Genomic analyses of non-model organisms with RAD-seq and Stacks

Instructor:

Julian Catchen <jcatchen@uoregon.edu>

Institute of Ecology and Evolution, University of Oregon

Objectives:

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, and from an organism with a reference genome to identify signatures of selection.

Students will learn how to:

1. Prepare raw RAD Illumina data for analysis by removing low quality reads and de-multiplexing a set of barcoded samples.
2. Align RAD sequences against a reference genome
3. Use *Stacks* to assemble RAD loci, call SNPs, genotypes, and haplotypes for each individual from two populations.
4. Calculate population genetic statistics and plot these across the genome

By the end of this workshop you will be expected to know how to:

5. Manipulate raw RAD Illumina data for analysis using a variety of different parameters.
6. Align RAD tags against a reference genome to identify signatures of selection.
7. Extend what was learned to more complicated 'on your own' problems.

Datasets and Software

We will analyze two datasets using *Stacks*, and *Bowtie*

- **Data sets - All are produced using an Illumina GAII or HiSeq2000 sequencer**
    - ***Dataset 1 (DS1)*** - 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) of data before alignment or assembly.
    - ***Dataset 2 (DS2)*** - This is a set of population genomic data from the threespine stickleback. The dataset comprises 8 individuals from each of two differentiated populations, for a total of 16 samples. The RAD data were prepared using the restriction enzyme *SbfI*, and sequenced using an Illumina sequencer. These data are unpublished, but similar to those published in Hohenlohe et al. 2010.
- **Software - All are open source software**
    - ***Stacks* (http://creskoloab.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.

Introduction

The maturation of short-read sequencing technologies is ushering in the possibility of true population genomic studies in organisms with and without a reference genome. For a variety of ecological, evolutionary and population genetic studies, a dream of biologists has long been to have complete genomic information from multiple individuals from the same or different populations.

Until just a few years ago, this goal of complete genomic information of multiple individuals has been out of reach for all but a small number of model organisms. For example, half a decade ago, producing a high density genetic map for an organism was a huge investment of resources to first produce and then type the large number of genetic markers needed to adequately cover the genome. In addition, identifying genomic regions associated with phenotypic variation, or involved in the adaptation of organisms to novel conditions, was restricted to organisms for which re-sequencing projects produced a dense battery of genetic markers at a significant cost.

With the advent of next generation sequencing, the costs of sequencing have been greatly reduced. Although one day population genomic studies will involve the complete re-sequencing of multiple individuals, these studies can now be performed with an approach called genotype-by-sequencing through the sequencing of reduced representation libraries (RRL), and subsequent identification and scoring of SNPs and inference of haplotypes. These approaches can provide data on hundreds of thousands of single nucleotide polymorphisms (SNPs) spread densely across a genome at a fraction of the cost of complete re-sequencing. We developed one such approach, called restriction site associated DNA (RAD) sequencing, which has been used to identify signatures of selection, produce a high density genetic map, help assemble genomes, and be useful for studies of allelic specific transcriptional profiling. Because these data are so new, and the sample sizes of sequences often so massive, a critical related breakthrough has been the development of algorithms and software pipelines for the analysis of such data. Several pipelines now exist, and we have produced one specifically for the analysis of RAD tags that we have titled *Stacks*.

In this workshop you will learn how to clean raw RAD data and then analyze RAD data from multiple individuals that compose two populations in order to identify signatures of selection. You will align reads to the reference genome using *Bowtie* and then generate loci and call SNPs using the *Stacks* pipeline.

For more information on RAD genotyping and related methods, in particular conceptual and statistical issues, see the papers listed at the end of this document.

Connect to the Cloud

When connecting to the cloud, do the following:

1. Select AMI **Stacks_CloudBioLinux_v4 (ami-9668c8ff)**
2. Choose "**Extra Large (m1.xlarge 15GB)**" as the instance type.
3. Choose "**us-east-1a**" as the availability zone.

