SHOREmap Manual

Version 2.0

Karl J.V. Nordström
Geo Velikkakam James
Stephan Ossowski
Korbinian Schneeberger

@ Max Planck Institute
07.05.2012

Contents

SHOREmap is an analysis pipeline for mapping and mutant identification in one step. See orginal publication: "SHOREmap: simultaneous mapping and mutation identification by deep sequencing", Schneeberger, Ossowski et al., Nat Meth, 2009. Confidence interval calculations have been introduced here: “Synteny-based Mapping-by-Sequencing enabled by Target Enrichment”, Galvão, Nordström et al., Plant J, 2012, and analysis of isogenic mapping populations using SHOREmap was first described here: “Fast isogenic mapping-by-sequencing of EMS-induced mutant bulks” Hartwig, Velikkakam James et al., under review.

In order to run SHOREmap you need to provide information of a resequencing analysis. The files need to be prepared in the right format. Any alignment and consensus calling tool can be used, if its output is converted into the right way for SHOREmap.

This README is updated in order to describe SHOREmap version 2.0.

1 Prerequisites

1.1 Download and Installation

Download SHOREmap from http://shoremap.org. After storing the SHOREmap_release_2.0.tar.gz somewhere on your hard drive, use the terminal application and change the working directory to the directory where you stored SHOREmap. Typing

```
tar zxf SHOREmap\_release\_2.0.tar.gz
```

will unpack SHOREmap. There is no further need for installing.

1.2 Installation of R

SHOREmap makes use of R for data visualization. R has to be installed and the installation path has to be added to the $PATH environmental variable. Furthermore, the new confidence interval analysis also requires the R-package bbmle to be installed (however SHOREmap can be executed without). After installing R, it is available from CRAN:

```
R # start R
> install.packages(‘‘bbmle’’) #in R, install the package
> q() # quit R
```

1.3 Installation of PERL and BioPerl

SHOREmap is written in PERL. To run SHOREmap you need to have PERL and BioPerl installed on your computer.

2 SHOREmap

SHOREmap allowes analysis of data from conventional mapping populations generated by outcrossing to a diverged parent and from isogenic mapping population generated by backcrossing to the non-mutagenized progenitor. Depending on the type of mapping population, select one of the following commands:

2.1 SHOREmap *outcross*

2.1.1 Description of parameters

Using the resequencing results of bulked F2 (or F3) individuals (that were selected for a particular phenotype), SHOREmap*outcross* tries to identify a region with a particular parental (mutant or non-mutant) allele frequency. Usually (i.e. in the case of recessive mutations) the allele frequency is 1 for the mutant background, however this can be adjusted.

In addition to simple sliding window-based summary methods, SHOREmap*outcross* can define a probabilistic mapping interval based on the likelihood that all markers within a given window agree with the target frequency. In this section, we describe the parameters that can be adjusted to tune the algorithm to your problem. Note that all, except the three first, of the below described parameters have stable default values and this is the minimal SHOREmap outcross command:

```
> SHOREmap.pl outcross --consensus=consensus.txt --marker=marker.txt
--chrsizes=chrsizes.txt
```

The input data is defined by three parameters that inidicate where SHOREmap can find the input files. `--chrsizes` indicates a file describing the length of each reference sequence in bp, `--marker` a file listing the marker positions and, `--consen` a file describing the base counts per marker position. Latter is the typical output of a resequencing program, but needs to be formated in a SHOREmap specific way. See tables below for the required file formats.

First, SHOREmap filters the data, and markers with extreme read counts are discarded. These boundaries of accepted coverages are defined by `--min-coverage` and `--max-coverage`. As the data still can contain artifacts from the short read alignment step, in particular the reads that are not mapped to the correct position (cross-mapping). Marker positions with cross-mapped reads might not reveal the right allele frequency. Thus, the SHOREmap can run an outlier removal step (`-outlier`). For this, SHOREmap estimates the allele frequency in a (by default 200 kb region) surrounding each marker. This window size can be adjust by `--outlier-window-size`. SHOREmap then tests whether the marker could be drawn from the estimated frequency. If this probability is lower than a cutoff, set with `--outlier-pvalue`, the marker is not used in further calculations. This exclusion includes the testing of less extreme outlier candidates as well. If single markers a visualized in the final plot, outliers are marked with a gray cross, compared to the blue dots of the used markers. Outlier removal increases runtime a lot (however might be worse it).

