

SDpop User Manual

Jos Käfer

February 29, 2024

jos.kafer@cnrs.fr

SDpop is a program that calculates probabilities for sex-linkage in population sequencing data. Individual genotypes need to be provided, as well as the sex of all individuals. SDpop is accompanied by some helper programs, especially a program, `popsum`, to convert `vcf` or `reads2snp`'s `gen` format to the `cnt` format SDpop uses.

1 Installation

SDpop is written in C, with some functionalities from C++. It depends on C and C++ standard libraries, as well as some (minor) facilities from the GNU Scientific Library (GSL). The C++ features need at least C++11 support.

On a reasonably up-to-date linux system, the instruction in the makefile should be sufficient. Thus,

```
make all
```

should do. If this doesn't work, the first thing to do would be to use a more recent compiler.

If you intend to run the programs on a computer cluster (i.e. compilation is done on a different machine than running), you might want to compile the programs statically:

```
make all CXXFLAGS=" -static"
```

2 Usage

2.1 Input files

Either a `vcf` file with genotyping information, or a file in `reads2snp`'s `gen` format. The `vcf` file should at least have a "GT" (genotype) field, and this field should be the first one in each sample.

`popsum` reads the genotype file and extract counts of genotypes for polymorphic positions. Furthermore, genotyping information for at least one individual of both sexes needs to be available. To run `popsum` on a `vcf` file `input.vcf` to create `cnt` file `output.cnt`, type

```
popsum input.vcf output.cnt v female1,female2,... male1,male2,...
```

The third parameter indicates `popsum` will consider `vcf` input. The last two arguments are the lists of names of the individuals of which genotypes should be counted. These names should exactly match the names in the `vcf` file. If not all names found in the `vcf` file are given, `popsum` issues a warning; equally, when names given in the arguments are not found, a warning is produced. Thus, `popsum` can be used to count genotypes for only a subset of the `vcf` file. Note that the convention adopted here is to start with the female names. This will have consequences for the inference of XY or ZW segregation types by `sdpop`.

To run `popsum` on a file in `reads2snp`'s `gen` format, type

```
popsum input.gen output.cnt r female1,female2,... male1,male2,...
```

2.2 cnt intermediate file

The `cnt` file thus obtained has two types of lines, the contig lines starting with ">", and the position lines. Contig lines have seven fields: the contig name, the number of positions with two observed alleles, the number of positions with three observed alleles, the number of positions with four observed alleles, Watterson's θ , and, if a `gen` file was used, the number of genotypes read, and the average genotype probability. Example:

```
>sl_contig1 39 0 0 0.009684 16000 0.000125
```

For each position for which two alleles have been observed, popsum outputs one line, starting with the position identifier (the position number given in the vcf or gen file), the two observed alleles (in random order), and the counts of the observed genotypes. These counts represent the number of homozygotes for allele 1, heterozygotes, and homozygotes for allele 2, for both sexes (female and male). For example,

```
4 T,A 0 3 5 0 1 7
```

indicates that at position 4 in the contig, there are 3 females AT and 5 AA, and 1 male AT and 7 AA.

If the cnt file was created from a gen file, the average error rate is calculated from the genotype probabilities. The last line of the cnt file will be like

```
#mean error rate: 0.000204
```

2.3 SDpop

The program SDpop allows to run different models of different degrees of complexity on the data. Its arguments are:

- the input file (cnt format)
- the output file
- the error mode:
 - **n**: no errors. This will likely cause numerical errors.
 - **f**: a fixed error rate, that will not be optimized. If this option is used, a rate should be given as the next argument.
 - **r**: a fixed error rate, that is read from the last line of the cnt file (see above).
 - **e**: error rate to be optimized.
- the sex chromosome system to fit; choose between:
 - **n**: no sex chromosomes
 - **x**: XY-type sex chromosomes
 - **z**: ZW-type sex chromosomes
 - **b**: both XY- and ZW-type sex chromosomes
- whether haploid segregation should be considered (1) or not (0)
- whether paralogous segregation should be considered (1) or not (0)
- whether the output should be in “simple” (**s**) or “expert” (**e**) mode. The simple mode outputs the recommended statistics, i.e. those that were used for the performance calculations in the manuscript. The expert mode provides some alternative statistics as well, as described in the manuscript’s Appendix; the user is free to base the inference on those, but thorough testing of their performance should be done beforehand.
- optionally, the average number of individuals genotyped at the positions in a contig below which the contig is excluded from the analysis.

