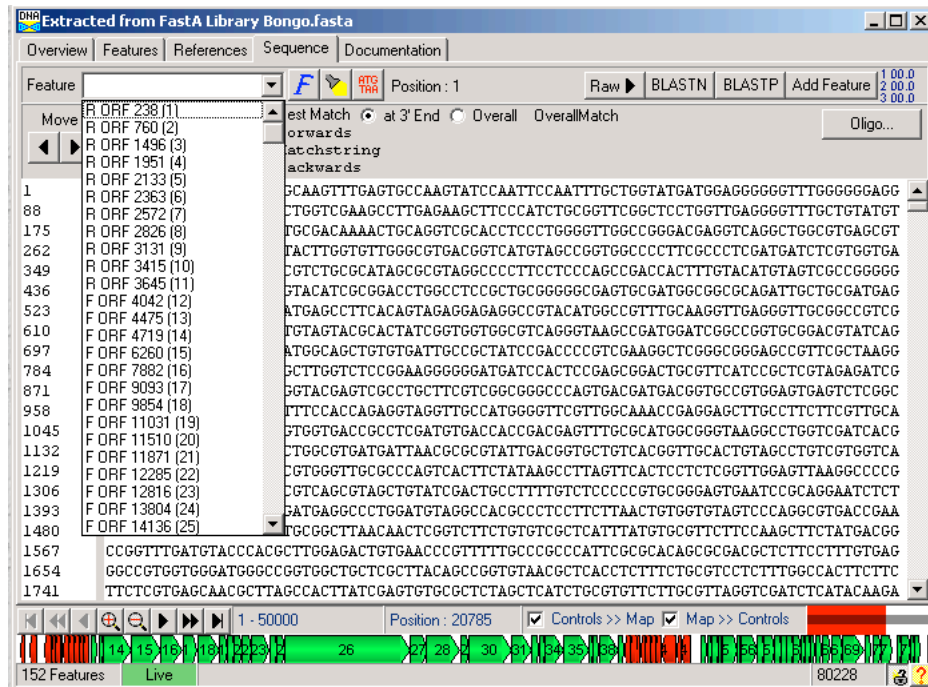


Figure 4.7

4.4.4 Viewing ORFs in the Frames window

The Frames window is an especially important one for determining and assessing start site choices. To open the Frames window (we use Angelica in the example below) select:

DNA → Frames

A window will open that has a graphical representation of the six possible reading frames, with each row representing one reading frame. Full-row-height vertical lines represent in-frame stop codons, and half-row-height vertical lines are possible start codons. At the lower left in the window is a box displaying the nucleotide coordinate corresponding to the position of your pointer as you mouse over the display. There are also buttons that allow you to scroll through your genome and zoom in and out.

At the lower right corner of the Frames window, there are six additional buttons. Click on the button labeled 'ORFs' (red circle in Figure 4.8).

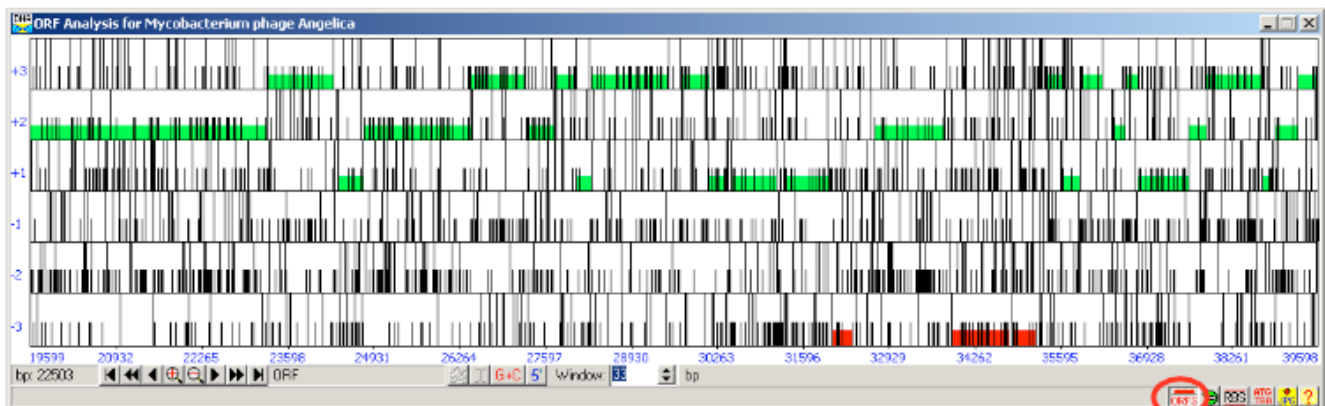


Figure 4.8

This will highlight all the features currently in your feature table as shown in the screenshot above. Genes in forward reading frames are green, those in reverse reading frames are red, and tRNAs are blue.

Next click on the frames window within the box that contains a highlighted gene.

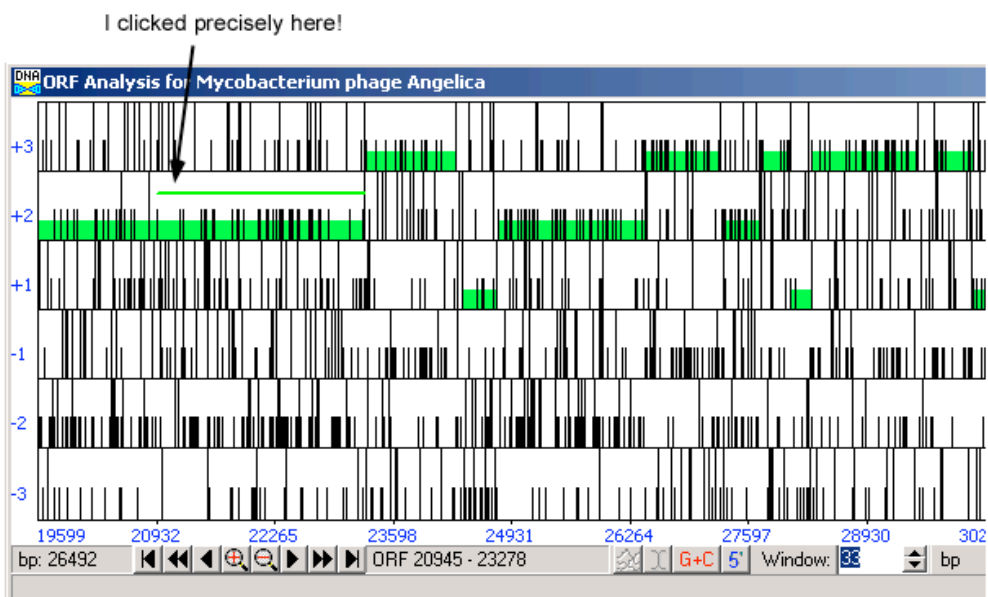


Figure 4.9

A thin, horizontal green line will appear that extends from the nearest upstream start codon to the next downstream stop codon.

Now click on the 'RBS' (ribosomal binding site) button in the bottom right corner of the Frames window.

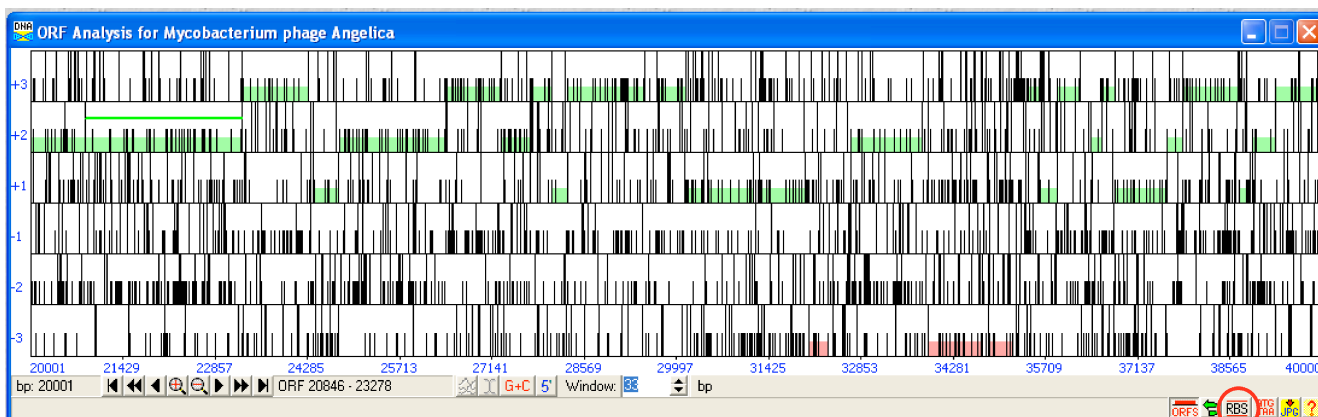


Figure 4.10

Another window titled "Choose ORF start" will appear, shown in Figure 4.11.

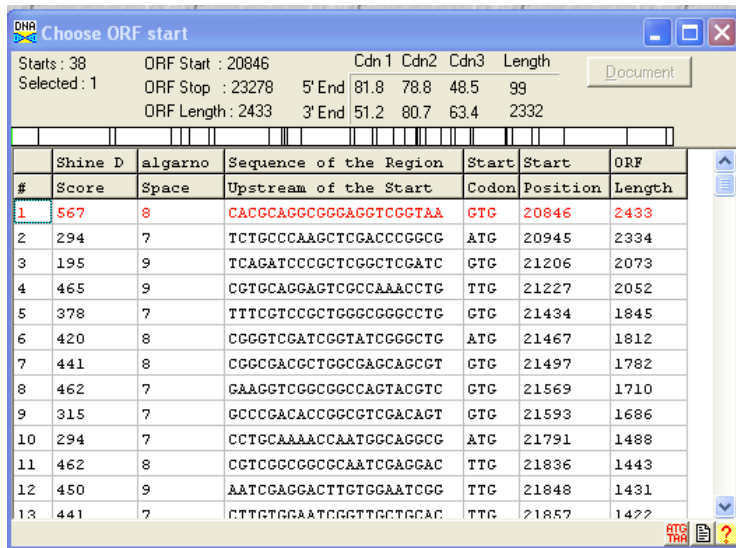


Figure 4.11

This window lists all of the possible start codons in the ORF you clicked on in the Frames window, the corresponding upstream nucleotide sequence, the gene length resulting from that start, and a score for the Shine-Dalgarno sequence (higher is better). One line's text may be red, and this is because that row corresponds to the start site immediately upstream of where you clicked in the Frames window.

When evaluating your gene calls and choosing between possible start sites, you may find it helpful to have all three windows open at once, as shown in Figure 4.12 for the Etude genome.



Figure 4.12

4.5 Running the BLAST function

When determining the settings for the automated annotation above, we cautioned about the time it takes to run the BLAST function and you may have elected to skip BLASTing. Sooner or later, however, you will need to do this. When you can allow an hour or so for DNA Master to run uninterrupted, you should run the BLAST function. To do so (we use phage Bongo below), go to:

Genome → BLAST All Genes

- In the dialog box, we recommend that you use the settings shown in **Figure 4.13**.

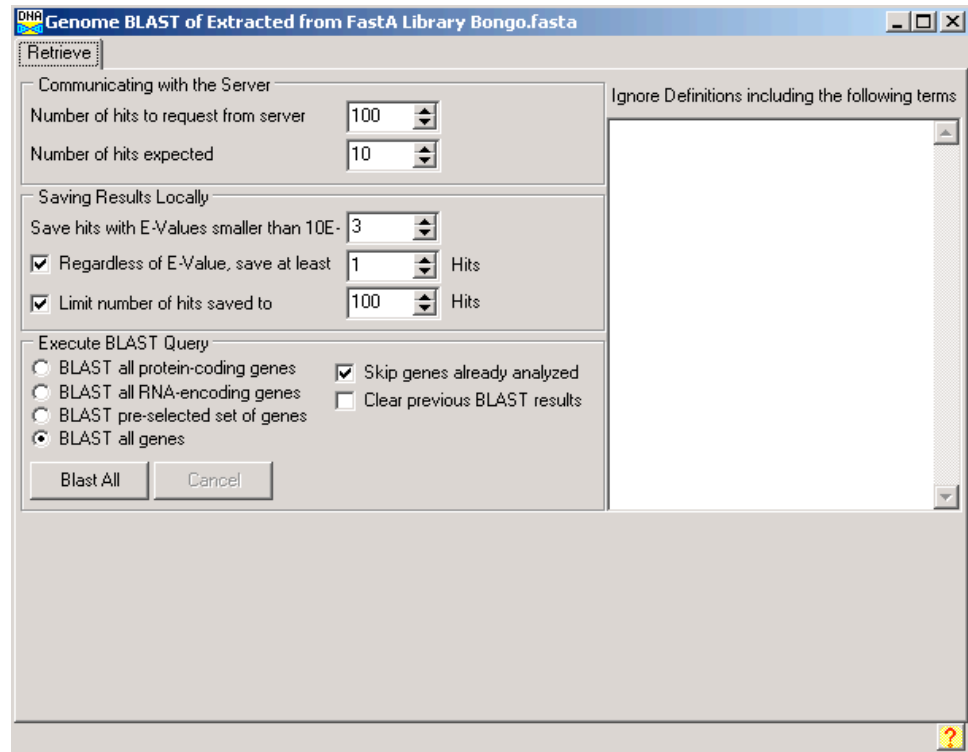


Figure 4.13

- Click on '**Blast All**'.
- DNA Master will send the predicted protein sequences in your file in batches to the NCBI server, then retrieve the results and store them. Be patient during this process! Windows may briefly indicate that DNA Master is "Not Responding" during this period, but that's because it's processing!

Even though you still only have a draft annotation that was generated automatically, it is very helpful to do the BLAST search **before** finalizing gene calls, because the data will be extremely helpful during the process of annotation refinement.

When all BLAST searches are complete, DNA Master will report "**Genome BLAST has been completed**" as shown in **Figure 4.14**.

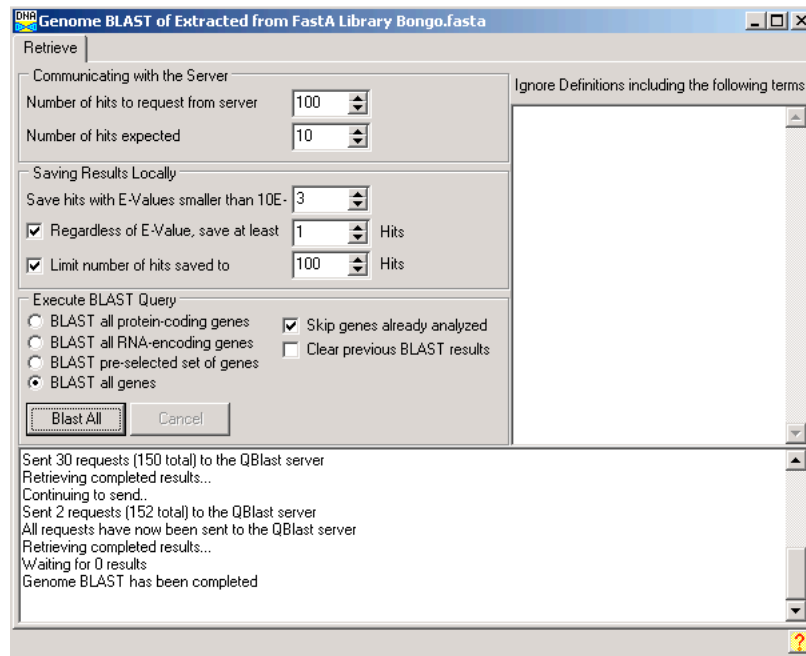


Figure 4.14

- You may now close this BLAST window.
- You can now view BLAST results for any gene by returning to the [Feature] tab and selecting a gene, then clicking on the [[Blast]] sub-tab to the right.

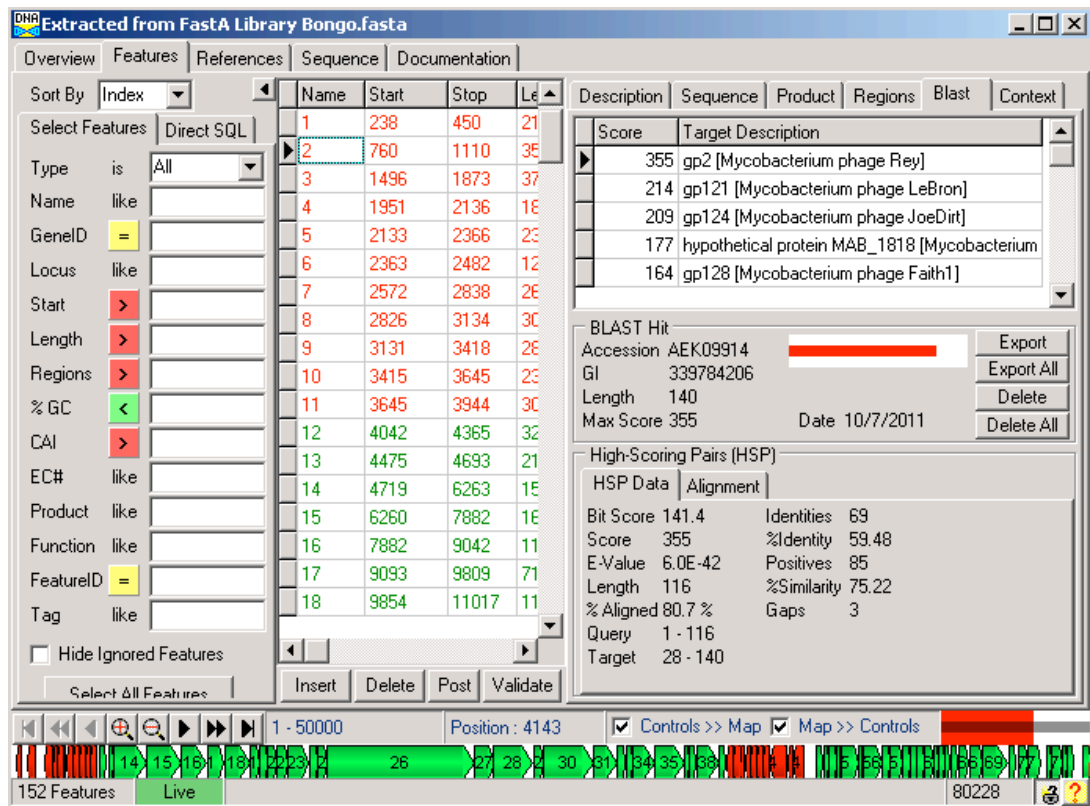


Figure 4.15

In the example above we clicked on gene 2. Under the **[[Blast]]** sub-tab, you can see a window with the BLAST hits listed, with a score and a description. Below that is a pictorial report on the extent of the match (shown as a red bar depicting the part of the gene product – i.e. gp2 in this case – that matches the selected subject). Below that are the data for the hit (HSP Data), and if you click on the **[[Alignment]]** sub-sub-tab it will show the actual alignment.

In the example shown in **Figure 4.16**, we clicked on the second BLAST hit and then clicked on the **[[Alignment]]** sub-sub-tab. Note that you can now see the amino-acid matches in the bottom right pane.

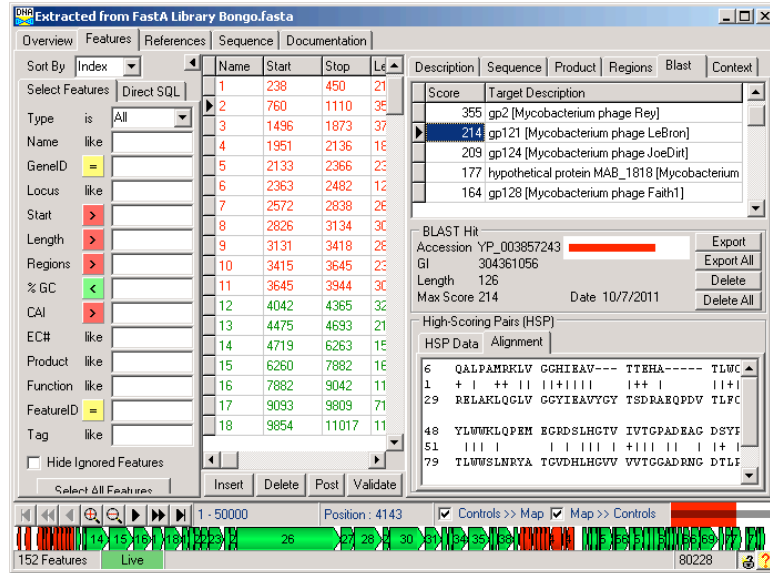


Figure 4.16

- Save your file as described in **Section 4.3** to ensure your BLAST data are stored.

4.6 Re-opening an archived (saved) file

When you save files, Opening archived (saved) files is straightforward. Go to:

File → Open → Archived DNA Master file

- Browse to your saved .dnam5 file and select and open it.

5 Gathering additional information for refining your annotation

There are three additional pieces of data that we recommend gathering at this point. The first is a **six-frame translation** of your sequence labeled with your predicted genes. The second is a **provisional genome map**. The third is a **graphical output of the GeneMark-Smeg analysis**. Depending on your genome, you may also need the **tRNA predictions** from the web-based Aragorn and tRNAscan-SE algorithms. The output of these programs will be used in **Section 8**.

5.1 Generating a six-frame translation

With your genome open in DNA Master (we used Etude below), go to:

Genome → Six-frame translation

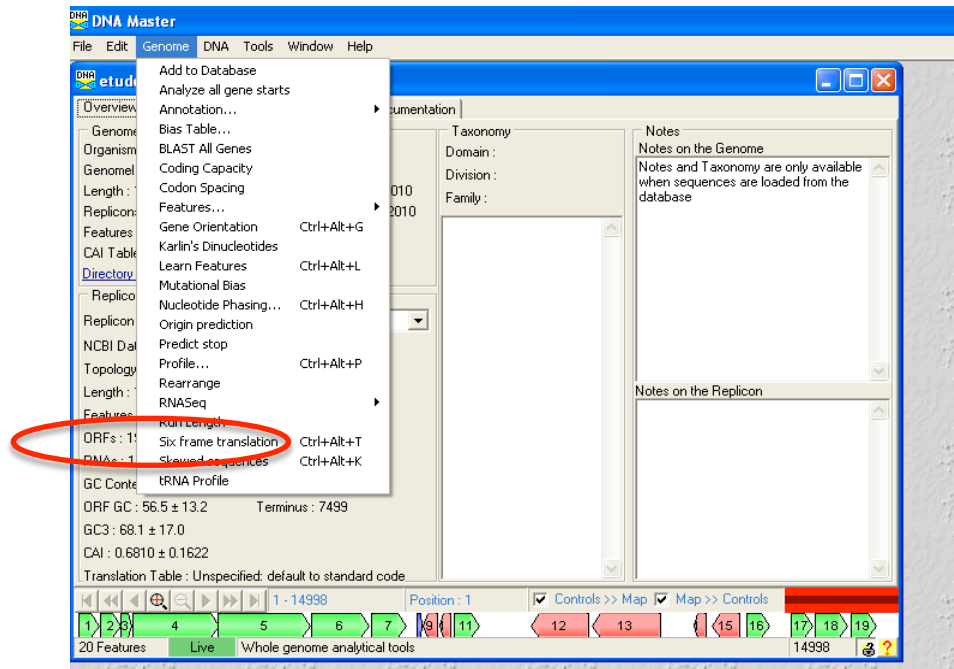


Figure 5.1

The six-frame translation window will open.

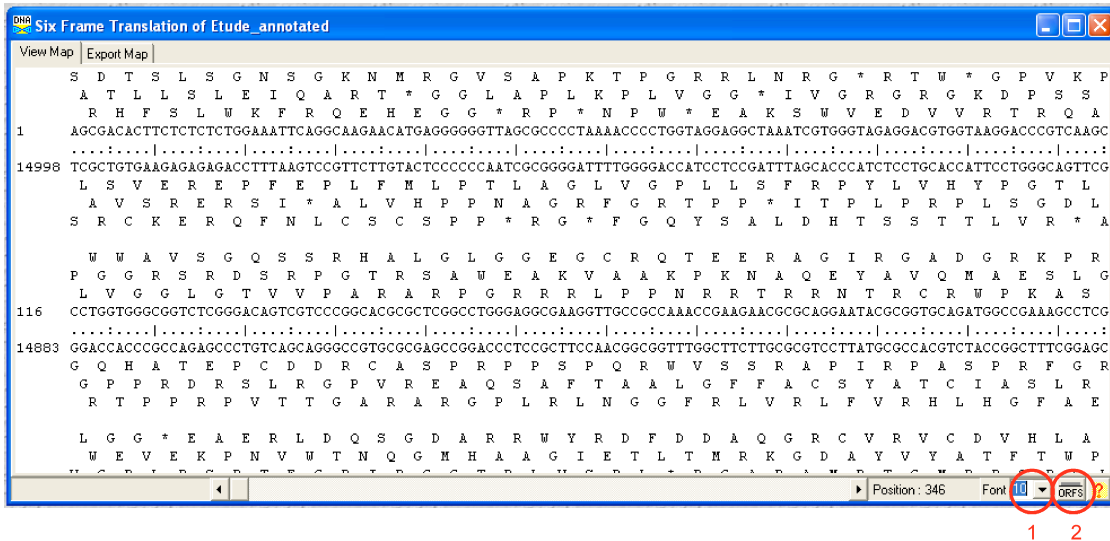


Figure 5.2

- Adjust the size of the font by entering '8' in red circle #1 in Figure 5.2.
- Click on the ORFs button in the red circle #2 in Figure 5.2.

Note that the ORFs predicted in your auto-annotation are now highlighted. Also note that this window scrolls right and left rather than up and down. When you first click on the ORFs button you may not see highlighted text if there is no gene predicted in the extreme left end of your genome (which is what is shown by default). If you like, you can scroll to the right using the scroll bar at the bottom to see more sequence.

But you can also be assured that your selection has been chosen because the ORFs button at the bottom right is now shown in red (see Figure 5.3).

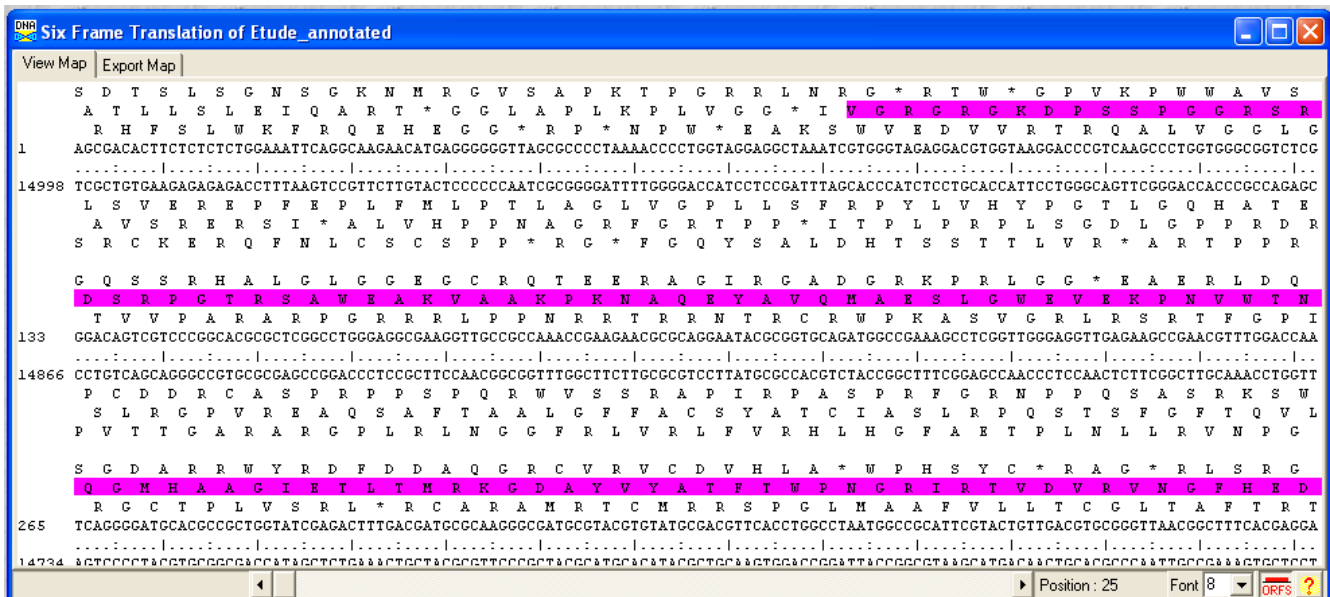


Figure 5.3

Now click on the [Export Map] tab at the top left of this window. We recommend using the default settings as shown in Figure 5.4 below.

5.2 Generating a provisional genome map in DNA Master

Another useful tool in DNA Master is the ability to make a genome map. This map is not comparative (though you will make a comparative map using Phamerator in the next section), but rather just a separate file of the map shown at the bottom of the sequence panel. Still, it is a useful way to see your gene calls in the context of the entire genome.

To make a genome map (we use mycobacteriophage Timshel below), go to:

DNA → Export Map

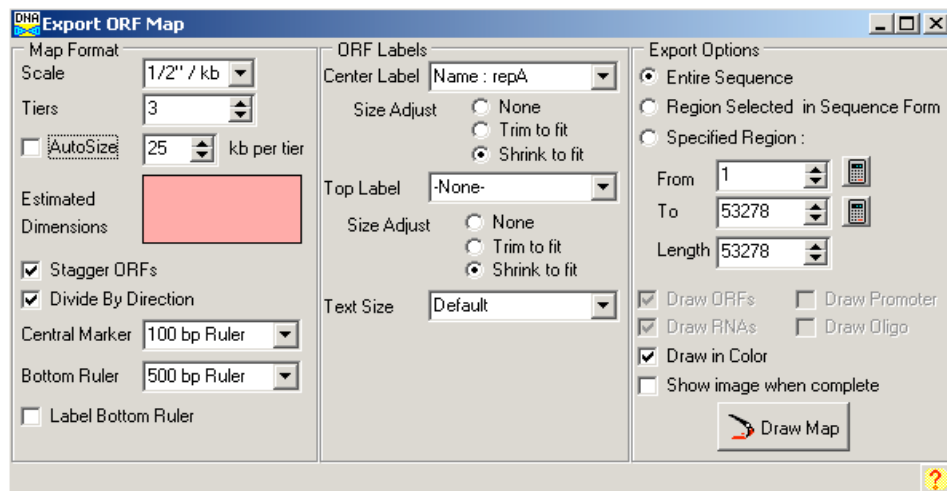


Figure 5.6

- In the dialog box that appears, many options are available. We recommend you use the settings shown in **Figure 5.6**, except that the 'Tiers' field may need to be adjusted. Three or four tiers are acceptable for a genome of up to about 60 - 80 kb in length. If your genome is larger, increase the number of tiers accordingly.
- Click on 'Draw Map'.
- Choose a filename and location to save to, then click 'Save'.

The file will be saved as YourFileName.wmf (Windows metafile). This file can be opened by Preview (on a Mac), Paint, Canvas, or similar drawing programs. Depending on the program, you can manipulate this file in numerous ways. At the very least, you should see a graphical illustration of your genome, similar to one shown in **Figure 5.7**.

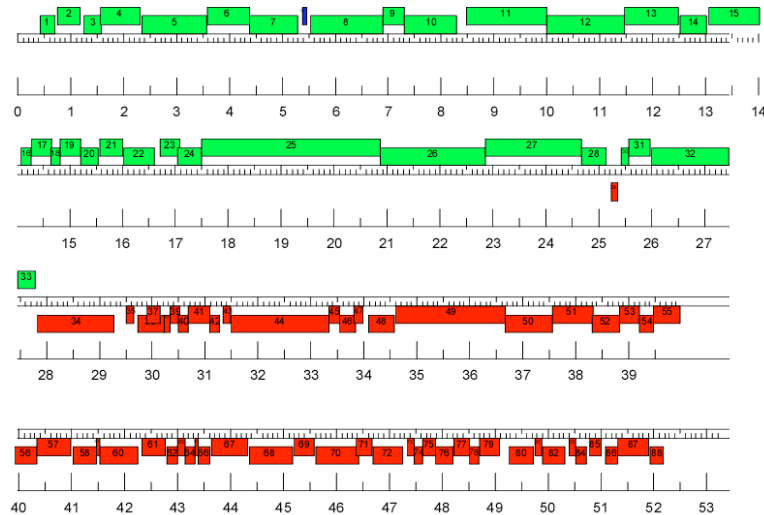


Figure 5.7

5.3 Generating a graphical output from GeneMark

As we noted above, GeneMark is a gene prediction program, and the version embedded in DNA Master runs heuristically, using parts of the genome you enter to train the program to identify coding potential. When using the stand-alone version on the web, you can:

1. Use an existing coding model to predict the genes.
2. Generate a graphical output.

The host profile we recommend using is that of *Mycobacterium smegmatis*, assuming that you used this host to isolate your phage. If you used a different host, you will obviously need to select a different bacterial profile for GeneMark.

To run web-based GeneMark (we use mycobacteriophage Bongo below), go to:

- http://opal.biology.gatech.edu/GeneMark/genemark_prok_gms_plus.cgi Also found on the Links page of <http://phagesdb.org>
- Select '**Browse**', then find and select your sequence file. This is the same YourPhage.fasta file that you imported into DNA Master.
- Enter your phage's name in the '**Title**' box.
- From the '**Species**' dropdown box, select '*Mycobacterium_smegmatis*' (assuming you are annotating a mycobacteriophage genome).
- Maintain the default option of *E. coli* as the RBS model (there is no other).
- Maintain the default options for Window size, Step size, and Threshold.
- In the '**Graphical output options**' section, check each box in the first column except '**Generate PostScript graphics (email)**' and '**Mark putative exon splice sites**'. You do **not** need to enter an email address.
- Uncheck all boxes under the '**Text output options**' heading.
- Click on the '**Start GeneMark**' button at the bottom left.

GeneMark Version 2.5 [\(Reload this page\)](#)

Reference: Borodovsky M. and McIninch J. GeneMark: parallel gene recognition for both DNA strands, *Computers & Chemistry*, 1993, Vol. 17, No. 19, pp. 123-133. [\[Download PDF \]](#)

Prediction models ready for a total of [265](#) completely sequenced prokaryotic genomes in NCBI RefSeq database. Pre-calculated prediction [database](#) for these genomes

Input Sequence

Title (optional):
Bongo

Sequence:

Sequence File upload:
/Users/welkin/Documents/Mycophages/SEA phages/2010-2011/ [Browse...](#)

Running Options

Species: Mycobacterium_tuberculosis_H37Rv
RBS model: E.coli
Window size: 96 bp
Step size: 12 bp
Threshold: 0.5 %

Use alternate genetic code:
 Eukaryote (e.g. Yeast, ATG = only start)
 Mycoplasma (TGA = Tryptophan)

Output Options

Graphical output options

- Generate PDF graphics (screen)
- Generate PostScript graphics (email)
- Mark orfs on graph
- Mark regions on graph
- Mark stop codons on graph
- Mark start codons on graph
- Mark frameshifts on graph
- Mark putative exon splice sites
- Print graph in landscape format

Email address (required for PostScript email output):

Text output options

- List open reading frames (ORFs) predicted as coding sequences (CDSs)
- List regions of interest
- List putative eukaryotic splice sites
- Write protein translations of ORFs
- Write nucleotide transcripts of ORFs
- Write protein translations of regions
- Write nucleotide transcripts of regions
- Write protein translations of putative exons
- Write nucleotide transcripts of putative exons

Run

[Start GeneMark](#) [Default](#)

Please send any suggestions for improvements or problems to the web page [maintainer](#).

Figure 5.8

Once GeneMark has run, a new window will appear and in the middle it will have a heading “**Result of last submittal**”, as shown in Figure 5.9.

- Click on the link ‘**View PDF Graphical Output**’ just below.

Result of last submittal

GeneMark Results

[View PDF Graphical Output](#)

```
Sequence: Bongo
Sequence length: 80228
GC Content: 61.62%
Window length: 96
Window step: 12
Threshold value: 0.500
---
Matrix: Mycobacterium tuberculosis H37Rv, Thu Oct 27 16:10:50 2005
Matrix author: Dr. Borodovsky Laboratory, School of Biology, Georgia Tech
Matrix order: 5
```

Figure 5.9

- Save and open the pdf.

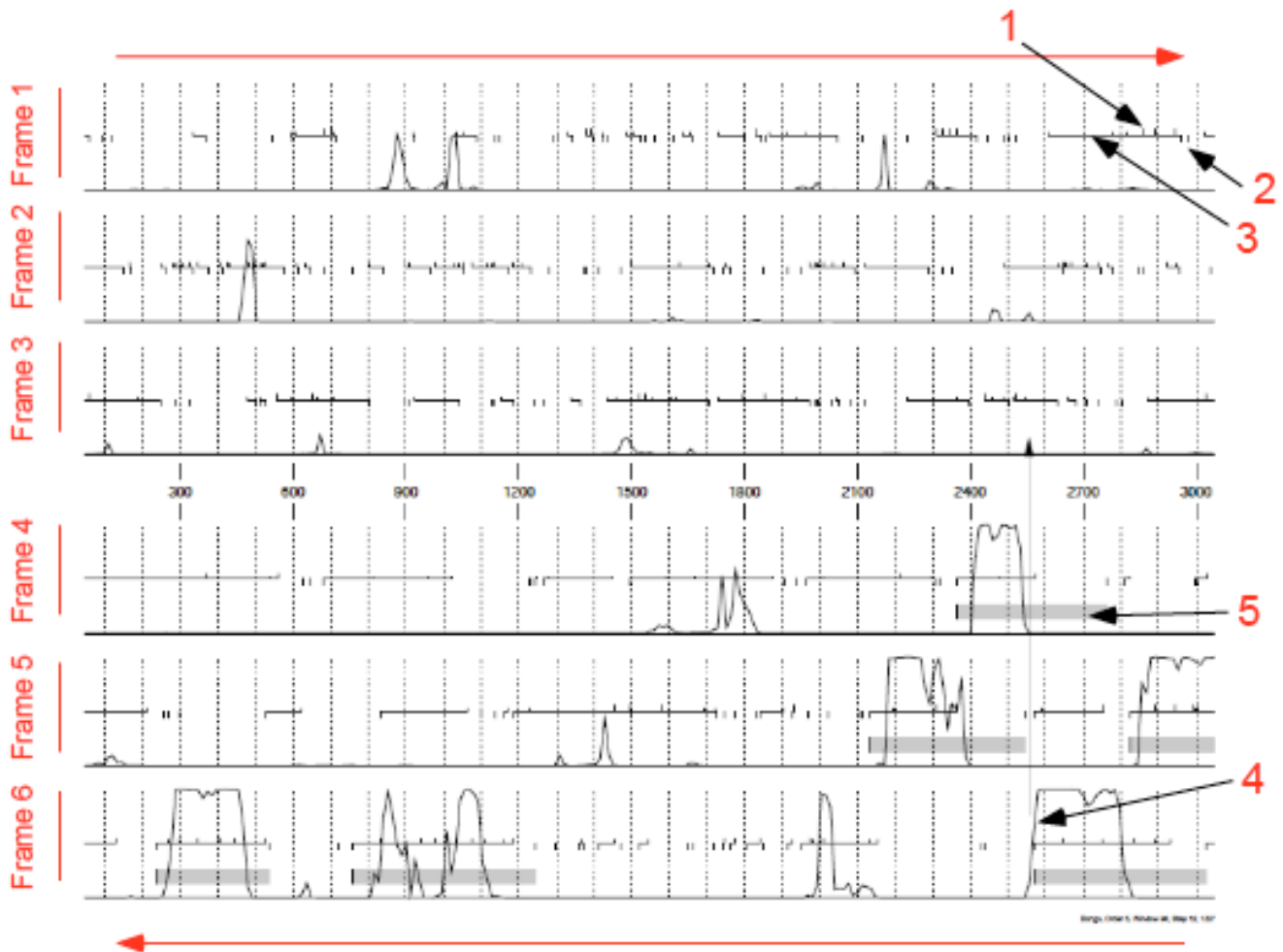


Figure 5.10

We recommend that you **print** this file because it is a good place to make notes as you refine your annotation. Below, several features of this output are described.

- ✓ All six frames are represented and are separated from one another by solid horizontal lines.
- ✓ The top three frames are in the forward orientation; the bottom three in the reverse orientation.
- ✓ In each frame, the start codons are shown as small upward facing ticks (#1 in figure).
- ✓ In each frame, the stop codons are shown as small downward facing ticks (#2).
- ✓ The horizontal lines in the middle of each row represent open reading frames (ORFs) (#3).
- ✓ A graphical representation of coding potential is shown (#4).
- ✓ The shaded areas (#5) signify regions that GeneMark predicts as likely coding regions, based on coding potential and positioning of stop codons, but for the most part is of limited utility in gene identification.

6 Using Phamerator to assist with annotation

6.1 Overview

Phamerator is a Linux-based program that compares phage genomes, their genes, and their gene products, and then displays the results of these comparisons in a variety of useful ways. Phamerator is comprised of two basic parts: an underlying database that contains the results of the comparisons, and a graphical interface to that database.

One of Phamerator's key features is that it groups gene products into "**Phamilies**" (generally referred to as "**Phams**") when the pairwise alignment scores (using BLASTP and ClustalW) are above a defined threshold.

Phams are thus groups of proteins with a high degree of similarity to one another, though there is one caveat to be aware of. If protein A is similar to protein B and protein B is similar to protein C, all three will be grouped into the same Pham, even if proteins A and C are not above the threshold scores when compared directly. This can be very useful in identifying proteins with multiple domains that may be fused in one phage genome and split in another.

Phamerator is especially useful for generating and comparing genome maps of multiple phages through the visual interface that displays whole genome nucleotide and protein sequence relationships, as well as the conserved domains within genes.

For more on Phamerator and its mechanics, see the following paper.

Cresawn SG, Bogel M, Day N, Jacobs-Sera D, Hendrix RW, Hatfull GF. "Phamerator: a bioinformatic tool for comparative bacteriophage genomics." *BMC Bioinformatics*. 2011 Oct 12; 12(1):395.

6.2 Why Phamerator is useful to you at this stage of your annotation

Phamerator maps provide an easy-to-understand representation of how your genome compares to similar genomes. This is useful during annotation because it draws attention to places where your automated annotation diverges from the finalized annotation of a closely related (and often GenBank-published) genome. It also provides a genome-wide perspective and thus a context for the annotation refinement, functional analysis, and other explorations to follow.

6.3 How did my genome get into Phamerator already?

In order expedite your annotation workflow, we have taken each newly sequenced genome, generated an automated annotation (just as you did in **Section 4**), and entered all of these files into a Phamerator database that contains all sequenced mycobacteriophages. The database generated is called 'Mycobacteriophage_Draft' because it contains auto-annotated draft genomes along with finalized and published annotations. The auto-annotated genome names are given the suffix "_Draft," so as to distinguish them from the GenBank-quality files. At a later time, when you've refined your annotation and it is submitted to GenBank, your draft annotation may be replaced in Phamerator with your final annotation.

6.4 Making Phamerator maps

- Open the Phamerator program. (Allow up to a minute for the main window to appear, as Phamerator will check for new databases when it boots.)
- Click on 'Phages' in the left 'Sources' pane.
- The name of the current database will be displayed at the top of the window (red oval in diagram below). Make sure the database is "Mycobacteriophage_Draft". If not, go to **Edit** → **Preferences** and select Mycobacteriophage_Draft from the Database dropdown menu.

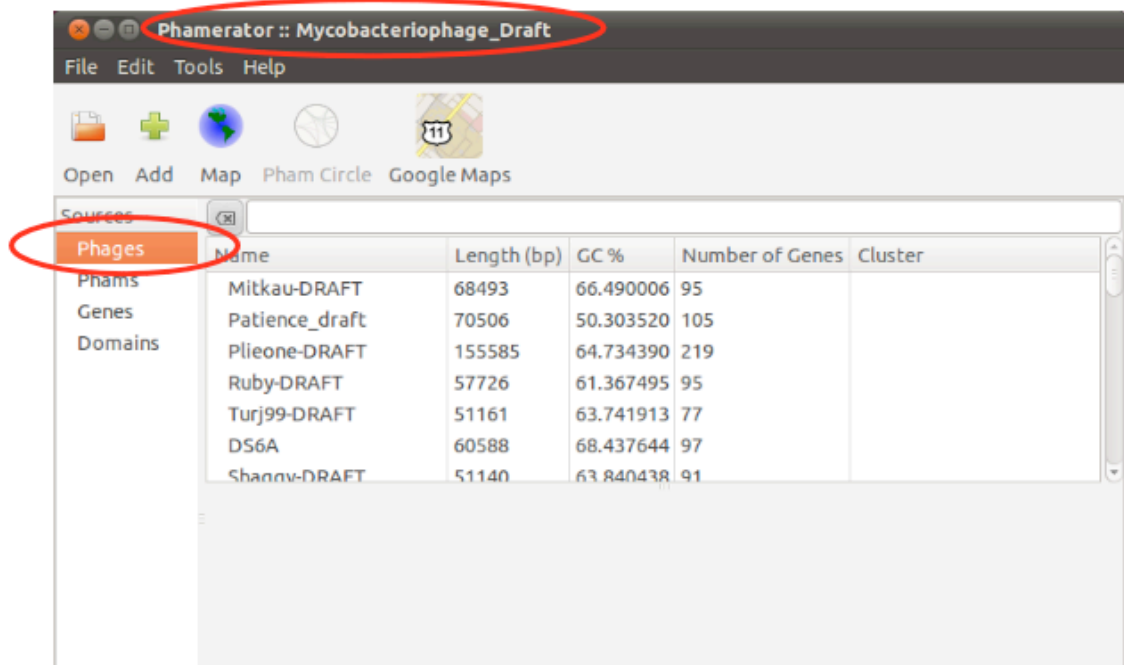


Figure 6.1

You can now choose genomes you want to compare to one another. We recommend:

- Your phage
- Some closely related phages (in the same cluster or subcluster)

You should decide carefully which genomes you want to compare. For example you may not want to compare all of the genomes from a particular cluster if there are a large number. If your phage belongs in a cluster with several different subclusters, you may want to use a representative of each subcluster.

A good rule of thumb is to shoot for no more than about six genomes to start with. You can always return to this and generate more maps as you need them.

- Scroll through the list—or use the search bar—to find your phage.
- Click on it to select it. It will be highlighted.
- To add additional genomes to your selection, scroll through to find the genome you want (if you used the search function, make sure you clear all search terms so that you can see all of the genomes).

- Use Ctrl-click (or equivalent if using an emulator—on Macs it is often Ctrl-Shift-click) to add another genome to your selection. You can also select consecutive genomes in the list by using Shift-click.
- Repeat to select as many genomes as you want to include.
- The phages can also be sorted by simply clicking on the column headers—such as Cluster, Length, GC%—to help find relevant genomes.

In **Figure 6.2**, four genomes are currently selected, indicated by the orange highlight.

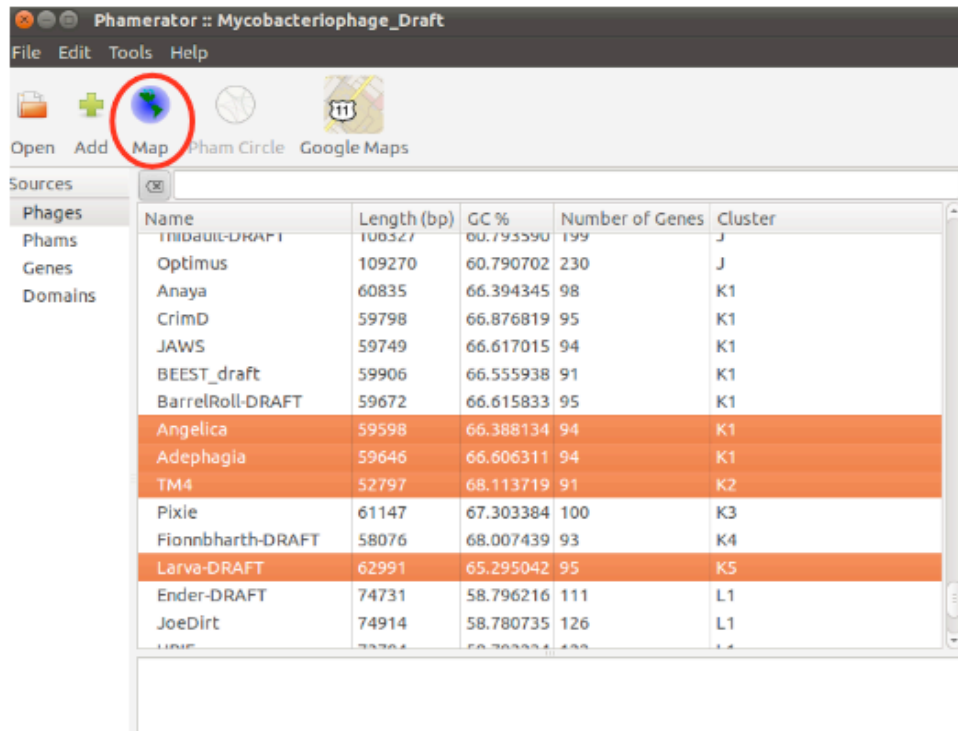


Figure 6.2

- Once you've finished selecting genomes, click on the button that says 'Map' (red circle in **Figure 6.2**). Be patient, as it can take a minute (or more for a large number of genomes) to generate the map.
- When the map window appears, you will see something like this:



Figure 6.3

Congratulations! You've made a Phamerator map using your phage's draft annotation.

6.5 Understanding and using the genome maps made by Phamerator

When the **Genome Map** window appears, you will probably only be able to see a small portion of the genomes. You can resize the window to see more, but you probably won't be able to see the entire picture unless you change the zoom factor. A sample is shown in **Figure 6.4**.

- To see a view of your entire genome, click the '**Zoom Out**' icon at the top left repeatedly until you can see the genome ends.



Figure 6.4

Each genome is represented as a hash-marked horizontal bar. Forward-transcribed genes are shown as rectangles above the bar, and reverse-transcribed genes as rectangles below the bar. Each gene is colored according to the **Pham** to which it belongs, making it easy to see relatives in other genomes.

You may have noticed that some genes appear to have smaller yellow boxes within them. These represent matches to the NCBI Conserved Domain Database. These will be particularly useful later when attempting to determine gene functions, but they can be confusing at this stage. Fortunately, Phamerator makes it easy to toggle the display of these domains. Just go to:

View → Show Domains, then click to unselect this option.

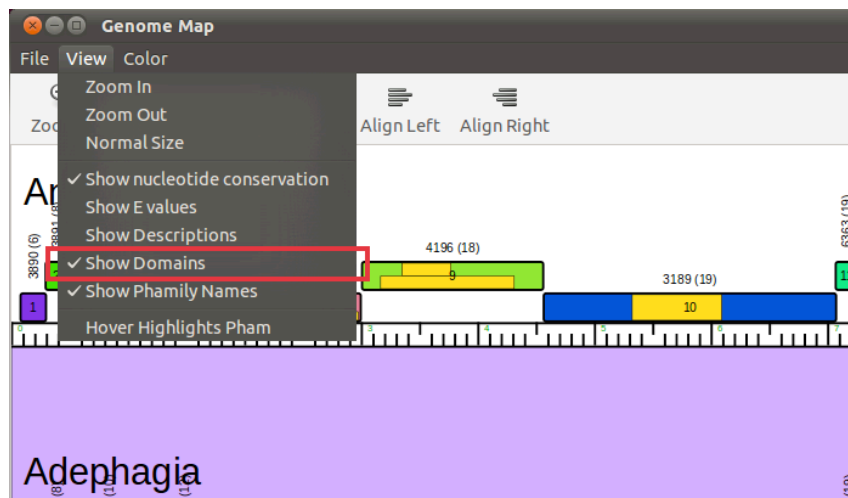


Figure 6.5

Lots of information is displayed on Phamerator maps.

- Click the 'Zoom In' icon several times to get a closer look.

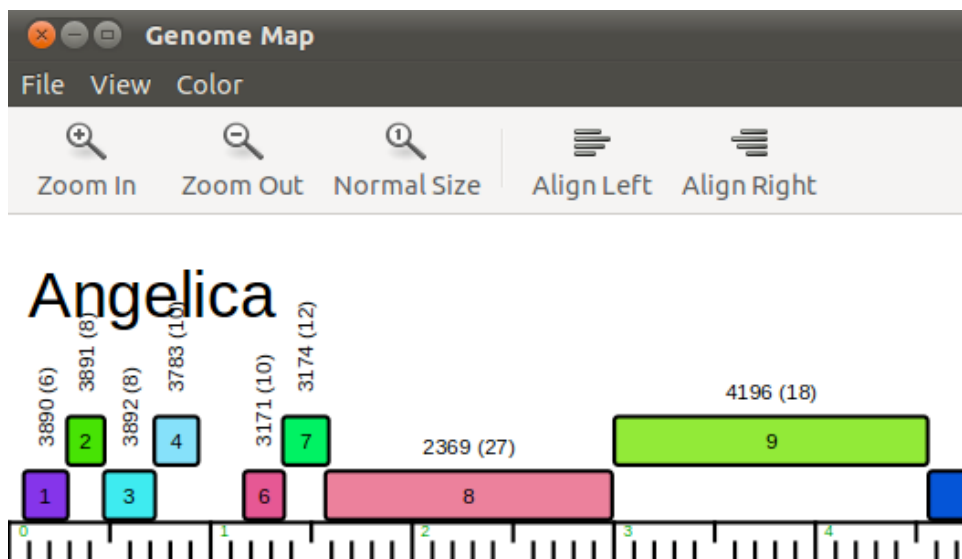


Figure 6.6

Again, the white bar at the bottom represents the genome sequence itself, and is marked with green numbers every 1,000 base pairs (bp). The small hash marks coming up from the bottom show 100 bp intervals, while the ones coming down from the top show 500 bp intervals.

Each gene's box has a number within it that represents that gene's number in this genome. There are also two numbers above each gene; the first is the number of the Pham this gene belongs to, and the second—in parentheses—is the total number of members of that Pham.

Putting all this together, we can determine that Angelica's gene 8 begins at ~1600 bp, ends at ~3000 bp, is a member of Pham 2369, and that there are 26 other members in that Pham:

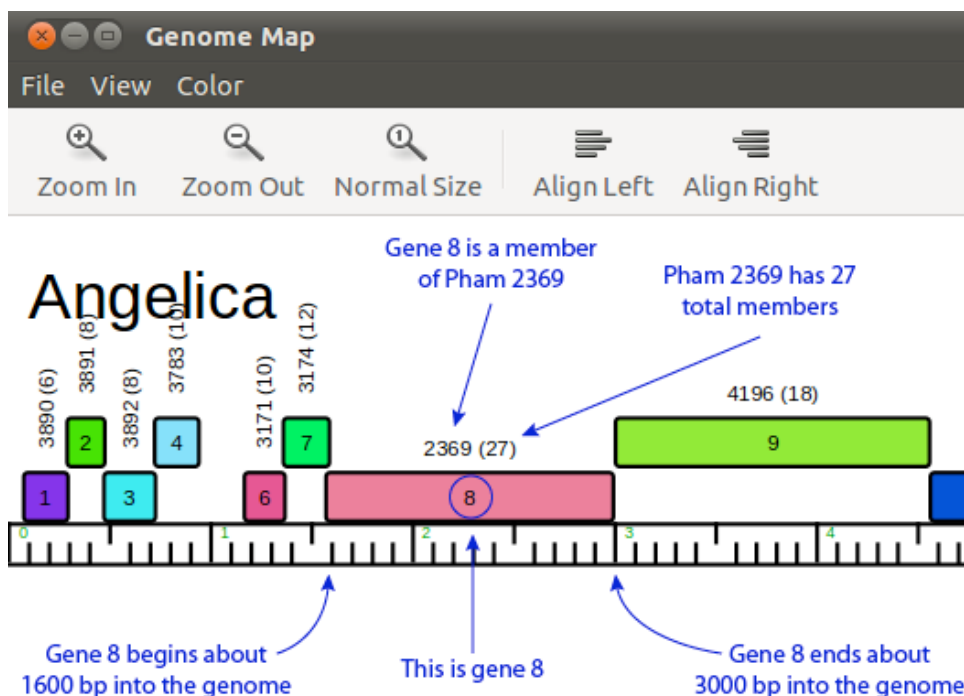


Figure 6.7

6.6 Viewing nucleotide sequence similarities in Phamerator

A NOTE ON TWO DIFFERENT TYPES OF SIMILARITY

Nucleotide sequence similarity is a comparison of the **DNA sequence** (A, C, G, T) of two **genomes**. It is often determined by running BLASTN. On Phamerator maps, nucleotide similarity is shown by colored vertical boxes between genomes.

Protein similarity is a comparison of the **amino acid sequence** of two **proteins**. It is often determined by BLASTP or ClustalW. On Phamerator maps, protein similarity is shown by similarly colored gene boxes.

Phamilies, or **Phams**, are determined based on **protein similarity and NOT nucleotide similarity**.

Don't confuse these two types of similarity, or you may misinterpret the data that Phamerator is showing!

While Phamerator was conceived to compare protein sequences to other protein sequences, it can also show nucleotide sequence similarity between genomes. To enable this function:

View → Show nucleotide conservation should be checked (as in **Figure 6.8**).

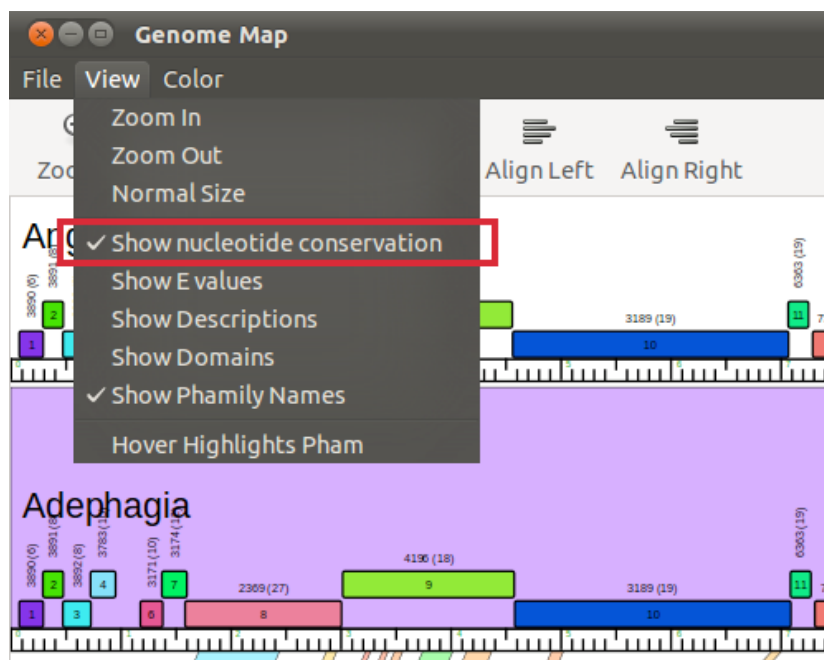


Figure 6.8

Once you've turned on 'Show nucleotide conservation', you may see colors between the genomes on your map, as shown in **Figure 6.9**.