SHOREmap thentries to pinpoint the exact location of the mutation by sliding window-based calculations. By default this is done with the peak eastimation called boost, but can be changed to the r-value calculation described in the initial SHOREmap publication. Both calculations will estimate a peak, which can be used as starting point for SHOREmap annotate (see below). In some cases, however, it is desired to have clear boarder like a conventional mapping interval would provide. SHOREmap implements a confidence interval calculation that identifies the maximum extent of the region that represents the targeted allele frequency. Such confidence interval can be interpreted as mapping intervals.

Confidence interval calulations are computationally intense and are not performed by default, use `--conf-int` to switch on confidence interval calculations. The first step of the analysis is finding an starting region for the calculation of the final confidence interval. This is achieved by comparing the target frequency (`--target`) to estimates of the frequencies in sliding windows. If needed, `--mis-phenotyped`, can be set to introduce a tolerance for deviation from the target frequency (which correspondes to the expected mis-phenotyping rate). The window size is given by `--peak-window-size`. The start points of the windows are by default separated by 10,000 bp, which can be adjusted by setting `--peak-window-step`, however this can have drastic effect on the runtime. Finally, a confidence mapping interval at a given confidence level (`--conf`) is calculated.

The results are reported in a flexible graphical pdf-format. By default, it will be two graphs per reference sequence. The top panel contains the marker frequencies as dots and crosses (outliers), the windowed average frequency as a black line and, if it exists, an interval marking the mapping interval. The bottom panel displays the coverage of a marker, split into the two parental alleles. The plotted region can be limited by specifying a set of zooming parameters; `--chromosome` selects the chromosome, `--begin` and `--end` limits the x-axis of the top panel and `--minfreq` and `--maxfreq` controls the range of the y-axis in the top panel. The windowed average can be adjusted with `--window-size` and `--window-step`. Allele frequencies estimated at single markers can be shown with `-marker`.

In addition, there are three parameters, `--referrors`, which defines a set of markers that shall be ignored as the are likely to contain erroneous marerks, `--background2` shifts the calculation to consider the second sample as mutated and finally, `--verbose`.

2.1.2 Recommendations

The default parameters have proven stable for Arabidopsis thaliana data. As mentioned above, there are different ways to discard bad markers. Either by a coverage that deviates strongly from the other markers. In your data, you can identify the existence of such markers by studying the bottom panels for each chromosome. If these values for single markers deviate heavily from the expected coverage, it is recommended to look into these markers and if possibly discard them by setting a maximum coverage with `--max-coverage` (or simple to remove them from the input files completely). The other alternative is the outlier removal. Here, the window-size, `--outlier-window-size`, is important. It should be large enough to incorporate enough markers to give a reliable frequency estimate for the region surrounding the tested marker, without spanning multiple recombinations.

The window size of the confidence calculations, `--peak-window-size`, is also affected by the recombination rate. The difference is that this is only an initial analysis, and you do not need the reliability needed to remove outliers. Furthermore, the subsequent confidence interval calculation benefits from a smaller window size. Hence, this is limited by the marker density rather than the recombination rate. By default, a window must contain 10 markers to be considered and the window size should be set in relation to this. If this is set too small the interval calculation might be evoced at a wrong starting point, which in the worst case can result in an incorrect interval. Still, if your markers are sparse, it is possible to reduce the minimum number of markers in a window by setting the `--min-marker` flag.

If no interval is predicted, even though there is a peak on one of the chromosomes, it is worth checking whether this peak reaches the given target frequency. This can be done by zooming to the region, as described above. Adjustments can be performed by allowing a small level of mis-phenotyping (`--mis-phenotyped`). If the window with the frequency closest to the target has an estimated frequency within this given tolerance, the target frequency is adjusted to this value. As this is an automatic estimation, it is therefore quite rough. It is recommended to manually estimate the degree of mis-phenotyping from the graph and then re-run SHOREmap outcross with a new target frequency.

