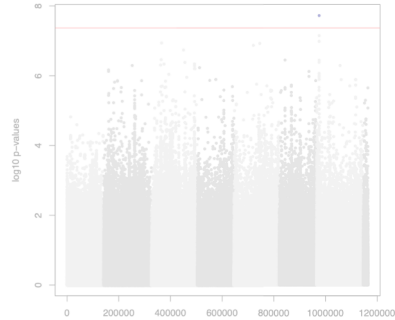
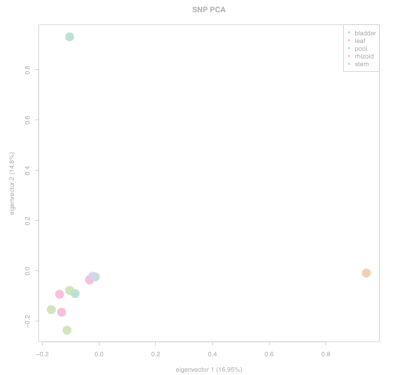




Hit the Ground Running with SNP Calling for PopGen and Evolutionary Analyses

Jacob B. Landis

School of Integrative Plant Science

Cornell University

and

BTI Computational Biology Center

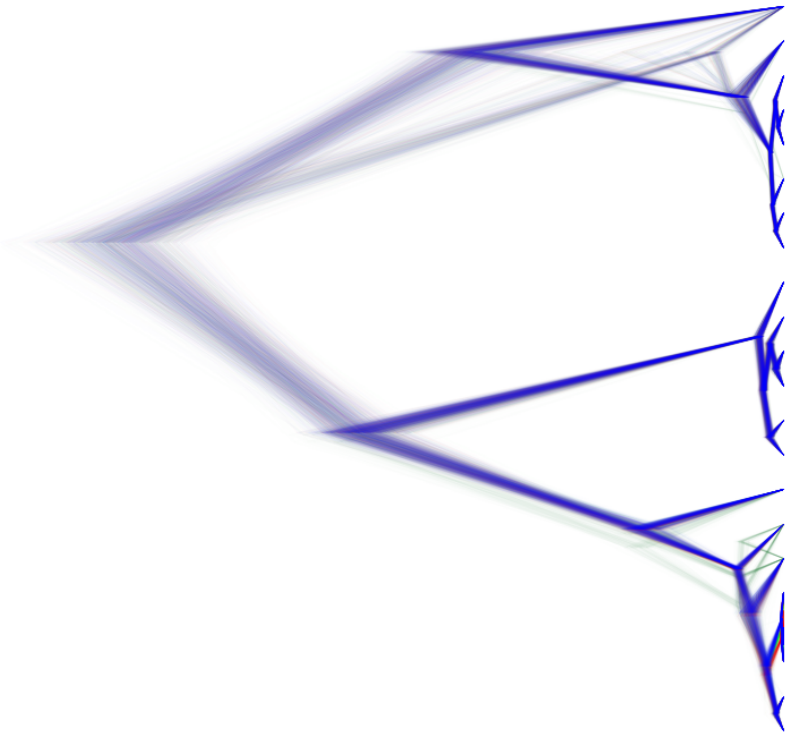
September 11th, 2020

@JLandisBotany

jbl256@cornell.edu

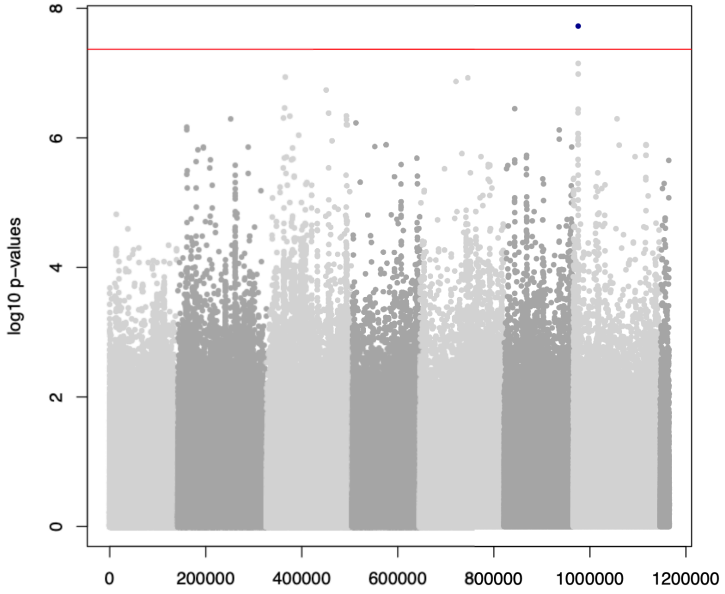
When I think of SNP analyses

Coalescent tree

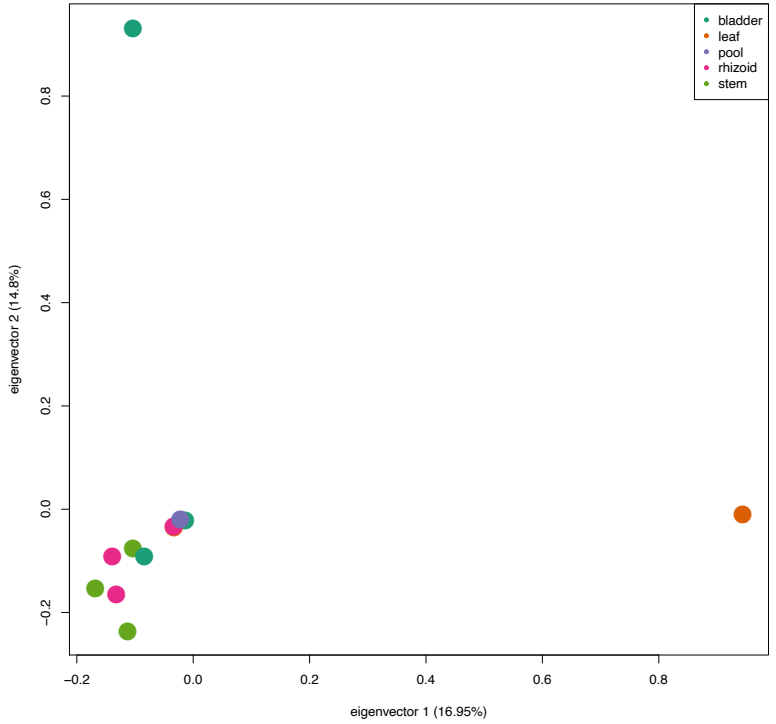


Structure/Admixture

GWAS



PCA

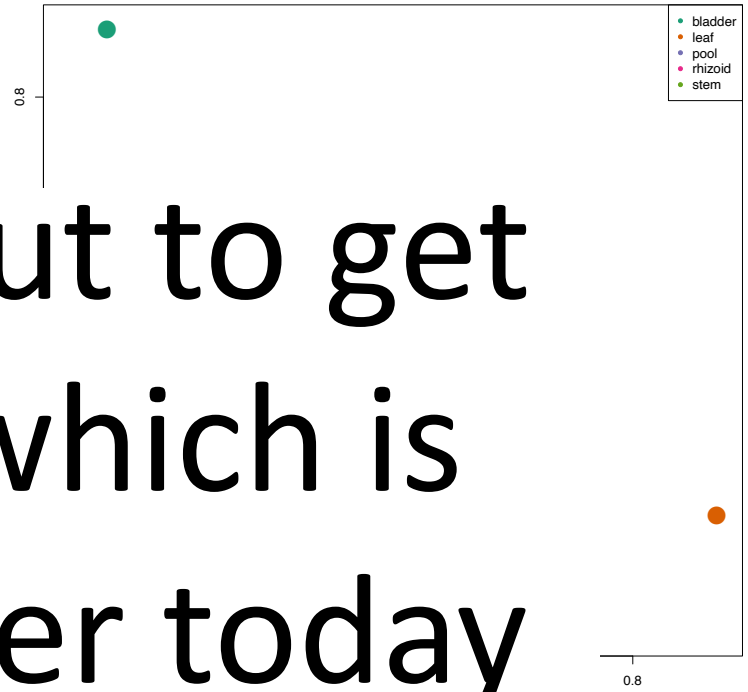


When I think of SNP analyses

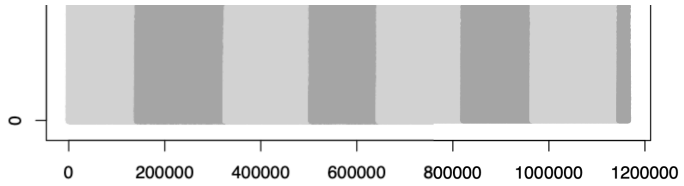
Coalescent tree

Structure/Admixture

PCA



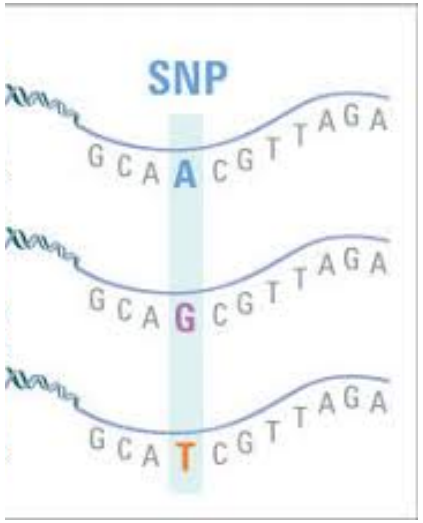
These are the goals but to get there takes work … which is what we are going over today



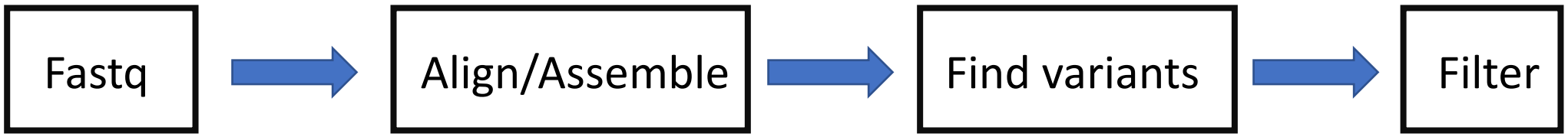
What I hope to provide

- An overview of different approaches
- Enough background on methods and scripts to get you started
- I am gearing this towards folks who do not know how to do SNP calling, but are comfortable with the command line
  - If you are not, that is ok, I will provide some options in the Discovery Environment but the resources there are quite limited
