Software for inferring Ancestral Recombination Graphs and subsequent population genetic analyses:

Margarita

Mark Minichiello

25 July 2007

1 Introduction

MARGARITA is a Java implementation of the algorithm described in [MD06] and can be downloaded from:

`http://www.sanger.ac.uk/Users/mjm`
`http://www.sanger.ac.uk/Software/analysis/margarita`

Margarita infers genealogies from population genotype data and uses these to map disease loci. The genealogies take the form of the Ancestral Recombination Graph (ARG). The ARG defines a genealogical tree for each locus, and as one moves along the chromosome the topologies of consecutive trees shift according to the impact of historical recombination events. There are two stages to the analysis. First, we infer plausible ARGs using a heuristic algorithm, which can handle unphased and missing data. Second, we test the genealogical tree at each locus for a clustering of the disease cases beneath a branch. Since the true ARG is unknown, we average this analysis over an ensemble of inferred ARGs.

2 Requirements for running Margarita

A recent version of Java is required (J2SE 5.0 or later), which can be downloaded from:

`http://java.sun.com/javase`

Download margarita.zip from the MARGARITA website. Then run:

```
unzip margarita.zip
```

To view the source code:

```
jar -xvf margarita.jar
```

If when running MARGARITA an error occurs like:

```
Exception in thread "main" java.lang.NoClassDefFoundError:
JSci/maths/statistics/ChiSqrDistribution
```

then the JSci library (`http://jsci.sourceforge.net`) is not installed and the directory structure resulting from `unzip margarita.zip` has not been preserved. The required file (`jsci-core.jar`) is included in the MARGARITA distribution. Either add the location of `jsci-core.jar` to your `CLASSPATH` environment variable, or copy `jsci-core.jar` into your `$JAVA_HOME/jre/lib/ext/` directory.

3 Running Margarita

The command line is:

```
java -jar margarita.jar INPUTFILE NUMARGS NUMPERMUTATIONS
```

If an out of memory error occurs, increase the maximum heap size; for example, to increase the maximum heap size to 300MB:

```
java -Xmx300M -jar margarita.jar INPUTFILE NUMARGS NUMPERMUTATIONS
```

`INPUTFILE` is the location of the file containing the sequence data; `NUMARGS` is the number of ARGs to infer and use in subsequent analyses; and `NUMPERMUTATIONS` is the number of permutations to perform in order to calculate P-values. We recommend setting the number of ARGs to infer, `NUMARGS`, to between 30 and 100.

Sample input files can be downloaded from the MARGARITA website and take the form:

```
NUMCASEHAPLOTYPES NUMCONTROLHAPLOTYPES NUMMARKERS
POSITION_OF_MARKER_1
POSITION_OF_MARKER_2
.
.
.
POSITION_OF_MARKER_NUMMARKERS
CASE_HAPLOTYPE_SEQUENCE_1
CASE_HAPLOTYPE_SEQUENCE_2
.
.
.
CASE_HAPLOTYPE_SEQUENCE_NUMCASEHAPLOTYPES-1
CASE_HAPLOTYPE_SEQUENCE_NUMCASEHAPLOTYPES
CONTROL_HAPLOTYPE_SEQUENCE_1
CONTROL_HAPLOTYPE_SEQUENCE_2
.
.
.
CONTROL_HAPLOTYPE_SEQUENCE_NUMCONTROLHAPLOTYPES-1
CONTROL_HAPLOTYPE_SEQUENCE_NUMCONTROLHAPLOTYPES
```

The two haplotype sequences for an individual must be written one after the other, that is, the first case individual has `CASE_HAPLOTYPE_SEQUENCE_1` and `CASE_HAPLOTYPE_SEQUENCE_2` as their haplotype sequences; the second case individual has `CASE_HAPLOTYPE_SEQUENCE_3` and `CASE_HAPLOTYPE_SEQUENCE_4` as their haplotype sequences, and so on.

