Variant calling from NGS data

Lecture 14, CPSC 499, Fall 2018

```
AGAATACCCTACGG reference
AGACTACCCTA-GG sample
   *       *
```

Steps in today’s workflow

- Linux, using Bowtie2 and Samtools
    - Build a Bowtie2 index for the Sorghum bicolor reference genome (~8 minutes)
    - Align reads to the genome, then sort and index the BAM files (~30 minutes)
    - Call and export variants to VCF
- R
    - Import data from VCF
    - Explore and visualize data
    - Predict protein coding changes from variants

What is variant analysis?

- Detection of sites where one or more samples differs from the reference genome
  - Identify genetic variation within a population, for downstream analysis such as genome-wide association
  - Identification of new mutations
  - Compare tumor to healthy tissue
- Files that you start with:
  - Reference genome, generally in FASTA format
  - Raw sequencing reads, generally as FASTQ

About today’s dataset

- *Sorghum bicolor*: grain crop grown in dry regions of Africa and Asia, Texas
- Sequence data from Thurber et al. (2013), study of climatic adaptation
- Produced by genotyping-by-sequencing
  - 95 samples multiplexed into one Illumina lane
  - Only sequencing adjacent to *Pst*I restriction sites
  - FASTQ files are demultiplexed, with barcode removed

FASTQ format

- Output format from next-generation sequencing (NGS) technologies (Illumina, 454, PacBio, etc.)
- Plain text
- Four lines per read
    - Comment line starting with @
    - Sequence line
    - Comment line starting with +
    - Characters indicating quality score for each base
- ShortRead package in BioConductor can be used for quality control

NCBI Sequence Read Archive

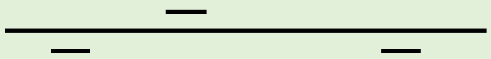
- Centralized repository for storing raw sequence reads
- Stored in specialized NCBI format, but generally uploaded as FASTQ and can be downloaded as FASTQ
- **BioProject**: Describes a study, why was it done, what species, etc
    - **BioSample**: Describes each tissue sample used in the project
        - **Experiment**: One sequencing library, or one index within sequencing library. Describes library preparation.
            - **Run**: One sequencing run of that experiment.

Short-read alignment software

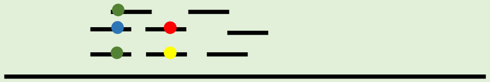
- Different from BLAST
  - Optimized for short sequence
  - Fast since it has to get through large files
  - Might not find the optimal alignment
- Bowtie2
- BWA
- GSNAP (available in BioConductor via gmapR package, but not for Windows)
- Last week we looked at RNA alignment, which could have large gaps, but this week we will do DNA alignment, which does not

Steps in any variant calling software

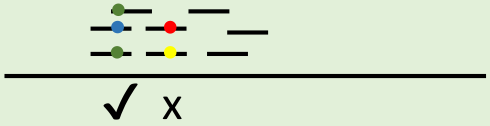
- Sequence reads must be aligned to reference genome (or to each other in non-reference pipeline)



- Identify positions where there is variation among samples, or differences between sample and reference genome



- Distinguish true variants from sequencing error



- Determine and output sample genotypes

Marker	Sample1	Sample2
Chr01:30005	A/A	A/G
Chr01:100040	C/T	T/T
Chr01:115788	C/C	C/A

Software for Calling Variants

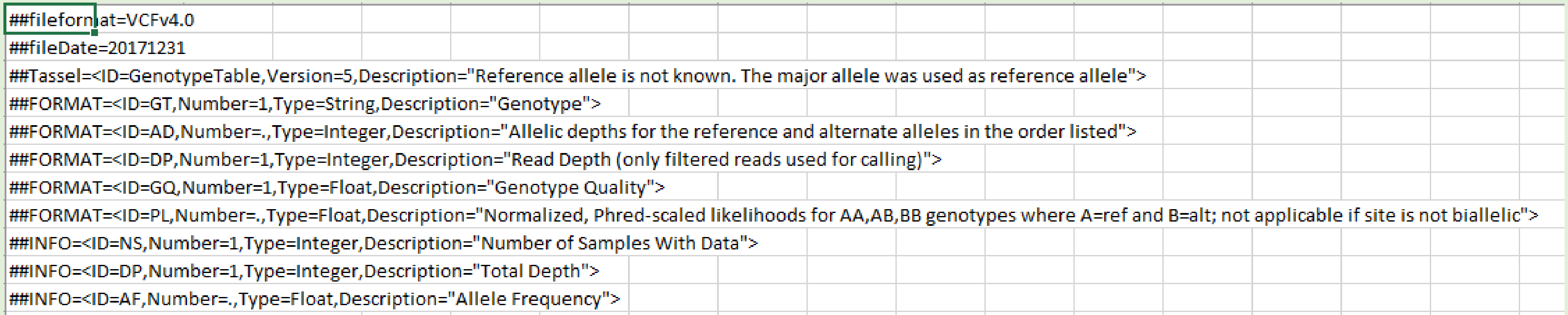
- Samtools – will use in class since we also used it for RNA-seq and viewing BAM, but not currently popular
- GATK – Genome Analysis Toolkit
- VariantTools package in Bioconductor
- Some specific for reduced-representation sequencing (GBS or RAD-seq)
  - TASSEL-GBS
  - Stacks

Variant Call Format (VCF)

- For every variant, contains position, alleles, quality statistics
- Also contains genotypes and read depth for all samples
- Tab-delimited text
- Self-documenting
  - Header contains info describing what each field means
  - You can have custom fields as long as they are documented