- A few tips and tricks speed up the learning curve
- Example data set to try out different methods

Before we get started…What is SNP analysis?



- At its simplest it is Single Nucleotide Polymorphism
- Why is this important?
  - Most common genetic variation
  - Can be linked to phenotype, environment, or heredity
- What is the basic workflow?



Options for generating SNPs

- Many factors go into deciding the most appropriate option
- Different levels of investments in terms of wet lab and bioinformatic
- Size of genome, number of individuals, how much of the genome do you need to sequence, and ultimate goal for analyses

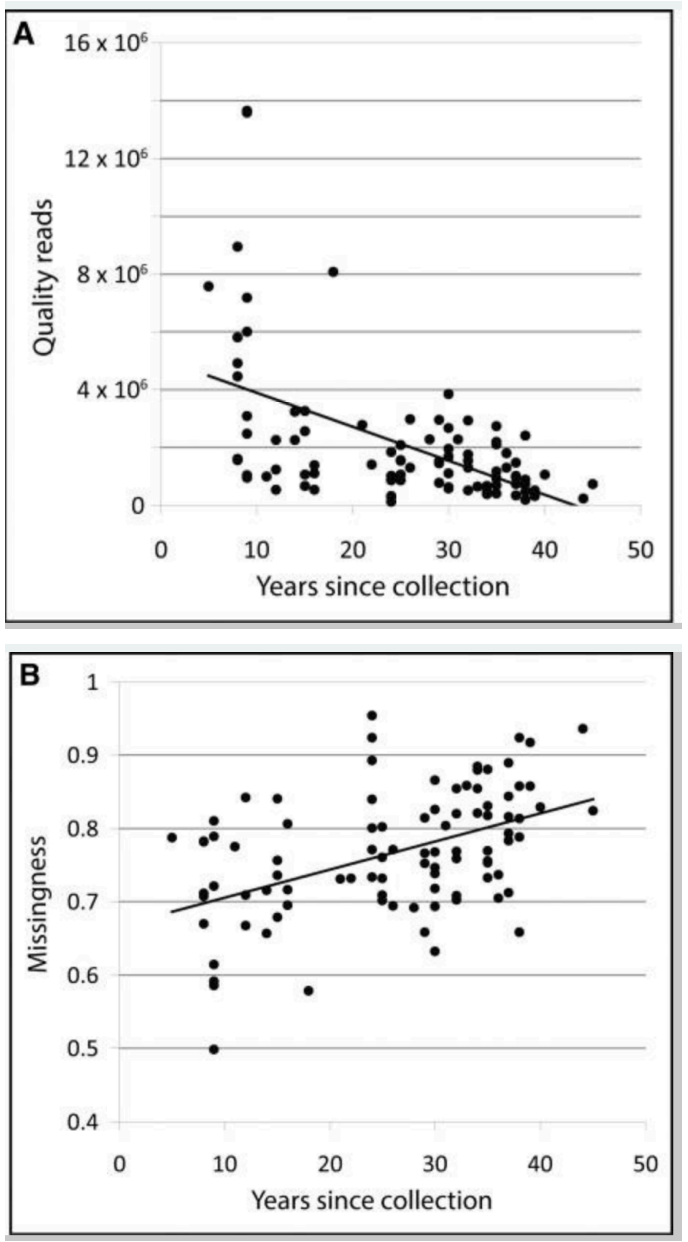
Phylogenomics approach	Genomic resources required	Initial bioinformatic investment	Ultimate bioinformatic investment	Initial laboratory cost	Ultimate cost per sample
Genome skimming	Yes	None	Medium	Low	Medium
RAD-Seq	No, but helpful	Medium	High	High	Low
RNA-Seq	No, but helpful	Low	High	Low	High
Hyb-Seq	Varies[b]	High[b]	Medium	Low[b]	Medium

Modified from Dodsworth et al., 2019

de novo RAD-Seq

- Basic CTAB or similar DNA extraction
- Lots of options for enzymes with different frequency of cut sites
- Silica dried material works great
- Herbarium samples or degraded samples can work
- iPyRad or Stacks