2.1.3 SHOREmap.pl outcross, command description

Mandatory:			
–chrsizes	STRING		Tabbed file with chromosome names and sizes.
–folder	STRING		Output folder (will be created and should not exist)
–marker	STRING		Marker file.
–consen	STRING		Consensus file.

Filter:			
–min-marker	INT	10	Minimum number of markers to be considered, either in a window, or in a confidence interval
–min-coverage	INT	0	Filter single markers with a read count below this value
–max-coverage	INT	Inf	Filter single markers with a read count above this value
-outlier			Filter outliers
–outlier-window-size	INT	200000	Window size to assess local allele frequency used for outlier removal
–outlier-pvalue	DOUBLE	0.05	p-value used for outlier removal

Confidence interval:			
–target	DOUBLE	1.0	Target allele frequency of the mutant allele (usually 1 for recessive mutants)
–mis-phenotyped	DOUBLE	0.0	Accepted degree of putatively mis-scored plants
–peak-window-size	INT	50000	Window size for initial peak prediction
–peak-window-step	INT	10000	Step size between windows used for initial peak prediction
–conf	DOUBLE	0.99	Confidence level for the confidence interval calculation

Zooming:

–chromosome	INT		Zoom plot to chromosome ..
–begin	INT		.. from here ..
–end	INT		.. to here, with a ...
–minfreq	INT		.. minimal to ..
–maxfreq	INT		.. maximal frequency.

Visualization:

–window-size	INT	50000	Window size for plotting
–window-step	INT	10000	Step size between windows used for plotting
-no-interval			Switch off confidence interval calculation
-no-marker			Do not plot single markers
-plot-r			Plot frequency calculation (“r”)
-plot-boost			Plot frequency calculation (“boost”)

Optional:

–referrors	STRING	Reference errors file
-background2		Mutation is in second parent
-verbose		Be talkative

2.2 SHOREmap *backcross*

SHOREmap *backcross* is used to analyse resequencing data of bulked individuals of isogenic mapping populations generated by backcrossing of a mutant to its progenitor. In contrast to conventional mapping population, no a priori marker do segregate in isogenic mapping populations. However, mutagen-induced markers can be used as novel markers. Usually the number of mutagen-induced changes is drastically lower than naturally occuring variation. Thus, SHOREmap cannot apply sliding window analyses or confidence interval calculations. Instead, SHOREmap *backcross* removes the background markers, in order to produce a list of purely mutant-specific mutations and visualizes them across the genome with respect to their frequency. Such vizulations are typically enough to identify a rough mapping interval (usually on the size of a couple of Mb), and to perform annotations of the mutations in the region under selection. Along with the visualization files, backcross analysis produce filtered marker files based on user specified criteria.

```
SHOREmap.pl backcross --marker marker.txt --chrsizes chrsizes.txt --out folder
```

After each filtering step, output files are written into the output folder. The file names indicate the criteria used for filtering.

Mandatory:

–marker	STRING		Point mutation file. See below for file format.
–chrsizes	STRING		Tab delimited file with chromosome names and sizes.
–out	STRING		Output folder. This folder will be created automatically.

Optional:

–marker-score	INT	25	Minimum quality score for filtering foreground marker.
–marker-freq	INT	20	Minimum frequency of foreground marker
–marker-cov	INT		Minimum read support of foreground marker
–bg	STRING		Background marker file. Multiple files can be given by comma separating their locations.
–bg-score	INT		Minimum quality score for filtering background markers.
–bg-freq	INT	20	Minimum frequency of background marker
–bg-cov	INT		Minimum read support of background marker

Plotting options:

-no-summary	Turn off ploting all chromosome in single page as summary
-no-filter	Plot all markers after background corretion without considering the filtering criterias used above
-non-EMS	Plot non-canonical EMS (as crosses) mutations along with canonical EMS (as points)
-other-mutagen	Plot all markers (marked as stars)without differentiating canonical EMS mutations. Applied when non-EMS mutagen used for screening.
-verbose	Be talkative

3 SHOREmap *annotate*

SHOREmap *annotate* outputs the predicted effects of mutations on genes and their putative encoding of amino acids.