The most elaborate model is thus run as:

```
sdpop input.cnt seb.out e b 1 1 e
```

This will optimize parameters for 7 segregation types (6 parameters), 3 subtypes (2 parameters) and the error rate (1 parameter). The simplest model would be:

```
sdpop input.cnt sfm.out f 0.001 n 0 0 s
```

without the optimization of any parameter.

The optimization starts from randomly chosen values for the segregation type proportions π_j , and stops when no parameter has changed more than 0.01% for 10 iterations. If a parameter value falls below 0.00001, it is still optimized, but its contribution to convergence is not evaluated anymore; this avoids spending too many iterations optimizing parameters that don't contribute significantly to the likelihood. Users wishing to change these parameters are invited to do so in the source code (look for “`double stop=`” and “`double noconv=`”).

For each iteration, the log-likelihood and the parameter values are written to the standard output. Upon convergence, the output is written to the output file. This file is composed as follows:

- sometimes: a warning message (“Warning: an error occurred, probably due to numerical problems (see stderr for details). The following output should not be used for other purposes than finding the error.”) This message is outputted when the likelihood decreases, instead of increases, as it should. The most likely (and not very worrying) case is that the likelihood decreased by a very small amount, due to rounding imprecisions. In this case, the message to the standard error will read “Warning: log-likelihood decreases by 0.000000; interrupting maximization and proceeding to output”, but the output can probably still be used. If other warnings are produced, one should try to find the cause.
- A line with SDpop's parameter values and model likelihood. This line gives all optimized proportions of the segregation types, in the following order: autosomal, haploid, paralog, x-hemizygote, xy, z-hemizygote, zw. Only proportions corresponding to modelled segregation types are given. Then, the proportions of the subtypes are given, if modelled. Next, the error rate is always indicated, as well as the log-likelihood, and the Bayesian Information Criterion, calculated in three different ways: by using the number of sites, the number of contigs, and the number of sites times the number of individuals.
- Next, contig- and site-wise posterior values are given. First, there are two lines starting with “#”, which indicate the fields of the contig lines (recognizable by a “>”) and the site lines.
- Contig line fields are (only fields corresponding to the model used are given; fields with an asterisk are only given in “expert” output mode):

contig_name name of the contig

N_sites number of polymorphic sites

mean_coverage average number of genotyped individuals per site

j_autosomal reminder of the number used for the autosomal segregation type (always 1)

posterior_autosomal* posterior probability for the autosomal segregation type based on the mean of the posterior probabilities per site, calculated using the prior

geometric_autosomal* posterior probability for the autosomal segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior

noprior_autosomal posterior probability for the autosomal segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior

j_haploid reminder of the number used for the haploid segregation type (always 2)

posterior_haploid* posterior probability for the haploid segregation type based on the mean of the posterior probabilities per site, calculated using the prior

geometric_haploid* posterior probability for the haploid segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior

noprior_haploid posterior probability for the haploid segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior

j_paralog reminder of the number used for the paralogous segregation type (always 3)

posterior_paralog* posterior probability for the paralogous segregation type based on the mean of the posterior probabilities per site, calculated using the prior

geometric_paralog* posterior probability for the paralogous segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior

noprior_paralog posterior probability for the paralogous segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior

