QTLTree User Manual

Ian Wilson

September 16, 2011

1 Installation

1.1 Required Libraries

1.1.1 TNT

This software needs the Template Numical Toolkit (TNT) for C++ to compile. Download this library and unzip into the src directory. This library is a collection of headers and requires no further installation. Commands to do this are included below, if you have `wget`.

```
wget  http://math.nist.gov/tnt/tnt_126.zip
unzip -d tnt tnt_126.zip   # unzips into directory tnt
```

1.1.2 GSL

The gnu scientific library (GSL) is also required. If this is not present on your system then you need to get hte administrator to install. It is available for all major linux flavours, and for windows and OSX.

1.2 Compilation

To compile on a linux distribution just tyope make in the `src` directory.

2 Introduction

Fastsplit is a program that splits a set of haplotypes into trees and then tests for difference in tree structure between different classes of haplotypes (generally cases and controls).

This document describes how to use the program on a real dataset. Example dataset that illustrates its use are in the directory `example`, along with some example command lines.

Generally the first port of call for help with this software is to use

```
QTLTree --help
```

which produces the following output

```
./QTLTree --help
Usage:
filename <Other Options>

Options
  --help                   produce help message
  --version                version number
  --R (CD/unrelateds.ped)  ped filename, gives the list of samples and
```

```
                          their regions
  --al (4)                Length of population abbreviation
  --b (/users/nijw/GAW/)  Base Directory
  --d (1)                 datasets to use (when have multiple QTL
                          datasets)
  --direction (C)         The direction of the tree, <L>eft, <R>ight
                          or <C>entral
  --k (10)                k - the number of disjoint node statistics
                          to collect
  --keep (false)          Keep all the randomisations - useful to
                          look at the distribution of randomisation
                          statistics but this produces lots of data,
                          use very carefully
  --pheno (CD/unr_phen)   pheno filename.
  --qtlpos (1)            Which QTL to use (1 offset)
  --r (1000)              replicates for the randomisation
  --s (A)                 Statistic to use <A>bsolute, <Z>squared,
                          <P>ositive or <N>egative
  --seed (1)              Random Number Seed
  --snp (CD/snp_info)     snp info filename, gives the snp codes
  --t (Extras/gene_info.4ago2010) File containing list of targets



Usage:
filename <Other Options>

Options used
  --help                  produce help message
  --version               version number
  --E ()                  A file that contains SNPs positions to exclude
  --L (2)                 Case Label
  --P ()                  positions filename (blank for <infile>.position)
  --R ()                  regions filename (blank for <infile>.region)
                          not needed unless you use regional randomisation
  --b (1)                 first SNP to split
  --direction (C)         The direction of the tree, <L>eft, <R>ight or <C>entral
  --end (-1)              The last position to analyse (in MB) - negative for
                          the position of the last SNP
  --f (beagleOut.phased)  Beagle input filename
  --input ()              Stem of input filenames  (.phased, .position and .trait
                          added to create file names)
                          if this is left blank then the individual file names are used
  --k (10)                k - the number of disjoint node statistics to collect
  --n (10)                Number of SNPs to split
  --r (100)               replicates for the randomisation
  --regional (false)      Use regional randomisation?
  --removeCentre (false)  Remove the First SNP
  --seed (1)              Random Number Seed
  --start (-1)            The first position to analyse (in MB) - negative for the
                          position of the first SNP
  --stat (P)              Test Statistic to use
                S for the Sevon test statistic
                Q for the sQuared Sevon statistic,
```

```
                A for the absolute Sevon test statistic
                G for a G-test statistic
                P for the exact binomial tail probability
                N for a normalised exact binomial tail probability (G test statistic)
                T for the Tree test statistic
                C for the 'Cherries' test statistic
                H for the 'Height' test statistic

  --t ()                  Traits filename (blank for <infile>.trait)
```

If you would like to ask for help it is also helpful to know the version used. For this use

```
QTLTree --version
```

3 Available Options

3.1 Input

Input is as haplotypes with addditional data files giving the positions of the SNPs and the traits (Case/Control) say.

3.1.1 Data File

`--f`: The main haplotype input file. This is in Beagle format. an example file is shopwn below. The first line is data for a marker (column 1), for SNP r23094315 (second column) and subsequent columns give the SNP at each haplotype.

```
# version 0.06, command line: WTCCCtoBEAGLE --f=0 --l=1.5 --s=58C,HT
M rs3094315 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
M rs6672353 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
M rs2980300 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
```

3.1.2 Positions File

`--P`: give the name of the positions file.

The file give the positions (as the distance along the chromosome in base pairs) for each of the SNPs in the data file. This file must containt the positions for all SNPs that are in the data file (but can also include SNPs that are not in the data file. An example file is given below. It has two columns. The first is the SNP label for each SNP that is to be used and the second is its position. Note that SNP `rs4040617` is not in the data file. This should not cause any problems.

```
# position file written by WTCCCtoBEAGLE
rs3094315 792429
rs6672353 817376
rs4040617 819185
rs2980300 825852
```

3.1.3 The Regional File

`--R`: Give the file name of the regional file.

This is optional and only used if we use use regional randomisation.

