Ecological Genomics Tutorials: Population & Landscape Genomics 1

September 11, 2023

Learning Objectives for 09/11/23

1. To get background on the study system (Red spruce, *Picea rubens*), and the experimental design of the exome capture data
2. To understand the general work flow or "pipeline" for processing and analyzing the exome capture sequence data
3. To visualize and interpret Illumina data quality (what is a fastq file; what are Phred (Q) scores?).
4. To learn how to make/write a bash script, and how to use bash commands to process files in batches
5. To trim the reads based on base quality scores in preparation for mapping to the reference genome

1. Red spruce, *Picea rubens*

Red spruce is a coniferous tree that plays a prominent role in montane communities throughout the Appalachians. It thrives in the cool, moist climates of the high elevation mountains of the Appalachians and northward along the coastal areas of Atlantic Canada. In the low-latitude trailing edge of the range, populations are highly fragmented and isolated on mountaintops. These "island" populations are remnants of spruce forests that covered the southern U.S. glaciers extended as far south as Long Island, NY. As the climate warmed at the end of the Pleistocene (~20K years ago), red spruce retreated upward in elevation to these mountaintop refugia, where they are now highly isolated from other such stands and from the core of the range further north.

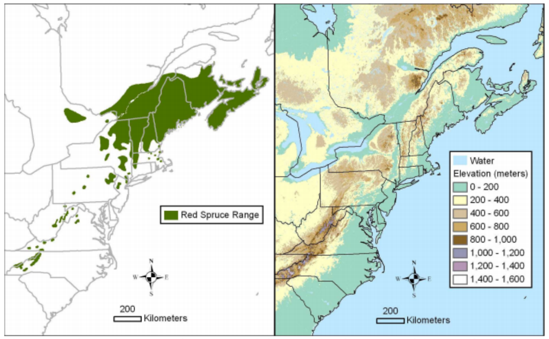

Figure 1. Left: Native range of red spruce (Little 1971); Right: Elevation map of northeastern USA (ESRI 2008).

Because of its preference for cool, moist climates, red spruce shows climate sensitivities that may make it especially vulnerable to climate change. This makes assessing the amount and distribution of genetic diversity across the landscape of red spruce an important conservation issue! Ultimately, we want to use genomic insights to help inform conservation biologists working to restore red spruce and evaluate the potential for assisted migration (a form of human-mediated dispersal) to offset the loss of adaptation under climate change. A close partner in this effort is the Nature Conservancy and the Central Appalachian Spruce Restoration Initiative (CASRI) – a multi-partner group dedicated to restoring and enhancing red spruce populations to promote their resilience under climate change.

CASRI | CENTRAL APPALACHIAN SPRUCE RESTORATION INITIATIVE

Some videos of the collaboration between UVM and CASRI on red spruce restoration:

- Building Resilience
- Seeds of Hope

Since 2015, the Keller Lab has been studying the genetic basis of climate adaptation across the entire distribution of *P. rubens.* Our main goal is to use genomics to aid conservation of red spruce under climate change through a better understanding of **(1) how genetic diversity diversity is distributed across the range and how that reflects the demographic history of population expansions, bottlenecks, gene flow, and divergence** and **(2) identify regions of the genome that show evidence of adaptation in response to abiotic climate gradients.** We hope to use this information to inform areas of the range most likely to experience climate mal-adaptation, and to help guide mitigation strategies such as sourcing seed for restoration and assisted migration.

Summary of prior population genomics research on red spruce

Our recent work funded by NSF (2017-2022) focused on early-life fitness of seedlings in response to population genomic variation and climate adaptation. That work sampled seeds and needle tissue from 340 mother trees at 65 populations spread throughout the range and generated population genomic data through exome capture sequencing. Seedlings from each mother were grown in multiple common gardens and measured for fitness traits. Based on these data, some of the insights we gleaned were:

- Red spruce has very low genetic diversity and has an Ne that has been declining for thousands of years Capblancq et al. 2020
- Populations are differentiated into 3 geographically separated clusters of genetic ancestry in the north (core) mid-latitude (margin) and southern (edge) regions of its range Capblancq et al. 2020)

- Local populations level of genetic diversity and frequency of deleterious mutations (aka, "genetic load") were related to how well seedlings survived and grew under greenhouse conditions Capblancq et al. 2021
- Seedling growth traits in common garden experiments showed heritable genetic variation and genetically-based trait divergence among the 3 ancestry groups Prakash et al. 2022
- Certain genomic regions showed strong allele frequency clines along climatic gradients indicative of selection, with gene functions often related to abiotic stress (heat and drought) Capblancq et al. 2023
- There was a hint in some of the results that hybridization with black spruce might be playing a role in some of the above results, but we lacked genomic data from black spruce to nail that down.

