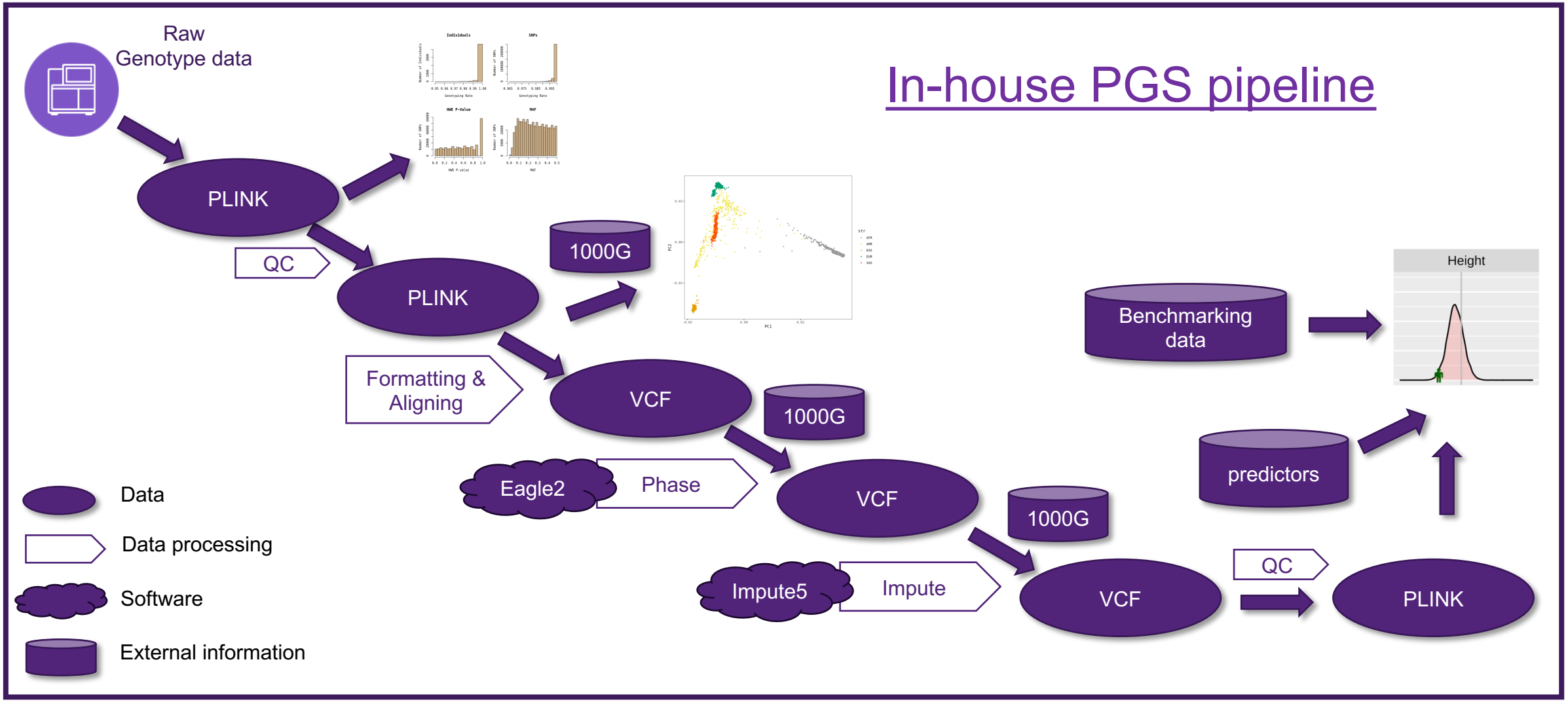
An in-house PGS pipeline

Genetics & Genomics Winter School

Module 5

schematic of technical pipeline



Genotype data from arrays

- Can assay ~1M SNPs per individual with ‘SNP chips’
- Data is typically ‘counts’ of a reference allele

genotype file:

	SNP1	SNP2	SNP3	SNP4
Bob	0	1	0	1
Fred	1	2	0	0
Jose	1	2	2	2
Andy	2	1	1	1

map file:

	chr	position	ref	alt
SNP1	1	52196307	A	T
SNP2	1	52462094	C	T
SNP3	1	52736008	A	G
SNP4	1	53010891	T	C

[borrowed from Keth]

Why a raw data is not ready for PGS profiling?

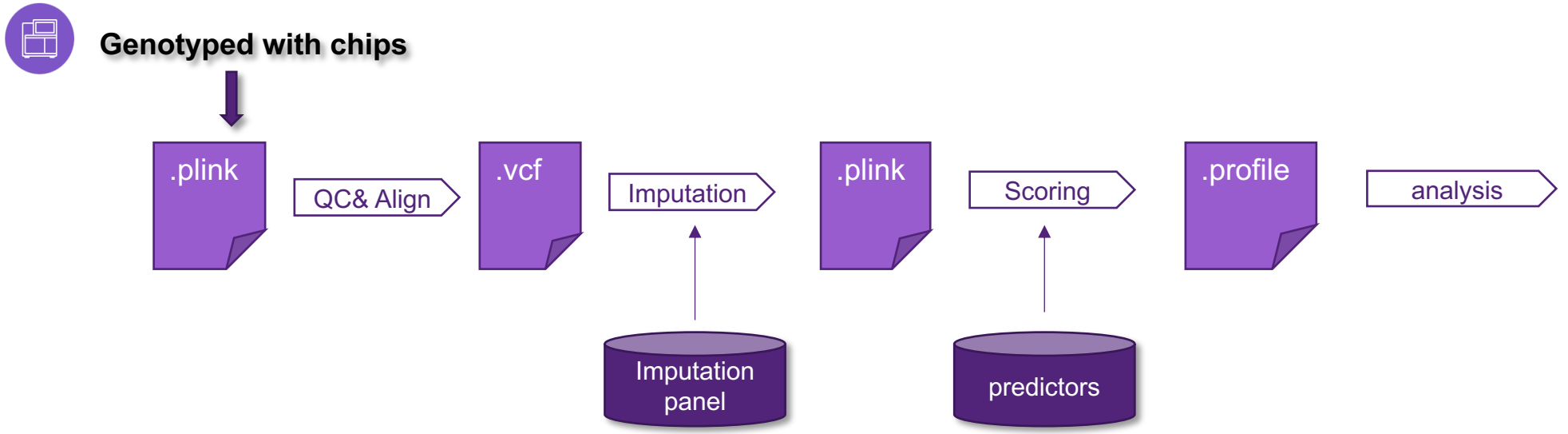
- Quality

- Coverage

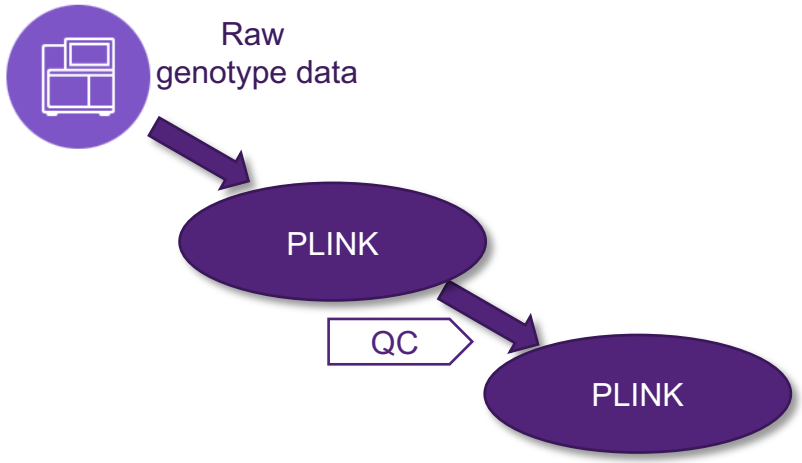
A high density SBayesRC Predictor – 7.3M SNPs

