

A bioinformatics workflow for detecting signatures of selection in genomic data

Murray Cadzow^{1,2}, James Boocock^{1,2}, Hoang Tan Nguyen^{1,2}, Phillip Wilcox^{2,3}, Tony R Merriman¹
and Michael A Black¹

¹Department of Biochemistry, University of Otago

²Department of Mathematics and Statistics, University of Otago

³Scion Research, Rotorua, New Zealand

September 9, 2013

Contents

1	Introduction	2
2	Getting Started	2
2.1	Prerequisites	2
2.2	Installation	3
2.3	Genetic Maps and Impute Haplotypes	3
2.4	Ancestral Fasta Files	4
3	Tutorial	4
3.1	Selection Signatures at the Lactase Locus	4
3.1.1	Getting the Data	4
3.2	Setting up Pipeline Run	4
3.3	Population Files	4
3.4	Data Analysis	5
3.4.1	Fst	5
3.4.2	Fey and Wu's H	5
3.4.3	iHS	5
3.4.4	Tajima's D	6
3.4.5	rSb	6
4	Command line Arguments	6
4.1	Multipopulation	6
4.1.1	Input Files	6
4.1.2	Output Files	7
4.1.3	Other parameters (Compulsory)	7
4.1.4	Other parameters (Optional)	7

4.2	Selection Pipeline	7
4.2.1	Input Files	7
4.2.2	Output Files	7
4.2.3	Other parameters(Compulsory)	8
4.2.4	Other parameters(Optional)	8
4.3	Ancestral Annotation	8
4.3.1	Input Files	8
4.3.2	Output Files	8
4.3.3	Other parameters	9
4.4	Configuration File	9
4.4.1	system	9
4.4.2	environment	9
4.4.3	selection_pipeline	9
4.4.4	vcf_tools	9
4.4.5	shapeit	10
4.4.6	impute2	10
4.4.7	plink	11
4.4.8	Rscript	11
4.4.9	python	11
4.4.10	ancestral_allele	11
5	Extra Features	11
5.1	Galaxy Intergration	11

1 Introduction

Mik to fill this in

2 Getting Started

2.1 Prerequisites

The selection pipeline was developed on a 64-bit ubuntu 13.04 system but should work on any 64-bit linux deriviant assuming some basic libraries and tools are installed on your system.

- python2 or python3
- bourne-again Shell (Bash)
- perl5
- git

2.2 Installation

To install the package standalone, requiring manual configuration of the config file run the following command.

```
./install.sh --standalone
```

The rest of this section will be dedicated to the automatic installation. To perform an automatic installation of the selection pipeline run the command.

```
./install.sh
```

Installation creates a default config file located in the base directory of the pipeline. Installation adds a program called selection_pipeline to the system path. To test the program is installed correctly run the following command at a terminal prompt.

```
selection_pipeline -h
```

2.3 Genetic Maps and Impute Haplotypes

To use the phasing and imputation features of the pipeline requires both genetic map files and haplotype files. For humans these files that conform to the format required for shapeit and impute2 can be found [here](#). For impute2 one reference is available [here](#), download and extract the archive to referencefiles/impute_ref and uncompress the contents. For shapeit2 a genetic map can be found [here](#), download and extract the archive to referencefiles/shapeit_ref.

To use other reference files with the selection pipeline requires setting a few options in the config file. The question mark character "?" in the config is substituted by the chromosome number, this is used for reference files that are split on chromosomes.

```
...
genetic_map_prefix=genetic_map_chr?_combined_b37.txt
...
impute_map_prefix=genetic_map_chr?_combined_b37.txt
impute_reference_prefix=ALL_1000G_phase1integrated_v3_chr?_impute
...
```

If you decide to store your reference files in another location, further options will require alterations.

```
...
genetic_map_dir= \${HOME}/MerrimanSelectionPipeline/referencefiles/shapeit_ref
...
impute_map_dir= \${HOME}/MerrimanSelectionPipeline/referencefiles/impute_ref
impute_reference_dir= \${HOME}/MerrimanSelectionPipeline/referencefiles/impute_ref
...
```

2.4 Ancestral Fasta Files

To generate results for iHS requires assigning the ancestral allele. The selection pipeline uses the 6-way EPO (Enredo-Pecan-Ortheus) alignment pipeline. The files can be downloaded from [here](#).