Setup your scratch space for use during the exercises. Execute the following commands from within the shell:

```
% sudo mkdir /mnt/working
% sudo chown ubuntu.ubuntu /mnt/working
```

Exercise 1. Data preparation

1. The first step in the analysis of all next generation sequencing data, including RAD data, is removing low quality sequences and separating out reads from different samples that were individually barcoded. This 'de-multiplexing' serves to associate reads with the different individuals or population samples from which they were derived.
2. In each exercise you will set up a directory structure on the remote server (in this case our Amazon Virtual Machine) that will hold your data and the different steps of your analysis. We will use the directory ~/`working` on the cloud to hold these analyses.
   - Each step of your analysis goes into the hierarchy of the workspace, and each step of the analysis takes its input from one directory and places it into another directory, this is known as a '**waterfall workspace**'. We will name the directories in a way that correspond to each stage and that allow us to remember where they are. A well organized workspace makes analyses easier and prevents data from being overwritten.
   - In ~/`working`, create a directory called `clean` to contain all the data for this exercise. Inside that directory, create three additional directories: `raw`, `samples`, and `hqual`. We will refer to the `clean` directory as the *working directory*.
   - Copy data set 1 (DS1):

     ```
     /data/clean/s_1_sequence.txt.gz
     ```

     to the `raw` directory.
   - Decompress the file using the `gunzip` command.
3. Your decompressed files has millions of reads in it, too many for you to examine in a spreadsheet or word processor. Examine the contents of the file in the terminal (the `head, more, tail` commands may be of use).
   - You should see multiple different lines with different encodings.
     - How does the FASTQ file format work?

    - How are quality scores encoded? (See the link to quality scores in Appendix)

  - Can you find strings of `B`s in the quality scores? What do these mean?

4. You probably noticed that not all of the data is high quality. In general, you will want to remove the lowest quality sequences from your data set before you proceed. However, the stringency of the filtering will depend on the final application. In general, higher stringency is needed for *de novo* assemblies as compared to alignments to a reference genome. However, low quality data will almost always affect downstream analysis, producing false positives, such as errant SNP predictions.

5. We will use the Stacks' program `process_radtags` to clean and demultiplex our samples.

  - You will need to specify the set of barcodes used in the construction of the RAD library. Remember, each P1 adaptor in RAD has a particular DNA sequence that gets sequenced first, allowing data to be associated with samples such as individuals or populations.

  - Enter the following barcodes into a file called `barcodes` in your working directory (make sure you enter them in the right format):

    - `TAATG     TACCA     TCAGA     TCGAG     TGACC`
    - `TGGTT     TTAAT     TTGGC     AAAAA     AAGGG`
    - `ACACG     ACGTA     AGAGT     AGGAC     ATGCT`

  - You will need to specify the restriction enzyme used to construct the library (*SbfI*), the input file (the `s_1_sequence.txt` file in the raw directory), the list of barcodes, the output directory (`samples`) and specify that `process_radtags` *clean*, *discard*, and *rescue* reads.

  - The `process_radtags` program will write a log file into the output directory. Examine the log and answer the following questions:

    - How many raw reads were there?
    - How many were retained?
    - Of those discarded, what were the reasons?

  - What can the list of "sequences not recorded" tell you about the data analyzed and about the design of barcodes in general?

6. Execute `process_radtags` a second time. This time increase the sliding window score threshold and store the output in the directory `hqual`.

    - Fix the barcodes file to add the missing data before executing.
    - What are the effects on the data of increasing the threshold?

7. Rename the output files in the samples directory to use more meaningful names:

```
sample_TAATG.fq indv_01.fq
sample_TACCA.fq indv_02.fq
sample_TCAGA.fq indv_03.fq
```

```
sample_TCGAG.fq indv_04.fq
sample_TGACC.fq indv_05.fq
sample_TGGTT.fq indv_06.fq
sample_TTAAT.fq indv_07.fq
sample_TTGGC.fq indv_08.fq
sample_AAAAA.fq indv_09.fq
sample_AAGGG.fq indv_10.fq
sample_ACACG.fq indv_11.fq
sample_ACGTA.fq indv_12.fq
sample_AGAGT.fq indv_13.fq
sample_AGGAC.fq indv_14.fq
sample_ATGCT.fq indv_16.fq
```

This task can be greatly simplified by writing a shell script in an editor, such as Emacs, and then using the search/replace function. Then, execute the shell script to actually rename the files.

If you have time

1. Remind yourself of the use of shell tools and regular expressions in Unix:
    - Using shell tools:
        - Count the number of reads in the file `s_1_sequence.txt`
        - Identify all of the barcodes in the file
        - Count the number of occurrences of each barcode.
    - (`cut, grep, sort, uniq`, and the pipe "|" are the commands you need)
    - Why are there more barcodes than you specified?
        - Can you see a pattern to the distribution of counts at barcodes?
        - Why do you think this occurs, and what does this tell you about designing barcodes for your adaptors?