- j_xhemizygote** reminder of the number used for the x-hemizygote segregation type (always 4)
 - posterior_xhemizygote*** posterior probability for the x-hemizygote segregation type based on the mean of the posterior probabilities per site, calculated using the prior
 - geometric_xhemizygote*** posterior probability for the x-hemizygote segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior
 - noprior_xhemizygote** posterior probability for the x-hemizygote segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior
 - j_xy** reminder of the number used for the xy segregation type (always 5)
 - posterior_xy*** posterior probability for the xy segregation type based on the mean of the posterior probabilities per site, calculated using the prior
 - geometric_xy*** posterior probability for the xy segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior
 - noprior_xy** posterior probability for the xy segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior
 - j_zhemizygote** reminder of the number used for the z-hemizygous segregation type (always 6)
 - posterior_zhemizygote*** posterior probability for the z-hemizygous segregation type based on the mean of the posterior probabilities per site, calculated using the prior
 - geometric_zhemizygote*** posterior probability for the z-hemizygous segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior
 - noprior_zhemizygote** posterior probability for the z-hemizygous segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior
 - j_zw** reminder of the number used for the zw segregation type (always 7)
 - posterior_zw*** posterior probability for the zw segregation type based on the mean of the posterior probabilities per site, calculated using the prior
 - geometric_zw*** posterior probability for the zw segregation type based on the geometric mean of the posterior probabilities per site, calculated using the prior
 - noprior_zw** posterior probability for the zw segregation type based on the geometric mean of the conditional likelihoods per site, calculated without the prior
 - max_posterior*** the segregation type with the highest posterior probability when using the mean of the posterior site probabilities
 - max_geometric*** the segregation type with the highest posterior probability when using the geometric mean of the posterior site probabilities
 - max_noprior** the segregation type with the highest posterior probability when using the geometric mean of the site conditional likelihoods
- For site lines, the fields are (fields with an asterisk are only given in “expert” output mode) :
 - position** position number
 - alleles** observed alleles (first, second)
 - N11F** observed number of females homozygote for allele 1
 - N12F** observed number of heterozygote females
 - N22F** observed number of females homozygote for allele 2
 - N11M** observed number of males homozygote for allele 1
 - N12M** observed number of heterozygote males
 - N22M** observed number of males homozygote for allele 2
 - fx_max*** frequency of allele 1 on the X, calculated from the subtype with the maximum posterior probability (do not use)
 - fy_max*** frequency of allele 1 on the Y, calculated from the subtype with the maximum posterior probability (do not use)

fx_mean frequency of allele 1 on the X, calculated from the weighed average of all subtypes
fy_mean frequency of allele 1 on the Y, calculated from the weighed average of all subtypes
fz_max* frequency of allele 1 on the Z, calculated from the subtype with the maximum posterior probability (do not use)
fw_max* frequency of allele 1 on the W, calculated from the subtype with the maximum posterior probability (do not use)
fz_mean frequency of allele 1 on the Z, calculated from the weighed average of all subtypes
fw_mean frequency of allele 1 on the W, calculated from the weighed average of all subtypes
subxy_1 the posterior probability of the subtype XY 1 (X-polymorphic, allele 1 is fixed on Y)
subxy_2 the posterior probability of the subtype XY 2 (X-polymorphic, allele 2 is fixed on Y)
subxy_3 the posterior probability of the subtype XY 3 (Y-polymorphic, allele 1 is fixed on X)
subxy_4 the posterior probability of the subtype XY 4 (Y-polymorphic, allele 2 is fixed on X)
subzw_1 the posterior probability of the subtype ZW 1 (Z-polymorphic, allele 1 is fixed on W)
subzw_2 the posterior probability of the subtype ZW 2 (Z-polymorphic, allele 2 is fixed on W)
subzw_3 the posterior probability of the subtype ZW 3 (W-polymorphic, allele 1 is fixed on Z)
subzw_4 the posterior probability of the subtype ZW 4 (W-polymorphic, allele 2 is fixed on Z)
logL_autosomal* the conditional log-likelihood of the autosomal segregation type
posterior_autosomal* the posterior probability of the autosomal segregation type
noprior_autosomal the posterior probability of the autosomal segregation type without prior
logL_haploid* the conditional log-likelihood of the haploid segregation type
posterior_haploid* the posterior probability of the haploid segregation type
noprior_haploid the posterior probability of the haploid segregation type without prior
logL_paralog* the conditional log-likelihood of the paralogous segregation type
posterior_paralog* the posterior probability of the paralogous segregation type
noprior_paralog the posterior probability of the paralogous segregation type without prior
logL_xhemizygote* the conditional log-likelihood of the x-hemizygous segregation type
posterior_xhemizygote* the posterior probability of the x-hemizygous segregation type
noprior_xhemizygote the posterior probability of the x-hemizygous segregation type without prior
logL_xy* the conditional log-likelihood of the xy segregation type
posterior_xy* the posterior probability of the xy segregation type
noprior_xy the posterior probability of the xy segregation type without prior
logL_zhemizygote* the conditional log-likelihood of the z-hemizygous segregation type
posterior_zhemizygote* the posterior probability of the z-hemizygous segregation type
noprior_zhemizygote the posterior probability of the z-hemizygous segregation type without prior
logL_zw* the conditional log-likelihood of the zw segregation type
posterior_zw* the posterior probability of the zw segregation type
noprior_zw the posterior probability of the zw segregation type without prior
j_max* the segregation type with the highest posterior probability
j_max_noprior the segregation type with the highest posterior probability