	Number of Nucleotide/Variants
Whole human genome haplotype	3 billion
TopMed	445 million
1000G	80 million
HRC	40 million
HapMap3	1 million
Illumina GSA chip	654 thousand

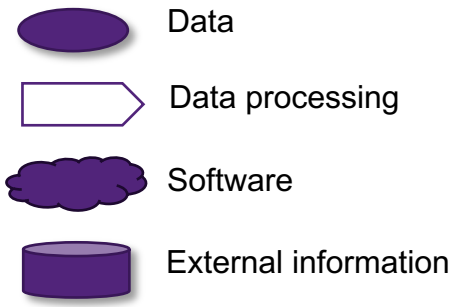
Overview of PGS pipeline



schematic of technical pipeline



In-house PGS pipeline



Revisit Genotype data QC for a GWAS study

- Per Individual QC
  - *1) removal of individuals with excess missing genotypes*
  - *2) removal of individuals with outlying homozygosity values*
  - *3) remove of samples showing a discordant sex*
  - *4) removal of related or duplicate samples, and*
  - *5) removal of ancestry outliers*

- Per SNP QC
  - *1) removal of SNPs with excess missing genotypes*
  - *2) removal of SNPs that deviate from Hardy-Weinberg equilibrium*
  - *3) removal of SNPs with low minor allele frequency*
  - *4) comparing allele frequency to known values*

Extra consideration in our practice

- Large number of SNPs with MAF = 0
  - Missing Alleles

- Replicates and relatives can exist

Genotype data QC

- Per Individual QC
  - *1) removal of individuals with excess missing genotypes*
  - *2) removal of individuals with outlying homozygosity values*
  - *3) remove of samples showing a discordant sex*
  - *4) removal of related or duplicate samples, and*
  - *5) removal of ancestry outliers*

- Per SNP QC
  - *1) removal of SNPs with excess missing genotypes*
  - *2) removal of SNPs that deviate from Hardy-Weinberg equilibrium*
  - *3) removal of SNPs with low minor allele frequency*
  - *4) comparing allele frequency to known values*

Extra consideration in practice

- Large number of SNPs with MAF = 0
  - Missing Alleles
- Replicates and relatives can exist
- Different genome build between raw data and imputation panel

Human Genome Assemblies

https://hgdownload.soe.ucsc.edu/downloads.html

Very similar / Same.
GRCh37 names them `chr1`, `chr2`, `chr3`, etc, while hg19 just has `1`, `2`, `3`.
Different Mitochondria contigs.

Human genomes

Jan. 2022 (T2T-CHM13 v2.0/hs1)

- Fileserver (bigBed, maf, fa, etc) annotations
- Standard genome sequence files and select annotations (2bit, GTF, GC-content, etc)
- LiftOver files
- Pairwise alignments

Telomere-to-Telomere

A haploid human genome without gaps

Dec. 2013 (GRCh38/hg38)

- Genome sequence files and select annotations (2bit, GTF, GC-content, etc)
- Sequence data by chromosome
- Annotations
- SNP-masked fasta files
- LiftOver files
- Pairwise alignments
- Multiple alignments
- Patches
- Data archive

hg19ToHg38.over.chain.gz

hg38ToHg19.over.chain.gz

hs1ToHg38.over.chain.gz

hs1ToHg19.over.chain.gz

hg38ToHs1.over.chain.gz

Feb. 2009 (GRCh37/hg19)

- Genome sequence files and select annotations (2bit, GTF, GC-content, etc)
- Sequence data by chromosome
- Annotations
- GC percent data
- Protein database for hg19
- SNP-masked fasta files
- LiftOver files
- Pairwise alignments (primates)
- Pairwise alignments (other mammals)
- Pairwise alignments (other vertebrates)
- Multiple alignments
- Patches
- Data archive

Mar. 2006 (NCBI36/hg18)

- Data and annotations

GRCh = Genome Reference Consortium Human Build

Liftover plink files

Best solution: recommend realigning the manifest files with BCFtools/gtc2vcf (http://github.com/freeseek/gtc2vcf)

Option 1. https://www.strand.org.uk

- update_build.sh <bed-file-stem> <strand-file> <output-file-stem>

Option 2. https://genome.sph.umich.edu/wiki/LiftOver

- python liftMap.py -m data_recoded.map -p data_recoded.ped -o data_recoded_lifted

Option 3. LiftOverPlink

- https://github.com/sritchie73/liftOverPlink/blob/master/README.md

Option 4. use reference file to update dbSNP locations in bim file or GWAS statistics

- Hg38 dbsnp_146.hg38.vcf.gz
- Hg19 dbsnp_138.hg19.vcf.gz

- Available: https://gatk.broadinstitute.org/hc/en-us/articles/360035890811-Resource-bundle

Extra consideration in practice

- Large number of SNPs with MAF = 0
  - Missing Alleles
- Replicates and relatives can exist
- Different genome build between raw data and imputation panel
- SNPs alleles from negative strand

