

Perl Scripts
for use with
QTL Cartographer

Christopher J. Basten

January 28, 2005

1 Automation

A nice feature of the UNIX operating system is the ability to write scripts to automate tasks. We have provided **csh** scripts as examples to do permutation tests and bootstrap analyses. Here we outline some **Perl** scripts to do these and other tasks. The scripts themselves reside in the **doc/scripts** subdirectory but should be installed into the same directory where you keep your *QTL Cartographer* binaries. The new **Permute.pl** and **Bootstrap.pl** scripts do everything their **csh** counterparts did and more.

1.1 Install Instructions

To install the c-shell scripts and perl scripts, first edit the *Makefile* and change it to suit your needs. The line

```
BINDIR = /Users/basten/bin
```

should be changed so that the scripts go into the same directory as your *QTL Cartographer* binaries. The lines

```
REMOVE = /bin/rm -f
INSTALL = /bin/cp
MOVE = /bin/mv -f
PERL = /usr/bin/perl
CSH = /bin/csh
```

should be changed to reflect the true locations of **rm**, **cp**, **mv**, **perl** and **csh**. You could try using **tcsh** in place of **csh** if you like.

Next, you will need to edit the file *scripts.cfg*. This file contains some default definitions for all of the scripts and will become the header for each of the perl scripts.

Once the *Makefile* and *scripts.cfg* have been edited, type

```
make install
```

to install the scripts. This step will prepend *scripts.cfg* to the perl scripts and change the first line of the **csh** commands.

2 Script Man Pages

In the UNIX world, a standard way of providing online documentation of programs is to write man pages. These are ASCII text files with embedded troff commands. UNIX versions of *QTL Cartographer* have man pages for all the programs and many of the scripts in the suite. On a UNIX system, if the man pages are in the correct subdirectory (in essence, if the subdirectory that contains the man pages is defined in the environmental variable MANPATH), then you can get the online help with a command such as

```
% man Rmap
```

We provide html versions of the man pages on the web server for Macintosh and Windows users. If you have World Wide Web access, first point your browser to our home page:

`http://statgen.ncsu.edu/qt1cart`

Then follow the link to the online man pages. You can also access the rest of the ***QTL Cartographer*** manual. The manual is written in L^AT_EX2e and has been translated into HTML by the program **html2latex**.

Since the documentation will change regularly, it is a good idea to check the Web site for the current online manual. The Web pages will always be updated with the manual updates.

3 NAME

Bootstrap.pl - Do a bootstrap analysis with Zmapqtl

4 SYNOPSIS

```
Bootstrap.pl [-b bin] [ -X stem] [-m model] [-r iterations] [-e email]
              [-H hypothesis] [-a rep] [-s] [-h]
```

5 DESCRIPTION

Bootstrap.pl iterates using **Prune** and **Zmapqtl** to determine sampling variances based on a bootstrap resampling.

6 OPTIONS

If the *qtlcart.rc* file if it exists, **Bootstrap.pl** will first set its parameter values from that file. Any command line options will override the *qtlcart.rc* values. If a parameter has not been set by either the *qtlcart.rc* file or command line parameters, default values are set.

-b

This option requires the path to the **QTL Cartographer** binaries and perl scripts.

-r

requires an integer to control how many bootstrap iterations you want to do. The default is 1000.

-X

This option allows you to specify the filename stem. The default is *qtlcart*.

-e

This option requires an email address. The temporary log file will be sent to this address to indicate that the bootstrap is complete. If blank, then no email message will be sent. The default is not to use this option.

-m

This option allows you to specify the **Zmapqtl** model to use. The default is interval mapping (3).

-H

Use this option to specify which hypothesis test you want to use. The usual values are 1 or 30. The default is 1: Unless there are three genotypic classes, do not change this option..

-s

requires no operand. This tells **Bootstrap.pl** to save the bootstrapped datasets and their analytical results. If used with a large number of bootstraps, a great deal of harddisk space will be used up. It is mainly for debugging purposes.

-a

requires an integer operand indicating the last completed iteration. Useful if your machine crashed during the bootstrap.

-h

requires no operand. If used, it prints a usage message and exits.

7 EXAMPLE

Suppose the files *mletest.map* and *mletest.cro* are in the current working directory.

```
% Zmapqtl -X mletest -M 3 -A -V
% Bootstrap.pl -b /home/basten/bin -r 500
```

These commands assume that the **QTL Cartographer** programs are in */home/basten/bin*. It will use interval mapping and do 500 bootstrap iterations. Note that you need to do an initial **Zmapqtl** run before beginning the bootstrap, and that initial run will set the filename stem and the model for analysis.

If your machine crashed during the bootstrap analysis, then you can restart where you left off (provided that the crash occurred during the **Zmapqtl** run). Suppose the above completed 356 iterations and crashed during iteration 357. Then

```
% Bootstrap.pl -b /home/basten/bin -r 500 -a 356
```

would pick up the bootstrap at iteration 357 and complete at 500.

8 CAVEATS

The **-s** option allows you to save the bootstrapped datasets and analytical results. For iteration *i*, model *m*, filename stem *qtlcart* and single trait analysis, there will be files *qtlcart.cro.i*, *qtlcart.zm.i* and *qtlcart.zm.bi*.

This option is mainly for debugging purposes. Be aware that a large number of iterations will use a great deal of disk space. You could modify the **Bootstrap.pl** script to compress these files to save disk space.

9 SEE ALSO

Zmapqtl(1), **Prune(1)**, **SSupdate.pl(1)**

10 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

11 NAME

SSupdate.pl - Update the the sum and sum of squares for the bootstrap

12 SYNOPSIS

```
SSupdate.pl [-l lrcol] [-a additive] [-d dominance] [-I hypothesis]
            [-f SSfile] [-h] [-c] < input > output
```

13 DESCRIPTION

SSupdate.pl reads from the standard input and writes to the standard output. It can do one of two things. The first is to initialize a file of sums and sums of squares in the *Ziboot.out* format from a *Zmapqtl.out* file. The second is to read the results of a run of **Zmapqtl** and the current *Ziboot.out* file and update the results. It is a **Perl** script meant to be run in a loop with **Prune** and **Zmapqtl**.

