IRiS User Manual

April 22, 2011

Contents

1 Introduction **2**

2 Phase 1: Detecting Recombinations **4**

2.1 Input format 4

2.2 Command Line Parameters 4

2.3 Output format 6

3 Phase 2: Constructing subARG **8**

3.1 Command line parameters 8

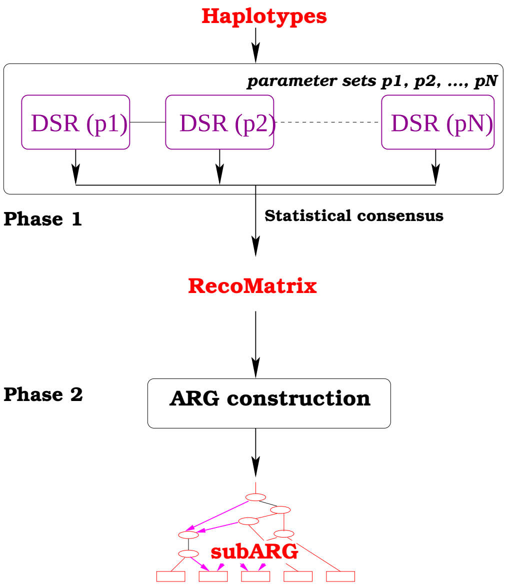
3.2 Output format 8

4 Example **10**

Chapter 1

Introduction

Genetic recombinations play a key role in shaping the chromosomal landscapes. The structure that captures these genetic events as the common evolutionary history of a set samples is called an ancestral recombinations graph (ARG) in population genetics literature. The statistical and combinatoric tools for identification of recombinations in sequences (IRiS) is described here. The reconstructed ARG of a collection of samples is necessarily a subgraph of the true ARG, hence we call it a subARG.



Given a collection of haplotypes, IRiS produces a subARG in two phases. A combinatorial algorithm called the DSR [7] is a model-based approach to detecting recombinations in haplotypes (with a guaranteed approximation factor [6]). In the first phase DSR is run multiple times with different sets of parameters and statistical consensus [3] is derived from them to produce a matrix of recombination information called the recomatrix. This encodes the local topology information of only the high confidence recombination events detected in the first phase. The subARG is constructed from the recomatrix in the second phase [1].

IRiS is implemented in C++ and the binary executable files can be downloaded for common computing platforms from `http://researcher.watson.ibm.com/researcher/view_project.php?id=2303`

The zipped folders contain three files.

- *iris1* - executable file for the first phase
- *iris2* - executable file for the second phase
- *seqs.txt* - an example input file for iris1

If IRiS is used in published analysis, it should be cited as:

> Javed, A., Pybus, M., Melé, M., Utro, F., Bertranpetit, J., Calafell, F., and Parida, L., IRiS: Construction of ARG network at genomic scales, 2011

Chapter 2

Phase 1: Detecting Recombinations

IRiS takes haplotypic data as input. The current implementation does not allow for missing values. Therefore genotypic data needs to be phased and imputed before using the application. The methodology has been tested with haplotypes inferred using *Phase* [8]; and is shown to be robust to genotyping and phasing errors [3].

A unique feature of IRiS is that in addition to the extant sequences, it allows the user to define the haplotype at their most recent common ancestor (i.e. the root of the ancestral recombination graph). This information (if available) is used by DSR only if the defined local ancestral SNP patterns are in concordance with the extant sequences.

2.1 Input format

For n chromosomes genotyped at m biallelic SNPs, the input file should be formatted as

1. The first two lines are blank.
2. The third line contains the ancestral sequence; it may be left blank as well. If the ancestral sequence is defined, the line consists of a haplotype identifier followed by the sequence. SNPs are assumed to be sorted based on their chromosomal positions and in agreement with the extant sequences. Note that blank values are not permitted. Markers with unknown ancestral allele can be indicated by assigning an allele different from the extant sequences.
3. The next four lines are blank.
4. The following n lines correspond to the n input haplotypes. Each line contains two fields: a unique haplotype identifier followed by the haplotype. Additional fields following the haplotype are ignored. The SNPs are assumed to be sorted based on their chromosomal position and consistent across the dataset.