Chromosomes, strands and SNP alleles

Paternal chromosome

Positive strand

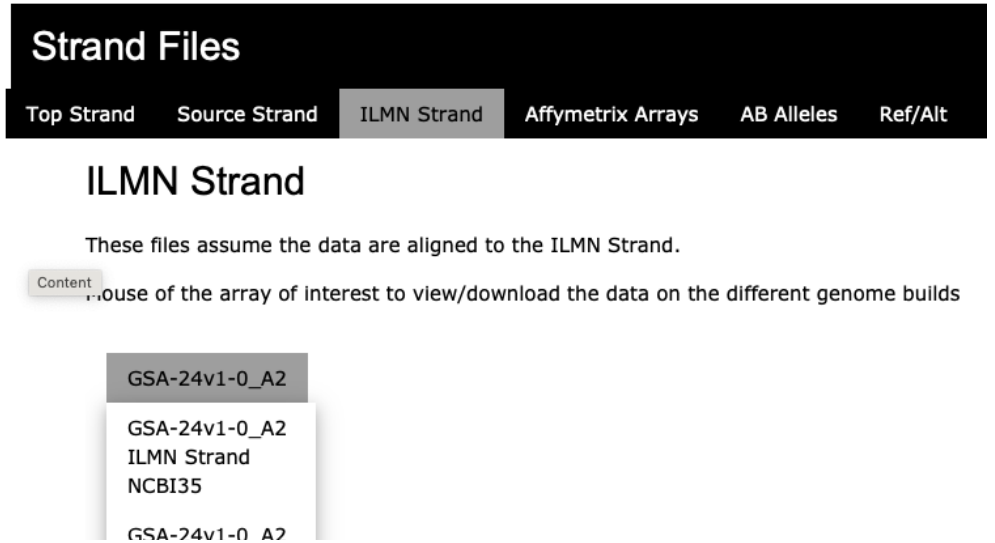
Negative strand

Flip the strand

Maternal chromosome

Strand resource

https://www.strand.org.uk



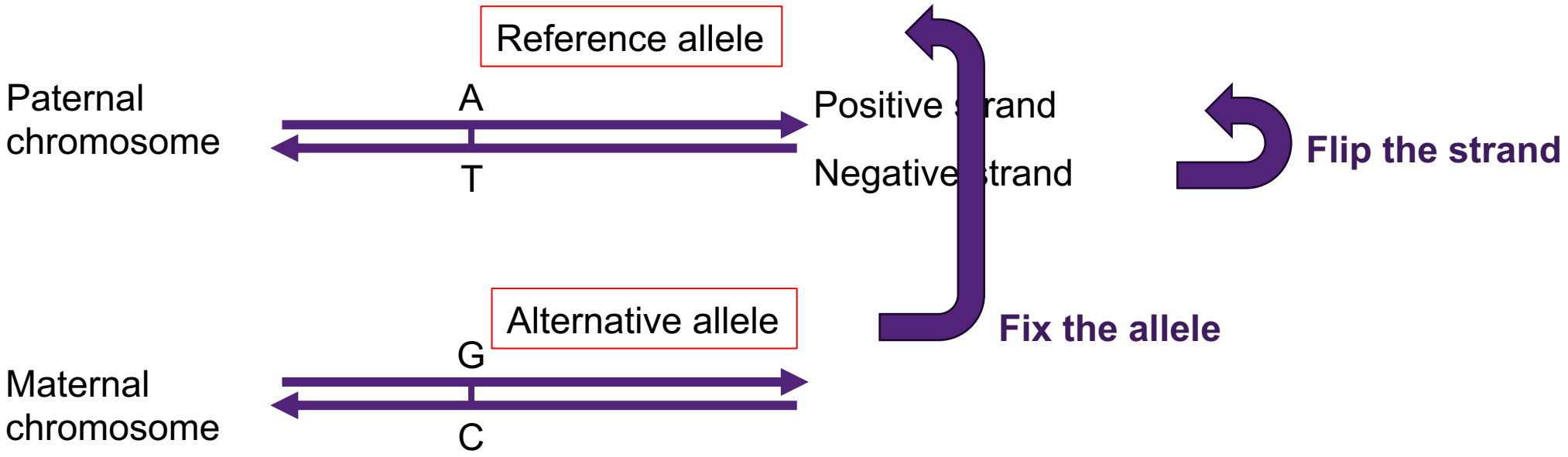
Usage is:
update_build.sh <bed-file-stem> <strand-file> <output-file-stem>
where:

<bed-file-stem>	is the name of your binary ped set minus the .bed, .bim or .fam extension
<strand-file>	is appropriate strand file for you chip and current strand orientation (TOP, SOURCE, ILMN)
<output-file-stem>	is the name of the new output file to create again minus the .bed, .bim or .fam extension

GSA-24v3-0_A2

GSAMD-24v1-0_20011747_A1

Chromosomes, strands and SNP alleles

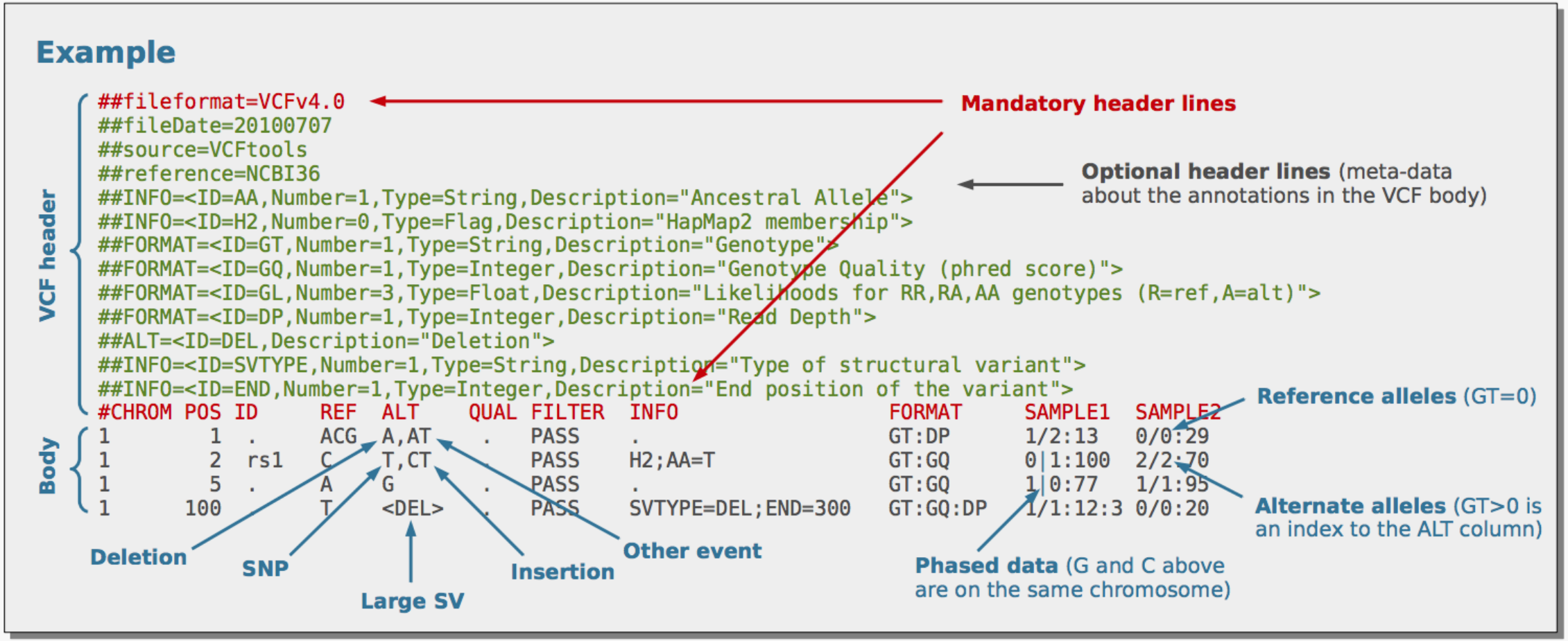


Example script to fix ref allele

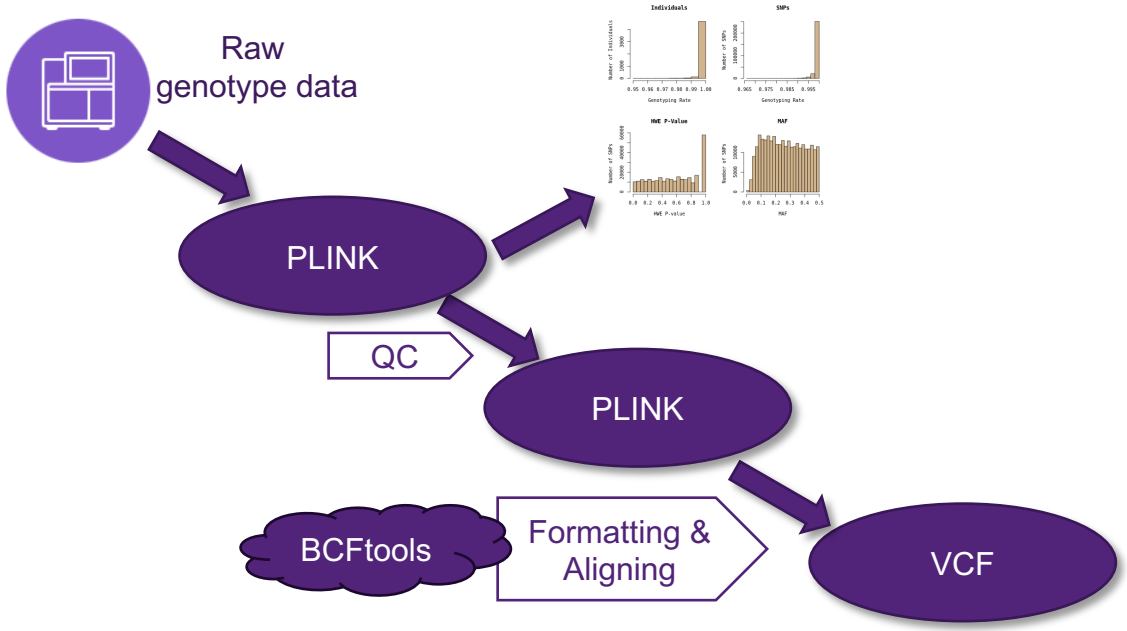
```
chr=22

# Pull out data for relevant chromosome and convert to VCF.
plink --bfile ${data}_chr${chr}  --recode vcf --out  ${data}_chr${chr}

# Sort and compress the VCF file
vcf-sort ${data}_chr${chr}.vcf | bgzip -c > ${data}_chr${chr}.vcf.gz

# Fix the reference allele to match the GRCh37 reference fasta (human_g1k_v37.fasta).
ref2fix=${refpath}/human_g1k_v37.fasta
BCFTOOLS_PLUGINS=/software/bin/
bcftools \
  +fixref \
  ${data}_chr${chr}.vcf.gz \
  -Oz \
  -o fixed_${data}_chr${chr}.vcf.gz  \
  -- -d \
  -f ${ref2fix} \
  -m flip

zcat fixed_${data}_chr${chr}.vcf.gz | bgzip -c > indexed_fixed_${data}_chr${chr}.vcf.gz

# create index file.
tabix indexed_fixed_${data}_chr${chr}.vcf.gz
```