14 OPTIONS

-f

This option requires an input filename that must exist. It allows the user to specify the *Ziboot.out* file for processing. If it is not given, then the script assumes that an initial file will be created.

-h

requires no operand. If used, it prints a usage message and exits.

-l

requires an integer operand. This should be the column from the *Zmapqtl.out* file with the likelihood ratio that you want processed. By default, it is 4.

-a

requires an integer operand. This should be the column from the *Zmapqtl.out* file with the additive effect that you want processed. By default, it is 0, which means that the sum and sum of squares for the additive effect will not be updated.

-d

requires an integer operand. This should be the column from the *Zmapqtl.out* file with the dominance effect that you want processed. By default, it is 0, which means that the sum and sum of squares for the dominance effect will not be updated.

-I

requires an integer operand. This should be an hypothesis test code. Possible values are 1, 30, 31, 32, 10 and 20. Using this option sets the proper values for the **-l**, **-a** and **-d** options, hence they are ignored in the presence of the **-I** option.

-h

requires no operand. If used, it processes the file specified with the **-f** option. This processing calculates the means and variances of the likelihood ratio and effects for each position. The program exits before reading from the standard input.

15 EXAMPLE

This **Perl** program was meant to be run in a shell script. Here is an example of a **c** shell program that allows calculate the sum and sum of squares for the likelihood ratio and additive effect in a bootstrap experiment.

```
#!/bin/csh
#   Bootstrap
#   Copyright (C) 2000 Christopher J. Basten
# Usage governed by the terms of the
# GNU General Public License, version 2 or higher
# See the file COPYING in this directory
#
#   This file was meant as an example. You will need to edit it
#   to work on your particular system with your data files.
#
#   Start by setting the variables needed.
#
set stem=corn                # filename stem
set hypo=1                   # hypothesis for SSupdate
set model=3                   # analysis model
set reps=1000                # number of bootstraps
set email=basten@statgen.ncsu.edu # email address for notice
set templog=temp.log         # temporary log file
set qbin=/usr/local/bin      # where are the QTL Cart binaries
set bin=/usr/bin              # where are the system programs
#
$bin/rm -f $templog
echo "Bootstrap experiment started " > $templog
$bin/date >> $templog
$bin/echo "Stem: " $stem >> $templog
$bin/echo "Model: " $model >> $templog
$bin/echo "Reps: " $reps >> $templog
```



```

$bin/echo "Email: " $email >> $templog
$bin/mv $stem.log $stem.logsave
$bin/rm -f $stem.z${model}a
$qbin/SSupdate -I $hypo < $stem.z > $stem.z$model.boot
$bin/mv $stem.z $stem.zsave
set i=1
while ( $i <= $reps )
$qbin/Prune -A -V -i $stem.cro -b 1 >>& $templog
$bin/nice $qbin/Zmapqtl -A -V -M $model -i $stem.crb >>& $templog
$qbin/SSupdate -I $hypo -f $stem.z$model.boot < $stem.z > $stem.z$model.new
$bin/mv $stem.z$model.new $stem.z$model.boot
$bin/rm $stem.z
@ i++
end
$qbin/SSupdate -I $hypo -c -f $stem.z$model.boot > $stem.z$model.booted
$bin/mv $stem.logsave $stem.log
$bin/mv $stem.zsave $stem.z
$bin/echo "Bootstrap experiment ended " >> $templog
$bin/date >> $templog
/usr/ucb/mail $email < $templog

```

Suppose you had a data set *corn.cro* and a map file *corn.map*. To use the above shell script, create a directory called *cornboot* and copy the two files into it. Run **Qstats** on the files to initialize the *qtlcart.rc* file, and **SRmapqtl** to rank a set of markers for use with composite interval mapping. Make sure that the **QTL Cartographer** programs are installed in the */usr/local/bin* subdirectory (or change the *qbin* line above). Run the bootstrap with the following command:

```
% Bootstrap &
```

The script will email you a message when it is complete. The example above uses interval mapping and does 1,000 bootstraps. The script above has been rewritten in **Perl**: Please look at the **Bootstrap.pl** man page for more information.

Note that the above example uses **-I 1** for the **SSupdate** line in the loop (this is set with the **set hypo=1** line in the script). This indicates that the dataset are the result of a backcross or recombinant inbred line. If you want to use a different column of likelihood ratios, you can change that option. You could create multiple files of the format *Ziboot.out* and collect the appropriate sums and sums of squares from different hypothesis tests by having multiple instances of **SSupdate** in the loop.

16 HYPOTHESIS TESTS

Using the **-I** option is an easier way to set the columns from the *Zmapqtl.out* file that you want to process. The following values are valid:

1

should be used with backcrosses or recombinant inbreds, that is only those crosses with two distinguishable marker types. It will read the likelihood ratio from column 4 and the additive effect from column 7. The dominance effect will be ignored. The likelihood ratio is for $H_1:H_0$.

10

can be used when more than three marker genotypes are distinguished. Likelihood ratios come from column 11, additive effects from column 7 and dominance effects are ignored. The likelihood ratio is for $H_1:H_0$.

20

can be used when more than three marker genotypes are distinguished. Likelihood ratios come from column 12, dominance effects from column 9 and additive effects are ignored. The likelihood ratio is for $H_2:H_0$.

30

can be used when more than three marker genotypes are distinguished. Likelihood ratios come from column 4, additive effects from column 8 and dominance effects from column 10. The likelihood ratio is for $H_3:H_0$.

31

can be used when more than three marker genotypes are distinguished. Likelihood ratios come from column 5, additive effects from column 8 and dominance effects from column 10. The likelihood ratio is for $H_3:H_1$.