Let's brainstorm about some issues these data couldn't address, or where the inference of population history or climate adaptation were constrained by aspects of the experimental design?

What questions would we want to ask next?

A new dataset for analysis

The data we'll be analyzing here consist of exome capture data for adult red spruce growing in a provenance trial in northern New Hampshire near the town of Colebrook. Here are the details:

- 95 individuals sampled from 12 populations across the range (N=190 red spruce)
- Individuals were grown from seed and planted out into the provenance trial as 2 year old seedlings in 1960
- Multiple studies have assessed survival, growth (height DBH), and cold tolerance of these individuals at multiple time points over the last 60 years
- Needle tissue was sampled from surviving red spruce in the trial in May 2020 for genomic DNA
- We also sampled 18 black spruce individuals from natural stands in 2 locations distant from red spruce's range (MN and MI).
    - These will be useful for detecting black spruce ancestry in the red spruce populations, if it exists
- We used the same exome-capture probe set as Capblancq et al. (2020).
    - *Why exome capture instead of alternatives (WGS, or RAD/GBS)??*
- Exome-capture was designed based on transcriptomes from multiple tissues and developmental stages in the related species, white spruce (*P. glauca*).
- Bait design used 2 transcriptomes previously assembled by Rigault et al. (2011) and Yeaman et al. (2014).
- A total of 80,000 120 bp probes were designed, including 75,732 probes within or overlapping exomic regions, and an additional 4,268 probes in intergenic regions.
- Each probe was required to represent a single blast hit to the *P. glauca* reference genome of at least 90bp long and 85% identity, covering **38,570 unigenes**.
- Libraries were made by random mechanical shearing of DNA (250 ng -1ug) to an average size of 400 bp followed by ligation of barcoded adapters, and PCR-amplification of the library. SureSelect probes (Agilent Technologies: Santa Clara, CA) were used for targeted enrichment following the SureSelect Target Enrichment System for Illumina Paired-End Multiplexed Sequencing Library protocol.
- Libraries were sequenced on an Illumina HiSeq X to generate paired-end 150-bp reads.

Here's the table of sample population codes used in file naming and their source localities

PopCode	PopName	State/Province	Country	Latitude	Longitude
2019	Indian_Gap	North_Carolina	USA	-83.45000	35.60000
2020	Glade_Run	West_Virginia	USA	-79.83333	38.63333
2021	Bear_Meadows	Pennsylvania	USA	-77.75000	40.72910
2022	October_Mtn_State_Forest	Massachusetts	USA	-73.25000	42.36667
2024	Upper_Jay	New-York	USA	-73.66667	44.41667
2027	Pillsbury_State_Forest	New-Hampshire	USA	-72.08333	43.20000
2030	Amherst	Maine	USA	-68.38333	44.90000
2030	Amherst	Maine	USA	-68.38333	44.90000
2032	Valcartier_Forest_Experimental_Station	Quebec	Canada	-71.55000	46.91667
2100	Sheet_Harbour_Waters	Nova-Scotia	Canada	-62.73333	45.20000
2101	Corberrie	Nova-Scotia	Canada	-65.90000	44.16667
2103	Centra_Acadia_Forest_Experiment_Station	New-Brunswick	Canada	-66.33333	46.03333
2505	Eastern_Acadia_Forest_Experiment_Station	New-Brunswick	Canada	-66.20000	46.03333

2. Here's our "pipeline"

- Visualize the quality of raw data (Program: FastQC)
- Clean raw data (Program: Trimmomatic)
- Visualize the quality of cleaned data (Program: FastQC)
- Calculate #'s of cleaned, high quality reads going into mapping

We'll then use these cleaned reads to align to the reference genome next time so that we can start estimating genomic diversity and population structure.

3.-5. Visualize, Clean, and Visualize again

Whenever you get a new batch of NGS data, the first step is to look at the data quality of coming off the sequencer and see if we notice any problems with base quality, sequence length, PCR duplicates, or adapter contamination. A lot of this info is stored in the raw data files you get from the core lab after sequencing, which are in *"fastq" format*.

