**Installation & Maintenance Guide**

GenomePro 2.0 is a web application for processing genomic data files. The application runs online and can be found on this internet address: <http://genomepro.cis.fiu.edu>. The site runs on Apache2 and Postgres, and uses PHP and HTML as well as CSS to display client side information. ANSI C programs solve GenomePro’s tools.

The following are brief, yet descriptive steps on how to install the application.

**Requirements**

1. LINUX/UNIX Operating System
2. Cron support
3. Apache2 Web Service\*
4. Postgres Database\*

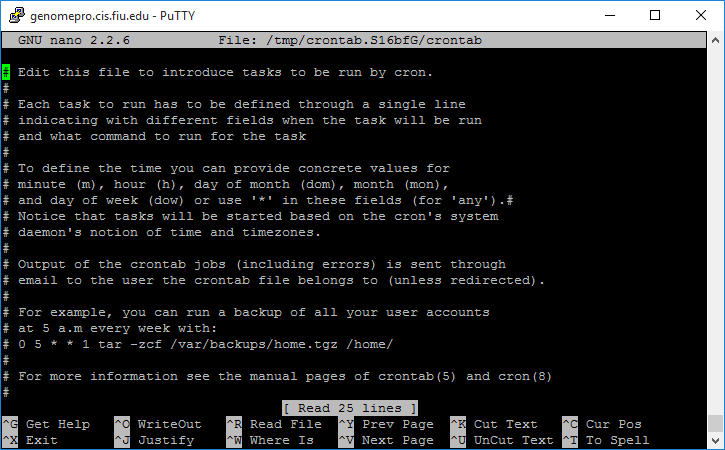
*\* Tutorials on this can be found online or below at the appendix*

**Step 1 – Moving the Website**

Save the project root directory contents to the new root directory where you would like to host the site. It is usually ‘/var/www/html/’, but it might change due to your system installation settings. The site should immediately be accessible at the site address, which is ‘http://localhost/’ on the local machine. Please make sure not to rename anything! If you get a 404 error or the likes, then Apache2 was not installed successfully, or the permissions of the files are not set correctly. Please fix them using ‘chmod’. A good suggestion is 755.

**Step 2 – Installing CRON Jobs**

The server has three different CRON jobs that must be installed on the system. Execute command ‘sudo crontab –e’ on the terminal to do this. Enter your password if asked. A dialogue similar to this one should appear if everything worked well:



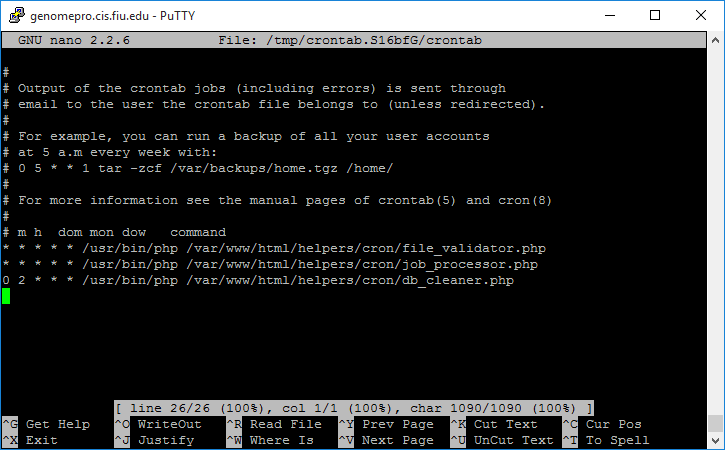
At the bottom of the file, add the following lines:

\* \* \* \* \* /usr/bin/php /var/www/html/helpers/cron/file\_validator.php

\* \* \* \* \* /usr/bin/php /var/www/html/helpers/cron/job\_processor.php

0 2 \* \* \* /usr/bin/php /var/www/html/helpers/cron/db\_cleaner.php

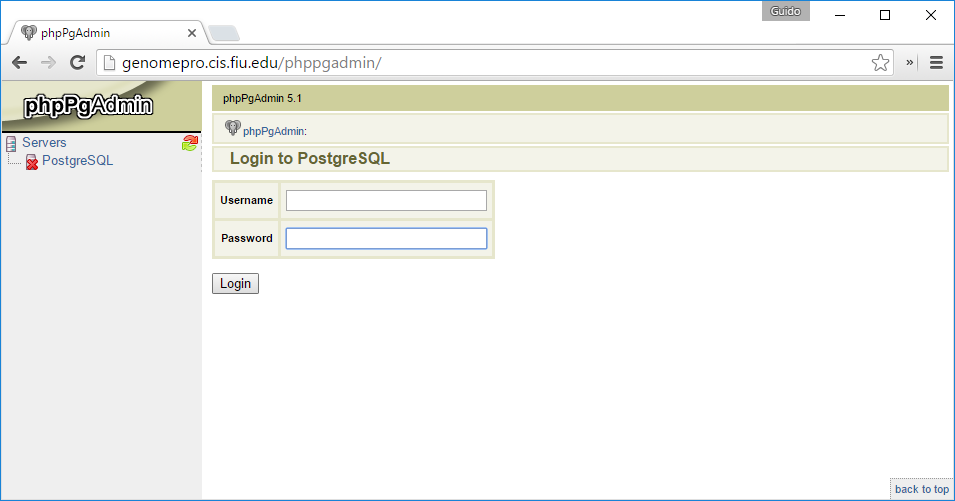
Make sure that the lines are changed depending on your system. The second argument, ‘/usr/bin/php’, is where PHP is installed. That may change depending on your system. Same goes for where the scripts are, which is, ‘/var/www/html/helpers/cron/file\_validator.php’. Also note that the arguments are absolute paths, not relative. The GenomePro server crontab looks like this:



