

Installation instructions for a local copy of the mammot chip annotator.

The mammot annotator was written on a PC platform running Fedora Linux Core 2. Although everything should work in other linux distributions and OS-X please bear this in mind when installing.

Prerequisites

MySQL: Version 3.23 and over.

PHP: 4.3.4. These scripts were originally written with 'register_globals' set to **on**. This variable is set in the php.ini file. Please refer to the documentation of your OS release to where this file is kept (/etc/php.ini in Fedora Core 2). They should work with globals off (the default PHP setting for releases later than 4.3), but if you have any problems, especially with the viewer, please set to on.

Perl v5.6.0 or over

Apache 1.3 or over. The site was written with Apache 2.0 but as it doesn't do anything out of the ordinary 1.3 should be fine. Please make sure however that the PHP modules are compiled – most modern distributions should do this anyway.

Installation instructions:

NOTE: All files must be gunzip'd and untar'd before use.

These instructions should work fine on most modern Linux installations.

Gunzip and untar the mammot_scripts file into your scripts directory (or wherever you put perl scripts). The scripts should be executable, but if not set them by chmod +x. Please set your PATH environment variable to include the scripts directory if you want to run them from anywhere. Instructions on how to do this are shown below.

Installing Primer:

For the design of the primers the tiling script uses Primer3 (written by Helen J. Skaletsky and Steve Rozen). Instructions for downloading, compiling and installing Primer3 can be found at http://frodo.wi.mit.edu/primer3/primer3_code.html. In order for the tiling script to work properly, you will need to add the PATH to Primer3 in the .bashrc script located in your home directory (eg add the line PATH="\$HOME/primer3/primer3_0_6/src", or add it to your existing PATH).

Installing BLAST

BLAST source code and executables, as well as full installation instructions can be found on the NCBI website at <http://www.ncbi.nlm.nih.gov>. You will need BLAST if you want to do alignments on your tiles and primers, otherwise this isn't needed by the software.

Installing the chip web site:

Installing the web site is relatively straightforward once you have a working copy of the Apache web server and associated PHP modules up and running. To install the site, simply create a directory called `chip` in your Apache-defined `html` folder (in Fedora Core 2 this is `/var/www/html` by default) and `gunzip` and `untar` the `mammot_chip_site.tar.gz` file into it.

To store the BED files produced by the experiment exporter you need to create a directory called `chip_files` within the `chip` folder and give it full write access by `chmod 0777`. You may also need for apache to own it by using `chown apache:apache`.

Creating the MySQL database:

This presumes that a MySQL database is up and running.

1. Run the database using `mysql -u root -p` and enter the password when prompted.
2. Create a database called `chip`: `create database chip`
3. Give permissions to `chip_user`: `grant select, delete, insert, update to chip_user@localhost`
4. The database should now be set up!

Now to add the database structure, log out of MySQL and use the command `mysql -u root -p chip < chip_data_structure.sql`.

Populating the database:

For the chip description and experimental data, the chip website provides a friendly front end to make things easier. There are other datasets available however (ie those describing genome structure that the chip and viewer need) that have to be uploaded to the database manually at this time.

To upload data log into MySQL, change the database to `chip` (`use chip`) and use the command `load data local infile "/path_to_file/filename" into table table`;

e.g. `load data local infile "/home/ed/species.out" into table species`;

Species and genomes data:

The tables `species` and `g_release` describe the species used in the dataset and the genome releases. For convenience data for fly, mouse and human data for the latest releases are available in the file **species_genomes.tar.gz**. The tables are very simple and new data can be added manually if needed (please refer to www.mysql.com and the `insert` command).

Genome structure:

There are three tables describing genome structure: `gene_structure`, `mammot_repeats` and `genome_go`. Sample files are currently available for *Drosophila*

(r3.1), mouse (m33), human (NCBI35) and dog (BROAD1). The data files are reformatted from data available from Ensembl (<http://www.ensembl.org/>) and the UCSC Genome Bioinformatics Site (<http://genome.ucsc.edu/index.html>) and therefore new releases of data (eg *D. melanogaster* 4.0) are really dependant on those sites. Some of the files are very large (eg the repeats file) and may take a long time to upload.

Upload the datafiles into their respective database tables using the load data command shown above.