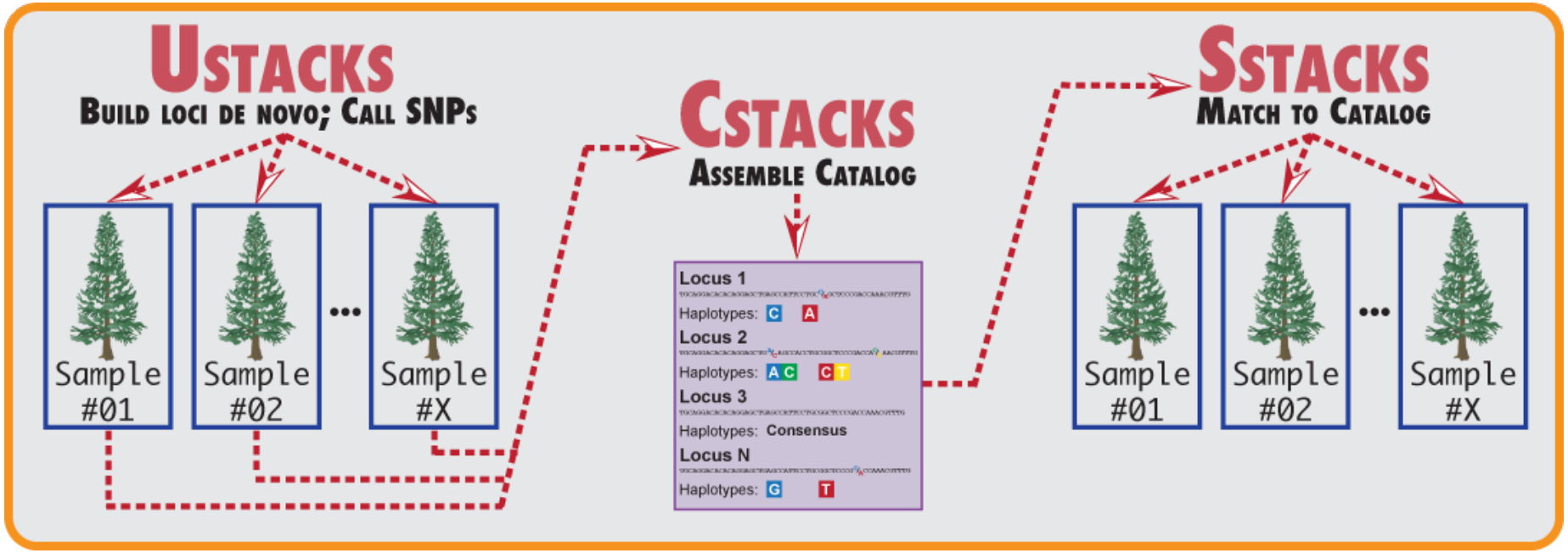
Change over time



Beck and Semple, 2015

de novo RAD-Seq

- Stacks denovo_map.pl script -> specify fastq files and population map



- Assembles loci in each individual and allows specification of number of nucleotide differences to define a locus, then assembles a catalog of all loci, then matches each sample to catalog for SNP calling

Stacks user manual

de novo RAD-Seq input

Calling the program — input — input

```
~/stacks/2.X/bin/denovo_map.pl --samples fastq_files/ --popmap population_map.txt -o de_novo_wrapper/ -T 8
```

output — additional options

Example population map - populations

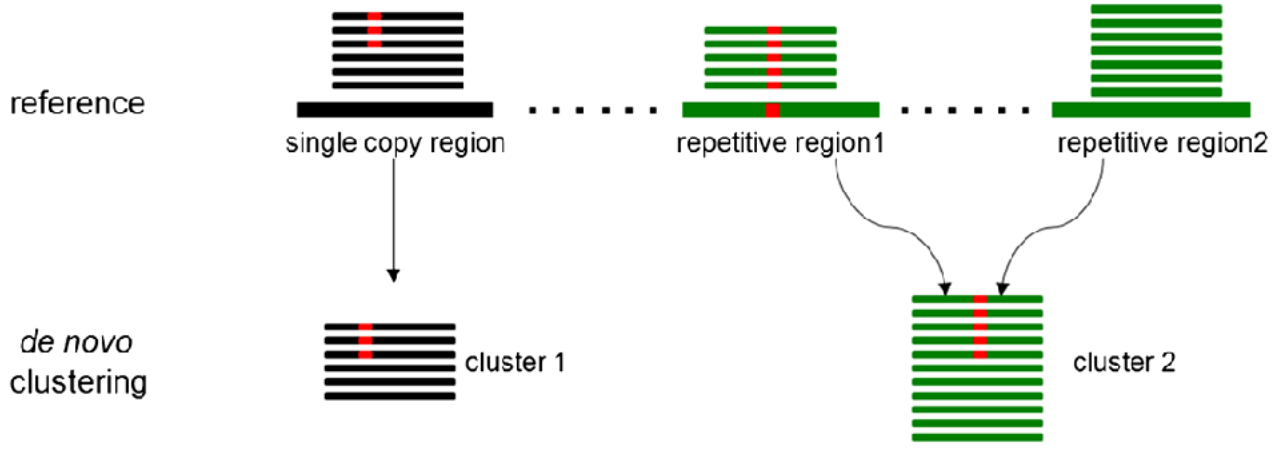
```
% more popmap
indv_01     6
indv_02     6
indv_03     6
indv_04     2
indv_05     2
indv_06     2
```

Example population map - individuals

```
LA2100  LA2100
LA2103  LA2103
LA2105  LA2105
LA2106  LA2106
LA2114  LA2114
LA2119  LA2119
LA2128  LA2128
LA2855  LA2855
```

Reference based RAD-Seq

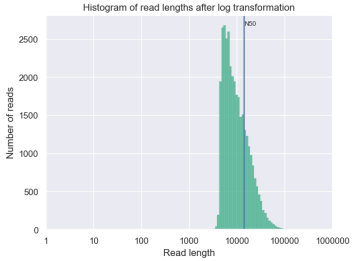
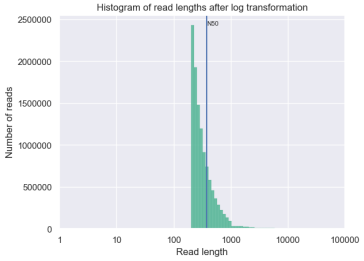
- Wet lab preparation same as for *de novo* approach
- Need to have some form of reference genome to map reads to
- Helps make sure nonhomologous loci are not collapsed

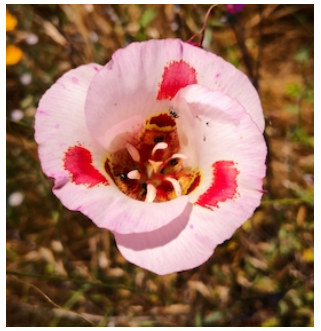


Dou et al. 2011

Does a reference genome help?

Adriana Hernandez

	de novo	Nanopore	Illumina	Hybrid
Genome Assembly		397 MB N50= 14,352 bp	3 GB N50= 220 bp	4.3 GB N50= 377 bp
				
