SNPbinner

SNPbinner is a Python 2.7 package and command line utility for the generation of genotype binmaps based on SNP genotype data across populations of recombinant inbred lines (RILs). Analysis using SNPbinner is performed in three parts: `crosspoints`, `bins`, and `visualize`.

Table of Contents

Installation and Usage

***SNPbinner requires Python 2.7**. Python 3 is currently not supported.*
The only non-standard dependency of SNPbinner is Pillow, a PIL fork.

To install the SNPbinner utility, download or clone the repository and run

```
$ pip install REPO-PATH
```

Once installed, one can execute any of the commands below like so

```
$ snpbinner COMMAND [ARGS...]
```

Alternatively, without installing the package, one can execute any of the commands below using

```
$ python REPO-PATH/snpbinner COMMAND [ARGS...]
```

Commands

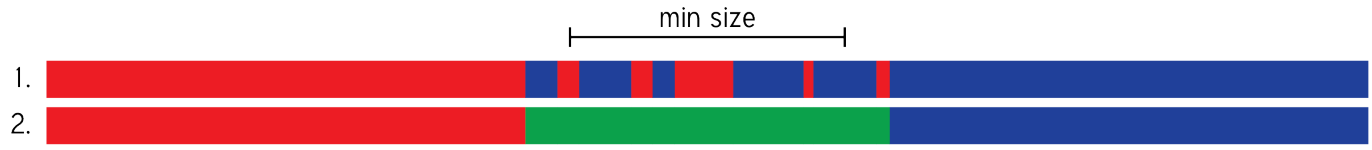
crosspoints

Description

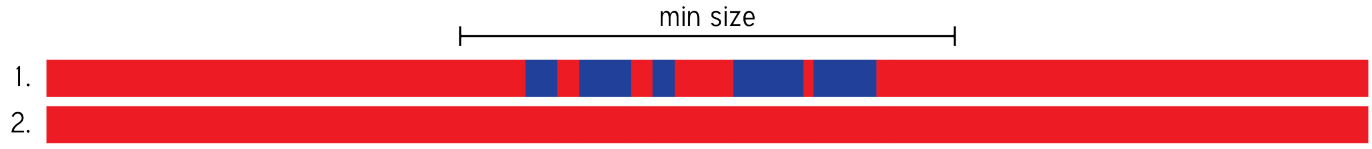
`crosspoints` uses genotyped SNP data to identify likely crossover points. First, the script uses a pair of hidden Markov models (HMM) to predict genotype regions along the chromosome both with (3-state) and without (2-state) heterozygous regions. Then, the script identifies groupings of regions which are too short (based on a minimum distance between crosspoints set by the user). After that it follows the rules below to find crosspoints. The script then outputs the crosspoints for each RIL and the genotyped regions between them to a CSV file.

---

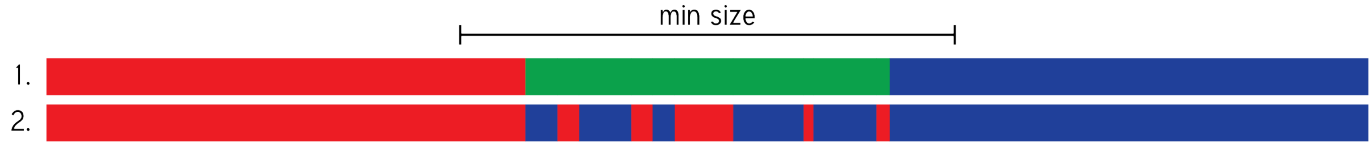
1. If a group of too-short regions is long enough to be its own acceptably-long genotype region, it will be treated as such and assigned the most likely genotype using the 3-state HMM.