Creating a genome tile:

Mammot_tile.pl:

Mammot_tile takes a fasta formatted file and creates a minimal overlap genome tile based on various user settings. As well as an interactive setup, a separate input file can also be used. When designing tiles in regions of high repeats it may desirable to miss these areas out; to this end mammot_tile can work on fasta files that have been screened with RepeatMasker to mask out repeats with N's. The program will split the sequence into separate jobs based on the N content of a particular stretch of DNA. Other scripts can turn this back into unmasked sequence (or more usefully, sequence that has been masked to lowercase) when the tile design phase has finished for easier downstream analysis.

After one round of primer design, there are likely to be areas where Primer3 could not design any good primers. To help plug the gaps, two further rounds are performed using increasingly more 'liberal' conditions- you can tell which round the primer was designed in by the presence of a suffix (either -2 or -3).

The tiling script was mainly designed to make small – medium tiling paths for use in ChIP on chip experiments – it's not really cut out for doing entire genomes (although there's no reason why you can't use it for this) and I'm sure better software exists for this purpose.

A quick note about file formats: all of the scripts here require a file with linux carriage returns, otherwise the script may act unpredictably or not at all. If you have edited files in Windows or MacOS, please run them through the `newline_converter.pl` script first to be on the safe side.

Run the script with the command `mammot_tile.pl`. If you choose interactive setup a number of options will appear.

Enter Sequence File Name:: name of the fasta file to design the tile on.

Enter Project Name:: name of the project (e.g. OMFG_PROJECT)

Enter minimum primer Tm:: needed for Primer3.

Enter Optimal Tile Size:: size you want your tiles to be in bp (e.g. 1000)

Enter Minimum Tile Size:: minimum size of a tile that is acceptable (e.g. 500)

Enter Tile Overlap:: maximum size in bp of how much the tiles can overlap (e.g. 30 – I wouldn't recommend going much lower than that unless you want a lot of gaps in your tiling sequence).

Enter N threshold:: Number of N's in a row before the script separates the sequence into 2 separate jobs and removes repeat from the design process. For example, a value of 150 will keep small repeats in the main tile but large repeats will be cut out.

Enter chr:: Chromosome used to create the tile (e.g. 2L)

Enter genome start position:: Base position of the tile on the genome (e.g. 3519537)

Two example files are included (`dros_test.seq` and `settings.in`) to help you test that everything is working fine.

After the script has run (time to completion will depend on how fast your computer is and how long the tiling path is) there should be a large number of files produced. Most of these are used by the script at various times or used for debugging. The one that contains the information you actually want is called **projectname_chip_import.out** (where projectname is the name of the project you specified when running the script) – it is this file which is used in the chip creation part of the web site.

mammot_shuffle.pl and plate_layout.pl

When designing arrays, it is best if the PCR products are placed randomly on the slide to help avoid any biasing during the analysis. To this end `mammot_shuffle.pl` will randomise the list of tiles and subsequently `plate_layout.pl` will sort the list into well positions, primer names and primer sequences to simplify ordering oligos from various companies.

finalout_to_seq_files.pl, mammot_blast.pl and parse_blast.pl

These scripts require a working local BLASTn server and an associated sequence database to function. The script `finalout_to_seq_files.pl` produces fasta format sequence files which `mammot_blast.pl` then aligns to the specified genome sequence, and `parse_blast.pl` collates the information into a file suitable for uploading into the mysql database.

convert-to-unmasked-sequence.pl

If you've designed a tiling path using RepeatMasker N-masked sequence and wish to know what the N's actually are you can use this script to restore the sequence back to its original state or more usefully, turn the N's into lowercase masked sequence.

Now that you've (hopefully) designed a tiling path to a region of interest it's time to upload the data into the chip web site so you can start doing something useful with it.

Using the website to create and manage chips and experiments:

Access the website by going to your web browser (I recommend FireFox) and typing <http://localhost/chip/index.php> in the address bar. The main page should come up;

Adding and managing chips:

Choose *Chip Admin* from the main menu. To create a new chip click on *Add a new chip* and follow the instructions for adding the species and genome release. When you get to step 3 add the name of the chip and supplementary information (the start, stop and chr information is only used for the BED file export and is safe to leave blank if you want). When it comes to pasting in the chip information, open up the `project_chip_import.out` file and copy and paste the contents into the text box. Clicking on *add data to database* should complete the upload to `chip_details` and give a 'data added successfully' message. If it doesn't please check that you have entered everything correctly. The site also has options to append an existing chip (for example if you want to combine a few tiling paths on to one chip for arraying) and deleting a chip.

For annotating the tiling path you can enter PCR amplification results using *Add PCR results* page. The gel mock-up used follows the system of the ABgene 96-well agarose gel used in our lab. For different variations and gel systems you'll have to find a friendly HTML/PHP coder, or you can make a tab-delimited output file of results and upload them into `mammot_gels` manually (the table structure is given in the appendix).