- Next, the actual SDpop outputs are given per contig and site, with their fields corresponding to the header line.

Note that many of the output fields are given to provide the possibility to understand the calculations, and for debugging purposes.

3 Interpreting the output

3.1 SDpop's parameter estimates

Parameters estimated during the likelihood maximization are

- the proportions π of sites of each segregation type,
- if applicable, the proportion ρ of subtypes within several segregation types
- the error rate
- the log-likelihood and BIC

It is important to note that SDpop only allows to calculate which one of the segregation types is the most probable given the data; if none of the segregation types is appropriate, this will not be visible in the proportions or the posterior probabilities, as they always sum to 1. It is thus recommended to carefully inspect the parameter estimates, particularly the proportions π . SDpop assumes that autosomal genes are in Hardy-Weinberg equilibrium, and although this is a strong hypothesis, most panmictic populations will have only slight deviations from this equilibrium, and those will not cause inferences of other segregation types. Thus, if π_A , the proportion of sites under autosomal segregation, is low, inspecting the data more carefully is recommended. The estimates of the other π values might give a clue to what is happening.

A high proportion of sites under haploid segregation could indicate:

- presence of mitochondrial, chloroplastic or bacterial sequences (real haploidy);
- high level of redundancy in the mapping reference (alleles will not be mapped together but on different positions in the reference);
- individuals are sampled from different, isolated or distant populations (lack of heterozygotes);
- genotyping errors (failure to correctly identify heterozygous genotypes, possibly due to low coverage).

In the first two cases, the positions inferred as being under haploid segregation are expected to occur in proximity (i.e. in the same contig or genome region), while in the latter two, haploid sites would appear more scattered throughout the genome.

A high proportion of sites under paralogous segregation could indicate:

- recent genome duplication (true paralogy);
- usage of a too distant reference for mapping;
- mapping to a reference from which many genes are missing.

SDpop is not designed to obtain correct estimates of the error rate; the error rate is modeled because a model without errors would not produce sensible estimates for several segregation types if an error nevertheless occurs. However, if no prior information about the error rate is known, this rate can be optimised. But since the allele frequencies are directly estimated from the data without optimisation, they do thus not incorporate the possibility of error. As a consequence, error rates are almost certainly underestimated by SDpop.

3.2 Post-optimization estimates

SDpop optimizes the parameters by iteratively calculating probabilities per site. Upon convergence, the posterior probabilities per site corresponding to the optimized model are outputted, as well as a few other values. Users are invited to look at some sites more closely to develop an idea of how SDpop works.