The fastq files for our project are stored in this path: `/netfiles/ecogen/PopulationGenomics/fastq/red_spruce`

`cd` over there and `ll` to see the files. There should be 190 fastq files – 2 for each of the 95 samples (2 files/sample because these are paired-end reads, and each sample gets a file with the forward reads (R1) and another with the reverse reads (R2)).

The naming convention for our data is: `<PopCode>_<RowID>_<ColumnID>_<ReadDirection>.fast.gz`

Together, `<PopCode>_<RowID>_<ColumnID>` define the unique individual ID for each DNA sample, and there should be 2 files per sample (and R1 and an R2)

So...what is a .fastq file anyway?

A fastq file is the standard sequence data format for NGS. It contains the sequence of the read itself, the corresponding quality scores for each base, and some meta-data about the read.

The files are big (typically many Gb compressed), so we can't open them completely. Instead, we can peek inside the file using `head`. But size these files are compressed (note the .gz ending in the filenames), and we want them to stay compressed while we peek. Bash has a solution to that called `zcat`. This lets us look at the .gz file without uncompressing it all the way. Let's peek inside a file:

```
zcat 2505_9_C_R2.fastq.gz | head -n 4
```

```
@A00354:455:HYG3FDSXY:1:1101:3893:1031 2:N:0:CATCAAGT+TACTCCTT
GTGGAAAATCAAAACCCTAATGCTGAAAGGAATCCAAATCAAATAAATATTTTCACCGACCTGTTTCGATGCCAGAATTGTCTGCGCAGAAC
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF:FF
```

Note: `zcat` lets us open a .gz (gzipped) file; we then "pipe" `|` this output from `zcat` to the `head` command and print just the top 4 lines `-n4`

The fastq file format has 4 lines for each read:

Line	Description
1	Always begins with '@' and then information about the read
2	The actual DNA sequence
3	Always begins with a '+' and sometimes the same info in line 1
4	A string of characters which represent the **quality** scores; always has same number of characters as line 2

Here's a useful reference for understanding Quality (Phred) scores. If P is the probability that a base call is an error, then:

Q = -10*log10(P)

So:

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%

The Phred Q score is translated to ASCII characters so that a two digit number can be represented by a single character.

```
 Quality encoding: !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
                   |         |         |         |         |
    Quality score: 0........10........20........30........40
```

What kind of characters do you want to see in your quality score?

Visualize using FastQC

We're going to use the program FastQC (already installed on our server). FastQC looks at the quality collectively across all reads in a sample.

First, let's `cd` back to our home directories `~/` and set up some new folders to store our work. We'll make 3 directories to store our data, scripts, and results:

```
mkdir mydata/
mkdir myscripts/
mkdir myresults/
```

Then let's cd into the `myresults/` folder then use `pwd` to prove to yourself that you're in the `myresults/` folder within your home directory. It should look like this (but with your home directory info instead of mine):

```
[kellrlab@ecogen myresults]$ pwd
/users/k/e/kellrlab/myresults
```

Now within `myresults/` let's make another folder called `fastqc/` to hold the outputs from our QC analysis. Do that on your own, just like we did above, then cd into the `fastqc/` folder and type `pwd` again to prove to yourself you did it right.

Now, we're ready to run FastQC to look at the quality of our sequencing. The basic command is like this:

```
fastqc filename.fastq.gz -o outputdirectory/
```

This will generate an .html output file for each input file you've run.

Once you've got results, let's use Filezilla to transfer the folder `~/myresults/fastqc/` over to your laptop and into the `results` folder in your github repo. Once the file transfer is complete, go to where you saved your files *on your laptop* and try double-clicking one of the html outputs. It should open with a web browser.

How does the quality look?

Since we made some changes (added files) to our github repo, we should practice committing these and then pushing to GitHub!

Take notes on your workflow today so you can remember what you did for the future!

An important final step is taking good notes on your workflow so you can remebber what you did down the road (your "future self") and share your process with others (reproducible science!). It's also really important once you start making detailed decisions that will affect the analysis outcome of your data, so you can recreate the results and explore the effect of different assumptions/decisions.

We want you to keep such a notebook for each module in the course (kind of like you would for each experiment, or each thesis chapter you will work on). We've provided you with a notebook template based on the markdown (md) language.

Notebook template in markdown (md) format

You should save a copy of this template to your github repo. You can then open it up and edit this file directly within RStudio, taking notes using either the `Source` interface if you know the markdown language (or want to learn) or you can use RStudio's built-in markdown GUI editor under the `Visual` tab. This works very similar to Word.

Here's a cheatsheet for markdown language if you want to write using `Source` code:

Markdown cheatsheet