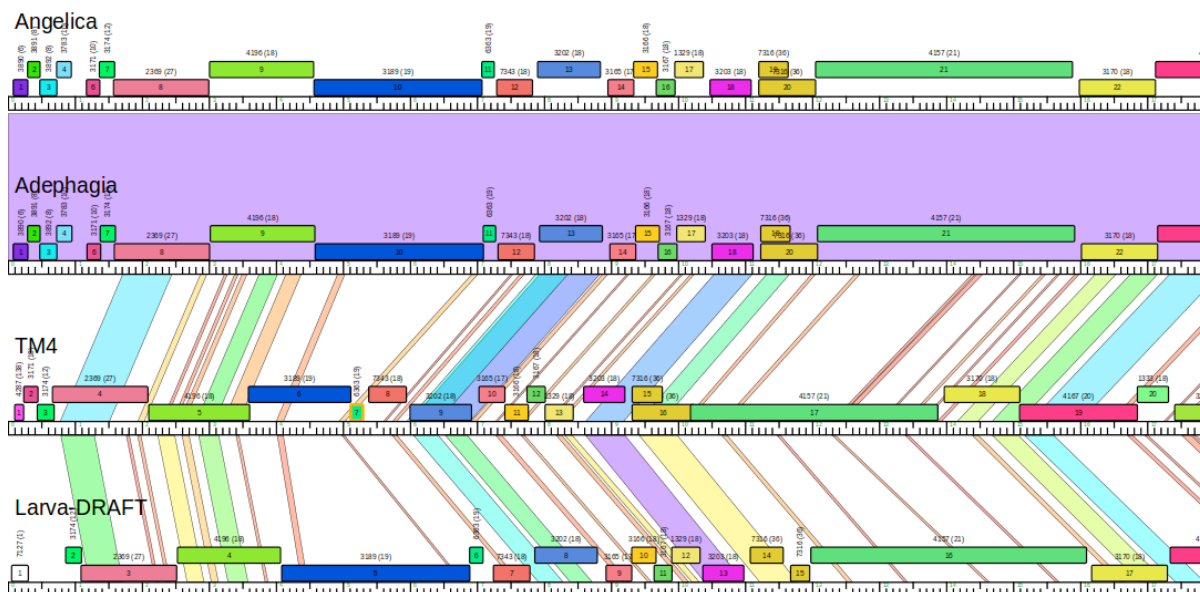


Figure 6.9

Nucleotide sequence similarity is shown by the (often slanted) shaded regions (boxes) **between genomes**. Each box represents one BLASTN alignment, and is colored based on its E value,

with violet representing the best matches (lowest E values) and red the worst matches (highest E values). White areas indicate that there is **no** nucleotide similarity in those regions.

Looking at the screenshot above, it is apparent that the top two phages (Adephagia and Angelica) have widespread nucleotide similarity to one another, as indicated by the solid purple between the two genome maps. The other two phages shown (TM4 and Larva) have multiple regions of nucleotide similarity, though these areas are interrupted by dissimilar (white) areas and have higher E values. This segmented similarity is a reflection of what you saw in the BLAST searches performed earlier. The top two genomes are members of Subcluster K1, while the bottom two are members of other subclusters within Cluster K.

Phamerator-generated maps can be extremely helpful when trying to evaluate a gene start codon in your novel genome that (for example) produces a bigger gene than in the compared genomes. A quick look at the Phamerator-generated map lets you know that the upstream sequence does or does not have sequence similarity.

6.7 Other Phamerator features

There are many other functions in Phamerator. Several examples are below.

1. Click on the colored portion of any gene's box to select it, and the nucleotide and amino acid sequences of that gene are shown in the bottom panels.
2. You can move the order of genomes around in the display. This is important, because the nucleotide similarities are only displayed by comparing two adjacent genomes in the display. To do this, click and hold on the **NAME** of a phage you want to move (it is on the extreme left, and you may need to scroll over to it), then drag the genome either up or down to where you want it and release it.
3. You can move a genome to the left or right to better compare it to its neighbors. To do this, Ctrl-Click-hold on the **NAME** of the phage (on a Mac, this might be Ctrl-Shift-Click-hold), then drag to the left or right and release.
4. You can also align genes from multiple genomes, such as those within a particular Pham. For example, you may have noticed that gene 13 in Adephagia is in the same Pham as gene 9 in TM4. Select gene 13 from Adephagia, then Ctrl-click to select gene 9 from TM4, and verify that both genes are highlighted. Then press the "Align Left" or "Align Right" button at the top of the genome map.
5. You may want to also explore the '**Hover Highlights Pham**' function, available in the **View** menu.

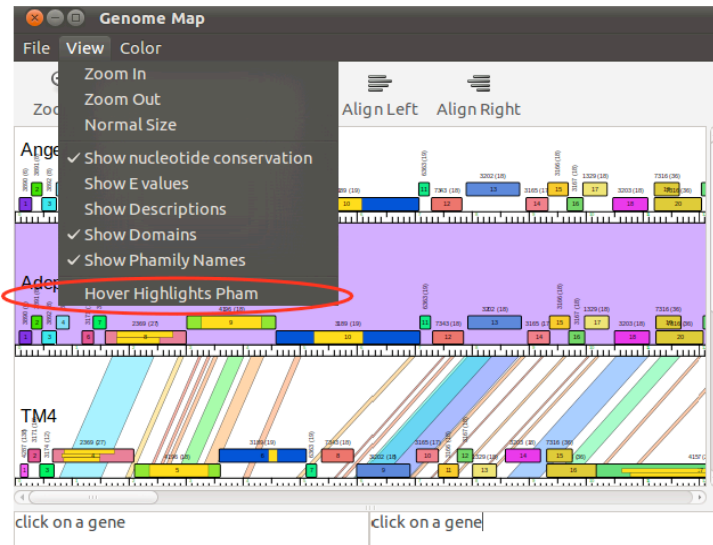


Figure 6.10

This function's use is that when your mouse hovers over any gene, only the gene members of that particular Pham are shown in color, while all others go white. This is a very useful function for easily seeing gene conservation or loss in different genomes.

6.8 Saving Phamerator maps

Finally, if you would like to save the map as a file, from the Genome Map window go to:

- **File** → **Save As**
- Enter a name and select your desired file type (pdf files are a good choice).
- Click '**Save**'.

7 Guiding Principles of Bacteriophage Genome Annotation

7.1 Overview

Though the automated annotation you have created using DNA Master will usually identify more than 80% of genes correctly, some genes will need to be manually added, modified, or deleted. Therefore, all gene calls must be reviewed to identify those that must be changed. In this section, we provide a set of principles that should guide you as you evaluate and improve upon your draft annotation.

It is helpful to think of the process of evaluating your draft annotation's gene calls as an application of these principles: together they will help you make the best possible gene predictions. It is essential to understand that any annotation consists of making a **prediction** as to how the genetic information is organized and used. In the absence of experimental evidence to support a given gene call, there is no right or wrong answer; there are, however, well-supported or ill-supported predictions.

As with any set of principles, the ones presented here will conflict with one another at times. It's your job to weigh one against another and make the best gene calls possible.

Because of the importance of these principles, this section is dedicated wholly to presenting them. Read them carefully before beginning an annotation, and keep them nearby as you work.

7.2 The Guiding Principles

The following two pages list the principles themselves. As mentioned above, we recommend that you print those two pages, read them carefully, and keep them close at hand as you refine your gene calls.

Because these are principles, and not unbreakable rules, you'll see words like "usually," "generally," and "typically" used quite frequently. Remember that phages are famous for finding exceptions to "rules", so very little is truly set in stone.

GUIDING PRINCIPLES OF BACTERIOPHAGE GENOME ANNOTATION

1. In any segment of DNA, typically only one frame in one strand is used for a protein-coding gene. That is, each double-stranded segment of DNA is generally part of only one gene.
2. Genes do not often overlap by more than a few bp, although up to about 30 bp is legitimate.
3. The gene density in phage genomes is very high, so genes tend to be tightly packed. Thus, there are typically not large non-coding gaps between genes.
4. If there are two genes transcribed in opposite directions whose start sites are near one another, there typically has to be space between them for transcription promoters in both directions. This usually requires at least a 50 bp gap.
5. Protein-coding genes are generally at least 120 bp (40 codons) long. There are a small number of exceptions. Genes below about 200 bp require careful examination.
6. Protein-coding genes should have coding potential predicted by *either* Glimmer, GeneMark, or GeneMark TB. Start sites are chosen to include areas of strong coding potential.
7. Switches in gene orientation (from forward to reverse, or vice versa) are relatively rare. In other words, it is common to find groups of genes transcribed in the same direction.
8. Each protein-coding gene ends with a stop codon (TAG, TGA, or TAA).
9. Each protein-coding gene starts with an initiation codon, ATG, GTG, or TTG. But note that TTG is used rarely (about 7% of all genes). ATG and GTG are used at almost equivalent frequencies.

CONTINUED...

GUIDING PRINCIPLES OF BACTERIOPHAGE GENOME ANNOTATION

...CONTINUED

10. An important task is choosing between different possible translation initiation (i.e., start) codons. The correct start site can often be distinguished by association with a credible ribosome-binding site (RBS; Shine-Dalgarno (SD) sequence). Identifying the correct start site, however, is not always easy and is predicated on the following sub-principles:
 - a. The preferred start site usually has one of the higher SD scores of all the potential start codons, but not necessarily the highest.
 - b. Manual inspection can be helpful to distinguish between possible start sites. The consensus is as follows: **AAGGAGG – 3-12 bp – start codon.**
 - c. The relationship to the closest upstream gene is important. Usually, there is neither a large gap nor a large overlap (i.e., more than about 4 bp). A short overlap of 1-4 bp—where the start codon overlaps the stop codon of the upstream gene—is very common.
 - d. The position of the start site is often conserved among homologues of genes. Therefore, the start site of a gene in your phage is likely to be in the same position as those in related genes in other genomes. But be aware that one or more previously annotated and published genes could be suboptimal, and you may have the opportunity to help change it to a more optimal one.
 - e. Your final start-site selection will likely represent a compromise of these sub-principles. For example:
 - i. A start codon that overlaps the stop codon of a previous gene trumps a somewhat lower score.
 - ii. A higher SD score or canonical RBS trumps a more extended gene overlap.
 - iii. If choosing between several starts with similar SD scores, it is usually best to choose the one that gives the longest open reading frame.
11. tRNA genes are not called precisely in the program embedded in DNA Master, and require extra attention. (Please refer to **Section 9.5.**)

8 Gene by gene: evaluating and improving your draft annotation

8.1 Overview

This section describes the heart of the matter: how to go through a draft annotation, one gene at a time, and decide whether or not the automated annotation called the gene correctly. You will spend most of your annotation time in this section, because you'll need to follow the steps here between 50 and 250 times per genome, once per gene!

If you've been following this guide step-by-step, you probably have all the items listed below ready to use. If you've jumped directly to this step, you may want to gather the items listed below to assist you as you go.

1. Your draft annotation file (from **Section 4**) open in DNA Master. (It is helpful to have DNA Master's Frames window open as well, with the windows arranged as shown as the last figure in **Section 4.4.4**.)
2. A printout of the Guiding Principles of Bacteriophage Annotation (**Section 7.2**).
3. Phamerator running, preferably with a map displaying your genome and related genomes (**Section 6**).
4. A printout of your GeneMark-Smeg output (**Section 5.3**).
5. (Optional) A printout of your DNA Master-generated map (**Section 5.2**).
6. (Optional) A printed six-frame translation of your sequence (**Section 5.1**).

One useful configuration is to have a pair of annotators work together on a genome, using two computers, one with DNA Master running, and the other with Phamerator.

8.2 *Button-pushing mechanics reserved for Section 9*

The goal of this section is to help you **decide** what modifications need to be made to your draft annotation. In order to keep this section manageable and streamlined, we've moved the detailed **mechanics** (button-pushing) of many of these operations to **Section 9** of this guide.

Section 9 should be used more as an à-la-carte reference than as a step-by-step guide. For example, you probably won't need to read **Section 9.4.1** about properly annotating a programmed translational frameshift until you come across one during your annotation review.

8.3 *Decision Tree for evaluating the draft annotation*

To help clarify how to use Sections 8 through 12 of this guide, a decision tree is shown in **Figure 8.1**. There are three beginning tracks depending on what feature of your genome you're currently investigating: one for **Protein-Coding Genes** (**Section 8.4**), one for **Gaps in the Annotation** (**Section 8.5**), and one for **tRNA Genes** (**Section 8.6**).

Blue boxes are **decision** points, most of which are covered in the rest of **Section 8**. To answer the question in each decision box you'll need to keep in mind the Guiding Principles described in **Section 7** of this guide as well as the rest of the information in this section.

Purple boxes are **action** points where you implement the changes you've decided on. These actions are described in detail in parts of **Section 9**.

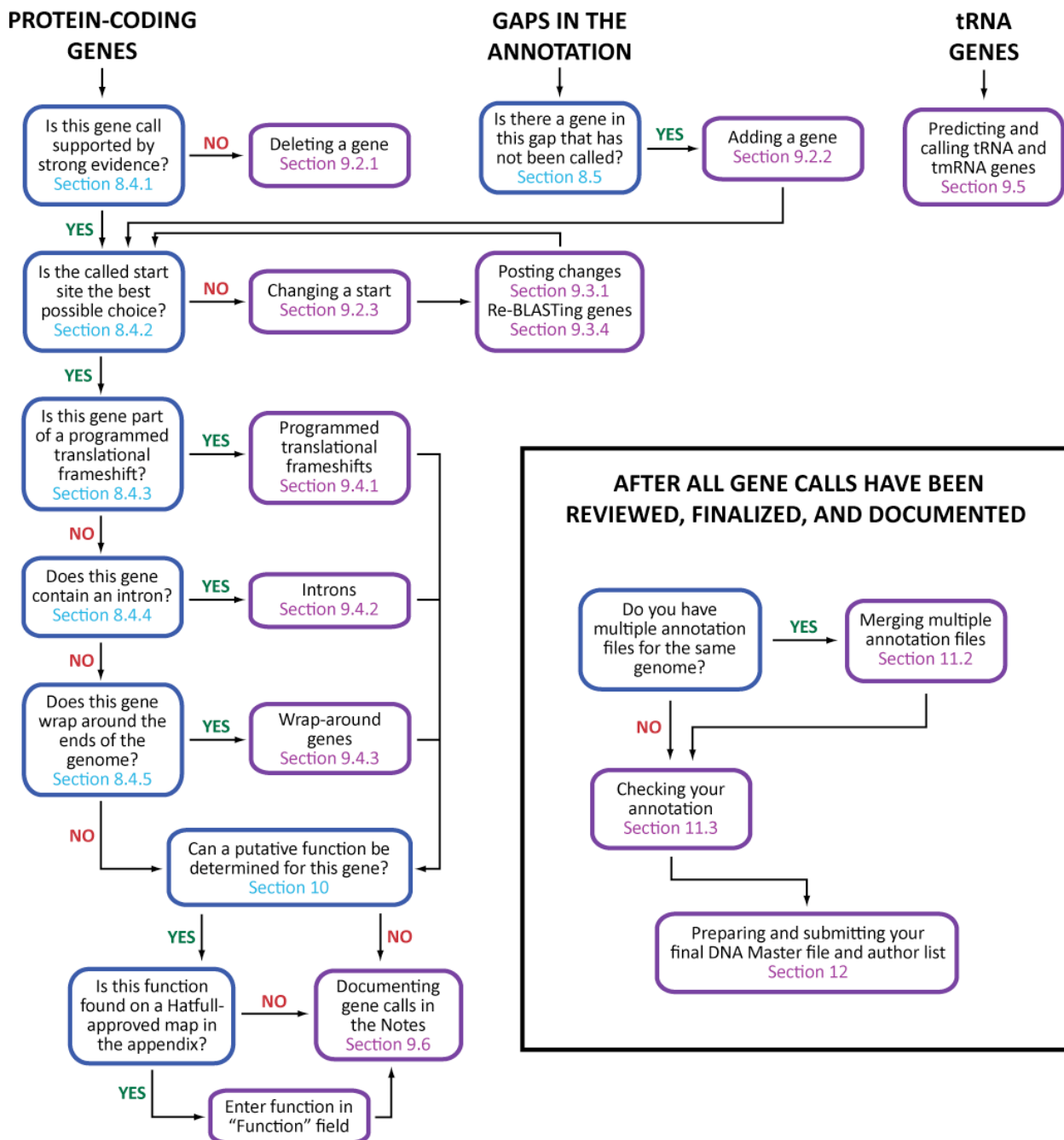


Figure 8.1

8.4 Evaluating protein-coding gene calls

The vast majority of features you will need to investigate are protein-coding genes, so you will use this section extensively. The first few genes you review will probably take some time as you become quite familiar with the process, but as you gain experience things will move faster.

It is best to start with your first open reading frame, which will typically be called gene '1' in the DNA Master feature table.

In evaluating the veracity of the prediction of this gene that was performed automatically by Glimmer and GeneMark, there are several questions you should ask, described in the following sub-sections. We'll use a sample gene, but you can proceed with your genome from here on.

It is also recommended that—in accordance with good lab practice—you keep notes of your thoughts and decisions as you proceed. You'll use them to enter your final Notes (Section 9.6).

8.4.1 Is the designation of this ORF as a gene well-supported?

- If it's not already selected, click on the **[Features]** tab.
- In the central column, click on the gene in question to select it. A small black triangle will appear to the left of that gene, indicating that it is active.
- Look at the "Notes" field under the **[Description]** sub-tab.

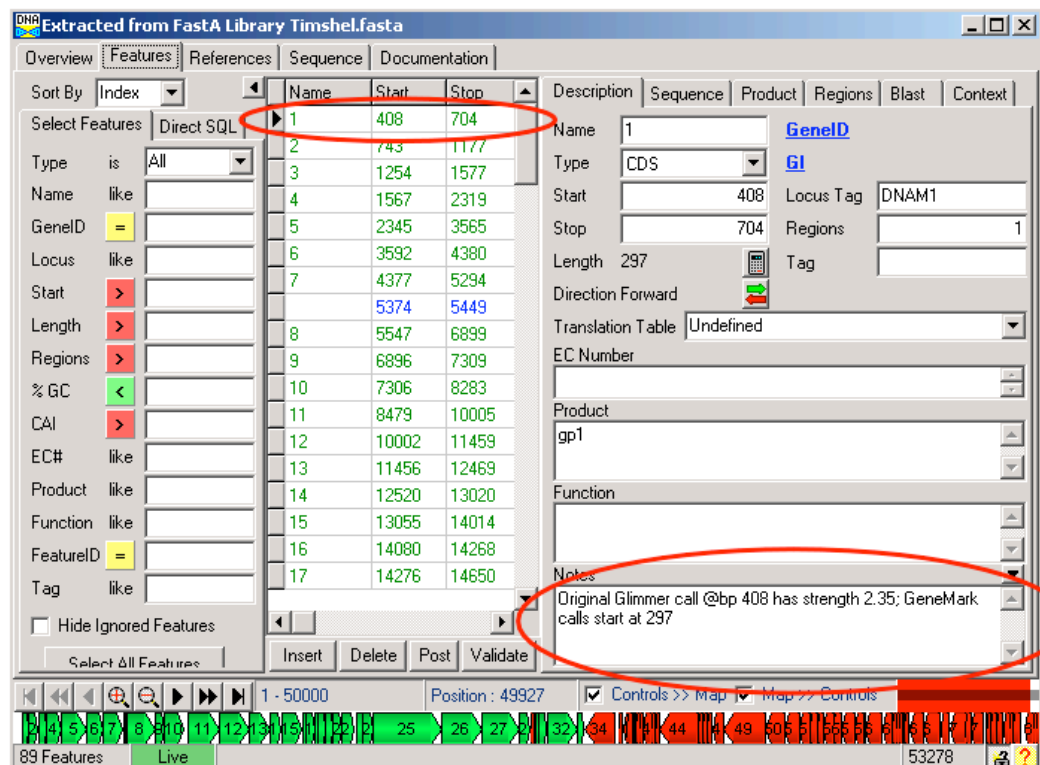


Figure 8.2

The notes should report whether Glimmer and/or GeneMark made the prediction. In the example above, both Glimmer and GeneMark did predict the gene, although the predicted starts sites are different. (Remember that if both programs agree, only one program's output

is reported.) The gene was called by both programs, which supports its legitimacy. Good so far.

- Find this gene in your GeneMark-Smeg output, and check if there is coding potential that supports this gene call.

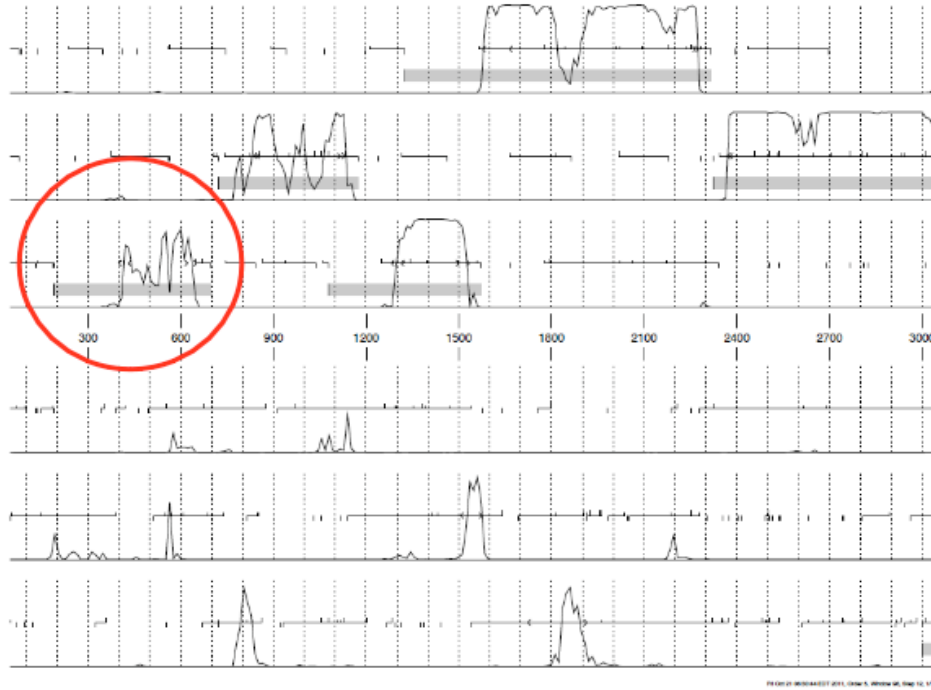


Figure 8.3

In **Figure 8.3**, the region of gene 1 is circled. You can find a gene by looking at its coordinates in the Feature Table, then finding those coordinates on the GeneMark output. It appears that this ORF has coding potential starting near position 400 and ending near 650. This is further evidence that there is a real protein-coding gene here.

- Examine the BLAST data under the **[[Blast]]** sub-tab, and see if there are genes in the databases that are high-quality matches to this one.

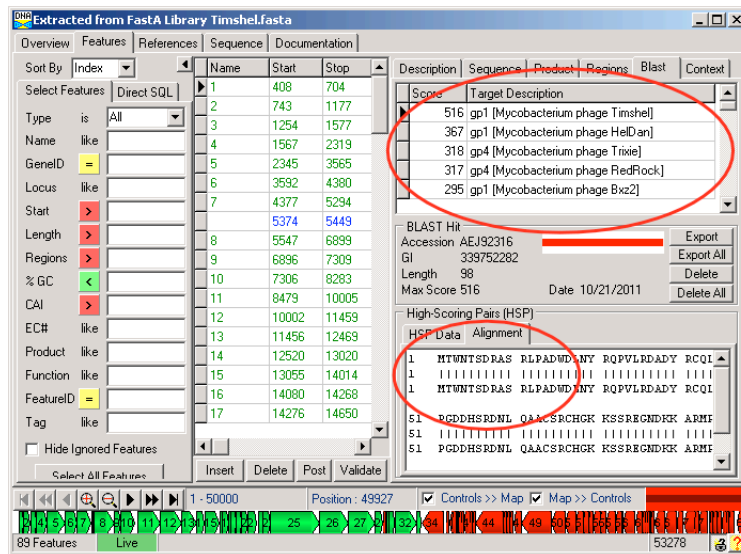


Figure 8.4

In **Figure 8.4** above, you can see that there are several good matches, which further supports this gene call's legitimacy.

- Review gene length to make sure it meets the expected parameters. You will recall (see **Section 7.2**) that you should carefully examine genes less than 200 bp in length with an eye towards gauging their legitimacy, and genes below 120 bp should be viewed very skeptically.
- You can check gene length by using the scroll bar to move to the right in the central column of the Feature table (see **Figure 8.5**), or you can select your gene and the length will be listed under the **[Description]** sub-tab to the right (see **Figure 8.5**).

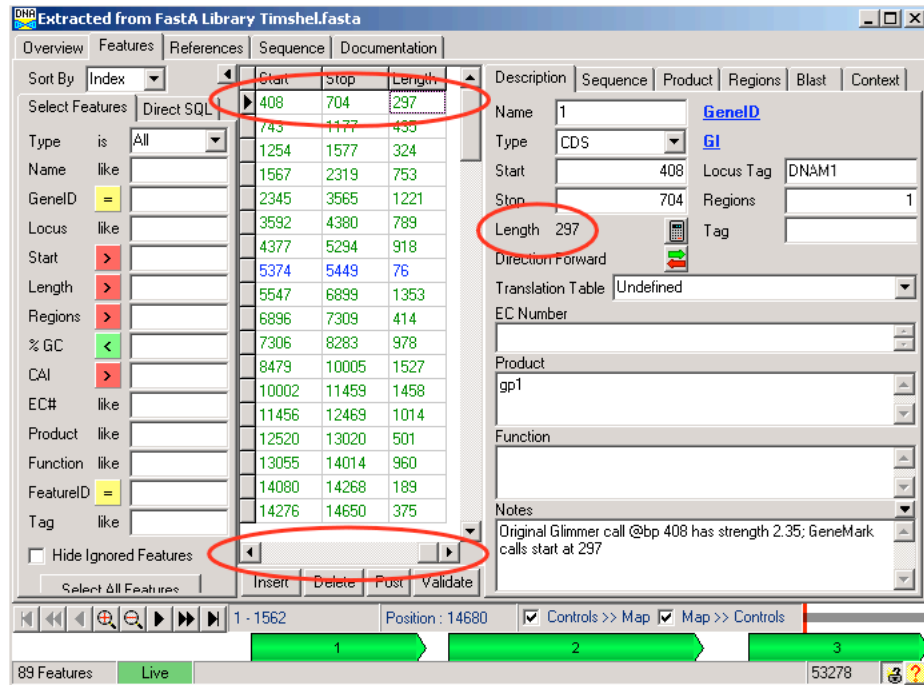


Figure 8.5

In this case, the gene length (297 bp) is not a concern nor requires special attention.

- Verify that there is only one gene called in this region of DNA, as per Guiding Principle #1. The easiest way to do this is by viewing either the Phamerator map or DNA Master map you've generated to see if there are other genes called that substantially overlap this one on either strand.

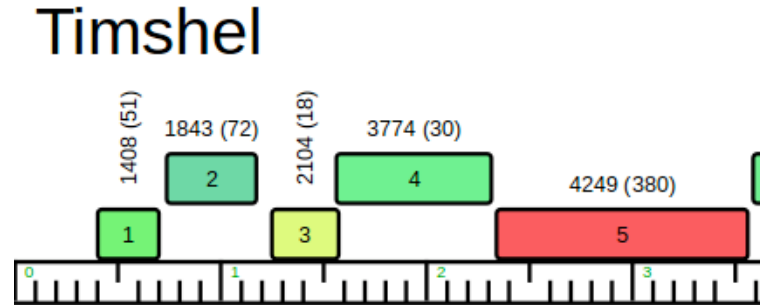


Figure 8.6

In this example above, we can see from the Phamerator map that there are no other genes occupying the same portion of DNA. Good.

DECISION TIME: Is the designation of this ORF as a gene well-supported?	
GUIDANCE: Most gene calls will pass this stage. Exceptions are genes that are called by only one program, have little or no coding potential, have very weak or no BLAST matches, are too short, and/or substantially overlap other genes.	
YES	NO
ACTION: Continue to Section 8.4.2 .	ACTION: You need to delete this gene. Go to Section 9.2.1 for instructions.

8.4.2 Is the called start site for this gene the best possible choice?

This can be a tricky, but the simplest way to answer is to address the following questions.

- **Does the currently predicted start site include all of the coding potential in the GeneMark output?** The current start position for our example is in location **A** in **Figure 8.7** below, and captures all of the coding potential. A hypothetical start at position **B**, however, would be a poor choice because it excludes about half of the area with coding potential.

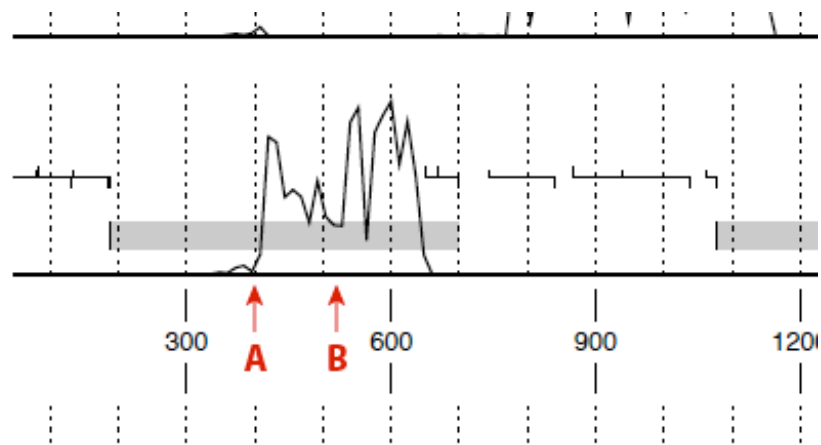


Figure 8.7

- **Did Glimmer and GeneMark agree on the start for this gene?** Check the 'Notes' field under the [Feature] tab and the [[Description]] sub-tab to answer this question. In our example, shown in Figure 8.8, the two programs disagree; Glimmer has called the start at position 408, and GeneMark at 297.

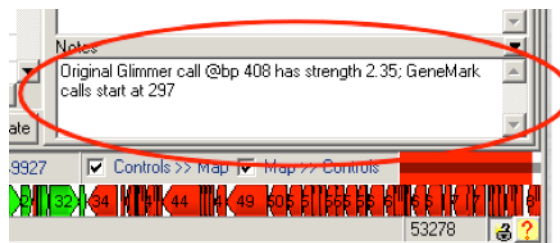


Figure 8.8

- **Does the predicted start have an associated ribosome binding site [RBS; Shine-Dalgarno (SD)] with a high score or recognizable sequence?** You may recall that you can use the Frames window (DNA → Frames) to review the SD scores for all start options in a given ORF. (Check Section 4.4.4 for details on how to open the "Choose ORF start" window and select a particular ORF.) Figure 8.9 shows that there are four possible start codons for our example. Their SD scores are shown in the red box. There is also a snippet of the upstream sequence so it can be inspected manually.

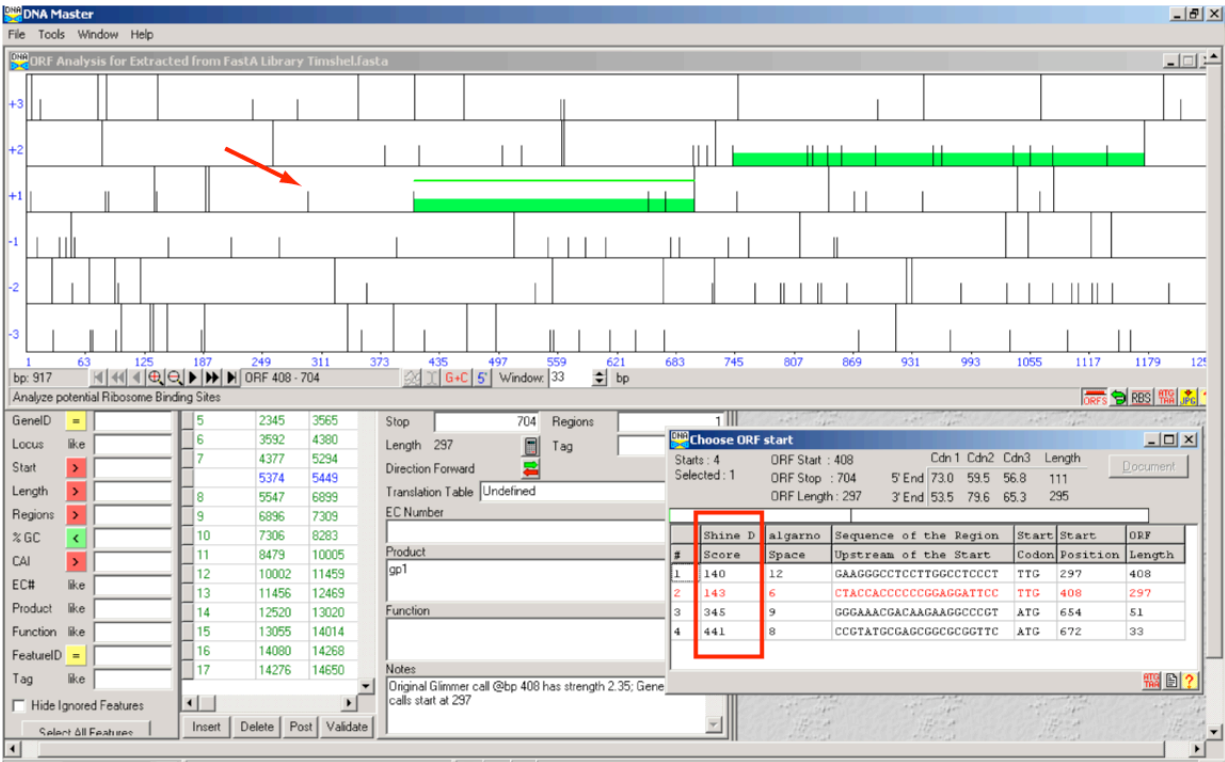


Figure 8.9

- **Is the predicted start codon the longest possible for the ORF without causing excessive overlap?** According to Guiding Principle #3, bacteriophage genomes tend to have very small gaps between genes. Therefore, you may want to consider changing to a further upstream start codon if it provides better gene packing. For example, the start codon indicated by a red arrow in the figure above is worth investigating, because it would provide a longer gene without any overlap.
- **Does the start site match other starts for similar genes in GenBank?** To view the relevant information, go to the **[[Blast]]** sub-tab, then the **[[Alignment]]** sub-sub-tab. You can select different BLAST alignments in the top pane to see how your start compares to those in a variety of other genomes.

In **Figure 8.10**, we're looking at our gene compared to two others. On the left is the alignment to gp1 of Bx2, and on the right is the alignment to HelDan gp1.

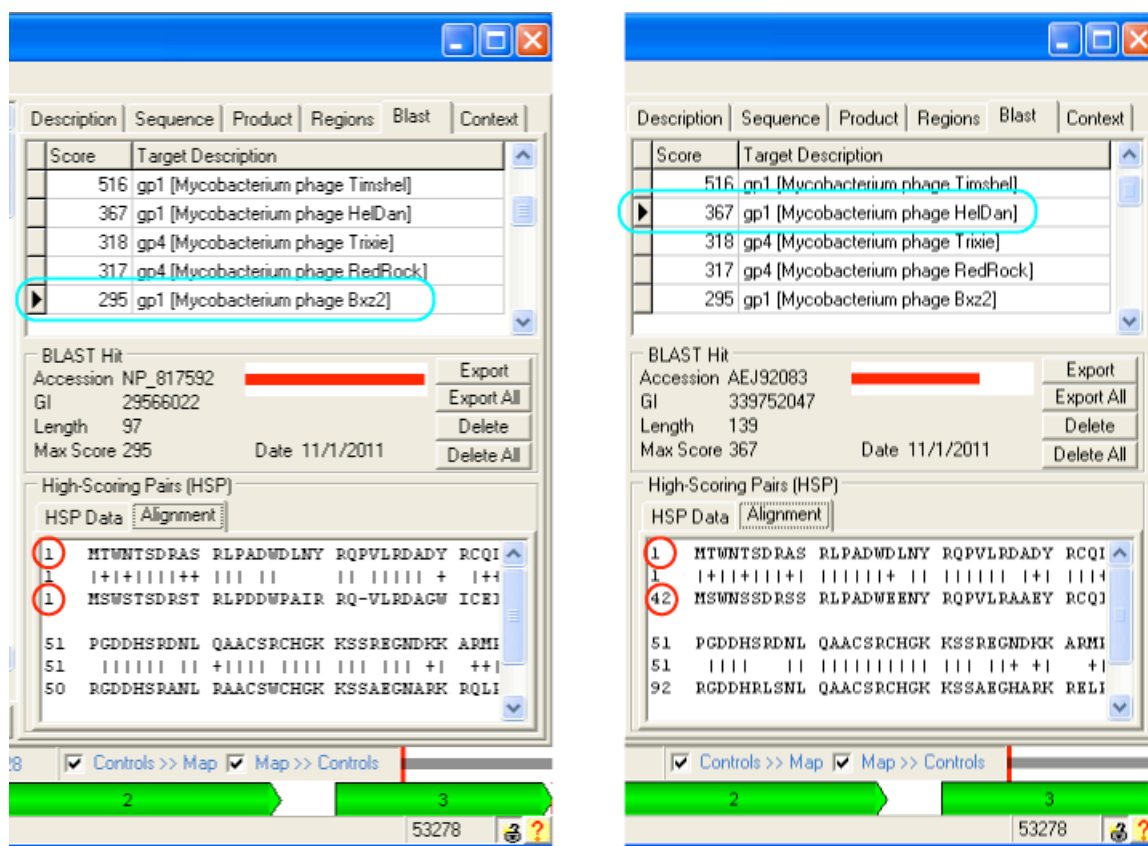


Figure 8.10

The numbers highlighted with red circles are critical. The left pane shows a “1-to-1” match, meaning both proteins start at the same position—supporting evidence for this start. The right pane, however, shows a amino acid 1 of our protein matches amino acid 42 of HelDan gp1, meaning that gene’s published start is further upstream. Because these differ, it’s wise to check several BLAST alignments before making any decision. In this case, HelDan is the exception and most of the BLAST alignments are 1-to-1.

Remember to not be overly enthusiastic about alignment to other gene products, because you don’t know *a priori* whether these were correctly identified. You just know that someone made that choice during a previous annotation.

You now need to put this information together to make the best choice. Occasionally, the answer to all five questions above will be “yes,” and the auto-annotated start will clearly be the best choice. More commonly, the answers will conflict, and human judgment becomes necessary. This is the meat of the annotation process.

For our example:

By looking at the “Choose ORF start” window (shown in **Figure 8.9**), we see that there are 4 possible start sites for this gene. Numbers 3 and 4 can be easily dismissed because they both cut off some coding potential (counter to Guiding Principle #6) AND lead to very short genes (51 bp and 33 bp, respectively, counter to Guiding Principle #5).

That leaves us with two possible start sites to consider, one at 408 (called by Glimmer) and one at 297 (called by GeneMark). The SD scores for these are similar, though the one at 408 is a bit higher. Also, note that both utilize a TTG start codon.

In this instance, because the starts are rather similar in their features, it is also helpful to take a manual inspection of the upstream sequences. In this case, the GGAGGA, located about 4 bp upstream of the TTG start site at position 408 is a good match to the consensus sequence (remember Guiding Principle #10b).

Though the start at 297 would give a longer gene and was called by GeneMark, the best choice here is the start at 408. It was called by Glimmer, the RBS has a higher score and a better consensus sequence, the GeneMark coding potential begins near this start, and the majority of homologous genes have start sites at the same position.

DECISION TIME: Is the currently called start site for the gene the best choice?	
GUIDANCE: Ten percent or more of your genome's start sites will likely have to be changed, and in some cases NEITHER Glimmer nor GeneMark will call the correct start. For each gene, gather the information described in this sub-section, and try to weigh all possibilities to arrive at the best call.	
YES	NO
ACTION: Continue to Section 8.4.3 .	ACTION: You need to change this gene's start. Go to Section 9.2.3 for instructions.

8.4.3 Is this gene part of a programmed translational frameshift?

This is the first of three less common features you may come across during your annotation (followed by Introns and Wrap-around Genes).

Many dsDNA tailed phages encode a pair of genes that are expressed via a programmed translational frameshift, meaning the ribosome changes reading frame in the middle of translation. The resulting gene products are usually involved in assembly of phage tails, and the two genes are often just upstream of the tape measure protein gene.

The prototypical example for this is in phage lambda where the two genes are called 'G' and 'T'. The proteins expressed are, however, gpG (the product of the first gene), and gpGT, which starts at the beginning of 'G' (with the same start as gpG) but ends at the end of the second open reading frame. It accomplishes this by shifting translational reading frames about 8-9 codons upstream of the stop codon of 'G', and into the frame of 'T' (**Figure 8.11**).

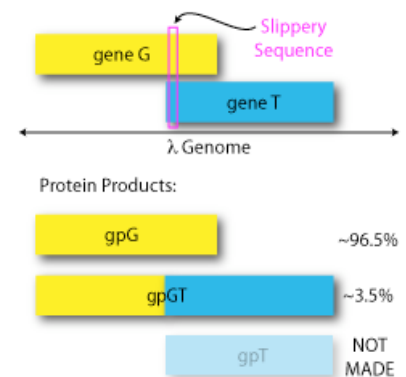


Figure 8.11

There are a few quick ways to determine whether your gene might be part of a programmed translational frameshift.

- It is one of the two genes immediately upstream of the tape measure protein gene. The tape measure protein gene is almost always the longest gene in mycobacteriophage genomes, and is thus relatively easy to locate. Not all tape measure protein genes are preceded by frameshifts, however.
- A Phamerator map shows that homologues in finalized annotations of similar genomes are parts of a frameshift. In Figure 8.12, Angelica's frameshift is shown in the blue box, and appears as two genes that start at the same position but have different stops. Fionnbharth's annotation is just a draft, however, and so the equivalent region (shown in the red box) does not yet have the frameshift called properly.

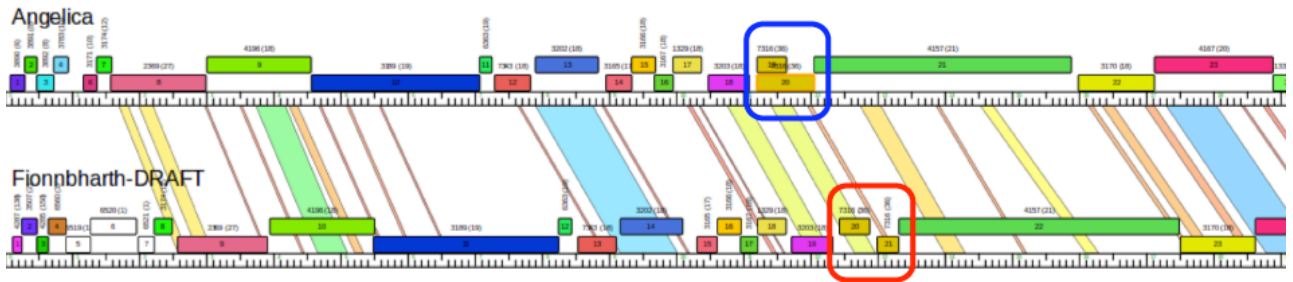


Figure 8.12

- The GeneMark-Smeg output shows strong coding potential, but no way to call the second gene without significant overlap or excluding strong coding potential.

In Figure 8.13, we see that both genes have strong coding potential, but that to include all the coding potential for the gene that's the equivalent of lambda's 'T', we would have to choose a start site that would lead to a nearly 100 bp overlap between these genes (overlapping region shaded blue). Guiding Principle #2 tells us that this size overlap is very uncommon. The auto-annotated start site, on the other hand, is too far downstream and misses strong coding potential. Seeing a situation like this on the GeneMark output should draw your attention and make you think of a possible frameshift.

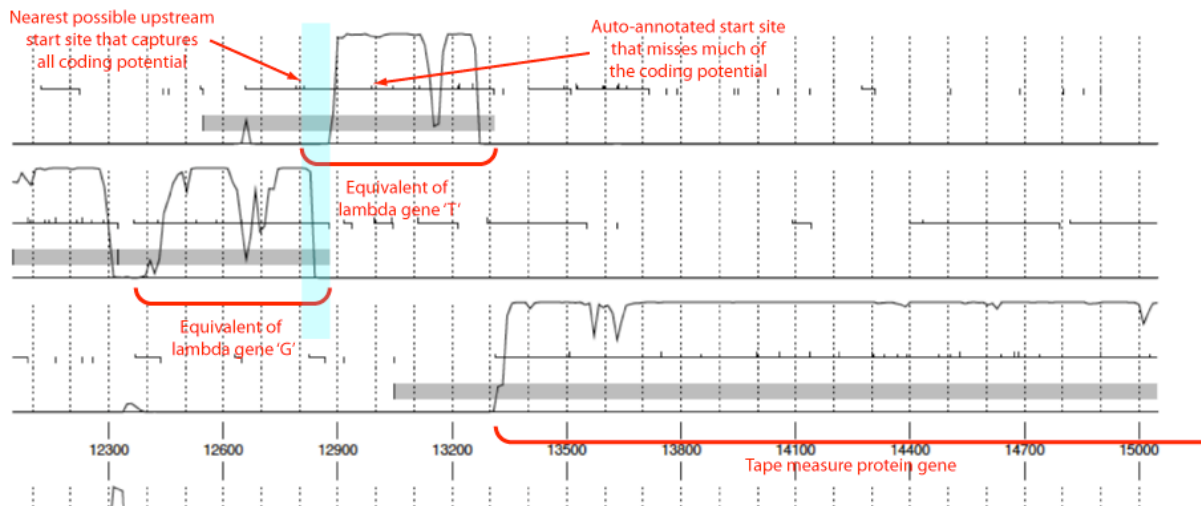


Figure 8.13

For more on frameshifts in phages see the following paper.

'Conserved translational frameshift in dsDNA bacteriophage tail assembly genes' Xu J, Hendrix RW, Duda RL Mol Cell. (2004)16:11-21

DECISION TIME: Is the gene part of a programmed translational frameshift?	
GUIDANCE: Each mycobacteriophage genome usually has only one or zero frameshifts. The most common location is just upstream of the tape measure protein gene, but this is not a guarantee. If you suspect that you may have found a frameshift, investigate it carefully.	
YES	NO
ACTION: You need to properly annotate this frameshift. Go to Section 9.4.1 .	ACTION: Continue to Section 8.4.4 .

8.4.4 Does this gene contain an intron?

Even rarer than frameshifts in mycobacteriophage genomes are introns. We are only aware of perhaps two instances. Identification of potential introns is easiest using Phamerator maps that compare your genome to similar genomes. In **Figure 8.14**, the major capsid protein in Omega is labeled. In LittleE's genome, this protein appears to have been split into two pieces with a reverse-transcribed gene between them.

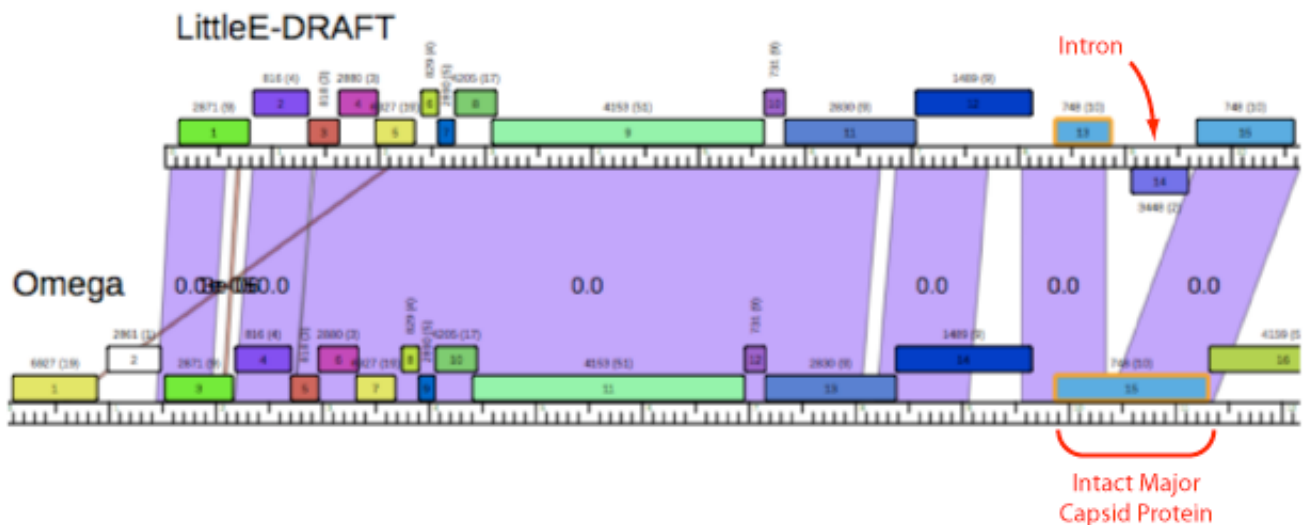


Figure 8.14

Of course, any tailed phage needs a functioning major capsid protein to be viable, so this situation in LittleE drew our attention. Subsequent experimental evidence has since verified that this is, in fact, an intron spliced out at the RNA level.

DECISION TIME: Does this gene contain an intron?	
GUIDANCE: As mentioned above, introns are quite rare in mycobacteriophage genomes, so most of the time the answer is an easy “No.” If you’re lucky enough to locate a potential intron, you should use a variety of bioinformatic approaches to see if the putative intron shares any similarity to known introns. Experimental evidence may ultimately be needed for verification.	
YES	NO
ACTION: You need to properly annotate this intron. Go to Section 9.4.2 .	ACTION: Continue to Section 8.4.5 .

8.4.5 Does this gene wrap around the ends of the genome?

In genomes that do not have defined ends, the left end of the genome is defined arbitrarily, and occasionally a gene may extend from the right end of the genome into the left end. We refer to these as “Wrap-around genes.”

To even possibly be a wrap-around gene, an ORF **must** meet the following two criteria:

1. It is positioned at the right end of the genome.
2. It is transcribed in the forward direction.

If it does not meet these criteria, it is not a wrap-around gene, and you may proceed to the next steps.

If it does meet these criteria, you should check the GeneMark-Smeg output (**Section 5.3**) for strong coding potential near the right end of a genome with no stop codon present (see **Figure 8.15** below). The presence of strong coding potential implies that a wrap-around gene may be present.

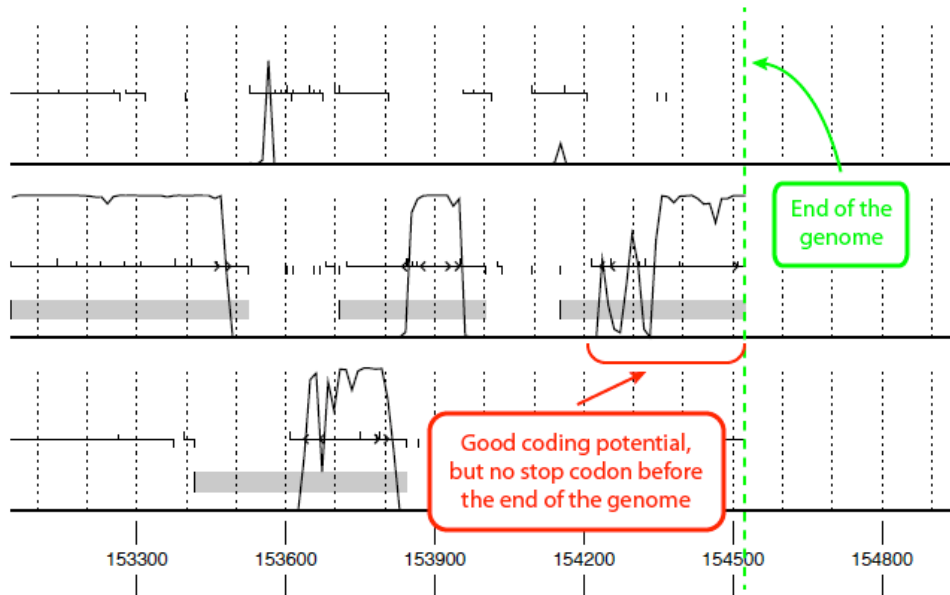


Figure 8.15

<p>DECISION TIME: Does this gene wrap around the ends of the genome?</p>	
<p>GUIDANCE: This question only applies at the extreme right end of circularly permuted genomes, so most of the time it is not a concern. The GeneMark-Smeg output is critical for locating these genes when they do exist.</p>	
<p>YES</p> <p>ACTION: You need to properly annotate this wrap-around gene. Go to Section 9.4.3.</p>	<p>NO</p> <p>ACTION: You need to determine if your gene has a known function. Go to Section 10.</p>

8.5 Checking gaps in the draft annotation for uncalled genes

According to Guiding Principle #3, the genes in phage genomes are generally tightly packed, so any large gaps (>50 bp) in your annotation should be reviewed.

In circumstances where you have a series of genes in the same orientation that are likely to be expressed as an operon, these genes are typically nestled closely end-to-end. However, non-coding gaps are perfectly legitimate and to be expected, and filling gaps with poorly justified gene calls is not appropriate.

There are two basic things you should look for in gaps.

- **Can the start site of the downstream gene be extended so that the gene covers more of the gap?** Carefully consider all possible start sites for the downstream gene. If a longer one is available, compare it to the current start site to see if it is a similar or better choice. All other things being equal, a longer call is usually preferable, but do not extend genes just to fill a gap.
If YES, go to Section 9.2.3 to change the start site.
- **Is there a protein-coding gene in this gap?** You have several resources to help answer this question. First, you can use Phamerator maps to see if any similar genomes have a gene called in this gap. Second, you can look at the GeneMark-Smeg output to see if any of the reading frames in this gap show some coding potential. Third, you can copy the DNA sequence from your gap and use it to run a BLASTX search on NCBI. The combination of these techniques may yield convincing evidence that the gap contains a protein-coding gene that was missed by both Glimmer and GeneMark.
If YES, go to Section 9.2.2 to add a gene.

Remember too that you should expect non-coding gaps between divergently transcribed genes as there is a strong prediction that promoters lie within these regions. For example, in **Figure 8.16**, we should expect some gap between gene 47 (transcribed leftwards) and gene 48 (transcribed rightwards).

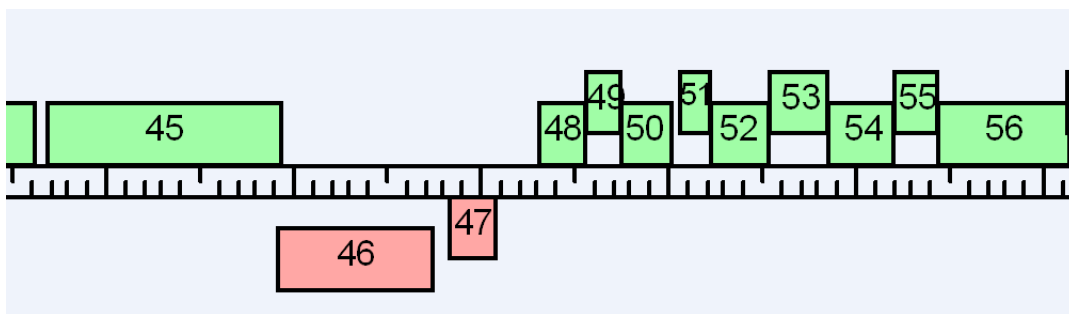


Figure 8.16

8.6 Finding and refining tRNA and tmRNA genes

DNA Master searches for tRNAs by default, but may miss some tRNAs that other approaches can find, or may miscall the precise boundaries of these genes. See **Section 9.5** for information on how to search for and call tRNAs and tmRNAs.

8.7 *Completing your annotation refinement*

Much of the work of annotation is following the steps above—for each gene and gap in your genome—until you’ve settled on the best calls for each with the information given.

As a double-check, you should scroll through the Feature table and the genome map (using buttons at the bottom of the **[Feature]** tab) to make sure that all the changes you’ve made have been committed to the file.

Several important steps remain, however.

1. **Documenting your gene calls.** You can use the Notes field (under **[Feature]** **[[Description]]**) to record notes about each gene as you go. Your final submitted file, however, should have each gene’s Notes field filled in according to specific instructions so as to facilitate checking the annotation. These documenting instructions are described in **Section 9.6**.
2. **Determining putative functions.** You’ve figured out where the genes are (and aren’t), so the next step is to see if you can make a well-supported guess as to what they do. This process is covered in **Section 10**.
3. **Merging several different portions of the annotation into a single file.** In a classroom setting, often you will choose to split the genome into sections and have different groups or students work on different sections. If you’ve split the genome up, now is the time to bring everyone’s work back together or “**Merge**” the different annotations. This process is described in the first part of **Section 11**.
4. **Checking the final annotation.** Once you’ve produced a nearly final annotation, it still needs a (relatively) expert eye to double-check it, as described in **Section 11**.
5. **Submitting final files.** When you’re confident in your annotation, have investigated every nook and cranny, and are ready to send it out the door, you’ll need to generate and submit a final DNA Master file, as well as a list of those who have worked on the annotation and should be authors on the GenBank submission. This is described in **Section 12**.

9 The mechanics of making changes to your annotation

9.1 Overview

This section, unlike most sections of this guide, is not intended to be a sequential step-by-step description of any part of the annotation process. Rather, it is intended to be used as a reference section for how to make specific changes to your annotation. The actual decision-making steps were described in **Section 8**, and a graphical summary can be seen in the Decision Tree in **Section 8.3**.