2. Try using the `process_radtags` program with a range of parameters
    - Specify a couple of different (incorrect) restriction enzymes
        - How many reads were retained this time?
        - Of those discarded, what were the reasons?
    - Vary the sliding window score threshold
        - As you did before store the output in the directory `hqual`.
        - Examine the change in reads retained based upon the score you set (you could use the R commands that you will learn below to plot the relationship)
        - Can you come up with a way to determine an optimal sliding window score threshold?
        - How might the value you use depend upon the experiment that you are performing?

Exercise II. Population genomics with a reference genome

1. Population genetics is a very old field that has a rich mathematical theory, and a core set of statistical approaches for inferring parameters from genetic data. These statistics are such things as nucleotide diversity (π), differentiation statistics (i.e. $\mathbf{F_{st}}$), and measures of genetic covariance such as Linkage Disequilibrium (**D** and **D′**). However, because of methodological limitations, the majority of the theoretical, statistical and empirical work in population genetics has focused on a small number of loci. With the advent of next generation sequencing, tens or hundreds of thousands of genetic markers can now be examined in dozens of individuals, allowing the field of population genomics to truly come to fruition. An exciting new activity in population genomics is the identification of signatures of selection in wild populations. Today you will process RAD data from one oceanic and one freshwater population of threespine stickleback from Middleton Island, which is located off the coast of Alaska. One set of data comes from an ancestral oceanic population, whereas the other is from a derived freshwater population that is likely less than 60 years old. We will align these data to the stickleback reference genome using *Bowtie*, and then feed the alignments into *Stacks* in two separate batches. After *Stacks* determines the loci and associated alleles present in each population, we will export the data and calculate several population genomic statistics, including F_{ST}. Performing a study like this was nearly impossible before the advent of next generation sequencing. *For more information on population genomics, see the papers listed in that section at the end of this document.*

2. 10 minute mini-lecture on diversity and divergence parameters, kernel smoothing, and signatures.

3. Acquire and process DS3 (Middleton Island).

   - In your workspace, create a directory called `scan` to contain all the data for this exercise. Inside that directory, create three directories: `samples`, `aligned`, and `stacks`. To save time, we have already cleaned and demultiplexed this data set and will start from the cleaned samples stage.
   - Copy data set 2 from

     ```
     /data/scan/middleton_scan.tar.gz
     ```

     to the `samples` directory.
   - `Untar` and decompress the file.

4. Align the stickleback sequences against the genome with *Bowtie*

   - Run *Bowtie* on the first freshwater sample: samples/s13_fw_01.fa
     - Running bowtie with no parameters will give you a list of all options.
     - We only want to keep alignments that have a single, best alignment to the genome. With the right combination of parameters, *Bowtie* can do this natively.

       Some hints: allow only a fixed number of mismatches in the alignment, require that alignment hits are broken up into *strata*, that only the *best* strata is kept, and that only alignments with a single member per strata are used.

- The stickleback *Bowtie* database is located within this directory:

  ```
  /data/bowtie_db
  ```

  the Bowtie database is stored in several files, we will specify only the common prefix of those files and Bowtie will know how many files to read in.

- Run *Bowtie* again with the first oceanic sample: samples/s13_an_01.fa
- To save time, the remaining 14 alignments can be found here:

  ```
  /data/scan/s13_bowtie.tar.gz
  ```

  `Untar` these remaining *Bowtie* alignments into the `aligned` directory.

5. Create a new MySQL database called `middleton_radtags` and populate the tables by loading the table definitions from:

```
/usr/local/share/stacks/sql/stacks.sql
```

If you view this file, you will see all the SQL commands necessary to create tables to hold our Stacks data. We need to create the database and then feed these commands to the MySQL server:

```
    % mysql --defaults-file=/usr/local/share/stacks/sql/mysql.cnf -e "CREATE
DATABASE middleton_radtags"

    % mysql --defaults-file=/usr/local/share/stacks/sql/mysql.cnf
middleton_radtags < /usr/local/share/stacks/sql/stacks.sql
```

