UPA

Matthew Jobin

1. Setup

1.1. Pre-Install

You will need to have installed the following for full operation of UPA. Please note that failure to install any of these might result in less-than-obvious error output.

- plink: https://www.cog-genomics.org/plink2/
- Bcftools: https://samtools.github.io/bcftools/
- Samtools: https://github.com/samtools/samtools
- R: https://www.r-project.org
- SNPRelate: https://bioconductor.org/packages/release/bioc/html/SNPRelate.html
- yHaplo: https://github.com/23andMe/yhaplo
- Haplogrep: You will need to get in touch with the folks who created and maintain Haplogrp https://haplogrep.uibk.ac.at In order to obtain a Java program that allows querying of haplogroups. Without this, the .hsd files generated can still be uploaded manually at the Haplogrep site.
- Java:https://java.com/

2. Input

UPA takes BAM files as input. One available structure is a text list of BAM files defining their locations on disk. The second is a “barcode file” list, wherein sequence ID’s samples and their barcodes (if any) are listed. This is to allow UPA input to stay in sync with a data pipeline by this author, Batpipe.

2.1. Barcode files

A simple list of BAM files, with one BAM file per line. Note that the file should not contain any blank lines. A typical barcode file, with annotations for the columns, is shown below:

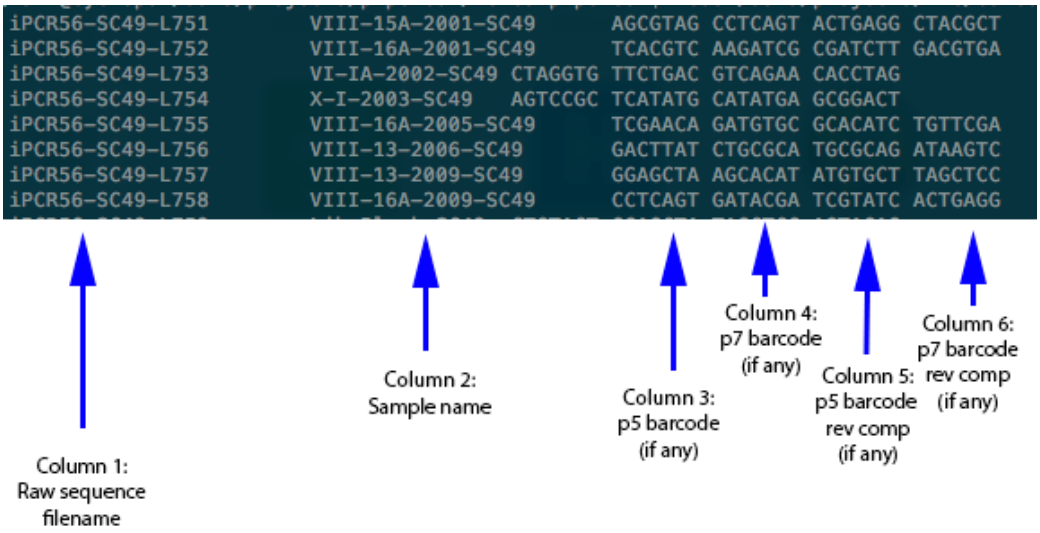

Figure 1:

2.2. BAM files

These must be in subfolders arranged by name and then sub-subfolder named BWA_ plus the name of the mapped reference. The file name must contain the reference name and a .q extension followed but he quality score.

2.3. VCF file

The user may wish to input a VCF (Danecek et al. 2011) directly into UPA so that he/she may specify a method for calling bases before UPA runs. This is usually done because you would like to use a calling method apart from the provided bcftools pileup method, such as ANGSD or ATLAS.