The posterior probabilities for the segregation types per site are obtained by multiplying the conditional likelihood by the prior (i.e., the optimized parameter value), and normalizing all probabilities to sum to one. Thus, in the case of very large differences between the priors, for instance if the proportion of autosomal sites is estimated to be close to 1, and the proportion of XY sites close to 0 (let's say 1.0×10^{-5}), the posterior probability for autosomal segregation might be higher than the one for XY segregation even if the data look like XY segregation, and the conditional probability for XY is a lot higher than the one for autosomal segregation. As an example, consider a simulation with a very small proportion of sex-linked contigs and very recent suppression of the recombination.

SDpop estimates π_A , the proportion of sites under autosomal segregation, to be 0.999946, and π_{XY} , the proportion of XY sites, to be 0.000054. One site in this simulation (20 individuals per sex were used) has 20 females and 14 males homozygous for the same allele, and 6 heterozygous males. The conditional log-likelihood for the autosomal segregation type is -17.2 , and for the XY segregation type, -13.0 ; this means that the XY segregation type is about 67 times ($e^{-13.0}/e^{-17.2}$) more likely than the autosomal one. However, the parameter estimates of π give the prior expectation (i.e., without looking at the data) that the autosomal segregation is about 19000 (π_A/π_{XY}) times more likely than XY segregation. The posterior probabilities, taking both the priors and the conditional likelihoods into account, are 0.996321 for autosomal segregation, and 0.003679 for XY segregation, i.e., autosomal segregation would be 270 more likely given the observed data and the model parameters.

Because the posterior probabilities per site depend on the parameter estimates of the proportions π , it is not straightforward to define a threshold above which a SNP should be considered sex-linked. By default, SDpop gives the segregation type with the highest posterior probability, but this classification should not be used without any precaution. Also, comparing the number of SNPs exceeding a certain posterior probability threshold between runs with different parameter estimates is not straightforward. Users might find it more informative to calculate posterior probabilities with non-informative priors, i.e., considering that all segregation types are equally likely a priori. This can easily be done from SDpop's output; this amounts to comparing the conditional likelihoods.

The recommended way to calculate contig-wise posterior probabilities indeed uses non-informative priors, and is based on the geometric mean of the conditional likelihoods. The geometric mean also has the advantage that sites with the most information, i.e. those with the largest differences in the conditional likelihood of the segregation types, contribute more than sites for which the conditional likelihoods are similar for all segregation types.

Another type of estimate that can be useful for further analysis is the the predicted frequency of alleles in X and Y (or Z and W) copies of sex-linked genes. These frequencies are given for all sites, regardless of their posterior probability or their conditional likelihood. Two ways to calculate these frequencies are implemented. The recommended way considers the empirical frequencies under each subtype of XY (or ZW) segregation, and calculates an average based on the posterior probability of each subtype. Alternatively, one could uniquely use the frequencies corresponding to the subtype with the highest posterior probability. By doing so, one would be sure that polymorphism is assigned to one of the copies only, but this is not recommended, as it would hide model uncertainty.

3.2.1 Producing gametolog consensus sequences

The predicted frequency of the alleles of each SNP on the gametolog copies can be used to produce gametolog consensus sequences. The utility `wxyz_genotyper` is designed to do this based on sdpop's output and a genotype file in `reads2snp`'s `gen` format or a `vcf` file. **Importantly**, the `gen` or `vcf` files should contain information about all sites, not only variant sites. Usually, `vcf` files are filtered to contain only variant sites; usage of such files will yield consensus sequences of the variant sites only.

`wxyz_genotyper` selects contigs from the sdpop output file that have a probability higher than a certain threshold to be XY or ZW, and looks for the monomorphic sites in the genotype file. Ideally, one should thus use the same genotype file in this step as the one that was used by `popsum` to create the `cnt` file. The field in sdpop's output file that should be used for comparison with the probability threshold can be chosen by the user. I recommend using the fields "noprior_xy" or "noprior_zw".