6. We next want to run *Stacks* on the freshwater and anadromous population.

- Run the *Stacks* `ref_map.pl` pipeline program. This program will run pstacks, cstacks, and sstacks on the members of the population, accounting for the alignments of each read.
  - Information on `ref_map.pl` and its parameters can be found online:
    - http://creskolab.uoregon.edu/stacks/comp/ref_map.php
  - Specify each *Bowtie*-aligned individual as a "sample" to ref_map.pl. Specify the `stacks` directory as the output location. **To save time, you will want to disable database interaction.**

7. Examine the *Stacks* log and output files when execution is complete.

- From the log: how are the different programs, `pstacks`, `cstacks`, and `sstacks` executed?
- How many reads are used in each `pstacks` execution?
- Familiarize yourself with the output of each Stacks' component:
  - `pstacks`: *.tags.tsv, *.snps.tsv, *.alleles.tsv
  - `cstacks`: batch_1.catalog.tags.tsv, batch_1.catalog.snps.tsv, batch_1.catalog.alleles.tsv
  - `sstacks`: *.matches.tsv

- Notice that each locus has a chromosome/base pair specified in each of the *.tags.tsv files and in the catalog files.
- Examine the Stacks output through the web interface:
    - http://instance-address.amazonaws.com/stacks/

      <the instructor will provide this address>

  Expand the details for one or more loci (click on a locus). Turn on the allele depths and click on some alleles to see the actual stack that corresponds.

8. Now calculate population genetic statistics for each SNP in the two populations. The program `populations` does this for one level of population subdivision, as we have here, so it will calculate expected and observed heterozygosity, π, F_{IS} and it includes F_{ST} as a measure of genetic differentiation between populations. It uses the same method for calculating F_{ST} as was used in the human HapMap project.

- Create a file in the **scan** directory called *popmap* that is formatted like this:

  ```
  <sample file prefix><tab><population ID>
  ```

  Include all 16 samples in this file and specify which individuals belong to which populations.
- Execute the `populations` program, supply the population map and enable kernel-smoothing.
- Now look at the output in the file `batch_1.sumstats.tsv`. There are a large number of statistics calculated at each SNP, such as the frequency of the major allele (*P*), and the observed and expected heterozygosity, and $\mathrm{F_{IS}}$. Use UNIX commands like `head`, `cat`, `cut`, `more`, `column`, and `sort` to focus on some. How are these summary statistics related to Hardy-Weinberg equilibrium?
- Examine catalog **locus 65** in the web interface (place the locus number in the catalog filter at the top of the web page). Use `grep` to extract **locus 65** from the `batch_1.sumstats.tsv` file. The sumstats file provides a measure of *P*, the most frequent allele present in the population. Verify that for the two SNPs in **locus 65** that *P* properly corresponds to the number of alleles present in the web interface.

9. Because RAD produces so many genetic markers, and because we have a reference genome sequence, we can examine population genetic statistics like F_{ST} as continuous distributions along the genome. The population program does this using a kernel-smoothing sliding window approach.

- The output file `batch_1.fst_1-2.tsv` contains F_{ST}, a measure of genetic differentiation between the two populations. What is the maximum value of F_{ST} at any SNP? How many SNPs reach this F_{ST} value?
- Look at the genomic distribution of F_{ST} in the file `batch_1.fst_1-2.tsv`. Use UNIX commands like `cut`, `sort`, and `grep` to find the genomic regions that show the highest levels of population differentiation.

  - What does the p-value generated by Fisher's exact test tell you about a particular F_{ST} measurement? How about the LOD score?

- Now plot F_{ST} over a single linkage group. First use **`grep`** to produce a new file with only the F_{ST} values for Linkage Group IV (labeled groupIV), call it **`batch_1.fst_1-2_lg4.tsv`**. Now plot this file using gnuplot. This can be done by typing in gnuplot:

```
% gnuplot
gnuplot> set terminal pdf enhanced size 8in,6in;
gnuplot> set output "./groupIV_fst.pdf";
gnuplot> set xtics ("10Mb" 10000000,"15Mb" 15000000,"20Mb" 20000000, \
                    "25Mb" 25000000,"30Mb" 30000000, "40Mb" 40000000);
gnuplot> set yrange [-0.25:1.05];
gnuplot> plot 'batch_1.fst_1-2_lg4.tsv' using 6:9 with points title “Fst”, \
         '' using 6:15 with lines lw 5 title “Smoothed Fst”;
```

- Download the resulting PDF file and open it. The red crosses represent the raw F_{ST} measures while the green line is the kernel-smoothed average value.