BCFtools
VCFtools

Example VCF files

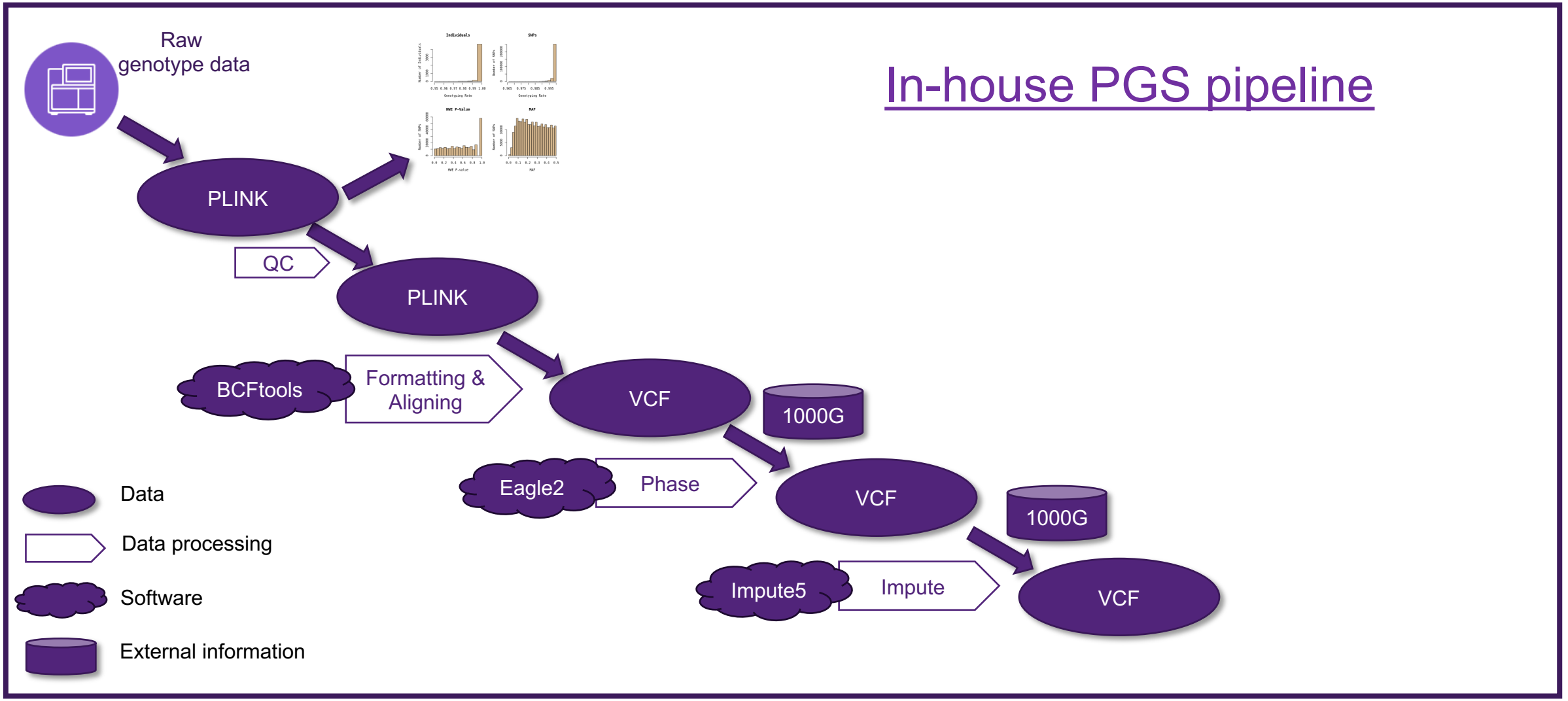


schematic of technical pipeline



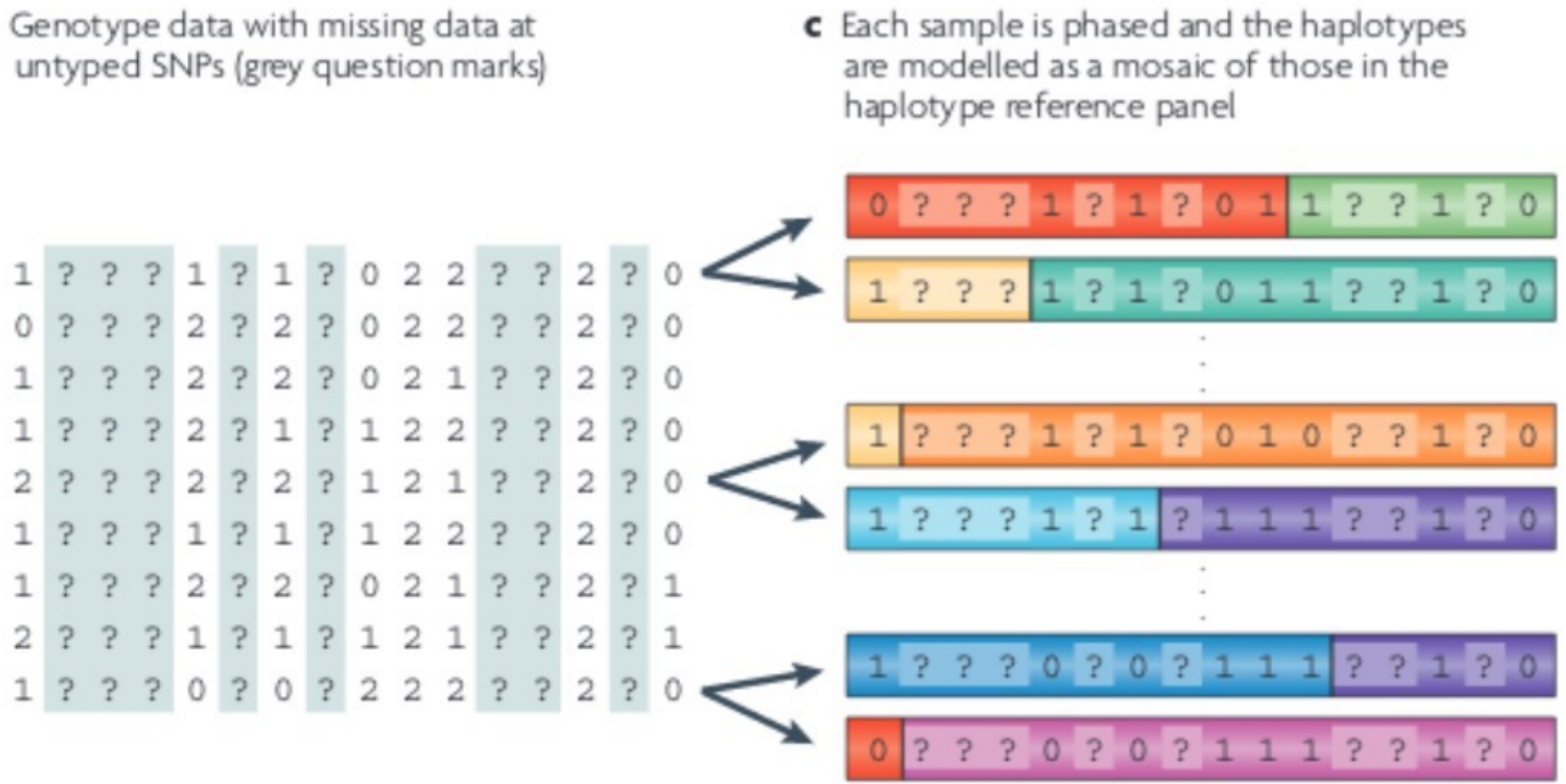
In-house PGS pipeline

schematic of technical pipeline



In-house PGS pipeline

phasing


a Genotype data with missing data at untyped SNPs (grey question marks)
c Each sample is phased and the haplotypes are modelled as a mosaic of those in the haplotype reference panel