32

can be used when more than three marker genotypes are distinguished. Likelihood ratios come from column 6, additive effects from column 8 and dominance effects from column 10. The likelihood ratio is for $H_3:H_2$.

Recall that when you have two marker classes, there are two hypotheses:

 H_0

No QTL, that is the additive effect is zero

 H_1

The additive effect is nonzero.

In contrast, when you have three marker classes, then you have four hypotheses:

 H_0

No QTL, that is the additive and dominance effects are zero

H1

The additive effect is nonzero, but the dominance effect is zero.

H2

The dominance effect is nonzero, but the additive effect is zero.

H3

Both the additive and dominance effects are nonzero.

17 SEE ALSO

Zmapqtl(1), Prune(1), Qstats(1), SRmapqtl(1)

18 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

19 NAME

Permute.pl - Do a permutation analysis with Zmapqtl or JZmapqtl

20 SYNOPSIS

```
Permute.pl [-b bin] [-X stem] [-m model] [-r reps] [-e email] [-c column]
           [-t trait] [-I ihypo] [-a atrep] [-s] [-u] [-h]
```

21 DESCRIPTION

Permute.pl iterates using **Prune** and **Zmapqtl** or **JZmapqtl** to determine significance thresholds based on permutation testing.

22 OPTIONS

If the *qtlcart.rc* file if it exists, **Permute.pl** will first set its parameter values from that file. Any command line options will override the *qtlcart.rc* values. If a parameter has not been set by either the *qtlcart.rc* file or command line parameters, default values are set.

-b

This option requires the path to the **QTL Cartographer** binaries and perl scripts. The default is *~/bin* directory.

-r

requires an integer to control how many bootstrap iterations you want to do. The default is 1000.

-X

This option allows you to specify the filename stem. The default is *qtlcart*.

-e

This option requires an email address. The temporary log file will be sent to this address to indicate that the bootstrap is complete. If blank, then no email message will be sent. The default is not to use this option.

-m

This option allows you to specify the **Zmapqtl** model to use. The default is interval mapping (3).

-c

Use this option to specify which column in the **Zmapqtl** output file should be processed. The default is 4.

-t

Use this option to specify which trait to analyze. If there are t traits, then a value greater than zero and less than or equal to t will cause **Zmapqtl** to be used. Otherwise, **JZmapqtl** will be used. The default is 1. Note that using this option with a negative value will set this option to 1.

-I

Use this option to specify the hypothesis test for **JZmapqtl**. The default is 10. It is ignored if **Zmapqtl** is used.

-s

requires no operand. This tells **Permute.pl** to save the permuted datasets and their analytical results. If used with a large number of permutations, a great deal of hard-disk space will be used up. It is mainly for debugging purposes.

-a

requires an integer operand indicating the last completed iteration. Useful if your machine crashed during the permutation test.

-u

requires no operand. This is the Unger flag. If used, then the script will also calculate the sum and sum of squares for each permutation over the entire genome, and for each site over all permutations. These values will be saved to files that can be processed with **RCpermute.pl** to calculate site and permutation means and variances.

-d

requires no operand. This is the Doust flag. If used, then the script will also calculate the sum and sum of squares for the GxE scores. It was put in at the request of Andrew Doust. It only works with **JZmapqtl** and models 14 or 34.

-h

requires no operand. If used, it prints a usage message and exits.

23 EXAMPLE

Suppose the files *mletest.map* and *mletest.cro* are in the current working directory.

```
% Zmapqtl -X mletest -M 3 -A -V
% Permute.pl -b /usr/local/bin -c 4 -r 500
```

Will assume that the **QTL Cartographer** programs are in */usr/local/bin*. It will use interval mapping and do 500 permutations. Note that you need to do an initial **Zmapqtl** run before beginning the permutation test. This initial run will have created a *qtlcart.rc* file that contains the model and stem information.

If your computer went down during the permutation run, you can pick up where you left off. Suppose that **Permute.pl** had finished 323 permutations in the above example. You would observe a file *mletest.z3.cwt.323* in the current working directory. You could then run

```
% Permute.pl -b /usr/local/bin -c 4 -r 500 -a 323
```

to continue with the permutation test starting at the 324th iteration and finishing with iteration 500.

Suppose we have another data set with multiple traits: The map is in *multitest.map* and the data in *multitest.cro*. Further suppose that this data set has four traits and it is a backcross. As above, assume that the binaries are in */usr/local/bin*.

```
% JZmapqtl -X multitest -M 3 -t 5 -I 10
% Permute.pl -b /usr/local/bin -c 5 -r 500 -I 10 -t 5
```

Will use all the traits in a multitrait analysis and permute the data 500 times. The likelihood ratio in the *multitest.z0* file will be the focus of the test. We need to specify column 5 for use with **JZmapqtl**.

24 CAVEATS

The **-s** option allows you to save the permuted datasets and analytical results. For iteration *i*, model *m*, filename stem *qtlcart* and single trait analysis, there will be files *qtlcart.cro.i*, *qtlcart.zm.i* and *qtlcart.zm.cwt.i*. For multitrait analysis, the *qtlcart.cro.i* will be saved along with the set of trait analysis files *qtlcart.zt.i*, where *t* is the trait.

This option is mainly for debugging purposes. Be aware that a large number of permutations will use a great deal of disk space. You could modify the **Permute.pl** script to compress these files to save disk space.

25 SEE ALSO

Zmapqtl(1), **JZmapqtl(1)**, **Prune(1)**, **CWTupdate.pl(1)**, **GetMaxLR.pl(1)**, **EWThreshold.pl(1)**, **RCpermute.pl(1)**

26 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

27 NAME

GetMaxLR.pl - The the maximum likelihood ratio from a Zmapqtl output file

28 SYNOPSIS

```
GetMaxLR.pl [-C column] [-i] [-h] [-j] [-r rep] [-t trait] < input > output
```

29 DESCRIPTION

GetMaxLR.pl reads from the standard input and writes to the standard output. It can do one of two things. The first is to initialize a file of likelihood ratio maxima so that the file would have the same output as a *ZipermE.out* file. The second is to read the results of a run of **Zmapqtl** and pull out the largest value in the user-specified column. It is a **Perl** script meant to be run in a loop with **Prune** and **Zmapqtl** or **JZmapqtl**.

