

# UPA

Matthew Jobin

## 1. Introduction

The Universal Pipeline Accessory is a Python tool for automating processing and population genetics analyses downstream of generating BAM files from NGS sequencing runs. It was developed in response to numerous tasks that the members of the UCSC Human Paleogenomics lab performed on a regular basis with a myriad of separate scripts and one-liners in bash, in an attempt to standardize and simplify work in population genetics. While the Human Paleogenomics lab largely works with ancient DNA, many of the functions are applicable to modern DNA data.

## 2. Setup

### 2.1. 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> and Rscript.
- 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/>
- Biopython:<https://biopython.org>
- The EIGENSOFT packages, including convertf, smartpca and qp3pop:  
<https://www.hsph.harvard.edu/alkes-price/software/>
- The following Python modules:
  - progressbar
  - linecache
  - re

## 2.2. Installing UPA

The simplest way to install UPA is from GitHub. From your install directory, type: **git clone <https://github.com/mjobin/UPA.git>** and then **cd** into UPA. You can either place the UPA folder in your PATH or move all the ups scripts to whichever directory in your path you would like to use.

## 2.3. Making permanent modifications

You might want to modify the defaults for UPA, say so that your own scripts and executables folder does not need to be explicitly invoked from the command line. Any text editor will allow you to do this, though be aware that pulling from git will overwrite these changes, so you might need to make them again after an update.

## 3. 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.


### 3.1. 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:

iPCR56-SC49-L751	VIII-15A-2001-SC49	AGCGTAG	CCTCAGT	ACTGAGG	CTACGCT
iPCR56-SC49-L752	VIII-16A-2001-SC49	TCACGTC	AAGATCG	CGATCTT	GACGTGA
iPCR56-SC49-L753	VI-IA-2002-SC49	CTAGGTG	TTCTGAC	GTCAGAA	CACCTAG
iPCR56-SC49-L754	X-I-2003-SC49	AGTCCGC	TCATATG	CATATGA	GCGGACT
iPCR56-SC49-L755	VIII-16A-2005-SC49	TCGAACA	GATGTGC	GCACATC	TGTTCGA
iPCR56-SC49-L756	VIII-13-2006-SC49	GACTTAT	CTGCGCA	TGCGCAG	ATAAGTC
iPCR56-SC49-L757	VIII-13-2009-SC49	GGAGCTA	AGCACAT	ATGTGCT	TAGCTCC
iPCR56-SC49-L758	VIII-16A-2009-SC49	CCTCAGT	GATACGA	TCGTATC	ACTGAGG


  




Column 1:  
Raw sequence  
filename




Column 2:  
Sample name




Column 3:  
p5 barcode  
(if any)



Column 4:  
p7 barcode  
(if any)



Column 5:  
p5 barcode  
rev comp  
(if any)



Column 6:  
p7 barcode  
rev comp  
(if any)