3. Program Flow

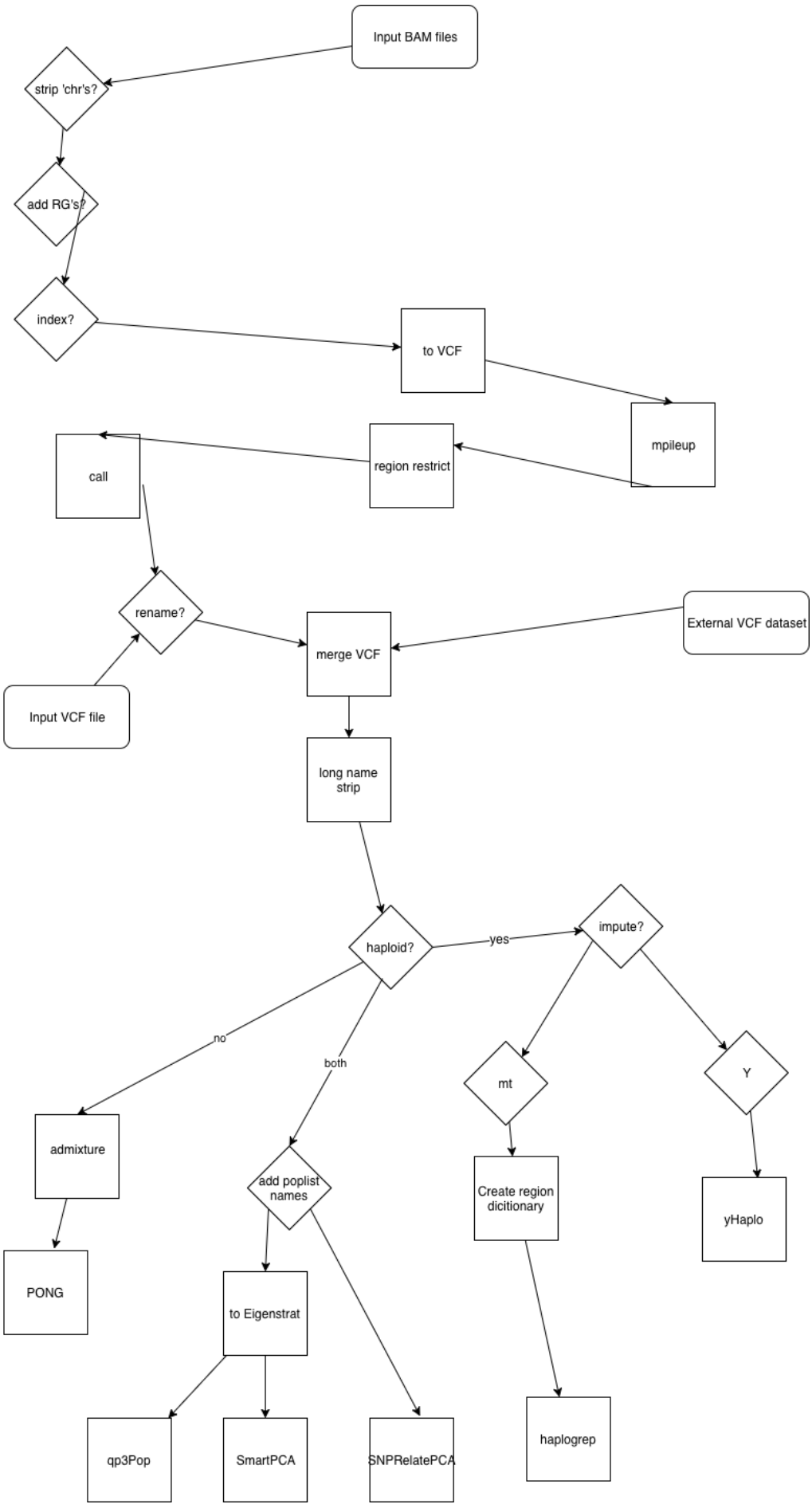
Input BAM files
strip 'chr's?
add RG's?
index?
to VCF
mpileup
region restrict
call
rename?
External VCF dataset
merge VCF
Input VCF file
long name strip
haploid?
yes
impute?
no
both
mt
Y
admixture
add poplist names
Create region dicitionary
yHaplo
PONG
to Eigenstrat
haplogrep
qp3Pop
SmartPCA
SNPRelatePCA

Figure 2: *UPA Program Flow*

def bcfmpileup(flist, ref, bcname, regionrestrict, diploid, cmdfile, logfile): print "mpileup consensus and writing as a VCF..." mpileupcmd = "bcftools mpileup -I -d 8000 -Ov -f " + ref + " " for i in range(len(flist)): sample = flist[i] mpileupcmd = mpileupcmd + sample + ".bam" + " " if regionrestrict: mpileupcmd = mpileupcmd + " -r " + regionrestrict mpileupcmd = mpileupcmd + " | bcftools call -Oz -m -o " + bcname + "-samples.vcf.gz - " if diploid: pass else: mpileupcmd = mpileupcmd + " --ploidy 1 " upa_util.bash_command(mpileupcmd, False, cmdfile, logfile) return bcname + "-samples.vcf.gz"

3.1. Options

Command-line Option	Description	Default
-bc_file	Location of barcode-style input file.	None
-bc_leftspec	Extensions or name to the left of the reference in sample name.	.M.cf.
-bc_rightspec	Extensions of names to the right of the quality score in sample name.	.s
-bam_list	Location of BAM list-style input file.	None
-vcf_file	Location of VCF input file.	None
-wd	Working directory	.
-verbose	Print verbose output.	False
-overwrite	Overwrite existing files and directories.	False
-threads	Number of threads where applicable (e.g. ADMIXTURE).	23