30 OPTIONS

-C

This option requires an integer value. It allows the user to specify the *Zmapqtl.out* file column for processing.

-i

requires no operand. If used, the script outputs a *ZipermE.out* header to the standard output and exits. It does require an input file of the *Zmapqtl.out* format. **GetMaxLR** will get the model, cross, window size and number of background parameters from the *Zmapqtl.out* file.

-h

requires no operand. If used, it prints a usage message and exits.

-j

requires no operand. It is a flag to indicate that the input file is a joint-mapping file from **JZmapqtl**.

-r

requires the repetition number for the bootstrap or permutation test.

-t

used with an integer is simply a way to pass on the trait value when using **JZmapqtl**. It should only be used in conjunction with the **-j** and **-i** options.

31 EXAMPLE

This **Perl** program was meant to be run in a shell script. Here is an example of a **c** shell program that allows the user to calculate the experimentwise threshold as well as comparisonwise values.

```
#!/bin/csh
#           Permute
#   Copyright (C) 2000 Christopher J. Basten
#   Usage governed by the terms of the GNU General Public License, version 2 or 1
#   See the file COPYING in this directory
#
#   This file was meant as an example. You will need to edit it
#   to work on your particular system with your data files.
#
#   Start by setting the variables needed.
#
set stem=corn                # filename stem
set column=4                 # LR column to process
set model=3                  # analysis model
set reps=1000                # number of bootstraps
set email=basten@statgen.ncsu.edu # email address for notice
set templog=temp.log         # temporary log file
set qbin=/user/local/bin     # where are the QTL Cart binaries
set bin=/usr/bin             # where are the system programs
#
#   Should only need to change what is above.
#
$bin/rm -f $templog
echo "Permutation test started " > $templog
$bin/date >> $templog
$bin/echo "Stem: " $stem >> $templog
$bin/echo "Model: " $model >> $templog
$bin/echo "Reps: " $reps >> $templog
$bin/echo "Email: " $email >> $templog
$bin/mv $stem.log $stem.logsave
$bin/mv $stem.z $stem.zsave
$bin/rm -f $stem.z{$model}e
set i=1
$qbin/GetMaxLR -i < $stem.zsave > $stem.z{$model}.ewt
$qbin/CWTupdate -C $column < $stem.zsave > $stem.z{$model}.cwt
while ( $i <= $reps )
  $qbin/Prune -A -V -i $stem.cro -b 2 >>& $templog
  $bin/nice $qbin/Zmapqtl -A -V -M $model -i $stem.crb >>& $templog
```



```

$qbin/GetMaxLR -r $i -C $column < $stem.z >> $stem.z{$model}.ewt
$qbin/CWTupdate -f $stem.z{$model}.cwt -C $column < $stem.z > $stem.z{$model}.n
$bin/mv $stem.z{$model}.newcwt $stem.z{$model}.cwt
$bin/rm -f $stem.z
@ i++
end
$bin/echo "Now your can run EWThreshold on $stem.z$model.ewt" >> $templog
$bin/mv $stem.zsave $stem.z
$bin/mv $stem.logsave $stem.log
$bin/date >> $templog
/usr/ucb/mail $email < $templog

```

Suppose you had a data set *corn.cro* and a map file *corn.map*. To use the above shell script, create a directory called *cornperm* and copy the two files into it. Run **Qstats** on the files to initialize the *qtlcart.rc* file, and **SRmapqtl** to rank a set of markers for use with composite interval mapping. Make sure that the **QTL Cartographer** programs and scripts are installed in the */usr/local/bin* subdirectory (or change the line setting *qbin* above). The script above is set to do 1,000 permutations using interval mapping and restricting itself to column four of the *Zmapqtl.out* file. Run it as follows:

```
% Permute    &
```

The script will email you a message when it is complete.

Note that the above example uses **-C 4** for the **GetMaxLR** line in the loop. If you want to use a different column of likelihood ratios, you can change that option. You could create multiple files of the format *ZipermE.out* and collect the maximal likelihood ratio from different hypothesis tests by having multiple instances of **GetMaxLR** in the loop.

Once finished, you will have two files of interest. The first will be *corn.z3.e* and will contain the maxima of the likelihoods for each of the permuted data sets. Run **EWThreshold** on this file to get your experimentwise thresholds. The other file, *corn.z3.cwt*, will contain the comparisonwise thresholds.

The **Permute** script above has been rewritten as a **Perl** script and is described in a man page **Permute.pl(1)**.

32 SEE ALSO

Zmapqtl(1), **Prune(1)**, **EWThreshold.pl(1)**, **CWTupdate.pl(1)**

33 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University

Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

34 NAME

EWThreshold.pl - Calculate the Experimentwise Threshold from a permutation test

35 SYNOPSIS

```
EWThreshold [-s size] [-h] < input > output
```

36 DESCRIPTION

EWThreshold.pl reads from the standard input and writes to the standard output. It reads the results of permutation test that are in the format of **ZipermE.out**. It sorts the likelihood maxima from that file and prints out the 100(1 - size)th percentile.

37 OPTIONS

-s

This option requires an a real value. It should be the size of the test, that is the Type I error probability. The default is 0.05.

-h

requires no operand. If used, it prints a usage message and exits.

38 EXAMPLE

After running the example outlined in the **GetMaxLR** manpage, you can run

```
% EWThreshold.pl -s 0.01 < corn.z3.e
```

to get the experimentwise threshold for a 1% test. The threshold will be printed to the standard output.