Exercise III. *de novo* assembly of RAD tags without a genome

1. Genetic linkage maps are very useful tools for many genetic and genomic studies. Using a defined pedigree (F_2, backcross or pseudo-testcross for example), or a genetic resource such as a radiation hybrid panel, one can examine the segregation of markers to infer their order with respect one another. Alleles at closely linked loci will co-vary more than those at distant loci, and the degree of covariance can be used to infer genetic distances *via* a mapping function. The construction of genetic linkage maps is a key step in the production of a genome assembly, especially using existing next generation sequence technologies. While whole genome shotgun sequencing will allow the assembly of many contigs, some of considerable size, structural aspects of genomes such as repetitive regions will often make complete genome assembly difficult. In this case, dense genetic maps, such as the one that you'll be producing here, can be used to tile the contigs into scaffolds that will often cover the entire genome. Even when a high quality reference genome is available, genetic maps can be very useful in identifying chromosomal features, such as inversions or translocations, that are segregating within and among populations. *For more information on genetic mapping, see the papers listed in that section at the end of this document.*

2. 10 minute mini-lecture on *Stacks*, primary/secondary reads, and parameters.

3. Make sure your scratch space is set up for use during the exercise if you have restarted the Amazon instance. Execute the following commands from within the shell:

```
% sudo mkdir /mnt/working
% sudo chown ubuntu.ubuntu /mnt/working
```

Start up the web server:

```
% sudo /etc/init.d/apache2 start
```

4. In your workspace, create a directory called `map` to contain all the data for this exercise. Inside that directory, create two additional directories: `samples`, and `stacks`. To save time, we have already cleaned and demultiplexed this data and will start from the cleaned samples stage.

   - Copy data set 2 (DS2) from `/data/map/gar_map.tar.gz` to the `samples` directory.
   - `Untar` and decompress the file.

5. Create a new MySQL database called `loc_radtags` and populate the tables by loading the table definitions from:

```
/usr/local/share/stacks/sql/stacks.sql
```

If you view this file, you will see all the SQL commands necessary to create tables to hold our Stacks data. We need to create the database and then feed these commands to the MySQL server:

```
    % mysql --defaults-file=/usr/local/share/stacks/sql/
mysql.cnf -e “CREATE DATABASE loc_radtags”

    % mysql --defaults-file=/usr/local/share/stacks/sql/
mysql.cnf loc_radtags < /usr/local/share/stacks/sql/stacks.sql
```

6. Run the Stacks’ `denovo_map.pl` pipeline program. This program will run ustacks, cstacks, and sstacks on the parents and progeny for the genetic map.

- There are three important parameters that must be specified to `denovo_map.pl`, and we must specify each of the parents and progeny from the cross.
  - Information on `denovo_map.pl` and its parameters can be found online:
    - http://creskolab.uoregon.edu/stacks/comp/denovo_map.php
- Set the `stacks` directory as the output, and set the three main parameters: *minimum stacks depth*, *distance allowed between stacks*, and *distance allowed between catalog loci*. Finally, specify 1 as the batch number.
- You will need to specify each parent in the dataset with ‘-p’ and each progeny with ‘-r’ Since there are a lot of individuals (94), it will be helpful to create the command ahead of executing it. There are several ways you could do this:

  1. Edit the command together in an external editor, say on your local laptop, then cut and paste the command into the terminal window.
  2. If you know how, create a shell script containing the command using a UNIX editor such as emacs, vi, or nano.
  3. Simply type the command out. This will take some time, but you can use tab-completion to make it easier.
- Execute the Stacks pipeline.

7. Examine the *Stacks* log and output files when execution is complete.

- From the log: how are the different programs, `ustacks`, `cstacks`, and `sstacks` executed?
- How many reads are used in each `ustacks` execution?
- Find the output of the genotypes program in the log.
  - How many markers were identified?
  - How many progeny genotypes were examined?
  - Why are corrections being made?
- Familiarize yourself with the output of each Stacks’ component:
  - `ustacks`: *.tags.tsv, *.snps.tsv, *.alleles.tsv
  - `cstacks`: batch_1.catalog.tags.tsv, batch_1.catalog.snps.tsv, batch_1.catalog.alleles.tsv
  - `sstacks`: *.matches.tsv