SNPs				
Raw	5,903	50,723	203,143	258,958
Filtered	2,188	4,976	8,660	15,533

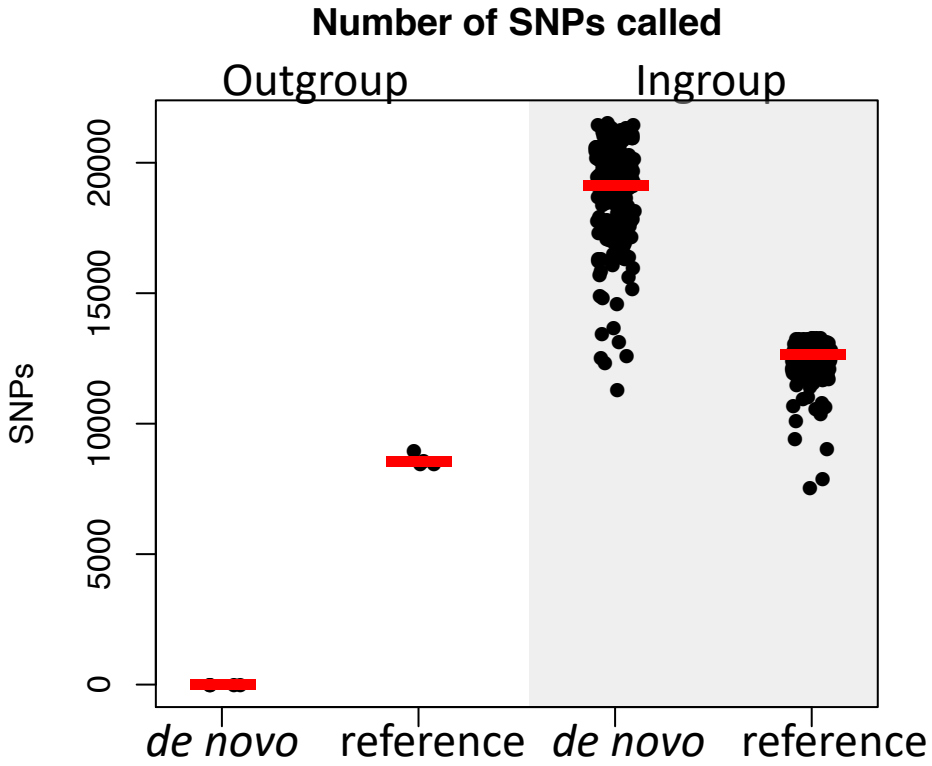


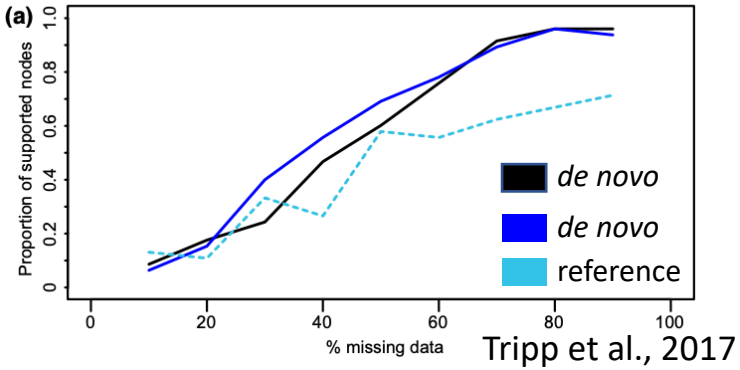
Calochortus venustus
Estimated genome size of 5.5 GB

Even a poor draft genome increases the ability to call SNPs

Does a reference genome help?

- For many analyses having an outgroup is helpful if not necessary
- If the outgroups are quite distinct genetically calling SNPs in *de novo* framework may leave them out
- Some concern that using a reference my lead to some bias





Tripp et al., 2017

Lorena Villanueva

Washingtonia filifera

How do I get a reference genome?

- Assemble your own using short- and long-read sequencing data

For a 1GB genome

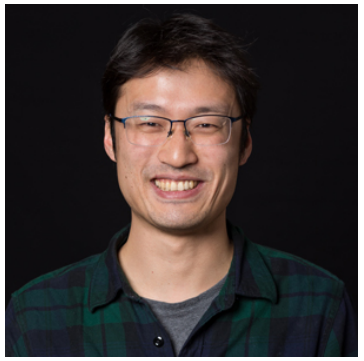
- 50X Illumina:
  - 50Gb x $26.5/Gb = **$1,325**
- 50X nanopore:
  - 50Gb x $40/Gb = **$2,000**

$3,325

- Organized a collaborative workshop covering genome assembly and annotation at Botany 2020

https://github.com/bcbc-group/Botany2020NMGWorkshop

Susan Strickler



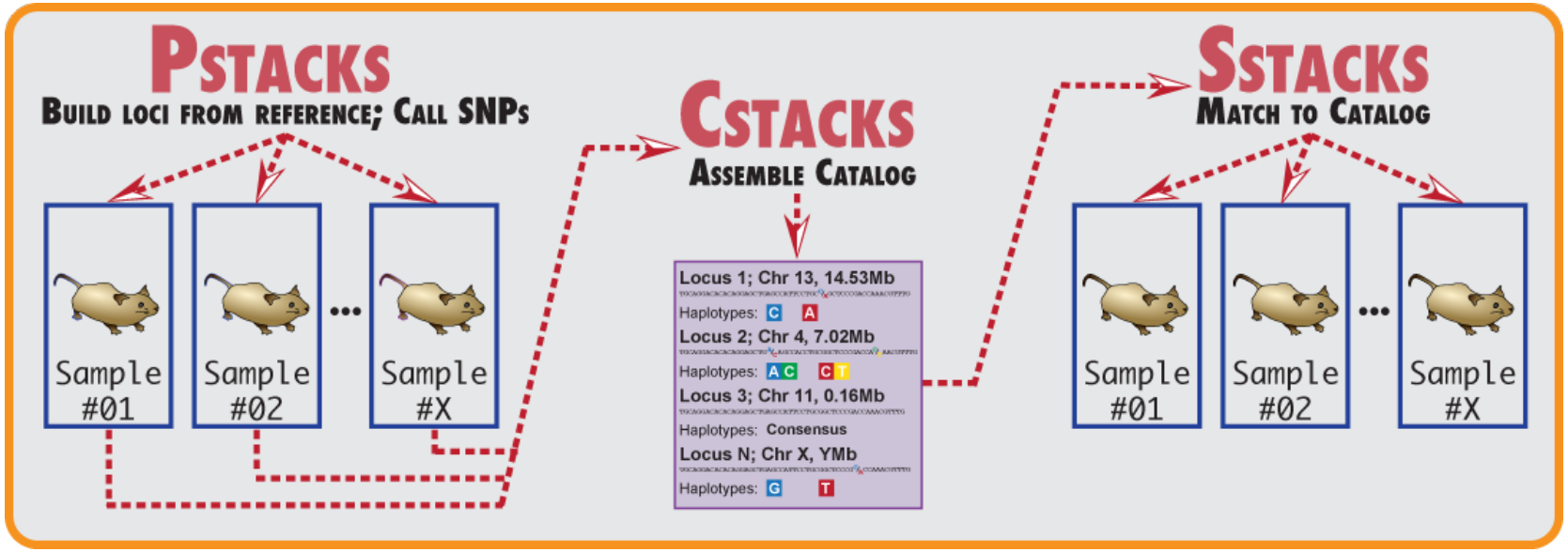
Fay-Wei Li

Andrew Nelson

CYVERSE®

Reference based RAD-Seq

- Map reads to reference
- refmap.pl -> specify bam files and population map



- Take aligned reads and calling SNPs in each locus, then make catalog and match loci based on genomic location not sequence similarity

Stacks user manual

Reference based RAD-Seq code

```
~/stacks/2.X/bin/ref_map.pl --samples sorted_bam_files/ --popmap population_map.txt -o ref_wrapper/ -T 8
```

Example population map - populations

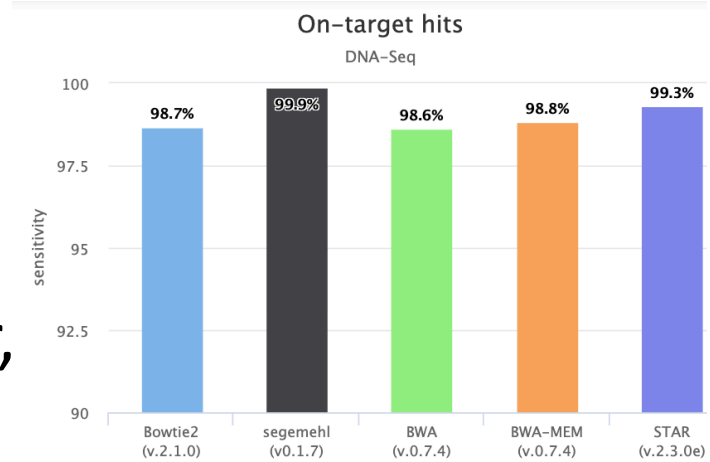
```
% more popmap
indv_01     6
indv_02     6
indv_03     6
indv_04     2
indv_05     2
indv_06     2
```