2.2 Command Line Parameters

The binary file `iris1` can be executed with command line parameters:

```
ANC      2222222222



NEW0     1211221122
NEW1     1211221122
NEW2     1211212112
NEW3     1222212112
NEW4     2122112112
```

Figure 2.1: An example input file for five extant sequences genotyped at ten markers with available ancestral haplotype information.

```
iris1 <input> mergePATS=<m> pure=<p> ancestral=<a> nocluster=<n> threshold=<t>
   parental_fact=<f> peak_distance=<d> grain=<g>
```

where:

- `iris1` assumes the name of the input file is <input file name>.txt. And it is in the format described in Section 2.1;
- `m`: patterns within the provided hamming distance will be merged (default 0);
- `p`: 1/0 determines if an ancestral allele should be inferred or not (default 0);
- `a`: 1/0 determines if ancestral haplotype is provided or not (default 0);
- `n`: 0/1 determines whether to cluster or not (default 1);
- `t`: determines the minimum threshold that a recombination must reach across multiple runs to be considered valid (default 42);
- `f`: determines the fraction of the recombination peak that the parental sequences have to reach to be considered valid (default 3);
- `d`: determines the maximum distance between parental and recombinant peaks (default 15);
- `g`: determines the grain sizes. The code assumes that they are separated by underscores (default 5_10_20).

Note that this description can also be retrieved by executing the binary file without any parameters. Otherwise, input file must be defined at command line. If any of the remaining parameters is not set, it is assigned the default value. The default values are set to the parameter values used in [1, 3].

For example, to execute `iris1` using the input file `seqs.txt`, a possible command line is:

```
iris1 seqs mergePATS=0 pure=1 ancestral=1 threshold=42 parental_fact=3
   peak_distance=15 grain=5_10_20
```

```
5 1
          rec_1
          4
          0
NEW0      2
NEW1      2
NEW2      1
NEW3      3
NEW4      0
```

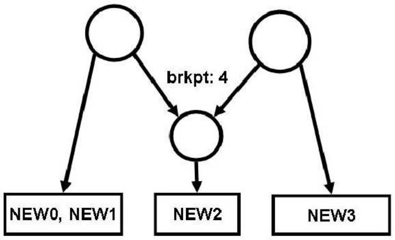


Figure 2.2: An example recomatrix file for 5 sequences indicating their role in a detected recombination *rec*_1. This relation is pictorially shown on the right.

2.3 Output format

The binary file `iris1` generates two text files:

- < *input* > _*recombinations.txt* gives a list of detected recombinations.
- < *input* > *.dat* provides the *recomatrix* which serves as input for *iris2*

Each line in < *input* > _*recombinations.txt* represents a detected recombination. The list of extant descendants of the recombination event along with the estimated breakpoint is defined. The list is sorted based on chromosomal locations.

An example recombination indicating one descendant *NEW2* with the breakpoint location between SNPs 3 and 4 (starting the SNP count from 0) would be

NEW2 4

Note that 0.5 is added to the breakpoint position if it is not clear on which side of the marker it occurs. For example, a value of 6.5 indicates that it could be between SNPs 5 and 6, or 6 and 7.

< *input* > *.dat* contains the recomatrix which defines the local sequence relations articulated by each recombination.

