**Classification model for distinguishing copy number variation from cancer-related alterations: software description**

Irina Ostrovnaya ([ostrovni@mskcc.org](mailto:ostrovni@mskcc.org)), Adam Olshen ([olshena@mskcc.org](javascript:sendMail('olshena@mskcc.org')))

This file contains instructions on using models for CNVs prediction developed in our manuscript. All the analyses are done in R (http://www.r-project.org), and require package randomForest (http://cran.r-project.org/web/packages/randomForest/index.html).

library(randomForest)

Two files are necessary for applying prediction models: “prediction code.R” and “CNV\_prediction\_objects.RData” available for download from http://www.mskcc.org/mskcc/html/72726.cfm

These two R files have to be downloaded in the current R workspace:

source(“prediction code.R")

load(“CNV\_prediction\_objects.RData”)

To obtain the predictions the following function must be called:

predict.CNVs Classification of segments of tumors gains and losses as germline CNVs or cancer-related CNAs

**Description**

For each segment that satisfies criteria of a “candidate CNV” as described in the manuscript, the function calculates a prediction TRUE (CNV) or FALSE (CNA) based on the random forest prediction model developed on the TCGA glioblastoma data.

**Usage**

predict.CNVs(dataseg,use.cohort=TRUE,smoothed=TRUE, glad=FALSE,gainloss.defined=FALSE,nmad=1)

**Arguments**

dataseg The input, dataseg, is the dataset that consists of genomic locations and log-ratios from the CGH arrays of tumors of interest, as well as their segmentation results. The dataset has to be in the format of the output of function ‘segment’ from the Bioconductor package ‘DNAcopy’ that implements the Circular Binary Segmentation algorithm, although the segmentation doesn’t have to be obtained by any specific algorithm.

Object dataseg has to contain two sub-objects:

* dataseg$data is a data frame with log ratios. First column is a chromosome (number between 1 and 22), second column is a genomic location (in bases). The names for the first two columns have to be “chrom” and “maploc”. The remaining columns correspond to samples. For example:

chrom maploc Sample1 Sample2

1 554268 0.4431502 -0.1088813

1 554287 0.8610094 1.6408030

1 639581 NA 0.4602401

1 736483 0.1829559 0.2692948

* dataseg$output contains segmentation results in the following format: a data frame with six columns. Each row of the data frame contains a segment for which there are six variables: the sample id, the chromosome number, the map position of the start of the segment, the map position of the end of the segment, the number of markers in the segment, and the average value in the segment.

For example:

ID chrom loc.start loc.end num.mark seg.mean

Sample1 1 554268 150819451 10646 0.0102

Sample1 1 150823073 150848509 5 1.4890

Sample1 1 150852858 151027357 31 -0.0083

Sample1 1 151033188 151036435 2 -5.4296

If the gain/loss calls are already determined, dataseg$output might contain an additional column ‘state’ with “Gain”, “Loss”, or “Normal” call for each segment.

use.cohort Is TRUE if in order to predict CNVs in any particular sample the model should use data from candidate segments of other samples in the cohort. If there are multiple samples and they are independent use.cohort=TRUE is likely to lead to more accurate prediction.

smoothed Is TRUE if outliers were removed prior to segmenting the data.

glad TRUE if the model from the manuscript developed based on GLAD segmentation is preferred (assumes smoothed=TRUE).

gainloss.defined TRUE if dataseg$output contains additional column ‘state’ with “Gain”, “Loss”, or “Normal” call for each segment. Otherwise, MAD criteria will be used to assign these calls.

nmad Number of MADs (median absolute deviation) that is used for Gain/Loss calls if gainloss.defined=FALSE. For each array MAD of its residuals (that is, data minus segmentation means, which represent the noise revel) is calculated. Any segment of this array that has a mean at least nmad MADs above or below arrays’s median is called a gain or loss.

**Output**

The function returns data frame dataseg$output that contains predictors as described in the manuscript. Column predicted.CNV has TRUE if a candidate segment is predicted to be a CNV, FALSE if it is predicted to be a CNA and NA for non-candidate segments. This data frame potentially has fewer rows than dataseg$output since adjacent candidate segments of the same sign are merged.