If you downloaded your reference to a different location you can set the following setting in your config file.

```
...
ancestral_fasta_dir = # directory you downloaded alignment to #
...
```

3 Tutorial

3.1 Selection Signatures at the Lactase Locus

3.1.1 Getting the Data

The lactase gene is located on Chromosome 2 between 136,545,410-136,594,750 positions. For the example we will use a 10 megabase region containing the Lactase gene and the CEU and YRI populations from the 1000 genomes. In order to demonstrate how to use the pipeline we will use the chromosome 2 region 130,000,000-140,000,000. To download the example dataset enter the command below.

```
wget http://tutorial_file_location.com
```

Extract the example data into a new folder.

3.2 Setting up Pipeline Run

To perform the analysis the first step will be to navigate to the folder you extracted the tutorial data.

3.3 Population Files

Population files are required for any cross population comparisons. The commands below will initiate the data generation step. Population files are line separated files the first line contains the population name every successive line contains and individual ID from that population.

```
multipop\_selection_pipeline -p POP1.txt -p POP2.txt -i input.vcf --c defaults.cfg
```

The generated folders and current folder have all the data required to perform further selection analysis. Within each population folder 4 output files are generated these contain tajima's D, iHH, an updated VCF and Fey and Wu's H statistic these files are located in the results folder inside each population subfolder. Between each each population Fst are generated and located in the fst folder. Fst calculations are generated using the weir and cockerham estimator.

3.4 Data Analysis

The pipelines purpose is to generate signatures of selection from a VCF formatted input file. In order to express the usefulness of the pipeline it is pertinent to illustrate the effectiveness of the pipeline. The next section describes some basic plotting of these data using the R programming language. All following commands are run in a R session with the working directory in the base directory you are running the tutorial in.

3.4.1 Fst

```
CEU_CHB_weir_fst=read.table("fst/CEU_CHB_chr2.weir.fst", header=TRUE)
weirFst=CEU\_CHB\_weir\_fst
pop1="CEU"
pop2="CHB"
weirMean=mean(weirFst[,3][weirFst[,3] != "NaN"])
weirThresUpper =quantile(weirFst[,3], 1-0.025, na.rm=TRUE)
weirThresLower =quantile(weirFst[,3], 0.025, na.rm=TRUE)
plot(weirFst[,3]~weirFst[,2], pch=16, cex=.4, type="p",
xlab=paste("Chr",chr,"(Mbp)",sep=" "), ylab="Weir Fst",
main=paste("Weir Fst for",pop1,"and",pop2,"Populations", sep=" " ),
xlim=c(0,300e6),ylim=c(-0.2,1), yaxt="n", xaxt="n")
axis(side=1, tick=TRUE,at=c(0, 50e6,100e6,150e6,200e6,250e6,300e6),
labels=c("0","50","100","150","200","250","300"))
axis(side=2, tick=TRUE, at=seq(0,1,0.2 ) )
abline(h=weirThresUpper, lty=3, col = "black")
abline(h=weirThresLower, lty=3, col="black")
abline(h=weirMean, lty=5,col="black")
```

3.4.2 Fey and Wu's H

3.4.3 iHS

```
CEU\_CHB\_weir\_fst=read.table("CEU\_CHB\_chr2.weir.fst", header=TRUE)
CEU\_ihh=read.table("CEU\_chr2.iHH")
CEU\_ihs=ihh2ihs(CEU\_ihh)
#plot density
distribplot(CEU\_ihs$res.ihs$iHS, main="1000 Genomes CEU iHS Density")
#plot iHS values
ihspplot(CEU\_ihs$res.ihs,plot.pval=FALSE,ylim.scan=2,main="CEU iHS",pch=".")
#plot iHS Pvalues
ihspplot(CEU\_ihs$res.ihs,plot.pval=TRUE,ylim.scan=2,main="CEU iHS",pch=".")
```