The user should also define two cutoff values for the frequencies of the alleles:

- a threshold value above which an allele is considered as fixed. Such alleles are printed in capital letters in the output consensus sequence.
- a threshold value above which an allele is considered as the majority allele. Such alleles are printed in lowercase letters in the output consensus sequence.

Recommended use of this program for an XY system is:

```
wxyz_genotyper sdpopfile genfile r outputfile 0.8 0.95 0.6 noprior_xy XY mean
```

This will produce X and Y consensus sequences for all genes for which the value of `noprior_xy` is larger than 0.8. In these consensus sequences, alleles with a frequency higher than 0.95 are considered fixed and printed in capital letters, and those with frequencies higher than 0.6 (but lower than 0.95) will be printed in lowercase letters. If the frequency is lower than 0.6, an "n" (lowercase) will be outputted. If the genotype file has some polymorphism that has not been taken into account by SDpop (e.g. because only genotypes of one sex were known), the output nucleotide will be "X". The argument "XY" indicates X and Y frequencies should be read, and the argument "mean"

that frequencies averaged over all subtypes should be used (i.e., with “XY mean”, wxyz_genotyper will use the fields “fx_mean” and “fy_mean”).

The output file has the following structure:

- a header line: “#contig_name N_poly_sites length reference_length snp_genotyped_length non_snps non_snp_length N_fixed_diff N_refalleles pi divergence”. This corresponds to the fields in the lines that start the contig. These fields are:

#contig_name name of the contig. It ends with _X for X or Z consensus sequences, and with _Y for Y or W consensus sequences.

N_poly_sites number of polymorphic sites

length total length of the sequence

reference_length length of the reference sequence

snp_genotyped_length number of genotyped non-polymorphic and SNP sites

non_snps number of polymorphisms that are not SNPs (including indels)

non_snp_length total length of the above

N_fixed_diff number of sites with fixed XY or ZW differences

N_refalleles number of sites with fixed differences for which the allele present in this sequence was in the reference

N_total total number of sites in the contig

pi the nucleotide diversity

divergence the XY or ZW divergence

- Four lines for each XY or ZW gene:
 - a line with the name and the values of the aforementioned fields for the X or Z consensus sequence, starting with “>”
 - a line with the X or Z consensus sequence itself
 - a line with the name and the values of the aforementioned fields for the Y or W consensus sequence, starting with “>”
 - a line with the Y or W consensus sequence itself

Note that information about reference alleles can only be used when the vcf file is constructed in such a way that the first allele is the reference allele.

Appendix: obtaining the input vcf files

For the whole suite of programs to work, especially the `wxyz_genotyper`, I recommend using vcf files that also include information about non-variant and missing sites. Thus, a site at which all alleles are the same as the reference allele should have “0/0” for all individuals, while a site with only missing data should have “./.” for all. Most genotyping pipelines discard these sites, which leads to substantially smaller vcf files, but the distinction between missing and non-variant sites is essential to infer haplotypes, divergence and levels of polymorphism.

Furthermore, as site-wise posterior probabilities have a lot of variation, we would like to aggregate those into contig-wise posterior probabilities (using the geometric mean without priors). If one has a chromosome-level assemble (as for the human genome, for instance), calculating chromosome-wide posterior probabilities makes little sense. Thus, we would like, for instance, to calculate posterior probabilities per gene or per exon, and thus need to change the vcf file in order to group sites per gene, not per chromosome.

bcftools

This is the pipeline used on the 1000 genomes data analyzed in the manuscript. The general idea is to merge the samples (individuals) with `samtools mpileup`, to call genotypes with `bcftools call`, to extract the exons and to replace chromosome names with the gene names.

Here is the detailed pipeline I used, with all the commands. It can surely be done faster and better by a more experienced bioinformatician, and it is very likely that this pipeline is outdated by the time you read this¹. I assume all `bam` files have been obtained. `$reference` is the genome reference file which has been indexed to obtain `$reference.fai`.

