Introduction

This document provides a tutorial for an R implementation of the DiNAMIC procedure introduced in the paper "DiNAMIC: A Method To Identify Recurrent DNA Copy Number Aberrations in Tumors" by Walter, Nobel, and Wright. All files described in this document can be found in the directory http://www.bios.unc.edu/research/genomic_software/DiNAMIC. There are three sections in the document: (1) Definitions of Symbols and Function Arguments, (2) Descriptions of Functions, and (3) A Sample Session Using DiNAMIC. First-time users may wish to skip to Section 3, which contains the R code available at Sample_Session_code, as well as a description of the associated output.

Section 1: Definitions of Symbols and Function Arguments

n = the number of subjects.

m = the number of markers.

x = an $n \times m$ matrix of numeric copy number data from markers in chr1 - chr22. Copy number data from chrX and chrY should not be included. Column headers are not assumed to be present. See wilmsdata.nobias.txt for an example.

marker.data = an $m \times r$ data frame containing information about the markers in chr1 - chr22, where r is at least 2. Additional marker information - e.g. rs numbers, marker names, etc. - may appear in columns 3 through r, but this information is not required. Column headers are assumed to be present, and the headers will be used to label the output. See wilms.markers.txt for an example.

annot.file = a four-column matrix containing cytoband annotation data. Each row contains the chromosome number (column 1), the cytoband start locus in base pairs (column 2), the cytoband end locus in base pairs (column 3), and the cytoband (column 4). Although information about cytobands does not appear in the output, this information is used internally by the 'peeling' function. Column headers are not assumed to be present. See annot.file.txt for an example.

num.perms = a positive integer indicating the number of cyclic shifts used when creating the empirical null distribution for the Detailed Look or Quick Look procedures.

num.iters = a positive integer indicating the number of aberrant markers that will be analyzed.

gain.loss = a character string that is used to determine whether copy number gains or losses are analyzed. Copy number gains are analyzed by default (gain.loss = "gain"), but losses can be analyzed by defining gain.loss = "loss."

random.seed (optional) = a positive integer that specifies the value of the random seed. The default value of random.seed is NULL.

Section 2: Descriptions of Functions

zero.make = function(binary.vec, start.site)
Input: (1) A binary vector of length m (binary.vec), (2) a natural number (start.site).
Output: a binary vector of length m (output.vec) that has only one of the contiguous strings of 1's that appears in binary.vec, namely the one at start.site.
Description: For convenience we write k instead of start.site. The k^{th} entry of output.vec is defined to be 1. Starting at $j = k + 1$ and continuing for all $j > k$, the j^{th} entry of output.vec is 1 if and only if (1) the j^{th} entry of binary.vec is 1, and (2) for all l such that $k \leq l \leq j$, the l^{th} entry of binary.vec is 1. The case $j < k$ is handled similarly.
Requires these functions: none.
Is called by these functions: peeling.

make.cytoband = function(marker.data, annot.file)
Input: (1) An $m \times r$ data frame of marker data (marker.data), (2) a four-column matrix containing cytoband annotation data (annot.file).
Output: A vector of length m (output.vec) that contains the cytoband location of each marker.
Description: This function finds the cytoband location of each marker.
Requires these functions: none.
Is called by these functions: peeling.

find.null = function(x, num.perms, random.seed)
Input: (1) an $n \times m$ data matrix (x), (2) a natural number (num.perms), (3) a random seed (random.seed) with default value equal to NULL.
Output: A vector of length num.perms (shift.vec)
Description: This function uses num.perms iterations of the cyclic shift procedure to find an empirical null distribution for either $T_{\text{gain}}(X)$ or $T_{\text{loss}}(X)$.
Requires these functions: none.
Is called by these functions: detailed.look, quick.look.

peeling = function(x, marker.data, cytoband, k)
Input: (1) An $n \times m$ data matrix (x), (2) an $m \times r$ data frame of marker data (marker.data), (3) a vector of length m (cytoband) that contains the cytoband for each marker, and (4) a natural number (k).
Output: A list containing the following items: (1) An $n \times m$ data matrix (final.matrix) obtained by applying the peeling procedure to marker k in x, (2) the peak interval containing k.
Description: This function applies the peeling procedure to the input data matrix x starting at marker k.
Requires these functions: zero.make.
Is called by these functions: detailed.look, quick.look.

