Upload genomes ant tracks to GBIB and enabling BLAT

Instructions from: <http://genomewiki.ucsc.edu/index.php/GBiB:_From_download_to_BLAT_at_assembly_hubs>

After installing and running the UCSC GBIB we want to upload genomes and annotations.

The data file we want to upload need to be in a folder that is shared with the gbib VM.

You need to go to the setting of the virtualbox (instructions here: <https://genome.ucsc.edu/goldenpath/help/gbib.html#YourTracks>)

The first step is to generate a hub file which is the main container of all our genomes.

It needs to be in the top folder and called hub.txt

An example file:

hub ICCI

shortLabel genomes

longLabel genome assemblies from ICCI

genomesFile genomes.txt

email udiland@tauex.tau.ac.il

The following rules must be obeyed:

* hub: name without spaces.
* shortLabel: limited to 17 characters.
* longLabel: limited to 80 characters.

more Instructions are here: https://genomes.crick.ac.uk/goldenPath/help/hgTrackHubHelp.html#Setup

Than in the same folder we create another file that will describe the genoms we have in our hub (each genome will be display in a different assembly)

the file name is genomes.txt

An example:

genome Longissima

trackDb long/trackDb.txt

groups long/groups.txt

description 190708\_Ae\_longissima\_pseudomolecules\_V1

twoBitPath long/longissima.2bit

organism Aegilops longissima

defaultPos chr1S:1000000-2000000

scientificName Aegilops longissima

This is what is an example of one genome we will have as many as we want in this file.

Lets go over it.

We have the genome name, a path to a file caled trackDb.txt, a path to a file called groups.txt

(both of them in the same folder), description, a path to a 2bit file (this is the genome sequence)

, organism name, default position to display and scientific name.

Additional information: http://genomewiki.ucsc.edu/index.php/Assembly\_Hubs

Another example of a genome (it is in the **same** genomes.file seperated from the previous assembly by a blanck line)

genome Sharonensis

trackDb shar/trackDb.txt

groups shar/groups.txt

description 181220\_Ae\_sharonensis\_AS1644\_pseudomolecules\_v1

twoBitPath shar/sharonensis.2bit

organism Aegilops sharonensis

defaultPos chr1S:1000000-2000000

scientificName Aegilops sharonensis

Now you need to open a new folder for each genome (like ‘shar’ and ‘long’ in the example above)

And than put a fasta file of the genome

usualy we begain with fasta file or gff and we need to convert them to files that can be uploaded to the genome browser.

There we have two options:

1. go to that directory yourself and create there the data we need

many executable you need can be downlaod from <http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/>

2. work on the shell that comes with the gbib.

To acces it type:

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

The password is: browser

Than you are redirected to the terminal that controls the gbib

##############################################################################

here you can use several commands that control settings of the GBIB.

For example:

* browser@browserbox:$ gbibAddTools - will download many tools that you need to mnipulate files it will also download the software you need to run BLAT.
* browser@browserbox:$ gbibOffline – will disconnect the gbib from the web. Important when you want to protect your data.

After running those commands you need to disconnect so type:

browser@browserbox:$ sudo shutdown -r now

To recconect, go again:

$ ssh browser@localhost -p 1235

more commands: <https://genome.ucsc.edu/goldenpath/help/gbib.html#YourTracks>

To go to the data you in the shared folder:

browser@browserbox:$ cd /folders/sf\_gbib\_data/

The following command where made directly on the data folder from the shell using executables that I have downloaded from:<http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/>

and some from the gbib shell

#######################################################################

in oreder to upload genome and/or annotation you need to have file in 2bit / bigbed formats.

- Take the fasta sequence and make a 2bit file (The executables are in gbib\_data folder)

$ ./faToTwoBit /storage/nr/db/nucleotide/160509\_Chinese\_Spring\_v1\_pseudomolecules.fasta trit\_a/Triticum\_aestivum.2bit

- Get chromosome size data from the 2bit file

$ ./twoBitInfo trit\_a/Triticum\_aestivum.2bit stdout | sort -k2nr > trit\_a/trit\_a\_chromSizes\_sorted.txt

-Build an AGP file from the fasta file, marking all N's as gaps, using the hgFakeAgp command.

$ ./hgFakeAgp -minContigGap=1

/storage/nr/db/nucleotide/160509\_Chinese\_Spring\_v1\_pseudomolecules.fasta Triticum\_aestivum.agp

- Check if the new AGP file matches the fasta file:

$ sort -k1,1 -k2n,2n Triticum\_aestivum.agp > Triticum\_aestivum\_sorted.agp

$ ./checkAgpAndFa trit\_a/Triticum\_aestivum\_sorted.agp trit\_a/Triticum\_aestivum.2bit

We also want to look at the places where there are genes in that genome so we will upload an annotation file.