3.4.4 Tajima's D

```
CEU\_CHB\_weir\_fst=read.table("CEU\_CHB\_chr2.weir.fst", header=TRUE)
CEU\_tajimaD=read.table(file="CEU\_chr4.Tajima.D", header=TRUE)

tajimaMean=mean(CEU\_tajima[CEU\_tajima$TajimaD != "NaN",]$TajimaD)
tajimaThresUpper = quantile(CEU\_tajima$TajimaD, (1-0.025))
tajimaThresLower = quantile(CEU\_tajima$TajimaD, 0.025)

#plot Tajima's D
plot(CEU\_tajimaD$BIN\_START,CEU\_tajimaD$TajimaD, pch=16, cex=0.4,
      frame=FALSE, ylab="Tajima's D", ylim=c(-3,6), xaxt="n", yaxt="n",
      xlim=c(0,300e6), xlab=paste("Chromosome",chr,"(Mbp)",sep=" "))
axis(side=1, tick=TRUE,at=c(0, 50e6,100e6,150e6,200e6,250e6,300e6),
      labels=c("0","50","100","150","200","250","300"))
axis(side=2, tick=TRUE, at=c(-3:6) )
title(main="CEU Tajima's D Chromosome 2)
abline(h=tajimaMean, lty="dashed", lwd=2)
abline(h=c(tajimaThresUpper, tajimaThresLower), lty="dotted", lwd=2)
```

3.4.5 rSb



4 Command line Arguments

The selection pipeline contains three major scripts selection_pipeline aa_annotate and multipopulation. The selection pipeline does all the within population statistics calculations. The multipopulation program calculates all the between population statistics and calls the selection pipeline. The aa_annotate program annotates a haplotype file or a phased vcf file with the ancestral allele from the 6-way EPO alignment, for other species or alternative ancestral annotation the feature will be added promptly.

4.1 Multipopulation

4.1.1 Input Files

- -i <vcf input file> VCF file containing all the populations you want to analyse from one chromosome or a part of a chromosome only.

4.1.2 Output Files

- FST

Fst results are stored in the fst folder with the chromosome number followed by the two populations. e.g 2CEUYRI.fst

- Selection Pipeline Results

All single population pipeline results are stored in the subdirectory of the population in a folder named results.

These contain the ihh, tajimasD and a population VCF file.

4.1.3 Other parameters (Compulsory)

- -c <Chromosome>

Chromosome name used for labelling outputs.

- -a <Arguments to the selection pipeline>

Quoted string containing any extra arguments to the selection_pipeline program. e.g "-imputation"

- -C <path to config file>

Path to the selection pipeline config file an example config file is located in the base directory of the extracted package.

4.1.4 Other parameters (Optional)

- -fst-window-size <FST window size>

Argument is passed directly to the VCF tools command line.

- -fst-window-step <FST window step>

Argument is passed directly to the VCF tools command line.

4.2 Selection Pipeline

4.2.1 Input Files

- -i <VCF input file>

Single population single chromosome VCF input file.

4.2.2 Output Files

The Results directory contains all the output files.

- .ihh file

The outputted iHH data for each SNP

- .taj_d file

Tajima's D output

- .vcf file

Single population VCF updated by the pipeline, can contain.

4.2.3 Other parameters(Compulsory)

- `-config-file <Config File path>`

Path to the selection pipeline config file an example config file is located in the base directory of the extracted package.

4.2.4 Other parameters(Optional)

- `-maf <minimum MAF>`

Minor allele frequency filter threshold any SNPs below this threshold will be discarded from the analysis.

- `-hwe <hardy-weinberg minimum p-value>`

A hardy weinberg test is performed on every snp any snps failing the test will be discarded.

- `-daf <Minimum derived allele frequency>`

Derived allele frequencies below this minimum will be discarded.

- `-remove-missing <Inclusion threshold for missing genotypes>`

Inclusion criteria for SNPs with missing data. SNPs with less than this value will be removed from analysis.

- `-TajimaD <tajimas D bin size>`

Tajima's D statistic bin size.

4.3 Ancestral Annotation

The program *ancestral_annotation* is installed on the program path. The program annotates haps and vcfs files with ancestral allele annotation from the 6-way IPO alignment or the human reference genome.

4.3.1 Input Files

- `-i` or `-haps <HAPS File>`

Haplotype File (.haps)

- `-v <Phased VCF file>`

Phased VCF file (.vcf), phased VCF genotypes denoted by a bar (|) for each sample.