*Figure 1: The “barcode”-type input file standard.*

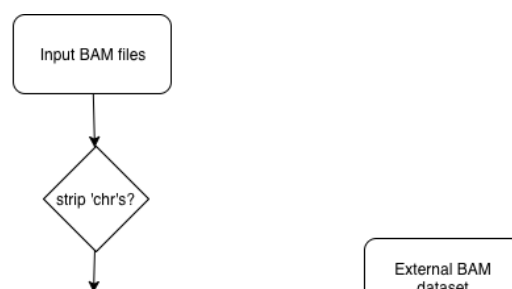
### 3.2. BAM files

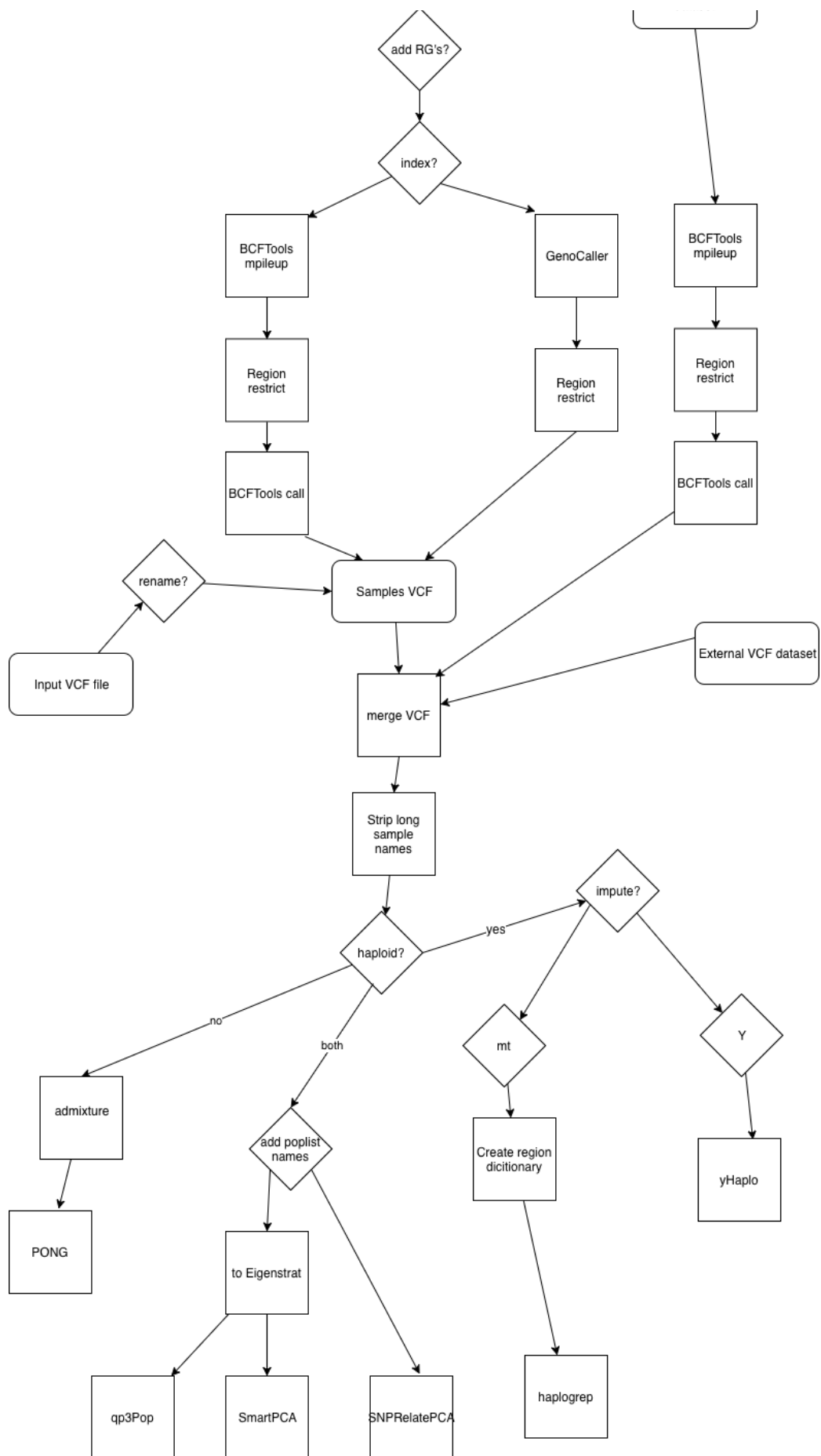
These must be in subfolders arranged by name and then sub-subfolder named **BWA\_<name of the mapped reference>**. The file name must contain the reference name and a .q extension followed by the quality score.

### 3.3. 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.

## 4. Program Flow





**Figure 2: UPA Program Flow**

## 4.1. 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.	
-	File with two columns, one for your BAMs existing chromosomes names	

vcfchromrename	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 correct sequencing error.	2
-msize	Minimum threshold number of neighbors to impute missing.	3
-ncollect	IMPUTOR neighbor-collection method (rootward, hops, distance).	rootward
-maxhops	Maximum number of hops to search in IMPUTOR's hops method.	5
-yhaplo	Location of yHaplo scripts. Leave blank to prevent yHaplo fro running.	

	Text file for population assignment.	
-poplistfile	Also functions as a keep list. Plink formatted: first column is population, second is individual.	
-lowk	Lowest K value for Admixture run.	2
-hik	Highest K value for Admixture run.	10
-reps	Number of Admixture replicates per K.	10
-tohaploid	Convert to haploid before running Admixture.	
-tvonly	Keep only transversions in adpipe functions.	
-termcrit	Termination criterion for ADMIXTURE.	0.0001
-optmethod	Optimizaiton method for ADMIXTURE. Can be em or block.	block
-haplogrepjava	Invoke java version of Haplogrep	
-maxgap	Maximum gap in read before it is counted as a new region.	1
-mindepth	Minimum depth (coverage) to be counted in a region.	1
-admixture	Run ADMIXTURE.	
-snprelatepca	Run SnpRelatePCA.	
-smartpca	Run smartpca.	
-ancient	Turn on arguments related to	

	processing ancient DNA.	
-scriptsloc	Location of external scripts.	/data/scripts
-binloc	Location of external binary executables.	/usr/local/bin
-stripchr	Strip out “chr” from chromosome names.	
-addreadgroup	Add read group (RG) back to your sample BAMs.	
-callmethod	Genotype calling method. Options: bcf, genocaller.	bcf
-gcbedfile	UCSC BED file for use with GenoCaller.	
-gcindent	Indent depth for use with GenoCaller.	2
-plinkgeno	Value for plink geno argument.	0.99
-annotate	Annotate the ID column of your samples’ VCF using an external dataset.	
-mergefoundonly	When merging VCF files, only keep sites found in both files	

**Table 1:** Command-line options for UPA

Where UPA options listed above refer to external software, they are usually passing those options straight to that software, and thus consulting the manual for that software will give the use more detail about the option’s effect.