8. View the result of the Stacks analysis through the web interface:

    - http://instance-address.amazonaws.com/stacks/
    - Explore the web interface
        - What do the different types of markers mean?
        - Why are some markers found in more progeny?
        - Why are the number of genotypes sometimes less than the number of progeny?
        - Expand the details for one or more loci (click on a locus). Click on `Allele Depths` and `Genotypes` to view additional information.
        - Click on some alleles to see the actual stack that corresponds to the catalog locus.
    - Compare the web interface data to the batch_1.haplotypes_1.tsv file.
        - Use the `cut` and `grep` commands to view all loci from a particular progeny, and all progeny from a particular locus.
        - Is the same data contained in both sources?

9. Export a set of genotypes for use in a linkage mapping program. Run the genotypes program to export a CP type map (an F1 pseudo-testcross), with a minimum of 75 progeny per locus. Enable automated corrections and generate a file for JoinMap.
    - Examine the genotypes output file. Why are the genotypes different than in the web interface?
10. Feed the genotypes we have generated into R/qtl (we have provided a file for you).
    - Copy the following files into the `map` directory you are working in:
        - `/data/map/rqtl_map.r`
        - `/data/map/gar_bc_genotypes.csv`
    - Examine the rqtl_map.r script in an editor. Can you tell what data will be read in, which analyses will be performed, and the plots that will be produced?
    - Execute R: `% R CMD BATCH rqtl_map.r`
11. Examine the plots generated from R/qtl by downloading them and opening them in a pdf viewer. Refer to the Broman R/qtl paper to help answer the following questions.
    - The first plot is of '`Missing_Data`'.
        - Do you see any samples or markers for which the data are particularly poor?
        - What information are you using to draw this conclusion?
    - Now examine the plot titled '`Genetic_Map`'.
        - Does this look correct given what we told you about the data?
        - If not, what could explain the discrepancy?
    - Finally, examine the file called '`Recombination_Frequency`'.

- It’s very pretty, but what is it?
- What do the red and blue regions mean?
- Do we expect to see the red aligning mostly on the diagonal?
- Why might some of these regions be ‘fatter’ than other regions?

If you have time

1. Run the *Stacks* programs (cstacks, ustacks and sstacks) separately on the data set without using the Perl wrapper.
    1. Can you tell what each program is doing?
    2. Think of circumstances where you might be interested in using only some of the programs, but not all of them in the wrapper.
2. Work through the Broman guide (Genetic map construction with R/qtl).
    1. Examine additional mapping statistics.
    2. Produce the different evaluative plots that Broman suggests.
    3. Modify the mapping parameters such as LOD score and recombination frequency and see how this affects your output.
    4. Can you see how creating a genetic map is not a one step, ‘plug and play’, but involves a lot of evaluation of parameters, data and map quality?

Citations and Readings

Core readings for the lecture and workshop

Amores, A., et al. 2011. Genome evolution and meiotic maps by massively parallel DNA sequencing. **Genetics** 188:799-808.

Broman, K. W. 2010. Genetic map construction with R/qtl. Univ. Wisc. Technical Report #214.

Catchen, J. M. et al. 2011. *Stacks*: building and genotyping loci de novo from short-read sequences. **G3: Genes, Genomes and Genetics** 1; 171-182.

Davey, J. W., et al. 2011. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. **Nature Reviews Genetics** 12:499-510.

Etter, P. D., et al. 2011. SNP Discovery and Genotyping for Evolutionary Genetics using RAD sequencing. *in* Molecular Methods in Evolutionary Genetics, Rockman, M., and Orgonogozo, V., eds. (*in press*).

Ekblom, R., and J. Galindo. 2010. Applications of next generation sequencing in molecular ecology of non-model organisms. **Heredity** 107:1-15.

Hohenlohe, P. A. et al. 2010. Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. **PLoS Genetics** 6. 1-23.

NGS population genomics background, concepts and statistical considerations

Broman, K. W., et al. 2003. R/qtl: QTL mapping in experimental crosses. **Bioinformatics** 19:889-890.

Broman, K. W., and S. Sen. 2009. **A Guide to QTL Mapping with R/qtl**. Springer.

Gompert, Z., and C. A. Buerkle. 2011a. A hierarchical Bayesian model for next-generation population genomics. **Genetics** 187:903-917.