I put a gff3 file in the folder via the machine and then I need to convert it to bigbed filename

so I go to the local host: ssh [browser@localhost](mailto:browser@localhst) -p 1235

and than:

$ gff3ToGenePred iwgsc\_refseqv1.0\_2017Mar13.gff3 stdout | sort -k2,2 -k4n,4n > trit\_a.genePred

# It gave an error of gff3 header. To fix it I needed to logout of the local host and do it through the regular file system

# add a gff header as a first line

$ sed -i '1 i\##gff-version 3' iwgsc\_refseqv1.0\_2017Mar13.gff3

# than change gff to genePred (the executable is in another folder)

$ ./gff3ToGenePred ../trit\_a/iwgsc\_refseqv1.0\_2017Mar13.gff3 stdout | sort -k2,2 -k4n,4n > ../trit\_a/trit\_a.genePred

# genePred to

$ ./genePredToBed ../trit\_a/trit\_a.genePred ../trit\_a/trit\_a.bed

# bed to bigbed

first get chrom sizes

$ samtools faidx /storage/nr/db/nucleotide/160509\_Chinese\_Spring\_v1\_pseudomolecules.fasta | cut -f1,2 > chromsizes

# Than

$ ./genePredToBed ../trit\_a/trit\_a.genePred ../trit\_a/trit\_a.bed

( In this case I got an error because there was a difference between the fasta and gff files, I fixed it manualy)

# And now make a bigbed file (with the option of name)

$ ./bedToBigBed -extraIndex=name ../trit\_a/trit\_a.bed ../trit\_a/trit\_a\_chromSizes\_sorted.txt ../trit\_a/trit\_a.bb

Now we have a file to present the sequence (.2bit) and a file for the annotaion (.bb)

You need to make a trackDb.txt file that will store the information.

An example:

track genes

type bigBed 12

shortLabel genes sharonensis

longLabel genes annotations Feb2020

bigDataUrl ./tracks/sharonensis.bb

priority 1

# Activate blat

( Because I already typed the command gbibAddTools I had the blat program installed)

Followed instructions from [here](http://genomewiki.ucsc.edu/index.php/GBiB:_From_download_to_BLAT_at_assembly_hubs)

From the folder where the 2bit file is create a log folder and than

browser@browserbox:$> ~browser/bin/blat/gfServer start 127.0.0.1 42420 -stepSize=5 -log=log/gfServer.log longissima.2bit &

I got an error (in the log file):

error: Sorry, can only index up to 4294967296 bases, longissima.2bit has 6702472785

This is because there is a memory limit to the indexing of the gfserver there is nothing I can do to change it.

So my option is to split the genome and run blat for each section seperatly.

I install a tool to split fasta

udiland@icci:~/gbib/gbib\_data/exe$ pip3 install split-fasta

Than split to chromosomes:

udiland@icci:~/gbib/gbib\_data/long$ splitfasta /storage/nr/db/nucleotide/190708\_Ae\_longissima\_pseudomolecules\_V1.fasta

Than I merged 4 chromosomes to each of two files using cat

And than make a 2bit file from each one

browser@browserbox:/folders/sf\_gbib\_data/long$ faToTwoBit 190708\_Ae\_longissima\_pseudomolecules\_V1\_chr1234.fasta 190708\_Ae\_longissima\_pseudomolecules\_V1\_chr1234.2bit

Now I tried to run the blat command again

browser@browserbox:/folders/sf\_gbib\_data/long$ ~browser/bin/blat/gfServer start 127.0.0.1 42420 -stepSize=5 -log=log/gfServer.log 190708\_Ae\_longissima\_pseudomolecules\_V1\_chr1234.2bit &

And than:

browser@browserbox:/folders/sf\_gbib\_data/long$ ~browser/bin/blat/gfServer start 127.0.0.1 42421 -trans -log=log/gfServer-trans.log 190708\_Ae\_longissima\_pseudomolecules\_V1\_chr1234.2bit &

(for the first time it can take some time…)

(Do it also to the other half of the genome)

And than:

* Edit the file genomes.txt of the assembly hub in order to include the lines relatives to blat and transBlat (pay attention to the capital 'B' at "transBlat"):

blat 127.0.0.1 42420

transBlat 127.0.0.1 42421

isPcr 127.0.0.1 42420

Now do it for the second part

For longissima:

$ ~browser/bin/blat/gfServer start 127.0.0.1 42422 -stepSize=5 -log=log/gfServer\_567Un.log 190708\_Ae\_longissima\_pseudomolecules\_V1\_chr567U.2bit &

$ ~browser/bin/blat/gfServer start 127.0.0.1 42420 -stepSize=5 -log=log/gfServer\_1234.log 190708\_Ae\_longissima\_pseudomolecules\_V1\_chr1234.2bit &

~browser/bin/blat/gfServer start 127.0.0.1 42430 -stepSize=5 -log=log/gfServer 221207\_longissima\_pseudomolecules\_v1\_1234.2bit &

~browser/bin/blat/gfServer start 127.0.0.1 42432 -stepSize=5 -log=log/gfServer 221207\_longissima\_pseudomolecules\_v1\_567un.2bit &