- `-a` or `-aa <Ancestral allele fasta>`

Ancestral allele annotation file. Currently only works on a the full 1000 genomes reference file from 1000 genomes or the single chromosome fasta files from the 6-way EPO alignment.

4.3.2 Output Files

- `-o` or `-output <Output file name>`

Output file name optional argument by default output is sent to the stdout stream.

4.3.3 Other parameters

- `-ref-fasta`

Denoting that you are using the human reference allele as the ancestral allele.

- `-f` or `-format <format>`

The 6-way EPO alignment denotes ancestral alleles with both high and low confidence. To use only ancestral alleles with high confidence use `-format high`. To use both high and low confident alleles use `-format low`. By default the program will use only highly confident alleles.

4.4 Configuration File

The selection pipeline requires a configuration file, by default the program looks in the current working directory for a file named `defaults.cfg` but you can point the program to any file using command line arguments. There are two main programs in the selection pipeline namely *selection_pipeline* and *multi_population*. These programs share a config file but certain configuration parameters can be omitted when using the *selection_pipeline* program exclusively. A clean install of the program generates an example configuration file containing default arguments for all the compulsory parameters. The default config file contains an example of the format.

4.4.1 system

- `threads_avaliable` Certain programs in the pipeline can take advantage of multicore computers. This option instructs the pipeline about the maximum number of concurrent processes it is allowed to use.

4.4.2 environment

- `LD_LIBRARY_PATH`

Set the library path when running the pipeline, this enables the pipeline to use the shared libraries that are used for some programs in the pipeline. (alter this option with caution!)

- `PERL5LIB`

Sets the `PERL5LIB` environment variable, this enables the pipeline to use the perl libraries required by `VCFTOOLS`. (alter this option with caution!)

4.4.3 selection_pipeline

- `selection_pipeline_executable`

Points to the location of the `selection_pipeline_executable`.

4.4.4 vcf_tools

- `vcf_tools_executable`

Points to the `vcftools` executable, by default it points to the `vcftools` executable installed with the pipeline.

- `vcf_subset_executable`

Points to the vcf-subset executable, by default pointing to the vcf-subset installed with the pipeline.

- `vcf_merge_executable`

Points to the vcf-merge executable, by default pointing to the vcf-subset installed with the pipeline.

- `extra_args`

A quoted string containing extra arguments to send to the `vcf_tools` executable.

4.4.5 `shapeit`

- `shapeit_executable`

Location of the shapeit executable.

- `genetic_map_dir`

Directory containing the genetic map for shapeit.

- `genetic_map_prefix`

The full file for the genetic map files with a "?" character representing the changing chromosome number.

- `extra_args` extra arguments to send to shapeit. (Warning: Certain options could potentially break to pipeline use with caution)

4.4.6 `impute2`

- `impute_executable`

Location of the impute2 executable

- `impute_map_dir`

Directory containing the genetic map for impute2

- `impute_reference_dir`

Directory containing the reference panel (`.legend` and `.hap`) files for impute2.

- `chromosome_split_size`

Window size for imputation calculation.

- `impute_map_prefix`

The full file name for the genetic map files with a "?" character representing the changing chromosome number

- `impute_reference_prefix`

The full file name for the reference panels minus the extension with a "?" character representing the changing chromosome number.

- `extra_args`

extra arguments to send to `impute2`. (Warning: Certain options could potentially break to pipeline use with caution)

4.4.7 **plink**

- `plink_executable`

Location of the plink executable

4.4.8 **Rscript**

- `rscript_executable`

Location of the rscript executable. (Program usually on path so just `Rscript` is the default)

- `indel_filter`

Location of the rscript `indel_filter` (`hap_indel_and_maf_filter.R`)i

4.4.9 **python**

- `python_executable`

location of the python executable (2 or 3)

4.4.10 **ancestral_allele**

- `ancestral_allele_script`

Location of the ancestral_annotation script (`aa_annotate.py`)

- `ancestral_fasta_dir`

Directory containing the ancestral reference files

- `ancestral_prefix`

Full file name for ancestral fasta files containing a "?" character

5 **Extra Features**

5.1 **Galaxy Intergration**

The galaxy folder contains the scripts required to add the selection pipeline to your local galaxy installation. The pipeline is also available on the galaxy toolshed at `galaxy_url`. To do intergrate the pipeline into galaxy.