Gompert, Z., and C. A. Buerkle. 2011b. Bayesian estimation of genomic clines. **Molecular Ecology** 20:2111-2127.

Lynch, M. 2009. Estimation of allele frequencies from high-coverage genome-sequencing projects. **Genetics** 182:295-301.

Nielsen, R., et al. 2005. Genomic scans for selective sweeps using SNP data. **Genome Research** 15:1566-1575.

Hohenlohe, P. A., et al. 2010. Using population genomics to detect selection in natural populations: Key concepts and methodological considerations. **International Journal of Plant Sciences** 171:1059-1071.

Stapley, J., et al. 2010. Adaptation genomics: the next generation. **Trends in Ecology and Evolution** 25:705-712.

Luikart, G.,et al. 2003. The power and promise of population genomics: from genotyping to genome typing. **Nature Reviews Genetics** 4:981-994.

Nielsen, R., et al. 2011. Genotype and SNP calling from next-generation sequencing data. **Nature Reviews Genetics** 12:443-451.

Genetic mapping using RRL and RAD sequencing

Altshuler, D., et al. 2000. An SNP map of the human genome generated by reduced representation shotgun sequencing. **Nature** 407:513-516.

Baxter, S. W., et al. 2011. Linkage mapping and comparative genomics using next-generation RAD sequencing of a non-model organism. **PLoS ONE** 6:e19315.

Chutimanitsakun, Y., et al. 2011. Construction and application for QTL analysis of a Restriction Site Associated DNA (RAD) linkage map in barley. **BMC Genomics** 12: 1-13.

Gore, M. A., et al. 2009. A first-generation haplotype map of maize. **Science** 326:1115-1117.

RAD-seq genotyping methodology

Baird, N. A., et al. 2008. Rapid SNP discovery and genetic mapping using sequenced RAD markers. **PLoS ONE** 3:e3376.

Emerson, K. J., et al. 2010. Resolving postglacial phylogeography using high-throughput sequencing. **Proceedings of the National Academy of Sciences** 107:16196-16200.

Etter, P. D., et al. 2011. Local De Novo Assembly of RAD Paired-End Contigs Using Short Sequencing Reads. **PLoS ONE** 6:e18561

Hohenlohe, P. A., et al. 2011. Next-generation RAD sequencing identifies thousands of SNPs for assessing hybridization between rainbow and westslope cutthroat trout. **Molecular Ecology Resources** 11 Suppl 1:117-122.

Miller, M. R., et al. 2007. Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. **Genome Research** 17:240-248.

Willing, E. M., et al. 2011. Paired-end RAD-seq for de novo assembly and marker design without available reference. **Bioinformatics** 27:2187-2193.

Other reduced representation library (RRL) methodologies

Andolfatto, P., et al. 2011. Multiplexed shotgun genotyping for rapid and efficient genetic mapping. **Genome Research** 21:610-617.

Elshire, R. J., et al. 2011. A Robust, Simple Genotyping-by-Sequencing (GBS) Approach for High Diversity Species. **PLoS ONE** 6:e19379.

Rigola, D., et al. 2009. High-Throughput Detection of Induced Mutations and Natural Variation Using KeyPoint™ Technology. **PLoS ONE** 4:e4761.

van Orsouw, N. J., et al. 2007. Complexity reduction of polymorphic sequences (CRoPS): a novel approach for large-scale polymorphism discovery in complex genomes. **PLoS ONE** 2:e1172.

van Tassell, C. P., et al. 2008. SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries. **Nature Methods** 5:247-252.

Useful links

1. Quality scores
    1. http://en.wikipedia.org/wiki/FASTQ_format
    2. http://en.wikipedia.org/wiki/Phred_quality_score
    3. http://www.phrap.com/phred/
    4. http://www.illumina.com/truseq/quality_101/quality_scores.ilmn

2. Basic Unix, R and PERL commands
    1. http://mally.stanford.edu/~sr/computing/basic-unix.html
    2. http://korflab.ucdavis.edu/Unix_and_Perl/
    3. http://www.r-project.org/
    4. http://cran.r-project.org/doc/manuals/R-intro.html
    5. http://manuals.bioinformatics.ucr.edu/home/programming-in-r

3. *Stacks* download and tutorials
    1. http://creskolab.uoregon.edu/stacks/

4. Great site for information on next gen sequencing
    1. http://seqanswers.com/