**Step 3 – Creating the Database**

The GenomePro server can work with any database that has the same structure as the previous one, as long as it is a Postgres database. There are SQL dump files that you can use to convert your Postgres database to our structure. Please find them under the ‘/Code/Database’ folder.

Visit ‘localhost/phppgadmin’ or ‘[your\_site\_doman]/phppgadmin’, depending on if you’re on the local or a remote PC, and log in by clicking ‘PostgreSQL’ on the left hand side. Use the credentials provided during installation, or the default ones. See picture below for a more visual explanation:

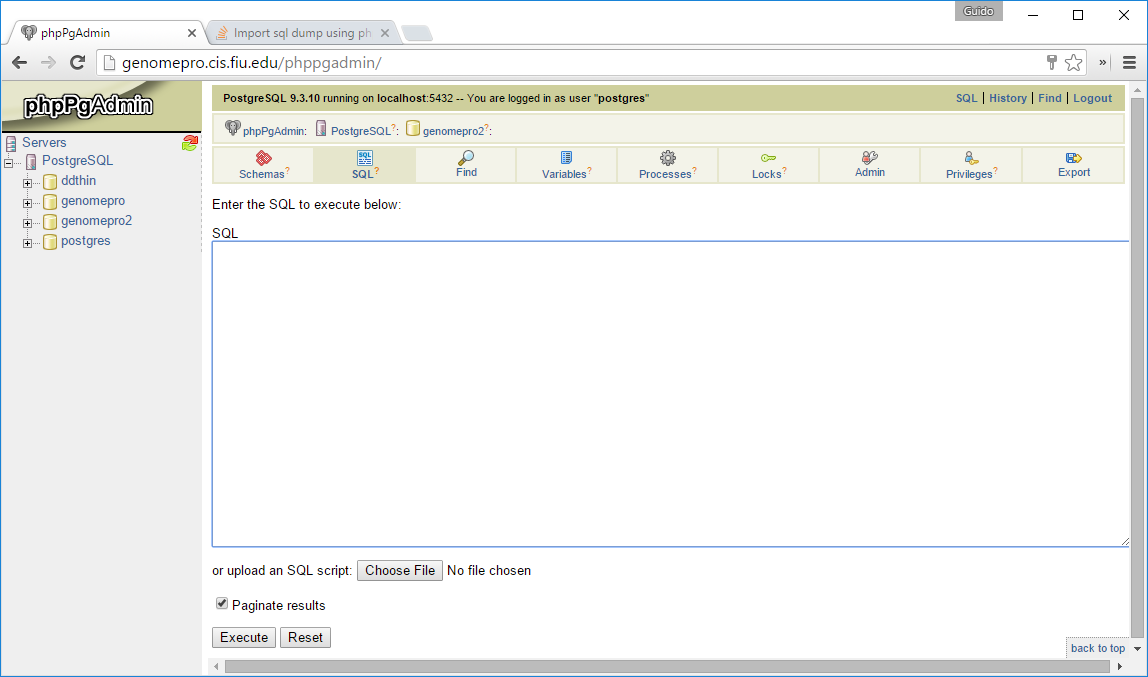


1

2

33

Once you are logged in, create a new database, click on ‘SQL’ at the middle and load your SQL script to create the database as well as the tables required for GenomePro. It is highly suggested you don’t change these either, as parts of the site will stop functioning if not handled for.