## **4.2. Processing input files**

A number of common steps need to be taken in many cases to process BAM files for population genetic analysis. One necessary step is the calling of genotypes from the raw BAMs, for which there are several methods. UPA provides two methods internally, and also allows user-defined VCF files to be imported, skipping the calling step and instead allowing the calling to be done externally. For internally-called BAMs, UPA also provides some optional steps for preparing files for analysis that should circumvent common bottlenecks in a data analysis pipeline.

### **4.2.1. Preparing BAM files for analysis**

There are a number of common preparatory steps to 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.

#### **4.2.1.1. 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

#### **4.2.1.2. Adding ReadGroup formation**

Processing during sequencing can sometimes strip read group information from BAM files. If the user uses the `-addreadgroup` argument, UPA will add in basic read group information by invoking Picard. The sample name will be taken from the file name, while other information such as RGPL will default to "Illumina".

### **4.2.2. Calling genotypes from BAM files**

UPA can use one of three methods for calling genotypes from BAM files. Each of

these have relative merits for speed, convenience and ability to work with degraded/ancient DNA. For a full comparison of these software, please consult their respective GitHub pages or manuals.

#### **4.2.2.1. Bcftools mpileup**

UPA can call genotypes using BCFTools's mpileup and call methods. This is also the approach taken if the user supplies external dataset in BAM format. The BCFTools options used by UPA are:

-C 50: Coefficient for downgrading mapping quality for reads containing excessive mismatches.

-d 8000: Maximum depth of 8000.

-f <ref>: Uses as reference the FASTA files supplied by -ref.

-q <q>: Uses a quality score supplied by the user.

#### **4.2.2.2. GenoCaller**

To use Genocaller ([https://github.com/kveeramah/GenoCaller\\_indent](https://github.com/kveeramah/GenoCaller_indent)) you will need a UCSC-style (non-binary) BED file. To obtain one from your external dataset in BED/BIM/FAM format, you can use steps similar to the below in plink:

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`

You will need the resulting BED file to use GenoCaller in UPA. Reference it using the -gcbefile argument. GenoCaller directly outputs VCF files, which UPA then merges into a samples VCF file before proceeding.

#### **4.2.3. Renaming chromosomes**

The merging step in the following section will not work if the sample files' conventions for chromosome names do not follow that of the external data.

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`

#### **4.2.4. Annotation**

It is often necessary to annotate (i.e. fill in missing information) for BAM files that are just returned from sequencing. In particular, the ID column of a VCF file generated from BAMs in UPA will need information in the ID column for successful merging with external data. The most straightforward way to do this is to annotate from the ID column of the external dataset you are merging, and this is the approach taken by UPA.

When troubleshooting, it is a good idea to check your external VCF dataset to make sure that it does indeed contain the annotation information necessary.

#### **4.2.5. Merging with an external dataset**

After the calling step, or after importing a sample VCF, UPA will attempt to merge the samples with a VCF from an external dataset if `-mergevcffile` is used. It is not uncommon for the REF alleles of the two files to fail to match, often an artifact of the assumptions used to generate that VCF file from BED/BIM/FAM format. If the user invokes UPA with `-mergevcffile` or `-mergebamfile` arguments, then it will merge that file with the VCF file generated from the BAM files or VCF file generated from the samples. The external data will be rearranged such that the REF allele matches that of the sample data, since the sample data is using a user-specified reference in the case of inputting BAM files. To save space, UPA will not merge lines where there are only missing alleles across all samples for the site. The name of the merged VCF file will be `<base name of your sample file>-MERGED.vcf`.

If you use the -annotate argument, the ID column of the merged file will be drawn from the external data file. If you use the -mergefoundonly argument, the merged file will only contain sites found in both files, while leaving an option out will fill in missing data for the external data wherever it does not have a site matching the samples.

NOTE: The QUAL and FILTER columns will preserve the information from your samples, NOT that of the external dataset.

## **5. Population genetics 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.

### **5.1. Autosomal only**

#### **5.1.1. ADMIXTURE**

UPA can invoke the software ADMIXTURE for demographic inference (Alexander et al. 2009). the user can specify the maximum and minimum K values, and the number of the replicates per K. The best run of each K is moved to a BEST directory based on its cross-validation score. Cross-validation plots are also generated in R using Rscript.

### **5.2. Y chromosome**

#### **5.2.1. yHaplo**

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

### **5.3. Mitochondrion**

#### **5.3.1. 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.

## **5.4. All**

### **5.4.1. ADMIXTURE**

### **5.4.2. 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.

### **5.4.3. Principal Components Analysis**

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

#### **5.4.3.1. SmartPCA**

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

#### **5.4.3.2. SNPRelate**

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

## **6. 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>.