The haplotype sequences are written:

$\texttt{X}_1\texttt{X}_2\texttt{X}_3\texttt{X}_4 \ldots \texttt{X}_{\texttt{NUMMARKERS}}$

where each of the $\texttt{X}_i$ is one of $\{0, 1, \texttt{M}, \texttt{U}\}$. 0 and 1 are the binary codings of SNP alleles, `M` means missing data, and `U` means unphased. As described above, the two haplotype sequences for an individual must follow one after the other, and when there is unphased data, unphased positions must be written as `U` in both haplotype sequences for an individual, for example:

```
10U0011U
10U0011U
```

4 Understanding output from Margarita

MARGARITA's output is printed to the terminal screen. First, the inferred ARGs are described:

```
%ARGINFERENCE
SEQS SNPS MUTS COAS RECS GECS TRCS SECS HEURP
```

There is one line for each inferred ARG, giving:

- `SEQS` the number of haplotype sequences on the leaves of that ARG.
- `SNPS` the number of markers.
- `MUTS` the number of inferred mutation events.
- `COAS` the number of coalescence events.
- `RECS` the number of recombination crossover events.
- `GECS` the number of gene conversion events.
- `TRCS` the total number of recombination break points (=`RECS`+2×`GECS`).
- `SECS` the time taken, in seconds, to construct that ARG.
- `HEURP` the heuristic parameter.

Then follow the lines:

```
%INTERPRETATION
MARKER POSITION ARG CUT_CHISQ_SCORE CUT_CHISQ_PVALUE CHROMOSOMES_UNDER_CUT
FREQ_CASES FREQ_CONTROLS
```

And for each marker and each ARG, there is one line, giving:

- `MARKER` the order position of the marker.
- `POSITION` its base pair position.
- `ARG` the ARG we are considering.
- `CUT_CHISQ_SCORE` the chi-square test statistic for the best cut (the branch with the highest such score) on that marginal tree.
- `CUT_CHISQ_PVALUE` the P-value of that chi-square test.
- `CHROMOSOMES_UNDER_CUT` which haplotype sequences fall under the best cut, which is written as a string of 0s and 1s, where a 1 in the ith position means that haplotype sequence i is under the best cut, and 0 means it is not.
- `FREQ_CASES` the proportion of case haplotypes under the best cut.
- `FREQ_CONTROLS`the proportion of control chromosomes under the best cut.

Then follow the lines:

```
%MAPPING
MARKER POSITION ARG_MAP_SCORE PERM_P-VALUE CHI_P-VALUE
```

There is a line for each marker, giving:

- `MARKER` its order position.
- `POSITION` its base pair position.
- `ARG_MAP_SCORE` is the chi-square test score at the branch which shows the strongest segregation of cases and controls, averaged over the ARGs (marginal trees) at that position.
- `PERM_P-VALUE` the MARGARITA marker-wise P-value calculated by permutation.
- `CHI_P-VALUE` the marker-wise P-value of the chi-square test.

To disable disease mapping, add the flag `-nomapping` to the command line.

To use "smart" permutations, add `-smart CUTOFF` to the command line. Up to `NUMPERMUTATIONS` are performed at a marker, unless `CUTOFF` permutations are found first which show greater association for that marker. The estimated P-value is then `CUTOFF` divided by the number of permutations taken to find that many more associated randomisations. This speeds up the calculation of P-values.

5 Viewing the ARGs and Marginal Trees

To output the ARGs inferred by MARGARITA, add the flag `-args` to the command line. The ARGs are output in the following way:

```
%ARGS
TIME OPERATION CHILD1 {CHILD2} PARENT1 {PARENT2} {LOCATION}
ARG 0
0 re 14 80 81 10
1 co 81 30 82
2 mu 82 83 11
.
.
.
ARG 1
.
.
.
```

- `ARG N` denotes the start of the Nth ARG.
- `A re B C D E` is a recombination event. `A` is the time order of the event. `B` is the child sequence on which the recombination occurs. `C` and `D` are the left and right recombination parents respectively. `E` is the location of the recombination breakpoint (the breakpoint is between markers `E` and `E`+1). The sequences are numbered from 0. Leaf sequences are numbered according to their order in the input file, with the first sequence having identifier 0. New (parent) sequences take the next available integer. Markers are numbered from 0, left to right.
- `A co B C D` is a coalescence event. At time `A`, sequences `B` and `C` coalesce to form parent `D`.
- `A mu B C D` is a mutation event. At time `A`, sequence `B` has the allele at position `D` flipped to the complement state. This yields parent sequence `C`.