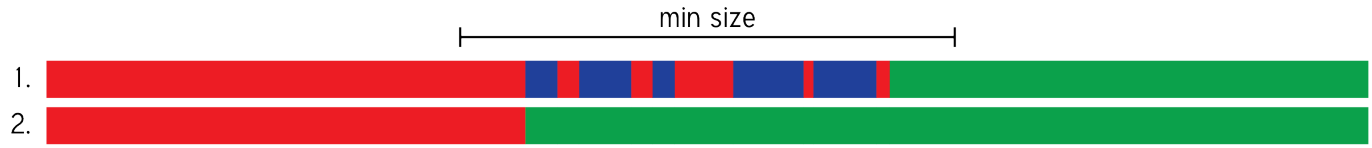
2. If a group of too-short regions is surrounded by regions of the same genotype, all regions within that group are assigned the surrounding genotype.



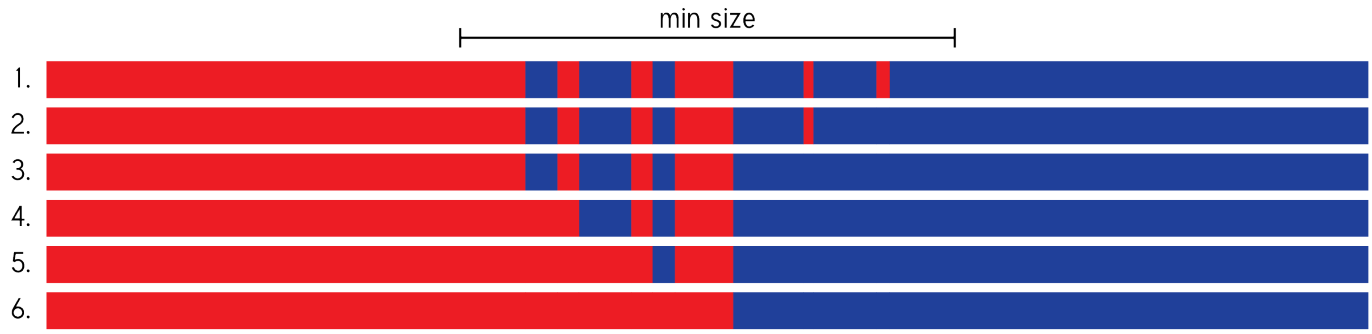
3. If a too-short region has been genotyped as heterozygous by the 3-state HMM, that section is replaced by the regions identified by the 2-sate HMM.



4. If the first or last too-short region is neighboring an acceptably-long heterozygous region, the whole grouping will be assigned the heterozygous genotype.



5. If neither the first or last too-short region is neighboring a heterozygous region, the shortest of those two regions will be assigned to the same genotype as its neighbor. This repeats until the group is empty.



Usage

Running the `crosspoints` command requires an input path, output path, and a minimum size argument. There are also three optional arguments which can be found in the table below.

```
$ snpbinner crosspoints --input PATH --output PATH (--min-length INT | --min-ratio FLOA
```

Required Arguments

		Type	Description
`-i`	`--input`	`PATH`	Path to a SNP TSV, multiple paths, or a glob (e.g. myGenome.chr*.tsv).
`-o`	`--output`	`PATH`	Path for the output CSV when there is a single input, or for a folder when there are multiple.
`-m`	`--min-length`	`INT`	Minimum distance between crosspoints in basepairs. Cannot be used with `min-ratio`.
`-r`	`--min-ratio`	`FLOAT`	Minimum distance between crosspoints as a ratio. (0.01 would be 1% of the chromosome.) Cannot be used with `min-length`.

Optional Arguments

		Type	Description
`-c`	`--cross-count`	`FLOAT`	Used to calculate transition probability. The state transition probability is this value divided by the chromosome length. (default: 4)
`-l`	`--chrom-len`	`INT`	The length of the chromosome/scaffold which the SNPs are on. If no length is provided (or multiple file are being processed), the last SNP is considered to be the last site on the chromosome.
`-p`	`--homogeneity`	`FLOAT`	Used to calculate emission probabilities. For example if 0.9 is used it is predicted that a region b-genotype would contain 90% b-genotype. (Default: 0.9)

Input Format

Sample input file

	Input should be formatted as a tab-separated value (TSV) file with the following columns.
0	The SNP marker ID.
1	The position of the marker in base pairs from the start of the chromosome.
2+	RIL ID (header) and the called genotype of the RIL at each position.