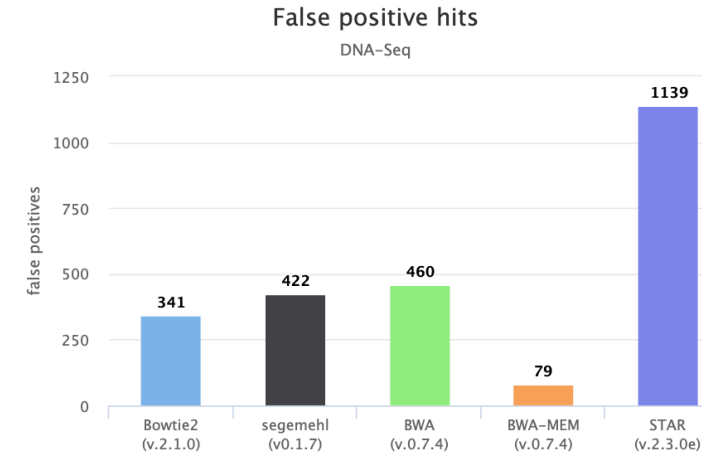
Example population map - individuals

```
LA2100  LA2100
LA2103  LA2103
LA2105  LA2105
LA2106  LA2106
LA2114  LA2114
LA2119  LA2119
LA2128  LA2128
LA2855  LA2855
```

Read mapping is often overlooked

- Many different options for mapping genomic data to a reference include BWA MEM, minimap2, bowtie, etc.
- “the portion of reads that can be mapped is one factor, but not necessarily the most appropriate one”
- BWA MEM often performs the best in comparisons
- To save computation space, convert SAM to BAM





http://merenlab.org/2015/06/23/comparing-different-mapping-software/

Otto et al., 2014

BWA MEM code

- Need to index the fasta file first to specify genetic coordinates

```
bwa index Genome_assembly.fasta
```

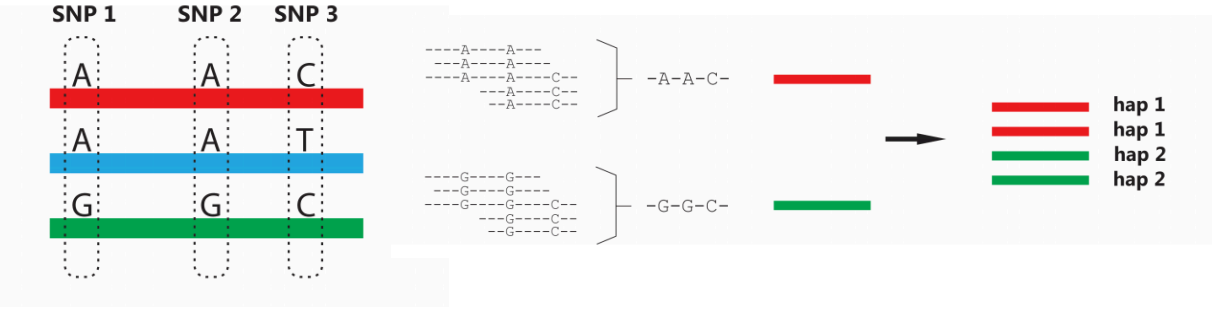
- Map reads from each sample to the reference using Read Group(RG) information for easy identification of samples
  - ID: is unique identifier of the samples
  - SM: is the sample name
  - PL: is the sequencing equipment
  - PU: is the run identifier
  - LB: is the library count

```
bwa mem -t 8 -R "@RG\tID:Sample1_A01\tSM:Sample1\tPL:HiSeq\tPU:HTNMKDSXX\tLB:RNA-Seq" Genome_assembly.fasta Sample1_R1.fastq.gz Sample1_R2.fastq.gz > Sample1.sam
```

Genome | Forward Read | Reverse Read | SAM file as output

Hyb-Seq and Genome resequencing

- No shortage in available programs or comparisons between programs
- Differences include maximum-likelihood vs Bayesian
- Haplotype vs site based



Bourke et al., 2018

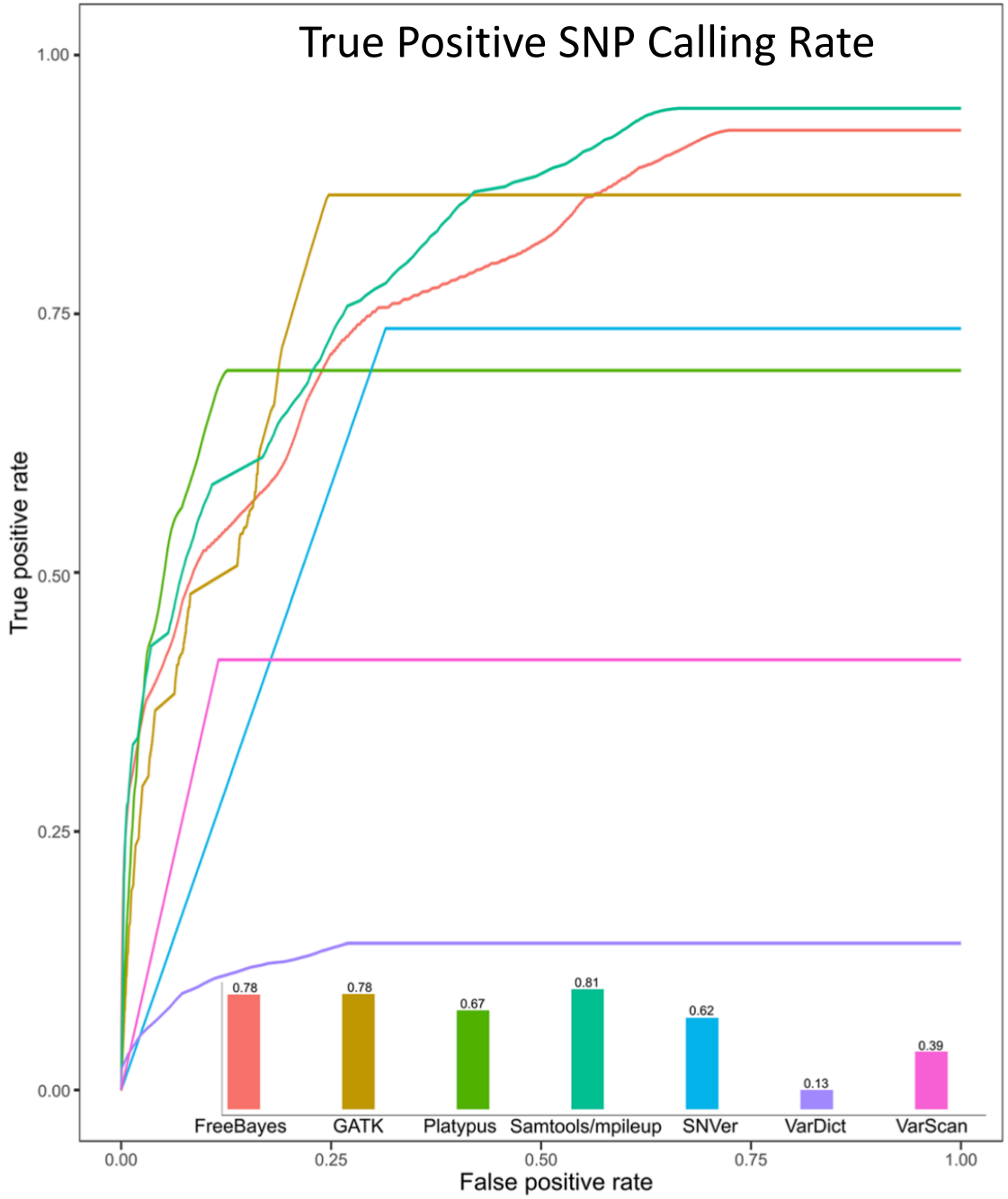
Commonly used programs

Variant tool	Version	Algorithm	Pipelines	Default filter	Reference
FreeBayes	v1.2.0–2	Haplotype-based Bayesian	FreeBayes	[b]10,[m]1	Garrison E, et al, 2012 [29]
GATK	4.0.11.0	Haplotype-based significant test	MarkDuplicates BaseRecalibrator HaplotypeCaller	[b]10,[m]20	DePristo M, et al, 2011 [27]
Platypus	0.8.1	Haplotype-based significant test	Platypus callVariants	[b]20,[m]20	Rimmer A, et al, 2014 [30]
Samtools /mpileup	1.9	Site align-based gt likelihoods	Samtools/mpileup bcftools call	[b]13,[m]0	Li H, 2011 [28]
SNVer	0.5.3	Site align-based MAF *p*-value	SNVerIndividual	[b]17,[m]20 [f]0.25,[r]1,[p]0.05	Wei Z, et al, 2011 [31]
VarScan	v2.3.9	Site-based allele frequency	Samtools/mpileup mpileup2snp	[b]15,[m]0 [f]0.2,[r]2,[p]0.01	Koboldt D, et al, 2012 [33]
VarDict	2018	Site-based alleles Fisher's	VarDict var2vcf_valid	[b]22.5,[m]0 [f]0.01,[r]2	Lai Z, et al, 2016 [32]

Yao et al., 2020

Which SNP caller to use?