Example:

```
>SHOREmap_annotate.pl --snp=mutations.txt --chrom=4 --start=16000000 --end=17000000
--genome=TAIR10.v1.fa --gff=TAIR10.gff
```

`--snp` is the file listing all mutations identified in the resequencing project and that need to be annotated. `--chrom, --start,` and `--end` indicate the interval in which the mutation shall be annotated. `--genome` is the reference sequence in fasta format (usually the file used to align the reads against). `--gff` is a GFF formatted description of the reference annotation. `--referr` describes all the positions, which will be discarded in the final output.

`--del` and `--ins` describe small indels found in the read mapping and which can be annotated as well.

Mandatory:		
–snp	STRING	Point mutation file.
–chrom	STRING	Chromosome to be annotated from...
–start	INT	...here to...
–end	INT	...here.

Functional annotation:		
–genome	STRING	Reference sequence
–gff	STRING	Gene annotation in GFF format

Optional:		
–referr	STRING	Known errors in the reference sequence.
–del	STRING	Deletions file.
–ins	STRING	Insertions file.

4 File formats

All columns described for all file formats have to be tab delimited. The number at the beginning of a column description indicates the column number.

4.1 Input files

4.1.1 Consensus file

	Column	Description
1	Chromosome	Chromosome identifier
2	Position	Position in the reference sequence
3	Reference allele	Allele within the reference sequence (currently not used, but taken from the marker file).
4	Coverage	Number of alignment covering this position.
5	A	Number of read alignments supporting A allele.
6	C	Number of read alignments supporting C allele.
7	G	Number of read alignments supporting G allele.
8	T	Number of read alignments supporting T allele.

4.1.2 Marker file

	Column	Description
1	ID	Sample ID.
2	Chromosome	Chromosome identifier
3	Position	Position in the reference sequence
4	Allele 1	Either A, C, G or T, describing the allele of the wild-type parent.
5	Allele 2	Either A, C, G or T, describing the allele of a diverged parent.

4.1.3 Chromosome sizes

	Column	Description
1	Chromosome	Chromosome identifier
2	Length	Length of the chromosome (=fasta entry in the reference sequence) in bp.

4.1.4 Reference errors

	Column	Description
1	Chromosome	Chromosome identifier
2	Position	Position in the reference sequence

4.1.5 Point mutation file

	Column	Description
1	ID	Sample ID.
2	Chromosome	Chromosome identifier
3	Position	Position in the reference sequence
4	Reference allele	Either A, C, G or T, describing the allele of the wild-type parent.
5	Mutant allele	Either A, C, G or T, describing the mutant allele.
6	Quality score	Base calling quality score (from a resequencing program)
7	Support	Total number of reads aligned to this mutation site.
8	Frequency	Allele frequency of the mutant allele. (Usually calculated as the fraction of reads supporting the mutant divided by the total number of reads.

4.1.6 Genome

Fasta file of the reference sequence.

4.1.7 Annotation GFF

See the general description of GFF files.

4.2 Output files

4.2.1 Prioritized SNP list

	Column	Description
1	Chromosome	Chromosome identifier
2	Position	Position in the reference sequence
3	Reference allele	Either A, C, G or T, describing the allele of the wild-type parent.
4	Mutant allele	Either A, C, G or T, describing the mutant allele.
5	Support	Total number of reads aligned to this mutation site.
6	Frequency	Frequency of the mutation
7	Quality score	Base calling quality score (from a resequencing program)
8	Type	Either this is a NEWSNP or a known REFERROR
9	Sequence feature	Type of DNA that is affected.
10	ID	Gene identifier (if mutation residesin gene)
11	Isoform	Isoform of the gene
12	Mutation position	Coding sequence position of the change.
13	Codon position	Codon position of the change
14	Type of change	Either syn or nonsyn.
15	Reference amino acid	
16	Mutation-induced amino acid	

We appreciate any kind of feedback. Please do not hesitate to contact us in case of any problems or questions. As SHOREmap undergoes modifications constantly, we usually run different versions of SHOREmap ourselves, which might lead to unidentified bugs in the release versions. Sorry for that.

Korbinian Schneeberger <korbinian.schneeberger@mpipz.mpg.de>