Cannabis Genome Browser and BLAT Tool

Mihai, Sierra, Ash, and Evan

Background

- In recent years, a variety of therapeutic and industrial uses of cannabis have been identified.
- However, breeding programs are limited by a lack of complete and standardized genetic tools.
- Genome assemblies, annotated transcriptomes and marker-associated genetic maps are inadequate compared to other economically significant crops.

Background

- The high heterogeneity of cannabis plants makes genetic tool development difficult compared to other crops.
- Many more subspecies/cultivars are needed to develop tools for heterogeneous crops.
- This requires funding and collaboration across research a breeding initiatives.
- Complete and standardized tools are the solution which would bring them together.

Collaborators

- **Steve Naraine:** founder and director of Aether Research Corp., specializing in Cannabis research and cultivation
- Also regulation advisor for Ministry of Justice of Colombia, Ministry of Agriculture, Agri-food Canada, Medipharm and more
- **Sunrise Genetics:** develops and applies genomics tools to assist breeders improve the desired traits of their crops and accelerate their breeding programs

Goal

- Proprietary annotation enhanced reference genome browser incorporating a BLAT query alignment tool
- Use existing BLAT code to index a genome assembly which users will align sequences to and find out where they reside
- The BLAT tool will interface with the JBrowse program which will allow users to visualize where on the chromosomes their sequences are located
- These are standard tools in the industry, used by WormBase.org for instance

Aims

- Aim 1: Download the source code for BLAT and JBrowse and set them up to work on our web browsers. Test out the functionality of BLAT by uploading a sample reference sequence and performing searches against it to visualize the results. Ensure that we can manipulate the data to learn how JBrowse functions and how it can be adapted.

- Aim 2: Test the search tool out using the GBS and RNA-seq data available online. Test out our implementation of the BLAT and JBrowse tools.

- Aim 3: Package these tools and everything we have learned into a web-based version that can be hosted online on a web server owned by either Aether Research or Sunrise Genetics.

Basic Local Alignment Tool (BLAT)

- BLAT is an alignment tool similar to BLAST
- Commonly used to look up the location of a sequence or determine exon structure of mRNA
- Rather than using GenBank, a stored index is used
- Index is made from an assembly of the genome which is stored in memory
- Can be run on a local machine and uses less than a gigabyte of RAM

BLAT Advantages

- Much faster than BLAST
- Can submit a long list of queries in FASTA format
- Five output sort options
- Direct link to the UCSC browser
- Alignment block details in natural genomic order
- Can launch alignment as part of a custom track

BLAT Installation

- Download the latest source code from https://users.soe.ucsc.edu/~kent/src/ and take the latest file (blatSrc35.zip)
- Install Ubuntu from the Windows 10 app store, set up a user account and create a folder called “unix” on your Windows desktop
- Create a link to that folder in your home directory by running the following command:

```
ln -s  '/mnt/c/Users/<yourwindowsusernamehere>/Desktop/unix' ~/unix
```

- Unzip the blatSrc35.zip folder into the unix directory and follow the instructions in the README file to install.
- If there is a MACHTYPE issue, it is because MACHTYPE is in your shell as a local variable and not an environmental variable. This can be solved by running:

```
export MACHTYPE=x86_64
```

JBrowse

- JBrowse is an embeddable genome browser with a fully dynamic AJAX interface
- It is very quick and scales to large datasets
- Mostly built in Javascript and HTML5 with formatting tools written in Perl
- Completes almost all of its work directly in the user’s web browser

JBrowse Features

- Fast and smooth scrolling and zooming
- Scales to large genomes and has deep-coverage sequencing
- Supports GFF3, BED, FASTA, Wiggle, BigWig, BAM, VCF, REST, and more
- Includes a track selector for large installations consisting of thousands of tracks
- Light server resource requirements
- Easy customization with a vast array of possible plugins

JBrowse Installation

- Download and uncompress JBrowse from http://jbrowse.org/install/
- Put JBrowse under a directory that's served by your web server (e.g. /var/www/html)
- Run `./setup.sh` to automatically install Perl prerequisites and format the example datasets
- Volvox algae sample genome can be viewed by navigating to localhost/jbrowse/jbrowse-1.13.1-release/index.html?data=sample_data/json/volvox

Uploading a Reference Sequence

- Download E.coli/Lizard/Snail genome from UCSC in fasta format (or .fai)
  - Chosen due to small size, good performance on a laptop
  - Proprietary assembly from Sunrise Genetics could not be sent to us
- cp to jbrowse folder under web server, run set up:

```
perl bin/prepare-refseqs.pl --fasta anoCar2.fa
```

  - Builds a ‘data’ folder that holds the necessary files
- Navigate to localhost/jbrowse/jbrowse-1.13.1-release/ to see the results
- Upload from a database, gff file, or config file

Uploading Custom Annotation Tracks

- Download annotations in GFF format from Ensembl
- JBrowse also accepts other formats such as BED and GBK

```
perl bin/flatfile-to-json.pl --gff ecoli.gff --tracklabel Annotations
```

- Once again navigate to localhost/jbrowse/jbrowse-1.13.1-release and check the Annotation track to see it

Running BLAT

- BLAT by default has a command-line interface
- The web version is known as webBlat, and requires a running gfServer instance to function
- The following things are required to host your own BLAT server:

1. A reference sequence in fasta or .2bit format
2. A web server of some kind to host the page locally, a DNS if you want to host it online (e.g. Apache2)
3. A web browser to visualize the results

Configuring webBlat

1. If the reference genome is a .fa file, you need to run faToTwoBit to convert it to a 2bit file:

   ```
   faToTwoBit ecoli.fa ecoli.2bit
   ```

2. You also need to modify webBlat.cfg to include the new gfServer instance:

   ```
   gfServer localhost 17779 /path/to/genome/ E coli
   ```

3. Navigate to the folder with your genome in .2bit form and run the following commands:

   ```
   sudo service apache2 start
   gfServer start localhost 17779 ecoli.2bit
   ```

4. Go to localhost/cgi-bin/webBlat to see it running

JBrowse Custom URIs

- JBrowse uses custom URIs to specify what tracks to use and other information when launching the JBrowse instance
- Unfortunately these are not human-readable as many special characters need to be “escaped” to prevent them from being interpreted incorrectly by the browser
- One challenge was to add a custom URL to the webBlat source code that launches a JBrowse instance with a custom track to visualize the hits
- webBlat is written in C, which also requires escaping special characters like %