User's Guide for R Routines to Perform Reference Marker Normalization

Stan Pounds and Charles Mullighan
St. Jude Children's Research Hospital
Memphis, TN 38135
USA

Version Date: January 29, 2008

Purpose

The purpose of this software is to normalize the signals of SNP or CGH arrays by matching the distribution of signals for markers that are believed to be diploid a priori. The normalized data are then useful for DNA copy number analysis. The algorithm is described in the supplementary materials of Mullighan et al (2007).

Licensing

The software is free to use provided that the associated manuscript is cited in all scientific presentations, abstracts, or manuscripts that result from use of the method. The software has no warranty.

Getting Started

The software is run using the freely available R software. Thus, R must be installed prior to use of this software. Visit www.r-project.org to download and install R on your machine.

Next, download the files refnorm-library-post.R, refnorm-example-post.R, and others from www.stjuderesearch.org/depts/biostats/refnorm.html.

Preparation of Data Files

The user must prepare 3 input files prior to using the algorithm: a data file including the input summarized (but not yet normalized) signals, a data file including the chromosomal locations of array features, and a data file including reference chromosome information for each sample. All files must be in tab-delimited text format and the first row must be a header row with column names. More detailed descriptions of each file are given below; you may wish to refer to the example data sets and code files for additional help.

In the data file, rows correspond to array features and columns correspond to samples. The data file must include a column with identifiers for array features and columns of unnormalized summary signals. Additional data columns (such as genotype calls) may be included for convenience; the algorithm will only normalize the columns listed specifically in the reference chromosome file (described later). Guidance on using dChip software (Lin et al. 2004) to obtain an input data file suitable for this application is

provided at the end of this user's guide. To use other software packages to obtain unnormalized summary signals, see the help materials of those packages.

In the annotation file, rows correspond to array features and columns give annotation information for each feature. The annotation file must include a column with the identifiers of the array features and a column with the chromosome on which the array features are located. Additional columns of annotation may be included for convenience; the normalization algorithm will only utilize these two columns.

The reference chromosome data file has rows corresponding to samples and columns giving information about the samples. The reference chromosome data file must include a column with identifiers of the signal columns to be normalized. These identifiers must exactly match the names of the signal columns in the data file to be normalized. It is critical that the reference chromosome file include all the columns to be normalized; the algorithm will normalize only those columns that appear in the reference chromosome file. The reference chromosome file must also include a column with a list of chromosomes for each signal column to be normalized. This column may contain chromosomes to be used a references or chromosomes to be excluded from the reference set. However, the same convention must be used for every row in the file. The chromosome list should be enclosed in quotation marks and commas used to separate chromosomes within the list. Additionally, chromosomes should be referred to using the same convention as in the annotation file.

Routines of Interest to Most Users

The R code file "refnorm-library.R" defines a set of routines that R can use to perform the normalization. The routines of greatest interest to end-users are `annsubset.file` and `refnorm.file`. The `annsubset.file` routine will subset the signal data file on those array features with annotation information in the annotation file. This performs an important data preparatory step because quality control array features should not be included in the normalization process. Including such features may skew the normalization. The `auto.refchrom.file` routine uses heterozygosity calls and signal data to automatically select reference chromosomes for each sample. The `refnorm.file` routine actually performs the reference marker normalization. These two routines are discussed in greater detail below.

The required arguments of the `annsubset.file` routine are `workdir`, `input.file`, `ann.file`, `output.file`, `input.marker.col`, and `ann.marker.col`. The argument `workdir` should be a character string giving the full file path for the working directory (which should contain all data files). Users should be aware that \\ should be used to separate different levels of the directory because \ is a special character in the R language (for example, use "D:\\SPounds\\RefNorm\\" instead of "D:\SPounds\Refnorm\"). The argument `input.file` should be a string giving the name of the file with the input data signals. The argument `ann.file` should be a string giving the name of the annotation file. The argument `output.file` should be a string giving the name of the output data file that is subset on those array features that are

included in the annotation file. The argument `input.marker.col` should be the name or numeric index of the column of the data file containing the identifiers of the array features. The argument `ann.marker.col` should be the name or numeric index of the column of the annotation file with the identifiers of the array features. The `annsubset.file` routine has one optional argument, `max.mem.datapts`, is the maximum number of data points to hold in memory at one time. The default value for `max.mem.datapts` is 10,000,000. Note that all character string arguments for this and other routines should be enclosed in quotation marks unless otherwise noted.