VCF headers

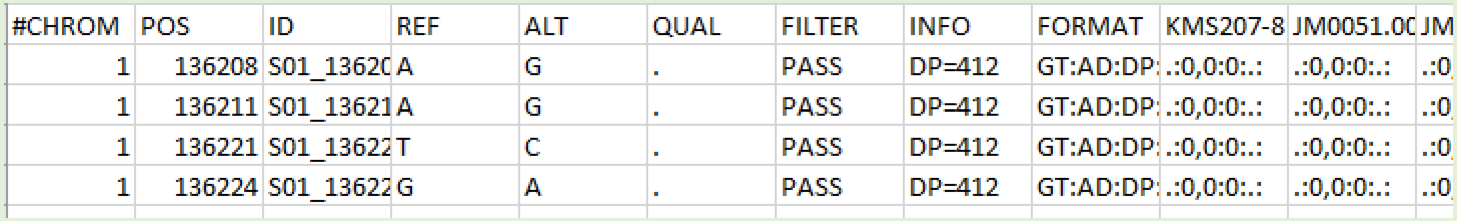
- Header lines start with ##
- INFO indicates information provided about each SNP (total depth, etc.)
- FORMAT indicates information provided about each genotype (SNP x sample)
- FILTER describes how the dataset was filtered
- SAMPLE can provide info about each sample



```
##fileformat=VCFv4.0
##fileDate=20171231
##Tassel=<ID=GenotypeTable,Version=5,Description="Reference allele is not known. The major allele was used as reference allele">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="Allelic depths for the reference and alternate alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth (only filtered reads used for calling)">
##FORMAT=<ID=GQ,Number=1,Type=Float,Description="Genotype Quality">
##FORMAT=<ID=PL,Number=.,Type=Float,Description="Normalized, Phred-scaled likelihoods for AA,AB,BB genotypes where A=ref and B=alt; not applicable if site is not biallelic">
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
```

VCF genotype table

- Column header line starts with #CHROM
- Nine mandatory headers:
  - #CHROM – chromosome name
  - POS – position on chromosome
  - ID – variant name
  - REF – allele in reference genome at this position
  - ALT – alternative allele(s)
  - QUAL – quality score
  - FILTER – whether the variant passes the filter
  - INFO – custom information about this variant
  - FORMAT – how are the individual genotypes formatted
- Every remaining column header is a sample name



#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	KMS207-8	JM0051.0(	JM
1	136208	S01_13620	A	G	.	PASS	DP=412	GT:AD:DP:	.:0,0:0:.:	.:0,0:0:.:	.:0
1	136211	S01_13621	A	G	.	PASS	DP=412	GT:AD:DP:	.:0,0:0:.:	.:0,0:0:.:	.:0
1	136221	S01_13622	T	C	.	PASS	DP=412	GT:AD:DP:	.:0,0:0:.:	.:0,0:0:.:	.:0
1	136224	S01_13622	G	A	.	PASS	DP=412	GT:AD:DP:	.:0,0:0:.:	.:0,0:0:.:	.:0

VCF genotype fields (sample x variant)

- Genotypes (GT) formatted like 0/1 for unphased, 0|1 for phased (period for missing data)
- Additional info can be stored in one genotype entry, separated by colon
- GL, GP, PL are various indicators of genotype probabilities
- DP has total read depth, AD has read depth of each allele.
- GT:AD:PL $\underbrace{0/0}_{\text{homozygote}}:\underbrace{5,0}_{\text{5 reads of reference allele, zero reads of alternative allele}}:\overbrace{0,10,55}^{\text{Homozygote most likely, heterozygote possible}}$

Uses for VCF files

- Good for archiving and sharing your genotype calls, since the data are complete and well documented

- Many programs for GWAS etc. will read VCF directly

- In BioConductor, the VariantAnnotation package has readVcf function to read all or part of a VCF

Reading VCFs with `VariantAnnotation::readVCF`

- File options
  - Give it name of VCF file to read whole thing
  - `bgzip` and `indexTabix` if you want to specify a genomic region to import
  - To loop through the VCF, use `TabixFile` and set `yieldSize` for how many SNPs to read at once
- Use `ScanVcfParam` to set which regions, samples, info fields and genotype fields to import
- Use genome argument if you need to rename chromosomes

Mini-exercise

- Use `exons` to get a `GRanges` object of all exons in sorghum genome from `TxDb`
- Pass it to `param` to just import SNPs within exons using `readVcf`

Filtering VCFs

- `filterVcf` function
- Input file --> Filter --> Output file
- Make your own functions for filtering, return TRUE/FALSE for each line
- Use `prefilters` to process each line as a text string, use `grepl` or similar to decide whether to keep
- Use `filters` to process as a `VCF` object
- Can use `ScanVcfParam`

Annotating variants

- VariantAnnotation package
- Are the variants in or near any genes? Can do quick lookups with `GRanges` and `TxDb` (`transcriptsByOverlaps`)
- Within CDS, can identify types of mutations
  - Synonymous: different codon but same amino acid
  - Non-synonymous: amino acid change
  - Premature stop codon
  - Frameshift: insertion or deletion that changes translation of all downstream sequence

Exporting SNP genotypes

- Genotypes are 0/0, 0/1, 1/1 in VCF
- `SNPMatrix` class in `snpStats` package
- Can export to
  - 0, 1, 2
  - A/A, A/B, B/B
- Can do math directly on `SNPMatrix` using `snpStats` package, or export to numeric to do stuff with normal R functions

Thursday’s lab

- Exploring a larger VCF file
- Custom VCF export