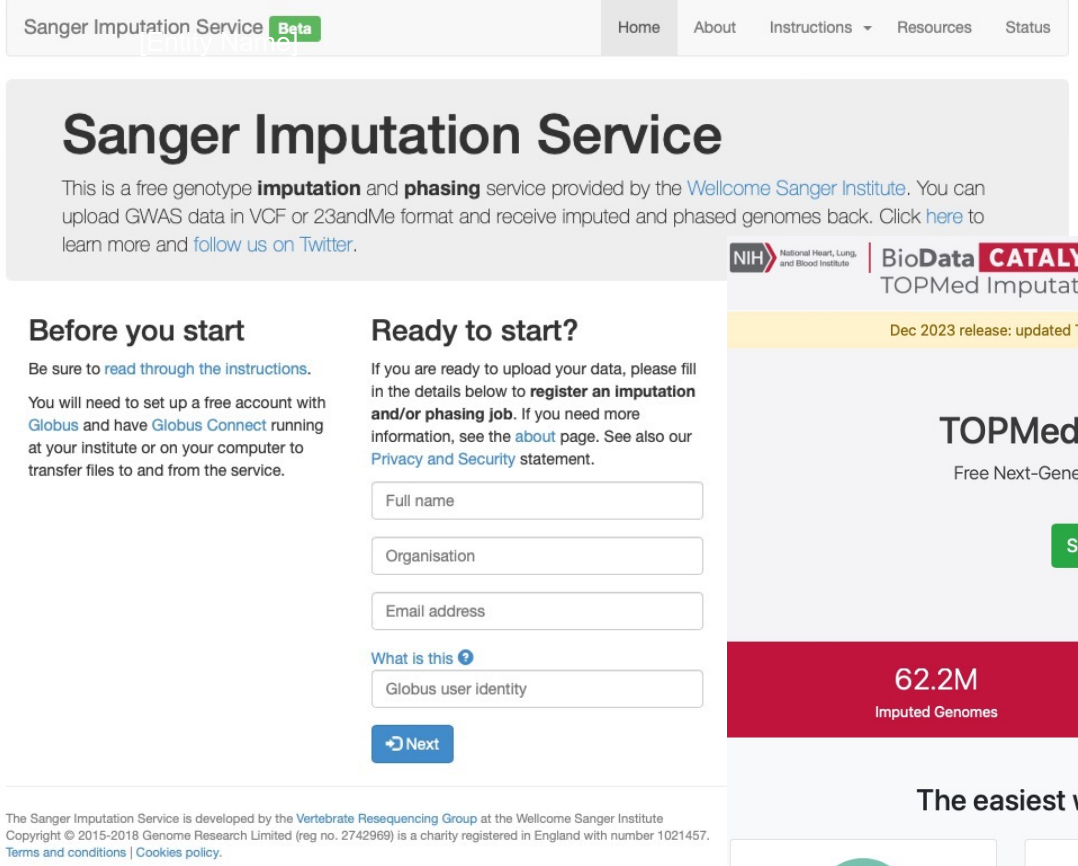
Imputation

Each sample is phased and the haplotypes are modelled as a mosaic of those in the haplotype reference panel

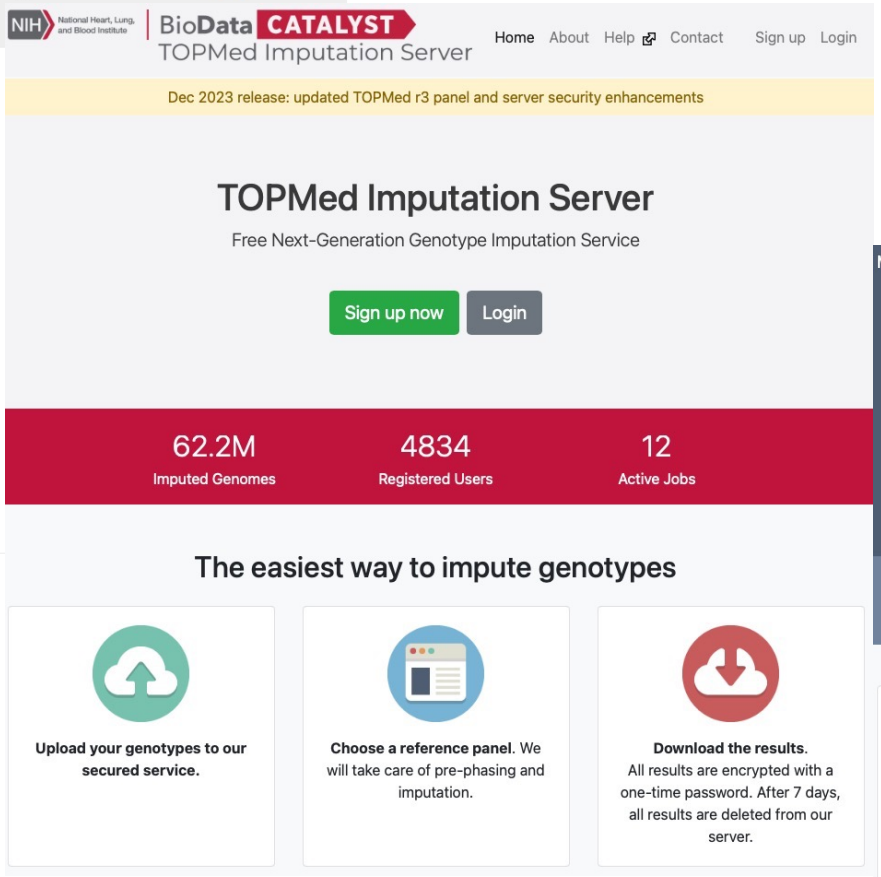
d Reference set of haplotypes, for example, HapMap

Panel options

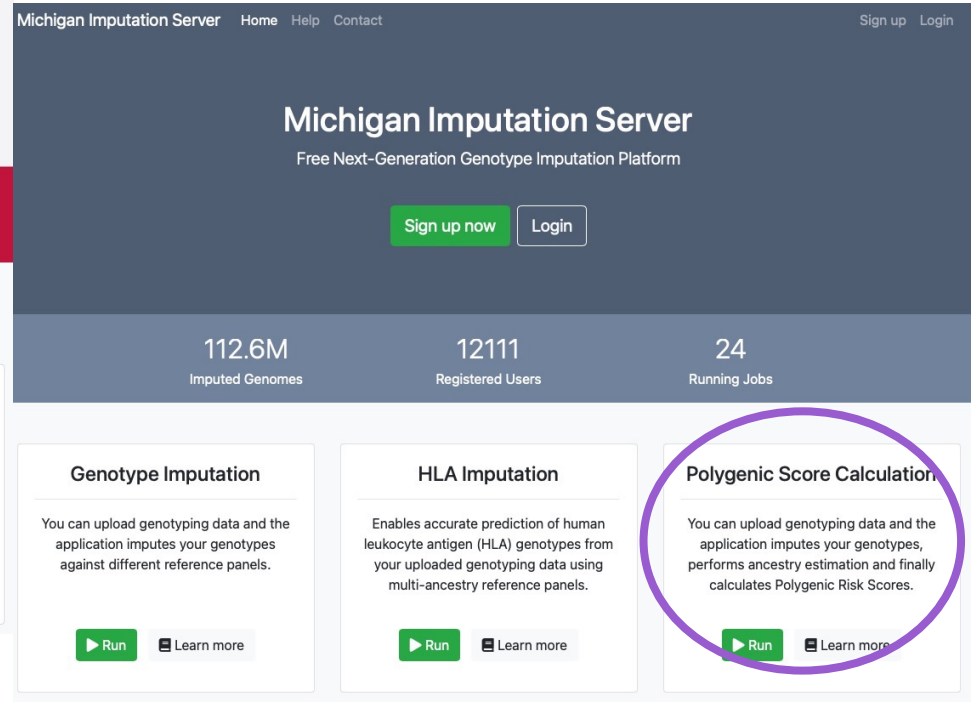
Reference Panel	Number of Individuals	Number of Variants	Population Focus
Haplotype Reference Consortium (HRC)	~32,000	~40 million	European
1000 Genomes Project	2,504	~88 million	Global, diverse
TOPMed	~62,000	>300 million	Diverse, underrepresented
UK10K	~3,800	~30 million	UK, European
GoT2D	~2,657	~20 million	Type 2 Diabetes, Metabolic