The required arguments of the `auto.refchrom.file` routine are `prenorm.file`, `snpann.file`, `refchr.file`, `snpann.snpcol`, and `snpann.chromcol`. The argument `prenorm.file` gives the full file path for the file with pre-normalized signals and genotype calls for each sample. The argument `snpann.file` gives the full file path for the SNP annotation file. The `refchr.file` gives the full file path for the reference chromosome file to be generated by auto.refchrom.file. The `snpann.snpcol` argument specifies the column of the SNP annotation file with the SNP identifiers. The `snpann.chromcol` argument specifies the column of the SNP annotation file with the chromosome information.

The `auto.refchrom.file` routine has some optional arguments as well. The `prenorm.snpcol` argument specifies the column of the prenormalized data file with the SNP identifiers (default = 1). The argument `min.hz` specifies the minimum heterozygosity rate that is acceptable for a reference chromosome (default = 0.15). The `exc.chrom` argument includes a list of chromosomes that should be excluded from the reference set for all samples (default = c(“X”,”Y”,23:24)). Finally, the `logt.signal` argument specifies whether signals should be log-transformed for purposes of reference selection (default = T).

The required arguments of the `refnorm.file` routine are `workdir`, `input.file`, `chrom.file`, `ann.file`, `output.file`, and `chrom.type`. The argument `workdir` should be a character string that specifies the full file path for the working directory that contains all data sets. The arguments `input.file`, `chrom.file`, `ann.file`, and `output.file` give character strings with the names of the input signal file, the reference chromosome file, the annotation file, and the output normalized signal file, respectively. The `chrom.type` argument should indicate the convention used in the reference chromosome file. It should indicate whether the listed chromosomes are those that are included (value =”include”) in the reference set or those that should be excluded (value = “exclude”) from the reference set.

The `refnorm.file` routine has a number of optional arguments. The argument `sex.chrom` gives a list of chromosome identifiers that correspond to the sex chromosome. The algorithm automatically excludes the sex chromosomes from the set of reference chromosomes. The default value is `c(23:24,"X","Y")` which would exclude the X and Y chromosomes for human studies (some annotation files use 23 and 24 as numeric conventions for the X and Y chromosomes).

There are several arguments that specify the columns with important information in each data file. The `input.marker.col` argument gives the name or numeric index of the column with feature identifiers in the input data file. The default is 1 because many data files include the feature identifiers as their first column. The `ann.marker.col` gives the name or numeric index of the feature identifier column in the annotation file (default =1). The `ann.chrom.col` gives the name or numeric index of the chromosome column of the annotation file (default =2). The arguments `chrom.file.IDcol` and `chrom.file.chromcol` give the name or numeric index of the columns with the sample identifiers (i.e. names of the columns in the data file to be normalized) and the chromosome list, respectively.

There are additional optional arguments that control specific aspects of the actual normalization process. The `trim.ref` argument accepts a logical variable (T/F) that indicates whether the reference set should be trimmed for outliers on the across-array unit-rank scale (default =T). See the paper for more details on the trimming. The `qtarget` argument gives the name of a function that evaluates the quantile function of the target distribution. The default is `qlnorm`, which is the built-in R function for the quantile function of the log-normal distribution. This function name should not be enclosed in quotation marks. To map signals directly to the log-transformed scale, use `qnorm` for this argument. The arguments `target.p1` and `target.p2` accept values for the first and second parameters of the quantile function. The defaults are 0 and 1 respectively. Thus, the defaults map the signals of the reference set to a log-normal distribution with μ=0 and σ=1.

There are three arguments that manage memory and file size. The argument `temp.prefix` is a character string that gives the prefix of temporary files that may be generated during the normalization procedure to reduce memory requirements. These temporary files are automatically deleted once the normalization is complete. The default is “tempnorm”, so that the files generated have names “tempnorm1.txt”, “tempnorm2.txt”, “tempnorm3.txt”, etc. The number of files generated depends on the size of the input data file and the value of the optional argument `max.datapts.mem`. The argument `max.datapts.mem` specifies the maximum number of data points to be kept in memory at once. If this argument is too large, the program will crash because it cannot obtain enough memory. However, setting this value very small will dramatically increase computing time. The default is 10,000,000 (= 1e7). The final argument, `ndigits`, specifies the number of decimal points to be included in the final normalized data file. Including a large number of decimal places in the final normalized data file will dramatically increase the size of that file. The default is 4.

Preparing the Normalization Program File

The information above provides guidance in preparing a very short and simple program to be run to perform the normalization. First, use the source command to make the library of routines in `refnorm-library.R` available for use. Next, use `annsubset.file` to generate a signal file with data only for annotated markers. If necessary, use `auto.refchrom.file` to generate a file with reference chromosomes

that are computationally selected. Finally, use `refnorm.file` to perform the normalization. The user needs only to provide the values of the arguments of `annsubset.file` and `refnorm.file` as described above. The file `refnorm-program.R` is an example program that can be run in R once the example data files are downloaded and the program is modified to reflect where those files are stored on the user's machine.