The three most common operations you'll need are covered first. They are:

- Deleting a gene
- Adding a gene
- Changing the start site for a gene

The following sub-sections describe some common steps you should take after making any changes to your annotation. They are:

- Posting changes
- Validating your calls
- Renumbering your genes
- Re-BLASTing a gene you've changed

There are also some less common operations that you may need. They are:

- Annotating a programmed translational frameshift
- Annotating introns
- Annotating wrap-around genes

Next is a sub-section on RNA genes. It is:

- Predicting tRNA and tmRNA genes

Finally, there is a sub-section of how to document the annotation work you've done:

- Documenting your gene calls

9.2 Making common changes to your annotation

9.2.1 Deleting a gene

- Select the **[Feature]** tab of your main genome file.
- In the center column, click on the feature you would like to delete to select it. (The selection can be verified by the presence of a black arrow to the left of the gene name.)

- Click the '**Delete**' button, found at the bottom of the center column.
- Click the '**Post**' button to commit your changes to the database.

9.2.2 Adding a gene

If it's not already open, open the Frames window by going to **DNA → Frames**

- Locate the ORF that corresponds to the gene you would like to add.
- Click within that ORF, and a green or red line will appear, depending on its orientation.
- Click on the '**RBS**' button in the lower-right corner.
- Confirm that you have selected the correct frame by verifying the coordinate of the **STOP** codon. There can be many possible starts for each ORF, but there is only one possible stop!
- Choose the best start, as described in **Section 8.4.2**, then click anywhere in that start site's row in the "Choose ORF start" window to select it.
- Return to the [**Feature**] tab and click on the '**Insert**' button at the bottom of the center column.
- A new window will appear that allows you to add the feature. Verify that the correct orientation (forward/reverse) is selected and that the coordinates are correct. Do not worry about adding the correct gene number or gene product (gp) number, as the genes will get renumbered using the Validation function when you are done.
- Check the boxes '**add to feature table**' and '**add to documentation**'.
- Click '**Add Feature**'.
- Click the '**Post**' button to commit your changes to the database. This is also a good time to save your file.
- Your new gene will likely be placed at the end of your feature list, because the default sorting is by index number, rather than genome position. To sort by position, find the dropdown box at the top left of the [**Feature**] tab labeled '**Sort by**', and change it from "Index" to "Start."
- You may want to collect BLAST data for your new gene. See **Section 9.3.4** for instructions.

9.2.3 Changing the start site for a gene

- Select the [**Feature**] tab of your main genome file.
- In the center column, click on the gene you want to change to select it.
- Click on the [**Description**] sub-tab to the right.
- In the box labeled "Start", third from the top under "Description", type in the new start coordinate you've selected.
- Click on the Calculator button (this is an icon of a calculator, found just to the right of the "Length" display) to recalculate the ORF length. The new length (in bp) will be shown and should reflect your change.

- Click the 'Post' button at the bottom of the central column to ensure your changes are saved to the database. This is also a good time to save your file.
- Because you've changed the start site, you'll probably want to re-BLAST this gene so that the BLAST results reflect your change. See **Section 9.3.4** to do so.

9.3 Common steps to take after making changes

9.3.1 Posting changes

When making gene changes—including changing start codons, deleting genes, annotating programmed frameshifts, adding notes to the Notes field, etc.—you need to both **enter** and **post** the changes. Simply entering them is insufficient, and the changes may be lost. Once you've learned how to post, it doesn't hurt to **post often!**

Normally, a selected gene in the feature table will be indicated by a triangle, as shown below.

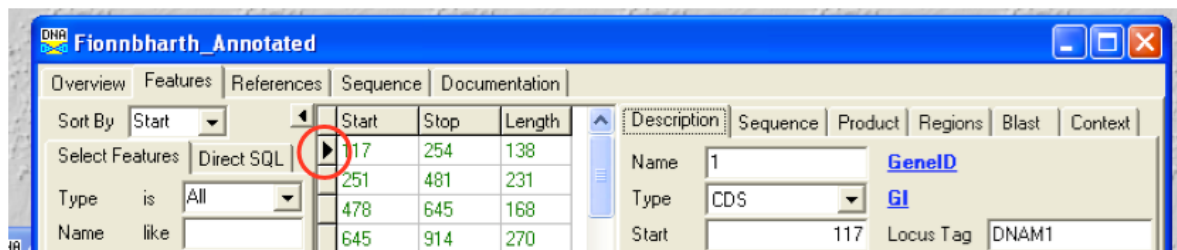


Figure 9.1

When you make a change to a feature listed in the Feature table (e.g., begin typing in the Notes field), the icon next to the feature changes to an Insert icon, as shown below.

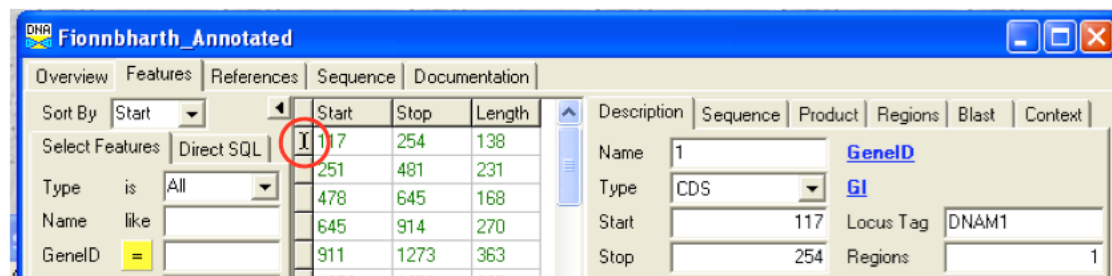


Figure 9.2

For the most part, this change to Insert Mode happens automatically when you start typing in any of the fields under the Description tab. Your changes, however, **won't be posted to the database until you exit Insert Mode.**

The following are ways to make sure your edits get posted to the database.

- ✓ Click on the 'Post' button at the bottom of the center column.
- ✓ Click on the **Calculator** icon, after changing a start or stop.
- ✓ Click on a different feature in the center column.

You will be able to tell that your changes have posted to the database because the Insert icon will change back to the right-pointing triangle.

Important Note: The follow are ways that your changes will **not be posted** to the database, and **WILL BE LOST**.

- ✘ Saving your file while still in Insert Mode.
- ✘ Clicking on a different tab or sub-tab while still in Insert Mode.

9.3.2 Validating your annotation

As you work through your genome, DNA Master has a handy **validate** feature that helps ensure your gene calls have valid start/stop codons and do not have any internal stop codons.

To perform a genome validation, follow the steps below.

- Click on the '**Validate**' button, at the bottom of the central column in the **[Features]** tab (located in the red circle in **Figure 9.3** below).

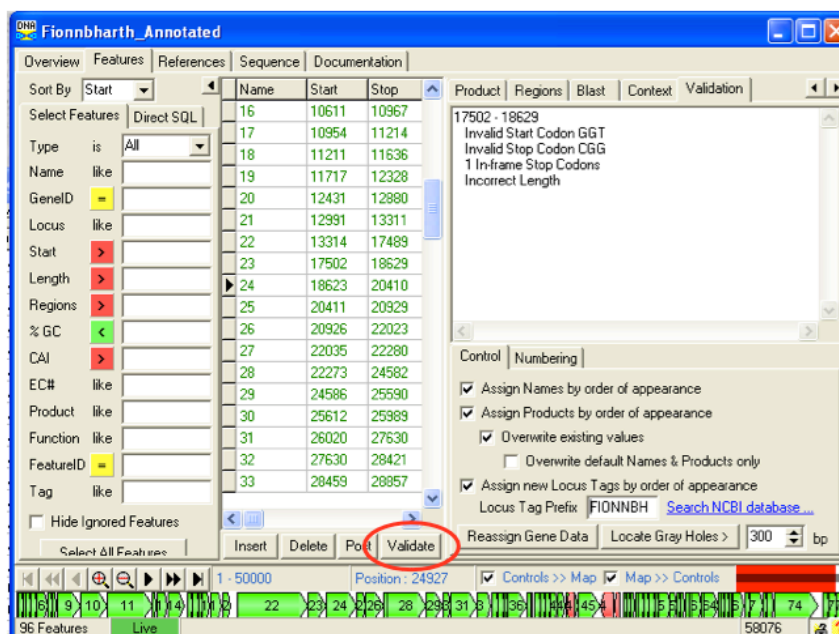


Figure 9.3

DNA Master will let you know when gene calls are not in frame or if they have incorrect start or stop codons. A genome is not complete unless validation returns as "All ORFs are valid".

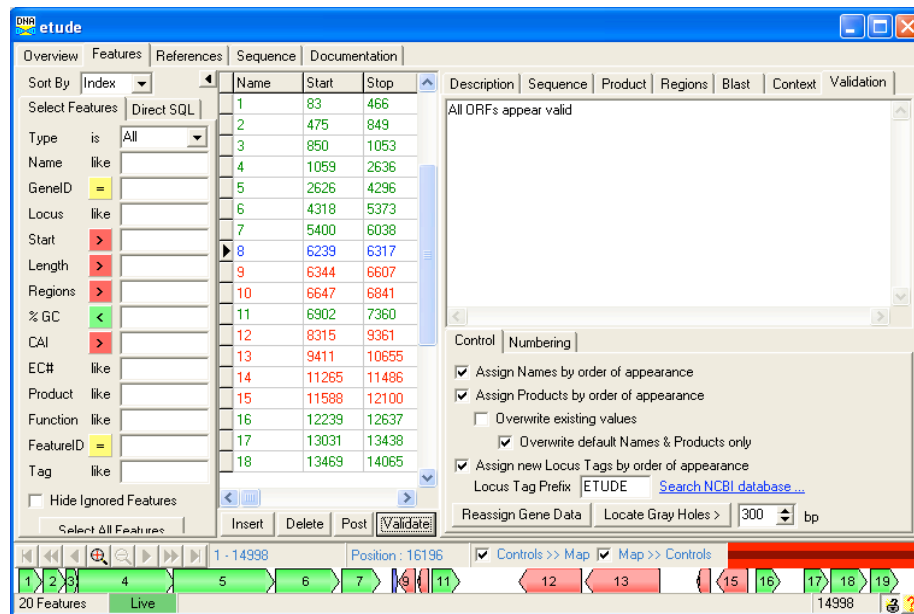


Figure 9.4

If the validation generates failures, you should check the coordinates in those features to see what might have gone wrong and make necessary changes. You can then re-run the validation to ensure all ORFs are valid.

9.3.3 Renumbering annotated features

When you add or delete a gene, you may want to renumber the genes to reflect the change. Genes added manually after auto-annotation will appear at the bottom of the feature list when sorted by **Index**. Sorting by **Start** will place the gene in its correct order by start coordinate.

To renumber your features:

- In the **[Features]** tab, click the 'Validate' button located at the bottom of the central column. This will open the **[Validation]** sub-tab on the right side.
- Check the boxes as shown in **Figure 9.5** below.

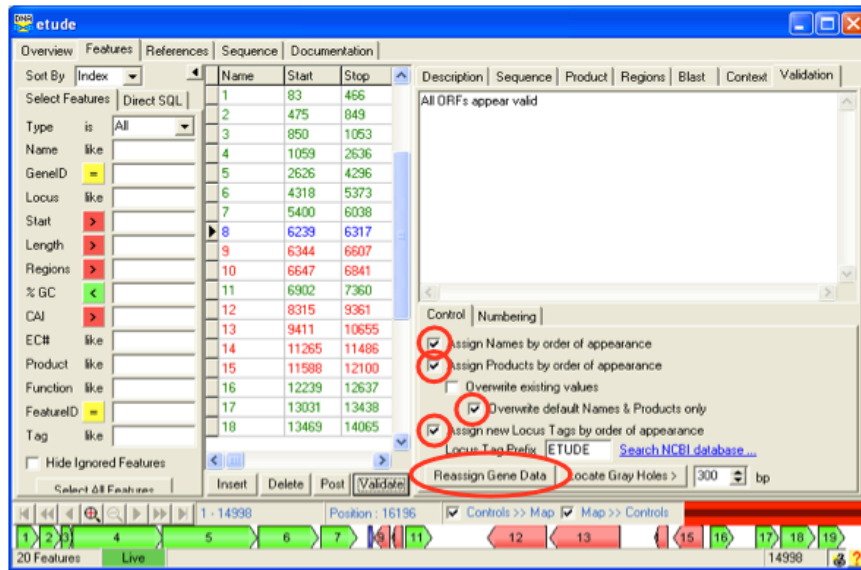


Figure 9.5

- In the field marked “Locus Tag Prefix”, type your phage’s name. (GenBank assigns a unique locus tag to every gene in GenBank, preferably constructed from the phage’s name and gene number.)
- Click the ‘**Reassign Gene Data**’ button.
- Click ‘**Yes**’ to confirm in the window that pops up.

Note: If you’re annotating a portion of a genome as one part of a larger group, you may not want to renumber genes because this may cause confusion if some groups do so and others do not. Make your own decisions, but bear this in mind. You can re-number as often or as little as you like.

9.3.4 Re-BLASTing a gene

Once you have finished adding a gene, changing a gene’s start site, or entering multiple regions for a gene, it can be useful to re-BLAST the gene. This is particularly helpful to check whether or not a gene’s modified start site now matches those published in GenBank.

- From the [Features] tab, select the [[Blast]] sub-tab.
- Click the ‘**Delete All**’ button, identified in **Figure 9.6**.

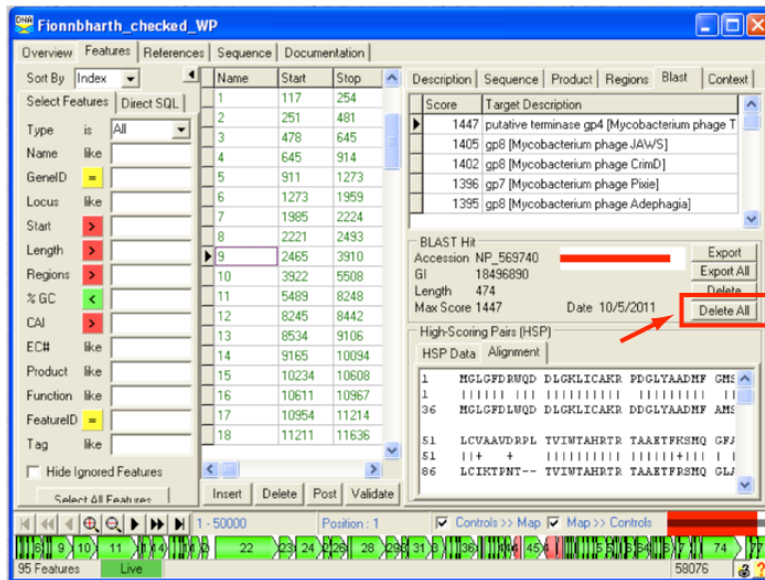


Figure 9.6

- A dialog box will pop up and ask if you really want to delete all the BLAST hits for this gene. Click 'Yes'. The BLAST tab will now be empty of hits, as shown below.

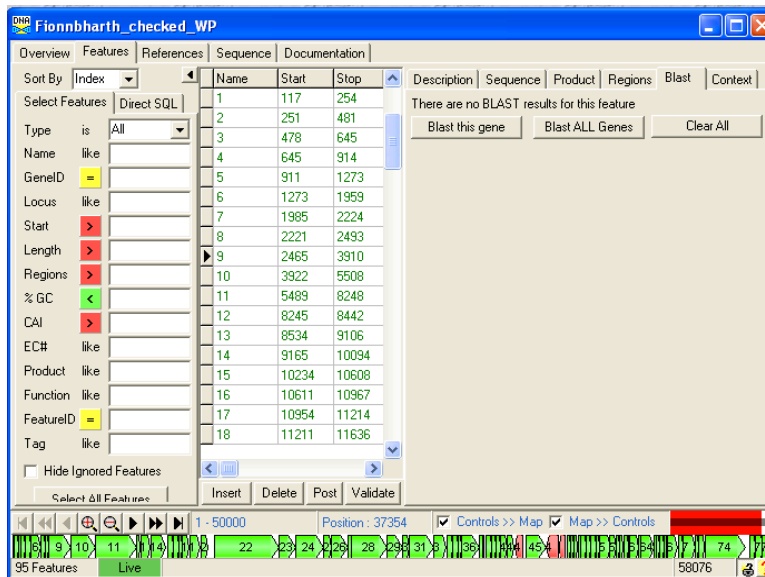


Figure 9.7

- Click the 'Blast this gene' button.
- A new window will appear, labeled "BLAST search for [your gene coordinates]". The status of the BLAST attempt will continually be updated in this window until the BLAST is done. When it is finished, the window will display the BLAST results as shown in Figure 9.8.

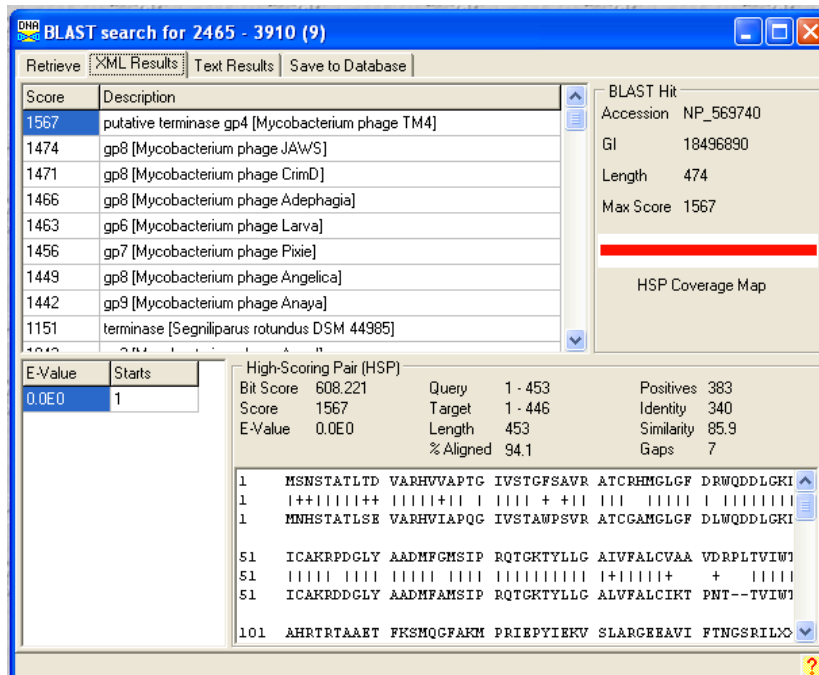


Figure 9.8

- To save your new BLAST hits to your genome file, select the [Save to Database] tab.

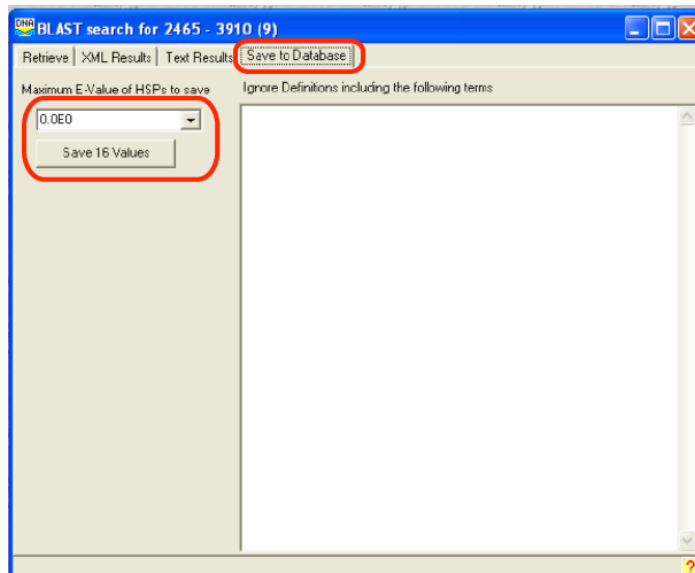


Figure 9.9

- Click on the drop-down arrow next to the empty field under 'Maximum E-Value of HSPs to save'.
- Scroll through the listed E-values (these are from your new BLAST matches) and pick an appropriate value (greater than 10^{-3}) that also gives you a useful number of matches (at least 10 or so). If you only have E-values higher than 10^{-3} , just pick at least one match so you will know that you have BLASTed this gene, and it doesn't have any good matches in GenBank.

- Click the ‘Save [n] Values’ button. The “n” will be automatically filled in for you based on the number of matches you picked from the drop-down menu. It should then say “[n] saved” in this window under the button. Close the BLAST window.
- Now your new BLAST hits should be listed in your genome file (you may not see them until you select a different feature and then reselect the one you just BLASTed to refresh the view).

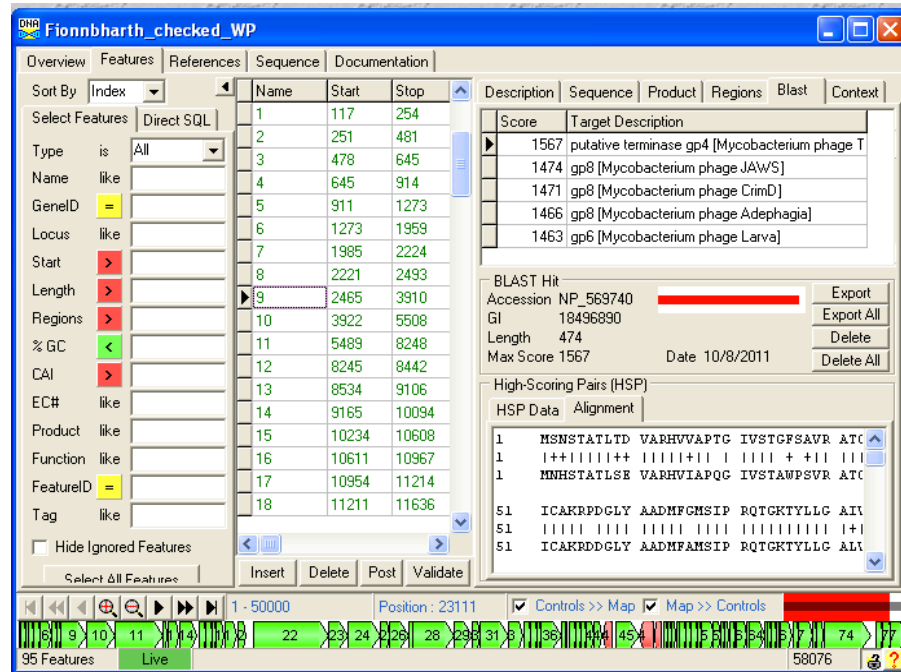


Figure 9.10

9.4 Making less common changes to your annotation

9.4.1 Annotating programmed translational frameshifts

Assuming you have identified the two genes involved in the frameshift (see Section 8.4.3), the next critical piece of correctly annotating a frameshift is locating the precise position where the shift occurs. A printed six-frame translation of the region in question is helpful during this process (see Section 5.1).

Frameshifting occurs when the ribosome encounters a “slippery” sequence in the mRNA, such as GGAAAA, and loses track of how to count to three. In the most common shift, the -1 shift, the first “A” of the above sequence is “counted” twice; it is read as the third nucleotide in the last codon of the upstream region, AND the first nucleotide in the first codon of the downstream region. (There are also examples of +1 shifts, in which a nucleotide is skipped, or -2 shifts, in which two nucleotides are counted twice.)

For those unfamiliar with finding the slippery sequences and determining where and how the shift is occurring, it is probably easiest to examine a similar phage’s genome in Phamerator that has a correctly annotated fusion gene, and compare it to the six-frame translation of your own phage’s fusion gene. This will help to determine what the correct amino acid sequence should be, and therefore which nucleotide the shift must occur at.

To annotate a programmed translational frameshift within your phage, you should do the following (we use Fionnbharth below).

Determine the precise location of the shift

- Using Phamerator or BLAST, find the most similar genome you can that has a correctly annotated frameshift. For Fionnbharth, we've selected Angelica.
- Make a Phamerator map using your genome plus the similar genome you've chosen (see **Section 6.4**).
- Click on the first gene in the correctly called frameshift in Phamerator to select it. Its border will change from black to orange to indicate that it's selected, and its nucleotide and amino acid sequences will be displayed in the panels at the bottom of the window, as shown in **Figure 9.11**.

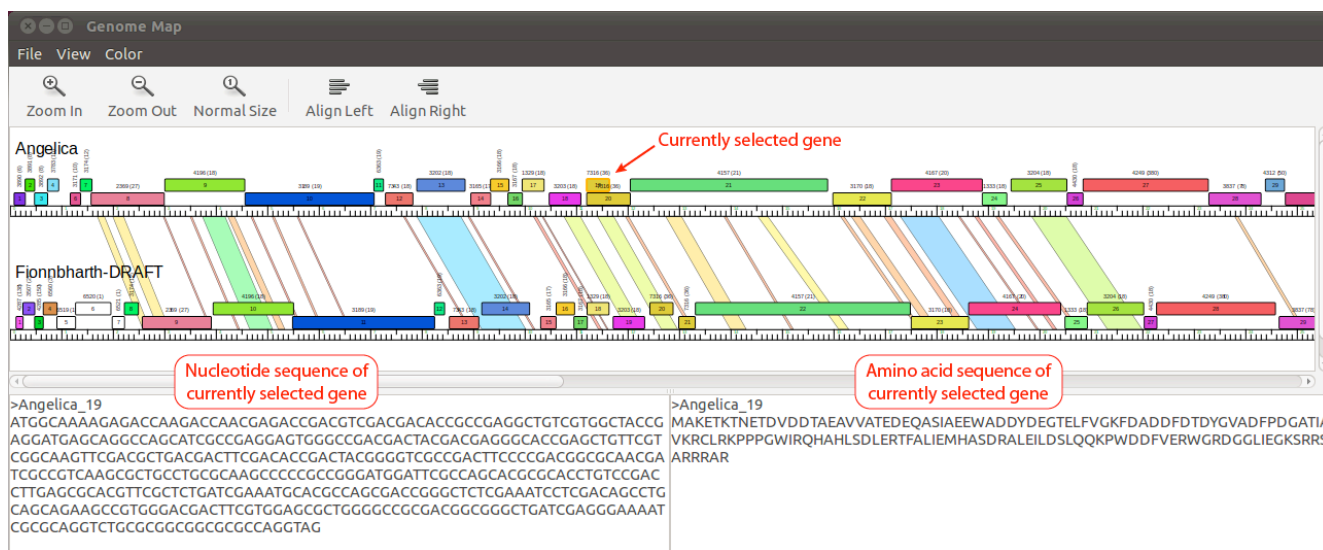


Figure 9.11

- Copy the amino acid sequence from the bottom-right panel and paste it into a new text file.
- Now select the second correctly called frameshift gene (just below the first), and copy and paste its amino acid sequence into a new text file as well.
- Locate the precise position where these two amino acid sequences diverge. (This can be done by manual inspection of the amino acid sequences, or by using BLASTP with the "Align two or more sequences" option checked.) In our example, the two Angelica sequences diverge after amino acid 135, as shown:
 - ... GGLIEGKSRRSA... in the first protein.
 - ... GGLIEGKIAQVC... in the second (fusion) protein.
- Now back to your genome. An examination of your six-frame translation shows the two genes as they were called by DNA Master's Auto-Annotate function.

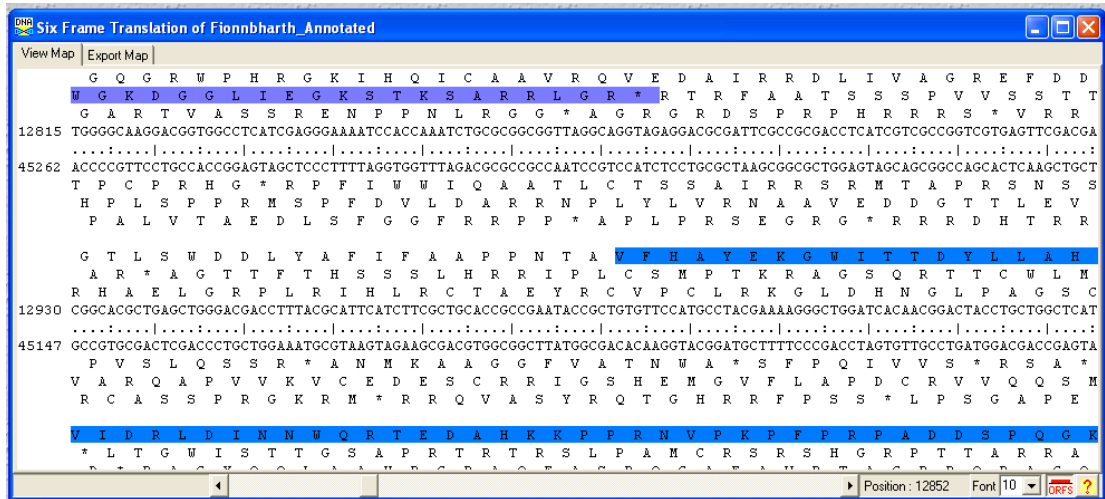


Figure 9.12

- In Figure 9.12, the purple bar shows the end of the first protein, and the blue bar shows the beginning of the auto-annotated version of the second protein. Note that the purple highlight is in reading frame 2 while the blue is in reading frame 1. This means that this phage likely has a -1 frameshift, and we need to identify a nucleotide somewhere in this region that should be “counted” twice by the ribosome.
- Near position 12841 there is an obvious slippery sequence, “GGGAAA” (underlined in red below). If we count the first A (at position 12844) of this sequence twice, we shift frames as shown by the red box, and generate the amino acid sequence ...GGLIEGKIHQIC... in the fusion protein. This sequence is not identical to Angelica’s fusion sequence, but it is very close. Counting carefully from the left, we can determine that the first “A” at position 12844 (underlined in green) is the coordinate of our frameshift.

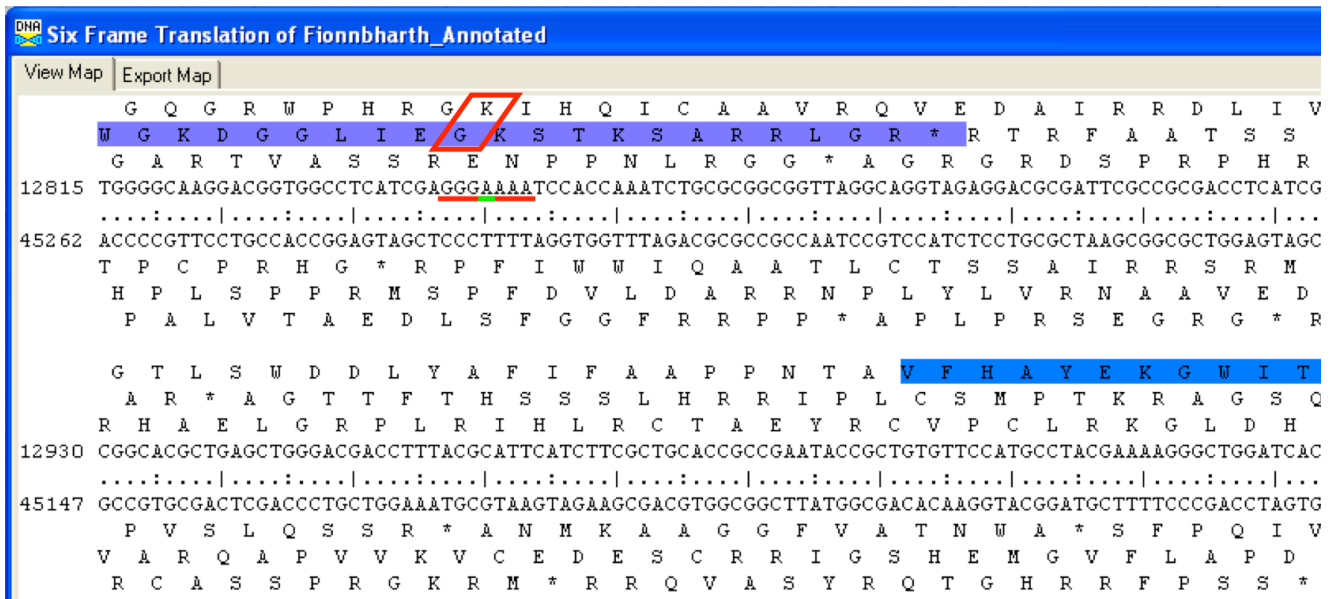


Figure 9.13

Annotate the frameshift in DNA Master

- Go to the **[Features]** tab and click on the **second** of the two genes involved in the frameshift. (We do not need to modify the first gene, only the second.)
- In the **[[Description]]** sub-tab in the right-hand section, locate the field labeled “Regions” (far right column, shown below). Change the number from “1” to “2”, then click the ‘Post’ button at the bottom of the central column to save this change.

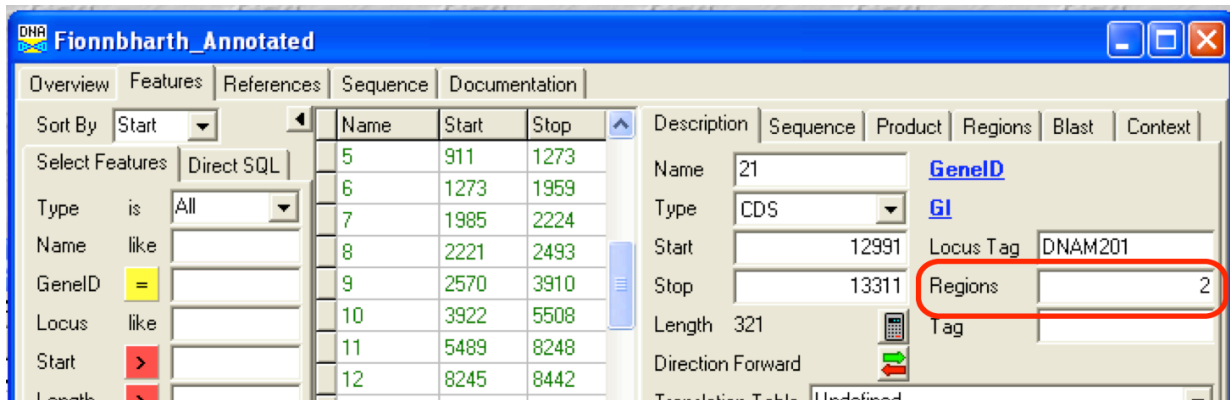


Figure 9.14

- Change from the **[[Description]]** sub-tab to the **[[Regions]]** sub-tab in the right-hand section of the Features tab.
- You will now enter the two regions that constitute the fusion protein. **These must be entered in order**, upstream first and downstream second.
- The **Start** coordinate for the **first region** is the start of the whole frameshift region (same as the start for the previous gene). The **Stop** coordinate for the first region is the position you’ve identified where the frameshift occurs; in our example it is 12844. For the **Length** field, just enter the number 1, because DNA Master will calculate this for us automatically in the following steps, but does require that some number be entered as a placeholder until then.

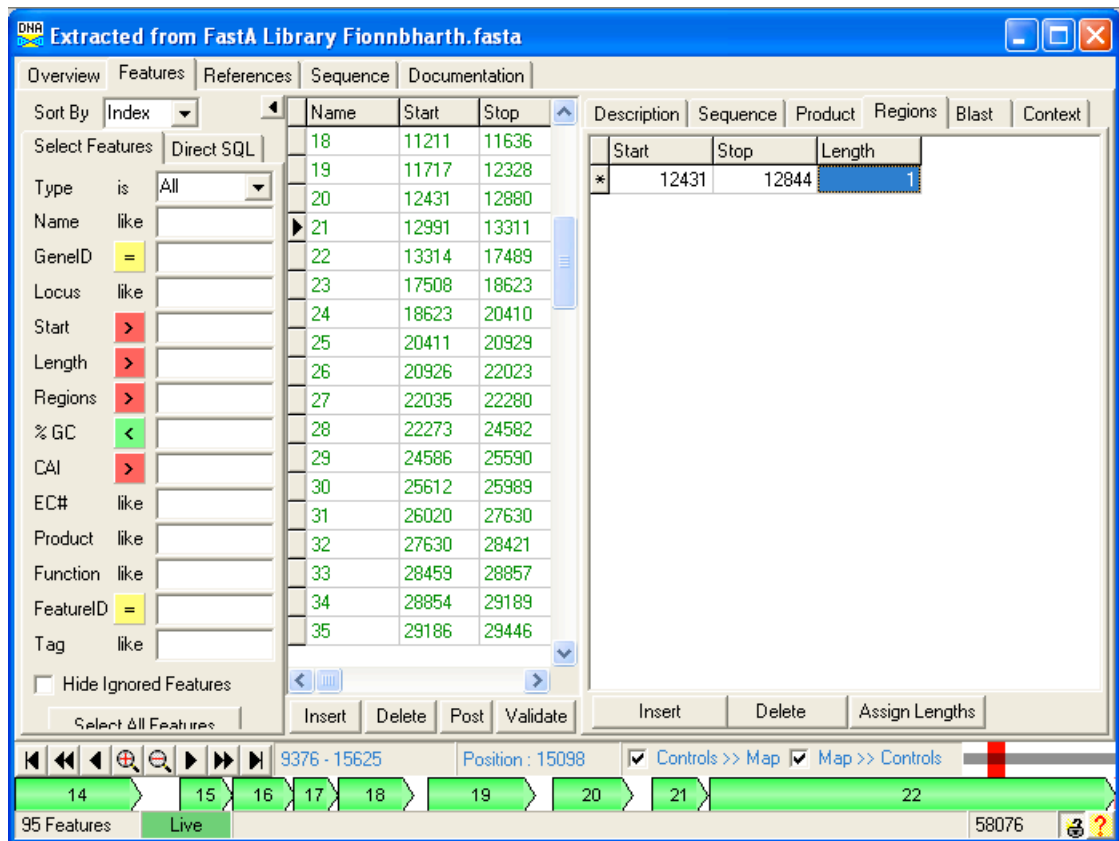


Figure 9.15

- With the “Length” field selected (as shown in **Figure 9.15** by the blue highlight), press **Tab** to move to the second line. For the **second region** of the fusion protein, the **Start** coordinate is the position of our frameshift (again, in our example this is 12844). The **Stop** coordinate is the previously called stop for the second gene (the end of the entire frameshift region, in our example 13311). Again, the **Length** should be entered as “1” for now.
- Click the ‘**Assign Lengths**’ button at the bottom of the **[[Regions]]** sub-tab (see below). DNA Master will calculate the length of each region and display it in the “Length” column.

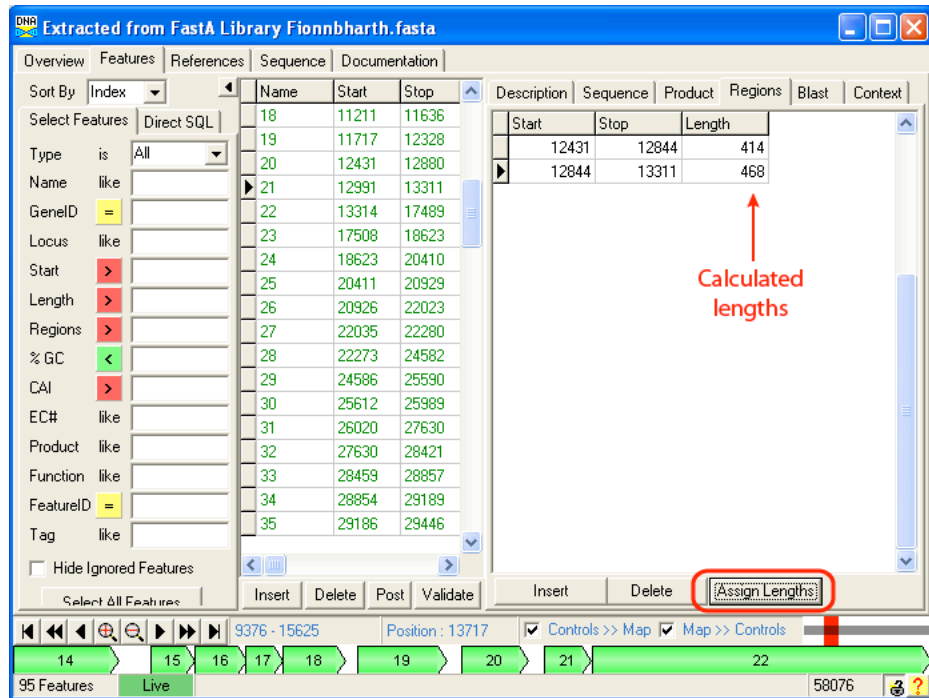


Figure 9.16

- Finally, change back to the **[Description]** sub-tab, and enter the correct start and stop coordinates for the entire gene (both regions). In our example, these coordinates are 12431 and 13311. Then click the **Calculator** icon to post changes and calculate the length of the entire gene.

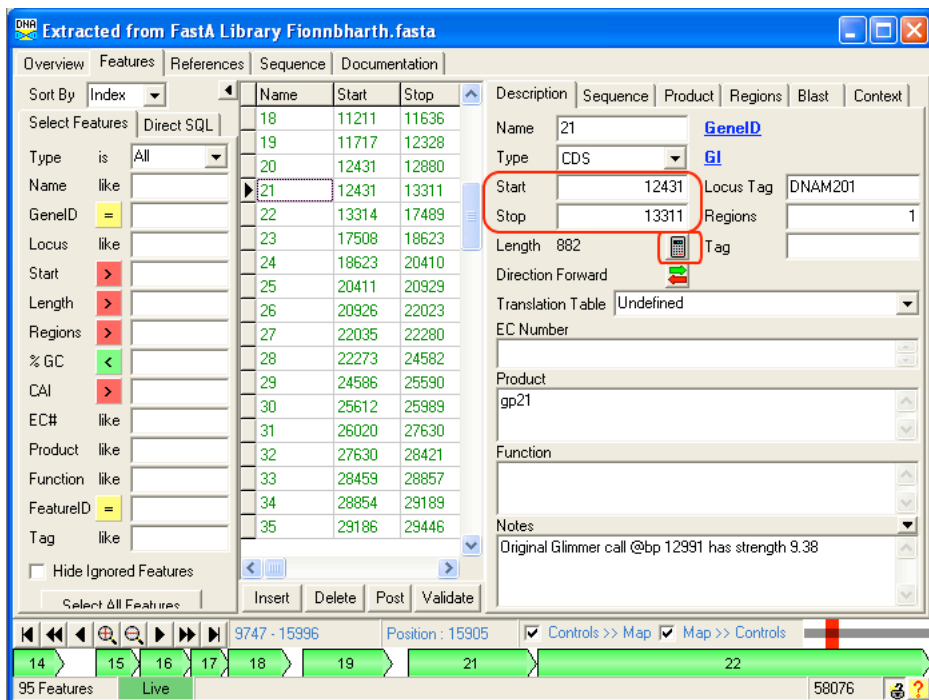


Figure 9.17

Now if you change back to the **Regions** sub-tab, you will see a graphic representation of your two frameshifted regions in black bars at the bottom of the tab, as shown in **Figure 9.18**. (You may need to select a different feature, then come back to this one to refresh the view.)

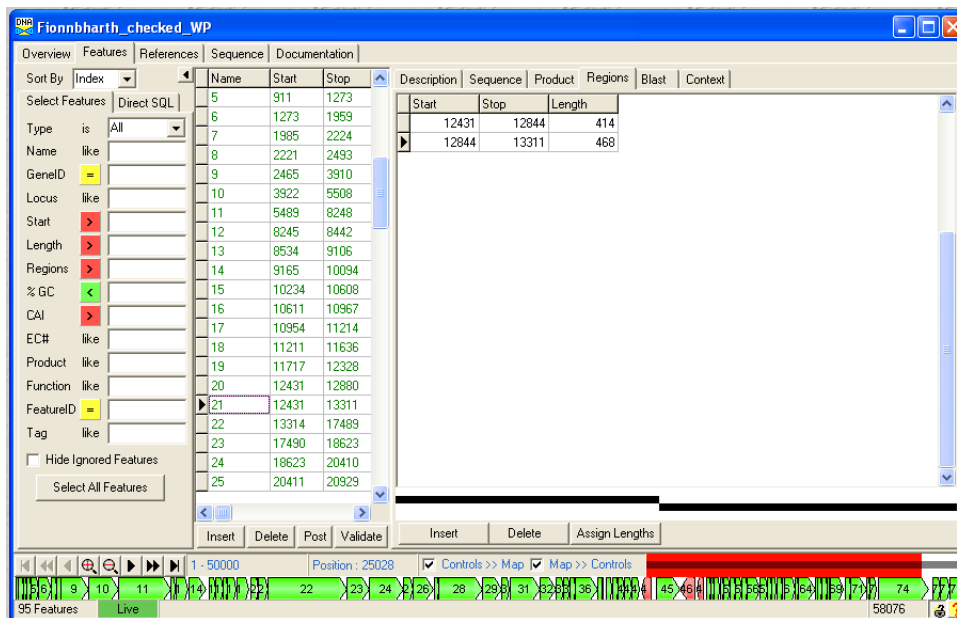


Figure 9.18

9.4.2 Annotating introns

Genes with introns in them can be annotated as two regions by following the procedure above under the heading “**Annotate the Frameshift in DNA Master.**” In this case, the two regions you enter will correspond to the exon portions of the gene. However, determining the precise boundaries of these regions is beyond the scope of this guide, and you need to refer to relevant literature or previous examples to figure this out.

9.4.3 Annotating wrap-around genes

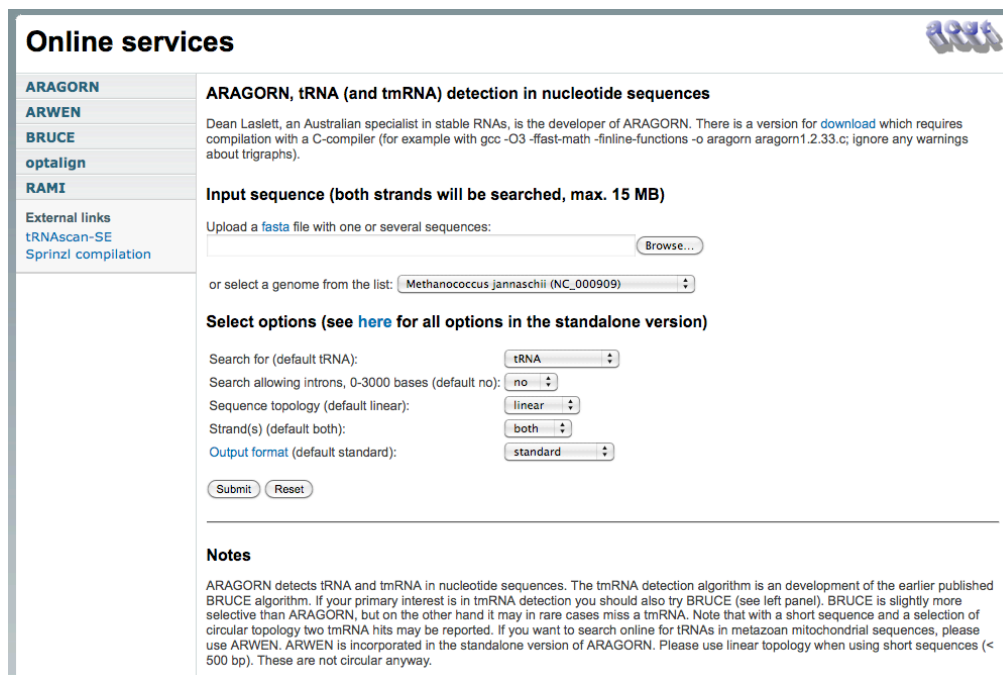
Wrap-around genes can be annotated by following the procedure above under the heading “**Annotate the Frameshift in DNA Master**”, (Section 9.4.1). In this case, the first region will be the portion of the gene at the right end of the genome, starting at your chosen start site and stopping at the end of the genome. The second region would be the portion of the gene at the left end of the genome, starting at position 1 and ending at the stop codon for the frame. For example, in a 60,000 bp genome, the two regions might be something like 58,734-60,000; and 1-4.

9.5 Predicting tRNA and tmRNA genes

DNA Master’s Auto-Annotate feature runs the tRNA search tool **Aragorn**, which may identify some tRNA genes in your genome. However, the version of Aragorn that is within DNA Master does not call the tRNAs (and their ends) as well as it could. There is a newer, web-based version of Aragorn is the best of the tRNA programs at determining the correct ends of tRNAs. The other web-based program, **tRNAscan-SE**, is useful for finding non-canonical tRNAs as it is possible to relax its search parameters.

9.5.1 Running web-based Aragorn (version 1.2.28)

- Go to: <http://130.235.46.10/ARAGORN/>



The screenshot shows the ARAGORN web interface. On the left is a navigation menu with links for ARAGORN, ARWEN, BRUCE, optalign, RAMI, and External links (tRNAscan-SE, Sprinzl compilation). The main content area is titled 'ARAGORN, tRNA (and tmRNA) detection in nucleotide sequences'. It includes a description of the tool, an 'Input sequence' section with a 'Browse...' button and a genome selection dropdown (currently set to Methanococcus jannaschii), and a 'Select options' section with dropdown menus for 'Search for' (tRNA), 'Search allowing introns' (no), 'Sequence topology' (linear), 'Strand(s)' (both), and 'Output format' (standard). 'Submit' and 'Reset' buttons are at the bottom of the options section. A 'Notes' section at the bottom provides additional information about the tool's capabilities and limitations.

Figure 9.19

- In the 'Input Sequence' section, click 'Browse...' then select your phage's DNA sequence as a FASTA file.
- Choose the following settings:
 - Search For: **tRNA & tmRNA**
 - Search allowing introns: **no**
 - Sequence topology: **circular** (because phage genomes circularize upon infection)
 - Strands: **both**
 - Output format: **standard**
- Click the 'Submit' button.
- Your results will load in a new page. The output includes the secondary structure of the tRNAs found. An example is shown in Figure 9.20.

```
-----  
ARAGORN v1.2      Dean Laslett  
-----
```

Please reference the following paper if you use this program as part of any published research.

Laslett, D. and Canback, B. (2004) ARAGORN, a program for the detection of transfer RNA and transfer-messenger RNA genes in nucleotide sequences. *Nucleic Acids Research*, 32;11-16.

Searching for tRNA genes with no introns
Searching for tmRNA genes
Assuming circular topology, search wraps around ends
Searching both strands
Using standard genetic code

Bongo Complete Sequence, 80228 bp including 11 bp 3' overhang (ACCTCCTGCAA), Cluster M
80228 nucleotides in sequence
Mean G+C content = 61.6%

1.

```
      g  
      c-g  
      t.t  
      c-g  
      a-t  
      c-g  
      g-c      tc  
      t      tgcc a  
gta  g      : : l g  
g    agcg      tgcg c  
c    l:l:l      c  tt  
a    tggc      ggg-c  
atg  a      g  c-g  
      ga a      g-c  
      g.g      g-c  
      g-c      g+t  
      g-c      g+t  
      g-c      g-c  
      a-t      a  a  
      t  t      t  g  
      t  g      t  c  a  
      ccg      tc
```

tRNA-Arg(ccg)
96 bases, %GC = 65.6
Sequence [32355,32450]

Figure 9.20

The principles underlying Aragorn are described in:

Laslett, D. & Canback, B. (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Res.* 32;11-16. [PMID: 14704338](https://pubmed.ncbi.nlm.nih.gov/14704338/)

9.5.2 Running tRNAscan-SE (version 1.21)

- Go to: <http://lowelab.ucsc.edu/tRNAscan-SE/>
- Next to the field labeled “or submit a file”, click the ‘Browse...’ button and select your phage’s DNA sequence as FASTA file.
- Choose the following settings:

Search mode: **Default**

Source: **Bacterial**

A note about settings: For most genomes, these default settings generate reliable results. However, you can always relax the parameters if you come across a suspicious area.

It is recommended that you run tRNAscan-SE using most of the website’s default parameters, with source set to “Bacterial” (as described above) **UNLESS** your phage is a member of Cluster C. To date, only Cluster C mycobacteriophage genomes have been shown to include these non-canonical (or pseudo-) tRNA sequences.

The relaxed settings include changing the Search Mode to “Cove only”, and setting the “Cove score cut-off to “2”, as shown in **Figure 9.21**.

Search Mode: Source:

Format:

Raw Sequence
 Sequence name (optional): (no spaces)

Other (FASTA, GenBank, EMBL, GCG, IG)

Paste your query sequence(s) here:

(Queries are limited to a total of less than 5 million nucleotides at any one time)

or submit a file:

Show results in this browser.
 Receive results by e-mail instead:

Extended Options:

Disable pseudo gene checking
 Display results in ACeDB format
 Show false positives from tRNAscan/EufindtRNA
 Show primary and secondary structure components to Cove scores
 Genetic Code for tRNA Isotype Prediction:
 Default cut-off values should only be changed for exceptional conditions
 Cove score cutoff:
 EufindtRNA search parameters:

Show origin of first-pass hits
 Show codons instead of tRNA anticodons

Intermediate score cutoff:

Figure 9.21

The output from this program looks like the sample in **Figure 9.22**, and when you click on the ‘View tRNA’ button, you will view tRNAscan-SE’s interpretation of the secondary structure (**Figure 9.23**).

Results

Sequence Name	tRNA #	tRNA Begin	Bounds End	tRNA Type	Anti Codon	Intron Begin	Bounds End	Cove Score
Bongo	1	54782	54852	Trp	CCA	0	0	54.55

Figure 9.22

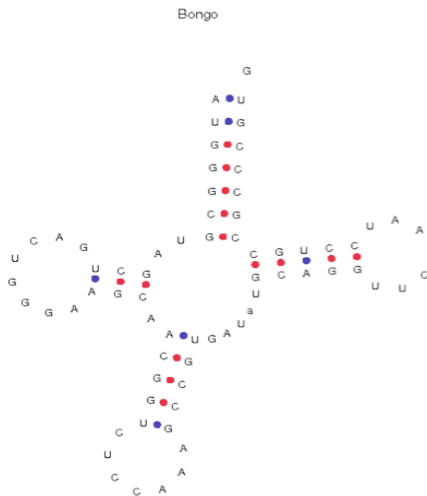


Figure 9.23

The principles underlying the tRNAscan-SE program are described in:

Lowe, T.M. and Eddy, S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*, 25, 955-964.

It is recommended that both tRNAscan-SE and Aragorn be run on every sequence.

9.5.3 tRNA secondary structure and end determination

Some manual checking is required to determine the precise 3' end of a tRNA gene.

In the tRNA schematic below, the 5' end of the tRNA is a 7 base-pair segment called the Acceptor Stem. The remainder of the tRNA is depicted in the diagram; it winds all the way through three additional stem-loops of variable lengths and then back to the matching base pairs of the acceptor stem. Conserved bases are labeled in nucleotide single-letter shorthand at the appropriate position. The tRNA algorithms score potential tRNAs based on their adherence to the conserved bases and stem-loop lengths.

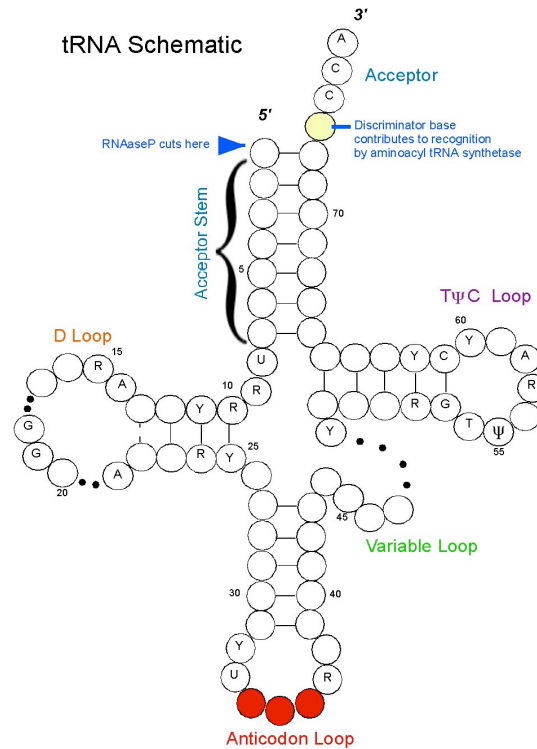


Figure 9.24

After the Acceptor Stem, the 3' end has up to four unpaired bases. The first is called the discriminator base, and it is part of the recognition system that the tRNA synthetase uses to charge the tRNA with the correct amino acid. The discriminator base is followed by the sequence CCA.

The ends of the tRNA must be carefully checked. The acceptor stem loop must be seven base pairs. The CCA sequence at the 3' end must be present on the final tRNA molecule for the tRNA to be charged. Sometimes in the tRNA gene within the DNA of the genome the CCA sequence is truncated, in which case the additional part of the CCA sequence is added after transcription. **Therefore if the 3' end of the sequence is not CCA, it should be trimmed at the first deviation from the CCA sequence, and the remainder should not be included in the gene call.**

The tRNA Schematic shown in **Figure 9.24** is an adaptation of the schematic found on the Lowe website <http://lowelab.ucsc.edu/tRNAscan-SE/> with review and guidance from Dr. Craig L. Peebles.

9.5.4 Entering a tRNA in DNA Master

DNA Master may have already called some of your tRNA genes. If so, go to the **[Feature]** tab and the **[[Description]]** sub-tab, and enter the following information. (See **Figure 9.25** for an example.)

- Type: tRNA (not CDS)
- Start & Stop: Exact coordinates as determined above

- Feature Product: “tRNA _____” (In the blank, write the amino acid 3-letter abbreviation, e.g. “Lys”.)
- Feature Notes and Function: “tRNA _____” (In the blank, write the amino acid 3-letter abbreviation followed by the anti-codon, e.g. “Lys (ttt)”.)