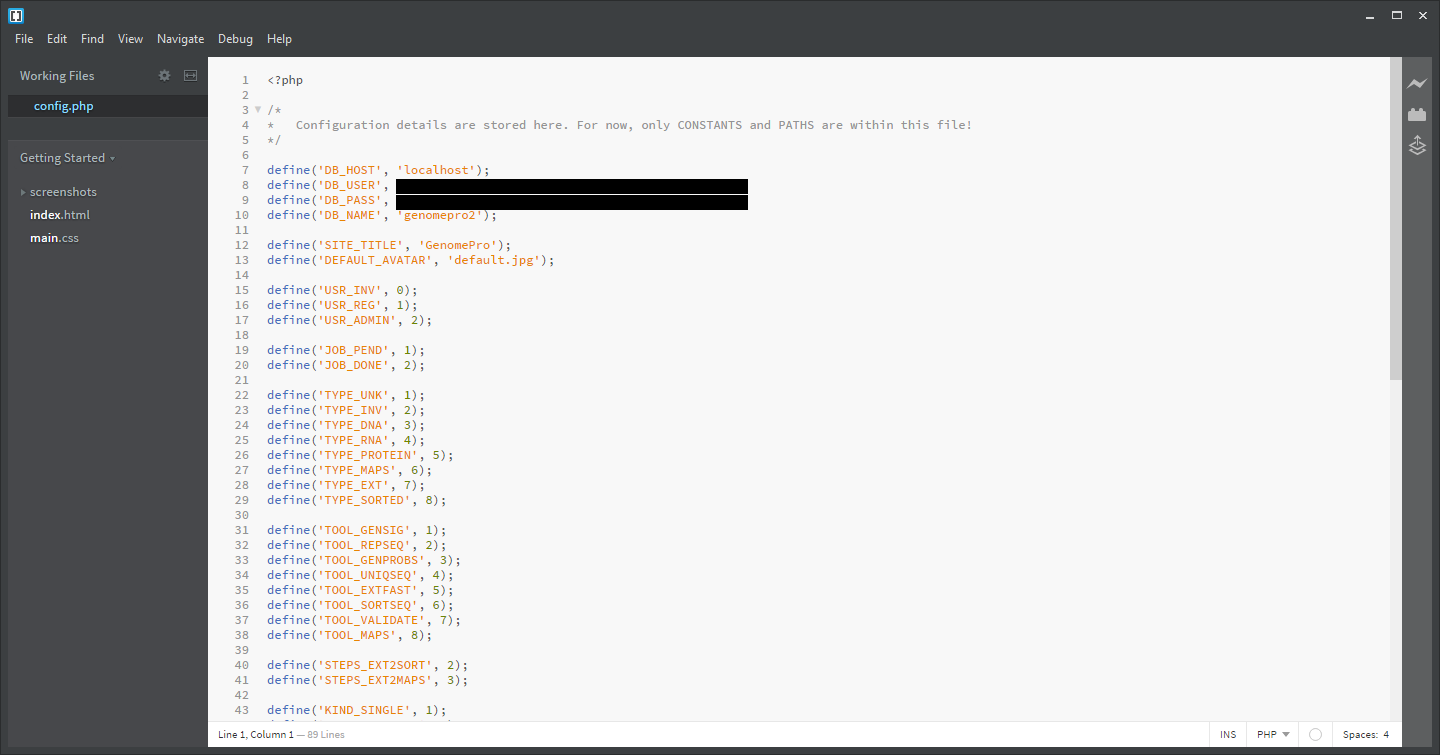
33

2

1

**Step 4 – Editing Config File**

The last step required so that GenomePro successfully works on your server is to make sure the ‘config’ file found under ‘[root]/config/config.php’ must be updated with new values. These include, but are not limited to, the database’s new name, username, password, and other global definitions. The changes should be minimal. Here’s a sample of what the file looks like:



**Appendix**

Below you can find some quick guides on how to install Apache2 and Postgres, courtesy of the GemomePro 1.0 Team, led by Yordan Alvarez and Yohan Santos. We hope that the below tutorials can assist you in the endeavor, and suggest verifying online, due to any updates to the implementation of Apache2 or Postgres that may have changed the procedure.

**Install PostgreSQL 9.4**

1. Install PostgreSQL
   * sudo apt-get install postgresql postgresql-contrib
2. Access PostgreSQL command prompt
   * sudo -u postgres psql postgres
3. Set “postgres” user password
   * postgres=# \password postgres
4. Enter new password:
5. Enter it again:
   * postgres=# \q

**Install phpPgAdmin**

1. Install phpPgAdmin:
   * sudo apt-get install phppgadmin
2. By default, you can access phppgadmin using:
   * http://localhost/phppgadmin

**Access Remote phpPgAdmin**

1. Edit file /etc/apache2/conf.d/phppgadmin:
   * sudo nano /etc/apache2/conf.d/phppgadmin
2. Comment the following line:
   * #allow from 127.0.0.0/255.0.0.0 ::1/128
3. Uncomment the following line to make phppgadmin from all systems:
   * allow from all
4. Edit /etc/apache2/apache2.conf:
   * sudo vi /etc/apache2/apache2.conf
5. Add the following line:
   * Include /etc/apache2/conf.d/phppgadmin
6. Then, restart apache service.
   * sudo service apache2 restart

**Configure phpPgAdmin**

1. Edit file /etc/phppgadmin/config.inc.php
   * sudo nano /etc/phppgadmin/config.inc.php
2. Find the following line:
   * $conf['servers'][0]['host'] = '';
3. Change it as shown below:
   * $conf['servers'][0]['host'] = 'localhost';
4. And find the line:
   * $conf['extra\_login\_security'] = true;
5. Change the value to false:
   * $conf['extra\_login\_security'] = false;
6. Find the line:
   * $conf['owned\_only'] = false;
7. Set the value as true.ru
   * $conf['owned\_only'] = true;

**Restart PostgreSQL service and Apache services**

1. Save and close the file.
   * sudo service postgresql restart
   * sudo service apache2 restart
2. Important Link.
   * http://www.unixmen.com/install-postgresql-9-4-phppgadmin-ubuntu-14-10/