Most user friendly



Largest panel



Multiple features

Imputation servers

In house Phasing with Eagle2

```
## Example script
geneticmap=genetic_map_chr${chr}_combined_b37.txt
reference=ALL.chr${chr}.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz

eagle \
 --vcfRef=$reference \
 --vcfTarget=indexed_fixed_${data}_chr${chr}.vcf.gz \
 --geneticMapFile=$geneticmap \
 --vcfOutFormat=z \
 --outPrefix=phased_chr${chr} > phasing.log
```

Alternative: SHAPEIT4

In house Imputation with Impute5

```
## Example script
impute5_1.1.5_static \
 --m $geneticmap \
 --h $reference \
 --g phased_chr${chr}.vcf.gz \
 --r ${chr}:${intstart}-${intend} \
 --ne 20000 \  ## effective sample size, default 10k~20k for human
 --threads 1 \
 --o imputed_chr${chr}_chunk.vcf.gz \
 --l imputed_chr${chr}_chunk.log
```

A chromosome can be imputed as chunks

After imputation

- Format

- Imputed data is output as a zipped VCF file. We usually change the format back to PLINK for following analysis.

Example script

```
## convert the format using plink
plink --vcf  imputed_chr${chr}.vcf.gz   \
            --id-delim '_'  \
            --keep-allele-order \
            --make-bed \
            --out  imputed_chr${chr}
```

After imputation

- Format

- Imputed data is output as a zipped VCF file. We usually change the format back to PLINK for following analysis.

- SNP ID

- Plink does not like duplicate and missing IDs.
  - Fill in dbSNP ID if it's not used, as in files from Michigan and TopMed server
  - Replace missing SNP IDs with "chr_pos"
  - Rename duplicate SNP IDs with "_dup"

After imputation

- Format

- Imputed data is output as a zipped VCF file. We usually change the format back to PLINK for following analysis.

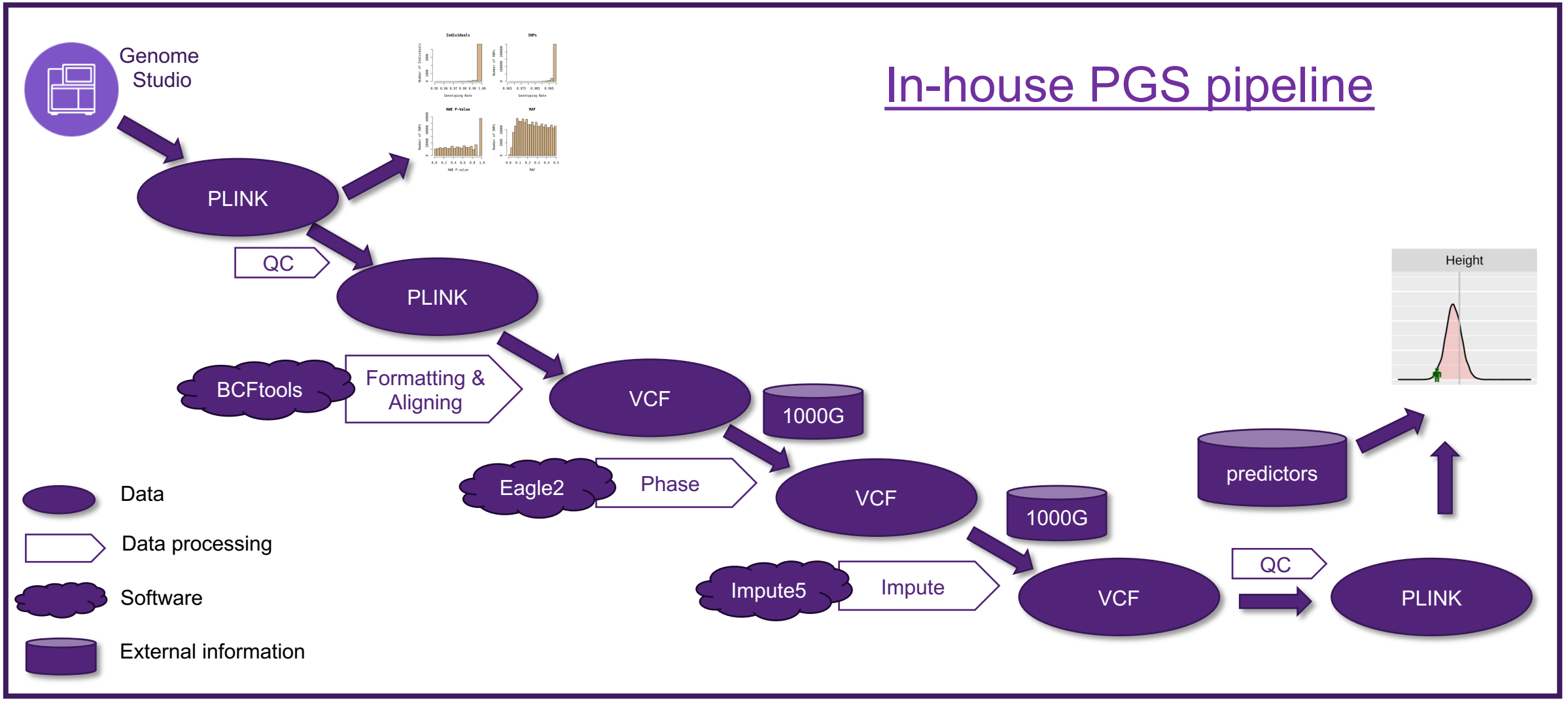
- SNP ID

- Plink does not like duplicate and missing IDs

- Quality

- We suggest to keep all the SNPs regardless of the info score and allele frequency for PGS profiling

schematic of technical pipeline



PGS profiling

```
## Example script
plink \
 --bfile  ${target}_chr${i} \
 --extract overlap_SNPs.txt \
 --score  ${trait}_SBayesRC_predictor.txt 1 2 3 header sum center \
 --out ${target}_${trait}_score_from_chr${chr}
```