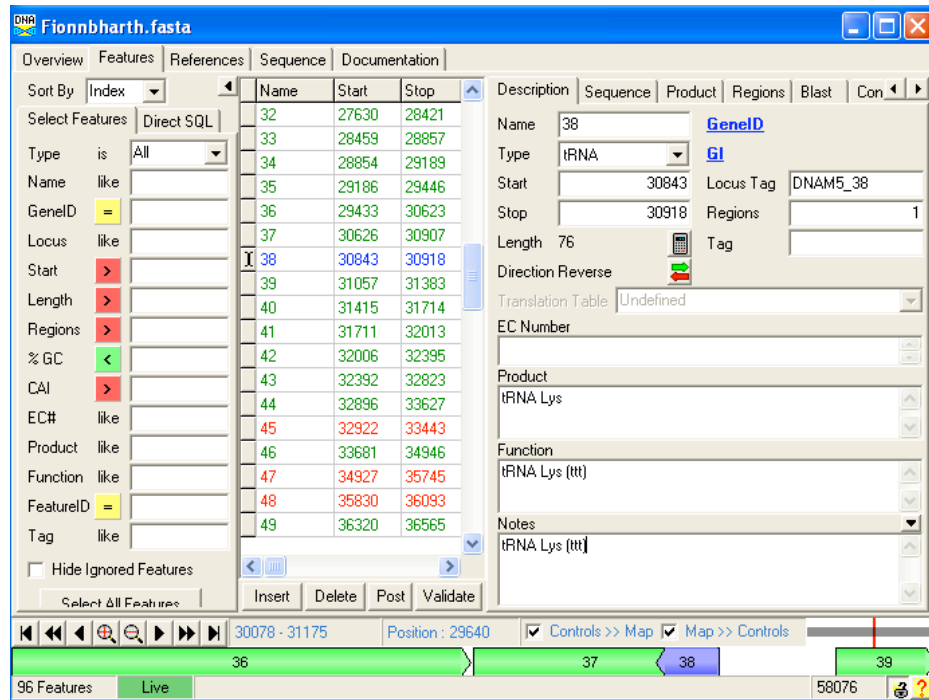


Figure 9.25

If you are adding a brand new tRNA, click the ‘**Insert**’ button at the bottom of the central column. Then enter in the above information in the window that opens and click ‘**Add Feature**’. (You can leave the name blank, and it will be automatically assigned when you renumber genes, as described in **Section 9.3.3**.)

9.5.5 Identifying and annotating tmRNA genes

Description from Wikipedia:

“Transfer-messenger RNA (**tmRNA**) is a bacterial RNA molecule with dual tRNA-like and messenger RNA-like properties. In *trans*-translation, tmRNA and its associated proteins bind to bacterial ribosomes which have stalled in the middle of protein biosynthesis, for example when reaching the end of a messenger RNA which has lost its stop codon. tmRNA can recycle the stalled ribosome, add a proteolysis-inducing tag to the unfinished polypeptide, and facilitate the degradation of the aberrant messenger RNA.”

The coordinates for tmRNAs can be annotated as web-based Aragorn (or the algorithm BRUCE on the Aragorn web page) calls them. Entering tmRNAs into your DNA Master annotation can be done using the same procedure as for entering tRNAs (**Section 9.5.4**), only the “**Type**” of feature in the should then be “tmRNA” (not CDS or tRNA).

9.6 Documenting your gene calls

Just like in at the wet bench, it is important to takes notes and document your findings during genome annotation. While you may want to keep an additional notebook or word document for lengthier rationales or questions, there is a good place to put an abbreviated version of your rationale for each gene in the DNA Master file. In the [Feature] tab and [[Description]] sub-tab, there is a convenient box marked “Notes” that will allow you to do this.

Every gene call should be documented in its Notes as described below. These notes are extremely important for the annotation review process. This is the place where you will want to advocate for those difficult calls. Once checked, these notes will be removed from the GenBank submission file.

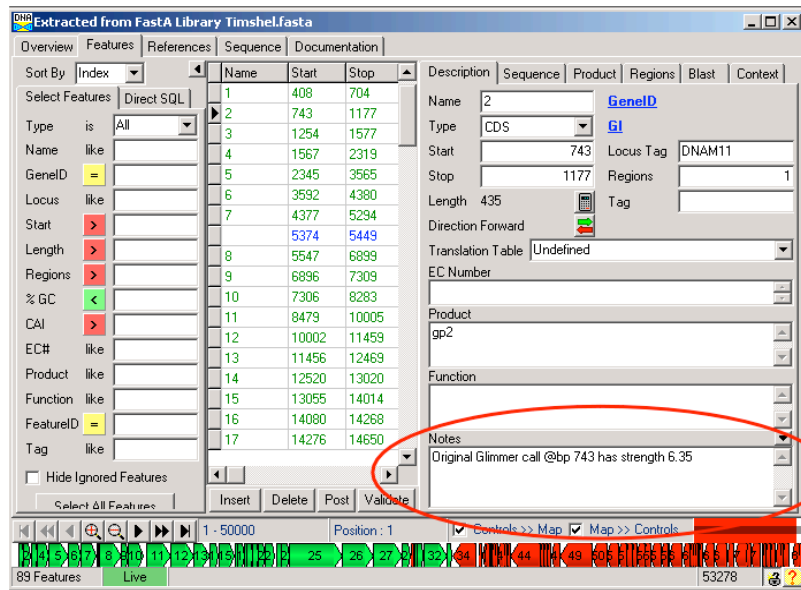


Figure 9.26

To edit the Notes field, simply click within the field and type. Make sure you Post changes (Section 9.3.1) when done so that you don't lose your work. The following information should be recorded for every gene, in order if possible.

- Start/stop coordinates. (This may seem redundant because there are “Start” and “Stop” fields that already contain this information, but it serves as a double-check that all changes you made are actually contained in the final file.)
- Any significant gap or overlap with preceding gene (in basepairs).
- Whether or not the gene was called by Glimmer and GeneMark, and if the start was called by same.
- Whether or not the coordinates you have chosen yield the longest possible gene for that ORF.
- Whether or not your start includes all the coding potential identified by GeneMark.
- Whether or not the start has the best SD score of all this ORF's possible starts.
- The best BLAST match, and the alignment of the gene start with that BLAST match. (For example, “Matches KBG gp32, Query 1 to Subject 1”, or “Aligns with Thibault gp45 q3:s45”.)

- If your gene start does not match the published starts of similar genes in GenBank, an explanation of why not. (“Published Thibault gp45 start not present in my sequence” or “Thibault start caused a 200 bp overlap with upstream gene”)
- Gene Function, and source for the function (see **Section 10**). If the function assignment comes from a Hatfull-approved map in the Appendix, please also enter it into the field labeled “Function” directly above the “Notes” field. Otherwise, only enter the putative functional assignment in the Notes.
- Anything else you think is important. In particular if you made a different choice than previous annotators have made in published genomes, and feel very strongly about your choice, this is the place to let us know.

An example of good Notes:

Start: 2435 Stop: 2650 (FWD). ORF Length: 213 bp; longest possible ORF. SD Score: 310, best score. Gap or Overlap with Previous Gene: 84 bp gap. Gene Predictions: Agrees with both Glimmer and GeneMark predictions. Coding Potential Support: ORF includes all coding potential shown on GeneMark-Smeg output. Best BLAST match: gp3 of Oline; Oline aa 1 aligns with query aa 1. Predicted Function(s): NKF (No Known Function).

10 Assigning gene functions

10.1 Overview

Before the age of bioinformatics, the only way to determine a gene function was to perform wet bench experiments: cloning and expressing a gene, or knocking a gene out, and then characterizing the resulting mutants. These kinds of studies are still the gold standard for determining gene function.

Because of recent advances in sequencing technology, however, we are identifying potential genes far more rapidly than we can perform the supporting wet bench experiments for functional determination. Bioinformatic tools can make some strong predictions through comparative approaches, especially by comparing the sequence of any particular gene to the sequences of genes with known functions (i.e., those that have been characterized experimentally).

Even with the new tools that are available, we are unable to assign functions to the majority of the genes that we annotate in bacteriophage genomes.

There are several categories in which genes can be assigned functions with some confidence.

1. **Virion structural and assembly genes**, i.e. those encoding proteins that are either components of virion particles or assist in their formation. These include genes encoding the terminase, portal, capsid maturation protease, scaffolding proteins, major capsid protein, major tail subunit, tail assembly chaperones, tape measure protein, and minor tail proteins.
2. **Genes involved in phage DNA replication**. These include DNA polymerase, DNA primase, DNA helicase, nucleotide metabolism genes, and ssDNA binding proteins.
3. **Genes involved in life cycle regulation**. These include various regulators such as repressors and activators, integrases, recombination directionality factors, etc.
4. **Genes involved in lysis**, including endolysins (referred to as Lysin A in the mycobacteriophages), Lysin B, and Holins.
5. **Other well-characterized genes**, including transcription factors, toxin/anti-toxin systems, peptidases, phosphatases, host gene homologues, methylases, nucleases, and DNA binding proteins, among others.

Not all phages contain all of the above genes—or at least genes that can be recognized as having these functions (e.g., we still are not sure where the tail assembly chaperones are in the cluster B phages). Even with a substantial body of knowledge about the mycobacteriophages, we can still only assign functions to 10-20% of the genes in a given genome. Remember that it is okay to write “No Known Function” or “NKF” for a gene.

For more information on the specific function of some of the above phage genes as they relate to mycobacteriophages, see:

<http://phagesdb.org/glossary/>

10.2 Using bioinformatic tools to assign gene function

There are three main tools that are useful for predicting potential gene functions. These are:

1. BLASTP
2. Conserved Domain Identification (either through NCBI or Phamerator)
3. HHpred

10.2.1 BLASTP

BLASTP [BLAST (Basic Local Alignment Search Tool) P (Protein)] is a program that searches your query protein sequence against all known predicted protein sequences. You have already come across this in the context of using BLAST to refine your annotations, but it is very useful for predicting potential gene functions.

There are two basic ways of doing BLASTP searches. They can be done within the DNA Master environment, or they can be done using the NCBI BLAST server. The web address for this program is:

<http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins>

When you BLASTP your protein sequence, you are comparing it to all the other protein sequences in GenBank. One important thing to remember is that anyone can submit information to GenBank—whether it is correct and high-quality or not—so any GenBank hits that provide putative gene functions must be carefully considered.

When assigning functions using BLASTP you should consider the following points.

E value. E values are a measure of the likelihood that this alignment would appear at random. Therefore, lower E values are better (less likely to be random) matches. For any potential functional match, the E value should be 10^{-4} or less. This is the perhaps the most important factor to consider, and if this condition is not met, you should not assign a function regardless of what kinds of functions appear in the results list.

The length of the alignment. Does the alignment extend the entire length of your protein? If it only matches a portion, you should interpret this cautiously. For example, if you find a relatively small segment of a protein that matches others at a statistically significant level, you may want to consider annotating this as a domain rather than a full protein function. For example, if a small segment of your protein matches other proteases, you might want to consider writing “peptidase domain”, rather than “peptidase” in your Notes.

Likelihood of the proposed match. Even if you have an exact match to a piece of a protein in *Vitis vinifera*, it is pretty unlikely that a protein from grapes has the same sequence and function as a protein in a mycobacteriophage. Most of the time when BLASTP aligns bacteriophage proteins with eukaryotic proteins, the alignment is occurring between repetitive sequences, rather than the functional domains of the protein.

Figure 10.1 is an example of a good BLASTP match, generated using NCBI’s web-based BLASTP, where a putative function can be assigned.

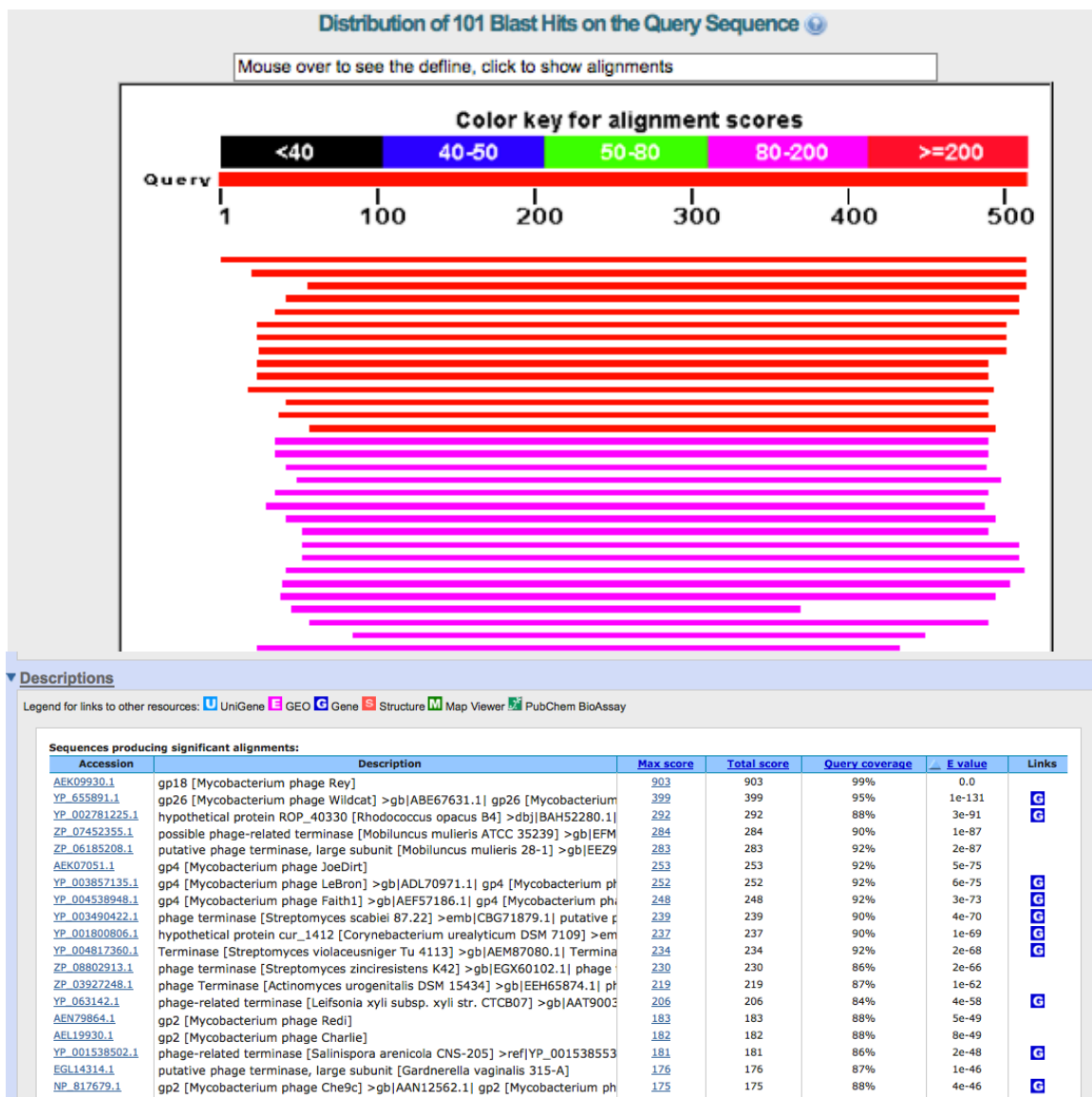


Figure 10.1

In the graphical portion of the results, there are many matches in red (the color for the highest match scores) that extend over the entire length of our query sequence. In the list of matches, we can see that all of the E values are well below 10^{-4} . And many of the hits have a Description that involves terminases. We can now say, with some confidence, that the protein we BLASTed is a terminase.

10.2.2 Conserved Domain Database

When you run your protein sequence through BLASTP on the NCBI webpage, one of the default settings is to examine your protein sequence for conserved domains. Conserved domains are smaller shorter amino acid sequences that are usually affiliated with a specific part (or domain) of a protein. These conserved domains also appear on Phamerator maps as yellow boxes *within* a gene's colored box.

If you have a conserved domain detected within your protein, the function assigned to the domain will be frequently—but not always—be similar to ones found in BLASTP matches.

Useful domains to indicate in your annotation are things like peptidases or phosphoesterases, but there are a wide variety that may appear.

Not all conserved domains will be useful. Some contain little information, such as “Conserved domain of unknown function, found in bacteriophages”. Others are false positives such as the “Structural maintenance of chromosomes” domain that often appears in structural proteins. Unfortunately, it is not clear *a priori* which are false fits and which are reliable. Consideration of the genomic context as well as the HHpred search described below are perhaps the most reliable indicators.

An example of a reliable Conserved Domain hit reported by BLASTP on the NCBI server might look like: If you hover your cursor over these boxes with the mouse, a pop-up window will appear that tells you about the conserved domain.

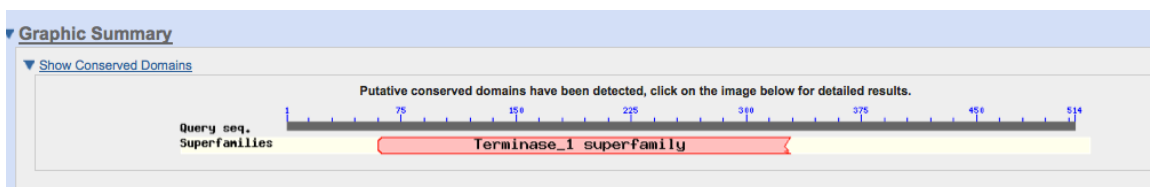


Figure 10.2

The same gene in a Phamerator map might look like:

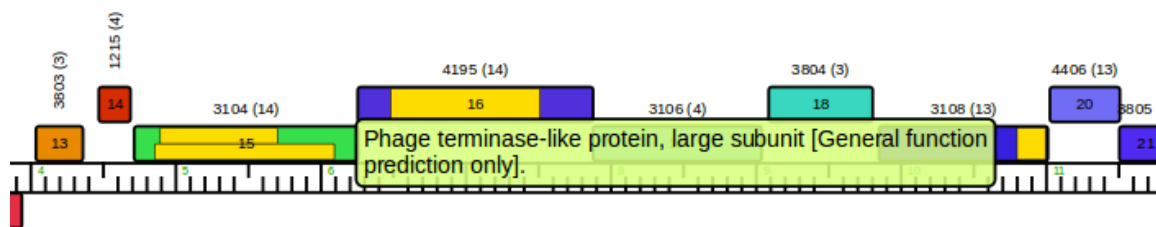


Figure 10.3

In this case, we moused over gene 15 in Figure 10.3, and the green box describing the domain appeared.

A less informative match on NCBI might look like:

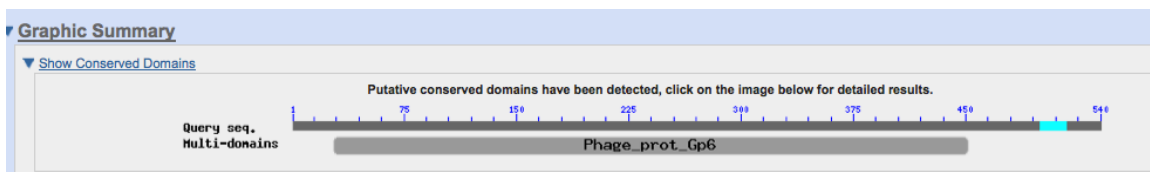


Figure 10.4

We already know that this is a phage protein, so this is not particularly useful information.

And the same gene in Phamerator:

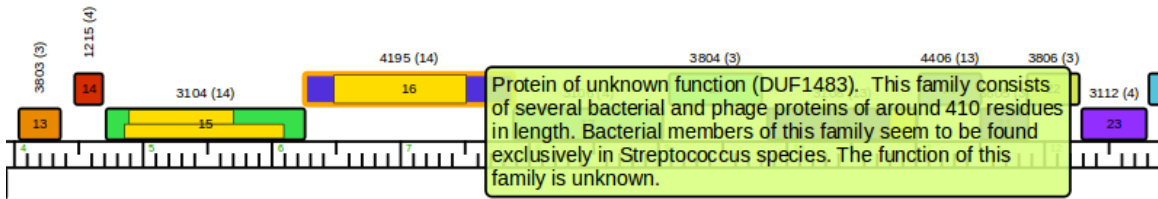


Figure 10.5

In this case, we moused over gene **16** in the above map, which is the well-characterized portal protein (shown in BLASTP hits). Based on the notes in the green box, we see that the Conserved Domain Database does not know that this is the portal protein. This is an example of the dependence of GenBank on its authors, who may not be as informed as they should be.

10.2.3 HHpred

HHpred is essentially a more sensitive way of searching for functions than BLASTP. In detail:

HHpred performs an iterated multiple sequence alignment using your query amino acid sequence and its best GenBank matches, using either PSI-BLAST or HHblits (Homology detection by iterative HMM-HMM comparison). It then builds a Hidden Markov Model (HMM) based on the alignment, and compares this model to HMMs based on the Protein DataBank (PDB) (which contains crystal structure coordinates for crystallized proteins). By comparing conserved residues to a 3-D coordinate map, we can sometimes detect and assign gene functions to genes that have very few informative matches using BLAST.

For more information about the design, abilities, and bioinformatics of HHpred, see:

http://toolkit.tuebingen.mpg.de/hhpred/help_ov

HHpred is accessible at:

<http://toolkit.tuebingen.mpg.de/hhpred>

Like BLAST, some matches in HHpred are very useful while others are more likely to be false positives.

An example of an informative HHpred match:

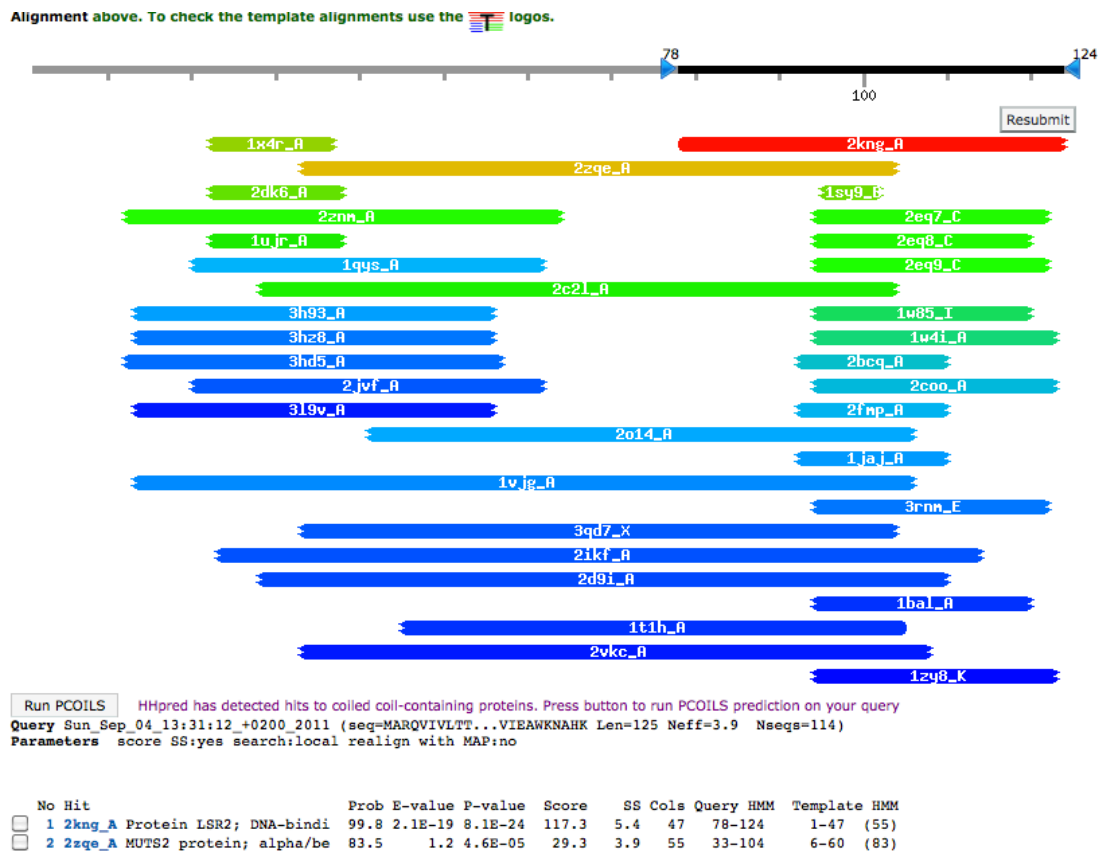


Figure 10.6

Like BLAST, HHpred provides a graphical view where the best matches are shown in red and lower-quality matches are dark blue or black. Also like BLAST, below the graphical representation is a list with useful information, including the score each hit gets. In the above screenshot, the best hit, "2kng_A" (this is the PDB designator, if you want to see the crystal structure), matches your protein with 99.8 % probability and an E value of 2.1×10^{-19} .

Good HHpred matches have high probabilities (80 or above), and low E values (the lower the better). The scientists who wrote HHpred claim that matches with probabilities above 30% might be real matches. However, if you are going to claim a function found in HHpred with a probability between 30 and 80%, supporting data (such as a conservation of a domain, or a function found in other mycobacteriophages) is necessary.

For more on determining if your HHpred hit is a real match, see:

http://toolkit.tuebingen.mpg.de/hhpred/help_faq#correct%20match

When we scroll down to the look at the specifics of the alignments, we see:

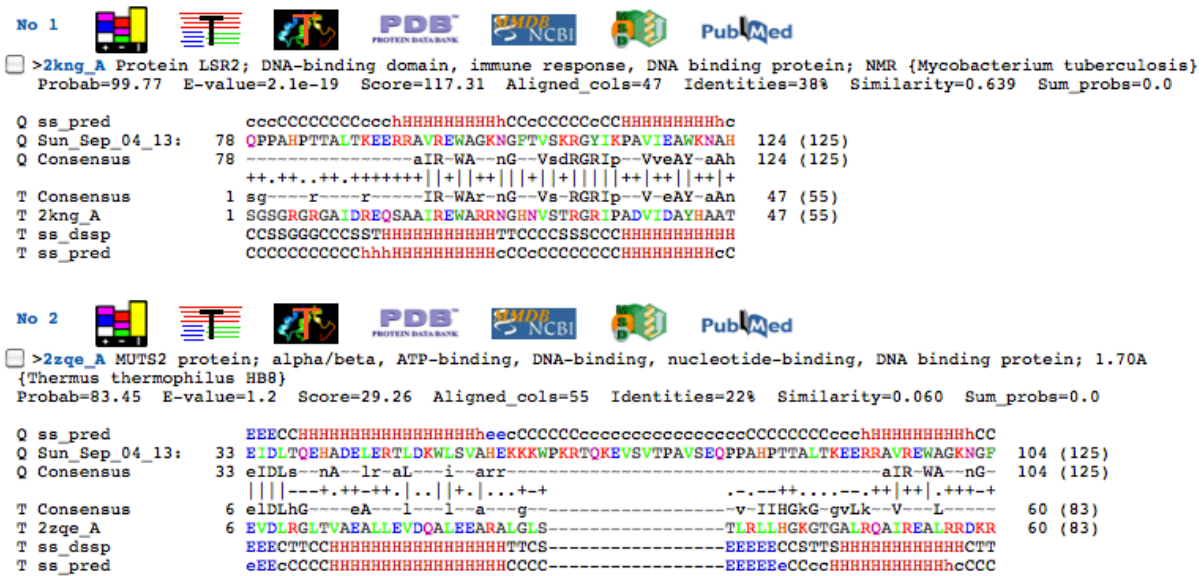


Figure 10.7

The first line of text describes what “2kng_A” is: the LSR2 protein from *M. tuberculosis*. This is encouraging in two ways: first, it matches a mycobacterial protein; and second, similar mycobacterial proteins have already been found in other mycobacteriophages.

Under the line of text is the alignment. The top line is the secondary structure prediction for your query sequence, the second line is your query sequence, and the third line is the consensus sequence that was built from the iterative search. The bottom part of the alignment refers to the subject sequences (in this case the crystal structure data), their consensus sequence, and secondary structure prediction or determination from two algorithms: PSI-PRED, and DSSP.

For more on interpreting HHpred results, see:

http://toolkit.tuebingen.mpg.de/hhpred/help_results

Another example of an informative report from HHpred is below.

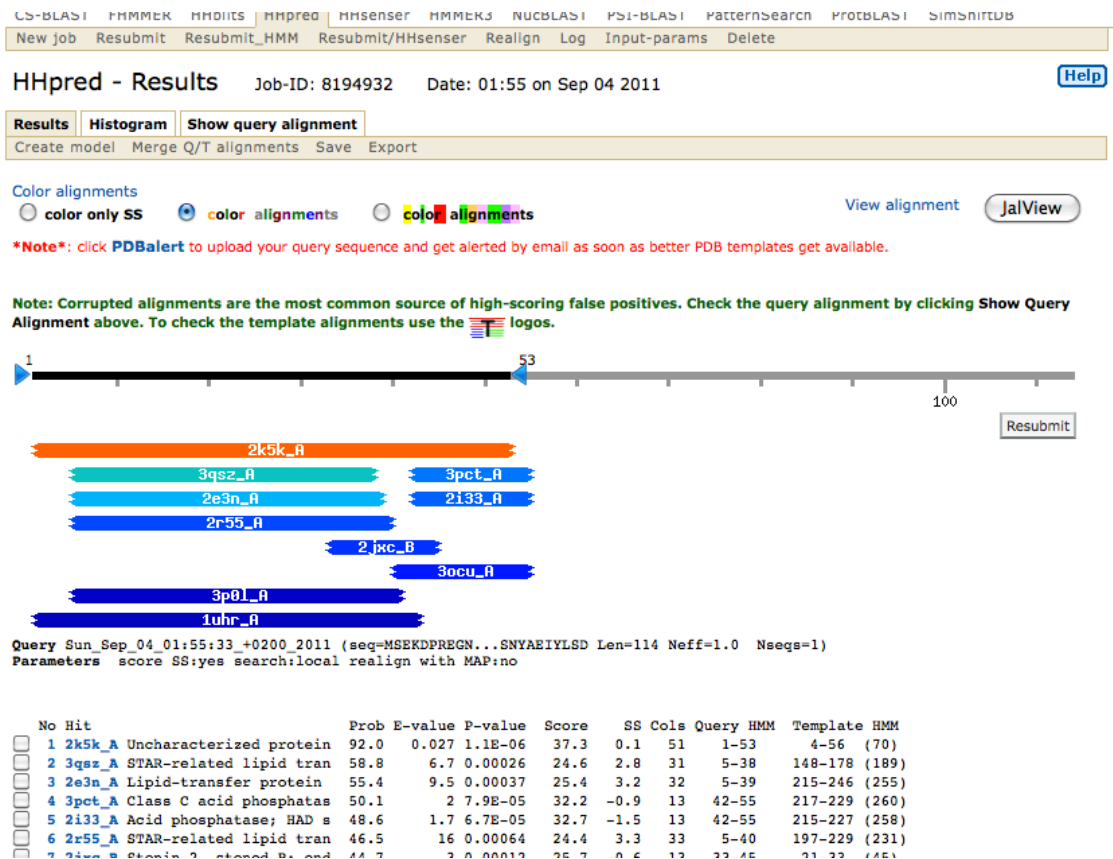


Figure 10.8

The top hit, to 2k5k_A, has a Probability score over 90, and it is an uncharacterized protein. The rest of the matches have low probabilities (80 or below), and high E values. So even though the other matches are to phosphatases, and one might be tempted to write “phosphatase”, this would not be a supportable functional prediction for this protein.

10.3 Other ways to assign gene function

10.3.1 Synteny

Many of the genes in bacteriophage genomes—but especially in the virion structure and assembly genes—appear in the same order (synteny). Therefore, sometimes functions can be inferred from gene order. The typical order is:

Terminase → Portal → Capsid Maturation Protease → Scaffolding → Major Capsid Subunit → Major Tail Subunit → Tail Assembly Chaperones → Tape measure → Minor Tail Proteins

Sometimes other smaller genes of unknown functions are interspersed within the structural genes, but in general the overall order remains conserved. While we may see conservation of gene order in some other areas of phage genomes, these other areas are far more mosaic than the structural genes are, and so the use of a synteny argument applies primarily when assigning gene function to the virion structure and assembly genes.

The longest gene in the genome of a phage with a flexible tail is almost always the tape measure protein gene. This gene is directly proportional to the length of the tail in the flexible-tailed phages.

10.3.2 Prior functional assignments

Many of the genes within the previously sequenced mycobacteriophages have already been assigned functions based on experiments, BLAST and/or HHpred matches, or synteny. Dr. Hatfull periodically reviews the mycobacteriophage genomes and assigns gene functions to the best of his knowledge. If you are trying to assign a function to a gene that has a BLAST match to or is in the same pham as one of the genes with an assigned function on one of his maps, you may assign your gene the same function. The most recent version of these maps, one per cluster, are included in the Appendices of this guide.

10.3.3 Phamerator

Many of the genomes in Phamerator have already been published according to the most recent functional assignments, but not all. We are constantly in the process of improving our gene calls, and so Phamerator functional assignments reflect our best effort at assigning gene functions **at the time the genome was entered** into Phamerator. This means that many of the more recent genomes might have better functional assignments than some of the older ones. If you're using comparisons in Phamerator to already-published genomes to determine function, try to compare to recently-published genomes.

11 Merging and checking annotations

11.1 Merging overview

In a classroom setting, different portions of a genome are often assigned to different students or groups of students to annotate. Once all portions have been annotated, they must be combined into a single file, and the “**Merge**” function in DNA Master performs this action. It takes multiple files from a single phage genome and creates a single master file that contains all of the gene calls from each individual file.

Note: merging will **only work on files that contain identical sequences**. If you are going to split a genome among different annotation groups, make sure that you keep the entire sequence intact, and simply work on a region identified by gene coordinates (e.g. between 20,000 and 30,000).

Typically, you’ll merge all of a given genome’s partial annotations together into a single file that can then be proofed and edited to become the final complete annotation. However, it is also possible to do several iterations of merging. For example, if two groups are working on the region from 10,000 to 20,000, you may want to merge their files first, come to a consensus on that region, then merge the newly checked version with the other final files from other sections of the genome. Merging is flexible enough to meet your pedagogical goals.

11.2 Merging multiple annotations into a single file

- Collect the files you’d like to merge into a single directory. Remember that these must all be from an identical DNA sequence (i.e., the same phage genome).
- Open DNA Master.
- Go to **File** → **Merge**

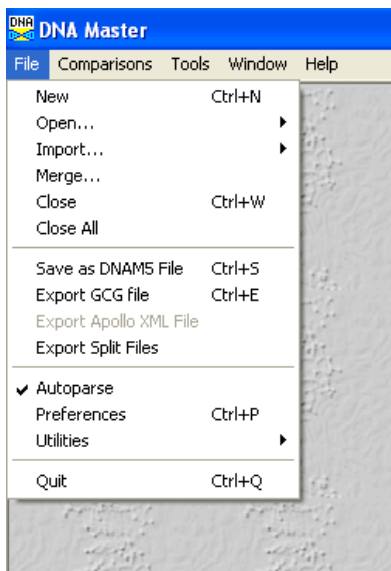


Figure 11.1

- A new window will open, as shown below.

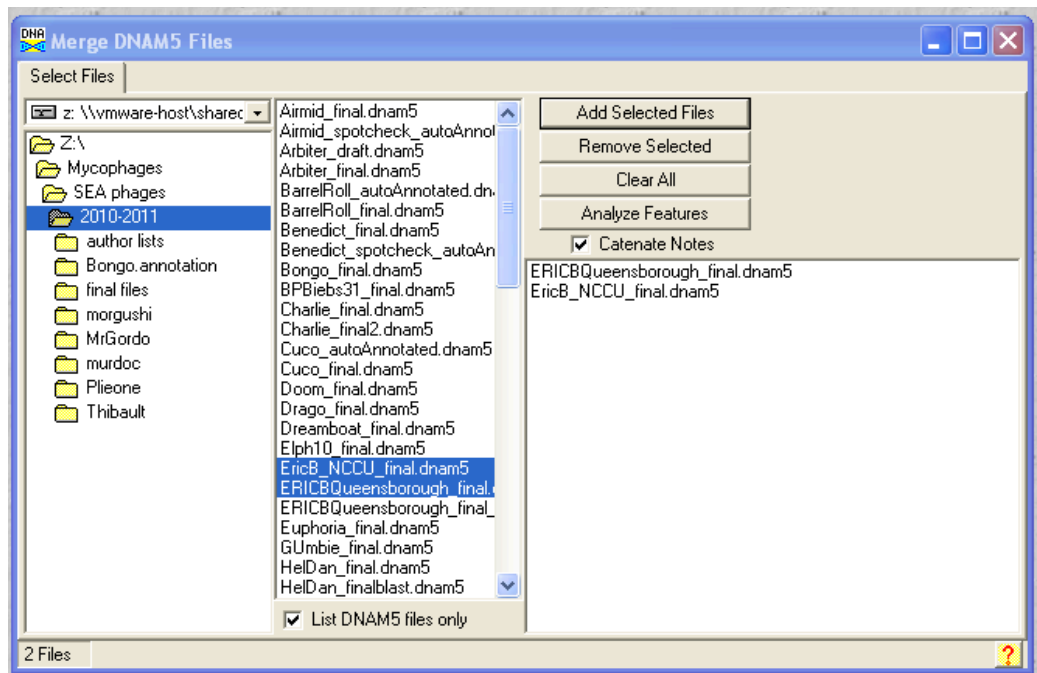


Figure 11.2

- In the left column, browse to the directory on your computer that contains the DNA Master (.dnam5) files that you want to merge.
- In the center column, click on files that you want to add to your merged file.
- Click the '**Add Selected Files**' button. The files will then appear in the empty white box on the right. You can browse to additional directories (if necessary) to add additional files.
- Once you all the files that you would like to merge are listed in the white, check the box marked "**Catenate Notes**".
- Click the '**Analyze Features**' button.
- The window will open a new tab, [**Merge Files**].

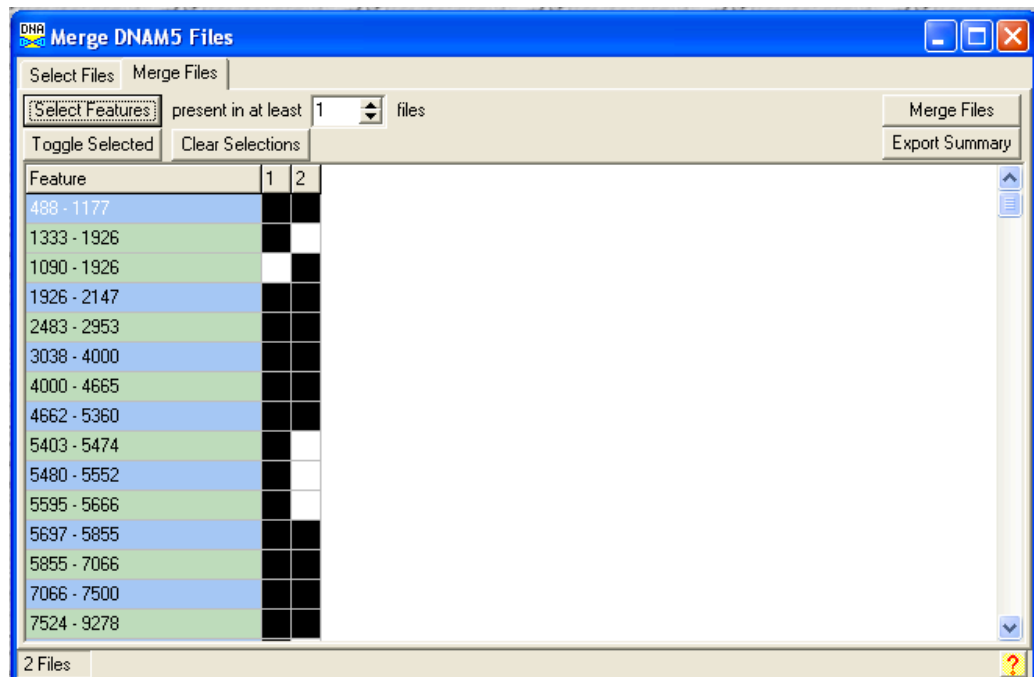


Figure 11.3

In the picture above, Features (or gene calls) are listed according to genome coordinates. Each file you selected is represented by a numbered column, displayed in the order that they were selected in the previous tab.

In each row, a black box is present if that file contains that feature, and a white box is present if the file does not contain that feature. The first feature, 488-1177, is present in both of the files that were merged. The next feature, from 1333-1926, was present only in the first file. The third feature, from 1090-1926, was present only in the second file. Because both of these features have the same stop codon, what we are looking at is a disagreement in the two files about where the start for this gene should be. File 1 calls it at 1333, while file 2 calls it at 1090.

- To export a spreadsheet that contains the above information (which can be useful to identify areas of disagreement that require further attention), click the **'Export Summary'** button in the top right of this window.

To create a .dnam5 file with all of the gene calls from the files to be merged:

- Click the **'Select Features'** button. (Selected features will turn red, as shown below.)

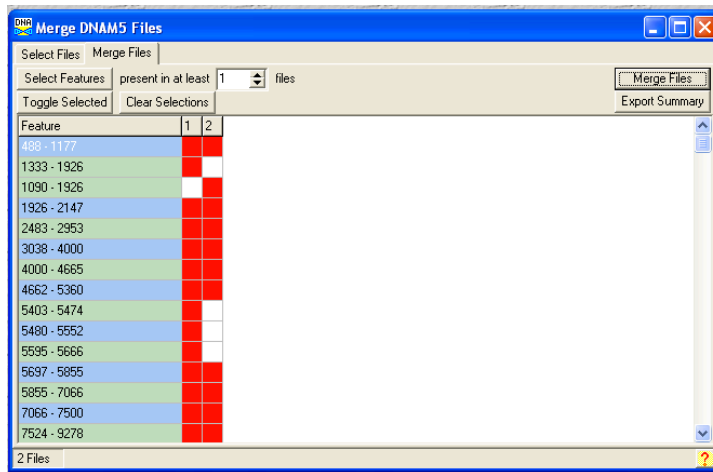


Figure 11.4

- You can tailor your selection by modifying the number in the dropdown box next to “present in at least ___ files”. After changing the number, click the ‘**Clear Selections**’ button to erase previously selected genes, then click the ‘**Select Features**’ button again to make your new selection. In the picture below, now only the features present in at least two (both) files are selected and shown in red.

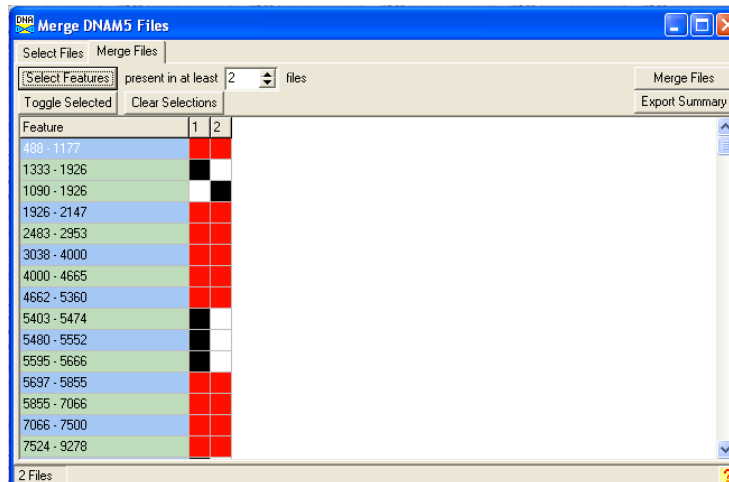


Figure 11.5

- Once you have selected the features you would like in your merged file (picking all of them is a good choice, disagreeing features can always be deleted from the merged file after review), click the ‘**Merge Files**’ button at the upper right corner.
- A new window titled ‘**Merged Sequence**’ will appear, as shown below.

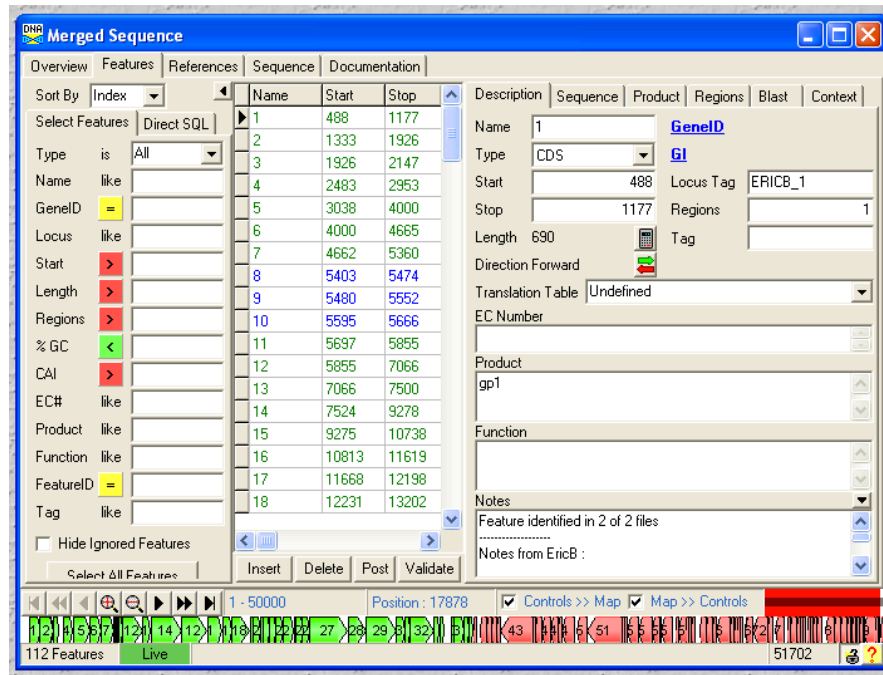


Figure 11.6

- Save your file immediately by going to: **File** → **Save as DNAM5 File**
- Select a meaningful name for the merged file, such as “YourPhageName_Merged.dnam5”.

In the above picture, we are looking at feature 1. In the “Notes” field on the lower right, the top line indicates that this feature was called in 2 of 2 files. Further down in the Notes box, both sets of notes have been concatenated.

How features and notes are reconciled when there is disagreement:

While all the genes from the unmerged files will be present within the features of the merged file, DNA Master will not treat all these genes equally. Features that share the same stop codon but have different start codons will be listed as separate features in the merged feature list. Features that were selected by the majority of the files in the merge will be given preference in the merged file, and will be listed first in the feature table if it is sorted by Index.

The most popular features will have concatenated notes. That is, all the notes from the unmerged files will be listed in the Notes field of the merged feature. Less popular features will be in the merged file, but will be listed at the end of the feature list when sorted by Index. Less popular features will have their original notes, not merged notes.

- To clearly see discrepant calls, go to the “Sort By” drop-down menu at the top left of the [Feature] tab, and select “Start” rather than “Index”.

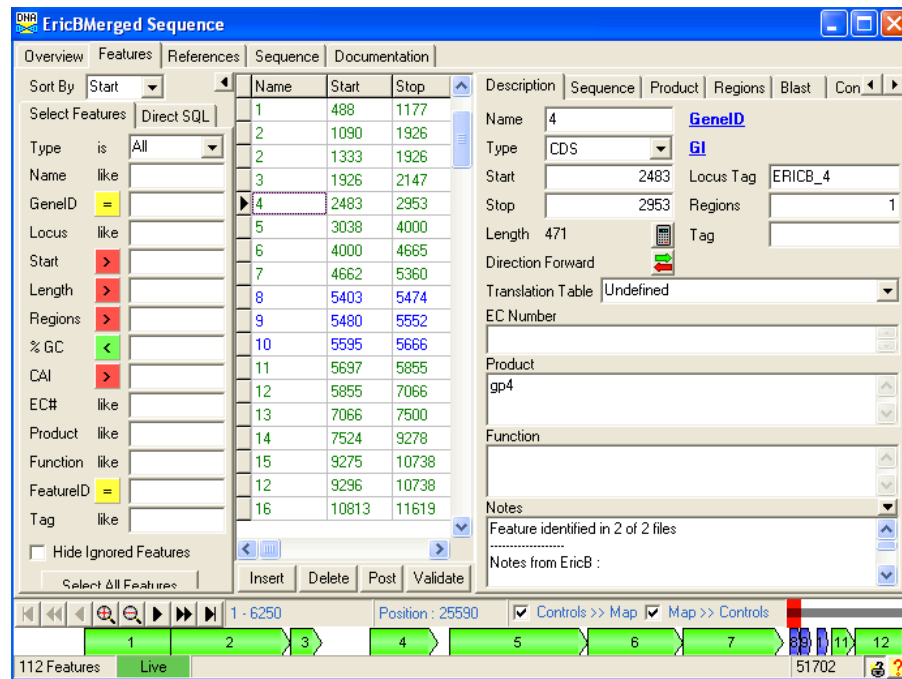


Figure 11.7

You can see that there are two versions of gene 2, one from each file, that share the same stop codon but differ in their choice of start codon. Now it's up to you to determine which is correct!

11.3 Checking an annotation

Once you've merged all files, made final decisions on each gene, and believe you've finished your annotation, there are a few final steps to take before submitting your genome for review and then GenBank submission. These steps below reflect what we typically do at the University of Pittsburgh to quality-control submitted annotations, so you can stay one step ahead and try to identify any remaining issues first.

- Click the 'Validate' button bottom of the central column in the [Feature] tab. The response should be "All ORFs appear valid." If you get a different message here, check the gene(s) identified for errors.
- Zoom in on the interactive map along the bottom of the sequence, and carefully scroll along the whole length of the genome. Do all the genes seem to be tightly packed? Look for large overlaps, gaps, or duplications.
- Open an interactive Phamerator map of your phage along with two or three closely related cluster members that are already in GenBank. (Remember that it is still your auto-annotated genome in Phamerator.) Are there any areas where your phage has orphans (white boxes) or otherwise diverges from similar phages that you have **not** addressed during your refinement?
- Create a "Genome profile". This is a spreadsheet (.csv format) of all the information in the Features table. While this won't give you any new information compared to simply scrolling through your features, it may help you make sure you don't miss anything.

Go to: **Genome** → **Profile**

In the window that opens, there are a number of settings. The default settings should be fine, but consider checking the “Export Notes” box if you’d like Notes included in your spreadsheet, and consider unchecking the “Load into Excel” box if you don’t have Excel or would like to open the file later.

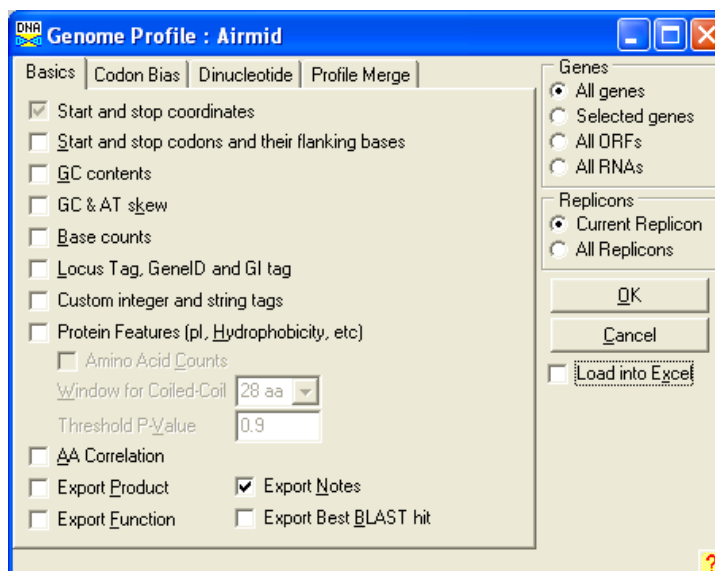


Figure 11.8

- Now check each gene individually.

Read the comments, and consider: Do the start and stop coordinates listed match the coordinates in the file? Does the gene have Glimmer/GeneMark support? A good RBS/Shine-Dalgarno score? Include all the GeneMark-Smeg coding potential? Is the gene as long as possible without overlapping the previous gene too much? Match its best BLAST hit 1-to-1? If the phage has close relatives in GenBank (you can tell pretty quickly by using Phamerator), our frequent default position is to make a newly annotated gene match the annotated genes already in GenBank. If it doesn’t, use your best judgment based on the other metrics.

Check the gene functions, and consider: Do they make sense? Are reported E values low (below 10^{-4})? Do they match the Hatfull-approved maps (where appropriate)? Is there a source listed for a function (HHpred, BLASTP, CDD, GFHmap, other)? If there is no known function, is “NKF” written?

When checking tRNAs, consider: Is the tRNA amino acid and anti-codon written in the notes and in the function boxes? Does the tRNA end with “CCA”, and if not is it trimmed correctly?

For gaps in your gene calls, consider: Is there an ORF with coding potential that was missed? Are there any BLASTX hits with good GenBank matches?

Keep track of any potential issues you encounter during checking, and revisit those areas of the genome to ensure the best call has been made.

12 Submitting final files for review and GenBank submission

You've made it. Plowed through gene after gene, pored over BLAST results and coding potential diagrams, perhaps argued over some start sites, and have merged all calls and come up with a final annotation. Congratulations!

The next step is to submit your files for expert QC and GenBank submission. Read below to make sure that your files are ready for submission, then submit a final DNA Master (.dnam5) file and a final Author List via e-mail to:

phage.submission@gmail.com.

After expert review, your annotation will be either accepted or returned. If accepted we will provide a GenBank flat file for your inspection. If not accepted, your file will be returned with an explanation and request for revisions.

12.1 Details of your final DNA Master (.dnam5) file

A final .dnam5 file is one that has the following properties.

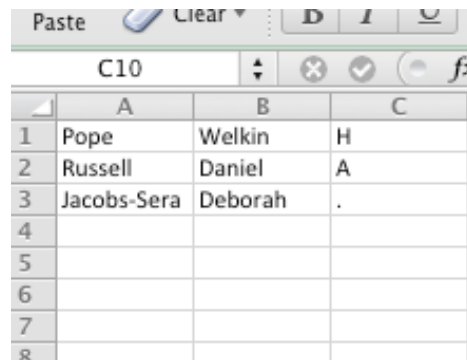
1. It must be named "YourPhageName_Final.dnam5", which will help distinguish it from other versions you may have been working on.
2. **It must contain one entry and set of notes per feature.** That means that if you have merged multiple files, you need to have evaluated the data from each source, come to a decision, and deleted erroneous versions of each feature. There should also be only **one set of notes** for each feature, and it should contain **everything** listed in **Section 9.6** about proper documentation of your gene calls. You may have to delete some notes, or even rewrite some notes from scratch to meet this criterion.
3. All features must be validated (**Section 9.3.2**).
4. All features must be re-numbered if necessary (**Section 9.3.3**).
5. All features must be re-BLASTed (**Section 9.3.4**).
6. Any functions are noted in the "Function" field as well as the "Notes" field.

12.2 Details of your author list

Please create a list (.csv formatted file) of the authors from your school who are to be included in this GenBank submission. Your author list should meet the following criteria.

- It contains **ONLY** authors from your school who deserve to be listed on the GenBank file. **Do not** include names from Pitt, HHMI, sequencing centers, or any other source.
- It is a .csv file. A .csv formatted file can be created in Excel, using the 'Save as...' function, and selecting .csv as the file type.
- It contains exactly three columns, with **NO HEADERS** at the top of each column.

- The first column contains the last name, the second column contains first name, and the third column contains a middle initial. **If no middle initial is needed, type a period in that column instead. All three columns should contain some information for each author.** See below for an example.



	A	B	C
1	Pope	Welkin	H
2	Russell	Daniel	A
3	Jacobs-Sera	Deborah	.
4			
5			
6			
7			
8			

Figure 12.1

Acknowledgements

DNA Master was designed and developed by Dr. Jeffrey G. Lawrence at the University of Pittsburgh. The program has gone through a multitude of advances, some of which were implemented by Dr. Adam Retchless when he was a graduate student with Jeffrey. Dr. Lawrence continues to provide support, updates and new functionalities to DNA Master.

DNA Master is much more than a genome annotation tool, although this is its main role in this guide. DNA Master has been developed for assisting in bioinformatic dissection of genomes – primarily microbial – with a view to understanding how they have evolved and how they are related. As you become familiar with the program and develop your own interests in genome evolution, we hope these utilities will be of use to you.

We are deeply grateful to Dr. Lawrence for making DNA Master available to us and for his constant willingness to listen to our suggestions and our particular needs. Over many years we have found DNA Master to be an incredibly effective platform for genome annotation and analysis, and Jeffrey's contributions cannot be overestimated.

We would also like to thank the literally hundreds of students and faculty who have used DNA Master and provided feedback that has helped us to develop and refine this annotation platform.

We thank our colleagues in the Science Education Program at HHMI, especially David Asai, Kevin Bradley, Lu Barker, Razi Khaja, and Melvina Lewis, for their tremendous insights and feedback. Melvina F. Lewis provided the terrific cover design.

See <http://phagesdb.org/DNAMaster> for **.pdfs** of all Appendices

Appendix I

System requirements and Installation of DNA Master

Appendix II

DNA Master Quick Start Guide

Appendix III

Gene Function with Bench Support and References

Appendix IV

Hatfull Genome Maps

Appendix V

Case Study: The Annotation of Etudee

Appendix I

System requirements and DNA Master installation

DNA Master Installation Guide

1. Minimum System Requirements for Installation of DNA Master

PC Minimum Requirements

OS: Windows XP/Vista/7 32-bit or 64-bit

CPU: Dual-core Processor 1.8GHz

Memory (RAM):

- XP: 1GB
- Vista/7: 2GB

Video Memory: 128MB

Free Disk Space: 5GB

DVD Drive

INTERNET CONNECTION

FULL ADMINISTRATOR RIGHTS

Mac Minimum Requirements

OS: Mac OS X 10.5 or Higher

CPU: Dual-core Intel Processor 1.8GHz

- Non-Intel Macs are NOT supported.

Memory (RAM):

- 2GB

Video Memory: 128MB

Free Disk Space: 25GB

DVD Drive

INTERNET CONNECTION

FULL ADMINISTRATOR RIGHTS

2. Installing DNAMaster on a Windows Computer

IMPORTANT: For Vista/Windows 7 users, this program must have **full administrative rights**. It is **not** sufficient to install this program on a User account with administrative-level rights, you must specific that the program has these rights too. During installation or when starting the program once installed, press “**Yes**” or “**OK**” if you are prompted to allow the program to continue, failing to do so **WILL NOT** allow the program to run properly and **WILL** cause errors. One easy way to make sure the program always has admin rights is to create a short-cut for the program as outlined below. If the program is then **ALWAYS** run from this short-cut, odd error messages should not occur. Otherwise, when starting the program, right-click on the program icon, and select “Run as administrator” every time.

DNA Master Installation

-DNA Master can be downloaded at the following link:

<http://cobamide2.bio.pitt.edu/computer.htm>

-Double click the installer, and follow the instructions to install the program.