Output Format

Sample output file

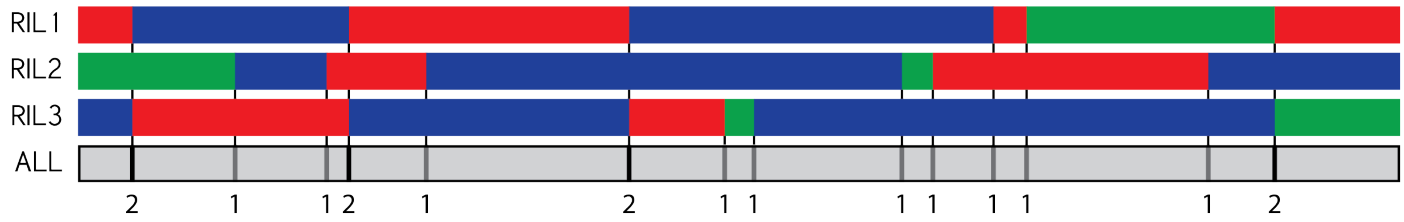
	Output is formatted as a comma-separated value (CSV) file with the following columns.
0	The RIL ID
Odd	Location of a crosspoint. (Empty after the chromosome ends.)
Even	Genotype in between the surrounding crosspoints. (Empty after the chromosome ends.)

bins

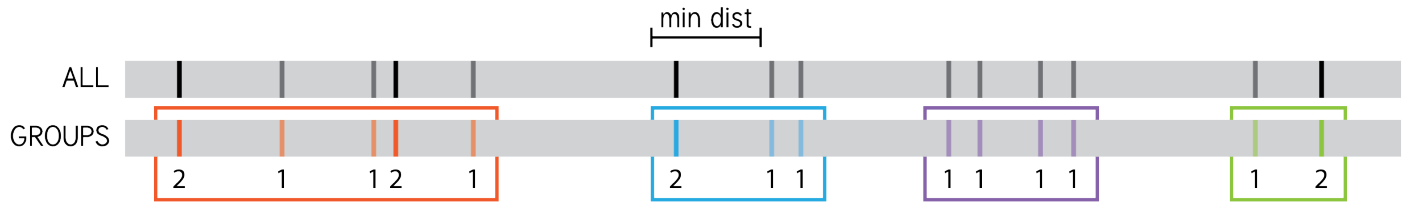
Description

`bins` takes the crosspoints predicted for each RIL and combines similar crosspoint locations to create a combined map of all crossover points across the RILs at a specified resolution. It then projects the genotype regions of the RIL back onto the map and outputs the average genotype of each RIL in each bin on the map. The procedure is as follows. *It should be noted that, to insure the changes are obvious, the illustrations below are showing a map with very low resolution (bin size) and therefore there is significant loss of information. A smaller bin size would create a more accurate map.*

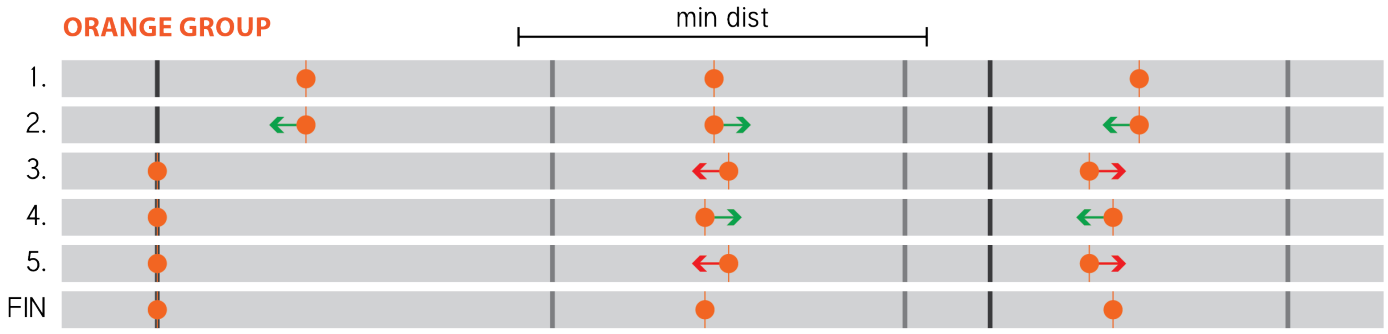
1. The script begins by combining the crosspoints from all lines, including duplicates occurring at the same location.