39 SEE ALSO

Zmapqtl(1), **Prune(1)**, **GetMaxLR.pl(1)**, **Permute.pl(1)**

40 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

41 NAME

CWTupdate.pl - Update the comparisonwise test results file during a permutation test

42 SYNOPSIS

```
CWTupdate.pl [-f CWTfile] [-h] < input > output
```

43 DESCRIPTION

CWTupdate.pl reads from the standard input and writes to the standard output. It can do one of two things. The first is to initialize a file of likelihood ratios in the *ZipermC.out* format from a *Zmapqtl.out* file. The second is to read the results of a run of **Zmapqtl** and the current *ZipermC.out* file and update the results. It is a **Perl** script meant to be run in a loop with **Prune** and **Zmapqtl**.

44 OPTIONS

-f

This option requires an input filename that must exist. It allows the user to specify the *ZipermC.out* file for processing. If it is not given, then the script assumes that an initial file will be created.

-h

requires no operand. If used, it prints a usage message and exits.

-C

requires an integer operand. This should be the column from the *Zmapqtl.out* file that you want processed.

45 EXAMPLE

See the example in the **GetMaxLR.pl** manpage. An alternate version of the shell script is given below, but the example in **GetMaxLR** is cleaner and simpler to understand. The one presented below has the advantage of being able to input command line parameters rather than having to edit the script. This script has also been rewritten as a **Perl** script with its own man page (**Permute.pl(1)**).

CWTupdate.pl was meant to be run in a shell script. Here is an example of a **c** shell program that allows calculate the experimentwise threshold.

```
#!/bin/csh
#   Permute.csh
#   Usage:  Permute.csh stem permutations email
```

```

# where stem is the filename stem.
#       permutations is the number of permutations
# and   email is the user's email address
# Note: This only works if you have set and used a filename stem.
#
if ( $1 == '-h' ) then
echo "      Usage:  Permute.csh stem model permutations email"
echo "Where"
echo "      stem   = filename stem"
echo "      model   = Zmapqtl Model"
echo "      permutations = number of permutations"
echo "      email   = user's email address"
echo " "
echo "Now exiting"
exit
endif
set templog=temp.log
/usr/bin/rm -f $templog
echo "Permutation test started " > $templog
/usr/bin/date >> $templog
echo "Stem: " $1 >> $templog
echo "Model: " $2 >> $templog
echo "Reps: " $3 >> $templog
echo "Email: " $4 >> $templog
set bindir=/usr/local/bin
set i=1
/usr/bin/mv $1.log $1.logsave
/usr/bin/mv $1.z $1.zsave
/usr/bin/rm -f $1.z$2e
$bindir/CWTupdate -C 4 < $1.z > $1.z$2c
while ( $i < $3 )
$bindir/Prune -A -V -i $1.cro -b 2 >>& $templog
nice $bindir/Zmapqtl -A -V -M $2 -i $1.crb >>& $templog
$bindir/CWTupdate -f $1.z$2c -C 4 < $1.z >> $1.z$2cc
/usr/bin/mv -f $1.z$2cc $1.z$2c
/usr/bin/rm -f $1.z
@ i++
end
/usr/bin/mv $1.zsave $1.z
/usr/bin/mv $1.logsave $1.log
/usr/bin/date >> $templog
/usr/ucb/mail $4 < $templog

```

Suppose you had a data set *corn.cro* and a map file *corn.map*. To use the above shell

script, create a directory called *cornperm* and copy the two files into it. Run **Qstats** on the files to initialize the *qtlcart.rc* file, and **SRmapqtl** to rank a set of markers for use with composite interval mapping. Make sure that the **QTL Cartographer** programs are installed in the */usr/local/bin* subdirectory (or change the 29th line above). Then, to do a permutation test using interval mapping with 1,000 repetitions, run

```
% Permute.csh corn 3 1000 your.email.address &
```

(substituting your real email address above). The script will email you a message when it is complete.

Note that the above example uses **-C 4** for the **CWTupdate** line in the loop. If you want to use a different column of likelihood ratios, you can change that option. You could create multiple files of the format *ZipermC.out* and collect the maximal likelihood ratio from different hypothesis tests by having multiple instances of **CWTupdate** in the loop.

46 SEE ALSO

Zmapqtl(1), **Prune(1)**, **GetMaxLR.pl(1)**, **Permute.pl(1)**

47 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

48 NAME

Model8.pl - Iterate Zmapqtl model 8 to determine a stable set of cofactors

49 SYNOPSIS

```
Model8.pl [-b bindir] [-X stem] [-S sig. threshold] [-i iterations]
          [-m max nbp] [-H hypothesis] [-h]
```

50 DESCRIPTION

Model8.pl iterates using **Zmapqtl** and **Eqtl** to determine a stable set of cofactors for composite interval mapping. First, interval mapping is run and the nearest markers to significant peaks are identified. These markers are used as cofactors in the first iteration of composite interval mapping. A new set of cofactors are identified by proximity to the likelihood peaks and the process is repeated.

51 OPTIONS

-b

This option requires the path to the **QTL Cartographer** binaries and perl scripts.

-i

requires an integer to control how many iterations you want to do.

-X

This option allows you to specify the filename stem.

-S

This option requires a real number to indicate the significance threshold for the likelihood ratio.

-m

This option allows you to specify the maximum number of background parameters in composite interval mapping.

-H

Use this option to specify which hypothesis test you want to use. The usual values are 10 or 30.

-h

requires no operand. If used, it prints a usage message and exits.

52 EXAMPLE

Suppose the files *mletest.map* and *mletest.cro* are in the current working directory.

```
% Model8.pl -b /home/basten/bin -X mletest -S 13.0 -m 25 -H 10 -i 15
```

Will assume that the **QTL Cartographer** programs are in */home/basten/bin*. It will use a significance threshold of 13.0 and allow for up to 25 markers in composite interval mapping. It will iterate 15 times in an attempt to find a stable set of cofactors. This script uses another script called **SRcompare.pl** which compares the set of cofactors (in *SRmapqtl.out* format) between consecutive runs and reports how many cofactors have been added or deleted.