Executing the Normalization Program

The program may be executed in an R session or in R batch mode. To execute the program in an R session, simply open R and then cut and paste the entire program to the command prompt. This is a simple way to execute the program, but batch mode processing may be preferred.

To perform the normalization in R batch mode in Windows, first go to the Start Menu and choose "Run…" Then type "CMD" in the "Run" window that appears to open a command window. At the command prompt type a command of the following form:

```
"Rterm.exe" --vanilla < "Program File" > "Log File"
```

This command will have R execute the program file and output messages to the specified log file. Note that only a single backslash (\) is needed to separate the directory levels in the location specification. The file location and names must be enclosed in double quotes. The full file location should be included in each set of quotes.

More details on how to execute the R program in other platforms (e.g. Unix, Linux, or Mac) may be found at www.r-project.org.

Lowering the Priority and Checking Progress

For large input files, the normalization may take some time. It may be desirable to assign the normalization process a lower priority so that the computer can be used for other activities while the program runs. In Windows, this can be done by pressing Control-Alt-Delete and choosing "Task Manager." Then, select the Processes Tab and click on "CPU" twice to sort the processes by decreasing CPU usage. This should be the Rterm.exe process to the top (or near the top) of the process list. Right-click on the Rterm.exe process and choose "Set Priority" and then "Low." This will lower the priority of the normalization process so that other activities (such as using Word) are not hindered by the normalization. This will not substantially extend the computing time unless there are other computationally intensive processes running at the same time. The priority can also be set in most Unix systems with the appropriate command.

You can check on the progress of the normalization procedure by viewing the log file. The program will write messages regarding the progress of the normalization to the log file. In Windows, Internet Explorer is a good choice for viewing the file. You want to avoid having a program other than R (such as Notepad or Word) write to the log file.

Using dChip to Obtain Unnormalized Summary Signal Data and Analyze Reference-Marker Normalized

In theory, any method that produces a summarized signal intensity and genotype for each SNP probe set for each sample should be satisfactory for the purposes of the normalization algorithm. In practice, we use dChip (www.dchip.org, Lin et al 2004) to summarize probe level data. A comprehensive guide to reading in array data and array pre-processing can be found at the dChip website.

Briefly, create a directory containing a **sample information file** of the basic format:

Array	Sample	Group	Gender	Ploidy(numeric)	Call
ALL1-N-250knsp		MLL	Male	2	95
ALL1-T-250knsp		MLL	Male		94
ALL2-N-250knsp		E2A	Female	2	93
ALL2-T-250knsp		E2A	Female		98

Save this as a tab delimited text file. (The sample column is optional, but can be used to add additional case-specific labels or information).

Create an **array list file** that specifies the ordering of the arrays for analysis in dChip. This is useful when analysing data from different arrays for the same samples, to ensure that exported files have matching columns, and also for generating paired analyses in dChip following normalization. For example:

ALL1-N-250knsp
ALL1-T-250knsp
--- Standardize---
ALL2-N-250knsp
ALL2-T-250knsp
--- Standardize---

For each array, dChip requires the array CEL file and either a SNP call TXT file, or the CHP file. Both the SNP call and TXT files can be generated by Affymetrix GTYPE software. Either copy/place all files in the current analysis directory, or create a **data file list** directing dChip to the folder containing all CEL/TXT/CHP files. This is convenient for large analyses. The data file list for a computer with Affy files stored in the default directory looks like this:.

C:\GeneChip\Affy_Data\Data\ ALL1-N-250knsp.CEL
C:\GeneChip\Affy_Data\Data\ ALL1-T-250knsp.CEL
C:\GeneChip\Affy_Data\Data\ ALL2-N-250knsp.CEL
C:\GeneChip\Affy_Data\Data\ ALL2-T-250knsp.CEL

Make sure that the ".CEL" suffix is added to each line.

Read the data into dChip**.**

Go to **Analysis → Open Group**
Enter a session name under "Group Name". Click either "data directory" or "data file list" and navigate to the CEL/TXT/CHP directory, or data file list respectively. Check "CEL". Specify the suffix of the SNP call file (TXT or CHP). Click "options" and specify a working directory. For convenience, check "DCP files in the working directory". For large analyses, **uncheck** "load probe data into memory". Click OK. Click the "other information" tab. Specify the array CDF (can be downloaded either from www.dChip.org or from Affymetrix). Specify array type (100K or 500K). At "sample" specify the sample information file created earlier.
Click OK. dChip should now read in the array cdf (and convert it to a binary file) and then read in the CEL and TXT/CHP files for each array. When completed, select **Tools → Array List File** and specify the array list file created earlier.