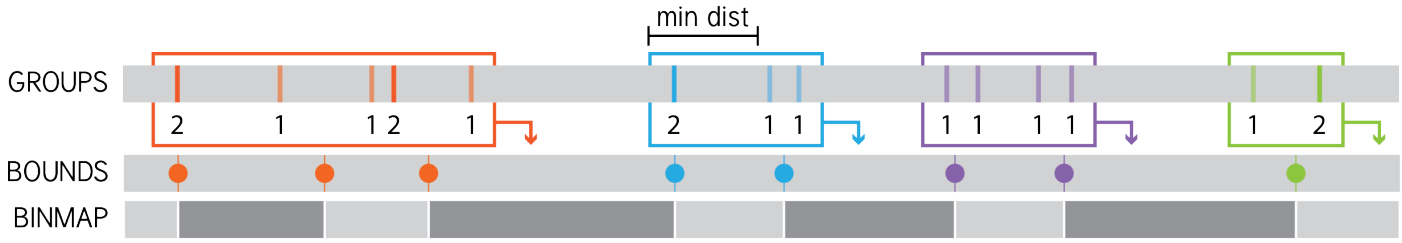
2. Contiguous series of crosspoints are then grouped together if they are closer to a neighbor than the specified minimum bin size.



3. One-dimensional k-means optimization is then used to find the best placement for the bin boundaries (steps 2 and 4 below). This is repeated for every possible number of boundaries that can fit in the span of each group. In order to account for the minimum bin-size constraint, once a possible set of boundaries has been converged upon by the k-means algorithm, each mean is adjusted to insure it is at least the minimum distance from it's neighbors (steps 3 and 4 below). If this enters a cycle instead of converging on a working solution, the script will accept the adjusted boundaries without the second optimization step. Otherwise, optimization continues until a solution is reached with appropriately spaced boundaries. *This k=3 example finishes due to a cycle (steps 3-5).*



4. For each group, the solution with a value of k leading to the least variance from the adjusted means are placed into a list of final boundaries. These boundaries are then used to create bins for the final binmap.



5. Each RIL is then projected onto this bin and the results are output as a CSV. Bins are genotyped as whatever genotype represents a plurality of its contents.

Usage

Running the bins command requires an input path, output path, and a minimum size argument. Optionally, a binmap ID may also be provided.

```
$ snpbinner bins --input PATH --output PATH --min-bin-size INT [--binmap-id ID]
```

Required Arguments

		Type	Description
`-i`	`--input`	`PATH`	Path to a crosspoints CSV, multiple paths, or a glob (e.g. myGenome.chr*.crosp.csv).
`-o`	`--output`	`PATH`	Path for the output CSV when there is a single input, or for a folder when there are multiple.
`-l`	`--min-bin-size`	`INT`	Sets the minimum size (in bp) of each bin.

Optional Arguments

		Type	Description
`-n`	`--binmap-id`	`ID`	If a binmap ID is provided, a header row will be added and each column labeled with the given string.

Input Format

`bins` uses the output from `crosspoints`.
For details, see the `crosspoints` Output Format.