Entering and viewing experiments:

Choose *Enter Experiments* from the main menu and follow the onscreen instructions. Please make the experiment description something that will make sense a few weeks later. You can copy and paste spot and value data directly into the text box from a spreadsheet of results. The data should have the format *spot_name score*. Clicking on *add data to database* should complete the upload to `exp_desc` and `exp_details` and give a 'data added successfully' message.

You can view your results, and combine them with previous results from other experiments by clicking on *View Results* and choosing which experiments you want. Clicking on *Export Experiments* will create a file in BED format and provide you a link to that file (which you can view again at a later date by clicking on *Previous Results* from the main menu) which you can then import into other applications for viewing (e.g. the UCSC Genome Browser at <http://genome.ucsc.edu/index.html> or the Ensembl Browser at <http://www.ensembl.org/>).

As well as exporting results, you can also view them using the internal mammot viewer. Although this does not have the wealth of supplementary information available from the dedicated genome browsers, it does have the distinct advantage of being tailored exactly to viewing the annotation of genome tiles and ChIP experiments.

The mammot viewer:

Clicking on *Mammoth Viewer* should open the application in a new window.

The screenshot displays the Mammoth Viewer interface, divided into three main sections:

- Top pane:** Contains search parameters including 'Enter search range' (114800 to 235000 bp), 'Project' (as-c), 'Chr' (X), and 'CG Graph' (unchecked). It also features a 'Select Experiments' dropdown menu with options like 'exp001 - as-c' and 'exp_002 - as-c'.
- Main pane:** The central visualization area showing a genomic map. It includes:
 - Navigation controls:** Arrows for navigating between tiles.
 - Tiling path:** A series of green squares representing individual tiling paths, labeled with IDs like 'as-c.001' through 'as-c.117'.
 - Feature track:** A horizontal line with markers representing genomic features.
 - Genes:** Labels for genes such as 'pc1' and 'CspH1'.
 - Experiment data:** A red bar chart at the bottom of the main pane representing signal intensity for different experiments.
- Information pane:** Located at the bottom, it provides 'PCR product details' for a selected tile (as-c.054), including 'From' (171440), 'To' (172494), 'R Primer' (CCCCAATGCTTCCTGTTC), and 'Tm' (%GC).

The top pane:

This close-up view of the top pane highlights the search and selection controls:

- Search Range:** '114800 to 235000 bp'.
- Project:** 'as-c'.
- Chromosome:** 'Chr X'.
- CG Graph:** A checkbox that is currently unchecked.
- Select Experiments:** A dropdown menu showing 'exp001 - as-c' as the selected experiment.

The top pane contains all the parameters for the search. Enter a scaffold range or tile name, the project and the chromosome to search. Selecting 'CG Graph' will display a trace of CG content of the tiles at the bottom of the main pane. You can also select experimental data to view: ctrl click to choose multiple experiments.

The main pane:

The main pane contains an overview of the tiling pathway in the specified range, and any genes and features that may be there. It also displays any experimental data in a bar chart

format- multiple experiments are shown next to each other and can be distinguished using the key. Clicking on a tile will display information about it in the information pane.

The information pane:

All	PCR	Blastn	Repeats	Sequence	Array
PCR product details					
Name	test.gap.458a	% Repeats	83		
From	105128452	To	105128742	PCR Product	291
F Primer	CGACACCAAAAAGACACC	R Primer	TTCCTCTTCTGTACTTATCCA		

The information pane displays various data about the chosen tile. Click on the tabs on the top to choose the type of data. Please note if you haven't uploaded information like blast results to the database it won't appear in the viewer. On the sequence tab, repeat information (if displayed in lowercase) will appear in red for easier identification.