- The first line contains the numbers of haplotypes n and inferred recombinations r.
- The second line contains the list of detected recombinations.
- The third line defines the corresponding breakpoint locations. This list is consistent with < *input* > _*recombinations.txt*.
- The fourth line can be ignored. Currently it contains zeros corresponding to each recombination. It is intended to add flexibility to output more information pertaining to each recombination.
- The remaining n lines represent the haplotypes. Each line starts with the sequence identifier, followed by r entries indicating its role in the corresponding detected recombination.
  - 0: no role

  - 1: descendant of the recombinant
  - 2: shared ancestry with the left donor haplotype
  - 3: shared ancestry with the right donor haplotype

Chapter 3

Phase 2: Constructing subARG

The recomatrix generated by *iris1* is used in the second phase to construct the subARG.

3.1 Command line parameters

The binary file `iris2` can be executed at command line as:

```
iris2 <input>
```

`iris2` assumes that the <input> *.dat* is in the format described in the preceding chapter.

3.2 Output format

The binary file `iris2` generates the following 4 text files:

1. $<$ *input* $>$ *.net* contains the subARG in format compatible with Pajek [5].
2. $<$ *input* $>$ *.dot* contains the subARG in a format compatible with Graphviz [4].
3. $<$ *input* $>$ *_node_age.txt* contains a list of subARG nodes along with their estimated age in units of the effective population size. The expected age of each node v is computed by Kimura and Ohta's formula [2]

   $$E(v) = \frac{-2p}{1-p}\ln(p),$$

   where p is the relative frequency of the extant descendants of the node. The age is estimated backwards in time with the present extant samples assigned 0.
4. $<$ *input* $>$ *.dis* contains the pairwise distance matrix computed between the input samples based on the subARG. The distance between every pair is computed as the average age of the lowest common ancestors shared by the samples. Since this matrix is symmetric, only the upper triangular entries are filled and remaining entries are assigned 0.

5. $< input >$ *_desc.txt* contains the information about the internal nodes of the subARG. This has been particularly used in PCA analysis of the output. Each row corresponds to an internal node of the subARG and is a binary vector of size n where n is the number of input samples. The jth element is one if a segment borne by this node reaches sample j and 0, otherwise.

Chapter 4

Example

The subARG can be generated for the example file `seqs.txt`, under default parameter settings, using the following commands in order.

1. `iris1 seqs`
2. `iris2 seqs`

`iris1` generates `seqs_recombinations.txt` and `seqs.dat`.
`iris2` generates `seqs.net`, `seqs.dot`, `seqs_node_age.txt` and `seqs.dis`.
These six example files are provided in the zipped folder example.zip at the download website.

Bibliography

[1] A. Javed, M. Melè, M. Pybus, P. Zalloua, M. Haber, D. Comas, M.G. Netea, O. Balanovsky, E. Balanovska, L. Jin, Y. Yang, G. Arunkumar, RM. Pitchappan, The Genographic Consortium, J. Bertranpetit, F. Calafell, and L Parida. Recombination networks as genetic markers: a human variation study of the old world. *under submission*.

[2] M. Kimura and T. Ohta. The average number of generations until fixation of a mutant gene in a finite population. *Genetics*, 61(30):763–71, 1969.

[3] M. Melè, A. Javed, M. Pybus, F. Calafelland L. Parida, J. Bertranpetit, and The Genographic Consortium. A new method to reconstruct recombination events at a genomic scale. *PLoS Comput Biol*, 6(11):e1001010, 2010.

[4] The Graphviz Home Page. http://www.graphviz.org/.

[5] The Pajeck Home Page. http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[6] L. Parida, A. Javed, M. Melè, F. Calafell, J. Bertranpetit, and The Genographic Consortium. Minimizing recombinations in consensus networks for phylogeographic studies. *BMC Bioinformatics*, 10(Suppl 1):S72, 2009.

[7] L. Parida, M. Melè, F. Calafell, J. Bertranpetit, and The Genographic Consortium. Estimating the ancestral recombinations graph (ARG) as compatible networks of SNP patterns. *Journal of Computational Biology*, 15(9):1133–1154, 2008.

[8] M. Stephens, N. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *AJHG*, 68:978–989, 2001.