Shortcut Creation

-Navigate to the DNA Master directory:

32-bit OS: My Computer → C:/Program Files/DNA Master

64-bit OS: My Computer → C:/Program Files(x86)/DNA Master

-Right-click on “DNAMas.exe” and select “Create Shortcut”

Windows XP: Drag the shortcut to the desktop.

Windows Vista/7: Click “Yes” on the dialogue box.

-Windows Vista/7 Users Only—MUST confer admin rights to the program:

Right Click on the newly created desktop shortcut:

-Click properties

-Click on the “Compatibility” tab

-Check the box next to “Run as Administrator” under “Privilege Level” at the bottom

-Click “OK”

Updating – INTERNET CONNECTION REQUIRED

- Double click your DNA Master shortcut.
- Go to **Help → Update DNA Master**
- Allow the update to run and restart the program.
- DNA Master is now updated and ready to use.

DNA Master is now installed and up-to-date. You can run it through the desktop shortcut.

3. Installing DNA Master on a Mac

To install DNAMaster on your Mac, you will need to install Windows as a second operating system on an emulator such as VirtualBox (<http://www.virtualbox.org/wiki/Downloads>). If you already have access to Windows on your Mac, skip down to “DNA Master Installation”. Once you have downloaded and installed VirtualBox, as per the VirtualBox website instructions, install Windows as below:

Windows Image Installation

- Obtain a copy of Windows XP, Vista, or 7 32-bit edition.
Windows XP is the cheapest option, and requires the least resources.
- Open VirtualBox.
- Click “New” to create a new virtual machine.
- Click “Continue”
- Name the machine “Windows”
- Under the “Operating System” menu choose “Microsoft Windows”
- Under the “Version” menu choose the version you are installing
- Click “Continue”
- For Windows XP, allocate at least 512MB memory, Vista/7 requires >1GB.
- Click “Continue”
- Check the box for “Boot Hard Disk”
- Click the “Create new hard disk” option.
- Click “Continue”
- Click “Continue”
- Click “Dynamically expanding storage”
- Click “Continue”
- Set the starting size of the virtual storage using the slider to a minimum of 20GB.
- Click “Continue”
- Click “Finish”
- Click “Finish”
- Now start the machine by selecting it from the list to the left and clicking “Start”
- Click “Continue” when the first run wizard starts
- Insert your Windows Installation Disk. Ignore any autorun prompts.
- The dialogue in the middle should read “Host Drive” followed by a letter.
- Click “Continue”
- Click “Finish”
- Follow the instructions on-screen to install your copy of Windows.
- After the installation completes, close the virtual machine.
Click “Machine”
Click “Shut Down”
- In the main window, click the settings button.

- Click on the display tab.
- Set the “Video Memory” to 64MB.
- Save the changes.
- Select the new machine in the main window and click “Start” to run it!

*Note: It is recommended that you update your copy of Windows before continuing

DNA Master Installation (Within your Windows Virtual Machine)

- DNA Master can be downloaded at the following link:
<http://cobamide2.bio.pitt.edu/computer.htm>
- Double click the installer, and follow the instructions to install the program.

IMPORTANT: For Vista/Windows 7 users, this program must have **full administrative rights**. It is **not** sufficient to install this program on a User account with administrative-level rights, you must specific that the program has these rights too. During installation or when starting the program once installed, press “Yes” or “OK” if you are prompted to allow the program to continue, failing to do so WILL NOT allow the program to run properly and WILL cause errors. One easy way to make sure the program always has admin rights is to create a short-cut for the program as outlined below. If the program is then **ALWAYS** run from this short-cut, odd error messages should not occur. Otherwise, when starting the program, right-click on the program icon, and select “Run as administrator” every time.

Shortcut Creation

- Navigate to the DNA Master directory:
 - 32-bit OS: My Computer -> C:/Program Files/DNA Master
 - 64-bit OS: My Computer -> C:/Program Files(x86)/DNA Master
- Right click on “DNAMas.exe” and click “Create Shortcut”:
 - Windows XP: Drag the shortcut to the desktop.
 - Windows Vista/7: Click “Yes” on the dialogue box.
- Windows Vista/7 Users Only—MUST confer admin rights to the program:**
 - Right Click on the newly created desktop shortcut:
 - Click properties
 - Click on the “Compatibility” tab
 - Check the box next to “Run as Administrator” under “Privilege Level” at the bottom
 - Click “OK”

Updating – INTERNET CONNECTION REQUIRED

- Double click your DNA Master shortcut.
- Go to **Help → Update DNA Master**
- Allow the update to run and restart the program.
- DNA Master is now updated and ready to use.

DNA Master is now installed and up-to-date. You can run it through your desktop shortcut.

Appendix II

DNA Master Quick Start Guide

DNA Master Quick Start Guide

Part 1: Creating a Draft Annotation

Setting Key Preferences

1. From the **File** menu, select **Preferences** to open the preference window.
2. Click on the **Local Settings** tab, then on the **Colors** sub-tab.
3. In the **LEFT** box, change the display colors for tRNAs, tmRNAs, and ORFs by clicking on the appropriate boxes. The default is black, but we strongly recommend the colors shown in **Figure 1**.
4. Still under the **Local Settings** tab, click on the **Codons** sub-tab.
5. If the box labeled “Use TTG start codons” is not checked, then check it.
6. Click the **Apply** button on the right to save changes, then **OK** to exit.

Importing a DNA Sequence

1. Verify that your DNA sequence, in fasta format, is saved in a known location (or you may download a fasta file from phagesdb.org).
NOTE: If using Virtual Box or another emulator to run Windows, you should copy the fasta file to the virtual machine desktop before proceeding.
2. From the **File** menu, select **Open** → **FastA Multiple Sequence File** as shown in **Figure 2**.
3. Browse to your genome’s fasta-formatted file, then click **Open**.
4. A window like the one shown in **Figure 3** should appear. Click on the button in the lower right hand corner that looks like a piece of paper with the upper right corner folded over, then select “Create Sequence from this entry only” and a new window titled “Extracted from YourPhage.fasta” will open within DNA Master.

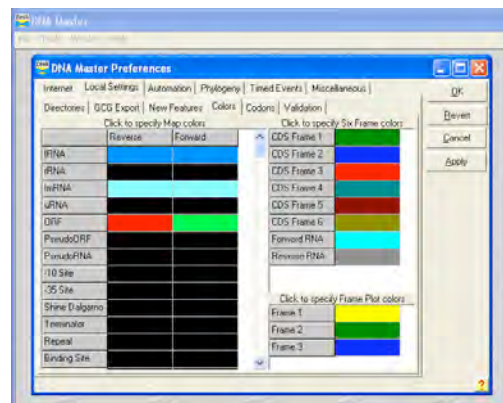


Figure 1: Setting Display Colors

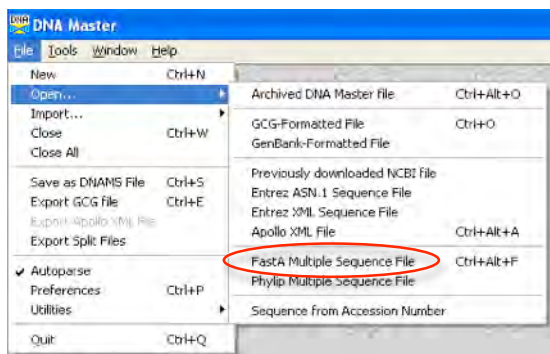


Figure 2: Opening a FastA File

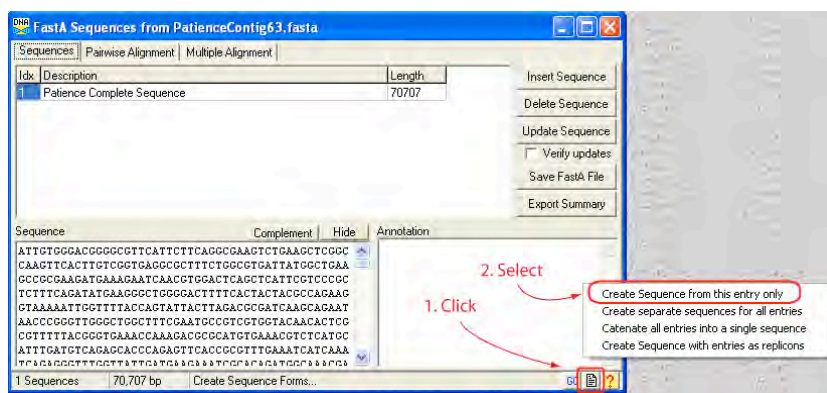


Figure 3: Importing the Sequence into DNA Master

Auto-Annotating a Genome

1. Make sure the “Extracted from YourPhage.fasta” window is open and selected, then go to the **Genome** menu at the top of the program, and select **Annotation** → **Auto-Annotate**.
NOTE: Many options are available in the Auto-Annotate window that opens. Our standard choices for each are shown in **Figure 4**.
NOTE: Be prepared to wait **30-90 minutes** if you choose to “Perform BLAST search on nr database” because BLAST searches take time. If you’d rather move on quickly, then un-check that BLAST box. You will still be able to BLAST all genes—and store the results in DNA Master—later.
2. After selecting your desired options, click **Annotate**.
3. When DNA Master asks if you want to “Erase features and annotate genome?”, click **Yes**.
4. Wait while DNA Master annotates your genome. This should be a fairly quick process (<5 minutes without BLASTing), and you can check the status in the lower-left corner of the Auto-Annotate window.
5. When Auto-Annotation is complete, the Auto-Annotate window will close, and you’ll be returned to the main window for your genome. Congratulations, you’ve created a draft genome annotation!

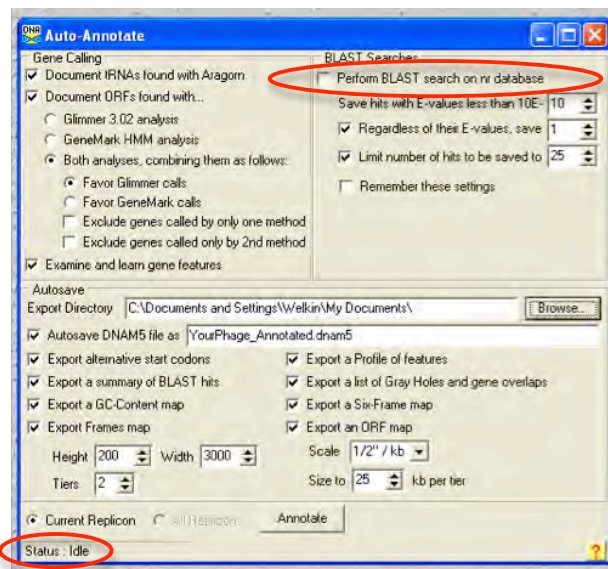


Figure 4: Auto-Annotation Options; BLAST Search Unchecked

Part 2: Refining Your Positional Annotation

Features (Genes)

1. Click on the **Features** tab at the top of your genome's window (the red box in [Figure 5](#)). Here you can see a list of all features (genes) in your current annotation with their left and right coordinates. If you've set up your color preferences properly, forward genes should be in green and reverse genes in red. At the bottom of this window a map of your features is displayed, also color-coded, along with the number of features and total sequence length.
2. The currently selected feature will be identified by a black triangle (the blue box in [Figure 5](#)). Data for this selected feature will be shown in the right pane. The right pane has several tabs of its own, the default being **Description** (see the green box in [Figure 5](#)). By changing tabs, you can see this gene's DNA sequence, amino acid product, any saved BLAST information, and many other statistics. Please explore!

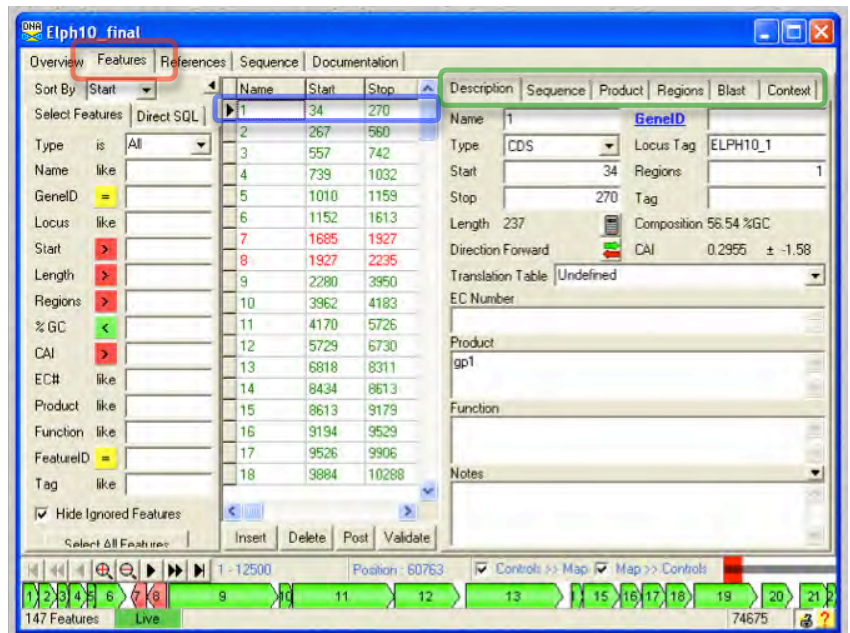


Figure 5: Looking at Auto-Annotated Features

Frames View

3. From the main DNA Master menu, select **DNA → Frames**. A new window titled "ORF Analysis..." will open where the top three rows represent the forwards reading frames, the bottom three rows the reverse reading frames. To see your current genes displayed in this view, click the ORFs button in the lower right corner of this Frames window (enlarged portion of [Figure 6](#)). You should now see green and/or red regions highlighted that represent currently called genes. (You may want to zoom in for better resolution by using the buttons near the bottom left of this window.) All potential start codons are shown as vertical lines that are half the height of a given frame, while stop codons are shown as vertical lines that are the full height of a given frame.

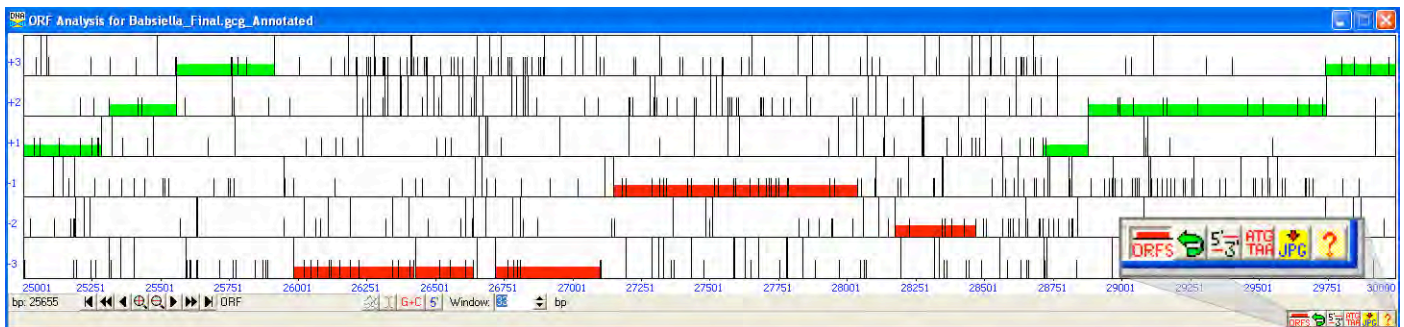


Figure 6: The Frames Window

Comparing Potential Start Codons

4. From the Frames window, select an ORF by clicking within it (DNA Master will draw a line showing your selection).
5. Click the button labeled **5'-3'** in the bottom right corner (enlarged portion of [Figure 6](#)), and a new window called "Choose ORF Start" will open. In this window will be a list of **each possible start codon** for the selected ORF, along with each Shine-Dalgarno score, upstream sequence, start position, and resulting ORF length.

Part 3: Functional Annotation

BLASTing Predicted Proteins

6. A powerful feature of DNA Master is the ability to BLAST all gene products from an annotation, then store the results in the archived file so that they can later be accessed as needed, even without an internet connection. If a feature has stored BLAST results, you can view them by going to the **Blast** tab for that feature (green box, [Figure 5](#)).
7. If you did not BLAST during auto-annotation, you can do so at any time by going to the **Blast** tab for any feature (green box, [Figure 5](#)), selecting **Blast ALL genes**, modifying settings if desired, then clicking **Blast All**. BLASTing a complete phage genome annotation takes 30+ minutes. Be patient.
8. Once finished, "Genome BLAST Complete" should display, and BLAST data for each feature are in the **Blast** tab.

Part 4: Important Features

Suggested Annotation Layout

The figure below shows how you might arrange windows in DNA Master to work on an annotation. The **Frames Window** (on top of the figure) has all 6 reading frames clearly delineated, and allows you to see currently called genes, other choices for start codons, as well as all other potential ORFs in the genome. The **Main Window** (bottom left in the figure) can be used to view precise coordinates, DNA or amino acid sequence, BLAST results (shown in figure), and more. It is also where you will change start positions, add notes, or add/delete genes. The **Choose ORF Start Window** (bottom right in the figure) allows you to see all potential start codons for a given reading frame along with the associated Shine-Dalgarno scores, start positions, and resulting ORF lengths.

Using all of DNA Master's capabilities in concert facilitates fast and accurate genome annotation.

The screenshot displays the DNA Master software interface. The top window, titled "ORF Analysis for Extracted from FastA Library Patience.fasta", shows a genomic map with six reading frames (+3, +2, +1, -1, -2, -3) and various ORFs highlighted in green. The main window, titled "Extracted from FastA Library Patience.fasta", shows a list of features with columns for Name, Start, Stop, Description, Sequence, Product, Regions, Blast, and Context. A BLAST Hit is shown for "gp19 [Mycobacterium phage Barnyard]" with a score of 132. The "Choose ORF start" window is open, showing a table of potential start codons with columns for Shine D, algarno, Sequence of the Region, Start, Start, ORF, and Length.

#	Shine D	algarno	Sequence of the Region	Start	Start	ORF
1	441	8	GGACAATTTAGGAGTTGGATAA	GTG	12413	822
2	364	6	CCCCGGCAACCAAGGACATCC	GTG	12449	786
3	399	7	TTCTTTCCCGTGAGCAAGCGA	ATG	12485	750
4	378	7	GTGGACCAAGCGAATGGACCCG	ATG	12494	741
5	345	9	GCGAATCGACCCGATGTTTCT	GTG	12503	732
6	345	9	CGAAAACCAAAAGCGCGTCCG	ATG	12791	444
7	273	7	CGCTGCTCTGAAAAATCAAGCT	GTG	12878	357
8	378	7	CGCCCCGAGATTTTCGAAAC	ATG	12977	258
9	399	7	TTTCCAAAGCATGGAAGTCGAT	TTG	12989	246
10	273	8	CGGAACGGTTACTGTTTCGGGT	GTG	13019	216
11	294	7	GGCTCTCGCTGAGAGTTCGCC	GTG	13208	27

Creating a Spreadsheet of Current Gene Calls

1. Make sure your main window is open and selected. From the top menu, select **Genome** → **Profile**.
2. Click **OK**, then save in the format you want.
3. Open the file you generated with Excel or a similar spreadsheet program.

Creating an ORF Map of Current Gene Calls

1. Make sure your main window is open and selected. From the top menu, select **DNA** → **Export Map**.
2. Modify any options you'd like to, then click **Draw**.
3. A Windows Meta File (.wmf) image is generated, and can be opened with most image-viewing programs.

References for gene functions

Appendix III

Appendix III: References for gene functions

TM4 structural genes

TUBERCLE AND LUNG DISEASE : THE OFFICIAL JOURNAL OF THE INTERNATIONAL UNION AGAINST TUBERCULOSIS AND. Vol 79, Issue 2, Pages 63-73, 1998.

Mycobacteriophage TM4: genome structure and gene expression.

M E Ford, C Stenstrom, R W Hendrix, G F Hatfull

PubMed ID: [10645443](#)

L5 integrase

JOURNAL OF BACTERIOLOGY. Vol 175, Issue 21, Pages 6836-41, 1993.

Mycobacteriophage L5 integrase-mediated site-specific integration in vitro.

M H Lee, G F Hatfull

PubMed ID: [8226625](#)

L5 Xis (gp 36)

MOLECULAR MICROBIOLOGY. Vol 35, Issue 2, Pages 350-60, 2000.

Identification and characterization of mycobacteriophage L5 excisionase.

J A Lewis, G F Hatfull

PubMed ID: [10652095](#)

Bxb1 serine integrase (gp 35)

MOLECULAR MICROBIOLOGY. Vol 50, Issue 2, Pages 463-73, 2003.

Mycobacteriophage Bxb1 integrates into the Mycobacterium smegmatis groEL1 gene.

Amy I Kim, Pallavi Ghosh, Michelle A Aaron, Lori A Bibb, Shruti Jain, Graham F Hatfull

PubMed ID: [14617171](#)

Bxb1 RDF (gp 46)

PLOS BIOLOGY. Vol 4, Issue 6, Pages e186, 2006.

Control of phage Bxb1 excision by a novel recombination directionality factor.

Pallavi Ghosh, Laura R Wasil, Graham F Hatfull

PubMed ID: [16719562](#)

D29 integrase

GENE. Vol 225, Issue 1-2, Pages 143-51, 1998.

Mycobacteriophage D29 integrase-mediated recombination: specificity of mycobacteriophage integration.

C E Peña, J Stoner, G F Hatfull

PubMed ID: [9931474](#)

Bethlehem DnaB intein

HE JOURNAL OF BIOLOGICAL CHEMISTRY. Vol 285, Issue 4, Pages 2515-26, 2010.

Splicing of the mycobacteriophage Bethlehem DnaB intein: identification of a new mechanistic class of inteins that contain an obligate block F nucleophile.

Kazuo Tori, Bareket Dassa, Margaret A Johnson, Maurice W Southworth, Lear E Brace, Yoshizumi Ishino, Shmuel Pietrokovski, Francine B Perler

PubMed ID: [19940146](#)

Che9c RecE and RecT (gp60 and 61)

MOLECULAR MICROBIOLOGY. Vol 67, Issue 5, Pages 1094-107, 2008.

Efficient point mutagenesis in mycobacteria using single-stranded DNA recombineering: characterization of antimycobacterial drug targets.

Julia C van Kessel, Graham F Hatfull

PubMed ID: [18221264](#)

L5 structural proteins

MOLECULAR MICROBIOLOGY. Vol 7, Issue 3, Pages 395-405, 1993.

DNA sequence, structure and gene expression of mycobacteriophage L5: a phage system for mycobacterial genetics.

G F Hatfull, G J Sarkis

PubMed ID: [8459766](#)

L5 Repressor (gp 71)

MOLECULAR MICROBIOLOGY. Vol 7, Issue 3, Pages 407-17, 1993.

Superinfection immunity of mycobacteriophage L5: applications for genetic transformation of mycobacteria.

M K Donnelly-Wu, W R Jacobs, G F Hatfull

PubMed ID: [8459767](#)

Bxb1 Repressor (gp 69)

MOLECULAR MICROBIOLOGY. Vol 38, Issue 5, Pages 971-85, 2000.

Transcriptional regulation and immunity in mycobacteriophage Bxb1.

S Jain, G F Hatfull

PubMed ID: [11123672](#)

L5 genes that are cytotoxic, (Set of genes that we know that the clones can not be transformed into a non-lysogen)

MICROBIOLOGY (READING, ENGLAND). Vol 154, Issue Pt 8, Pages 2304-14, 2008.

Identification of three cytotoxic early proteins of mycobacteriophage L5 leading to growth inhibition in *Mycobacterium smegmatis*.

Jan Rybniker, Georg Plum, Nirmal Robinson, Pamela L Small, Pia Hartmann

Cluster G repressors

MICROBIOLOGY (READING, ENGLAND). Vol 155, Issue Pt 9, Pages 2962-77, 2009.

Mycobacteriophages BPs, Angel and Halo: comparative genomics reveals a novel class of ultra-small mobile genetic elements.

Timothy Sampson, Gregory W Broussard, Laura J Marinelli, Deborah Jacobs-Sera, Mondira Ray, Ching-Chung Ko, Daniel Russell, Roger W Hendrix, Graham F Hatfull

PubMed ID: [19556295](#)

Lysins A and B

MOLECULAR MICROBIOLOGY. Vol 73, Issue 3, Pages 367-81, 2009.

Mycobacteriophage Lysin B is a novel mycolylarabinogalactan esterase.

Kimberly Payne, Qingan Sun, James Sacchettini, Graham F Hatfull

PubMed ID: [19555454](#)

WhiB of tm4

MOLECULAR MICROBIOLOGY. Vol 77, Issue 3, Pages 642-57, 2010.

Insights into the function of the WhiB-like protein of mycobacteriophage TM4--a transcriptional inhibitor of WhiB2.

Jan Rybniker, Angela Nowag, Edeltraud van Gumpel, Nicole Nissen, Nirmal Robinson, Georg Plum, Pia Hartmann

PubMed ID: [20545868](#)

Non-essential genes

PLOS ONE. Vol 3, Issue 12, Pages e3957, 2008.

BRED: a simple and powerful tool for constructing mutant and recombinant bacteriophage genomes.

Marinelli LJ, Piuri M, Swigonová Z, Balachandran A, Oldfield LM, van Kessel JC, Hatfull GF

PubMed ID: [19088849](#)

Tapemeasure of TM4, defect in infection of stationary phage cells

MOLECULAR MICROBIOLOGY. Vol 62, Issue 6, Pages 1569-85, 2006.

A peptidoglycan hydrolase motif within the mycobacteriophage TM4 tape measure protein promotes efficient infection of stationary phase cells.

Mariana Piuri, Graham F Hatfull

Phage I3 (cluster C) promoters and some structural genes

GENE. Vol 143, Issue 1, Pages 95-100, 1994.

Structural proteins of mycobacteriophage I3: cloning, expression and sequence analysis of a gene encoding a 70-kDa structural protein.

G R Ramesh, K P Gopinathan

PubMed ID: [8200544](#)

INDIAN JOURNAL OF BIOCHEMISTRY & BIOPHYSICS. Vol 33, Issue 1, Pages 83, 1996.

Cloning and characterization of mycobacteriophage I3 promoters.

G R Ramesh, K P Gopinathan

PubMed ID: [8744840](#)

TM4 non-essential genes, D29 non-essential genes

PROCEEDINGS OF THE NATIONAL ACADEMY OF SCIENCES OF THE UNITED STATES OF AMERICA. Vol 94, Issue 20, Pages 10961-6, 1997.

Conditionally replicating mycobacteriophages: a system for transposon delivery to Mycobacterium tuberculosis.

S Bardarov, J Kriakov, C Carriere, S Yu, C Vaamonde, R A McAdam, B R Bloom, G F Hatfull, W R Jacobs

PubMed ID: [9380742](#)

D29 and L5 non-essential genes

JOURNAL OF MOLECULAR BIOLOGY. Vol 279, Issue 1, Pages 143-64, 1998.

Genome structure of mycobacteriophage D29: implications for phage evolution.

M E Ford, G J Sarkis, A E Belanger, R W Hendrix, G F Hatfull

PubMed ID: [9636706](#)

L5 Promoters

MOLECULAR MICROBIOLOGY. Vol 17, Issue 6, Pages 1045-56, 1995.

Transcriptional regulation of repressor synthesis in mycobacteriophage L5.

C E Nesbit, M E Levin, M K Donnelly-Wu, G F Hatfull

PubMed ID: [8594325](#)

Unpublished:

Bps gene 22 :

 Mutations in this gene have expanded host range

Rosebush genes 32 and 42

 Mutations in these genes have expanded host range

Cluster N integrases and repressors, Brujita Integrase and repressor

Cluster J capsids:

Identified through N-terminal sequencing

Giles non-essential genes identified through BRED

Appendix IV

Genome Maps:

L5

Cluster B map

Cluster C map

Cluster D map

Cluster E map

Cluster F map

Cluster G map

Cluster I map

Cluster J map

Cluster K map

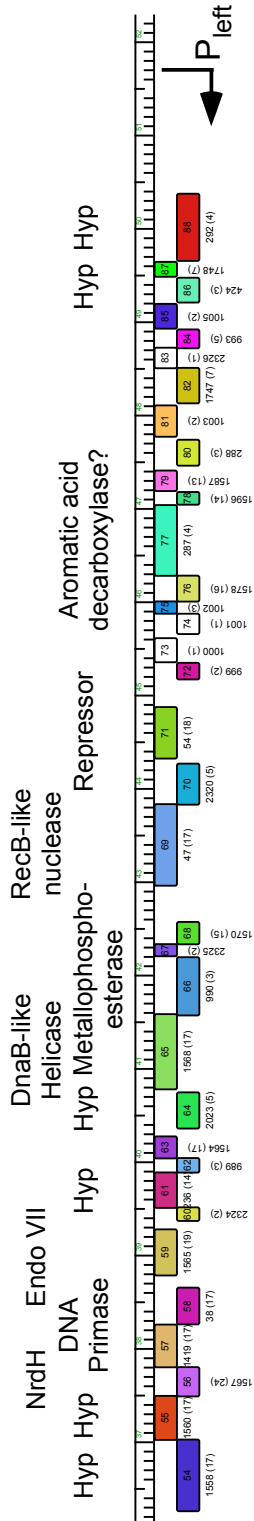
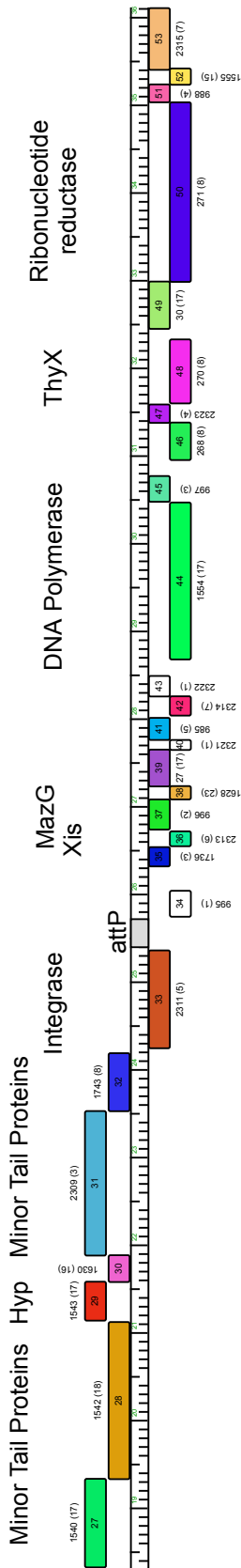
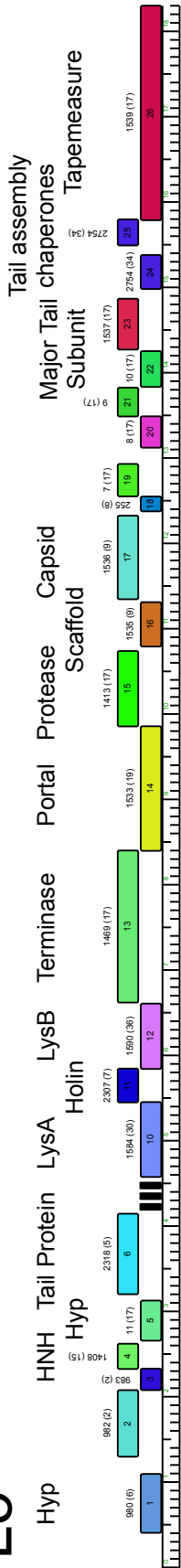
Cluster L map

Cluster O map

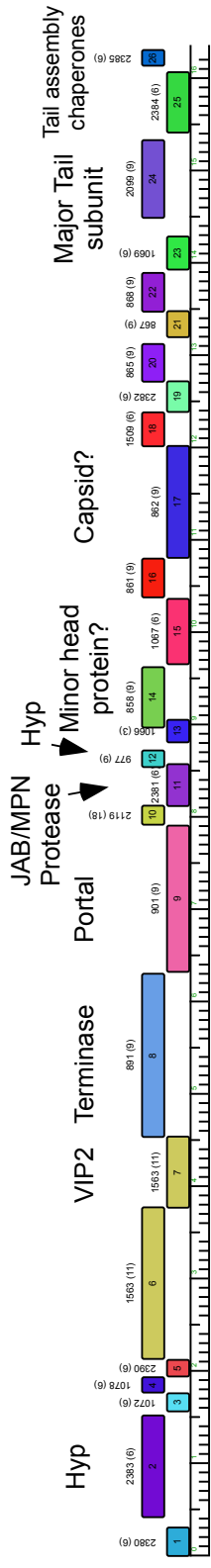
Giles map

Wildcat map

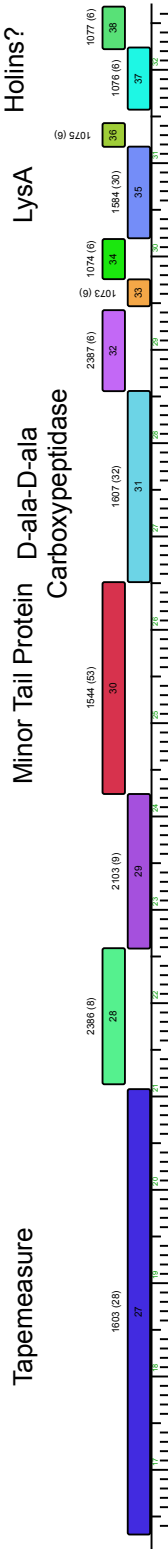
L5



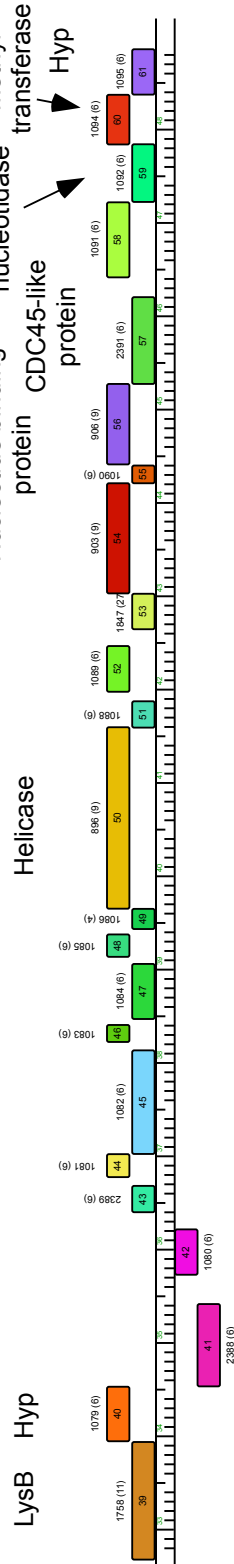
PBI1



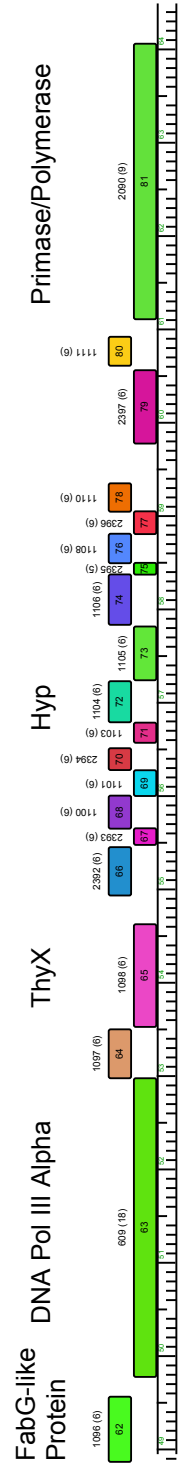
Tapemeasure



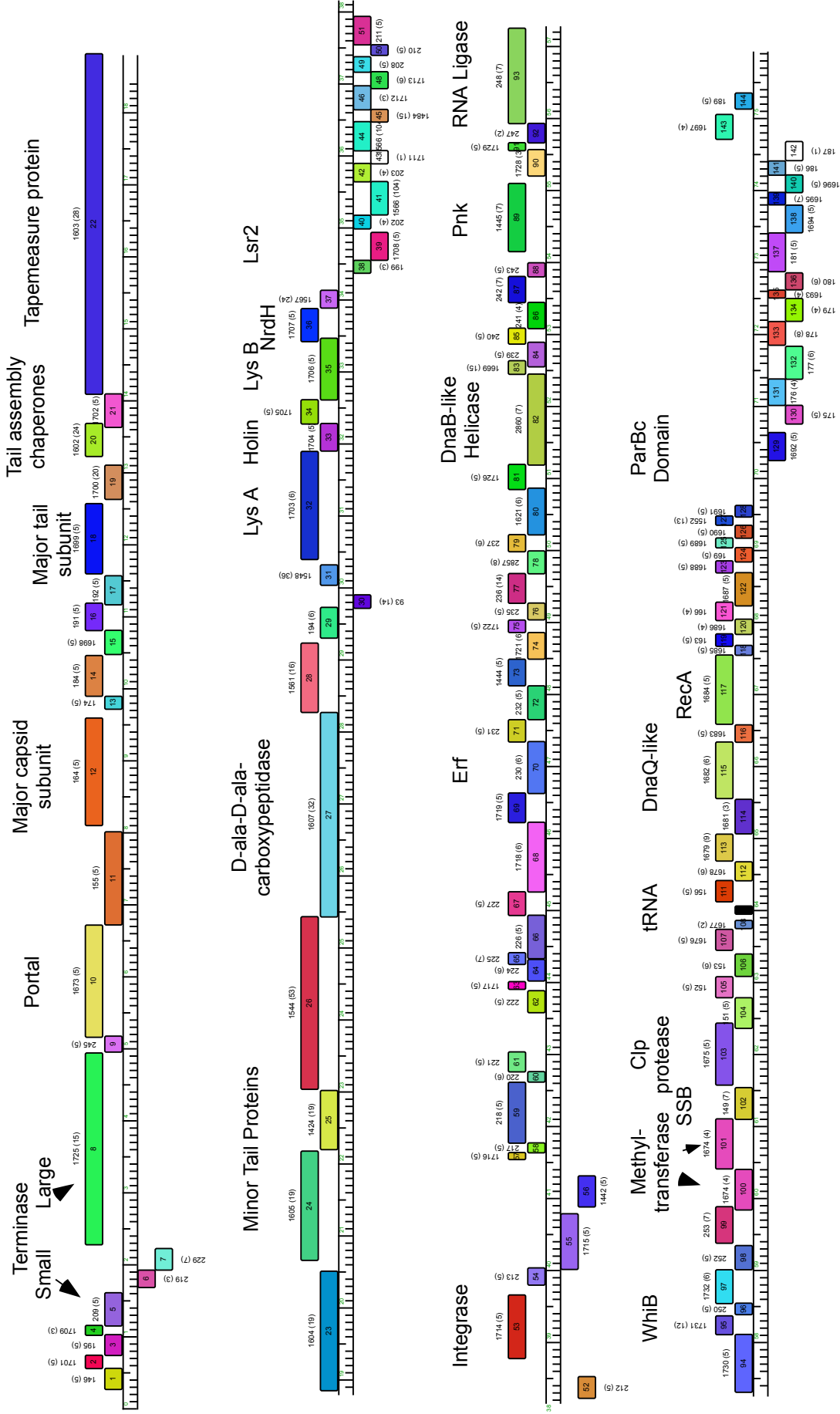
LysB Hyp



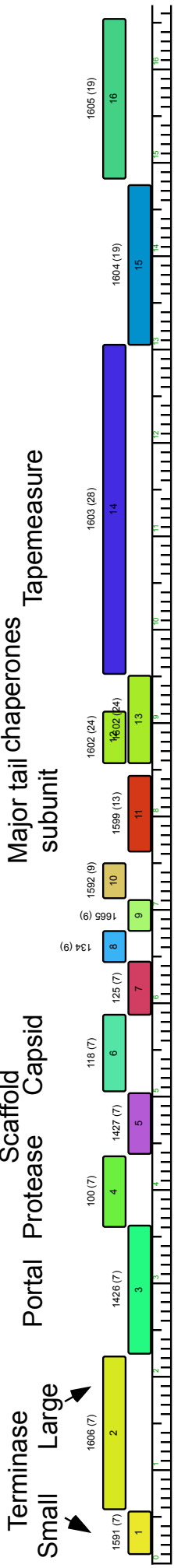
FabG-like Protein



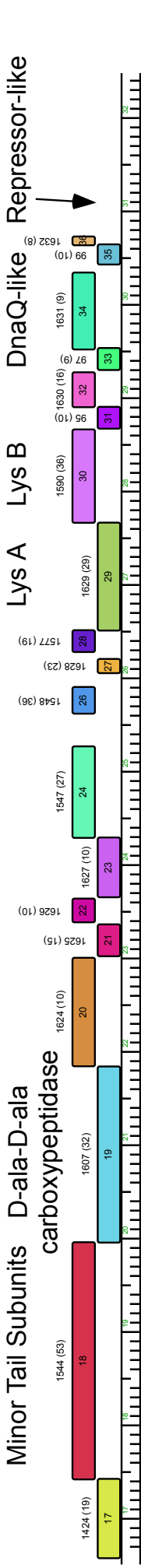
Cjw1



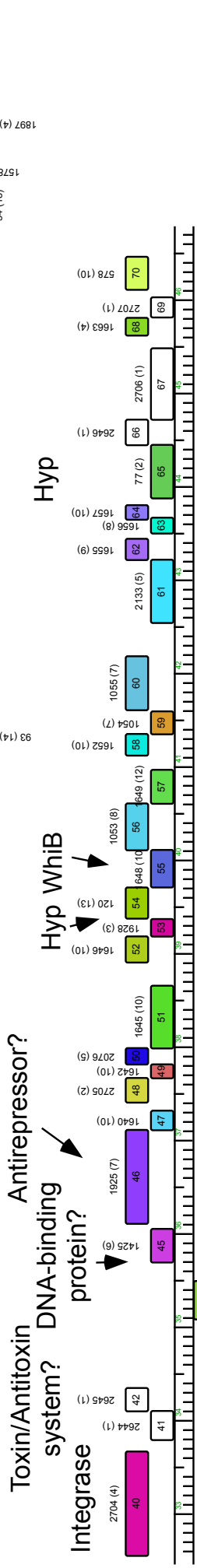
Fruitloop



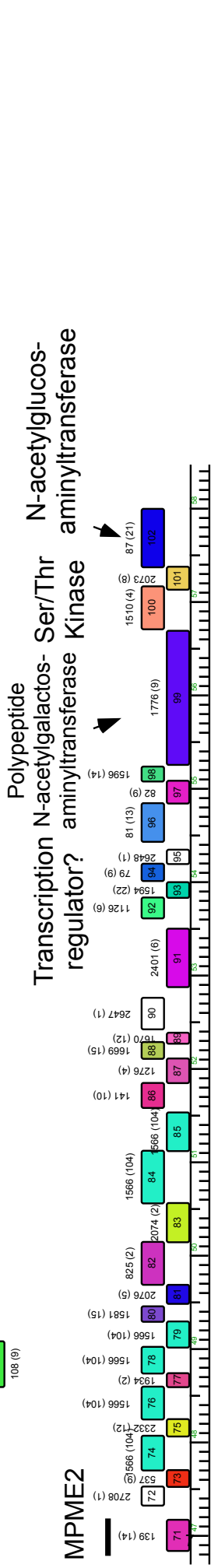
Minor Tail Subunits



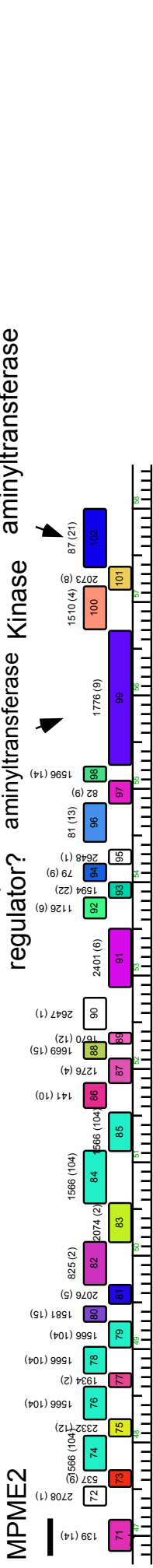
Toxin/Antitoxin system? DNA-binding protein? Antirepressor?



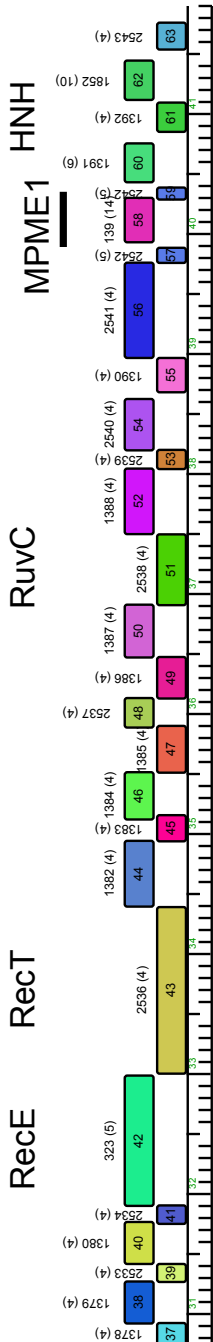
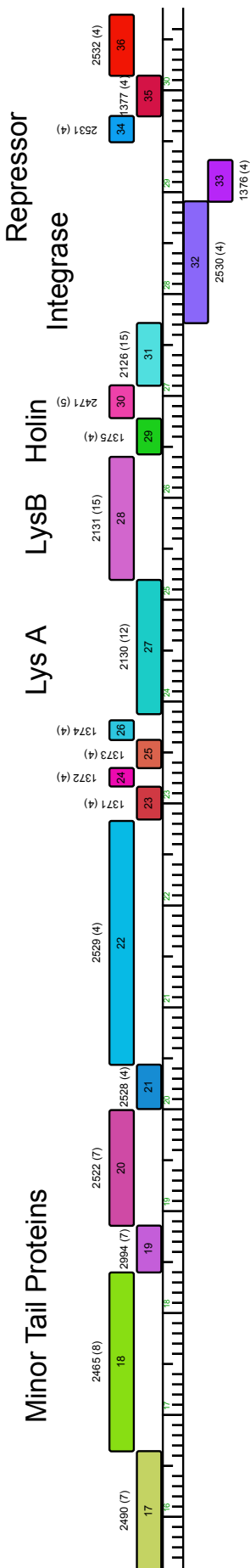
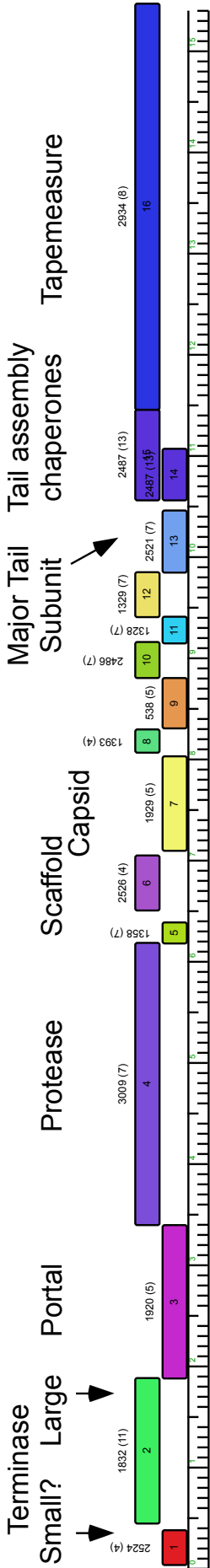
Polypeptide Transcription regulator? N-acetylgalactosyl aminyltransferase



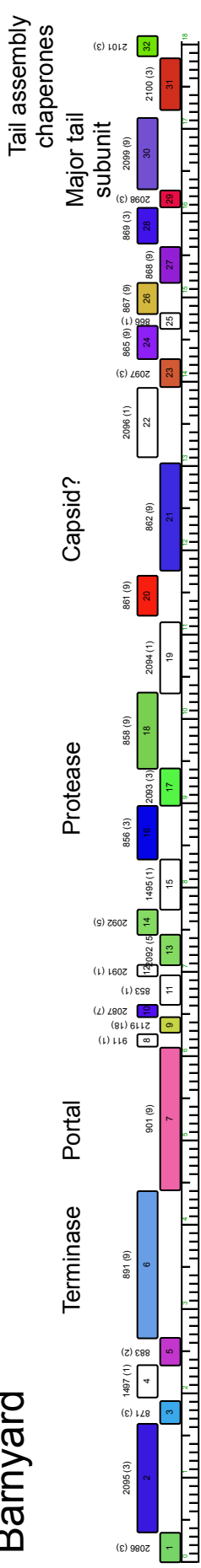
MPME2



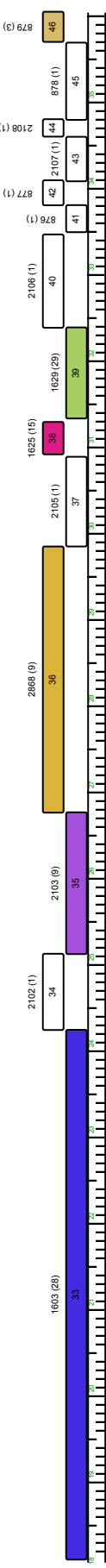
BPs



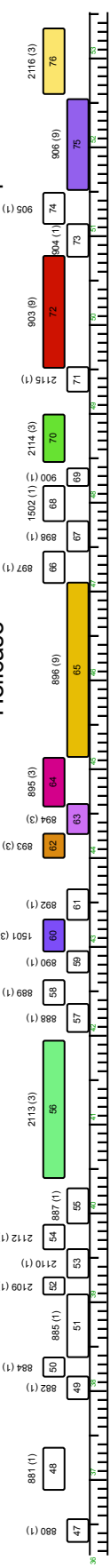
Barnyard



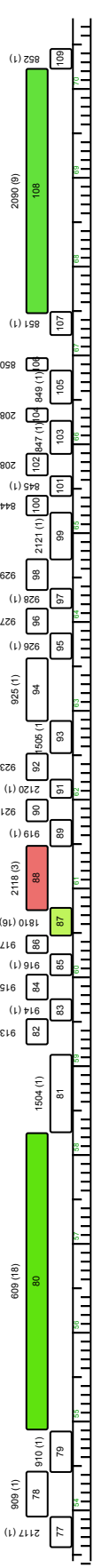
Tapemeasure protein



Nucleotide binding protein

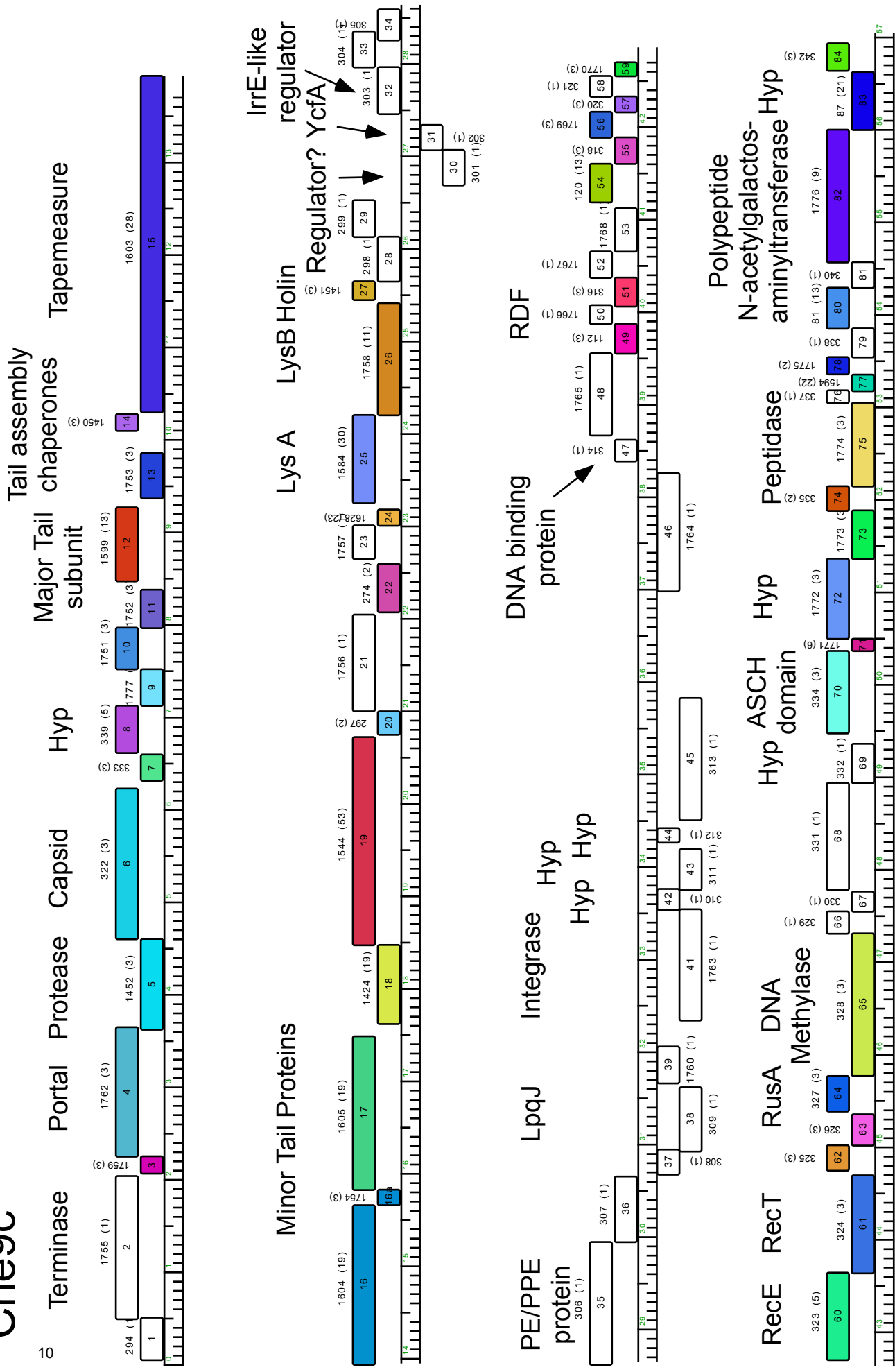


DNA Pol III Alpha

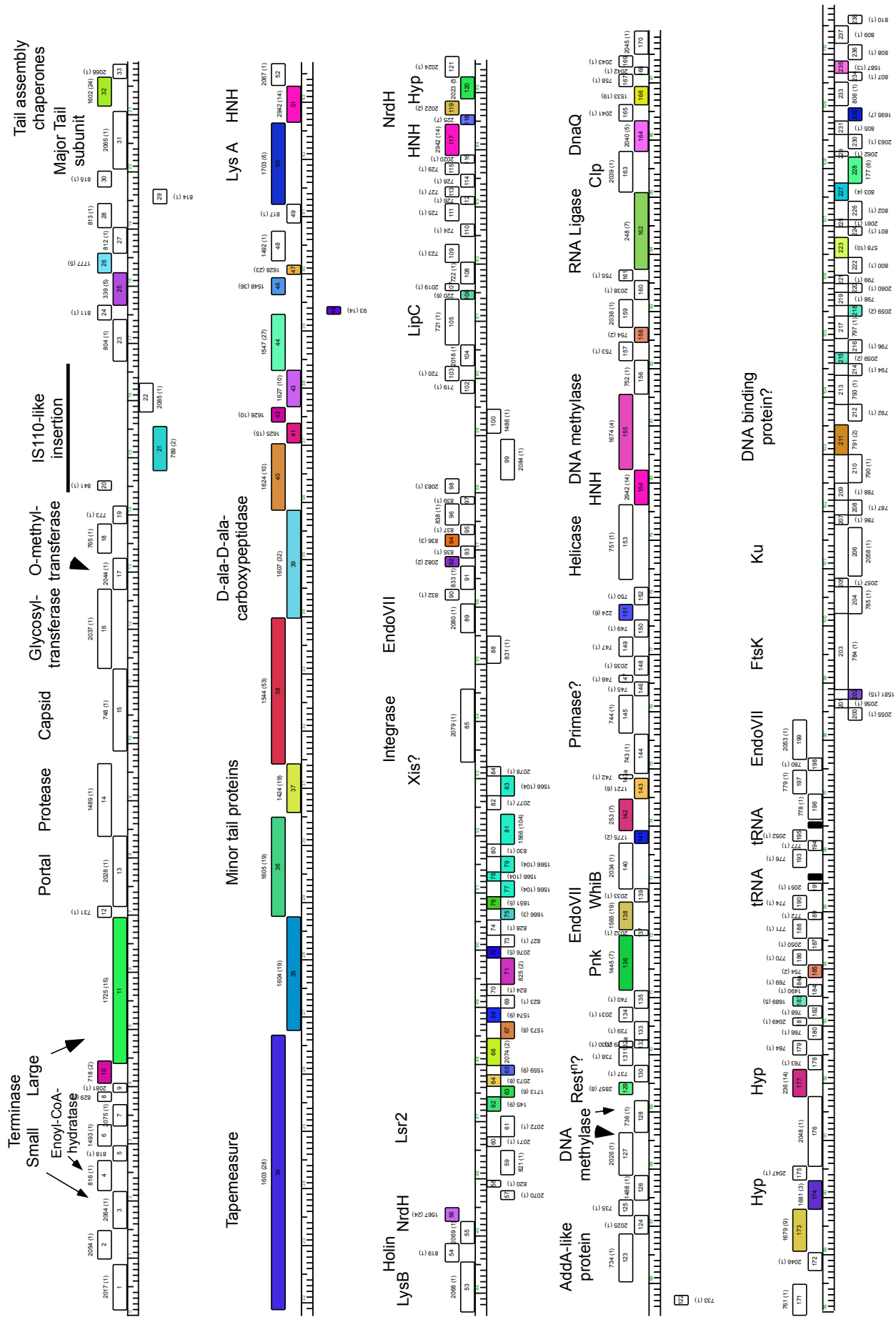


Che9c

10



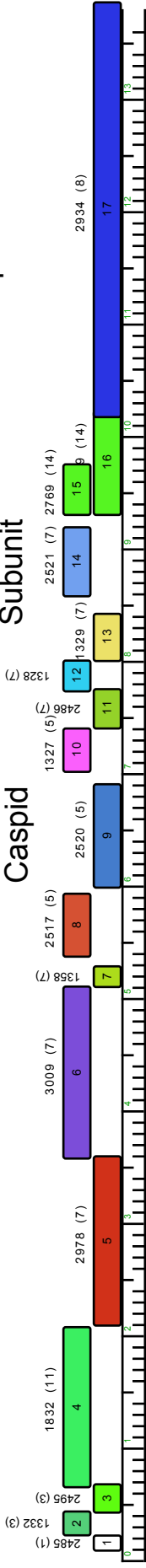
Omega



TM4

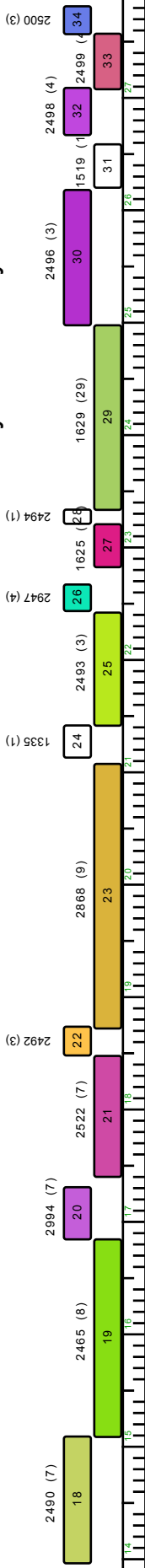
Tail assembly
chaperones
Major Tail
Subunit
Tapemeasure

Terminase Portal Protease Scaffold Caspid



Minor Tail Proteins

LysA LysB HoIn



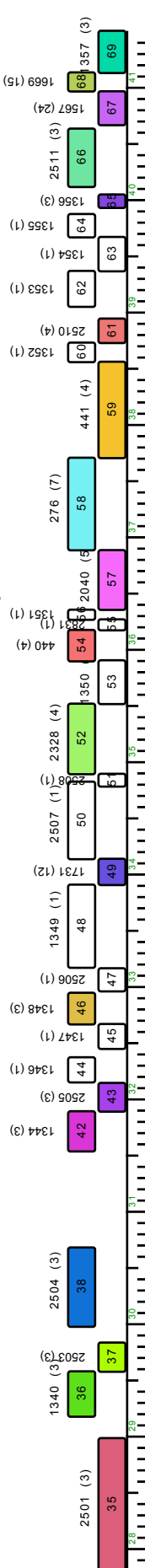
DNA-binding
protein?