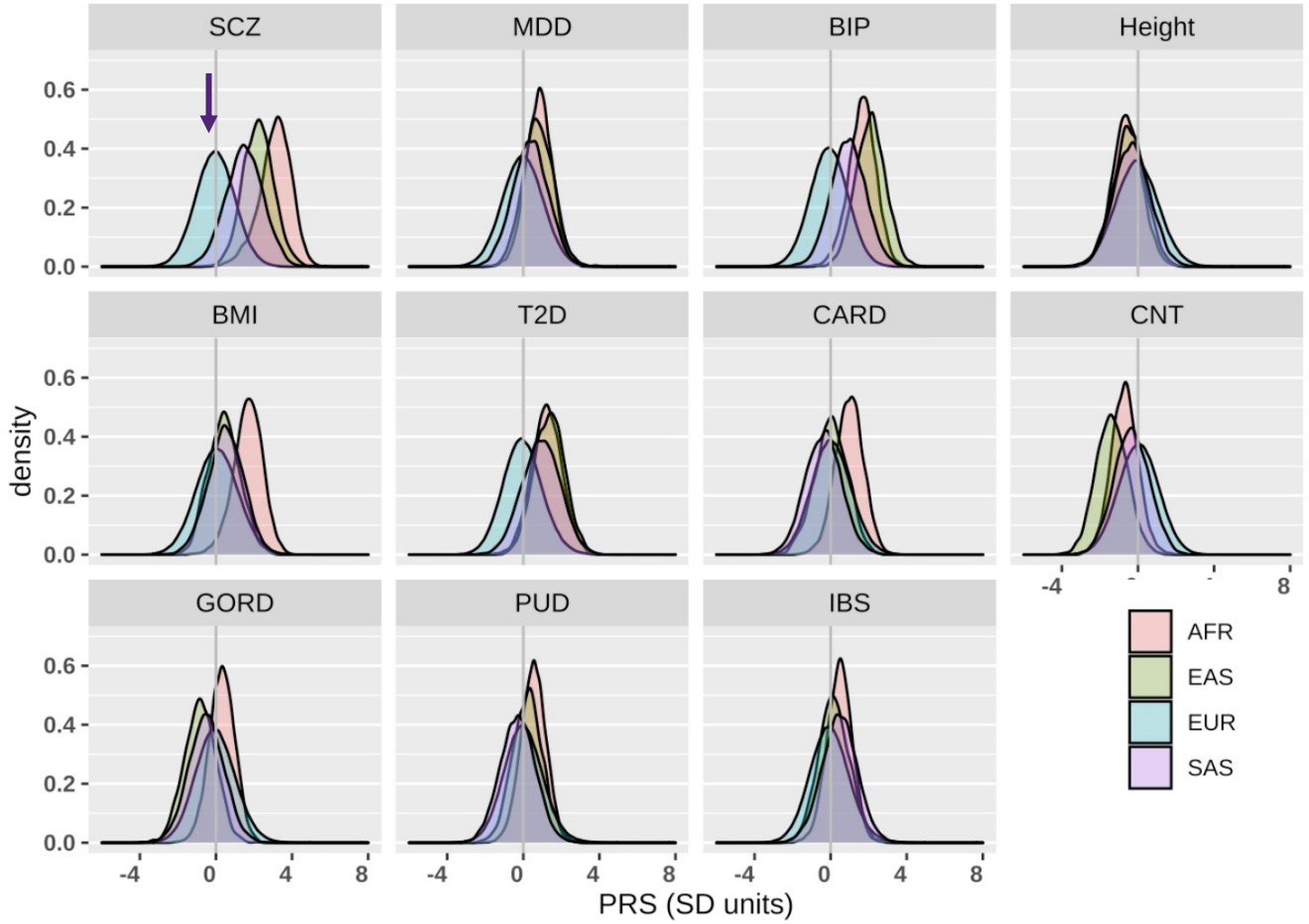
The parameters after your predictor file means

- 1 2 3: Take only the first three columns in the predictor file. The order should be columns of SNP, A1, Effect.
- header: The predictor file has a header row.
- center: The score all all samples will minus the mean value. The mean value of the scores will be zero.
- sum: PLINK prefers to divide the score by the number of SNPs in predictor. Using “sum” will prevent the division step.

Interpret PGS

- Case vs. Control?
- Benchmark with population-wise scores

Match ancestry when benchmarking the PGS

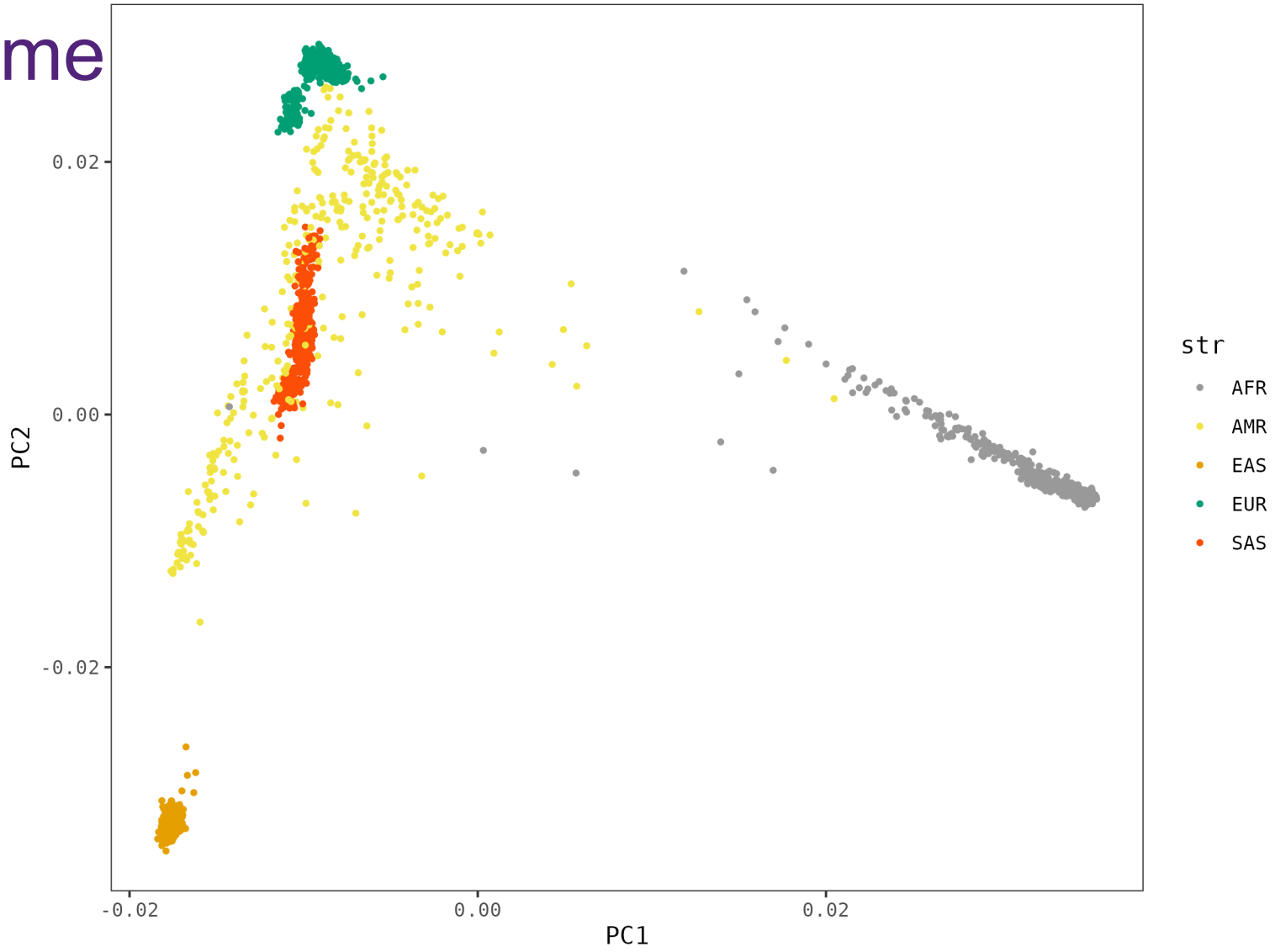


PC calculation using GCTA

- 1000G is the most widely used reference data

```
### generate GRM of reference data
     gcta --bfile ${refpath}/${pcref}.05  \
    --extract common.SNPs.txt \
    --make-grm \
    --out ${pcref}.05.common

### calculate PC of reference data
     gcta --grm     ${pcref}.05.common  --pca 3   --out  ${pcref}.05.common_pca3
```

