EACCR2 NID Node Training Course: Genomics

Lecture 5

Introduction to diversity analysis

Overview

We will be covering the following within this lecture and following workshop:

- What file types will be looking at
- Summary statistics can we use to analysis population diversity
  - Tajima's D
  - What is π
  - What is F_{ST}

- How can we use these to understand populations

File types

We will be using VCF files primarily which you should already be familiar with

Do you know what the differences between these key file types are we can use with VCF tools?

- vcf.gz files vs vcf files
- How can we convert a vcf.gz from a vcf file?
- How can we convert a vcf.gz into a vcf file?

- VCFtools which we will be using generates log files and tab seperated text files when you perform various analyses, you will need to become familiar with opening/closing/editing these file types

VCF files

- These have a .vcf extension and are human readable. The format of this file type can be found here http://www.internationalgenome.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-40/
- VCF files are human readable, compressed VCF files (gz) are not, and can be open using zless
- Some programmes require your VCF files to be compressed and indexed
- This file is comprised of two key parts, a header and the main body of the file. Header lines are started with #
- We can use the header to tell us information about the sample
- The body of the file is kept in the same format, with CHROM, the position the variant is in, and the REFerence and ALTernate allele as the first columns as shown below

```
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=FAO,Number=A,Type=Integer,Description="Flow Evaluator Alternate allele observation count">
##FORMAT=<ID=FDP,Number=1,Type=Integer,Description="Flow Evaluator Read Depth">
##FORMAT=<ID=FRO,Number=1,Type=Integer,Description="Flow Evaluator Reference allele observation count">
##FORMAT=<ID=FSAF,Number=A,Type=Integer,Description="Flow Evaluator Alternate allele observations on the forward strand">
##FORMAT=<ID=FSAR,Number=A,Type=Integer,Description="Flow Evaluator Alternate allele observations on the reverse strand">
##FORMAT=<ID=FSRF,Number=1,Type=Integer,Description="Flow Evaluator reference observations on the forward strand">
##FORMAT=<ID=FSRR,Number=1,Type=Integer,Description="Flow Evaluator reference observations on the reverse strand">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality, the Phred-scaled marginal (or unconditional) probability of the called genotype">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=RO,Number=1,Type=Integer,Description="Reference allele observation count">
##FORMAT=<ID=SAF,Number=A,Type=Integer,Description="Alternate allele observations on the forward strand">
##FORMAT=<ID=SAR,Number=A,Type=Integer,Description="Alternate allele observations on the reverse strand">
##FORMAT=<ID=SRF,Number=1,Type=Integer,Description="Number of reference observations on the forward strand">
##FORMAT=<ID=SRR,Number=1,Type=Integer,Description="Number of reference observations on the reverse strand">
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  Sample1 Sample2 Sample3
chr1    2488153 .   A   G   4476.14 PASS    AC=4;AF=1.00;AN=4;DP=648;FDP=195;FR=.;FRO=2;FSAF=115;FSAR=78;FSRF=2;FSRR=0;FWDB=0.00167703;FXX=0.0101518;HRUN=
chr1    2491258 .   C   G   2611.42 PASS    AC=2;AF=0.500;AN=4;AO=146;DP=1332;FAO=146;FDP=334;FR=.;FRO=188;FSAF=87;FSAR=59;FSRF=109;FSRR=79;FWDB=-0.00494
chr1    6528100 .   GGCCCCT GGCCCTC 10278.10    PASS    AC=2;AF=1.00;AN=2;AO=90;DP=655;FAO=638;FDP=638;FR=.,HEALED,HEALED,HEALED,HEALED,HEALED;FRO=0;FSAF=
chr1    6528468 .   C   T   1859.16 PASS    AC=2;AF=0.500;AN=4;AO=120;DP=893;FAO=120;FDP=236;FR=.;FRO=116;FSAF=32;FSAR=88;FSRF=41;FSRR=75;FWDB=0.0384332;
chr1    6529188 .   C   T   11263.97    PASS    AC=2;AF=0.500;AN=4;AO=606;DP=2960;FAO=640;FDP=946;FR=.,HEALED;FRO=306;FSAF=336;FSAR=304;FSRF=11;FSRR=295;
chr1    6529443 .   A   G   5283.78 PASS    AC=2;AF=0.500;AN=4;AO=331;DP=2207;FAO=361;FDP=708;FR=.,HEALED;FRO=347;FSAF=187;FSAR=174;FSRF=196;FSRR=151;FWDB
chr1    6529747 .   A   AT  1631.35 PASS    AC=1;AF=0.500;AN=2;AO=10;DP=478;FAO=228;FDP=468;FR=.;FRO=240;FSAF=93;FSAR=135;FSRF=108;FSRR=132;FWDB=-0.009871
```

Manipulating VCF files

There are two key packages we will be using, that use VCF files. These are vcftools and bcftools. Bcftools is newer and has the ability to convert VCF files into BCF files. Two other packages that are used to compress and index vcftools (bgzip and tabix) we will also use

First we should familiarize ourselves with the format of the VCF files we can do this is Sample1.vcf that we made earlier. To open the vcf file we can do the following:

less Sample1.vcf

We can see various information about this single sample. Can you tell what the original sample name was? They began with ERR and you should be able to see them in the header lines

Manipulating VCF files

Some tools will need you to feed in a vcf.gz file. We can convert our vcf to a vcf.gz using bgzip and tabix from the htslib package

First we should load this module

module load htslib

We should also make sure that vcftools and bcftools are loaded

module list (will show us what is loaded) if vcftools and bcftools are not here, we can load them as above