WhiB

DnaQ-like
protein

Hyp

NrdH

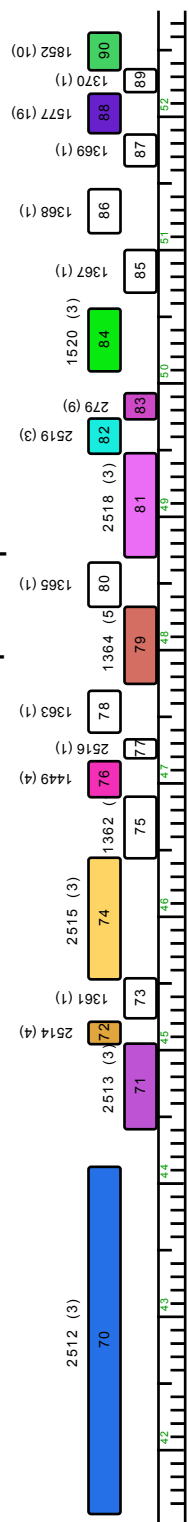


Primase/Helicase Rusa

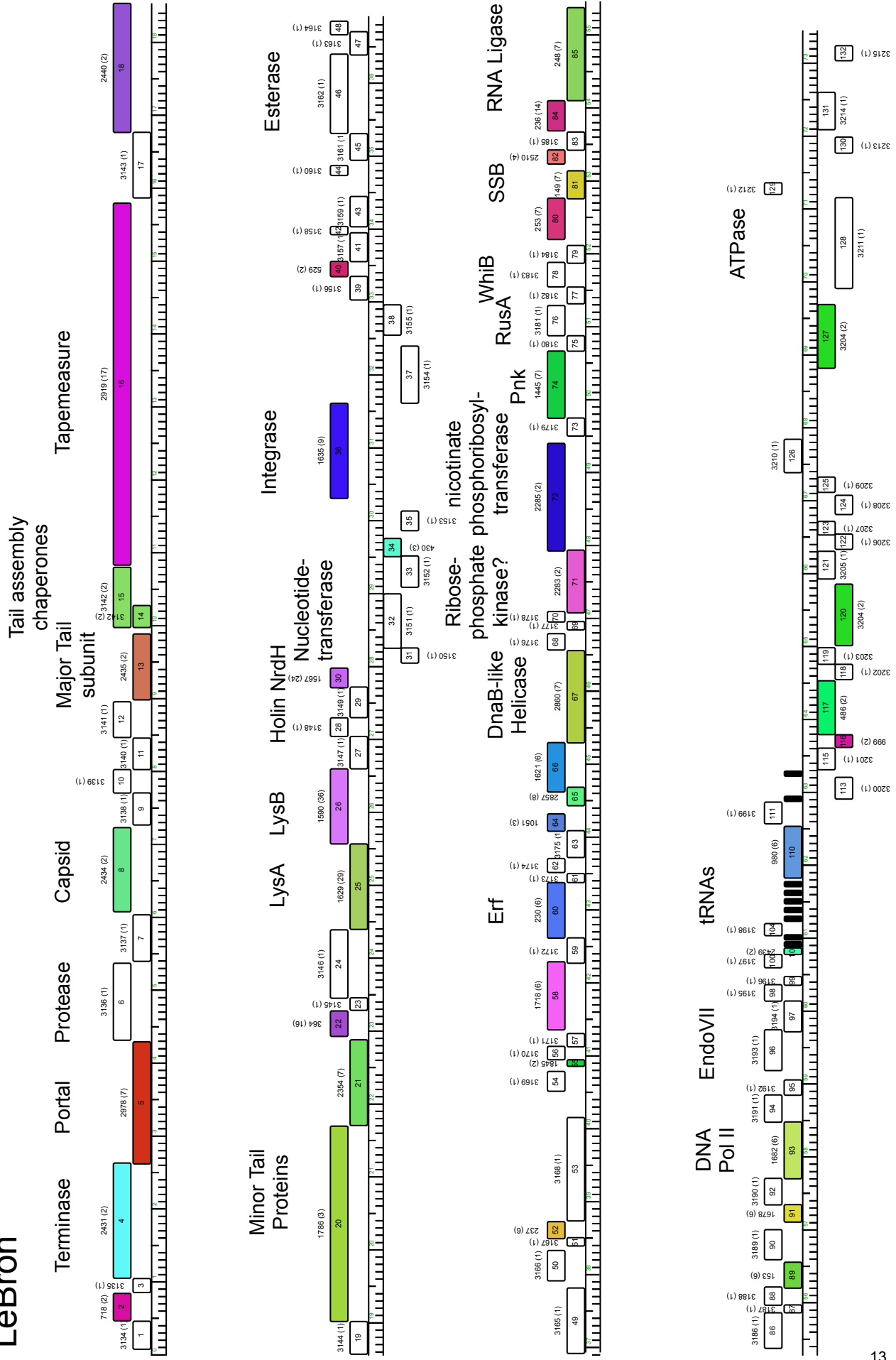
DNA-binding
protein?

SprT

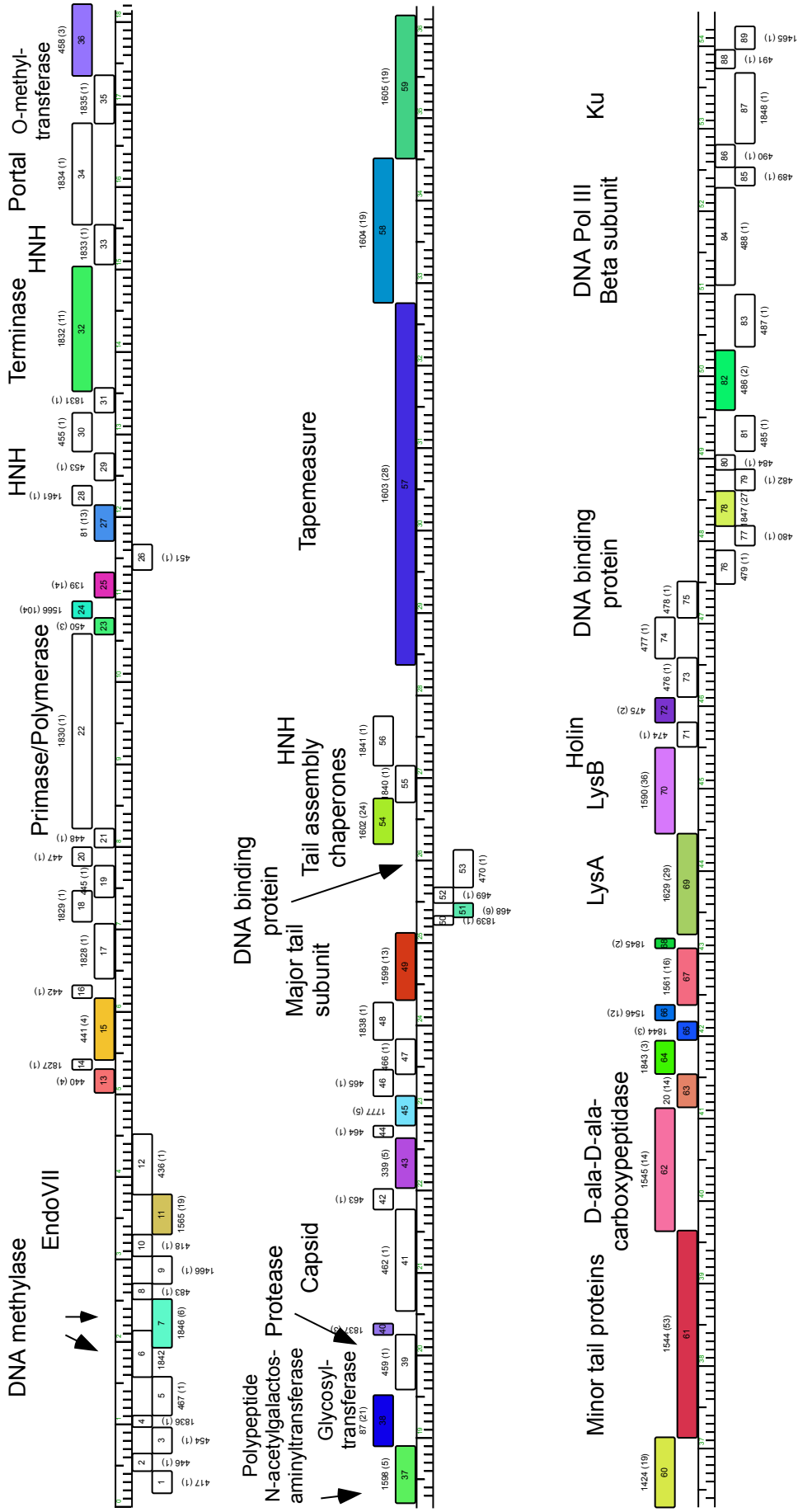
HNH



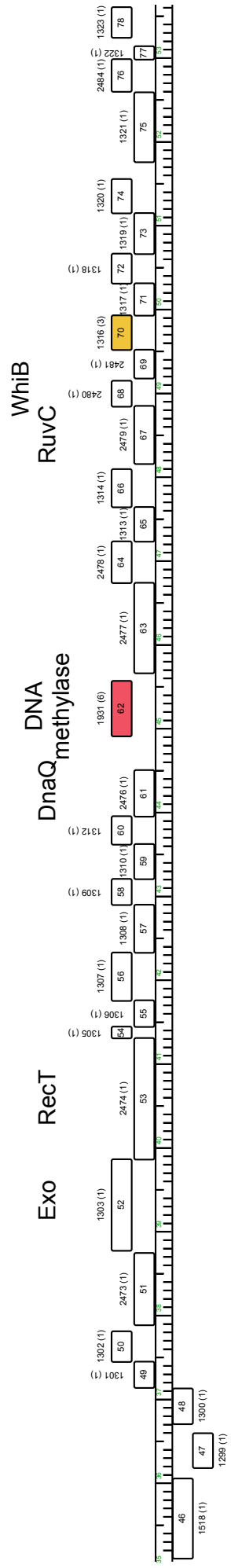
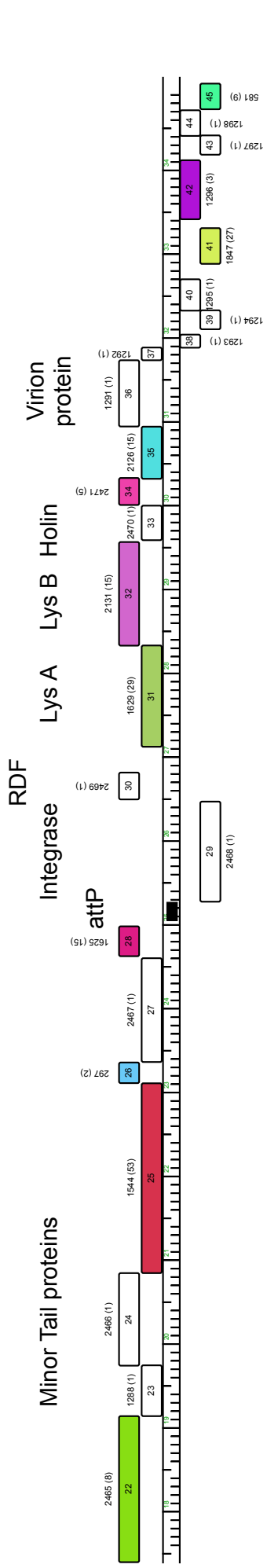
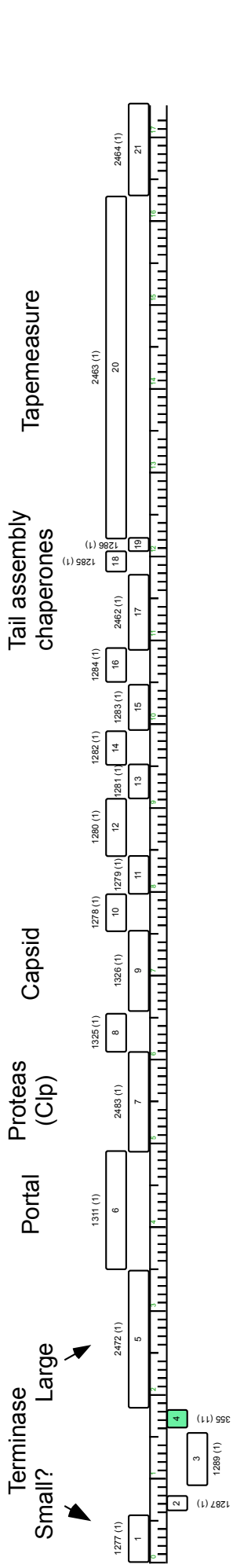
LeBron



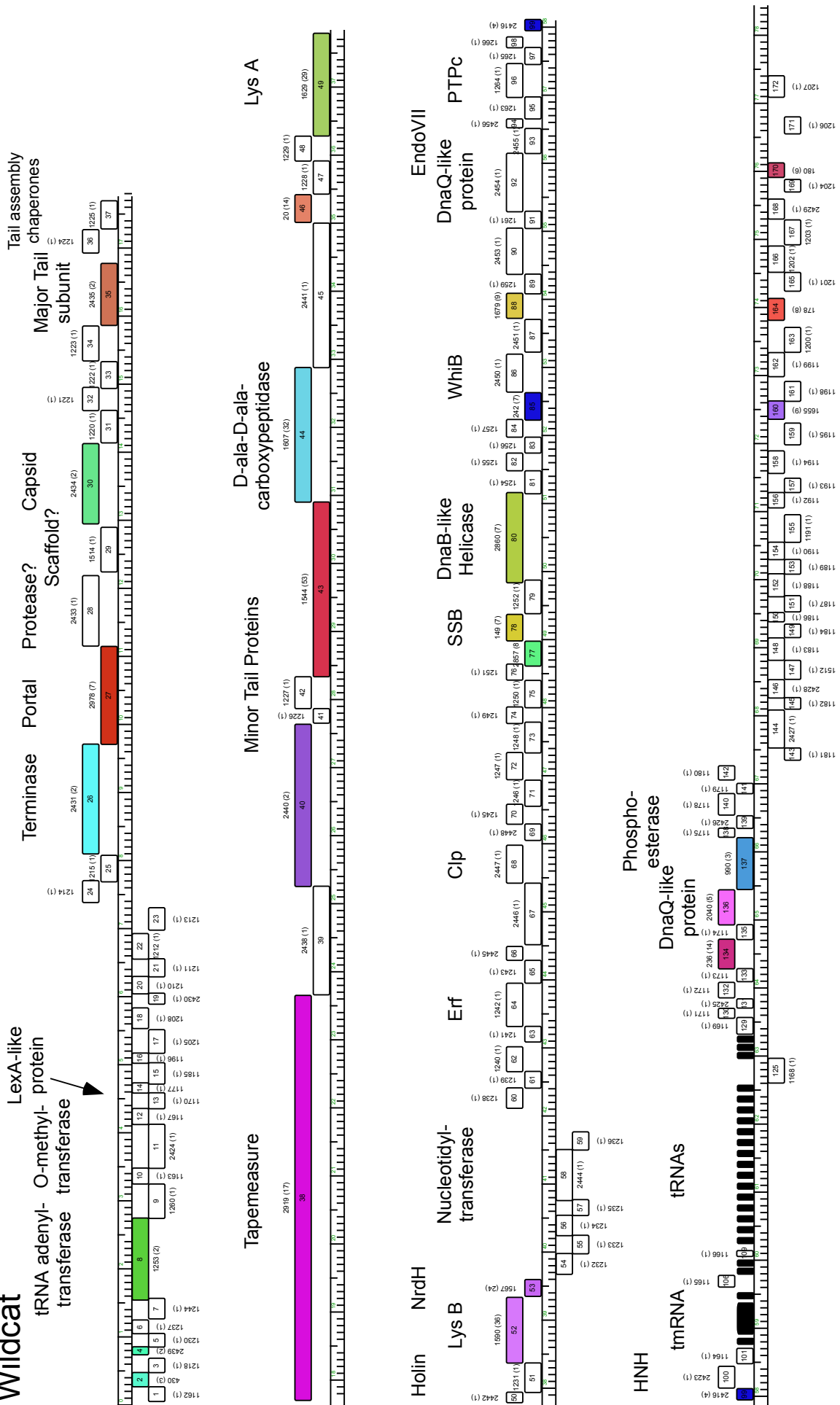
Corndog



Giles



Wildcat



Appendix V: Etude Annotation

First, BLAST Etude against phagesdb.org.

Go to the phagesdb.org BLAST page. Paste in the Etude sequence or browse to the FASTA file on your computer. Turn off the low complexity filter. Press BLAST.

Mycobacteriophage Database | BLAST

http://phagesdb.org/blast/

Home Phages Data Entry BLAST Publications Education Link

Local Phage BLAST

This tool will run a local BLAST search against our phage database. It will include some genomes that are not yet in GenBank and thus accessible via NCBI BLAST.

Choose program to use and database to search:

Program Database

Enter sequence below in **FASTA** format

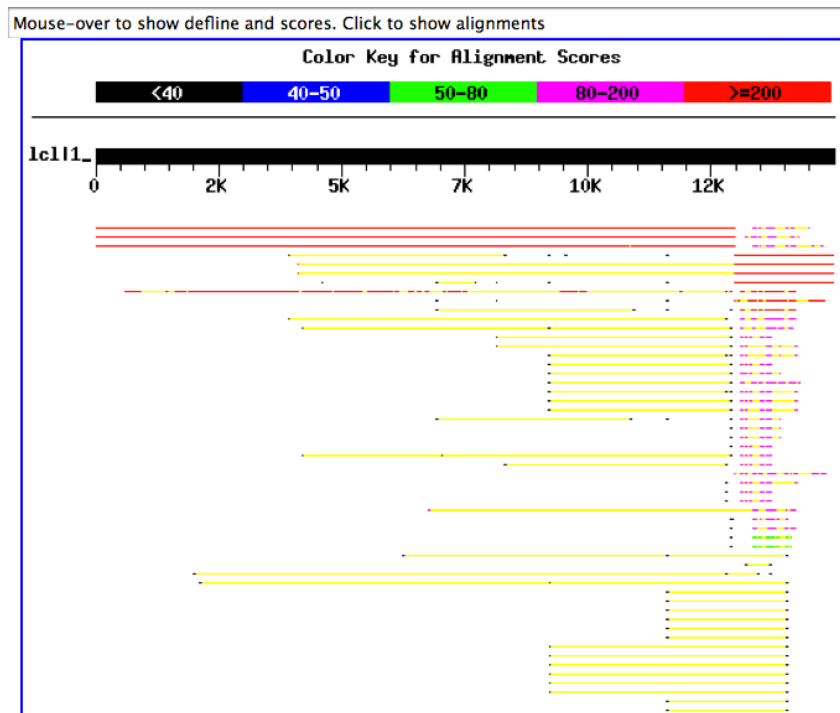
```
>etude
ACCGACACTTCTCTCTCGGAAATTCAGGCAAGAACATGAGGGGGTTAGCGCCCTAAA
ACCCCTGGTAGGAGGCTAAATCGTGGGTAGAGGACGTGGTAAGGACCCGTC AAGCCCTGG
TGGGGGTCTCGGGACATCCCGGGCACGGCTCGGCCTGGGAGGCGAAGTTGCCGC
CAAACCGAAGAACCGGCAGGAATACGCGGTGCAGATGGCCGAAAGCCCTCGGTGGGAGGT
TGAGAAGCCGAACGTTTGGACCAATCAGGGGATGCACGCCGCTGGTATCGAGACTTTGAC
GATGCGCAAGGGCGATGCTACGTGTATGCGACGTTACCTGGCCTAATGGCCGCATTGCG
```

Or load it from disk

When Etude is BLASTed against phagesdb.org, it appears that there is similarity to the Cluster L1 phages, and to the Cluster A3 phages.

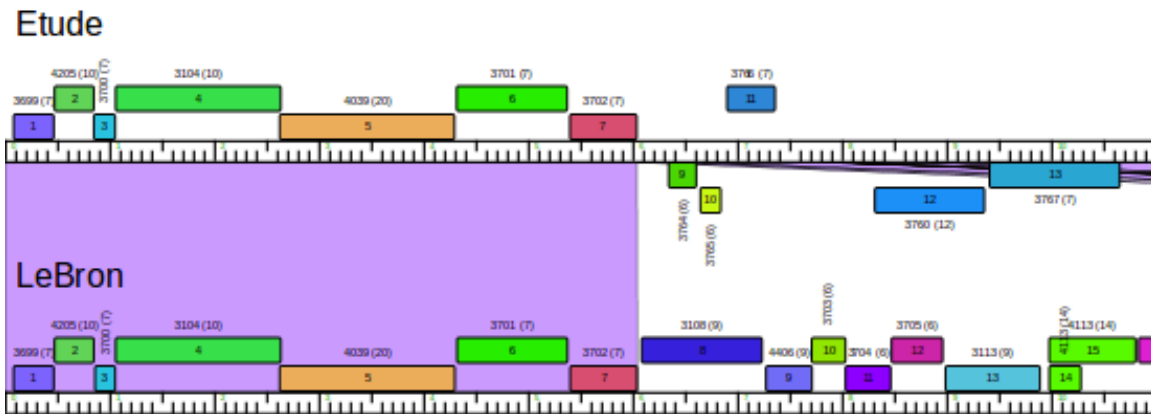
Query= etude
(14,998 letters)

Distribution of 392 Blast Hits on the Query Sequence



Sequences producing significant alignments:	Score (bits)	E Value
UPIE Complete Sequence, 73784 bp including 10 bp 3' overhang (TC...	1.314e+04	0.0
LeBron	1.178e+04	0.0
JoeDirt Final Sequence, 74914 bp including 10 bp 3' overhang (TC...	1.169e+04	0.0
Microwolf Final Sequence, 50864 bp including 10 bp 3' overhang, ...	4022	0.0
Vix Complete Sequence, 50963 bp including 10 bp 3' overhang (CGG...	3998	0.0
JHC117 Final Sequence, 50877 bp including 10 bp 3' overhang, Clu...	3998	0.0
Bxz2	3998	0.0
Faith1 Complete Sequence, 75960 bp including 10 bp 3' overhang (...)	1388	0.0
Rockstar Complete Sequence, 47780 bp including 10 bp 3' overhang...	232	2e-59
Peaches	212	2e-53
Eagle	204	4e-51
LHTSCC Complete Sequence (51813bp, including 10bp 3' overhang: C...	196	1e-48
George Final Sequence, 51578 bp including 10 bp 3' overhang, Clu...	137	8e-31

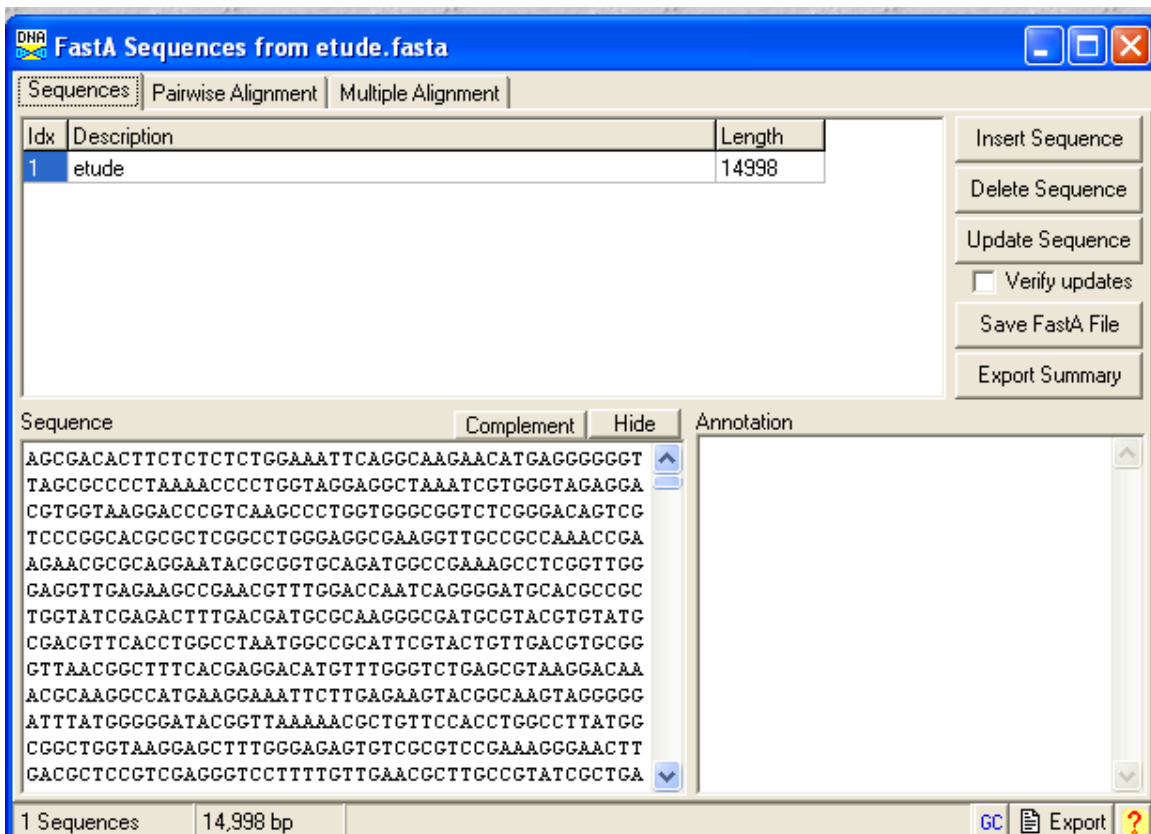
Now we pull up Etude in phamerator, next to its closest matches. I will use LeBron and Bxz3 for now, because these two phages have annotations in GenBank already—which means that when I look for individual genes using BLAST on the NCBI website, I should see these genes, and they will be genes that have already been curated and well-examined by the annotators. I will also check Upie, JoeDirt, and Microwolf's draft annotations in phamerator.



Notice how the purple between the two genomes indicates that the nucleotide sequence similarity is very high between Etude and LeBron for the first seven genes in both genomes. I will start with calling these seven genes.

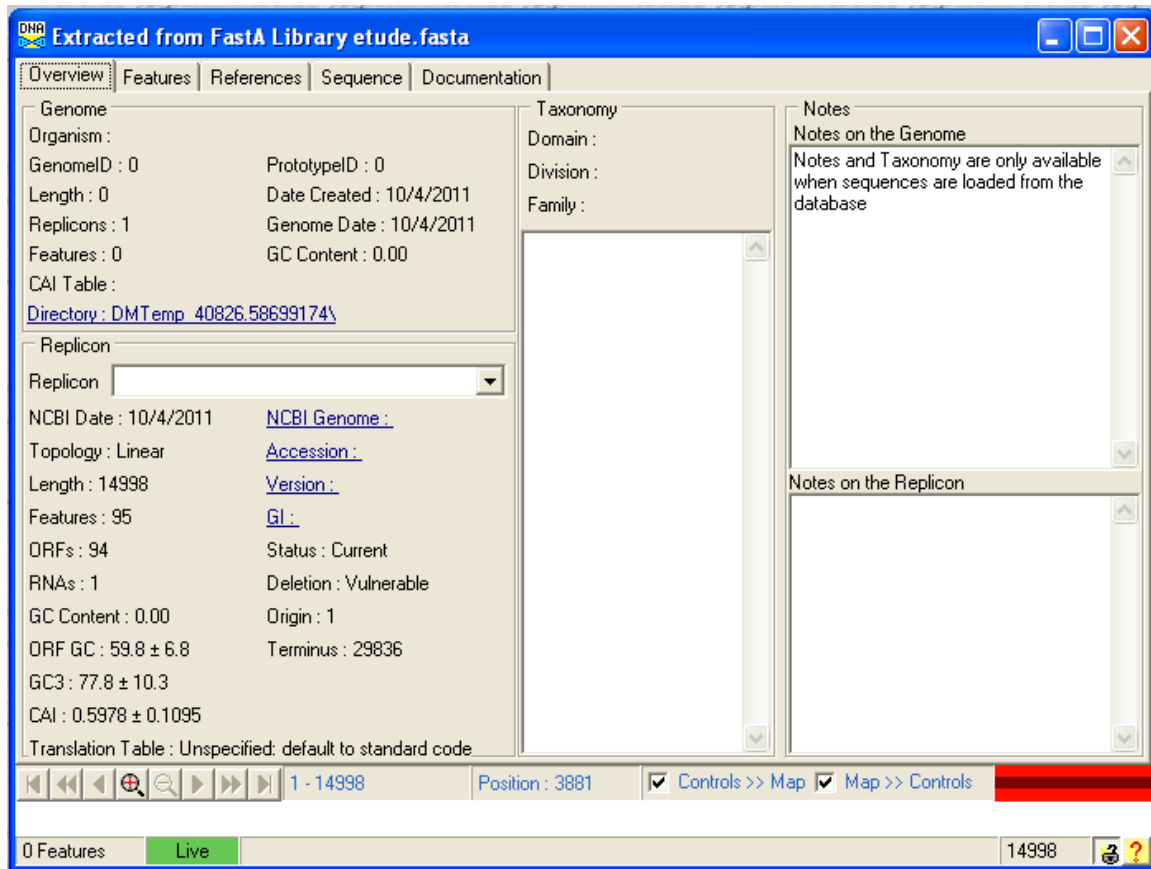
Next, I open DNA master, and load the Etude sequence from its fasta file.

-> File ->Open -> FastA Multiple Sequence File



I then click “export” in the lower right corner, and “Create sequence from this entry only” from the menu that appears.

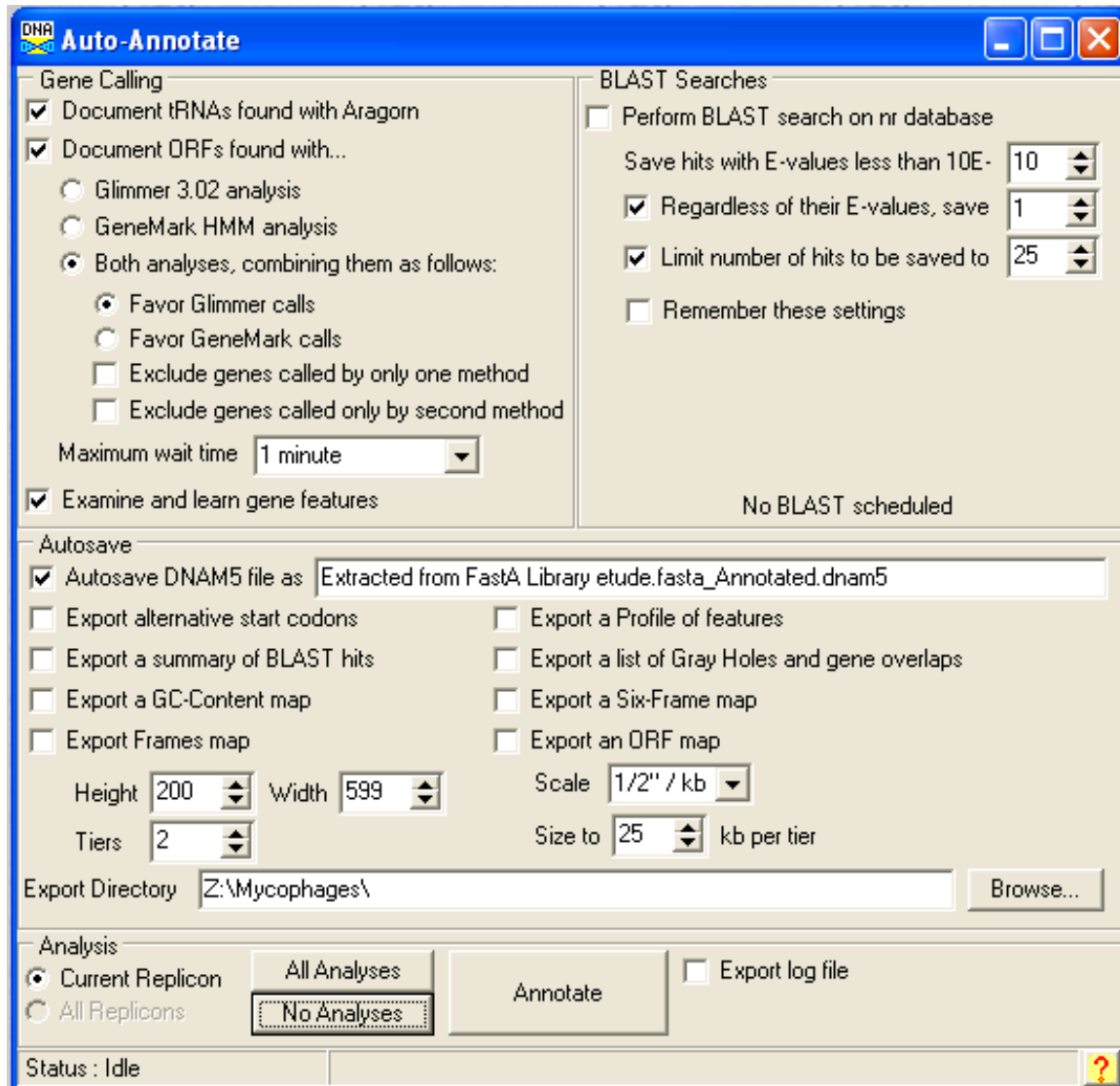
A DNA Master sequence file will be created:



This file is empty other than the imported sequence (viewable if you click the "Sequence" tab above).

Now I auto-annotate this file to generate and import the information from Glimmer, GeneMark, and Aragorn into the file.

Click Genome->Annotation-> Auto-Annotate



Uncheck all the analyses buttons and then click “annotate”. As Etude is a relatively short piece of DNA, I will check the “BLAST” box at the upper right.

Once my genome is annotated and BLASTed, I will save it as Etude_annotated.dnam5.

Now I will generate the data from the programs outside of DNA master that I will need to review the auto-annotation.

GeneMark TB:

GeneMark is located at

http://exon.gatech.edu/genemark/genemark_prok_gms_plus.cgi

There is also a link from the phagesdb website. Upload the etude.fasta file and use the Mycobacterium tuberculosis coding model (either strain is fine).

Sequence File upload:

Running Options

Species: Window size: bp

RBS model: Step size: bp

Use alternate genetic code: Eukaryote (e.g. Yeast, ATG = only start)
 Mycoplasma (TGA = Tryptophan)

Threshold: %

Output Options

Graphical output options

- Generate PDF graphics (screen)
- Generate PostScript graphics (email)
- Mark orfs on graph
- Mark regions on graph
- Mark stop codons on graph
- Mark start codons on graph
- Mark frameshifts on graph
- Mark putative exon splice sites
- Print graph in landscape format

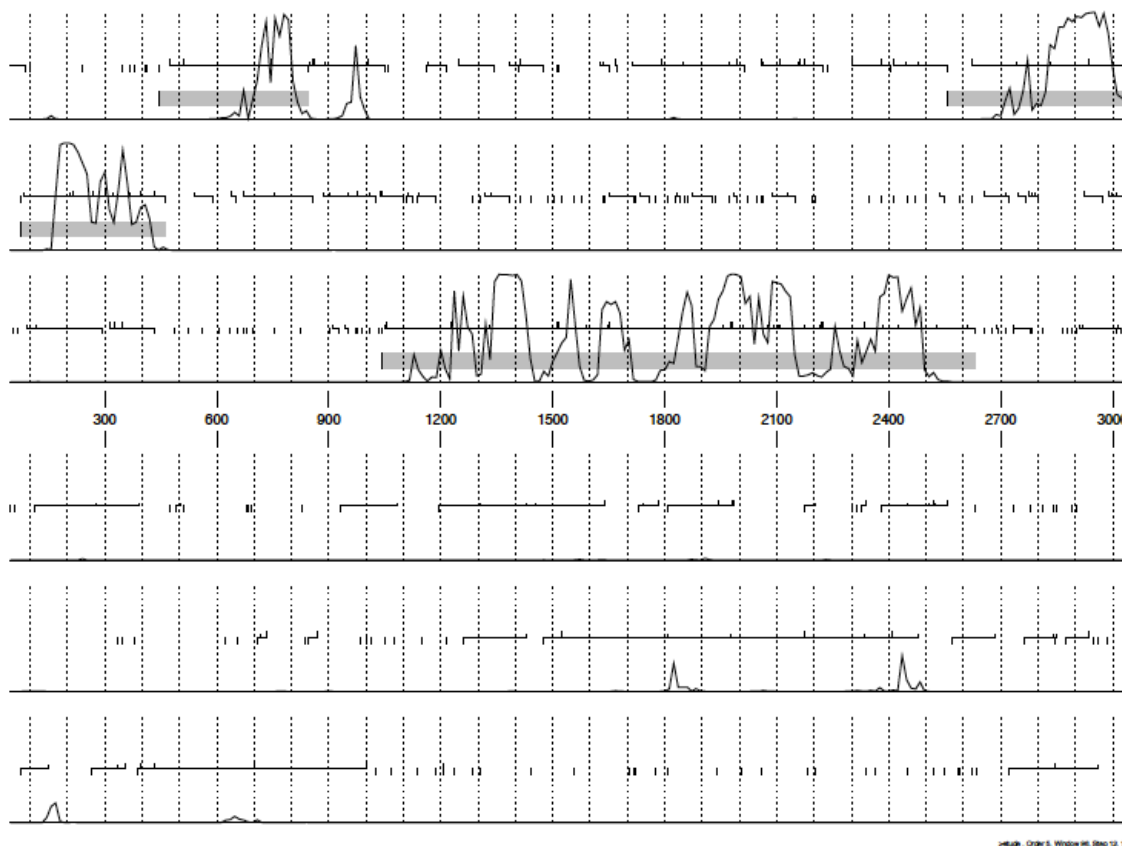
Text output options

- List open reading frames (ORFs) predicted as coding sequences (CDSs)
- List regions of interest
- List putative eukaryotic splice sites
- Write protein translations of ORFs
- Write nucleotide transcripts of ORFs
- Write protein translations of regions
- Write nucleotide transcripts of regions
- Write protein translations of putative exons
- Write nucleotide transcripts of putative exons

Email address (required for PostScript email output)

Run

Opening the .pdf of the GeneMark output should show you a 5 page document that begins like this:



Here are the first four and start of the fifth genes in Etude as called by GeneMark. Each tier represents a different reading frame, with upticks from the center horizontal in each

tier representing start codons in that frame and downticks representing stop codons in that frame. ORFs of significant length are shown as horizontal lines in each tier. Coding potential is shown by the wiggly trace lines. Areas that GeneMark has designated “regions of interest” are shown with gray bars through the coding potential. Sometimes these regions are actually genes, sometimes not. I find it easier to just ignore the gray bars completely.

Aragorn:

Aragorn is found at

<http://130.235.46.10/ARAGORN/>

or linked to from phagesdb.org.

Upload your FastA file, and change the default to “tRNA and tmRNA”.

The screenshot shows the ARAGORN web interface. At the top, there is a navigation bar with 'Home', 'Projects', 'Publications', and 'Online services'. The 'Online services' section is active, displaying a list of tools: ARAGORN, ARWEN, BRUCE, optalign, and RAMI. The ARAGORN tool is selected, showing its description: 'ARAGORN, tRNA (and tmRNA) detection in nucleotide sequences'. Below the description, there is a section for 'Input sequence (both strands will be searched, max. 15 MB)'. This section includes a text input field for a FASTA file path, a 'Browse...' button, and a dropdown menu for selecting a genome from a list (currently showing 'Methanococcus jannaschii (NC_000909)'). There is also a 'Select options (see here for all options in the standalone version)' section with several dropdown menus: 'Search for (default tRNA):' set to 'tRNA & tmRNA', 'Search allowing introns, 0-3000 bases (default no):' set to 'no', 'Sequence topology (default linear):' set to 'linear', 'Strand(s) (default both):' set to 'both', and 'Output format (default standard):' set to 'standard'. At the bottom of this section are 'Submit' and 'Reset' buttons.

Then click “Submit”

etude
14998 nucleotides in sequence
Mean G+C content = 60.2%

1.

```
      c
      a
      g+t
      g-c
      t-a
      c-g
      c-g
      t+g
      g-c      tg
      t      cacc a
taaa a      !!!!! g
t      cgg      gtggg c
g      !!!!! t      tt
g      gcc      c
caaa      t
      g+tg
      c-g
      g-c
      t-a
      c-g
      a-t
      c      a
      t      a
      caa
```

tRNA-Leu(caa)
75 bases, %GC = 56.0
Sequence [6240,6314]

Primary sequence for tRNA-Leu(caa)
1 . 10 . 20 . 30 . 40 . 50
ggtcctgtaggcaaattggcaaagccgctcactcaaaatgacgtgtctg
tgqattcaaatccccacccqaactac

The web-based Aragorn output shows a single tRNA that has a correctly-trimmed 3' end. I will come back to this when I get to the tRNA in my draft annotation.

tRNA-Scan SE:

tRNA-Scan SE is available at:

<http://lowelab.ucsc.edu/tRNAscan-SE/> or linked to from phagesdb.org

I leave the default settings alone, and browse to my file:

Search Mode:

Source:

Format:

Raw Sequence

Sequence name (optional): (no spaces)

Other (FASTA, GenBank, EMBL, GCG, IG)

Paste your query sequence(s) here:

(Queries are limited to a total of less than 5 million nucleotides at any one time)

or submit a file:

Show results in this browser.

Receive results by e-mail instead:

Now I click "Run tRNAscan-SE."

The results are similar to Aragorn, however tRNAscan SE has called the tRNA with one extra base:

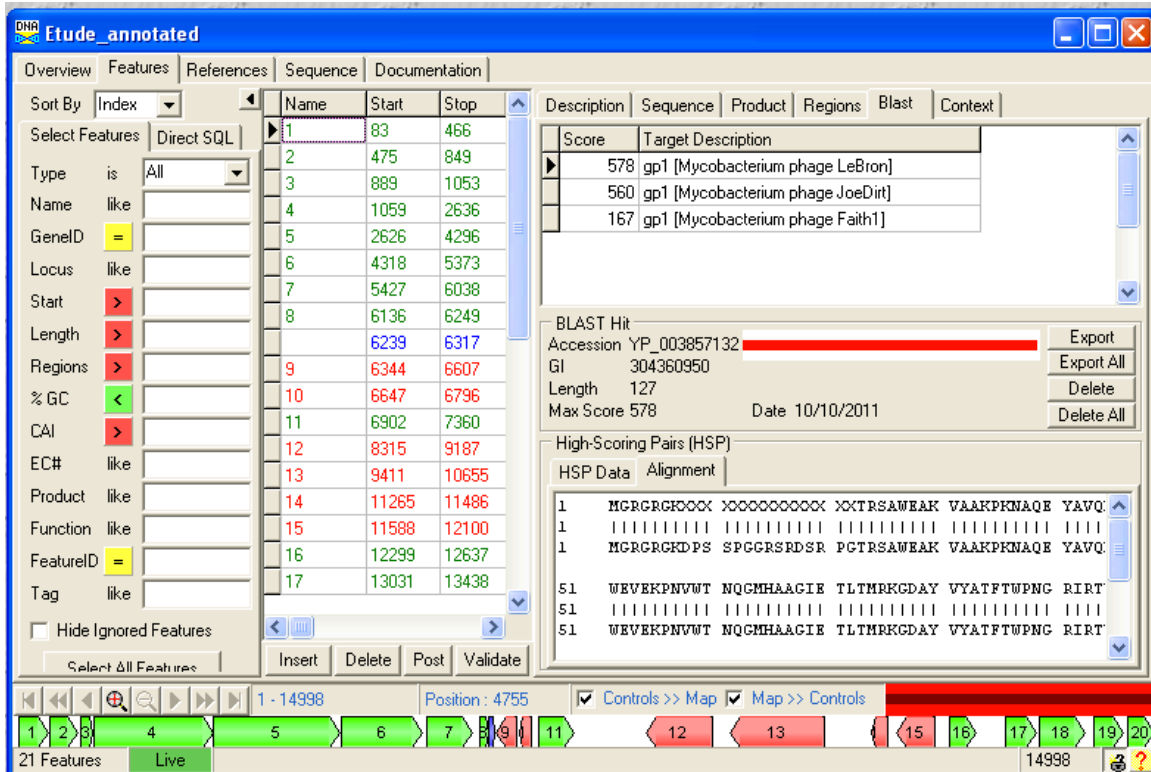
Results

Sequence Name	tRNA #	tRNA Begin	Bounds End	tRNA Type	Anti Codon	Intron Begin	Bounds End	Cove Score
Your-seq	1	6242	6315	Leu	CAA	0	0	62.03

I will evaluate these when I reach the tRNA in the genome sequence.

At this point, I might also make a genome map, however, this genome is so small I can visualize the entire thing at once in DNA Master, so I am going to skip the map.

Whole Genome overview:



My auto-annotation has 21 called genes and 1 tRNA. If I click on the BLAST tab for gene 1, I can see the scores of the alignments from GenBank and the actual alignment.

I can tell from the interactive map at the bottom of my Etude sequence file that there are some large gaps in my genome between genes 11 and 12 and 13 and 14. I will take a closer look at these areas when I reach them in my annotation. There is also a gene overlap with gene 8 and the tRNA. This will also need to be resolved.

Refining my annotation

I now open the Frames window:

Click →DNA→Frames

→Toggle on the features listed in my features table by clicking the “ORFs” button in the lower right-hand corner.



As you mouse over the Frames window, the lower left box will display in real-time the base pair coordinate of your cursor.

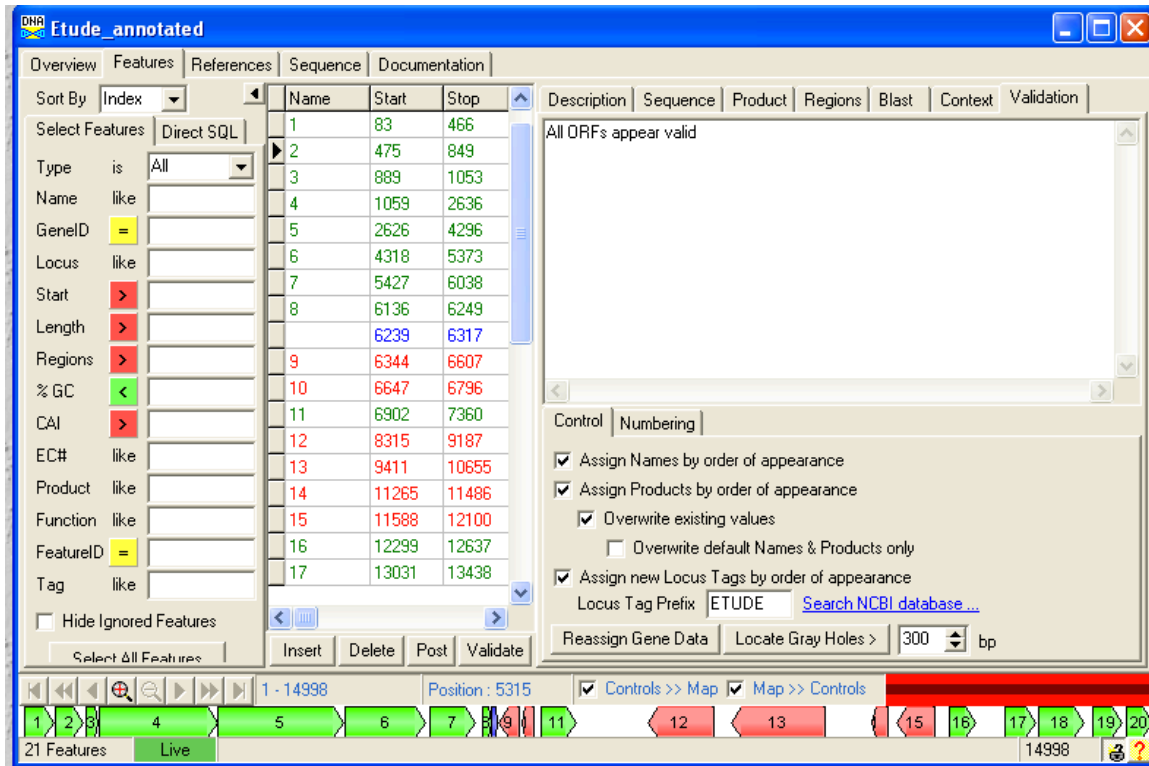
My screen isn't quite big enough to comfortably display both the frame window and my sequence window, so I will be flipping back and forth between the windows.

In the Frames window, I see my forwards transcribed ORFs in green, the reverse in red, and my tRNA in blue. The horizontal tiers represent the six-translational frames, and the full vertical lines within the tiers are stop codons while the half-vertical lines are start codons.

Before I start going through the genes one by one, I am going to validate the gene calls and renumber the genes. The auto-annotation function of DNA Master does not assign gene numbers to tRNAs, however, we do count them in our genbank files. To keep everything consistent, it is easiest to renumber genes initially and then again at the end of the gene identification process. As you grow more confident in your annotation ability, you may also want to renumber periodically as you add and delete genes from the annotation.

To renumber (and assign locus tags):

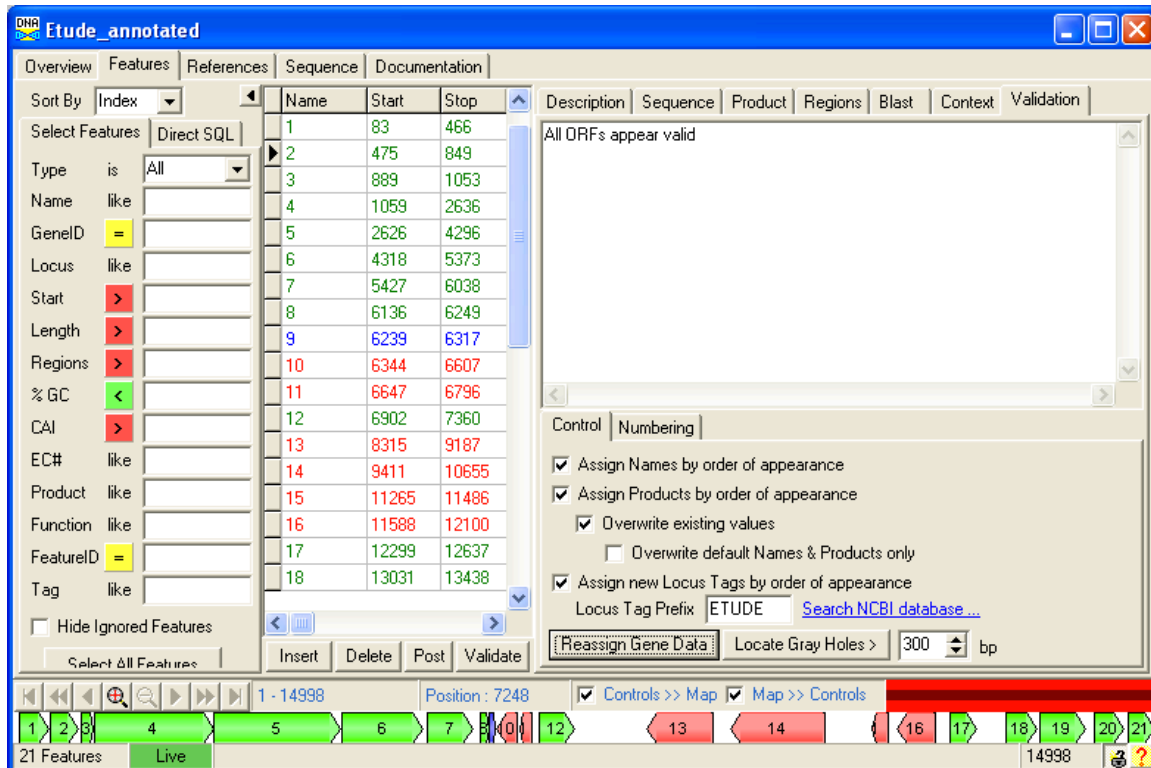
Click → Validate (at the bottom of the central column)



Check the box marked Overwrite existing values

Write the phage's name in the Locus Tag Prefix field. GenBank uses locus tags to assign a unique id to every gene in the database. We prefer to create our GenBank submission files with locus tags comprised of the phage's name and gene number already assigned, to prevent GenBank from assigning every gene a random number.

Click→Reassign Gene Data



The genes have been renumbered (notice the tRNA is now gene 9, whereas before it didn't have a number), and the locus tags have been adjusted.

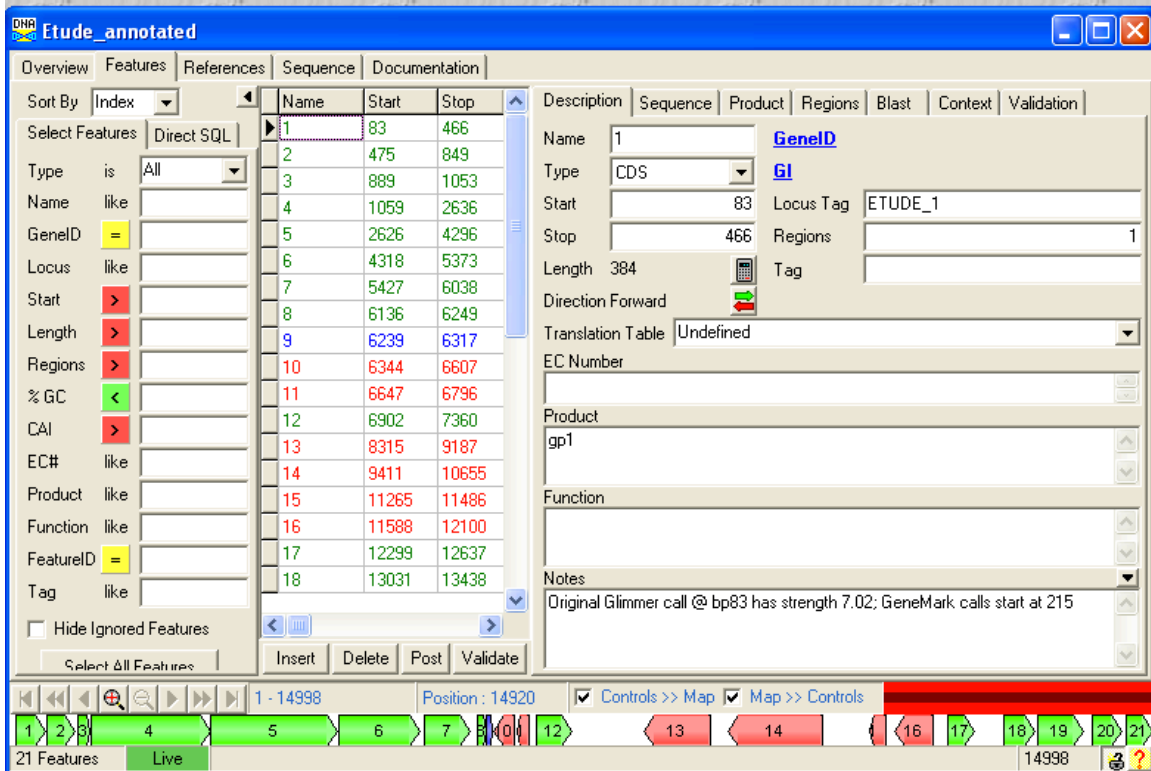
I will now start with Gene 1.

Gene 1:

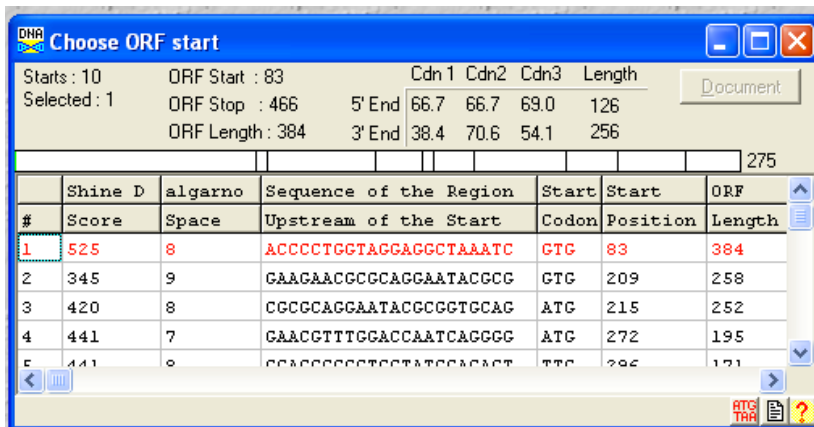
We need to decide: if this gene is a gene, if it is really gene 1, and where its start is. To do this, I will examine five pieces of data: coding potential in GeneMark TB, Glimmer/GeneMark calls, ribosome binding site (RBS) scores, gene gap/overlap with preceding gene, and BLAST alignment with previously annotated genes.