53 SEE ALSO

Zmapqtl(1), **Qstats(1)**, **Eqtl(1)**, **SRcompare.pl(1)**

54 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

55 NAME

SRcompare.pl - Compare a pair of **SRmapqtl** output files to see how many cofactors were gained or lost

56 SYNOPSIS

```
SRcompare.pl [-f SRmapqtl.old] [-h] < SRmapqtl.new > output
```

57 DESCRIPTION

SRcompare.pl reads from the standard input and writes to the standard output. It is meant to compare the set of cofactors in two **SRmapqtl** output files. It will print out the number of cofactors that are the same, lost or gained in the two files.

58 OPTIONS

-f

This option requires an input filename that must exist. It allows the user to specify the old *SRmapqtl.out* file for processing. If it is not given, then the script dies.

-h

requires no operand. If used, it prints a usage message and exits.

59 INPUT FILES

The input files should be of the same format as the output of **SRmapqtl**. You should have only one set of such results in a file to be processed by **SRcompare.pl**. These results are created by **Eqt1** as well as **SRmapqtl**.

60 OUTPUT

At this time the output will be a single line telling how many cofactors were lost, gained or remained, between the two files. For example:

```
10 are the same, 2 were lost and 0 were gained.
```

Ideally, you would like to see 0 lost and 0 gained.

61 EXAMPLE

Suppose we have a map in *qtlcart.map* and a data file in *qtlcart.cro*, and that there is no resource file in the current directory. For this example, assume that there are only two marker genotypes (that is a backcross or recombinant inbred line) and one trait. The following series of commands can be used to compare the cofactors chosen from interval mapping with those from composite interval mapping:

```
% Qstats -X qtlcart
% Zmapqtl -M 3 -A
% Eqt1 -S 12.0 -H 10 -I Z -A
% mv qtlcart.z qtlcart.z.1
% Zmapqtl -M 8 -A
% mv qtlcart.eqt qtlcart.eqt.1
% mv qtlcart.sr qtlcart.sr.1
% Eqt1 -S 12.0 -H 10 -I Z -A
% SRcompare.pl -f qtlcart.sr.1 < qtlcart.sr
```

In the above, **Qstats** is run to set the filename stem. Next, **Zmapqtl** does an interval mapping analysis and **Eqt1** picks a set of markers closest to the peaks from interval mapping: These peaks are only used if they have likelihood ratios greater than 12.0. Next, the interval mapping results are moved to a new file and composite interval mapping is run using the cofactors identified via interval mapping. The previous output files of **Eqt1** (*qtlcart.eqt* and *qtlcart.sr*) are renamed and new files are generated from the composite interval mapping results. Then, **SRcompare.pl** is used to compare the two sets of results, those from interval mapping (*qtlcart.sr.1*) and those from composite interval mapping (*qtlcart.sr*).

This type of analysis can be iterated until a stable set of cofactors are identified.

62 SEE ALSO

Eqt1(1), **Qstats(1)**, **SRmapqtl(1)**, **Zmapqtl(1)**

63 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

64 NAME

Prepraw.pl - Input a mapmaker.raw file and prepare it for QTL Cartographer

65 SYNOPSIS

```
Prepraw [-h] < input > output
```

66 DESCRIPTION

Prepraw.pl reads from the standard input and writes to the standard output. It reads a mapmaker.raw file and reformats it slightly. It will use standard tokens for each marker type. It will count the number of individuals typed for each marker, and print a warning if there are missing tokens.

67 OPTIONS

-h

requires no operand. If used, it prints a usage message and exits.

68 EXAMPLE

Suppose you have a **MAPMAKER** raw file called *corn.raw*. If you have trouble converting it with **Rcross**, then try this.

```
% Prepraw.pl < corn.raw > corn2.raw
```

and check if there are the proper number of data points for each marker and trait.

69 SEE ALSO

Rcross(1)

70 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

71 NAME

TestExamples.pl - Run a basic analysis on the example files

72 SYNOPSIS

```
TestExamples.pl [-D debug] [-t test dir ] [-e example dir ] [-b bin dir ] [-h]
```

73 DESCRIPTION

TestExamples.pl is a script that will use the **QTL Cartographer** commands to run basic analyses on all of the example data sets.

74 OPTIONS

-b

This option requires the path to the **QTL Cartographer** binaries and perl scripts.

-t

This option requires the path to the base directory for doing the analyses. This directory will be created if it doesn't exist. Each data set will have a subdirectory in the base directory for analysis.

-e

Use this to specify the location of the example subdirectory.

-D

Allows you to set the debug level. The default is zero. You can set it to any of 0, 1, 2, or 3.

-h

requires no operand. If used, it prints a usage message and exits.

75 EXAMPLE

Suppose that the example files are in */home/user/QTLCartUnix/example* and you want to run all the tests in a base directory */home/user/test*. Suppose further that you have installed the **QTL Cartographer** binaries in */usr/local/bin*. Use

```
% TestExamples.pl -b /usr/local/bin -t /home/user/test \  
    -e /home/user/QTLCartUnix/example
```

This will create the directory */home/user/test* if it doesn't exist, and then create sub-directories for each data set. It will copy the data file and genetic linkage map from the example subdirectory into the data set subdirectory of */home/user/test*. Then, it will convert the map and data files into the proper formats and run **Qstats**, **LRmapqtl**, **SRmapqtl**, **Zmapqtl** on all traits. If there are multiple traits, **JZmapqtl** will be run followed by **Eqtl** and **Preplot**. Then, **JZmapqtl** will convert the data so that **MultiRegress** can generate an initial model that **Rqtl** translates and **MImapqtl** refines. This will be done for all the data sets in the example directory that have filename extension *.inp*.

The script also runs analyses for the *mletest* and *sample* data sets, and does two simulations: One with a single trait and one with two traits.

At the end of the script, you will find a file */home/user/test/Verbiage.txt* that contains all of the verbal output from the programs. This file is deleted by the script if you run it again.