detailed.look = function(x, marker.data, annot.file, num.perms, num.iters, gain.loss = "gain", random.seed = NULL)
Input: (1) An $n \times m$ data matrix (x), (2) a $m \times r$ data frame of marker data (marker.data), (3) a cytoband annotation file (annot.file), (4) a natural number (num.perms), (5) a natural number (num.iters), and (6) a character string (gain.loss) with default value equal to "gain", and (7) an optional natural number (random.seed) with default value equal to NULL.
Output: A list containing two items: (1) a num.iters $\times (r+1)$ matrix B containing information about the most aberrant markers, and (2) a num.iters $\times r$ two-dimensional array of lists C containing information about the peak intervals around each aberrant marker. For any number l such that $1 \leq l \leq$ num.iters, the l^{th} row of B or C contains information about the l^{th} most aberrant marker k_l or the peak interval around k_l, respectively. The l^{th} row of B consists of the k_l^{th} row of marker.matrix, as well as the p-value for k_l. The $C[l,1]$ and $C[l,2]$ entries of C are lists containing the chromosome and the genomic position of the endpoints, respectively, of the l^{th} peak interval. For $j \geq 3$ the $C[l,j]$ entry of C is a list containing all distinct entries in the j^{th} column of marker.matrix that appear in the l^{th} peak interval, although naturally these lists do not exist if marker.data has only two columns.
Description: This function applies the Detailed Look procedure to the input data matrix x. Copy number gains are analyzed by default, but losses are analyzed if gain.loss = "loss." The function finds the num.iters most aberrant markers of the appropriate type (gains or losses), assesses their statistical significance, peels them, and finds the peak interval around each peeled marker.
Requires these functions: find.null, marker.finder, peeling, zero.make, make.cytoband.
Is called by these functions: none.

quick.look = function(x, marker.data, annot.file, num.perms, num.iters, gain.loss = "gain", random.seed = NULL)
Input: (1) An $n \times m$ data matrix (x), (2) a $m \times r$ data frame of marker data (marker.matrix), (3) a cytoband annotation file (annot.file), (4) a natural number (num.perms), (5) a natural number (num.iters), and (6) a character string (gain.loss) with default value equal to "gain", and (7) an optional natural number (random.seed) with default value equal to NULL.
Output: A list containing two items: (1) a num.iters $\times (r+1)$ matrix B containing information about the most aberrant markers, and (2) a num.iters $\times r$ two-dimensional array of lists C containing information about the peak intervals around each aberrant marker. For any number l such that $1 \leq l \leq$ num.iters, the l^{th} row of B or C contains information about the l^{th} most aberrant marker k_l or the peak interval around k_l, respectively. The l^{th} row of B consists of the k_l^{th} row of marker.matrix, as well as the p-value for k_l. The $C[l,1]$ and $C[l,2]$ entries of C are lists containing the chromosome and the genomic position of the endpoints, respectively, of the l^{th} peak interval. For $j \geq 3$ the $C[l,j]$ entry of C is a list containing all distinct entries in the j^{th} column of marker.matrix that appear in the l^{th} peak interval, although naturally these lists do not exist if marker.data has only two columns.
Description: This function applies the Quick Look procedure to the input data matrix x. Copy number gains are analyzed by default, but losses are analyzed if gain.loss = "loss." The function finds the num.iters most aberrant markers of the appropriate type (gains or losses), assesses their statistical significance, peels them, and finds the peak interval around each peeled marker.
Requires these functions: find.null, marker.finder, peeling, zero.make, make.cytoband.
Is called by these functions: none.

make.shift.array = function(seg.matrix, marker.data)
Input: (1) A six-column matrix (seg.matrix) produced by the segmentation algorithm DNAcopy, (2) an $m \times r$ data frame of marker data (marker.data).
Output: An $n \times m$ matrix.
Description: This function converts the six-column output of DNAcopy to an $n \times m$ matrix that can be analyzed with the cyclic shift procedure.
Requires these functions: none.
Is called by these functions: bias.correction.

bias.correction = function(x, marker.data)
Input: (1) An $n \times m$ data matrix (x), (2) an $m \times r$ data frame of marker data (marker.data).
Output: An $n \times m$ matrix.
Description: This function applies the bias-correction procedure to the input matrix x.
Requires these functions: make.shift.array. Also, the package DNAcopy is used.
Is called by these functions: none.

Section 3: A Sample Session Using DiNAMIC

We now illustrate DiNAMIC using the Wilms' tumor data of Natrajan et al. (2006) and the code provided in the file Sample__Session__code.

1. Load the bias-corrected version of the Wilms' tumor copy number data

   wilmsdata.nobias =

   read.table("http://www.bios.unc.edu/research/genomic_software/DiNAMIC/

   wilmsdata.nobias.txt", sep = "\t", header = FALSE)

   wilmsdata.nobias = as.matrix(wilmsdata.nobias)

   Note: This copy number matrix only contains data from chr1 - chr22. If you have copy number data from chrX and/or chrY, be sure to remove it before using DiNAMIC.

