

ICCR's server applications summary

Genome Browser:

The genome browser address is: <http://icci-2.tau.ac.il:8000> you need to be logged in to the tau network in order to access.

- If you're using the browser for the first time, you'll need to connect it to the hub.txt file. To do this, go to the 'My Data' menu and select 'Truck Hubs.' Switch to 'Connected Hubs' and enter the following URL in the designated window:
http://icci-2.tau.ac.il:8000/folders/sf_gbib_data/hub.txt
Finally, click on 'Add Hub'.

The server's genome browser is based on the Genome Browser In a Box (GBIB) implementation by UCSC. It utilizes a virtual machine (VM) to host the UCSC genome browser website framework with limited data. For information on how the program was installed as well as other information go to: <https://genome.ucsc.edu/goldenpath/help/gbib.html>

Basically, it is a locally installed copy of the UCSC genome browser that run on a virtual machine and can be accessed by multiple users simultaneously. I have installed and configured it and it should run smoothly.

I use the 'screen' program to keep the browser running constantly on the server, if you are not familiar with the program, here is short explanation on 'screen' and how to use it:

- The screen program on Linux is a command-line utility that allows you to create multiple terminal sessions within a single shell window, and to detach and re-attach those sessions as needed. It is commonly used to keep long-running processes running even after logging out of a remote shell session, or to allow multiple users to collaborate on a single terminal session.
With screen, you can create new terminal sessions, switch between them, and detach from them (leaving them running in the background) without terminating the processes running within them. You can also re-attach to detached sessions at any time, even from a different terminal or computer, and pick up where you left off.

Common uses of screen include:

Running long-running processes or programs that you want to keep running even after logging out of a remote shell session

- Overall, screen is a powerful and flexible tool that can greatly improve your productivity when working on the command line.
- Some screen commands:

screen -ls: list existing screen sessions

screen -S [name]: initiate a new screen session and label it with a name

ctrl-a + d: detach from screen session

screen -r [session id] or [name]: reattach to session

ctrl-a + k: kill session

Here are some basic commands to run on the server that you might find useful:

To start the VM in cases it is down (for example if the server has been turned off):

```
$ VBoxManage startvm browserbox --type headless
```

Close the VM:

```
$ VBoxManage controlvm browserbox acpipowerbutton
```

Example:

1. Start a screen session:

```
$ screen -S gbib
```

2. Turn on the GBIB machine:

```
$ VBoxManage startvm browserbox --type headless
```

3. Detach from screen: cntrl-a cntrl-d

Activate BLAT servers

In order to activate the BLAT applet of the genome browser, you must initiate a 'gfServer' session within the GBIB machine. To accomplish this, you need to connect to the machine and start the service. Keep in mind that the BLAT servers operate continuously in the background, so whenever the virtual machine is shut down, you will need to reactivate the service.

Currently there are 4 BLAT server processes (2 for the old *longissima* assembly and 2 for the new assembly) I needed to split each genome to two parts because the gfServer cannot handle large genomes.

How to activate BLAT server for the old *longissima* assembly chromosomes 5,6,7 and UN:

1. Log in to the screen session you already have (or open a new one):

```
$ screen -r gbib
```

2. Log in to the VM:

```
$ ssh browser@localhost -p 1235
```

3. Enter password:

```
$ browser
```

4. Activate the BLAT for the old *longissima* 6782 assembly of chromosomes 5,6,7,UN:

```
$ ~browser/bin/blat/gfServer start 127.0.0.1 42422 -  
stepSize=5 -  
log=/folders/sf_gbib_data/long_6782/log/gfServer_567Un.log  
/folders/sf_gbib_data/long_6782/190708_Ae_longissima_pseudomolecules_V1_chr567U.2bit &
```

You usually need to also activate the BLAT to the rest of the chromosomes and to the new assembly:

- a. ~browser/bin/blat/gfServer start 127.0.0.1 42420 -stepSize=5 -
log=/folders/sf_gbib_data/long_6782/log/gfServer_1234.log
/folders/sf_gbib_data/long_6782/190708_Ae_longissima_pseudomolecules_V1_chr1234.2bit &
- b. ~browser/bin/blat/gfServer start 127.0.0.1 42430 -stepSize=5 -
log=/folders/sf_gbib_data/long_6782_newV2/log/gfServer_1234.log
/folders/sf_gbib_data/long_6782_newV2/221207_longissima_pseudomolecules_v1_1234.2bit &
- c. ~browser/bin/blat/gfServer start 127.0.0.1 42432 -stepSize=5 -
log=/folders/sf_gbib_data/long_6782_newV2/log/gfServer_567un.log
/folders/sf_gbib_data/long_6782_newV2/221207_longissima_pseudomolecules_v1_567un.2bit &

GBIB files and configuration:

The login to the VM procedure described above is also used to prepare files to be displayed on the genome browser.

To get to the main folder of the browser data within the VM:

1. Log in to the VM:

```
$ ssh browser@localhost -p 1235
```

2. Enter password:

```
$ browser
```

3. Navigate to the main folder:

```
$ cd /folders/sf_gbib_data
```

There you can find the 'hub.txt' and 'genomes.txt' files as well as a folder for each genome.

You can also get to the GBIB files without logging to the VM from
/home/udiland/gbib/gbib_data

Be careful not change things that might cause the genome browser or some of its functions to fail.

Useful links:

http://genomewiki.ucsc.edu/index.php/GBiB:_From_download_to_BLAT_at_assembly_hubs

All file format available and their display setting:

<https://genome.ucsc.edu/goldenpath/help/trackDb/trackDbHub.html>

Shiny Server

The Shiny server is a locally installed server designed for hosting interactive web applications written in the R programming language.

The application designed to help users to post-process GWAS outputs created with GAPIT, the available applications can be found here:

- <http://icci-2:9100/manhattan/>
- <http://icci-2:9100/common/>
- http://icci-2:9100/common_sig/
- <http://icci-2:9100/compare/>
- http://icci-2:9100/plot_ld/

The full documentation of each application is elsewhere, more applications can be added by demand.

The code that execute the applications can be found here: /srv/shiny-server/

Here are some commands to control the shiny server:

Start the server:

```
$ sudo systemctl start shiny-server
```

Stop the server:

```
$ sudo systemctl stop shiny-server
```

Restart:

```
$ sudo systemctl restart shiny-server
```

Full documentation can be found here: <https://docs.posit.co/shiny-server/>

The configuration file of the server is in:

```
/etc/shiny-server/shiny-server.conf
```

The log file of the server is in (use it to diagnose server problems):

```
/var/log/shiny-server.log
```

Logs file for each application are in (use it to diagnose application problems):

```
/var/log/shiny-server/
```

Good Luck, contact me for questions.

Udi