Coding Potential: From looking at the GeneMark TB output, it appears that the coding potential starts in this genome around 200 or so bp. There aren't any ORFs upstream of the called Gene 1 with coding potential, so I am confident that this gene is, in fact, Gene1.

Glimmer/GeneMark: From the Notes field in my auto-annotation, I can tell that GeneMark and Glimmer have called this gene; Glimmer starting at bp 83, and GeneMark starting at bp 215. This means that the GeneMark call does not encompass all of the coding potential as shown by the GeneMark TB output. However, both programs called the gene, and there is good coding potential in the GeneMark TB output. So I am confident that this ORF is a gene, and now just need to resolve what the start coordinate should be.



RBS scores: Click on the first highlighted green bar in the “Frames” window, and then click the “RBS” button at the lower right of the Frames window. A new window will pop-up.



The start at position 83, the one called by Glimmer, has a higher Shine-Dalgarno (RBS) score (525) than the GeneMark start at position 215 (420). The start at position 83 is yields the longest possible gene as well.

Gap/Overlap: Since it is gene 1, we can omit determining the gap or overlap with the upstream gene (as there isn’t one!)

BLAST data: If I click on the BLAST tab (see below), I can see that the genes in GenBank that align well with my gene. Our top hit is (as expected from our phamerator view) to

The screenshot shows the 'Etude_annotated' software interface. The main window displays a protein sequence with various features and annotations. The sequence is: `MCRGRGRKDPSSPGGRSRD SRPCTRSANEAKVAAPKNAQETAVQMAESLQWEVEK PNVVWVWQGMHAAAGIETLTMRKGDAYVYATFTWPNGRIRTVDRVWNGFHEDMFGSE RFDKRRKAMKEILEKYGKZ`. The sequence is 128 residues long, with a molecular weight (MW) of 14.36 kd, a pI of 8.47, a Kyte Hydrophobicity of -0.0983, and an OMH Hydrophobicity of -0.0246. The interface includes a table of features with columns for Name, Start, and Stop. The features are numbered 1 through 18, with corresponding start and stop positions. The interface also includes a search bar, a 'Select Features' section, and a 'Sequence' section. The bottom of the interface shows a navigation bar with a position of 14708 and a map of the sequence.

Name	Start	Stop
1	83	466
2	475	849
3	889	1053
4	1059	2636
5	2626	4296
6	4318	5373
7	5427	6038
8	6136	6249
9	6239	6317
10	6344	6607
11	6647	6796
12	6902	7360
13	8315	9187
14	9411	10655
15	11265	11486
16	11588	12100
17	12299	12637
18	13031	13438

Copy this sequence and paste into NCBI's BLASTP page.

http://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&BLAST_PROGRAMS=blastp&PAGE_TYPE=BlastSearch&SHOW_DEFAULTS=on&LINK_LOC=blasthome

BLAST *Basic Local Alignment Search Tool*

Home Recent Results Saved Strategies Help

NCBI/ BLAST/ blastp suite

blastn **blastp** blastx tblastn tblastx

Enter Query Sequence BLASTP programs search protein databases using a protein query. [more...](#)

Enter accession number(s), gi(s), or FASTA sequence(s) [?](#) Clear Query subrange [?](#)

From
To

Or, upload file [?](#)

Job Title

Enter a descriptive title for your BLAST search [?](#)

Align two or more sequences [?](#)

Choose Search Set

Database [?](#)

Organism Exclude [?](#)

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown. [?](#)

Exclude Models (XM/XP) Uncultured/environmental sample sequences

Entrez Query

Enter an Entrez query to limit search [?](#)

Program Selection

Algorithm

blastp (protein-protein BLAST)

PSI-BLAST (Position-Specific Iterated BLAST)

PHI-BLAST (Pattern Hit Initiated BLAST)

Choose a BLAST algorithm [?](#)

BLAST | Search database Non-redundant protein sequences (nr) using Blastp (protein-protein BLAST)

By pressing the “BLAST” button at the bottom of the page, we get the following result:

Color key for alignment scores

Query 1 20 40 60 80 100 120

Descriptions

Legend for links to other resources: [U](#) UniGene [G](#) GEO [G](#) Gene [S](#) Structure [M](#) Map Viewer [P](#) PubChem BioAssay

Sequences producing significant alignments:

Accession	Description	Max score	Total score	Query coverage	E value	Links
YP_003857132.1	gp1 [Mycobacterium phage LeBron] >gb ADL70968.1 gp1 [Mycobact	263	263	100%	7e-69	G
XP_001528612.1	suppressor of stem-loop protein 1 [Lodderomyces elongisporus NRRL]	36.2	36.2	77%	1.4	G
ZP_03489925.1	hypothetical protein EUBIFOR_02530 [Eubacterium bifforme DSM 3985	33.9	33.9	32%	7.5	
NP_147595.2	putative ornithine cyclodeaminase [Aeropyrum pernix K1] >dbj BAA75	33.1	33.1	63%	9.9	G

Alignments

Select All [Get selected sequences](#) [Distance tree of results](#) [Multiple alignment](#)

```
>ref|YP\_003857132.1| G gp1 [Mycobacterium phage LeBron]
  gb|ADL70968.1| G gp1 [Mycobacterium phage LeBron]
  Length=127

  GENE ID: 9711717_1 | gp1 [Mycobacterium phage LeBron]

  Score = 263 bits (671), Expect = 7e-69, Method: Compositional matrix adjust.
  Identities = 127/127 (100%), Positives = 127/127 (100%), Gaps = 0/127 (0%)

  Query 1  MGRGRGKDPSSPGGRSRDSRPGTRSAWEAKVAAKPKNAQEYAVQMAESLGWEVEKPNVWT 60
             MGRGRGKDPSSPGGRSRDSRPGTRSAWEAKVAAKPKNAQEYAVQMAESLGWEVEKPNVWT
  Sbjct 1  MGRGRGKDPSSPGGRSRDSRPGTRSAWEAKVAAKPKNAQEYAVQMAESLGWEVEKPNVWT 60

  Query 61  NQGMHAAGIETLIMRKGDAYVYATFTWPNGRIRITVDRVVNGFHEDMFGSERDKRKKAMKE 120
             NQGMHAAGIETLIMRKGDAYVYATFTWPNGRIRITVDRVVNGFHEDMFGSERDKRKKAMKE
```

Our top hit is (as expected from our Phamerator view) to LeBron gene 1. LeBron is the only hit listed above that has an acceptable E value, and therefore we discount the other hits and their possible functional assignments.

HHPred.

<http://toolkit.tuebingen.mpg.de/hhpred>

Copy the amino acid sequence into HHPred’s sequence field. Make sure to remove the “Z” at the end (DNA Master represents stop codons with a Z in the product field).

HOME Login PDBAlert Personal Databases Contact Imprint Disclaimer Help

Bioinformatics Toolkit
Max-Planck Institute for Developmental Biology

Quickfinder

Search Alignment Sequence Analysis 2ary Structure 3ary Structure Classification Utils

CS-BLAST HHblits HHpred HHsenser HMMER3 PSI-BLAST PatternSearch ProtBLAST SimShiftDB

HHpred - Homology detection & structure prediction by HMM-HMM comparison [Help](#)

NEW: Official server results for CASP9 structure prediction benchmark

Input

Paste protein sequence or multiple alignment

or upload a local file

Select input format

Search Options

Select HMM databases (hold Ctr to select several)

Standard
 pdb70_8Oct11
 pdb_on_hold_6Oct11
 scop70_1.75
 Interpro_34.0
 pfamA_v25.0

Genomes
 Arabidopsis_thaliana
 Caenorhabditis_elegans
 Drosophila_melanogaster
 Homo_sapiens
 Mus_musculus

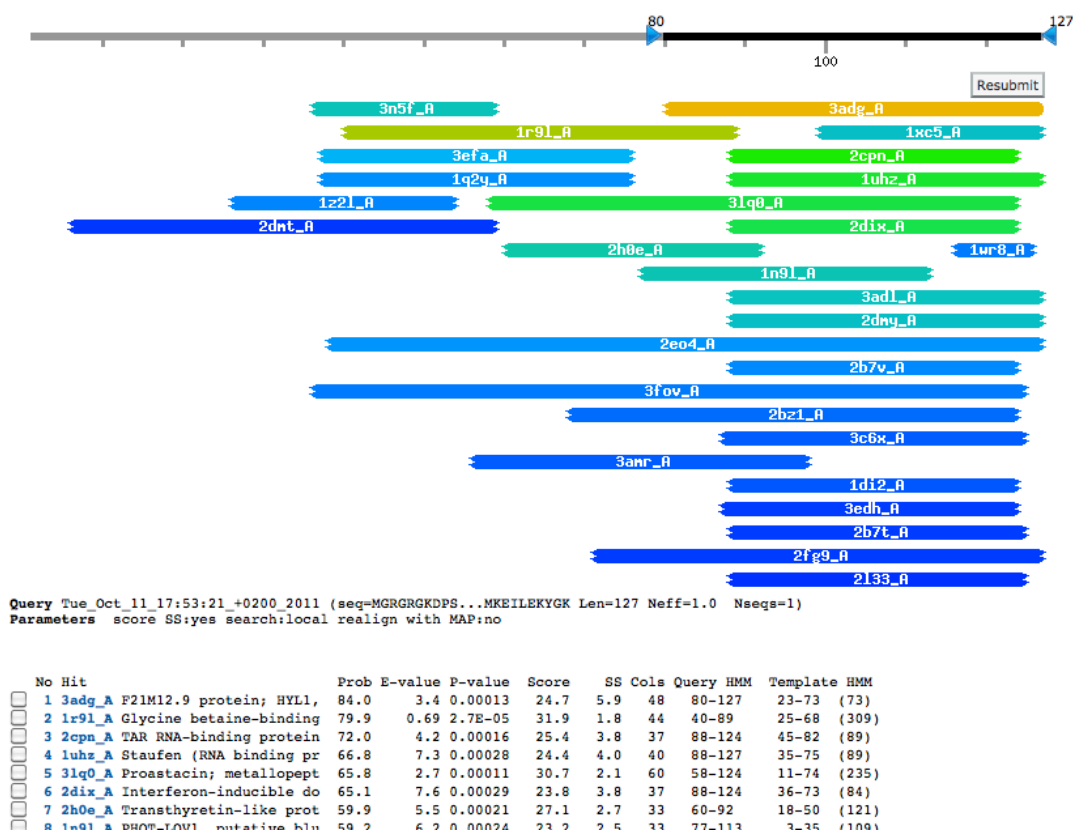
MSA Generation Method HHblits Psiblast

Max. HHblits iterations

Score secondary structure yes no predicted vs predicted only

Alignment mode local global

Click "Submit job" (at the far right, just above the beige bar labeled "Search Options")



Only one of these alignments has a Probability score of above 80, and it from a small portion of our query to a protein in *Arabidopsis thaliana*. We will consider this not a match. Finally, the Hatfull-labeled maps also suggest that there is no known function for gene 1 in LeBron.

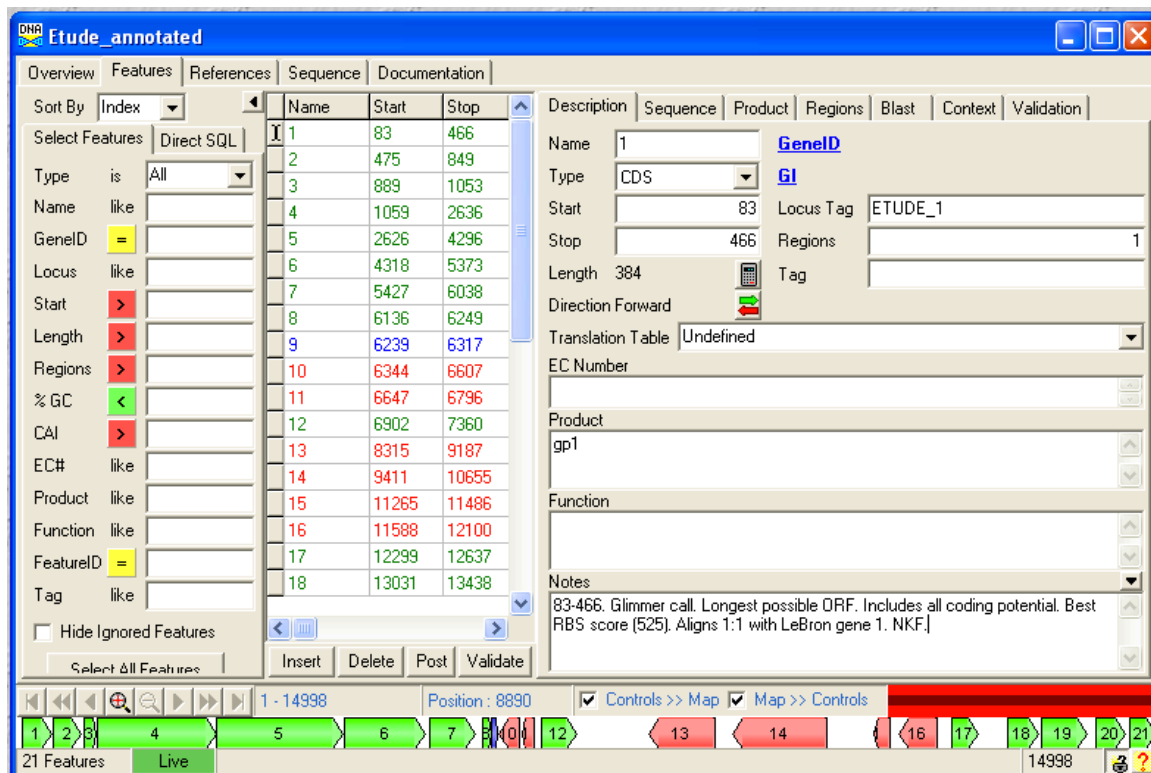
Gene 1 has no known function (NKF).

Our last task is to add our annotation rationale to the Notes field for this gene.

In the Click on the Description tab in the right-hand section of the Features tab.

In the Notes field, add your notes.

Things to include: The gene coordinates for your gene call. Is this the longest possible gene for this gene call? Is this the Glimmer/GeneMark call? What is the gap or overlap between this gene's start and the previous gene's stop? Does this start have the best RBS score? Does this gene match anything in GenBank when you BLAST it? If so, what? What is the alignment between the start that you chose and the closest GenBank match? Is there a known function?



It is important that you physically type in the gene coordinates into your comments, as in some cases I have received files in which people believe they have changed their start coordinates and were not actually able to. In case there are any discrepancies between the gene coordinates that you think you are choosing and the gene coordinates that are actually saved into the file, it is important that you write what your gene coordinate choices are into the notes here.

It is also important that you report the BLAST alignments here, for two reasons: It is possible to generate spreadsheets of the gene data fields, including the notes, that can be very useful for genome checking. These spreadsheets will not include data from the BLAST tab. And if you ever accidentally lose your BLAST data (say, from parsing your documentation, or from corrupting your file) you'll have a record of what the alignment was without having to BLAST your entire genome again.

Post your changes to the notes field, either by clicking "Post", or by moving to gene 2 by clicking on the corresponding row in the central column.

Gene 2:

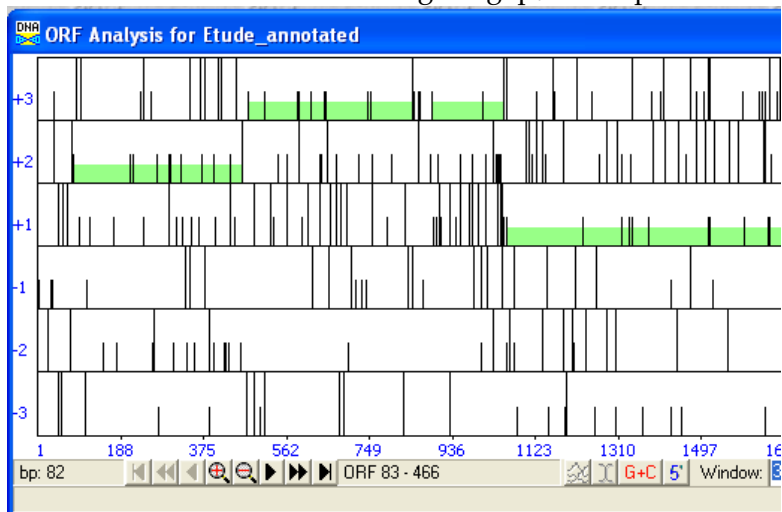
On the feature tab, click "2" in the central column.

In the Notes for gene 2, we can once again see that Glimmer and GeneMark have disagreed on the start for the gene (475 for Glimmer and 514 for GeneMark). However, as with gene 1, we can see that both programs have called the gene, and that there is good coding potential for the gene in the GeneMark TB output. So we will agree that this is a gene, and now just need to resolve its start.

Now we check our three criteria for start selection: coding potential, gene gap/overlap and RBS scores.

Coding Potential: the trace for the GeneMark TB coding potential doesn't start to rise until about bp 600 or so, so both the Glimmer and GeneMark start codons encompass all the coding potential.

Look at the Frames window for gene gap/overlap:



We can see here that the called start (the Glimmer start) represents the longest possible start for this gene—any extension would run into the upstream stop codon. There is no gene overlap, and there is a 9bp gap.

The RBS scores: Click in the box with the second green highlighted bar, and then click the RBS button on the lower right side of the frames window.

Choose ORF start

Starts : 9 ORF Start : 475 Cdn1 Cdn2 Cdn3 Length
 Selected : 1 ORF Stop : 849 5' End 38.5 69.2 46.2 39
 ORF Length : 375 3' End 41.1 68.5 62.2 334

Document

#	Shine D	algarno	Sequence of the Region	Start	Start	ORF
	Score	Space	Upstream of the Start	Codon	Position	Length
1	378	8	AGTACGGCAAGTAGGGGGATT	ATG	475	375
2	150	9	AAACGCTGTTCCACCTGCGCCTT	ATG	514	336
3	210	7	TGACGCTCCGTCGAGGGTCCTT	TTG	586	264
4	504	7	CGCTCCGTCGAGGGTCCTTTG	TTG	589	261
5	252	7	CCCTCCGCTATCCCTCAGACC	TTG	616	224

795

Again, the Glimmer call at 475 has a higher score than the GeneMark call at 514.

Examine the BLAST tab:

Etude_annotated

Overview Features References Sequence Documentation

Sort By Index

Select Features Direct SQL

Type is All

Name like

GenelD =

Locus like

Start >

Length >

Regions >

% GC <

CAI >

EC# like

Product like

Function like

FeatureID =

Tag like

Hide Ignored Features

Select All Features

Name	Start	Stop
1	83	466
2	475	849
3	889	1053
4	1059	2636
5	2626	4296
6	4318	5373
7	5427	6038
8	6136	6249
9	6239	6317
10	6344	6607
11	6647	6796
12	6902	7360
13	8315	9187
14	9411	10655
15	11265	11486
16	11588	12100
17	12299	12637
18	13031	13438

Insert Delete Post Validate

Score	Target Description
572	gp2 [Mycobacterium phage JoeDirt]
560	gp2 [Mycobacterium phage LeBron]
437	gp2 [Mycobacterium phage Faith1]
193	hypothetical protein SVEN_3985 [Streptomyces]
191	gp10 [Mycobacterium phage Omega]

BLAST Hit

Accession AEK07049

GI 339781215

Length 124

Max Score 572 Date 10/10/2011

Export
Export All
Delete
Delete All

High-Scoring Pairs (HSP)

HSP Data Alignment

```

1  MGDTVKNAV PGLMAAGKEL WESVASEREL DAPSRVLLLN ACRI
1  |
1  MGDTVKNAV PGLMAAGKEL WESVASEREL DAPSRVLLLN ACRI
51  LDQEIDGRLL SYNQRGDEVI NPLISEHRQQ YTTLANILGK MGLG
51  |
51  LDQEIDGRLL SYNQRGDEVI NPLISEHRQQ YTTLANILGK MGLG
  
```

1 - 14998 Position: 13780 Controls >> Map Map >> Controls

21 Features Live 14998

Once again, the best match aligns 1:1 with JoeDirt and LeBron.

So we will pick the Glimmer call at 475 as our gene start, and enter the appropriate description into the notes. Now that we have an upstream gene, we will also write the gap/overlap in bp of this gene with the previous one.

Functional assignment: If we BLASTP this gene outside of DNA Master on the NCBI website, or examine Phamerator, or the Hatfull-approved genome maps with functions, we will see this gene is the small subunit of the terminase.

Click on the product tab in DNA Master (or on the gene in Phamerator)

The screenshot shows the 'Etude_annotated' software interface. The 'Features' tab is active, displaying a table of features. Feature 2 is selected, and its details are shown in the right-hand pane. The amino acid sequence for feature 2 is displayed in the 'Sequence' pane.

Name	Start	Stop
1	83	466
2	475	849
3	889	1053
4	1059	2636
5	2626	4296
6	4318	5373
7	5427	6038
8	6136	6249
9	6239	6317
10	6344	6607
11	6647	6796
12	6902	7360
13	8315	9187
14	9411	10655
15	11265	11486
16	11588	12100
17	12299	12637
18	13031	13438

Feature 2 details:

- Description: 125 Residues
- MW = 13.76 kd
- Kyte Hydro = -0.0587
- pl = 7.75
- OMH Hydro = -0.0230

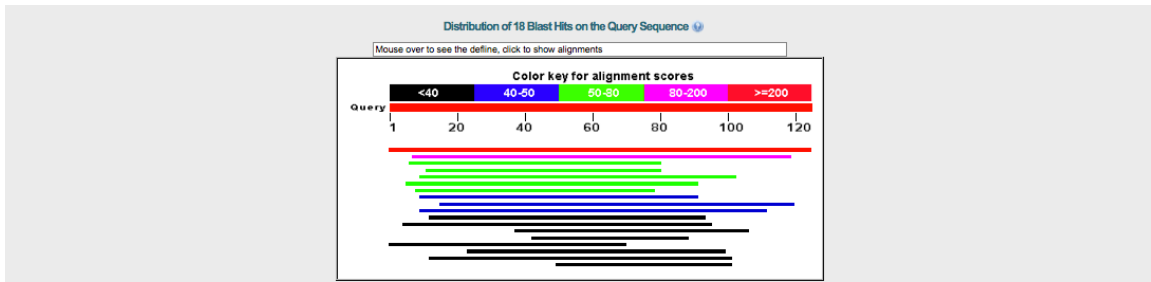
Amino acid sequence for Feature 2:

```

MCDTVKNAVPPGLMAAGKELWESVASEREILDAPSRVLLLNACRIADRLDQLDQEI
DGRLLSYNQRGDEVINPLISEHRQYYTTLANILGRMGLGELPKAKQENSRWDELA
KKRAERAAKAAQASZ
    
```

The interface also shows a histogram of amino acid frequencies at the bottom of the sequence pane and a navigation bar at the bottom of the window.

As in gene 1, copy and paste the amino acid sequence into the NCBI BLASTP page. The BLAST result show more hits this time, not only LeBron, but multiple other phages. This gene is the small subunit of the terminase, and so we must add the function into our annotation. The LeBron alignment is still "query 1 to subjct 1", indicating that gene 2 of LeBron and our gene 2 of Etude use the same start codon.



Descriptions

Legend for links to other resources: [UniGene](#) [GEO](#) [Gene](#) [Structure](#) [Map Viewer](#) [PubChem BioAssay](#)

Sequences producing significant alignments:

Accession	Description	Max score	Total score	Query coverage	E value	Links
YP_003857133.1	gp2 [Mycobacterium phage LeBron] >gb ADL70969.1 gp2 [Mycobacterium phage LeBron]	246	246	100%	5e-64	G
NP_818311.1	gp10 [Mycobacterium phage Omega] >gb AAN12654.1 gp10 [Mycobacterium phage Omega]	87.8	87.8	89%	4e-16	G
ZP_06825994.1	hypothetical protein SSBG_02573 [Streptomyces sp. SP874] >gb EDY44611.1 hypothetical protein SSBG_02573 [Streptomyces sp. SP874]	72.4	72.4	59%	2e-11	
YP_002882537.1	hypothetical protein Bcav_2527 [Beutenbergia cavernae DSM 12333] >gb ACQ80775.1	57.0	57.0	55%	7e-07	G
YP_002781224.1	hypothetical protein ROP_40320 [Rhodococcus opacus B4] >dbj BAH52279.1	56.2	56.2	75%	1e-06	G
YP_003162104.1	hypothetical protein Jden_2164 [Jonesia denitrificans DSM 20603] >gb ACV09801.1	53.5	53.5	69%	8e-06	G
YP_706493.1	hypothetical protein RHA1_ro06562 [Rhodococcus jostii RHA1] >gb ABG98335.1	50.1	50.1	56%	9e-05	G
ZP_06501003.1	phage terminase, small subunit, P27 family [Micrococcus luteus SK58] >gb EFD51948.1	48.5	48.5	66%	2e-04	
ZP_07714862.1	conserved hypothetical protein [Corynebacterium pseudogenitalium ATCC 33035] >gb EF	41.6	41.6	83%	0.029	
YP_655890.1	gp25 [Mycobacterium phage Wildcat] >gb ABE67630.1 gp25 [Mycobacterium phage Wildcat]	40.4	40.4	82%	0.074	G
ZP_06832385.1	conserved hypothetical protein [Rhodococcus equi ATCC 33707] >gb EFG59276.1	37.7	37.7	65%	0.43	
ZP_03646336.1	hypothetical protein BbiFN4_04215 [Bifidobacterium bifidum NCIMB 41171] >ref YP_003	37.7	37.7	73%	0.43	
YP_885333.1	serine 3-dehydrogenase [Mycobacterium smegmatis str. MC2 155] >gb ABK70390.1	37.7	37.7	55%	0.44	
YP_003886949.1	sodium/hydrogen exchanger [Cyanotheca sp. PCC 7822] >gb ADN13674.1 sodium/hydr	34.7	34.7	37%	4.4	G
ZP_04402134.1	MSHA biogenesis protein MshM [Vibrio cholerae TMA 21] >gb EEO15293.1	34.3	34.3	56%	5.1	
XP_864435.1	PREDICTED: similar to sterile alpha motif domain containing 4 isoform 4 [Canis familiaris	33.9	33.9	61%	5.9	UGM
YP_002894263.1	transketolase [Tolomonas auensis DSM 9187] >gb ACQ94677.1 transketolase [Tolomon	33.5	33.5	71%	9.0	G
ZP_06399327.1	acyl-CoA dehydrogenase domain protein [Micromonospora sp. L5] >ref YP_003833223.1	33.5	33.5	41%	9.7	

Alignments

Select All [Get selected sequences](#) [Distance tree of results](#) [Multiple alignment](#)

```
> ref|YP\_003857133.1 G gp2 [Mycobacterium phage LeBron]
  gb|ADL70969.1 G gp2 [Mycobacterium phage LeBron]
  Length=124

  GENE ID: 9711608.2 | gp2 [Mycobacterium phage LeBron]

  Score = 246 bits (629), Expect = 5e-64, Method: Compositional matrix adjust.
  Identities = 122/124 (99%), Positives = 122/124 (99%), Gaps = 0/124 (0%)

  Query 1  MGDTVKNAVPPGLMAAGKELWESVASERELDAPSRVLLLNACRIADRLDQLDQEQIDGRLL 60
           MGD VKN VPPGLMAAGKELWESVASERELDAPSRVLLLNACRIADRLDQLDQEQIDGRLL
  Sbjct 1  MGDGVKNTVPPGLMAAGKELWESVASERELDAPSRVLLLNACRIADRLDQLDQEQIDGRLL 60

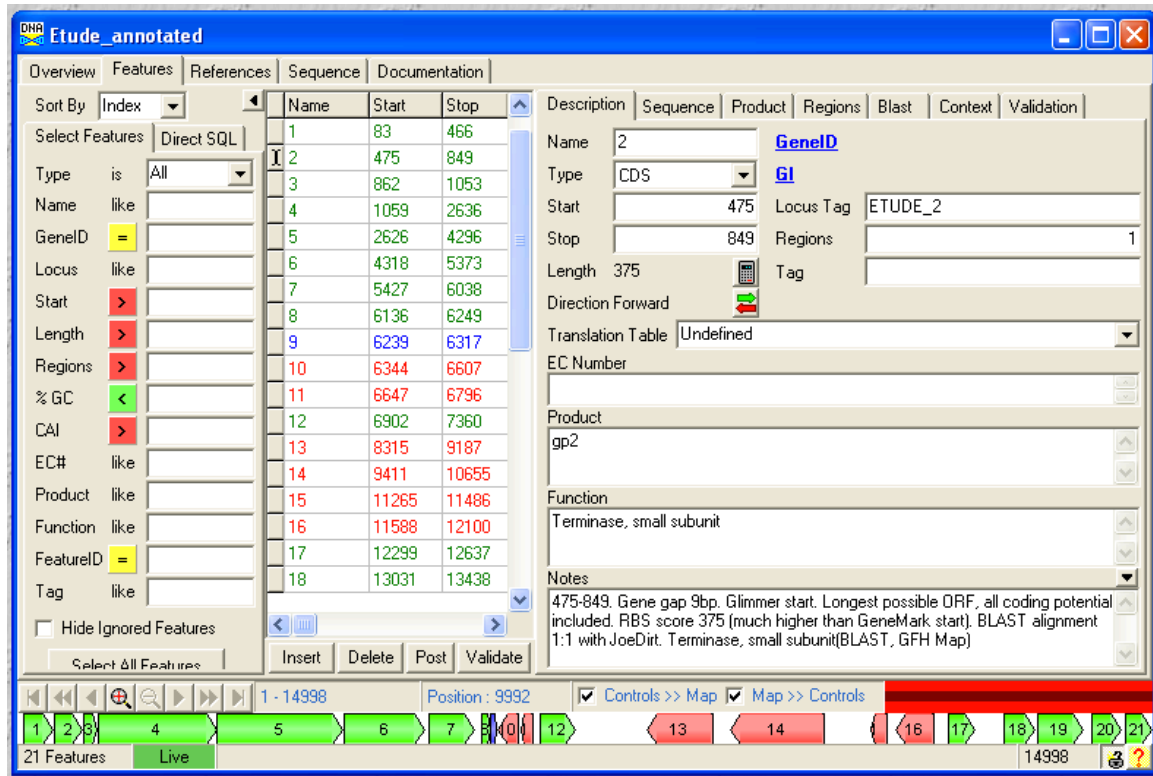
  Query 61 SYNQRGDEVINPLISEHRQQYTTLANILGKMGLGELPKAQENSRWDELAKKRAERAAKA 120
           SYNQRGDEVINPLISEHRQQYTTLANILGKMGLGELPKAQENSRWDELAKKRAERAAKA
  Sbjct 61 SYNQRGDEVINPLISEHRQQYTTLANILGKMGLGELPKAQENSRWDELAKKRAERAAKA 120

  Query 121 AQAS 124
           AQAS
  Sbjct 121 AQAS 124
```

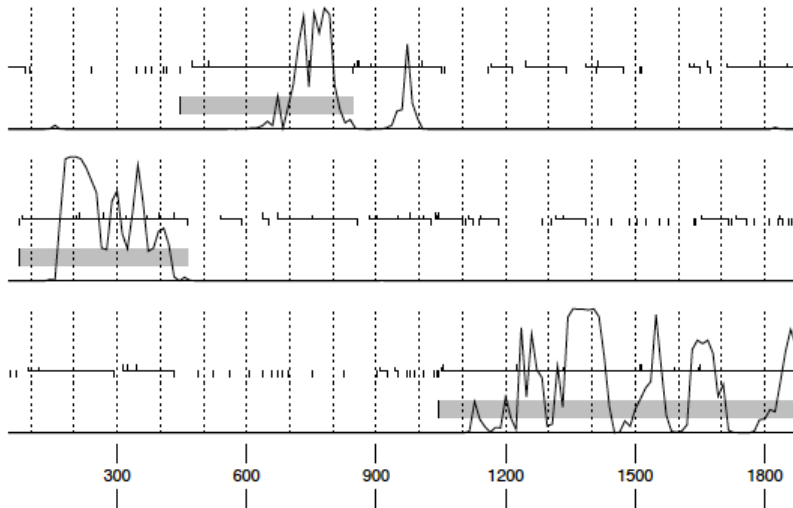
We are confident enough with the BLASTP and map assignments that it is not necessary to run HHPred.

Add detailed annotation notes as in gene 1. Make sure that you include the gene gap/overlap, the functional assignment, and the source of your functional assignment.

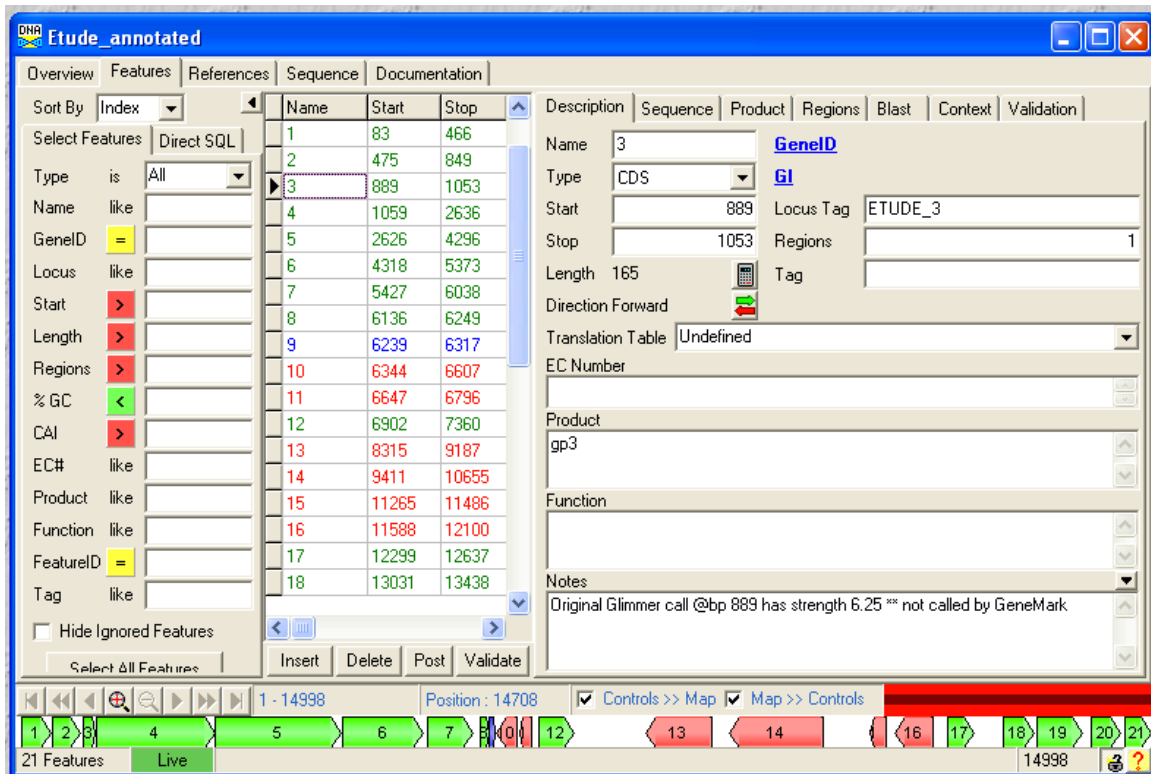
Since this is a Hatfull Map approved function, we will add it to the Function field as well:



Gene 3: Look at the coding potential trace in the GeneMark TB output. The coding potential after Gene 2 shows a smaller peak in the top tier following the gene 2 peak.

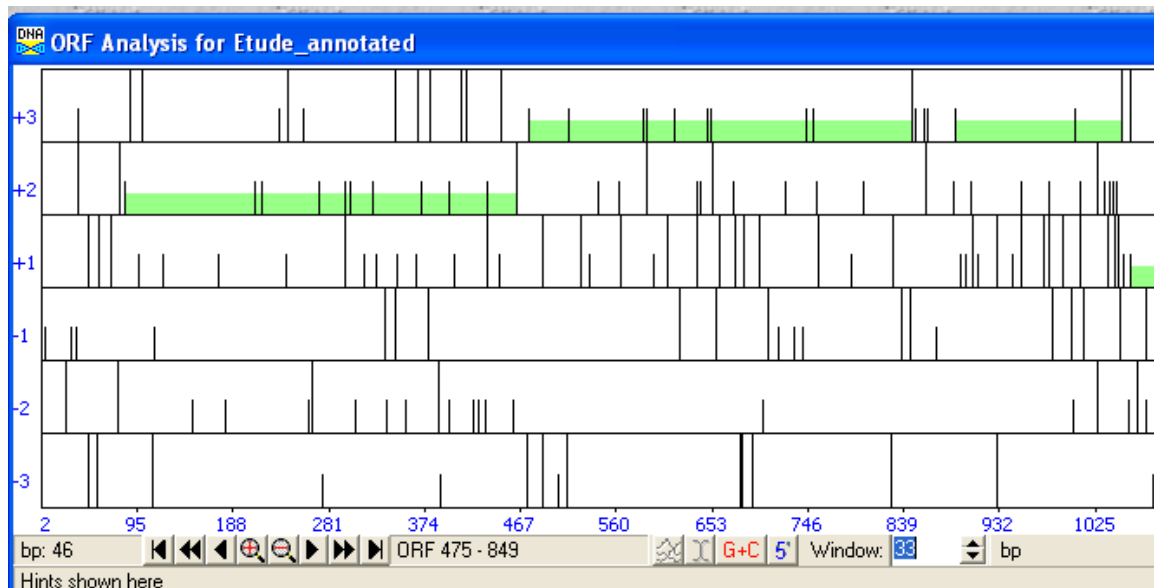


Glimmer has decided to call this ORF a gene, GeneMark has decided to omit it.



Since we see some coding potential in this frame and not in any others, and this ORF nicely fills a gap in the genome between gene 2 and gene 4, we are going to call this gene.

Now we need to pick a start codon. Examine the frames window:



There are at least four possible starts for gene 3 that will not overlap with gene 2 (we can't overlap gene 2 at all in this case, as genes 2 and 3 are in the same frame. Gene 2's stop codon would prevent translation of gene 3 from any earlier start.)

Now we use our five pieces of data to determine which start of the four possible starts we like the best.

Coding Potential: the earliest blip in the GeneMark TB coding potential trace is about 900 bp, so all four starts encompass all the coding potential.

Gene gap/overlap: The best start here is the longest start, which leaves no gap between genes. However, the tandem starts (start 2 and start 3) leave a fairly small gap as well, either 10 or 13 bp.

RBS scores:

#	Shine D	algarno	Sequence of the Region	Start	Start	ORF
#	Score	Space	Upstream of the Start	Codon	Position	Length
1	247	6	TAAAGCCGCTCAGGCCCTCTAG	GTG	850	204
2	247	6	TCAGGCCCTCCTAGCTGCCGGGG	ATG	859	195
3	312	6	GGCGTCCTAGGTCCGGGGGATC	ATG	862	192
4	252	7	GTCCATTTCGTTCCGGCCGGGCT	GTG	889	165
5	399	7	CTACTGCTATAGCGCCCTCAT	ATG	1006	48

According to DNA master, starts 1 and 2 have exactly the same score, which is only very slightly lower than start 4 (the Glimmer start). Start 3 has the best score of 312.

At this point, we need to weigh whether we minimize the gap (start 1 at bp 850) or pick the best SD score (start 3 at bp 862). An interesting biological discussion!

I will check the BLAST results, too:

The screenshot shows the Etude_annotated software interface. The main window displays a list of features with columns for Name, Start, and Stop. A BLAST hit is shown in the right panel, indicating a match with 'gp3 [Mycobacterium phage UPIE]' with a score of 282 and a length of 63. The HSP alignment shows a 9-amino acid gap at the start of the query sequence.

Name	Start	Stop
1	83	466
2	475	849
3	889	1053
4	1059	2636
5	2626	4296
6	4318	5373
7	5427	6038
8	6136	6249
9	6239	6317
10	6344	6607
11	6647	6796
12	6902	7360
13	8315	9187
14	9411	10655
15	11265	11486
16	11588	12100
17	12299	12637
18	13031	13438

This indicates that we are 9 amino acids short of the called start in the best BLAST hit, that of UPIE. 9 amino acids is 18 bp, or equivalent to the start at 862.

Without wet lab evidence, we can't really say. So at this point, I will choose the best SD score at (bp 862) and the BLAST alignment. I could just as easily choose to minimize the gap, as the SD score is not too terrible for start 1 and we frequently see phage genes that start and stop practically on top of each other in this manner. Both are valid choices.

Change the gene start in the description tab:

→ Write the new coordinate in the start field, and then click the calculator icon to change the gene length and post the change to the database.

The screenshot shows the DHA Etude_annotated software interface. The main window is titled "Etude_annotated" and has tabs for Overview, Features, References, Sequence, and Documentation. The "Features" tab is active, displaying a table of features with columns for Name, Start, and Stop. Feature 3 is selected, and its details are shown in the right-hand pane.

Name	Start	Stop
1	83	466
2	475	849
3	862	1053
4	1059	2636
5	2626	4296
6	4318	5373
7	5427	6038
8	6136	6249
9	6239	6317
10	6344	6607
11	6647	6796
12	6902	7360
13	8315	9187
14	9411	10655
15	11265	11486
16	11588	12100
17	12299	12637
18	13031	13438

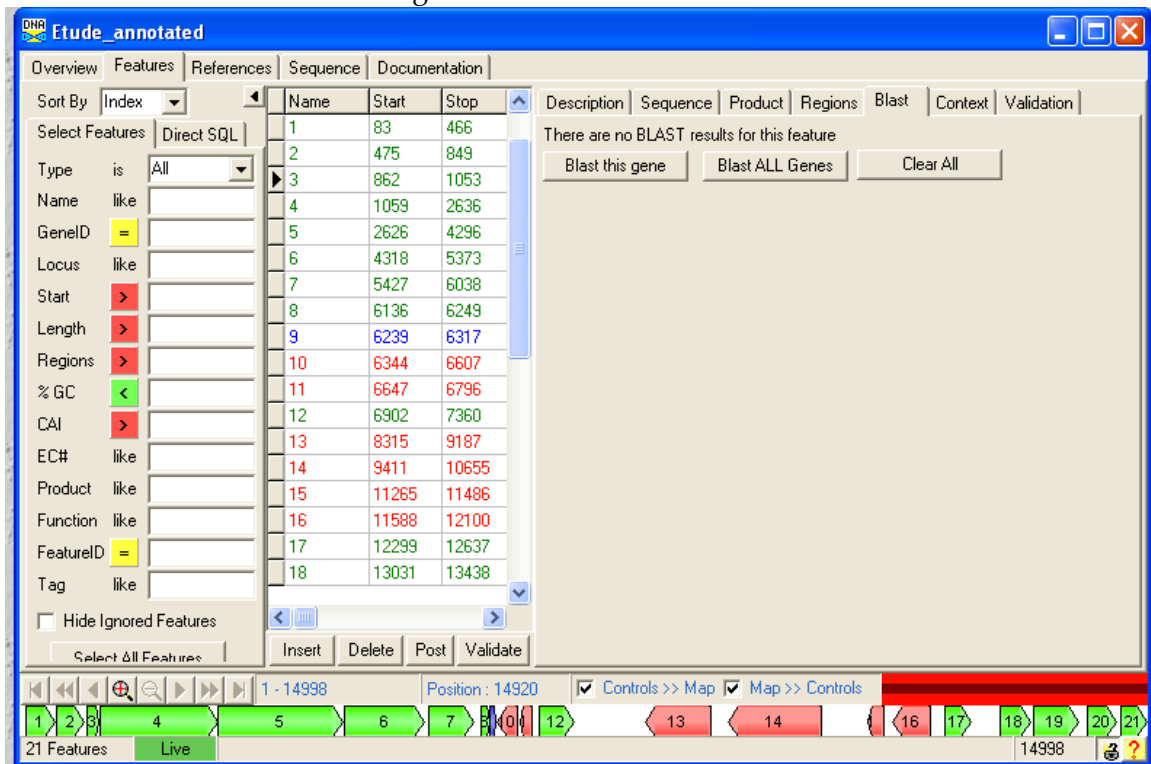
Details for Feature 3:

- Name: 3 (GenelD)
- Type: CDS (GI)
- Start: 862
- Stop: 1053
- Length: 192
- Direction: Forward
- Translation Table: Undefined
- EC Number: (empty)
- Product: gp3
- Function: (empty)
- Notes: Original Glimmer call @bp 889 has strength 6.25 ** not called by GeneMark

The bottom of the interface shows a genomic map with 21 features represented by colored arrows. The current position is 4445, and the range is 1-14998. The "BLAST" tab is highlighted in red.

Now reBLAST the gene: click the BLAST tab.

Click "Delete all". Click yes in the box that pops up that asks you if you really want to do this. Then click "BLAST this gene"

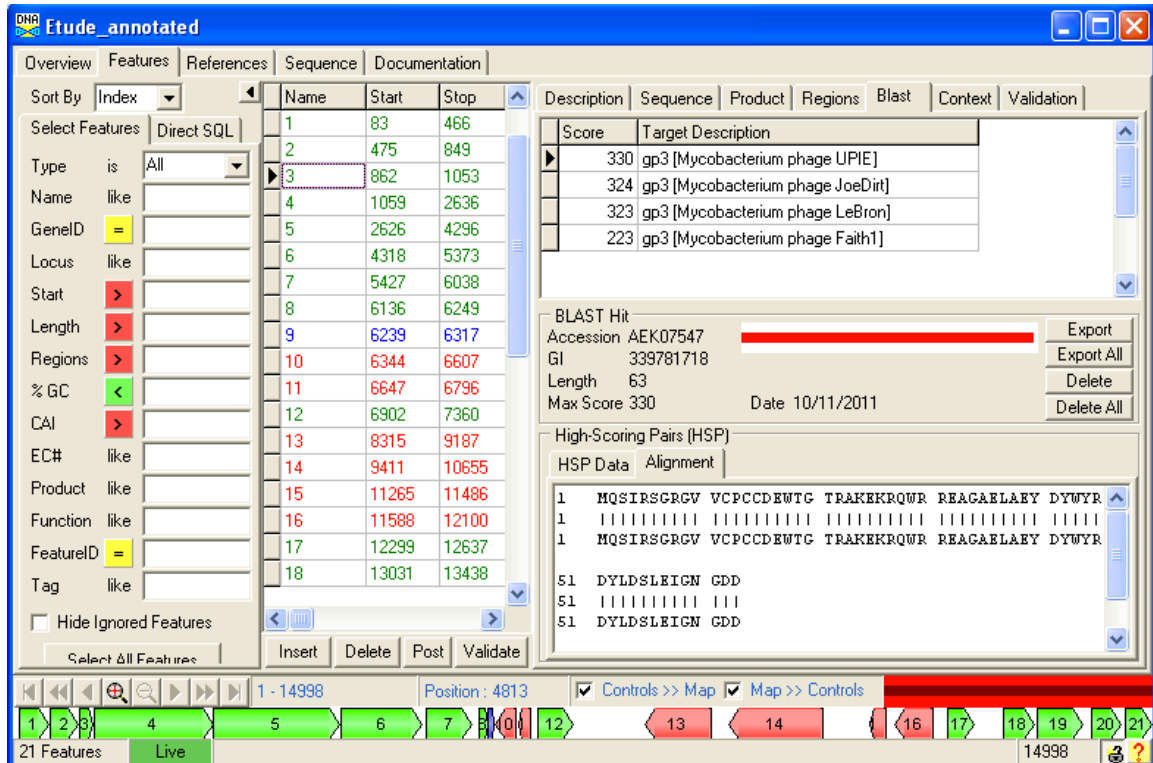


A new window will appear, showing your BLAST request status. When it finishes, it will load your data and look like this:

Click the tab that says “Save to Database”

Only the top four hits have reasonable E values, so we will only save those four to our genome database. I will click “save 4 values”. Then I close the window.

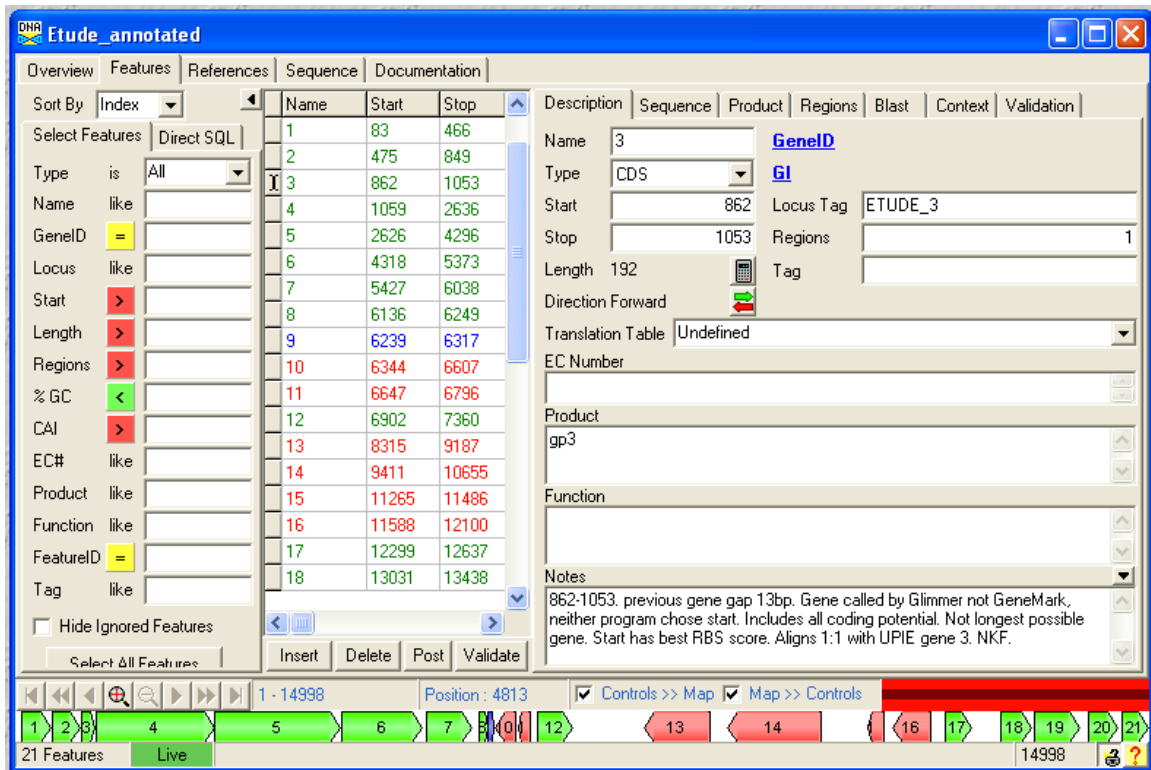
Back in my main genome window, the BLAST data has been altered for gene 3 (I had to click gene 2 and then back to gene 3 for it to load into the window):



Now when we BLAST the amino acid sequence of gene 3, it matches the UPIE annotation gene 3, with the "query 1" aligning with the "sbjct 1". This makes me feel better about the start that I chose, for even though both starts were good choices, it is nice to be consistent with a similar genome. That way, in the future, any wet bench data that we get about this gene from one phage will be easily applied to similar genomes.

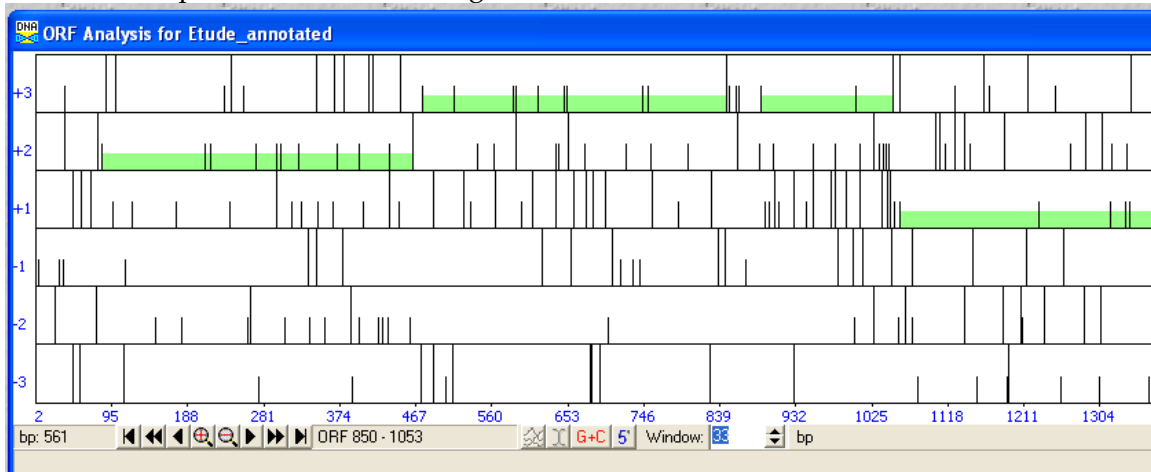
Check for functions via BLAST, Phamerator, HHPred, and the Hatfull Maps. None of these return a known or likely function (the best HHPred match is to a zinc-finger protein in Homo sapiens).

Add detailed annotation info to the notes on the Description tab:



Gene 4:

Back to the coding potential trace in GeneMark TB. Gene four is in frame three. It looks like the coding potential starts around bp ~980 or 990. According to the frames window, there are two possible starts for this gene:



1053 and 1059 (the start called by both Glimmer and GeneMark).

We already know that both starts encompass all the coding potential, and that both have minimal gaps. The final piece of data is the RBS score:

#	Shine D	algarno	Sequence of the Region	Start	Start	ORF
	Score	Space	Upstream of the Start	Codon	Position	Length
1	357	8	AGATTGGTAATGGTGATGATTA	GTG	1053	1584
2	483	8	GTAATGGTGATGATTAGTGGCA	ATG	1059	1578
3	315	8	TACATCCCTAGAAGACGACGGG	ATG	1230	1407
4	294	7	TCTGGGCCGCGACTAAAGAGGGT	TTG	1317	1320
5	420	7	CCCTTTCTCCTCCCGCCCGCAC	CTC	1325	1302

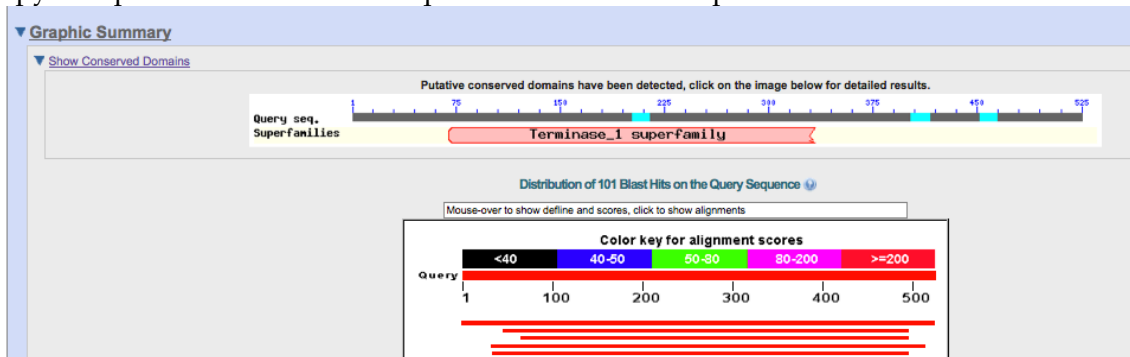
Here we come to one of those gray areas in annotation. The RBS score of the first start is lower, but not much lower. The gap between genes is smaller with the first start, but not much smaller. And both algorithms selected the second start.

BLAST: The data from the BLAST tab indicates that the algorithms have selected the same start as the genes already in GenBank.

So I am going to pick the Glimmer/GeneMark start.

Functional assignment:

Copy and paste the amino acid sequence into a BLASTP pane at NCBI.