-ref	Reference FASTA file. Must be indexed.	/data/genomes/hg19.fa
-q	BWA minimum quality.	20
-samindex	Generate indexes for SAM/BAM files.	
-diploid	Is data diploid?	
-regionrestrict	Restrict merge/call to a region of the genome.	
-vcfchromrename	File with two columns, one for your BAMs existing chromosomes names and one for the new names.	
-mergevcffile	Name of external dataset in VCF format.	
-mergebamfile	Name of external dataset in BAM format.	
-mito	Use mitochondrial functions.	
-ychr	Use Y chromosomal functions.	
-imputor	Imputation and correction (haploid data only).	
-imptree	Pre-made phylogenetic tree for Imputor.	
-maxheight	Maximum height for IMPUTOR root ward search.	3
-maxdepth	Maximum depth for IMPUTOR rootward search.	3
-nsize	Minimum threshold neighbors to	2

	correct sequencing error.	
-msize	Minimum threshold number of neighbors to impute missing.	3
-ncollect	IMPUTOR neighbor-collection method (rootward, hops, distance).	rootward
-maxhops		

Table 1: *Command-line options for UPA*

3.2. Processing Input

3.2.1. Preparing BAM files for Analysis

There are a number of common preparatory steps tp working with BAMs from a sequencer. When merging your samples with another dataset, you might find that the chromosome names do not match those of your samples. Some forms of mapping strips the read group information needed by genotype callers.

3.2.1.1. Stripping 'chr' from sample chromosomes

UPA provides a convenience function that strips the ‘chr’ element from the names of

chromosomes. This may be helpful when names with the convention “chr1, chr2…” etc are used in your samples but your reference set uses “1,2,3,..”. Use the argument stripchr to invoke this

3.2.2. Calling genotypes from BAM files

3.2.2.1. bcftools mpileup

3.2.2.2. GenoCaller

To use Genocaller (https://github.com/kveeramah/GenoCaller_indent) you will need a UCSC-style (non-binary) BED file.

1. plink --bfile 1240K --keep Keeplist.txt --make-bed --out 1240KTest --output-chr MT
2. plink --bfile 1240KTest --recode --output-chr MT --out 1240KTest
3. awk '{print $1, $4-1, $4}' 1240KTest.map > 1240KTest.bed

3.2.3. Renaming chromosomes

Hint: If you need to figure out what naming conventions were used for your chromosomes for a big external file, try something like: zgrep -o 'chrM’ <your file name>.vcf.gz | wc -l

3.2.4.

4. Functions

The functions of UPA are divided by type of data and function. The initial stages of the program perform data file conversion if necessary and/or requested, while the later stages perform commonly-used analyses based on the type of data and the external repositories available.

4.1. Nuclear

4.1.1. ADMIXTURE

UPA can invoke the software ADMIXTURE for (Alexander et al. 2009)

4.2. Y chromosome

4.2.1. yHaplo

The yHaplo software calls haplogroups for Y chromosomal data (Poznik 2016).

4.3. Mitochondrion

4.3.1. Haplogrep

UPA can generate HSD files for use on Haplogrep website (Weissensteiner et al. 2016). Using the -mito switch creates a user-submittable HSD files. If the user has installed the Haplogrep software (on request from the maintainers of the Haplogrep site. The haplogrepjava switch will submit the generated file to Haplogrep if the software is installed. Please note that all regions that are not SNPs will be stripped from the HSD file.

4.4. All

4.4.1. ADMIXTURE

4.4.2. Rename and filter for populations

An example of a population list file is shown below

If a poplistfile is submitted, then only those individuals who have a matching population in that file will be preserved for further processing.

4.4.3. Prinicpal Components Analysis

UPA can invoke one of two methods for performing Principal Components Analysis (PCA) on the samples:

4.4.3.1. SmartPCA

SmartPCA is part of the Eigensoft package (Patterson et al. 2006) (Price et al. 2006).

4.4.3.2. SNPRelate

SNPRelate is a parallel-processing PCA package for the R statistical suite (Zheng et al. 2012).

5. Bibliography

Alexander, D.H., Novembre, J. & Lange, K., 2009. Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, 19(9), pp.1655–1664. Available at: http://genome.cshlp.org/cgi/doi/10.1101/gr.094052.109.

Danecek, P. et al., 2011. The variant call format and VCFtools. *Bioinformatics*, 27(15), pp.2156–2158. Available at: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btr330.

Patterson, N., Price, A.L. & Reich, D., 2006. Population Structure and Eigenanalysis. *PLoS Genet*, 2(12), p.NaN–NaN. Available at: http://dx.plos.org/10.1371/journal.pgen.0020190.

Poznik, G.D., 2016. Identifying Y-chromosome haplogroups in arbitrarily large samples of sequenced or genotyped men. , pp.1–5. Available at: http://biorxiv.org/lookup/doi/10.1101/088716.

Price, A.L. et al., 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nat Genet*, 38(8), pp.904–909. Available at: http://www.nature.com/articles/ng1847.

Weissensteiner, H. et al., 2016. HaploGrep 2: mitochondrial haplogroup classification

in the era of high-throughput sequencing. *Nucleic Acids Research*, 44, p.W58. Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw233.

Zheng, X. et al., 2012. A high-performance computing toolset for relatedness and principal component analysis of SNP data. *Bioinformatics*, 28(24), pp.3326–3328. Available at: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bts606.