Appendix – MySQL tables:

```
mysql> desc chip ;
```

Field	Type	Null	Key	Default	Extra
chip_id	int(3)	YES	MUL	NULL	
chip_name	varchar(128)	YES		NULL	
g_id	int(3)	YES		NULL	
g_start	int(10)	YES		NULL	
g_stop	int(10)	YES		NULL	
chr	varchar(4)	YES		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> desc chip_details ;
```

Field	Type	Null	Key	Default	Extra
chip_id	int(3)	YES	MUL	NULL	
spot_name	varchar(64)	YES	MUL	NULL	
chr	varchar(8)	YES	MUL	NULL	
start	int(10)	YES		NULL	
stop	int(10)	YES		NULL	
5_primer	varchar(128)	YES		NULL	
3_primer	varchar(128)	YES		NULL	
sequence	text	YES		NULL	
notes	text	YES		NULL	
f_tm	float	YES		NULL	
f_cg	float	YES		NULL	
r_tm	float	YES		NULL	
r_cg	float	YES		NULL	
percent_n	int(3)	YES		NULL	
percent_cg	float	YES		NULL	
pcr_size	int(6)	YES		NULL	

```
16 rows in set (0.00 sec)
```

```
mysql> desc exp_desc;
```

Field	Type	Null	Key	Default	Extra
id	int(9)		PRI	NULL	auto_increment
user	varchar(128)	YES		NULL	
exp_name	varchar(128)	YES	MUL	NULL	
chip_id	int(3)	YES	MUL	NULL	
date	date	YES		NULL	
notes	text	YES		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> desc exp_details;
```

Field	Type	Null	Key	Default	Extra
exp_name	varchar(128)	YES	MUL	NULL	
chip_id	int(3)	YES	MUL	NULL	
spot_name	varchar(64)	YES	MUL	NULL	
score	float	YES		NULL	
stdev	float	YES		NULL	
signif	float	YES		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> desc g_release ;
```

Field	Type	Null	Key	Default	Extra
g_id	int(3)		PRI	NULL	auto_increment
species_id	int(3)	YES		NULL	
g_name	varchar(128)	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> desc gene_structure ;
```

Field	Type	Null	Key	Default	Extra
project	varchar(128)	YES	MUL	NULL	
gene_id	varchar(64)	YES	MUL	NULL	
transcript_id	varchar(25)	YES	MUL	NULL	
feature	varchar(10)	YES	MUL	NULL	
start	int(9)	YES		NULL	
stop	int(9)	YES		NULL	
strand	int(1)	YES	MUL	NULL	
chr	char(2)	YES	MUL	NULL	
gene_name	varchar(128)	YES	MUL	NULL	

```
9 rows in set (0.00 sec)
```

```
mysql> desc genome_go ;
```

Field	Type	Null	Key	Default	Extra
project	varchar(128)	YES	MUL	NULL	
gene_id	varchar(64)	YES	MUL	NULL	
gene_name	varchar(64)	YES	MUL	NULL	
go_id	varchar(32)	YES		NULL	
go_desc	text	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> desc mammot_blast ;
```

Field	Type	Null	Key	Default	Extra
project	char(64)	YES	MUL	NULL	
name	char(64)	YES	MUL	NULL	
name_type	char(64)	YES		NULL	
clone_hit	char(100)	YES		NULL	
identity	float	YES		NULL	
length	int(15)	YES		NULL	
mismatches	int(10)	YES		NULL	
gap_openings	int(10)	YES		NULL	
q_start	int(15)	YES		NULL	
q_end	int(15)	YES		NULL	
s_start	int(15)	YES		NULL	
s_end	int(15)	YES		NULL	
e_value	float	YES		NULL	
score	float	YES		NULL	
chr_details	char(255)	YES		NULL	

```
15 rows in set (0.00 sec)
```

```
mysql> desc mammot_gels ;
```

Field	Type	Null	Key	Default	Extra
project	int(4)	YES	MUL	NULL	
plate_id	char(255)	YES		NULL	
name	char(255)	YES		NULL	
well_pos	char(3)	YES		NULL	
pcr_worked	int(1)	YES		NULL	
conditions	char(64)	YES		NULL	
anneal	int(3)	YES		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> desc mammot_repeats ;
```

Field	Type	Null	Key	Default	Extra
project	char(64)	YES	MUL	NULL	
sw_score	int(5)	YES		NULL	
start	int(8)	YES		NULL	
stop	int(8)	YES		NULL	
repeat_match	char(20)	YES		NULL	
repeat_class	char(20)	YES		NULL	
rep_start	int(5)	YES		NULL	
rep_stop	int(5)	YES		NULL	
chr	char(3)	YES	MUL	NULL	

```
9 rows in set (0.00 sec)
```

```
mysql> desc species ;
```

Field	Type	Null	Key	Default	Extra
species_id	int(3)		PRI	NULL	auto_increment
species_name	varchar(128)	YES		NULL	

2 rows in set (0.00 sec)

```
mysql> desc urls ;
```

Field	Type	Null	Key	Default	Extra
url	varchar(255)	YES		NULL	
experiments	text	YES		NULL	

2 rows in set (0.00 sec)