To output the marginal trees inferred by MARGARITA, add the flag `-trees` to the command line. The output is then:

```
%TREES
TIME CHILD1 CHILD2 PARENT
TREE: ARG 0 MARKER 0
1 0 4
3 2 5
4 5 6
```

.
.
.
TREE: ARG 0 MARKER 1
.
.
.

The number after ARG gives the ARG from which this tree is extracted, and the number after MARKER gives the position of this tree. The first three numbers of each subsequent line define the tree. For example,

1 0 4

means that nodes 1 and 0 coalesce, and we call the parent 4. The input sequences are labelled from 0, in the same order as in the input file.

6 Imputing missing genotypes and untyped loci with Margarita

M characters in the input file correspond to missing genotypes, and will be imputed by MARGARITA. Add -imputation FILENAME to the command line in order to output the sequences with the missing genotypes filled in. The imputations from each for the ARGs will be output to the file FILENAME and the consensus imputation, taken by finding the most frequently imputed genotypes across the ARGs, will be output to a file FILENAME.consensusimputation.

7 Testing untyped loci with Multiple Imputation

The output from using the -imputation FILENAME option of MARGARITA can be used to test untyped loci for disease association. See [MD07] for a description of this. Combine the case-control sample with a more densely typed sample such as the HapMap, and use this as input to MARGARITA. You should use the -nomapping option to disable the standard MARGARITA disease mapping and the first line of the input file to MARGARITA, where NUMCASES and NUMCONTROLS is specified, must be such that NUMCASES+NUMCONTROLS equals the total number of haplotypes, with the dense sample included. The file must contain the cases first, then the controls, then the more densely typed sample. We recommend using 30 ARGs, giving 30 imputations. For example:

```
java -jar margarita.jar data.in 30 -nomapping -imputation imputations.out
```

Then run the MARGARITAMI Java program, which is included in the MARGARITA download. Run this as follows:

```
java -jar margaritaMI.jar FILENAME NUMCASES NUMCONTS NUMDENSE NUMMARS NUMIMPS
```

Where `FILENAME` corresponds to the file written by MARGARITA run with `-imputation FILENAME` (not the consensus imputation `FILENAME.consensusimputation`). `NUMCASES` is the number of case haplotypes in the data; `NUMCONTS` is the number of control haplotypes and `NUMDENSE` is the number of more densely typed haplotypes. `NUMMARS` is the number of markers. `NUMIMPS` is the number of ARGs (= the number of imputations in `FILENAME`) which MARGARITA produced. The output from MARGARITAMI is:

```
MARKER POSITION IMPUTED_LOGOR IMPUTED_LOGOR_VARIANCE IMPUTED_OR LCL UCL P-VALUE
```

- `MARKER` the order position of the marker.
- `POSITION` its base pair position.
- `IMPUTED_LOGOR` is the imputed log odds ratio for that marker.
- `IMPUTED_LOGOR_VARIANCE` is the between-imputation variance for the estimate.
- `IMPUTED_OR` is the imputed odds ratio for that marker.
- `LCL` lower limit for the 95% confidence interval for the imputed odds ratio.
- `UCL` upper limit of the 95% confidence interval for the imputed odds ratio.
- `P-VALUE` P-value for the imputed log odds ratio.

References

[MD06] Minichiello MJ, Durbin R (2006) Mapping trait loci by use of inferred ancestral recombination graphs. American Journal of Human Genetics.

[MD07] Minichiello MJ, Durbin R (2007) Imputing missing genotypes and untyped loci in association studies by use of inferred ancestral recombination graphs. Submitted.