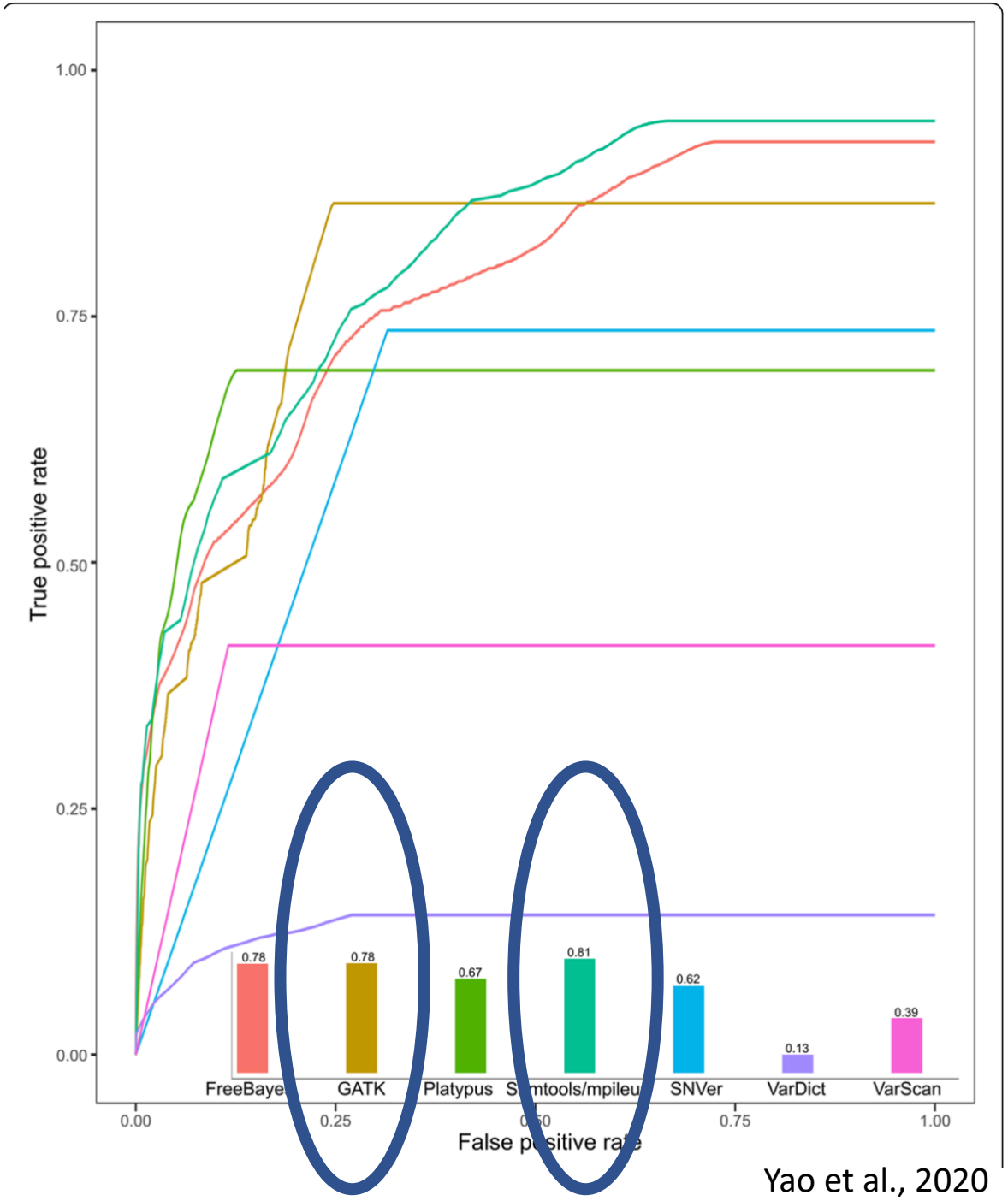
- **All SNP callers are NOT created equal**
- FreeBayes, GATK, and Samtools/mpileup had the lowest number of missed calls
- FreeBayes, VarScan and VarDict were most sensitive to unique calls
  - High sensitivity could result in a higher false positive rate
- Testing for true positives Samtools/mpileup called 81%, while GATK called 78.1% and FreeBayes called 77.7%



Yao et al., 2020

Which SNP caller to use?

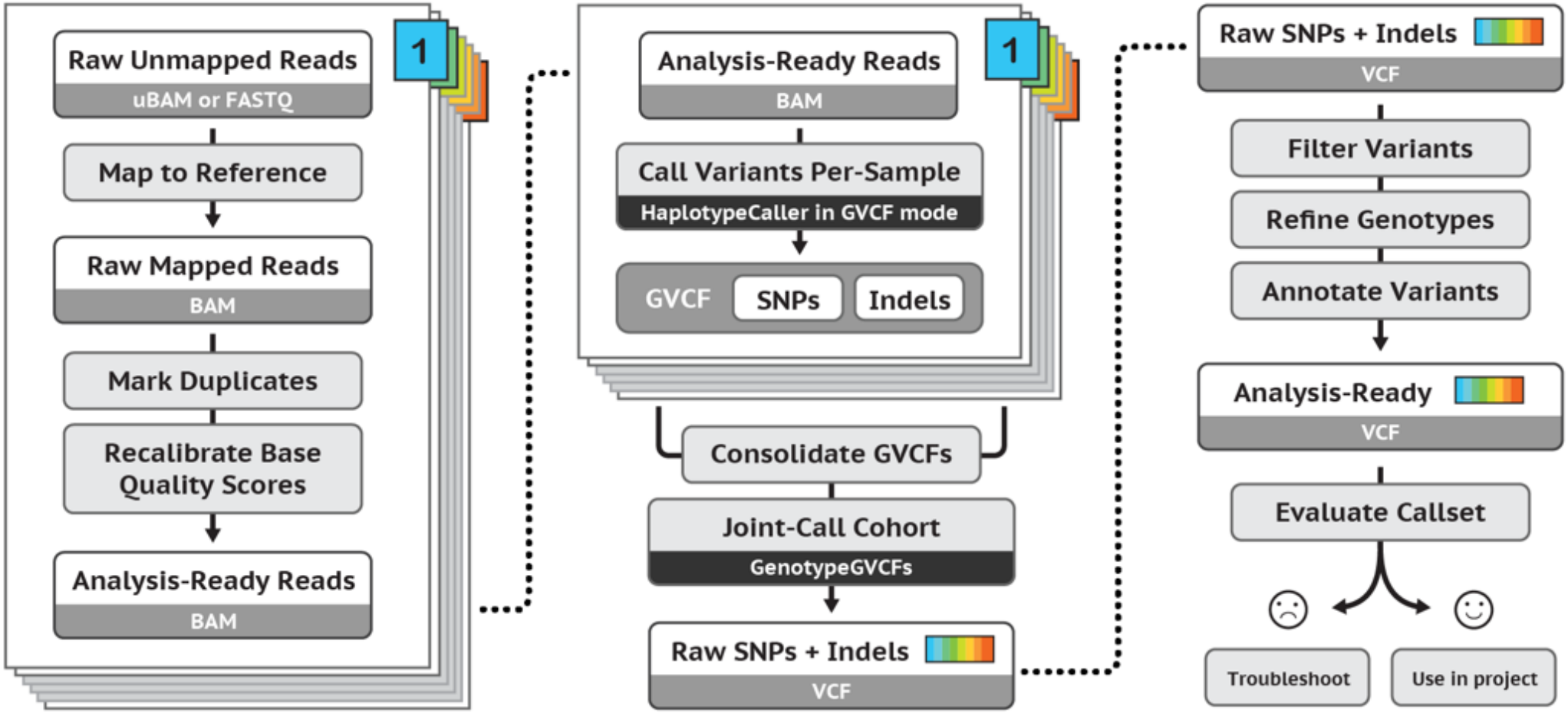
- **All SNP callers are NOT created equal**
- In many comparisons BWA MEM + GATK found to be the best for most genomes
- For complex genomes such as the large, polyploid wheat genome, BWA MEM + Samtools/mpileup is recommended