In order to compress and index the vcf file we first need to use bgzip to compress the vcf file as follows:

bgzip Sample1.vcf

This will generate a Sample1.vcf.gz file. In order to now generate a vcf index we need to now use tabix:

tabix –p vcf Sample1.vcf.gz

We can then uncompress this if needed using gzip

gzip Sample1.vcf.gz

Manipulating VCF files

- You can specify whether you're inputting a vcf or .vcf in vcftools using the --gzvcf Sample1.vcf.gz or --vcf Sample1.vcf
- We can either pipe output to a file using '>' or you can specify out which will give the prefix for the file using --out

For example:
We can calculate the allele frequency in a vcf file using --freq flag
vcftools --gzvcf Sample1.vcf.gz --freq --out Sample1

This says use a gzipped vcf, output a file with allele frequency (will end .freq in output) and output a file prefixed with Sample1

Our vcf file contains SNPs and INDELs. We can remove indels using --remove-indels flag. In this case we want to make a new vcf file, we can do this using the flag --recode

vcftools --vcf Sample1.vcf --remove-indels --recode --out Sample1_SNPs_only
This uses a vcf file instead, removes indels, and makes a new vcf file called Sample1_SNPs_only_recode.vcf

We can also filter using identifies in the reference, for example if we look at the reference fasta you will see each chromosome header has >chr1. We can filter for sites only on chromosome 1 using the following

vcftools --vcf Sample1.vcf --chr chr1 --out Chr1_only

Manipulating VCF files

There are lots of other utilities in vcftool and bcftools which we will show you later but you can find from the manual

Several of the most common uses/features you might want to try are the following?

- Can filter by a list of positions, which may represent genes you want to look at using **--positions** file.list where file.list is a tab seperated file with the chr and position
- Can also look at only indels with **--keep-only-indels**
- You can also filter sites by their minor allele frequency using **--maf**
- You can also filter for sites that only have a minimum level of coverage using **--min-meanDP**
- Can specify individuals to keep/filter your vcf for **--indv**

VCFtools can also be used to perform some statistical analyses

- We will use some of the following:
- --window-pi
- --weir-fst-pop
- --TajimaD

Manipulating VCF files

In the next workshop you will learn about population structure and how to identify and classify samples based on. For now we are going to give you a file containing multiple individuals, and two lists forming two populations so that we can perform some statistics on these. This multisample vcf is called **all_samples.vcf**

The two sample lists are within the course material folder, **population_list1** and **population_list2**

If you open these files you will see that they contain just a list of the sample names for each population

You will be using bcftools to filter using these samples list as follows because it has additional functions:

bcftools view -Ov -S population_list1 all_samples.vcf -o population_list1.vcf
This means open the multisample vcf, -Ov is the output type, v means output as uncompressed vcf, -S is the name of a file with a list of samples, -o means what is the output name.

bcftools view -Ov -S population_list1 all_samples.vcf > population_list1.vcf

The above command does the same, -o and > can be used interchangeably

Repeat this for both populations so that you have a vcf for each population

Nucleotide diversity (π)

Nucleotide diversity (π)

Nucleotide diversity is used to measure the degree of polymorphism within a population.

A common measure of nucleotide diversity was first introduced by Nei and Li in 1979.
This measure is defined as **the average number of nucleotide differences per site between two DNA sequences in all possible pairs in the sample population**.

We can calculate π using vcftools. We can do this per SNP, however if we take a window covering at least ten SNPs, this is often a far more accurate way to calculate, and can reduce the power of spurious mutations.

We can calculate this in vcftools using the following command

vcftools --vcf population_list1.vcf --window-pi 10000 --out population_list1_pi
This will generate a tab seperated file prefixed population_list1_pi with a value for every 10,000 bases across the genome

We can use these values to plot a graph of the distribution of π across the genome for each population

Can we test for selection

We can also perform statistics which allow us to see whether a population is neutrally evolving or whether it is under strong selective pressure. One of the measures we can use to test for this is **Tajima's D.**

Similarly we can use vcftools to perform this test

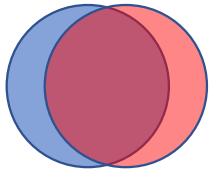
vcftools --vcf population_list1.vcf --TajimaD 10000 --out population_list1_tajD

As before we can test over a window, this is usually robust if the window covers at least 10 variants within the window. You can now perform this on both populations and the unseparated file of populations

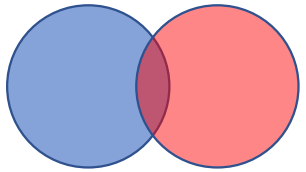
Calculating F_{ST}

Another important method used in population genetics fixation index (F_{ST}). F_{ST} ranges from 0 to 1. F_{ST} is a measure of population differentiation due to genetic structure. F_{ST} can be interpreted as measuring **how much closer two individuals from the same subpopulation are, compared to the total population**

If two populations are **closely related** and share many SNPs:
FST will be low: eg 0.1



If two populations are **less related** and share few SNPs,
FST will be low: eg 0.9



We can use this measure in order to identify the relationship between individuals and can be used to identify populations that are more, or less separated. We compare pairwise between a set of populations or between individuals in vcftools:

vcftools --vcf all_samples.vcf --weir-fst-pop population_list1 --weir-fst-pop population_list2 --out pop1_vs_2_FST

In the above command, --weir-fst is specified twice to give the lists of individuals from the two population file, that form each population. The population lists we had previously, have been used here to specify each population.

See: https://en.wikipedia.org/wiki/Fixation_index

Questions?