76 NOTE

You will need at least 36 megabytes of disk space to run this program.

77 SEE ALSO

QTLcart(1) and all programs referenced therein.

78 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

79 NAME

Ttransform.pl - Shell of a script to allow programming of trait transformations

80 SYNOPSIS

```
Ttransform.pl [-a a] [-b b] [-c c] [-f func] [-h] < input > output
```

81 DESCRIPTION

Ttransform.pl reads an *Rcross.out* formatted file from the standard input and writes same to the standard output. You can customize the script to transform trait values.

82 OPTIONS

You will need to customize the script to suit your needs. Look for the phrase LOOK HERE in the script.

-a

requires a real-valued operand. It is 0.0 by default and used as explained below.

-b

requires a real-valued operand. It is 1.0 by default and used as explained below.

-c

requires a real-valued operand. It is 0.0 by default and used as explained below.

-f

requires an integer-valued operand. It is 1 by default tells **Ttransform.pl** which function to use.

-h

requires no operand. If used, it prints a usage message and exits.

83 TRANSFORMS

There are some predefined transforms available. Let Y be the trait value, a, b and c real-valued coefficients and Z the transformed trait. Use the **-f** option and an integer from one to six to have the following transforms applied to all traits:

1. Subtract the mean and divide by the standard error. This is the default.

$$Z = (Y - \text{Mean}(Y)) / \text{STDERR}(Y)$$

2. Subtract the mean, square it and divide by the sample variance.

$$Z = (Y - \text{Mean}(Y))^2 / \text{VAR}(Y)$$

3. This is a linear or quadratic transform.

$$Z = a + b Y + c Y^2$$

4. This is a natural log transform.

$$Z = a + b \log(Y)$$

5. This is an exponential transform

$$Z = a \exp(b Y)$$

6. This is for you to code any transform that you like.

84 EXAMPLE

Suppose the file *mletest.cro* is in the current working directory.

```
% Ttransform.pl < mletest.cro > mletest.cro.out
```

would transform all the traits in *mletest.cro*. The default transform is to subtract the mean and divide by the standard error.

85 CAVEATS

You should know some Perl to use this script.

86 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

87 NAME

Vert.pl - Convert line endings

88 SYNOPSIS

```
Vert.pl [-i u|d|m ] [-o u|d|m ] [-h] < input > output
```

89 DESCRIPTION

Vert.pl reads from the standard input and writes to the standard output. It converts the line endings of text files from one format to another based on the options **-i** and **-o**. The default is to convert UNIX files into DOS files.

90 OPTIONS

-i

This option requires a one letter code indicating the type of input file. Options are **u** for UNIX, **d** for DOS and **m** for Macintosh.

-o

This option requires a one letter code indicating the type of output file. Options are **u** for UNIX, **d** for DOS and **m** for Macintosh.

-h

requires no operand. If used, it prints a usage message and exits.

91 EXAMPLE

Suppose *corn.cro* and *corn.map* are UNIX formatted files that you want to convert to DOS. The following sequence converts them to DOS format

```
% Vert.pl -i u -o d < corn.cro > dcorn.cro
% Vert.pl -i u -o d < corn.map > dcorn.map
```

No imagine that you have an entire directory of UNIX files and you want to convert them to Macintosh formatted files. You could use

```
% foreach i ( * )
? Vert.pl -i u -o m < $i > $i.mac
? /usr/bin/mv $i.mac $i
? end
%
```

The above would replace all of the files with Macintosh formatted versions. Alternatively, you could copy them into a new directory:

```
% mkdir macdir
% foreach i ( * )
? if ( -f $i ) Vert.pl -i u -o m < $i > macdir/$i
? end
%
```

This script is useful for shared filesystems, that is when a fileservers can be accessed by Windows, UNIX or Macintoshes.

92 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

93 NAME

Ztrim.pl - Trim a Zmapqtl output file for terminal reading

94 SYNOPSIS

```
Ztrim.pl [-H Hypothesis] [-l lines] [-w width] [-t] [-r]
        [-s] [-h] < input > output
```

95 DESCRIPTION

Ztrim.pl reads from the standard input and writes to the standard output. It reads the output of **Zmapqtl** and eliminates columns based on the value of *Hypothesis*. It is a quick way to make the output file fit in a terminal window without having the text wrap.

96 OPTIONS

-H

This option requires an integer value of 1, 10, 20, 30, 31, 32 or 33. The behavior of the program for each value is explained below.

-l

This option requires an integer value greater than 1. It specifies how often the column headers should be printed. The default value of 0 means that headers will be printed at the top of each block of output. If your terminal shows 25 lines, then it is useful to reprint the header every 21 lines so as to have a header on each page of the output. (The header is 3 lines long...if you use **more**, then you will need one more line for it.)

-w

This option requires an integer value greater than 1. It specifies character width of the output terminal. The default is 80, which is a common value. It only effects the horizontal bars defining the headers.

-h

requires no operand. If used, it prints a usage message and exits.

-r

Prints R squared values

-t

Prints total R squared values

-s

Prints residual test statistic values

97 OUTPUT

Different values used with the *-H* option allow the user to output specific columns from the *qtlcart.z* file. All options print out columns for the chromosome, marker and positions. All print at least one likelihood ratio test statistic and at least one parameter estimate.

For crosses with two marker classes (backcrosses and recombinant inbreds), we can estimate one parameter value, namely the additive effect. The test is H0 (no QTL) versus H1 (QTL with an additive effect a).

For crosses with three marker classes, we can estimate additive and dominance effects. Let a be the additive effect and d be the dominance effect. We can set up four hypotheses:

H0

No QTL: $a = 0$ and $d = 0$.

H1

QTL with an additive effect: a not 0 and $d = 0$. We refer to the estimate of the additive effect as $a1$ under this hypothesis.

H2

QTL with a dominance effect: $a = 0$ and d not 0. We refer to the estimate of the dominance effect as $d2$ under this hypothesis.