1. `mpileup`. This is typically single-thread in `bcftools`, but the `samtools` website² provides a solution for parallelization. The file `$bamlist` used here is simply the list of `bam` files to be used, one per line (e.g. `ls *.bam > bam.list`)

```
vcfutils.pl splitchr -l 500000 $reference.fai \
| xargs -i echo samtools mpileup -C50 -m3 -F0.0002 -DSuf $reference -r {} -b $bamlist \
| bcftools view - \> part-{}.bcf | parallel -j $nthreads
```

I used `vcfutils.pl`, `samtools` and `bcftools` versions 1.8.

One then needs to merge the vcf files created by each parallel process:

```
for j in `cut -f1 $reference.fai | tail -n +3`
do
  for i in `ls part-$j\:*`
  do
    coord=`echo $i | sed -e 's/part-1:\([0-9]*\)-.*/\1/'`
    echo $coord $i
  done | sort -n | cut -d' ' -f2 > chr${j}.list
  bcftools concat -f chr${j}.list -Oz -o chr${j}.vcf.gz
done
```

One now has one vcf file per chromosome. These should be sorted and indexed:

```
for i in `ls chr*.vcf.gz | grep "chr[0-9]*.vcf.gz"`
do
  echo "mv $i ${i}.unsorted ; bcftools sort -T /tmp/${i} -Oz -o $i ${i}.unsorted; bcftools index $i"
done | parallel -j $nthreads
```

2. Genotype calling. Only call genotypes in the targeted regions, described in a bed file `$targets_bed`. Note that the chromosome names need to correspond to those in the vcf files (in the 100 genome case, I had to remove the string “chr” in each chromosome name).

¹Note added 17 January 2023: yes, the pipeline is outdated. For example, `mpileup` is now part of `bcftools`, not of `samtools`. I hope to update this notice soon.

²<http://samtools.sourceforge.net/mpileup.shtml>

```
for i in `ls chr*.vcf.gz | grep "chr[0-9X]*.vcf.gz"`
do
echo "bcftools call -m -R $targets_bed -Oz -o ${i}/vcf/called.vcf} $i"
done | parallel -j $nthreads
ls *.called.vcf.gz > toconcat
bcftools concat -f toconcat -Oz -o genome.vcf.gz
```

Normalize and left-align indels. Multiallelic sites should represent only one line in the vcf file.

```
bcftools norm -f $reference -m +any genome.vcf > genome_norm.vcf
```

3. One can of course run SDpop on this vcf, but chromosome-level posterior scores aren't very interesting. I therefore recommend cutting the genome up into smaller pieces. A "natural" way to cut it up, is according to the genes, as we've used the exome-sequencing data here. `exons.bed` is a tab-separated file with information about the location of the protein coding genes in the genome.

First, get the header of the original vcf file:

```
bcftools view -h genome_norm.vcf | grep -v contig > genome_norm.genenames.vcf
```

Then, extract the exons. We'll pipe this through `awk` to reorder the columns according to vcf style. Some exons might be present twice, so pipe through `sort` and `uniq` afterwards:

```
bedtools intersect -a genome_norm.vcf -b exons.bed -wa -wb | \
awk '{printf "%s\t", $NF; for(i=2;i<NF-3;i++){printf "%s\t", $i} printf "\n"}' | \
sort -k 1,1 -k2,2n | uniq >> genome_norm.genenames.vcf
```

4. It's best to remove samples you'll not use from the vcf, as they might contain extra genetic variation. Note that SDpop only works on dimorphic sites, so sites with more than two alleles will be discarded. Also, for haplotype reconstruction, alleles not taken into account in the SDpop analysis will result in missing information ("X"). You can also do additional filtering here, for instance for the number of reads supporting a genotype.

```
vcftools --keep individuals.txt --minDP 10 --vcf genome_norm.genenames.vcf -c --recode | \
bcftools view -a > selection.vcf
```