3.1.4 Traits

`--t`: The file containing traits to analyse. Presently only uses the first trait, and just considers those entries that match the case label (option `--L`).

```
# file written by WTCCCtoBEAGLE
A NBS 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

`--L`: Gives the value that we should consider the Case trait. It is **2** by default.

3.1.5 A Shortcut

`--input`: Gives the stem of the input file.
This si a shortcut to allow the use of the same filename stem for all the different input files.
This is only used if the `--f`, `--P` and `--t` options are not used. Using
`--input=stem` is equivalent to using
`--f=stem.phased --P=stem.position --t=stem.trait`.

3.2 Positions to Analyse

The data files given above give us the raw materials to analyse. If no other options are added then **QTLTree** will make trees starting at each position.
`--start` and
`--end`:
Allow you to specify (in Megabases) the start and end points of the region to be analysed. Within this region trees will be started at each position (but also see the option `--E`).

3.3 Tree Building

The haplotype trees are built using a lexical sorting algorithm. Essentially a binary tree is produced where, as you consider each SNP in turn a 1 at a position put you in the left branch of the tree and a 0 puts you to the right. The final trees that are produced depend on a number of parameters.

`--direction`: The direction of the tree splitting. This can either be to the **L**eft of the centre, to the **R**ight, or **C**entrally - that is the positions to split are those in ascending order of distance away from the starting point.

`--n`: The number of positions to split at.

`--E`: A file with the SNP positions to exclude. T

`--removeCentre`: If this option is set then the first SNP is always excluded. This is a useful test if the first SNP is a known factor.

3.4 Test Statistics and Randomisation

The significance of test statistics is tested using randomisation.
`--stat`: Which test statistic to use
`--k`
`--r`: The number of replicate randomisations to use.
`--regional`: Use randomisation within regions.
`--seed` Use a different random number seed to the default for the randomisations.

4 Examples

The directory example contains examples of the use of **QTLTree**

Code	Local	
S	Y	The Sevon test statistic
Q	Y	the squared Sevon statistic
A	Y	the absolute Sevon test statistic
G	Y	a G-test statistic
P	Y	the exact binomial tail probability
N	Y	a normalised exact binomial tail probability (G test statistic)
T	N	The Tree test statistic
C	N	The 'Cherries' test statistic
H	N	The 'Height' test statistic

Table 1: Test statistics available in QTLTree.

4.1 Simple Tree Consistent with no recombination

The dataset in `simpletest.phased` shows some properties of the tree building algorithm. It contains data for five SNPs labelled a-e.

We can look at the data using the R commands for tree building described in appendix A.

```
> library(genomic)
> a <- read.beagle("../example/simpletest")
> cc <- gl(2, 40, labels = c("Case", "Control"))
> tb <- table(cc, apply(a, 1, paste, collapse = "-"))
> SevonStatistic <- function(x) {
+     n <- sum(x)
+     (x[1] - 0.5 * n)/sqrt(0.25 * n)
+ }
> TerminalNodeStats <- apply(tb, 2, SevonStatistic)
> tree1 <- Split(a, 1:40, 1:5)
> tree2 <- Split(a, 1:40, 5:1)
```

	Case	Control	Statistic
0-0-0-0-0	0	3	-1.73
0-0-0-0-1	0	10	-3.16
0-0-0-1-0	3	5	-0.71
0-0-1-0-0	1	2	-0.58
0-1-0-0-0	21	10	1.98
1-0-0-0-0	15	10	1.00

Table 2: Simpletest Data

4.1.1 QTLTree

We can recreate the parts of R analysis by using `QTLTree`. We do this using the command:

```
NULL
```

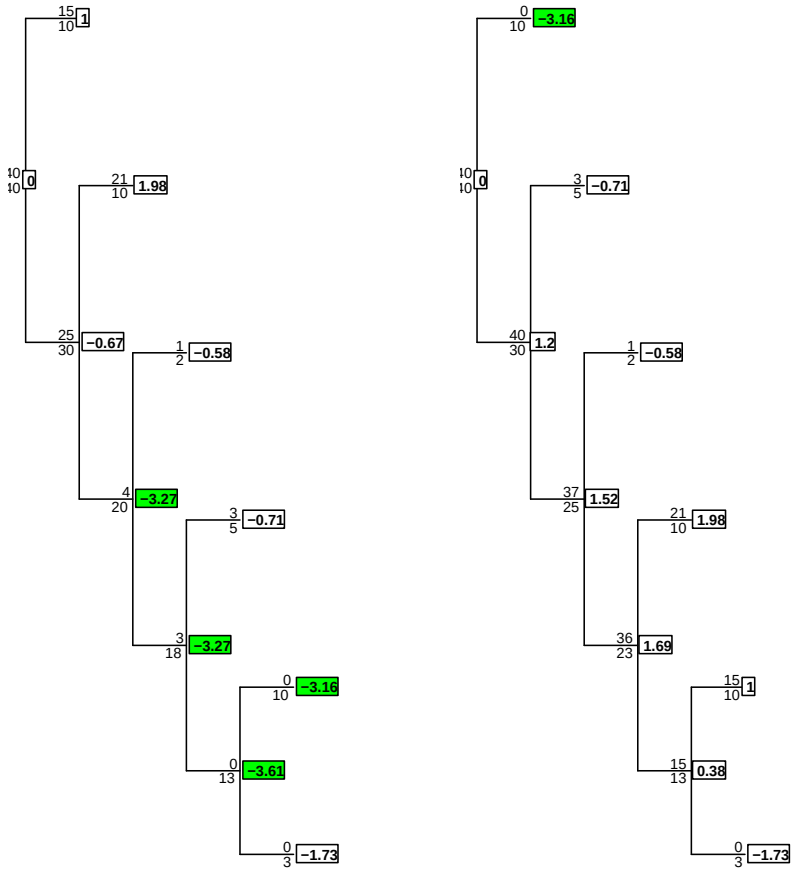

Figure 1: Trees constructed from simpletest. Trees constructioned from Left to Right (Left) and Right to Left (Right).