H3

QTL with both an additive and a dominance effect: a not 0 and d not 0. We refer to the estimate of the additive effect as $a3$ and that of the dominance effect as $d3$ under this hypothesis.

Using the following values with the *-H* option yields

1

This is the default and is useful for crosses with two genotypic classes. The output simply deletes the columns with zeros. You are left with the likelihood ratio of H1 to H0 and the estimate of the additive effect.

10

This essentially prints the same information as 1 above except it does it for crosses with three genotypic classes.

20

This prints the likelihood ratio for a test of dominance where the additive effect is zero. You get the H2 to H0 likelihood ratio, the estimate of dominance, the R squared estimates and the test statistic for normality of the residuals.

30

This prints the H3 to H0 likelihood ratio. You get the hypothesis test likelihood ratio, the estimates of additivity and dominance, the R squared estimates and the test statistic for normality of the residuals.

31

This option prints everything that option 30 prints, but adds the test of H3 to H1 and the *a1* estimate.

32

This option prints everything that option 30 prints, but adds the test of H3 to H2 and the *d2* estimate.

98 EXAMPLE

```
./Ztrim.pl -l 20 -w 78 -H 30 -s < qtlcart.z | more
```

will print the test statistic H3:H0 and the estimates of the additive and dominance effects for the results in *qtlcart.z*. It will also reprint the column headers every 20 lines. This works well for a terminal set to 24 lines by 80 columns. The headers take 3 lines each and the **more** command chews up the last line of each screen.

99 SEE ALSO

Zmapqtl(1)

100 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934

101 NAME

Dobasics.pl - Do a basic set of analyses for a QTL mapping data set

102 SYNOPSIS

```
Dobasics.pl [-b bindir] [-X stem] [-S sig. threshold] [-m mapinput]
            [-d datainput] [-t trait] [-H hypothesis] [-V] [-h]
```

103 DESCRIPTION

Dobasics.pl does a basic analysis of a QTL mapping data set. Map and data files are translated first. Then, basic single marker analyses are followed by interval mapping and composite interval mapping. **Eqt1** produces a summary of results and **Preplot** prepares the results for plotting with **Gnuplot**.

104 OPTIONS

-b

This option requires the path to the **QTL Cartographer** binaries and perl scripts.

-t

requires an integer telling which trait to analyze. If you want to do them all, set this to one more than the number of traits.

-X

This option allows you to specify the filename stem.

-S

This option requires a real number to indicate the significance threshold for the likelihood ratio.

-m

This allows you to specify an input file for the map data. It should be in the format of *map.inp* or **MAPMAKER/EXP** output. If not used, the script assumes that the map file already exists in the correct format.

-d

This allows you to specify an input file for the mapping data. It should be in the format of *cross.inp* or **MAPMAKER/EXP** raw format. If not used, the script assumes that the data file already exists in the correct format.

-H

Use this option to specify which hypothesis test you want to use. The usual values are 10 or 30.

-V

requires no operand. If used, it turns off messages indicating the progress of the script.

-h

requires no operand. If used, it prints a usage message and exits.

105 ANALYSES

Assume that we will use the filename stem *qtlcart* for the following. The script does the tasks in this order:

Rmap

Convert the map found in the file specified by the **-m** option into **QTL Cartographer** format with **Rmap**. If the option is not used, then the script assumes that the map is already in the correct format and resides in the file *qtlcart.map*.

Rcross

Convert the data found in the file specified by the **-d** option into **QTL Cartographer** format with **Rcross**. If the option is not used, then the script assumes that the data is already in the correct format and resides in the file *qtlcart.cro*.

Qstats

Run **Qstats** to generate basic statistics.

LRmapqtl

Do simple linear regression with **LRmapqtl**.

SRmapqtl

Do stepwise linear regression with **SRmapqtl**.

Zmapqtl

Do composite interval mapping with **Zmapqtl** and the model specified on the command line. The default is to use model 6.

JZmapqtl

Do multiple trait composite interval mapping with **JZmapqtl**. This will only be done if you specified more than one trait with the **-t** option.

Eqtl

Estimate QTL positions from the analyses above.

Preplot

Prepare the results for display with **GNU PLOT**. You will need to run **GNU PLOT** on your own.

JZmapqtl

This step converts the data into a format that can be analyzed by **MultiRegress**.

MultiRegress

Use least-squares analysis to estimate QTL postions.

Rqtl

Convert the output of **MultiRegress** into a format that can be read as an initial model for multiple interval mapping.

MImapqtl

Use the reformatted results from **MultiRegress** to estimate new QTL and epistatic effects.

106 EXAMPLE

Suppose the files *mletest.map* and *mletest.cro* are in the current working directory.

```
% Dobasics.pl -b /home/basten/bin -X mletest -S 13.0 -H 10
```

Will assume that the **QTL Cartographer** programs are in */home/basten/bin*. It will use a significance threshold of 13.0. It will do basic statistics, single marker analyses, interval and composite interval mapping, and prepare the data for **Gnuplot**.

Suppose the files *realdatm.inp* and *realdatac.inp* are in the current working directory. These files are formatted like *map.inp* and *cross.inp* and need to be translated. Use

```
% Dobasics.pl -b /home/basten/bin -X realdat -m realdatm.inp -d realdatac.inp -S
```

for this analysis.

107 SEE ALSO

Zmapqtl(1), **Qstats(1)**, **Eqtl(1)**, **LRmapqtl(1)**, **SRmapqtl(1)**, **Rmap(1)**, **Rcross(1)**, **Rqtl(1)**, **MultiRegress(1)**, **MImapqtl(1)**, **Preplot(1)**

108 AUTHORS

In general, it is best to contact us via email (basten@statgen.ncsu.edu).

Christopher J. Basten, B. S. Weir and Z.-B. Zeng
Department of Statistics, North Carolina State University
Raleigh, NC 27695-7566, USA
Phone: (919)515-1934