

# Unix, Perl and BioPerl

## Session 3: Sequence analysis with Perl (modules including BioPerl)

### Exercise 3: Setting up a web-based search script

Goal: Learn the basics of using the CGI module to set up a script that can be executed from a web page on hebrides.

See <http://jura.wi.mit.edu/bio/education/bioinfo-mini/unix-perl/> for course page.

#### To do:

- If you haven't already done so, make a directory called perl\_2 and copy all files from /home/george/perl\_2  
Starting script = george.cgi  
Web page = george.html
- IMPORTANT 1: Rename the files according to your username:  
username.html and username.cgi
- IMPORTANT 2: make sure that the permissions of username.html is set to 644 (so everyone can read it) and of username.cgi is set to 755 (so everyone can read it, execute it, but not write to it).
- Open (but no need to copy) the data file at /home/george/yeast/registry.genenames.tab.txt  
Note that the yeast data contains tab-delimited fields for ~5000 yeast genes.
- Look at the HTML file, especially the line  
<FORM METHOD="POST" ACTION="/cgi-bin/class/george.cgi">  
that shows the script to execute. Change the filename to match your CGI file.
- Look at the script. It uses the CGI (Common Gateway Interface) module and Carp, a sub-module of CGI to help handle errors during debugging. Without Carp, errors can be much harder to diagnose. A CGI object is created and the printing begins. A key line of code is `$term = param("term");` to get data (a parameter) from the web page. On the web page form, the data is called "term" and it's turned into a Perl variable (with the same name, but this isn't required).

- The key searching command of the script is simply  
`if (/ $term/)`

The variable `$term` is treated like a regular expression, although it's really just be a word or part of a word. If any part of a line in the data file contains this term, it's printed. The search could be expanded to look at specific fields, but in this case all fields are searched together.

- As with all web pages, the files must be in a directory that the web server (Apache) can "see"; the server doesn't have access to all of hebrides.

```
Copy username.html to /usr/local/apache2048/htdocs/class
```

```
Copy username.cgi to /usr/local/apache2048/cgi-bin/class
```

- **On your laptop**, point your browser to your own html file with a URL similar to this:

```
http://hebrides.wi.mit.edu/class/username.html
```

- Try out the search engine. If you get an error, first check that the permissions of the CGI script is still 755.
- (Optional) Modify the script to print out some explanatory information about the data along with the search results and/or a message if there no search results.