Basic Stacks (Atmosphere Images Tutorial)

External resources

1. http://creskolab.uoregon.edu/stacks/manual/

2. http://evomics.org/learning/genomics/stacks/

Note: most of the tutorial is adopted by the evomics tutorial (2nd link above)

Learn about allocations
Learn about CyVerse's allocation policies here.

Expected learning outcome

The goal of this exercise is to familiarize students with the use of next generation sequence data produced from Reduced Representation Libraries (RRL) such as Restriction site associated (RAD) tags. These libraries are often used for genotyping by sequencing, and can provide a dense set of single nucleotide polymorphism (SNP) markers that are spread evenly across a genome. Students will gain experience with a computational pipeline called Stacks that was designed for the analysis of such data. Data will be analyzed from an organism without a reference genome in order to create a genetic map.

Participants will learn how to:

1. Prepare raw RAD Illumina data for analysis by removing low quality reads and demultiplexing a set of barcoded samples.
2. Use Stacks to assemble RAD tags de novo from parents and progeny of an F1 mapping cross.
3. Call SNPs, genotypes, and haplotypes of these individuals within Stacks.

Log in to Atmosphere and select the app

To log in to the Atmosphere virtual machine, do the following:

1. Go to https://atmo.cyverse.org/
2. Log in using your CyVerse credentials.
3. After clicking launch **New Instance**, look for **Ubuntu 16 Stacks 1.46 v.2.0**, and then select it.
4. Under instance type, select **tiny2**.
5. Click **Launch Instance**.

Datasets and software

We will analyze one dataset using Stacks.

- **Datasets**: Produced using an Illumina GAII or HiSeq2000 sequencer.
    - **Dataset 1** – This is a ‘toy’ RAD data set that contains four barcoded samples. You will use these data to become familiar with the structure of RAD sequences, as well as to become proficient with the pre-processing (i.e. cleaning and de-multiplexing) as well as calling SNP markers and exporting them to different formats STACKs supports
- **Software:** All are open source software:
    - **Stacks** (http://creskolab.uoregon.edu/stacks/) – A set of interconnected open source programs designed initially for the de novo assembly of RAD sequences into loci and genetic maps, and extended to be used more flexibly in studies of organisms with and without a reference genome. The pipeline has a Perl wrapper allowing sets of programs to be run. However, the software is modular, allowing it to be applied to many scenarios. You will use the Perl wrapper in class and the modules on your own.
    - **Bowtie** (http://bowtie-bio.sourceforge.net/) – A component of the tuxedo suite of software, Bowtie is used for aligning sequences against a reference genome. We will use Bowtie to align RAD reads against the stickleback reference genome, and then analyze

these reads within the Stacks pipeline. Although we will use Bowtie for this exercise, many other algorithms and software exist for aligning against a reference genome, and these could be used in conjunction with Stacks as well.

Exercise 1

Note: All staged data we will be using for this exercise are under **/home/<your_cyverse_username>/stacks_demo.**

1. Connect to the Atmosphere image. (Detailed instructions are located here https://pods.iplantcollaborative.org/wiki/display/atmman/Logging+In+to+an+Instance.)
2. From the command line use **"mkdir"** to create a folder were we will work and save our work):
   ```
   cd ~
   mkdir ~/stacks_demo
   ```
3. After that use **"cd"** command to move into the newly created directory.
   ```
   cd stacks_demo
   ```

Clean and demultiplex the data

Note: Under external resources above, a detailed manual of stacks is mentioned. Under the specific link http://creskolab.uoregon.edu/stacks/comp/process_radtags.php, all parameters we are using are explained in detail.

1. Within the stacks_demo directory, create a subdirectory called **"demultiplex_samples"** by using:
   ```
   mkdir demuplix_samples
   ```
2. From the stacks_demo directory, execute the following command:
   ```
   process_radtags -p ~/stacks_demo/fastq_files/ -o ~/stacks_demo/demultiplex_samples/ -b ~/stacks_demo/61VBPAAXX_key.txt
   -e apeKI -r -c --q
   ```

Execute the *de novo* population pipeline

Note: Details about this pipeline can be found at the http://creskolab.uoregon.edu/stacks/manual/#prun

1. Create a "population_analysis" sub-folder under the "stacks_demo" folder.
2. Execute the denovo_map pipeline
   ```
   denovo_map.pl -S  -s ~/stacks_demo/demultiplex_samples/sample_CCAGCT.fq -s
   ~/stacks_demo/demultiplex_samples/sample_GAATTCA.fq -s ~/stacks_demo/demultiplex_samples/sample_TATTTTT.fq -s
   ~/stacks_demo/demultiplex_samples/sample_AACGCCT.fq -s ~/stacks_demo/demultiplex_samples/sample_CCGGATAT.fq -s
   ~/stacks_demo/demultiplex_samples/sample_GAGAAT.fq -s ~/stacks_demo/demultiplex_samples/sample_TCACC.fq -s
   ~/stacks_demo/demultiplex_samples/sample_AAGGATGC.fq -s ~/stacks_demo/demultiplex_samples/sample_CGAT.fq -s
   ~/stacks_demo/demultiplex_samples/sample_GAGATA.fq -s ~/stacks_demo/demultiplex_samples/sample_TCTCAGTC.fq -s
   ~/stacks_demo/demultiplex_samples/sample_AATATGC.fq -s ~/stacks_demo/demultiplex_samples/sample_CGCCTTAT.fq -s
   ~/stacks_demo/demultiplex_samples/sample_GCGT.fq -s ~/stacks_demo/demultiplex_samples/sample_TGCAAGGA.fq -b 1 -O
   ~/stacks_demo/map_rice1.txt -o ~/stacks_demo/population_analysis/
   ```
3. Run the population program by executing:
   ```
   populations -P ~/stacks_demo/population_analysis/ -M ~/stacks_demo/map_rice1.txt -b 1 --k
   ```

Again, details of the population command can be found at http://creskolab.uoregon.edu/stacks/comp/populations.php.

Execute the *de novo* map pipeline

```
denovo_map.pl -S -p ~/stacks_demo/demultiplex_samples/sample_CCAGCT.fq -p
~/stacks_demo/demultiplex_samples/sample_GAATTCA.fq -r ~/stacks_demo/demultiplex_samples/sample_TATTTTT.fq -r
~/stacks_demo/demultiplex_samples/sample_AACGCCT.fq -r ~/stacks_demo/demultiplex_samples/sample_CCGGATAT.fq -r
~/stacks_demo/demultiplex_samples/sample_GAGAAT.fq -r ~/stacks_demo/demultiplex_samples/sample_TCACC.fq -r
~/stacks_demo/demultiplex_samples/sample_AAGGATGC.fq -r ~/stacks_demo/demultiplex_samples/sample_CGAT.fq -r
~/stacks_demo/demultiplex_samples/sample_GAGATA.fq -r ~/stacks_demo/demultiplex_samples/sample_TCTCAGTC.fq -r
~/stacks_demo/demultiplex_samples/sample_AATATGC.fq -r ~/stacks_demo/demultiplex_samples/sample_CGCCTTAT.fq -r
~/stacks_demo/demultiplex_samples/sample_GCGT.fq -r ~/stacks_demo/demultiplex_samples/sample_TGCAAGGA.fq -b 1 -A CP
-o ~/stacks_demo/map_analysis/
```

Execute the genotypes program

Finally, we execute the genotypes program to filter and format the data to one of the formats that it supports:

```
genotypes -P ~/stacks_demo/map_analysis/ -b 1 -m 5 -r 3 -o joinmap -t BC1
```