Output Format

Sample output file

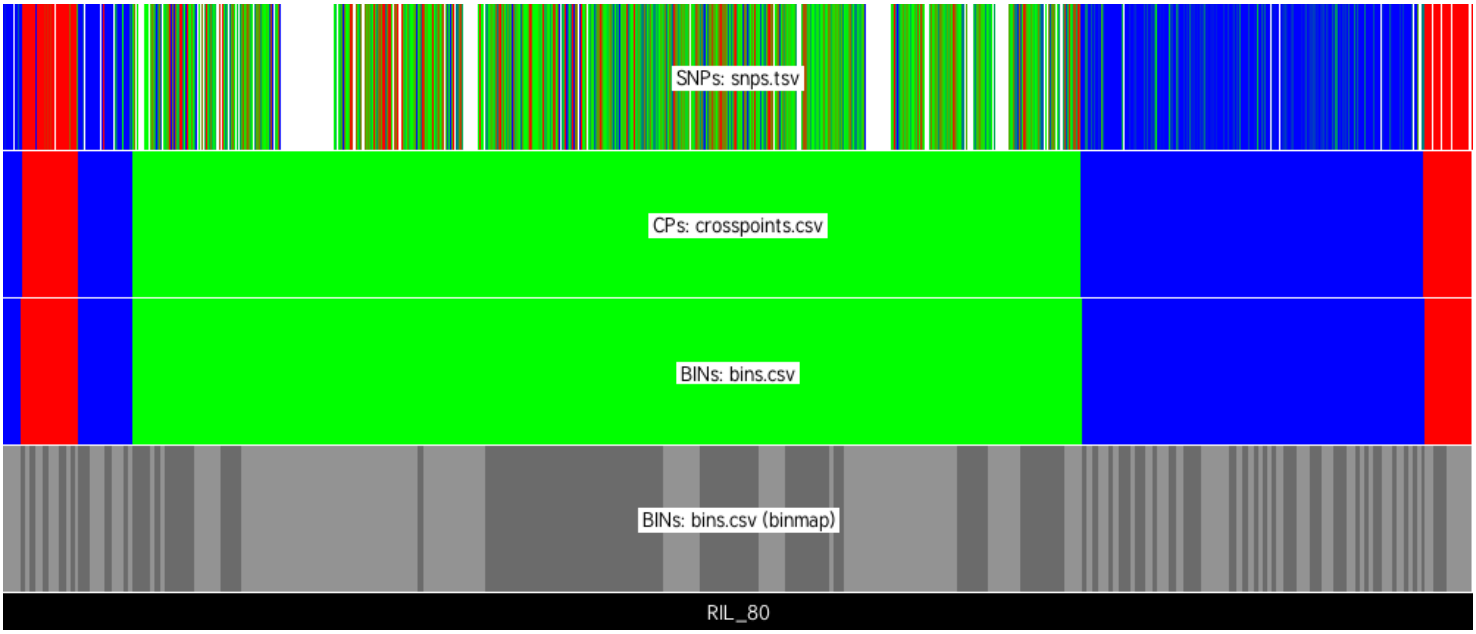
	Output is formatted as a comma-separated value (CSV) file and has the following rows.
0	(Optional) The binmap ID
1	The start of each bin (in base pairs).
2	The end of each bin (in base pairs).
3	The center of each bin (in base pairs).
4+	RIL ID in the first cell, then the genotypes of each bin for that RIL.

visualize

Description

`visualize` plots the inputs and outputs of `bins` and `crosspoints`. It can be used to visually check the results of the above commands to help determine the best values for each of the parameters. It can accept three filetypes (SNP input TSV, crosspoint CSV, and bin CSV). It then parses the files and groups the data by RIL, creating an image for each. In each row of the resulting images, regions are colored red, green, or blue, for genotype *a*, heterozygous, or genotype *b*, respectively. The binmap is represented in gray with adjacent bins alternating dark and light. The script can accept any combination or number of files for each of the different filetypes.

Example



Usage

```
$ snpbinner visualize --out PATH [--bins PATH]... [--crosspoints PATH]... [--snps PATH]
```

Required Arguments

		Type	Description
`-o`	`--out`	`PATH`	Folder to which the resulting images should be saved.

Optional Arguments

		Type	Description
`-b`	`--bins`	`PATH`	`bins` output file to be added to the visualization.
`-c`	`--crosspoints`	`PATH`	`crosspoints` output file to be added to the visualization.
`-s`	`--snps`	`PATH`	SNP (`crosspoints` input file) file to be added to the visualization.