2. Load the Wilms' tumor marker data

   wilms.markers =

   read.table("http://www.bios.unc.edu/research/genomic_software/DiNAMIC/

   wilms.markers.txt", sep = "\t", header = TRUE)

   Note: This marker file only contains information about the markers in chr1 - chr22. If you have marker information from chrX and/or chrY, be sure to remove it before using DiNAMIC.

3. Load the cytoband annotation file

   annot.file =

   read.table("http://www.bios.unc.edu/research/genomic_software/DiNAMIC/

   annot.file.txt", sep = "\t", header = FALSE)

   annot.file = as.matrix(annot.file)

4. Define other input parameters

   num.perms = 100 (the number of cyclic shifts used to define the null distribution)

   num.iters = 10 (the number of aberrant markers assessed)

5. Use the Detailed Look procedure to assess the significance of the 10 most aberrant markers in the bias-corrected version of the Wilms' tumor data

   try.it = detailed.look(wilmsdata.nobias, wilms.markers, annot.file, num.perms, num.iters, gain.loss = "gain", random.seed = NULL)

   Recall that try.it is a list consisting of two elements. The first is try.it[[1]], which equals the following matrix:

Chromosome	Position	Name	p-Value
1	155656176	R:A-MEXP-192:RP11-393K10	0.01
12	38270107	R:A-MEXP-192:RP11-519E12	0.01
8	4554176	R:A-MEXP-192:RP11-337D8	0.01
12	4666536	R:A-MEXP-192:RP11-417H1	0.01
7	142791137	R:A-MEXP-192:RP11-373O9	0.01
8	142097125	R:A-MEXP-192:RP11-534I23	0.01
9	135068909	R:A-MEXP-192:RP11-311J21	0.01
6	7302660	R:A-MEXP-192:RP11-318G16	0.01
6	147799209	R:A-MEXP-192:RP11-434K8	0.01
20	24934874	R:A-MEXP-192:RP11-461C10	0.01

   The second element of try.it is a two-dimensional array of lists. In particular, try.it[[2]] is

Chromosome	Position	Name
1	Integer,2	Character,179
12	Integer,2	Character,131
8	Integer,2	Character,62
12	Integer,2	Character,62
7	Integer,2	Character,158
8	Integer,2	Character,91
9	Integer,2	Character,78
6	Integer,2	Character,72
6	Integer,2	Character,86
20	Integer,2	Character,29

The value of unlist(try.it[[2]][1, 2]) is the vector c(142391262, 244273234), whose entries contain the genomic positions of the endpoints of the peak interval around the most significant marker in wilmsdata.nobias. A total of 179 markers lie in this interval, and their names are contained in the list try.it[[2]][1, 3]. For example, the names of the last two markers in the peak interval are unlist(try.it[[2]][1, 3])[178:179] = c("R:A-MEXP-192:RP11-551G24", "R:A-MEXP-192:RP11-438H8").

6. Use the Quick Look procedure to assess the significance of the 10 most aberrant markers in the bias-corrected version of the Wilms' tumor data. Here we specify a value for the random seed.

   try.it.again = quick.look(wilms.nobias, wilms.markers, annot.file, num.perms, num.iters, gain.loss = "gain", random.seed = 12345)

   Although the Detailed Look and Quick Look procedures produce output with the same format, the output itself need not be identical. The matrix try.it.again[[1]] is given below, and we see that the p-values are not the same as those that appear in try.it[[1]]. This example is illustrative of the additional power that Detailed Look has to detect aberrant markers when compared to Quick Look. We note that try.it.again[[2]] is identical to try.it[[2]].

Chromosome	Position	Name	p-Value
1	155656176	R:A-MEXP-192:RP11-393K10	0.01
12	38270107	R:A-MEXP-192:RP11-519E12	0.01
8	4554176	R:A-MEXP-192:RP11-337D8	0.01
12	4666536	R:A-MEXP-192:RP11-417H1	0.01
7	142791137	R:A-MEXP-192:RP11-373O9	0.01
8	142097125	R:A-MEXP-192:RP11-534I23	0.01
9	135068909	R:A-MEXP-192:RP11-311J21	0.02
6	7302660	R:A-MEXP-192:RP11-318G16	0.09
6	147799209	R:A-MEXP-192:RP11-434K8	0.17
20	24934874	R:A-MEXP-192:RP11-461C10	0.18