Yao et al., 2020

GATK

- GATK Best Practices: https://gatk.broadinstitute.org/hc/en-us/sections/360007226651-Best-Practices-Workflows



GATK code

- Prep the reference similar to how we did for BWA MEM

```
gatk CreateSequenceDictionary -R Genome_assembly.fasta -O Genome_assembly.dict
```

```
samtools faidx Genome_assembly.fasta
```

- Each sample that was mapped to the genome will need to be indexed then call SNPs and indels via local re-assembly of haplotypes

```
samtools index Sample1.bam
```

```
gatk HaplotypeCaller -R Genome_assembly.fasta -I Sample1.bam -O Sample1.g.vcf.gz -ERC GVCF
```

GATK code continued

- We technically have now called SNPs on each sample but only the variants for each sample individually
- We want a file representing all individuals and all variants
- Need to combine the files and the do joint genotyping

```
gatk CombineGVCFs -R Genome_assembly.fasta -V samples.list --output
All_samples_combined.g.vcf.gz
```

```
gatk GenotypeGVCFs -R Genome_assembly.fasta --variant
All_samples_combined.g.vcf.gz --output All_samples_variants.vcf.gz
```

Resulting file - Variant Call Format

Formatting and info about what is included for each score

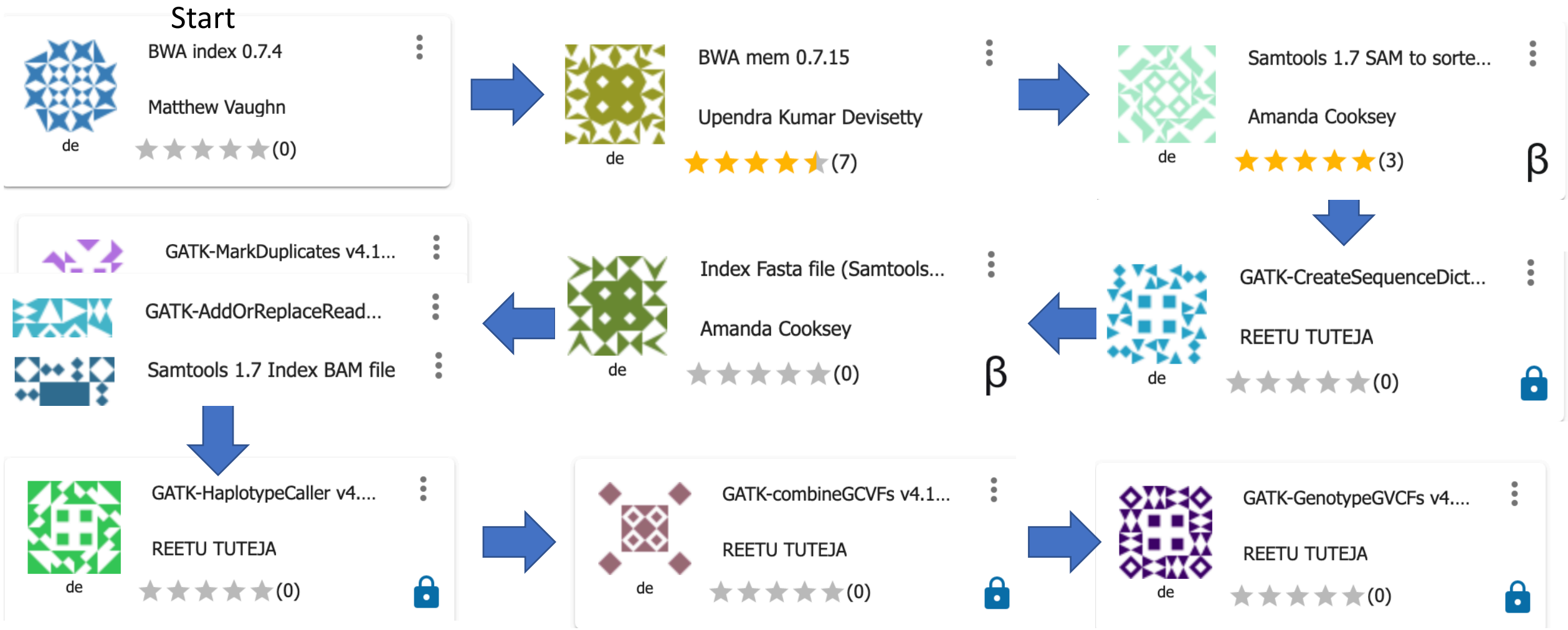
Each contig and how big they are

Each line is a variant, each column is a sample

```
##fileformat=VCFv4.2
##ALT=<ID=NON_REF,Description="Represents any possible alternative allele at this location">
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=MIN_DP,Number=1,Type=Integer,Description="Minimum DP observed within the GVCF block">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">##INFO=<ID=A
##INFO=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth; some reads may have been filtered">
##INFO=<ID=DS,Number=0,Type=Flag,Description="Were any of the samples downsampled?">
##INFO=<ID=END,Number=1,Type=Integer,Description="Stop position of the interval">
##INFO=<ID=RAW_MQandDP,Number=2,Type=Integer,Description="Raw data (sum of squared MQ and total depth) for improved RMS Mapping
##INFO=<ID=ReadPosRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt vs. Ref read position bi
##contig=<ID=Chr1,length=43270923>
##contig=<ID=Chr2,length=35937250>
##contig=<ID=Chr3,length=36413819>
##contig=<ID=Chr4,length=35502694>
##contig=<ID=Chr5,length=29958434>
##contig=<ID=Chr6,length=31248787>
##contig=<ID=Chr7,length=29697621>
##contig=<ID=Chr8,length=28443022>
##contig=<ID=Chr9,length=23012720>
##contig=<ID=Chr10,length=23207287>
##contig=<ID=Chr11,length=29021106>
##contig=<ID=Chr12,length=27531856>
##contig=<ID=ChrUn,length=633585>
##contig=<ID=ChrSy,length=592136>
##source=CombineGVCFs
##source=GenotypeGVCFs
##source=HaplotypeCaller
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  Arpashali_S242  Ceenova_S243    Marakissa_S241  Rice_Plate5_A01_19b Ri
ChrSy   1   .   T   .   0.01    LowQual DP=6    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
ChrSy   3   .   C   .   0.01    LowQual DP=6    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
ChrSy   4   .   T   .   0.01    LowQual DP=6    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
ChrSy   5   .   A   .   0.01    LowQual DP=6    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
ChrSy   6   .   G   .   0.03    LowQual DP=9    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
ChrSy   7   .   A   .   0.03    LowQual DP=9    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
ChrSy   8   .   T   .   0.03    LowQual DP=9    GT:AD:DP:RGQ    0/0:2,0:2:6 ./.:0,0:0   ./.:0,0:0   ./.:0,0:0   ./
```