Compute model-based signal values
When reading in array data is completed, go to **Analysis → Normalize & Model**
Important: do *not* normalize data here.
Ignore the "baseline array" option.
Uncheck "perform normalization" and "view normalization plot". Make sure "Compute model-based expression/signal values" is checked.
Click "options". For "model method" select" "Model-based expression". For "background subtraction", choose "Mismatch probe (PM/MM difference)". Click OK, then OK again. dChip should start to model the data.

Export summarized signal data.
You should now see "Modelled" in the lower right of the dChip window (but not "Normalized").
Select **Tools → Export Expression Value**.
In the dialog window, ctrl- or shift-click in "arrays to be exported" to select the arrays to be exported (you will probably want to select all).
Click the box below "output file" and navigate to a directory and specify a file name with a ".txt" suffix to save the data. Check "Has both signal and call".
Check the format of the exported file using a text editor (e.g. TextPad, www.textpad.com). Each array should have a signal intensity column, and a genotype column. Use this file for karyotype-guided normalization.

probe set	ALL1-N-250knsp	ALL1-N-250knsp call	ALL1-T-250knsp	ALL1-T-250knsp call
SNP_A-1886933	67.94	AB	97.66	AB
SNP_A-1902458	443.53	AB	395.45	AB
SNP_A-2131660	239.29	AB	130.59	AB
SNP_A-4221087	149.67	A	172.99	A
SNP_A-1884606	476.98	AB	386.08	AB
SNP_A-2235839	571.06	AB	583.2	AB
SNP_A-2218153	57.19	AB	51.28	NoCall
SNP_A-1919019	1466.42	B	1596.11	B
SNP_A-1815933	808.19	B	916.24	B
SNP_A-2115098	536.27	A	542.36	A
SNP_A-2264565	97.6	B	102.55	B
SNP_A-1788728	2243.18	B	1640.67	B
SNP_A-4218776	364.08	A	250.55	A
SNP_A-2082194	675.13	A	525.93	A
SNP_A-2135694	192.83	B	178.73	B
SNP_A-4220764	123.05	NoCall	101.96	B
SNP_A-1796345	64.56	B	74.89	B
SNP_A-1789472	889.39	B	915.06	B
SNP_A-1934233	550.67	B	555.07	B
SNP_A-4207031	537.35	B	545.62	B
SNP_A-2305014	121.43	A	75.85	A
SNP_A-2179242	189.81	A	156.35	A

Using dChip to Perform Copy number Analysis with reference-marker normalized data

To read in the normlized file, **Analysis → Get external data** and click "Data file" to specify the normalized data file. Under the "other information" tab, specify the sample information file as before. Proceed with copy number/LOH analysis.

Combining array data sets.
Normalized data for identical samples from different Affymetix SNP array platforms (Hind, Xba, Sty and Nsp) can be combined as follows.
Open each normalized data file (e.g. one for Sty, one for Nsp) in a text editor and check that the column headers match.
For small data sets, copy and paste one file below the other, and save as a new file. Delete the second header line halfway down the file.
For large data sets, open the second data file (e.g. Nsp data) using **Analysis → Get external data** as described above. Once loaded and the correct array list file has been specified (to ensure column ordering is correct), export the data (**Tools → Export Expression Value).** Under "Output file" specify the first set of array data (e.g. Sty) and then check "append to this file" and check OK.

This combined file can then be read into dChip. For copy number analysis, a combined genome information file, preferably with the same ordering of SNPs as the data file (for speed of loading) will be required. We have successfully combined 615,000 SNP data (Hind, Xba, Sty and Nsp) for over 300 samples using this approach.

Details on Lower Level Routines

Software developers may have interest in some of the lower level routines defined by `refnorm-library-post.R`. The routine `ref.norm` performs the reference normalization for one vector of signals with a corresponding vector indicating reference marker status. This routine could be used to perform reference normalization for a set that cannot be defined by a set of chromosomes. The routine `vcut.file` writes specific columns of an input file to another smaller file. This routine is used to generate small temporary data files that can be read into memory. The routine `vpaste.files` is used to generate a file that is the row-by-row concatenation of the input files. It is used to paste the temporary normalized files back together into one result file.

Works Cited

Lin M, Wei LJ, Sellers WR, Lieberfarb M, Wong WH, Li C (2004). dChipSNP: significance curve and clustering of SNP-array-based loss-of-heterozygosity data. *Bioinformatics*, 20:1233-1240.

Mullighan CG, Goorha S, Radtke I, Miller CB, Coustan-Smith E, Dalton JC, Girtman K, Mathew S, Ma J, Pounds SB, Su X, Pui C-H, Relling MV, Evans WE, Shurtleff SA, and

Downing JR (2007a). Genes regulating B cell development are mutated in acute lymphoid leukaemia. *Nature*, 446: 758-764.