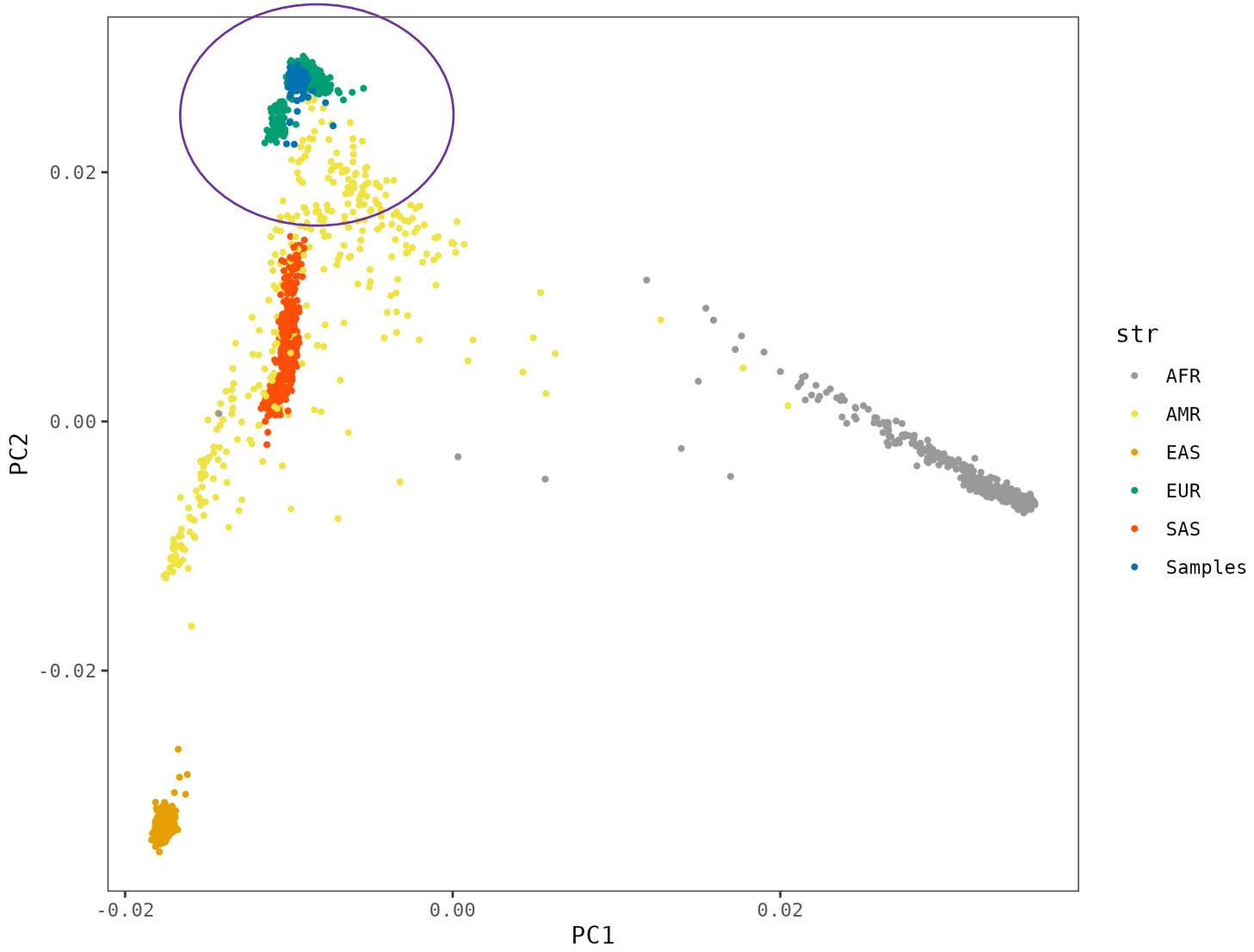
PC plot of 1000Genome



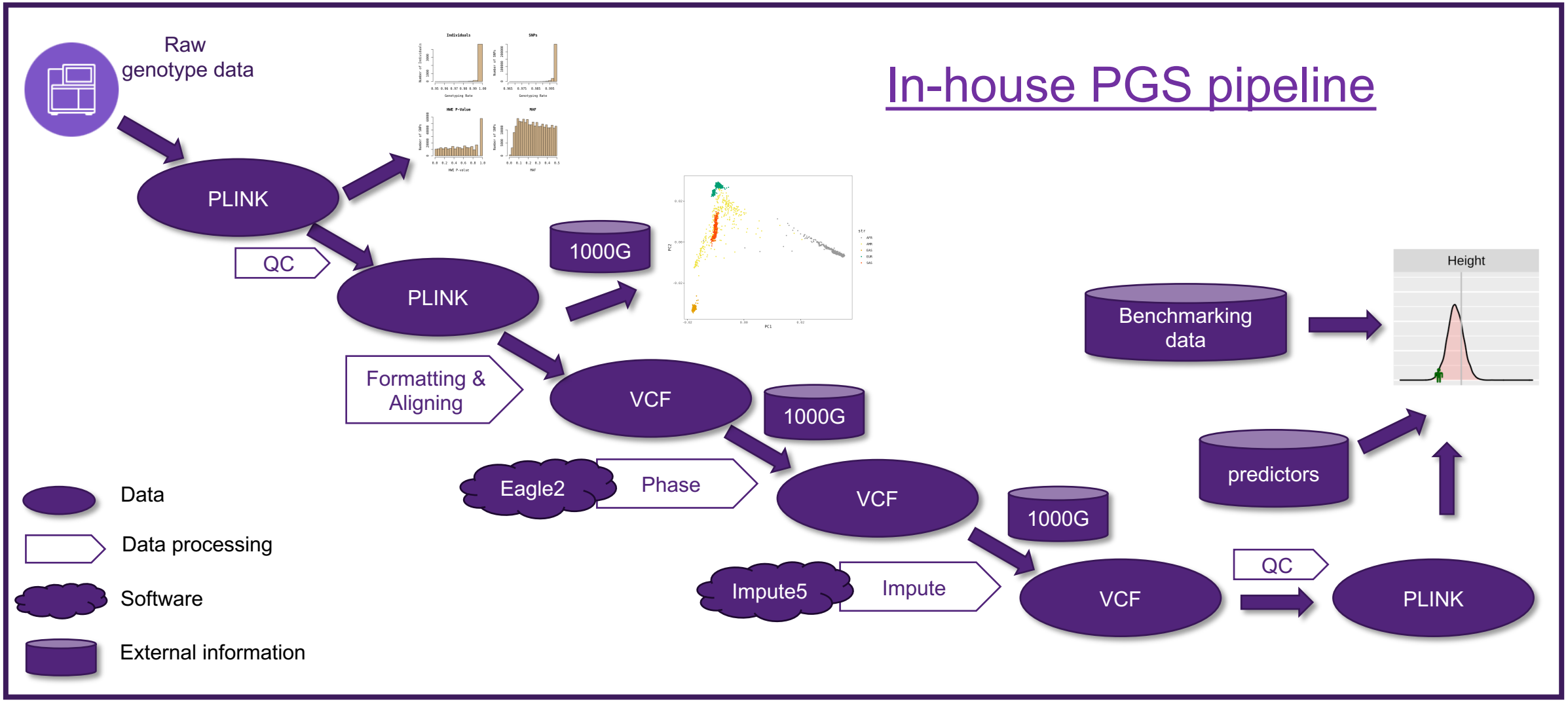
PC projection using GCTA

```
### PC loading
     gcta \
    --bfile  ${refpath}/${pcref}.05    \
    --extract  common.SNPs.txt  \
    --pc-loading   ${pcref}.05.common_pca3   \
    --out  ${pcref}.05.common_pca3_snp_loading

### PC projection
     gcta  \
    --bfile   ${data}  \
    --extract  common.SNPs.txt  \
    --project-loading   ${pcref}.05.common_pca3_snp_loading  3  \
    --out  ${data}_05.common_pca3
```

PC projection



schematic of technical pipeline



Questions?

Survey

https://www.jotform.com/build/241618833733056

Thank you!!