▼ Descriptions

Legend for links to other resources: [UniGene](#) [GEO](#) [Gene](#) [Structure](#) [Map Viewer](#) [PubChem BioAssay](#)

Sequences producing significant alignments:

Accession	Description	Max score	Total score	Query coverage	E value	Links
YP_003857135.1	gp4 [Mycobacterium phage LeBron] >gb ADL70971.1 gp4 [Mycobacterium phage LeBron]	1083	1083	100%	0.0	G
YP_655891.1	gp26 [Mycobacterium phage Wildcat] >gb ABE67631.1 gp26 [Mycobacterium phage Wil]	242	242	85%	8e-62	G
YP_002281225.1	hypothetical protein ROP_40330 [Rhodococcus opacus B4] >dbj BAH52280.1 hypothetic	221	221	82%	2e-55	G
ZP_03927248.1	phage Terminase [Actinomyces urogenitalis DSM 15434] >gb EH65874.1 phage Termin	216	216	91%	8e-54	G
YP_001800806.1	hypothetical protein cur_1412 [Corynebacterium urealyticum DSM 7109] >emb CAQ0537	211	211	88%	2e-52	G
ZP_06185208.1	putative phage terminase, large subunit [Mobiluncus mulieris 28-1] >gb EEZ90351.1 pu	197	197	85%	3e-48	
ZP_07452355.1	possible phage-related terminase [Mobiluncus mulieris ATCC 35239] >gb EFM46107.1 p	195	195	84%	1e-47	
YP_003490422.1	putative phage terminase [Streptomyces scabiei 87.22] >emb CBG71879.1 putative pha	194	194	88%	2e-47	G
ZP_07608233.1	Terminase [Streptomyces violaceusniger Tu 4113] >gb EFN16280.1 Terminase [Streptor	191	191	91%	3e-46	

▼ Alignments

Select All [Get selected sequences](#) [Distance tree of results](#) [Multiple alignment](#)

```
>ref|YP_003857135.1| G gp4 [Mycobacterium phage LeBron]
gb|ADL70971.1| G gp4 [Mycobacterium phage LeBron]
Length=525

GENE ID: 9711610_4 | gp4 [Mycobacterium phage LeBron]

Score = 1083 bits (2801), Expect = 0.0, Method: Compositional matrix adjust.
Identities = 525/525 (100%), Positives = 525/525 (100%), Gaps = 0/525 (0%)

Query 1   MTVIPSIPTDRTVSESDLWTFIDEKAREWSDKGLIGAQKPRLSNYPTFFTSLEDDGMDF 60
Sbjct 1   MTVIPSIPTDRTVSESDLWTFIDEKAREWSDKGLIGAQKPRLSNYPTFFTSLEDDGMDF 60

Query 61   IEAYGNLLPWQEQALFRASLGRTKEGLWSARQVCLIVPRQGGTELLEAREFFGLPGLNE 120
Sbjct 61   IEAYGNLLPWQEQALFRASLGRTKEGLWSARQVCLIVPRQGGTELLEAREFFGLPGLNE 120

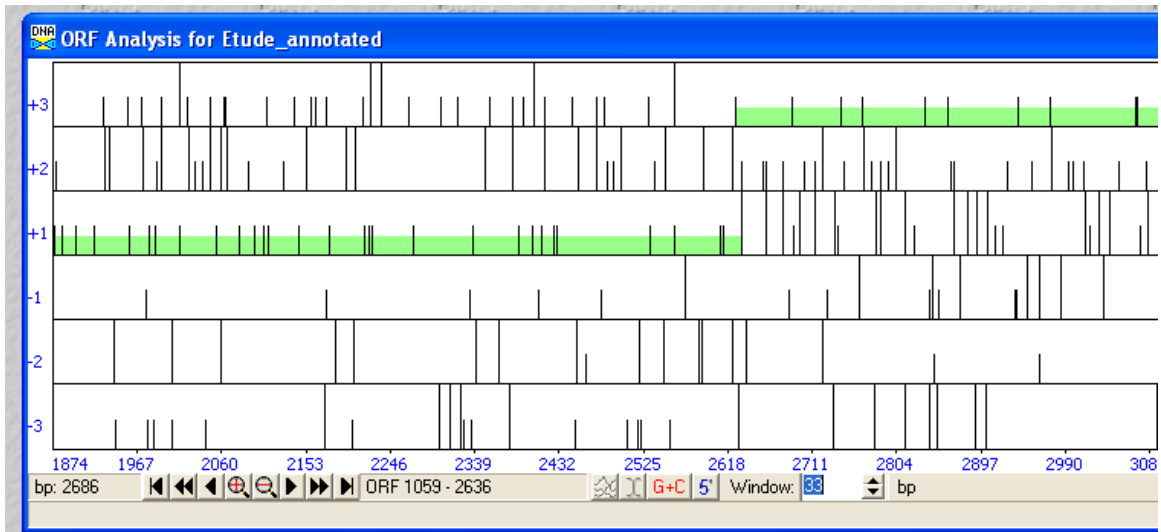
Query 121  RIFHTSQAKTNTQAWQSLTAKIDSPFDLEELMMPHKNNGGEVSIIRLKKTGSNPEPGFVR 180
Sbjct 121  RIFHTSQAKTNTQAWQSLTAKIDSPFDLEELMMPHKNNGGEVSIIRLKKTGSNPEPGFVR 180
```

Gene 4 is the large subunit of the terminase, which is part of the DNA packaging machinery (helps to stuff the DNA into the new phage head). We got a conserved domain hit and numerous phage hits. Once again, our best match is to LeBron, and we once again align perfectly, with the Query 1 matching the Sbjct 1. The assignment is supported by the Hatfull Maps, and running HHPred is not necessary.

I will make the appropriate notes in the Notes field and Function field.

Gene 5:

Gene 5 is the easiest gene by far that we have looked at. Both Glimmer and GeneMark call this gene, the GeneMark TB coding potential starts around ~2650, and there is only one start codon (at 2626) that neither overlaps gene 4 too much and encompasses all the coding potential. In fact, there is only one start codon in the correct frame for this gene. Notice there is a small 10bp overlap between genes now. This is OK, overlaps need to be much larger before we discount them.



BLAST tab results indicate that this gene is a perfect match 1:1 with LeBron gene 4.

Done!

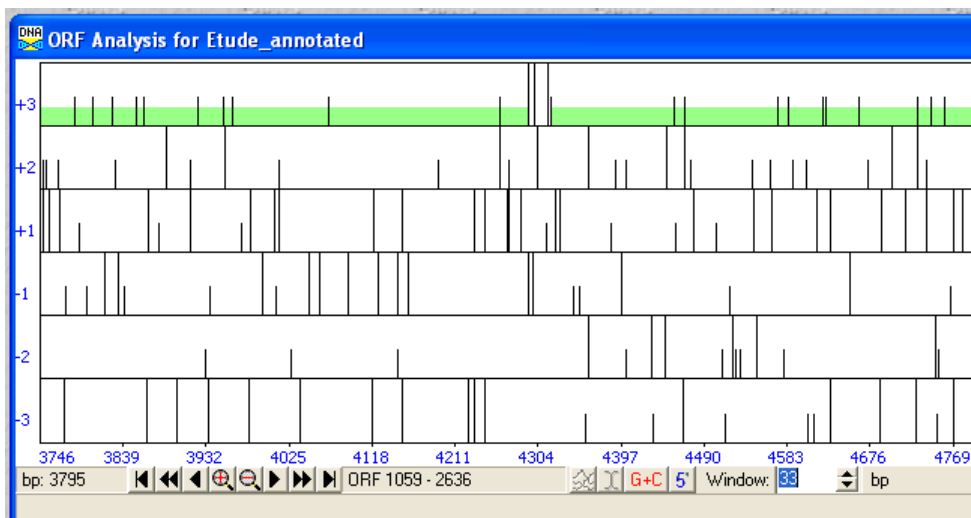
Paste the amino acid sequence into a BLAST p pane at NCBI.

This protein is a phage portal protein and forms a dodecameric ring at the vertex of the capsid that the DNA is threaded through and that the tail then joins to. We once again match the LeBron gene call perfectly. The Hatfull Maps support this assignment.

I will write the appropriate notes in the Notes field and Function field.

Gene 6:

Again, both the Glimmer and GeneMark calls agree on a single start that does not have any close starts near it in the same frame and is the longest possible start for this gene.



The BLAST tab data shows a 1:1 alignment with LeBron's gene 6. The Hatfull Map shows that this gene is the capsid maturation protease (frequently found after the portal gene in phage genomes). This protease cleaves the scaffolding protein in the immature

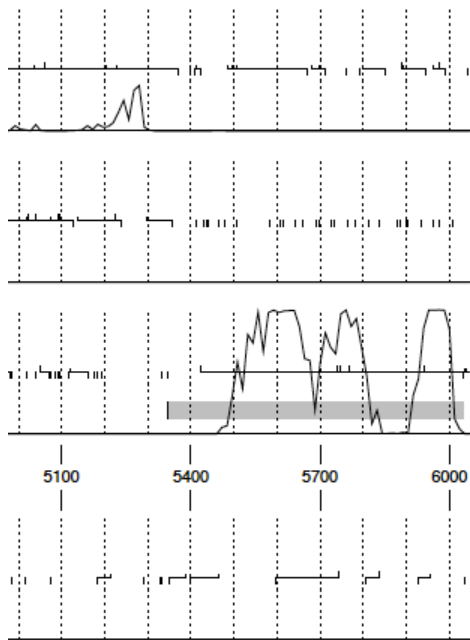
capsid (also called the procapsid) and allows the phage capsid to expand to its mature size during assembly and DNA packaging.

I will write the appropriate Notes in the Notes field, and Function field.

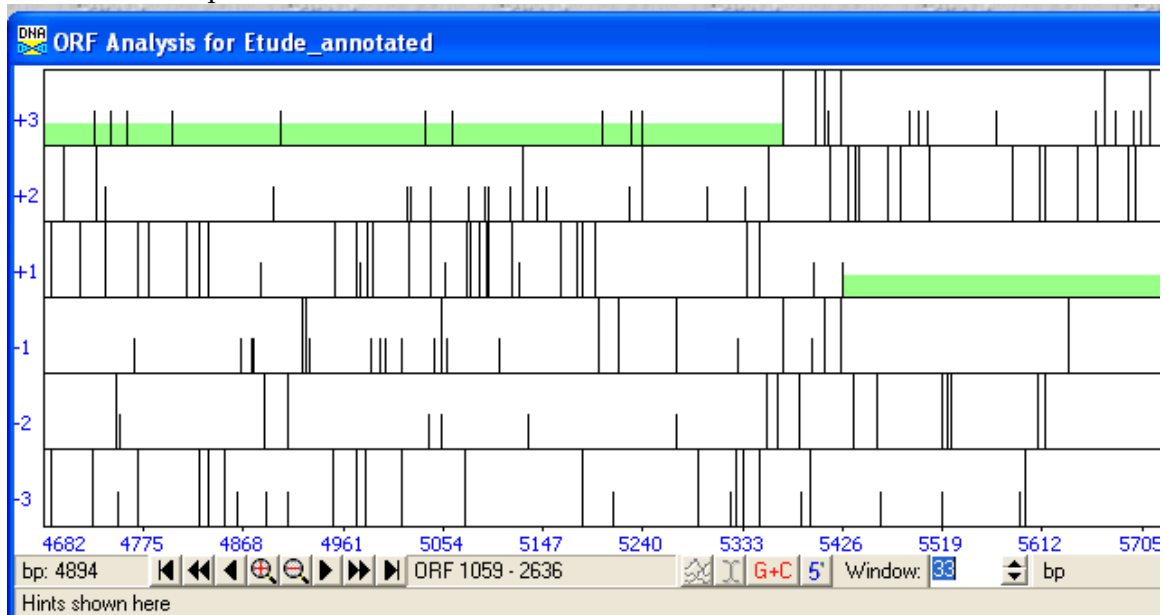
BLASTing the sequence at NCBI shows that we once again match LeBron gene 6's start exactly.

Gene 7:

Coding potential: The coding potential for this gene begins around 5450.



Gene 7 has two possible start choices:



The start that Glimmer and GeneMark have selected at 5427 and the TTG start at 5400. Both starts include all the coding potential. GeneMark never calls TTG starts and Glimmer undercalls them, so we must take that into consideration when deciding which start to pick (not a good time to say, "well, all things being equal, we will take the algorithms' call," because TTG starts are **NOT** equal from the point of view of the programs.)

When we look at the two starts in context of gap closing and SD scores,

#	Shine D Score	algarno Space	Sequence of the Region	Start Codon	Start Position	ORF Length
1	378	7	CGGGACACTCACCGCTTTTCAA	TTG	5400	639
2	462	7	TGCCTTAACTCAAGGAAAATTA	ATG	5427	612
3	294	7	GAACTGTCTAAGGCTGAGCCG	ATG	5742	297
4	435	9	GTCTAAGGCTCAGCCCATGGAG	ATG	5748	291
5	420	9	GCTTCCCGACCTACCTCAGCAG	ATG	5722	262

the TTG start at 5400 has a score of 378 and the ATG start has a SD score of 462. However, the size of the gap left between the stop codon of gene 6 is either 24bp or 54bp.

BLAST data: The BLAST tab shows that the longer gene start has been called for the genes already in GenBank (our alignment has a mismatch of 1:10).

Given that 24bp is already a sizable gap for a phage genome and the other GenBank phages use the longer start, we will pick the TTG start at 5400.

Change the gene start on the Description tab, and click the calculator button to recalculate gene length and to post the change to the database.

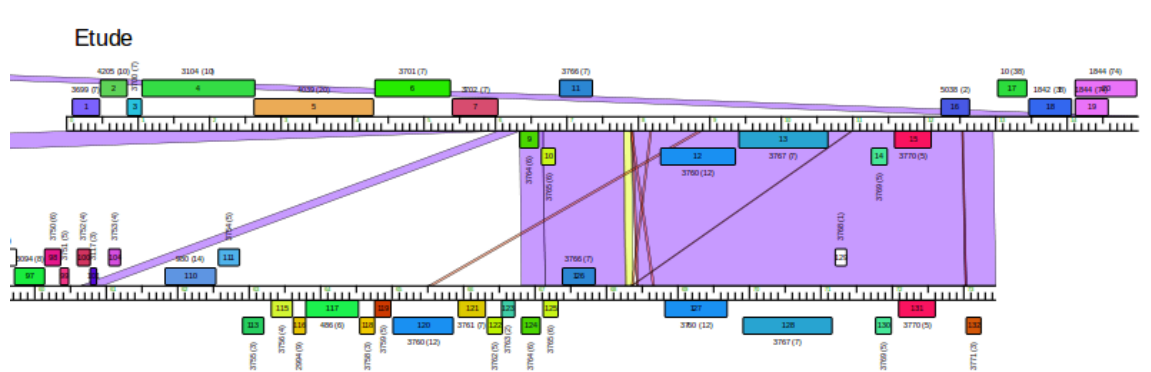
ReBLAST the gene through DNA Master to make sure that you see the correct alignment.

Functional assignment: the NCBI results indicated that this is a likely scaffolding protein in another phage. While this is not labeled as a scaffolding protein on the Hatfull Maps, synteny certainly supports this assignment. We will assign this gene the function of "Scaffolding", but won't enter it into the function field.

Gene 8:

Gene 8 is a very small gene (38 res) that exactly matches the beginning of LeBron gene 8 in the BLAST tab alignments. Normally, I would include this gene in an annotation, even though it is very small because it shows that a gene has been truncated and therefore is a good example of genome mosaicism and recombination. Unfortunately, gene 8 also overlaps a tRNA. Generally we do not see CDS and tRNA overlaps, except for possibly a few bases at the 3' end of both of them. The positioning of the tRNA, and the truncation of gene 8 suggests that the tRNA interrupted the original gene 8. I will therefore delete the called gene 8 from the auto-annotation. I am not going to renumber the genes again until I am done with the annotation.

Now we know from our phamerator alignment that Etude no longer has a high degree of similarity to LeBron after gene 8. But there are some lines in the phamerator map indicating that gene 9 of Etude is similar to something farther to the right in the LeBron genome. Slide the Etude map to the right in relation to LeBron and you will see:



LeBron is still the bottom genome, but you can't see the title any more because it is all the way over at the left end of the genome. From the map above, it looks like the next 8kbp are very similar to LeBron, with the final two kbp having no similarity.

Gene 9 the tRNA:

Click on the product tab:

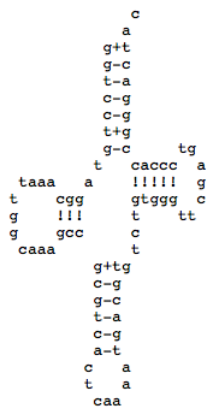
The screenshot shows the Etude_annotated software interface. The main window displays the tRNA structure for Aragorn v1.1. The structure is shown as a sequence of bases with stems and loops. The anticodon is TTT, and the product is 'Complement of TTG'. The 'Show Aragorn v1.1 Structure' checkbox is checked. The bottom status bar shows 'Position: 10975' and 'Controls >> Map'.

If you check the box at the top marked “Show Aragorn v1.1 Structure” the folded view of the tRNA will appear. Examine the top stem loop and 3’ end of the tRNA: does the stem loop have seven base pairs? No, it has eight. Is the 3’ end sequence after the stemloop NCCA? No, It is ngct. This means that we will need to look at the tRNA outputs from the other programs and trim the tRNA appropriately.

Web-Based Aragorn:

etude
 14998 nucleotides in sequence
 Mean G+C content = 60.2%

1.



tRNA-Leu(caa)
 75 bases, %GC = 56.0
 Sequence [6240,6314]

Primary sequence for tRNA-Leu(caa)
 1 . 10 . 20 . 30 . 40 . 50
 ggtcctgtaggcaaatggcaagccgctcactcaaaatgacgtgtctg
 tgaattcaatccccaccggaactac

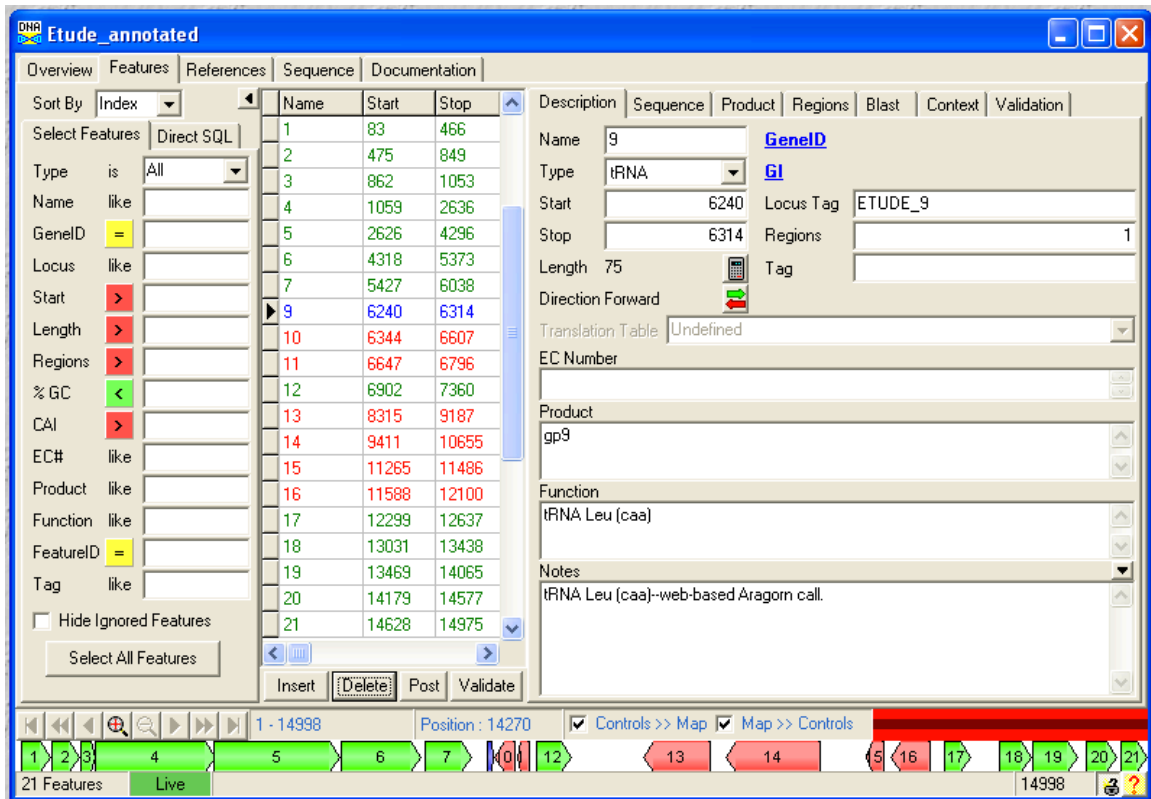
the output for Web-based Aragorn is much better: a seven base pair top stem loop, and on the 3' end there is a discriminator base (the A), followed by a single C from the CCA. This follows our tRNA rule: the trimming the CCA part of the sequence once it deviates from CCA.

The tRNAscan SE output is:

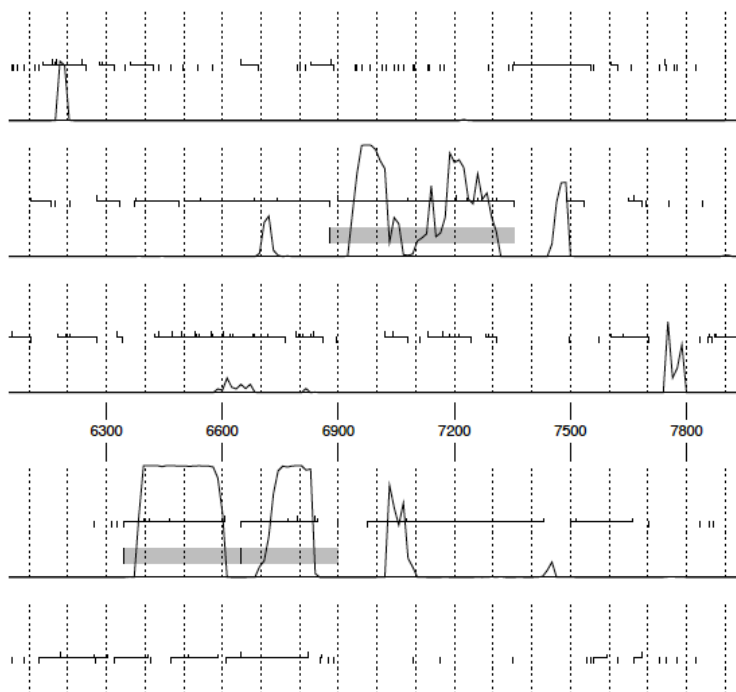
Results

Sequence Name	tRNA #	tRNA Begin	Bounds End	tRNA Type	Anti Codon	Intron Begin	Bounds End	Cove Score
Your-seq	1	6242	6315	Leu	CAA	0	0	62.03

[View tRNA](#)



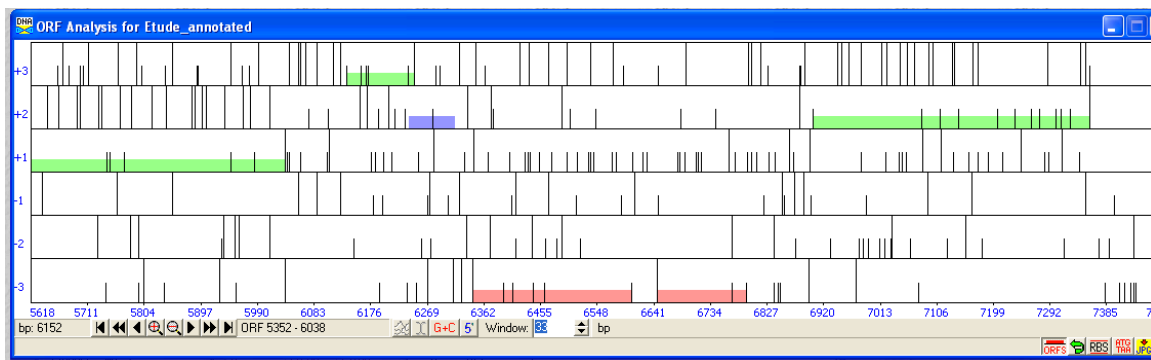
Gene 10: It seems pretty clear from GeneMark TB coding potential, from our phamerator alignment, and that the next gene is a reverse gene, rather than a forward gene like all the previous ones.



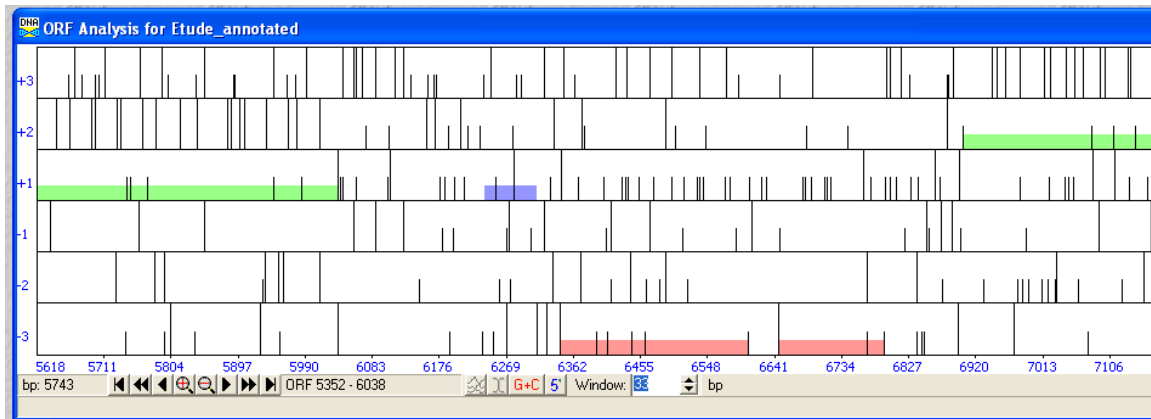
There are actually three separate ORFs in the fourth frame above, the first two corresponding to two different auto-annotated genes, with the final ORF overlapping

with the forward ORF in the second frame. Both algorithms decided to call the forward ORF as a gene over the third reverse ORF (you can't pick both; they overlap almost completely). The coding potential of the forward ORF is much better than the third reverse ORF—as shown by the higher, more extensive peaks, so we will also pick that gene when we get to gene 12.

Back to gene 10: when a forward gene and reverse gene meet stop codon to stop codon (or end of tRNA to stop codon as genes 9 and 10 do), it is not necessary to leave much space between the two ends of the genes. Several bases is sufficient. However in the opposite case, when the forward and reverse genes meet start codon to start codon, it is necessary to leave at least 50-60 bases between the two starts. This is because there will be a promoter for the RNA polymerase preceding the start codon of each gene. Since gene 10's stop codon has plenty of room after the tRNA transcript ends, we don't need to worry about this here (See below). We will need to worry about it when we select the starts for both gene 10 and gene 11.

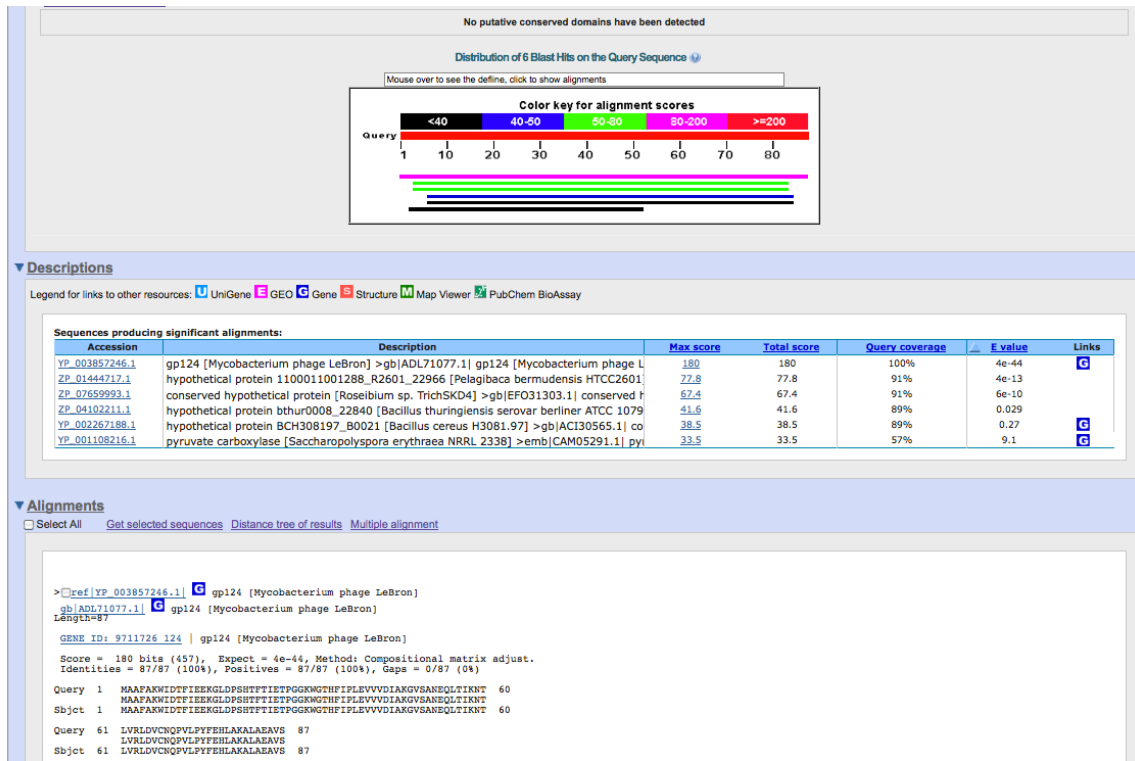


in the frames window, the deleted gene is still highlighted. Click the green “refresh” arrow at the lower right side (next to the ORFs button) to update the frames window.



Gene 9 only has one real choice for a start codon, and it was selected by both the GeneMark and Glimmer algorithms.

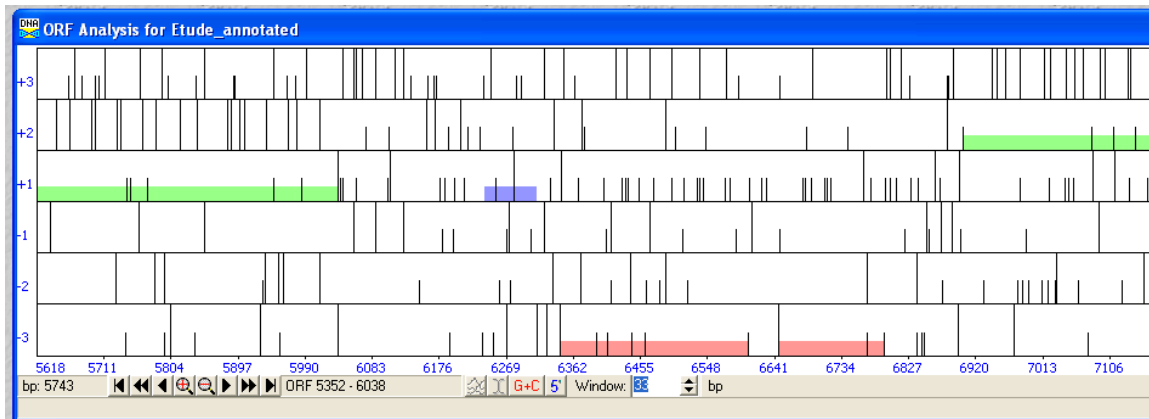
BLAST check:



We still are similar to LeBron, only now we are similar to gene 124. We still align perfectly with the same start codon as selected in the LeBron annotation.

In the annotation notes, when you are calculating the “gap/overlap” number, you should still look at the start codon of gene 10, only now you should compare to the stop codon of gene 11, because we are going in the reverse direction. The reason why we include these gap or overlap numbers is to see how well the start we chose fills out the genome from this gene to the neighboring one. Since we can’t change the stop codon, the only way to fill the genome is by changing the start codon. This number provides an extra reference as to how closely the genes are called in your annotation. LeBron gene 124 has no known function.

Genes 11 and 12: As mentioned above, gene 11 is a reverse gene while gene 12 is a forward gene. As they will be “head to head” (so to speak), we need to take care in choosing the starts for each of them that we leave at least 50-60 bases between the two genes.



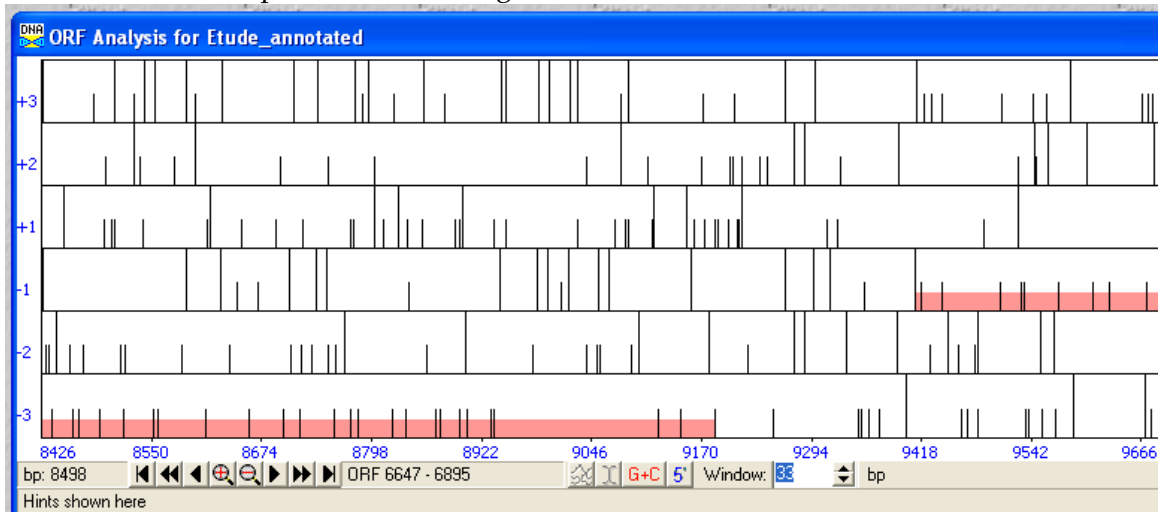
Currently, if we accept the two GeneMark calls for the genes, we will just barely squeak by with our minimum of 60 bases (6841 to 6902). Notice, however, in the above figure, that it is not possible to extend gene 12 to start any closer to gene 11 (there aren't any more start codons in the same frame any closer to gene 11 than the one already called). On the other hand, Gene 11 has four possible start codons, including the one used in the Glimmer call, which is way back at position 6796. From looking at the GeneMark TB coding potential trace, it is pretty clear that the Glimmer start does not encompass all the coding potential, so we will eliminate this choice as a possible start. We do still have three more starts to check: the one selected by GeneMark and the two immediately after it. If we check the RBS scores for all three starts; we find that the GeneMark start also has the highest RBS score. So we will accept the GeneMark call for gene 11. Gene 12 really only has one possibility for a start, and it is the one called by both programs above.

BLAST check: gene 11 aligns with LeBron gene 125 query 1 to sbjct 1. Gene 12 aligns with LeBron gene 126 query 22 to sbjct 22 (which still counts as picking the same start, just the beginnings of the genes are not as similar as the later portions.)

Gene 13: While there are some blips in the GeneMark TB coding potential, none of them align well within an open reading frame (if the peaks did fit better into an ORF I would be likely to include them as genes). So we will leave a fairly large gap between gene 12 and gene 13. This is also what the algorithms suggest.

Both the Glimmer and GeneMark calls suggest that gene 13 begins at position 9187, but this start leaves a huge gap between gene 14 and 13 (gene 13 is reverse, so we compare its start to the upstream gene). The BLAST data suggests that this gene is much shorter than the other similar entries in GenBank (UPIE 1:59).

There are five more possible starts for gene 13 between 9300 and 9400.



The 'Choose ORF start' dialog box shows the following information:

- Starts : 33
- Selected : 1
- ORF Start : 9400
- ORF Stop : 8315
- ORF Length : 1086
- 5' End : 50.0
- 3' End : 66.7
- Cdn1 : 75.0
- Cdn2 : 80.0
- Cdn3 : 100.0
- Length : 12

#	Shine D	algarno	Sequence of the Region	Start	Start	ORF
	Score	Space	Upstream of the Start	Codon	Position	Length
1	357	8	GCTAGCACCCCGTACCGAAGCG	GTG	9373	1059
2	525	8	TACCGAAGCGGTGCGGGCTCCT	TTG	9361	1047
3	420	8	GGTGGGGGTCCTTTGCTTTGC	GTG	9352	1038
4	143	6	GCGGGTCCCTTTGCTTTGCGTG	GTG	9349	1035
5	210	7	AGCCCCCGCTAACCGCGTCGGT	TTG	9253	939
6	483	7	ACACCACACCAGGAGGAACACC	ATG	9187	873
7	315	8	AACCGACACTGATATTCAGTAC	GTG	9148	834
8	357	8	GTTCCAAAGCTTCGCGCAATTC	CTG	9124	810
9	273	7	GGCCAGCGTCAAGGCTAAGGGC	ATG	8938	624
10	399	7	CAGCGTCAAGCGTAAGGCCATG	GTG	8935	621

The best RBS score goes to the TTG at 9361. We will pick this start.

BLAST check: while we now match UPIE, we are substantially longer than LeBron.(Query 59 aligns with Sbjct 1)—which is interesting. The two phages seem very similar according to phamerator (all that purple between the genomes), so why wouldn't the LeBron annotators have chosen the longer start that we did? While it is not necessary to match the starts for all genes 1 to 1 with a similar phage in genbank, I was still puzzled. To solve this, I actually loaded the LeBron sequence into the web-based GeneMark TB coding potential site, and examined the two outputs side by side. It turns out that LeBron has a point mutation in this area which causes an extra stop codon in this frame, and the start that we chose for Etude is not a possibility in LeBron. The point mutation is not a large enough sequence difference to show up as another color in phamerator. I am happy to proceed with our start selection.

Gene 14: Is another reverse gene, this time in the fifth tier in the GeneMark TB coding potential view. We will ignore the peak in the forward third tier because we can't call

both of them and the reverse one is larger and lovelier. There is only one possible start codon for gene 14.

BLAST Check: we match LeBron gene 128, with the start codons aligning perfectly.

Gene 15: while there is a teensy little peak in the forward direction in the second tier GeneMark TB coding potential trace nicely centered in an ORF, I am more inclined to omit it for simply being too small. It is also important to get a feel for you phage genome—are all the genes very tightly packed? Do multiple genes have very small blips of coding potential? You need to think about these things when you are making your decision. There is also a trend in most phage genomes for clusters of genes to be transcribed in the same direction, and Etude is no exception. And since we are going from a reverse gene 14 to another reverse gene --supported by the algorithms calls-- we will skip the teeny blip in the second tier.

However, the best way to really be sure would be to add this gene into your annotation, and BLAST the protein sequence to see if there are any similar genes in GenBank.

Gene 16 is the larger peak in the fourth tier after the smaller peak in the fifth tier, then gene 17 is forward again in the second tier(the peak in the top tier does not align well with an ORF).

Back to Gene 15:

There are really only two choices for a start for gene 15; either the GeneMark call or the Glimmer call. The GeneMark call has a higher RBS score so we will pick the GeneMark call—both for SD score and for being a longer gene.

When we do the BLAST check, we match LeBron gene 130 perfectly at the start.

At this point, it is worth revisiting whether or not we want to include the little forward ORF with the blip of coding potential to see if it aligns with LeBron gene 129. Our gene 15 matches LeBron 128, and our gene 16 matches LeBron 130. Our phamerator alignment indicates that there is still the highest level of sequence similarity between these genomes in this region, but again, a point mutation resulting in the loss of a start or stop codon would not be enough to change the nucleotide identity color in phamerator. And again, it is not necessary to make the genome annotations match, but it is worth looking at the data. When in doubt, BLAST the potential gene to see if there are any matches in GenBank. I will leave it as a exercise to see if LeBron gene 129 exists intact in Etude, and if so, should it be included in the annotation.

Gene 16:

Gene 15 has a stop codon almost immediately to the right the start codon called by Glimmer and Genemark (remember, we are in the reverse direction, so the gene is transcribed right to left). This means that the start codon selected by the two algorithms is already the longest possible start that can be selected for this gene.

The SD score is nice and high, and the start encompasses all the GeneMark TB coding potential. I am happy to accept the call as is.

BLAST check: we match gene 131 from LeBron with a perfect start codon alignment.

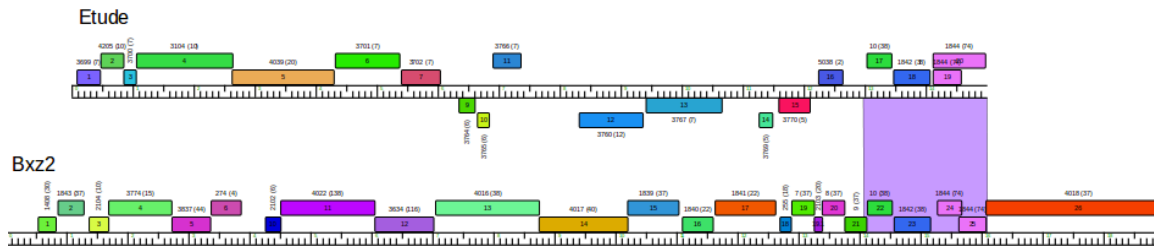
Gene 17: Gene 17 is the peak shown in the fourth tier of the GeneMark TB coding potential. Since we are once again switching from reverse to forward, we need to make sure that we leave at least 60 bases between the two gene calls.

There are three possible starts for gene 17 before a stop codon appears in the frame, including the Glimmer start:

All three start encompass all the coding potential shown in the GeneMark TB trace; however, the SD score is best for the second of the three starts; plus this gene call fills our genome gap a bit better without getting in the way of promoter sites. We will pick the second start, at 12242.

BLAST check: this gene aligns with UPIE, but not with LeBron. Even though phamerator shows that this sequence is still similar to LeBron, and the LeBron annotation does not call this gene, we are going to rely on our own data and Glimmer calls. It is possible that LeBron has another point mutation in this region, eliminating this ORF. We can check on the LeBron genome's coding potential in the GeneMark program again, if we want to.

Genes 18-21 The remainder of the genes are no longer similar to LeBron, but instead appear to match the A3 cluster phages.



Gene 18: There is a fairly big (400bp) gap between the end of gene 17 and the beginning of the Glimmer/ GeneMark calls for gene 18. While unusual in phage genomes, this is OK. There is no coding potential blips anywhere in the GeneMark TB coding potential traces between the two genes, and while there is an earlier alternate start codon at 12827, its RBS score is low, while the RBS score for the Glimmer/ GeneMark call is very high! So we will pick the Glimmer/ GeneMark call.

BLAST check: when we BLAST the amino acid sequence using BLASTp, we find that we align perfectly with Bxz2 gene 22, with the Query 1 aligning with Sbjct 1.

Gene 19: We know from the GeneMark TB coding trace that gene 18 and gene 19 are in the same frame, so the stop codon of 18 precludes any start codon for gene 19 earlier in the genome. The start codon selected by Glimmer and GeneMark is already the longest possible gene call for the gene, encompasses all the coding potential, and has a RBS good score. We will accept the algorithms' calls.

BLAST check:

We match the Bxz2 gene 23 perfectly, with a Query 1 aligning to Sbjct 1.

Functional assignment: This is the major tail subunit of the phage.

Gene 20: There are two possible starts to Gene 20, the one called by Glimmer and GeneMark, and the one upstream at 14116 (another TTG). If you check the coding potential GeneMark TB output, you will see that the Glimmer and GeneMark calls do

not encompass all the coding potential. When we check the SD scores, we get a higher score for the TTG start than the Glimmer/GeneMark start. We will pick the 14116 start.

BLAST check: This gene matches Microwolf gene 25 perfectly (aligns Query 1 to Sbjct 1).

Functional assignment: this is the first of the two tail assembly chaperones (the equivalent of G in phage lambda).

Gene 21: The GeneMark call for this gene overlaps with the end of gene 20, while the Glimmer call leaves a large gap. The GeneMark call encompasses all the coding potential while the Glimmer call does not. This is quite a large overlap.

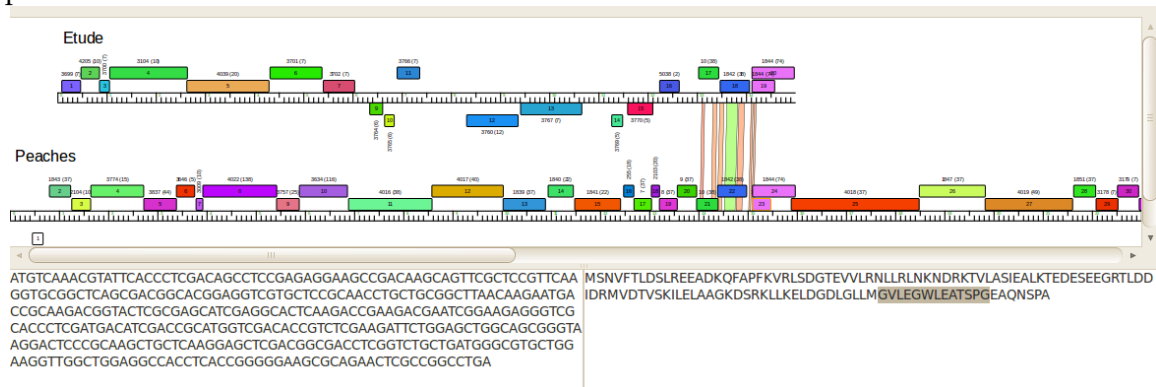
BLAST check: This gene matches Bxz2 gene 24 perfectly, but begins in the middle of the equivalent gene for Peaches and Eagle. This is because of its function: this is the second of the two tail assembly chaperones, and actually begins at 14116 and then frameshifts into the remainder of this gene, creating the G-T fusion.

Note about this function:

Bxz2 gene 23 is a tail assembly chaperone, (gene "G" in phage lambda), and with its partner gene 24 (gene "T" in lambda) makes a fusion protein that helps assembly the tail of the phage correctly. Both the first protein, the "G" like protein, and the fusion "G-T" like protein are produced, however, the second gene product, the T-like protein is not made on its own but only as part of the fusion. In the flexible tailed phages, the tail assembly chaperones frequently precede the tapemeasure gene (generally the longest gene in the genome), and are characterized by a "slippery sequence" that allows the ribosome to shift translational frame during protein synthesis. The ribosome will "slip" back a base, causing a -1 frameshift. Another way to think about it is that the ribosome reads the same base twice. The slippery sequence is generally rich in As but can begin with Gs as well. There are numerous examples of the G-T frameshift in phamerator (look for any phage genome that has two genes that begin at the same start codon, with one of them being longer than the other and followed by the longest gene in the genome). Notice I put it in the Etude annotation in phamerator already. In the phages that we have studied, the G-T slippery sequence almost always occurs at the C-terminal end of the G protein (in our case Etude gp 20).

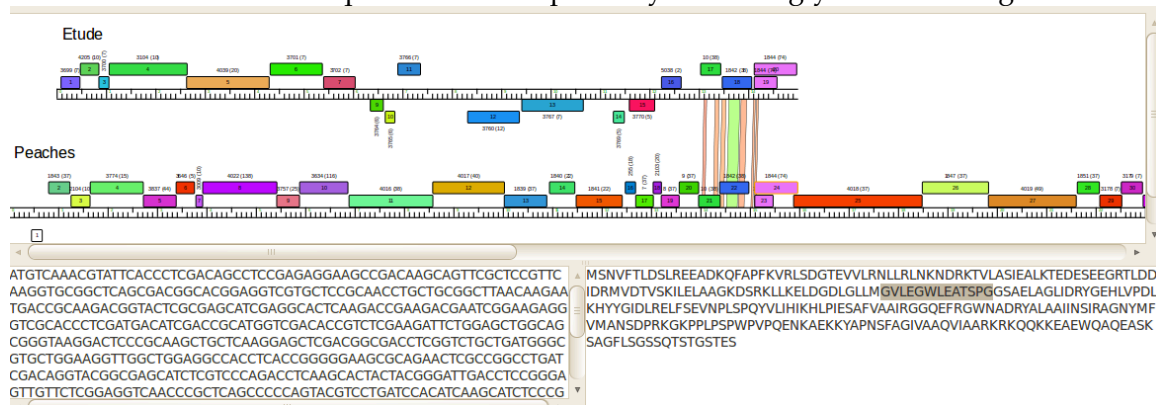
To find the coordinate, the easiest way is to find a closely related phage that already has the frameshift correctly annotated. If we look in our BLAST hits, you will see that phage Peaches is similar to Etude, and has the frameshift already correctly annotated in phamerator. While Peaches and Etude do not have a ton of nucleotide sequence similarity between them, notice the tail assembly proteins are in the same

pham.

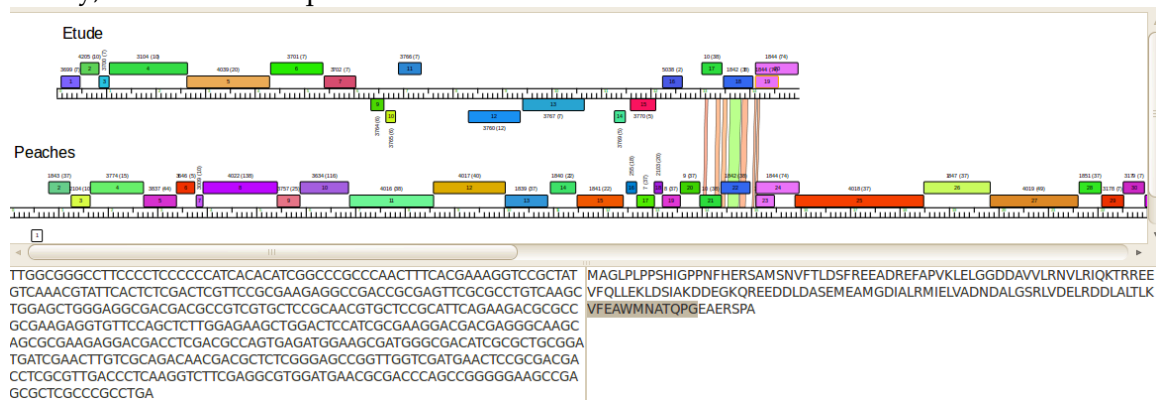


In the picture above, I have clicked on the first, shorter of the two purple Peaches genes (the G equivalent), and then highlighted the C-terminal portion of the protein sequence where frameshifting is likely to occur.

In this next picture, I have clicked on the longer, correctly annotated frameshifted G-T fusion protein, and highlighted the same amino acids as in the shorter previous gene. We can see that the the sequence matches perfectly until the glycine following SPG.

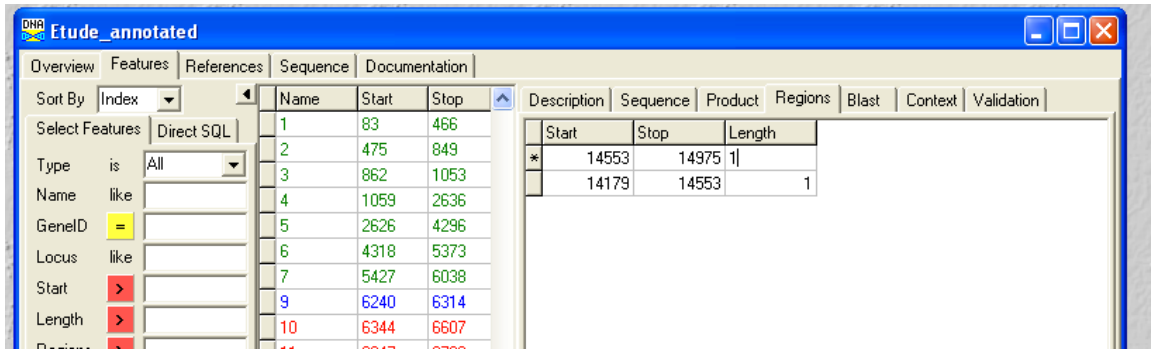


Finally, I look for the equiavalent area in Etude:

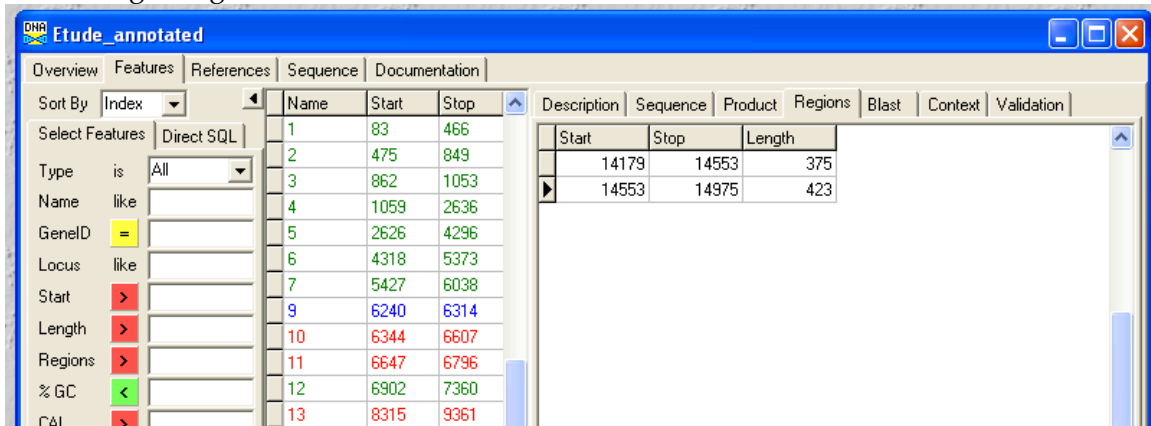


Again, above I have clicked on the first of two tail assembly genes, and then looked at the C-terminal end where frameshifting is likely to occur. Now, we go back into our six-frame translation of the sequence.

coordinates of the second region (it will first appear on top, but will later be correctly reordered):

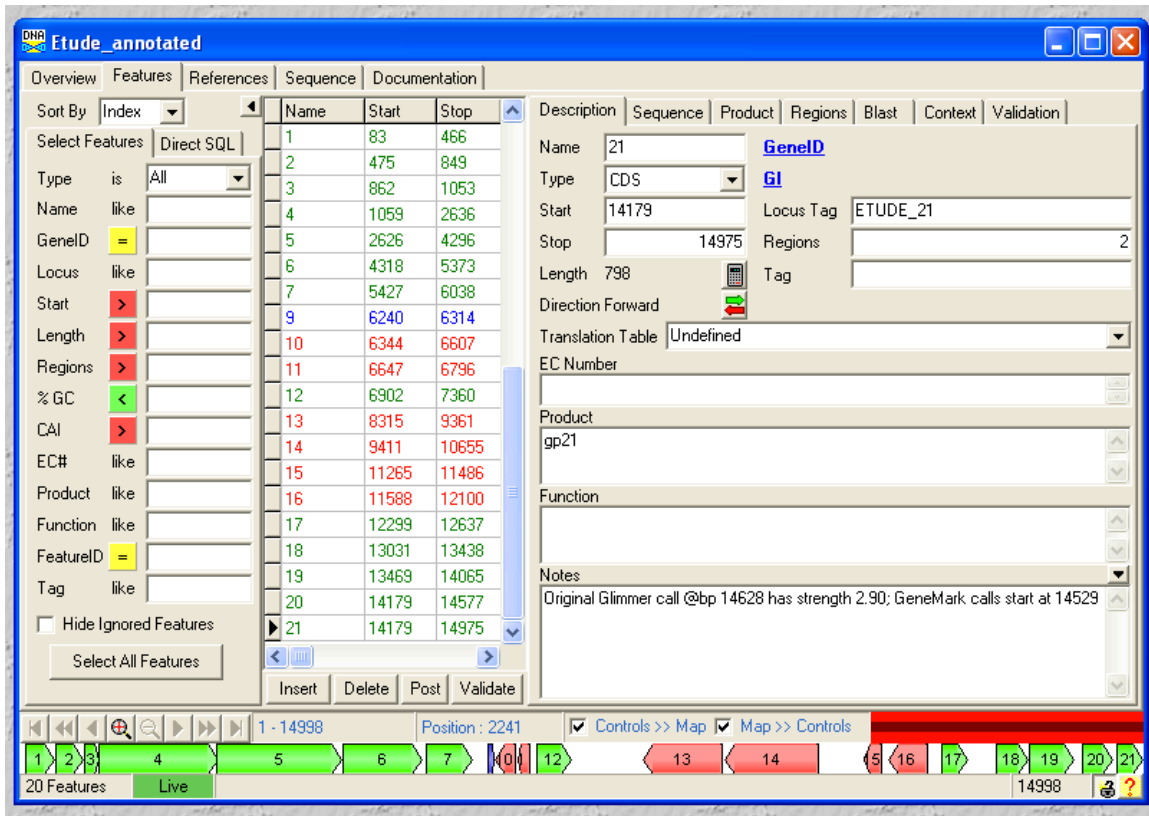


click "assign lengths"



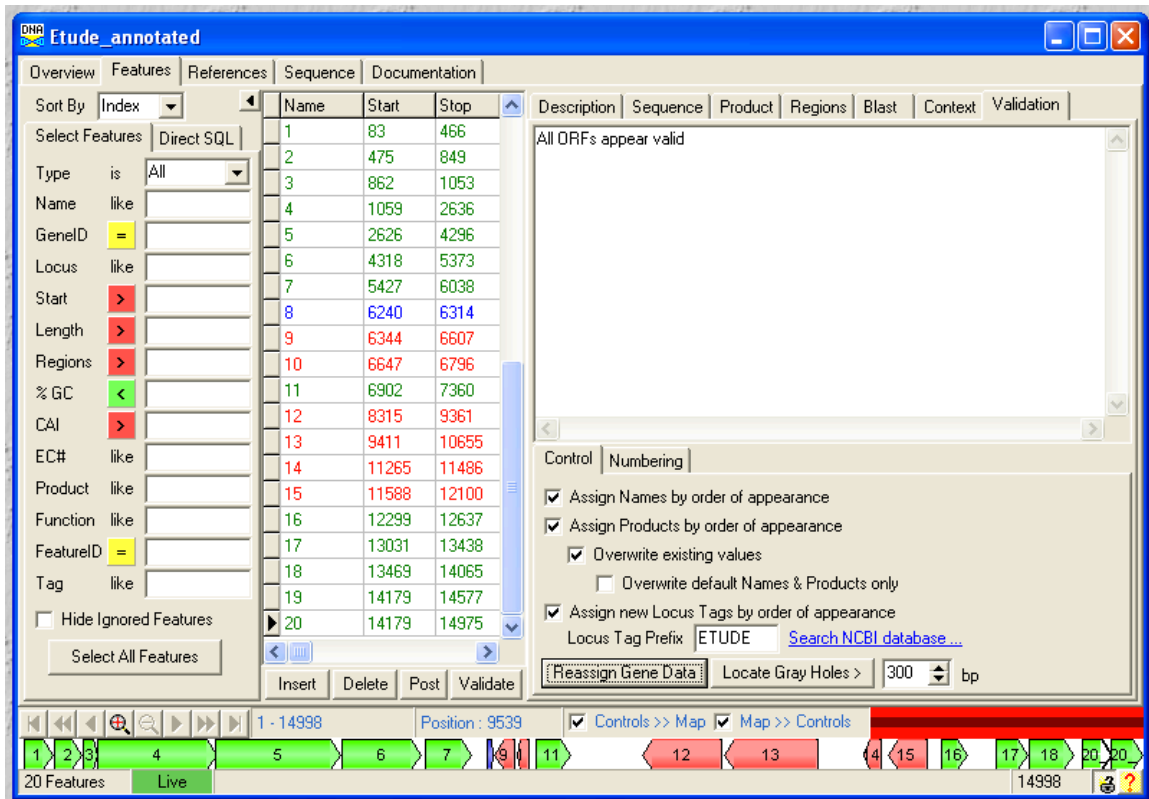
The correct numbers will be calculated.

Now return to the description tab, and adjust the start coordinate accordingly.



Enter your gene Notes.

Finally, validate and renumber all your genes.



If you have not been reBLASTing all your genes, now is a good time to delete all the BLAST hits from your file and do a new complete genome BLAST. Then start your QC.

-Review your notes (correct format? All the information?)

-check those gene gaps and overlaps one last time. Did you miss any?

Finally save your final file (yourphagename_final.dnam5 is a good name), and send it off to Pitt for review.