More info: https://samtools.github.io/hts-specs/VCFv4.2.pdf

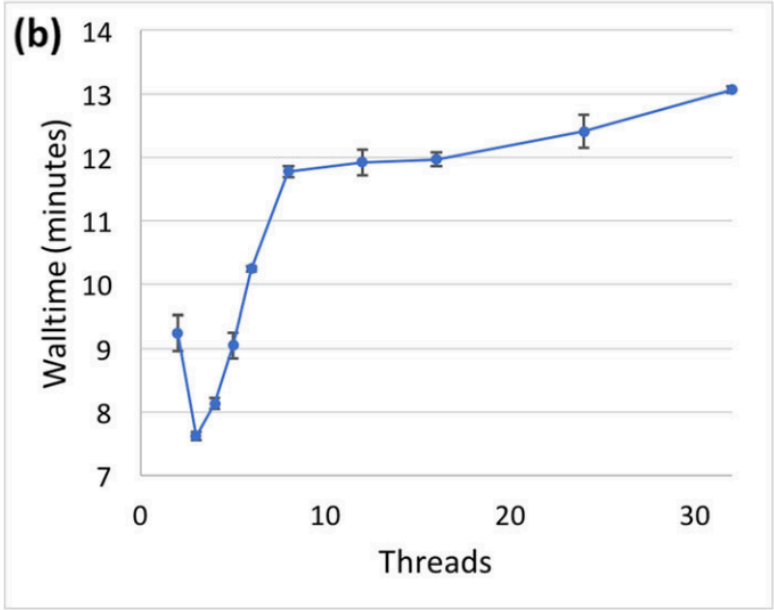
GATK CyVerse Discovery Environment



- Limited resources: 8 CPU and 16 GB RAM

Possible issues with GATK

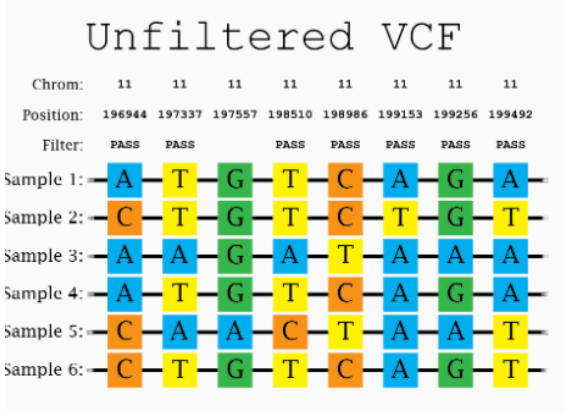
- Can be a difficult program to learn, however there is an extensive and active discussion board and tutorials available
- Scalability – Using more threads/processors doesn't always speed up analyses
- Version issues are real
  - When updates come out, some commands change with little documentation
  - Need to look at the updated tutorials from the Broad Institute

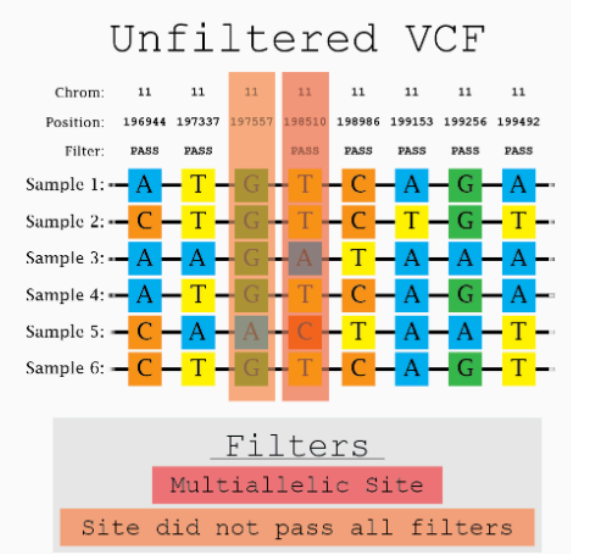


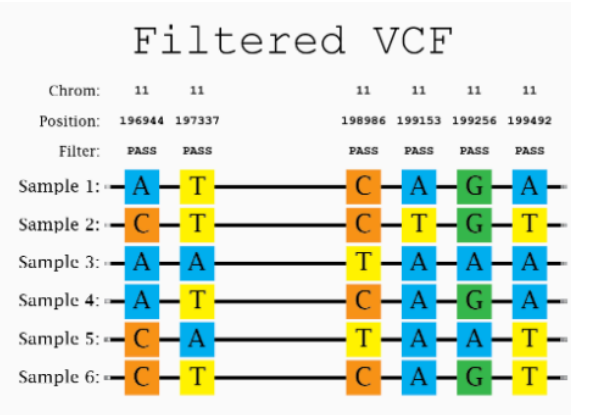
Heldenbrand et al., 2019

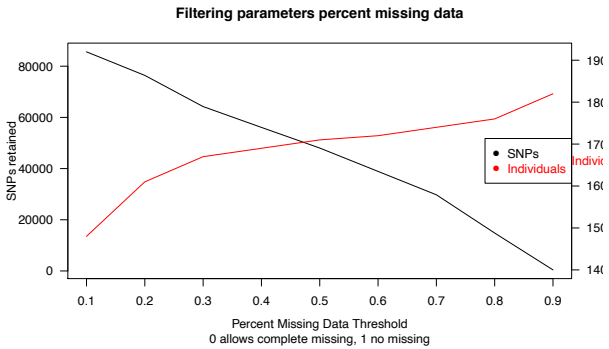
Filtering data

- VCFtools
  - Easy to implement; not very picky on specific formatting
  - Limited to options, but a clear user manual
  - Can be slow on large data sets (hundreds of taxa and millions of SNPs)
  - Cannot handle polyploid data
- BCFtools
  - Harder to implement for basic filtering, but more powerful
  - Much faster with large data sets and can handle polyploid data
  - Actively supported and distributed alongside Samtools
- GATK methods: https://gatk.broadinstitute.org/hc/en-us/articles/360035890471-Hard-filtering-germline-short-variants









VCFtools code

- If you wanted to keep only sites that were biallelic sites, at most 50% missing data, a read depth between 3-30x coverage, and a minor allele frequency of at least 5%

vcftools --vcf original.snps.vcf --max-missing 0.5 --min-alleles 2 --max-alleles 2 --min-meanDP 3 --max-meanDP 30 --maf 0.05 **--recode --recode-INFO-all** --out Filtered_SNPs

- Also very easy to in VCFtools to report read depth for each individual, percent of missing data, and heterozygosity values

vcftools --vcf Filtered_SNPs.recode.vcf –depth
vcftools --vcf Filtered_SNPs.recode.vcf –missing-indiv
vcftools --vcf Filtered_SNPs.recode.vcf --het

GitHub tutorial with *U. gibba*

- SNP calling walkthrough available: https://github.com/bcbc-group/CyVerese_SNP_calling_webinar
- Incorporates publicly available data using a high-quality genome assembly and RNA-Seq data for multiple organ types
  - Bladder, leaf, rhizoid, and stem
- Small data set that can be run on a local machine
- Examples for command line and Discovery Environment
- SNP calling using both Stacks and GATK
- Filtering and PCA using SNP data

Utricularia gibba
Humped bladderwort

Conclusions

- Every project may demand a modified SNP calling approach
- Things that may influence your methods may be large genomes, polyploidy events, availability and quality of a reference genome
- SNP filtering in some ways is an art; each data set should be explored to see what happens when adjusting parameters
- Hopefully this is a good start on the SNP calling journey but there are many intricacies to each of these programs along the way

CyVerse is supported by the National Science Foundation under Grants No. DBI-0735191, DBI-1265383 and DBI-1743442.

THE UNIVERSITY OF ARIZONA®

TACC
TEXAS ADVANCED COMPUTING CENTER

CSH Cold Spring Harbor Laboratory