Create Signature

Jeffrey T. Chang, William T. Barry, Joseph E. Lucas, Quanli Wang,
Joseph R. Nevins, and Mike West
2 November 2011

Summary: This module generates a gene expression signature from a *training set* that consists of a *train0* file and a *train1* file. The *train0* file contains gene expression samples (typically 3-10 samples) that are representative of a state of interest, and the *train1* file contains samples that represent the opposite state (for example, train0 might contain cells where a signaling pathway is inactive, and train1 contains cells where the pathway is active). The module will use this information to generate a gene expression signature. If a *test* file is also provided, it will predict the cellular state in the samples in this file.

For more examples on the use of this module, see:

1. Huang E, *et al*. Gene expression phenotypic models that predict the activity of oncogenic pathways. *Nat Genet*. 34(2):226-30, 2003.
2. Bild AH, *et al*. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*. 439(7074):353-7, 2006.

Parameters: train0, train1, test
train0 and *train1* contain the gene expression data that is representative of the cellular states of interest. *test* is an optional data set. If it is given, CreateSignature will predict the degree that each sample (in the file) resembles the state in *train1* over *train0*. These files can be provided in PCL, GCT, and some other formats.

Parameter: version
This sets the version of BinReg that is used to construct the model. By default, the version is 2. You should not change this unless you know what you are doing.

Parameters: num_genes, num_metagenes
This determines the number of genes and metagenes used to construct the model. These are determined empirically. *num_genes* is typically between 100 and 500, and *num_metagenes* between 2 and 5. You can provide either single values (e.g. "250") or multiple ones (e.g. "100,150,200,250,300").

Parameters: apply_quantile_normalization, apply_shiftscale_normalization, apply_dwd_normalization, apply_dwd_bild_normalization
These parameters are used to account for technical differences between the training and test data sets. By default, none of these are run, but each one can be activated. The options are to normalize by quantiles, Shift-Scale, DWD, or a variant of DWD used by the Bild laboratory.

Parameters: normalization_reference
Ordinarily, Shift-Scale will normalize the mean and variance of the test data set to fit that of the training data set. This means the prediction of each sample in the test data depends on what else is in the test data set. Sometimes this is not desirable. To address this, you can specify a data set to serve as a *normalization reference*. Each of the samples in the test data will be normalized in the context of this reference. This way, the predictions will not depend on which other samples are in the data set. Instead, they will depend on the contents of this reference.

Parameter: log_the_data
Whether the data files need to be logged. The algorithm requires gene expression signal values to be transformed by taking the log (base 2). RMA normalization does this automatically, but MAS5 does not. The default value is *auto*, which means that the module will automatically detect whether the files need to be logged. You should not have to change this--it will do the right thing in nearly all cases. However, you can force the algorithm to log the files by setting this parameter to *yes*, or force no logging by setting it to *no*.

Parameters: burnin, niter, skips
These parameters are for the Markov Chain Monte-Carlo simulation use to sample the model. They determine the number of samples for burn-in and for sampling the model. *skips* indicates which samples are used for the model. That is, if *skips* is 2, then sampling algorithm will only use every 2^{nd} sample for model and discard the rest.

Parameters: label_samples, draw_error_bars, ci
These parameters are used to control the plotting of the predictions. *label_samples* will determine whether the sample names are labeled on the plot. *draw_error_bars* will determine whether the error bar is drawn. *ci* sets the credible interval shown by the error bars.

For help, please email: jeffrey.chang@duke.edu.