RankNormalize Documentation

Description: Rank a dataset within columns and normalize across the rows.

Author: Pablo Tamayo, David Eby, gp-help@broadinstitute.org

Summary

This module ranks a dataset within columns and then normalizes the ranked data set by row, a process geared toward making features more comparable with one another. Rank normalization allows datasets created on different platforms or with different experimental parameters to be compared. This module is particularly useful for processing files from ExpressionFileCreator or normalizing data files that you intend to analyze with the NMF (Non-negative Matrix Factorization) module for clustering or projection.

The ranking algorithm creates a table in which the data values in the file are changed into ranks from lowest to highest in each column. Then the rank orders are normalized by dividing each rank by the total number of samples.

For example:

- Given a microarray consisting of a set of 5 probes and 2 samples:

#1.2			
5	2		
Name	Description	SampleA	Sample B
probe1	gene1	1.2	7.0
probe2	gene2	0.8	4.4
probe3	gene3	3.5	3.7
probe4	gene4	8.4	9.0
probe5	gene5	0.5	0.4

- The ranking algorithm orders them in ascending order:

#1.2			
5	2		
Name	Description	SampleA	Sample B
probe1	gene1	3	4
probe2	gene2	2	3
probe3	gene3	4	2
probe4	gene4	5	5
probe5	gene5	1	1

- The algorithm then normalizes them by dividing by the number of objects in the row (that is, 2):

#1.2			
5	2		
Name	Description	SampleA	Sample B
probe1	gene1	3.5	3.5
probe2	gene2	2.5	2.5
probe3	gene3	3	3
probe4	gene4	5	5
probe5	gene5	1	1

Ties in ranking are broken using the “average” method. That is, for a data set with N rows, each column will have its data points ranked from 1 (least) to N (greatest), with the score for any ties being averaged over those ties.

For example:

- Given a microarray with seven probes that you use to test two samples, getting the following results:

#1.2			
7	2		
Name	Description	Sample A	Sample B
probe1	gene1	1.2	7.0
probe2	gene2	0.8	4.4
probe3	gene3	3.5	3.7
probe4	gene4	8.4	9.0
probe5	gene5	0.5	0.4
probe6	gene6	1.2	7.0
probe7	gene7	1.2	6.5

- Using this as input to RankNormalize with default settings produces the following output, where the intensity values have been ranked from 1 to 7 in each column, with several ties ranked at 4 (for Sample A) and 5.5 (for Sample B):

#1.2			
7	2		
Name	Description	Sample A	Sample B
probe1	gene1	4	5.5
probe2	gene2	2	3
probe3	gene3	6	2
probe4	gene4	7	7
probe5	gene5	1	1
probe6	gene6	4	5.5
probe7	gene7	4	4

Scaling

The scale of the results is controlled by the *scale to value* parameter. By default, for a data set with N rows, the results will be scaled to values in the interval [1, N], directly passing the row count as the *scale to value*. If the user sets this parameter, it will override that default value. For instance, the use of 1 as the *scale to value* will scale results on the interval [0, 1] (or more precisely, on the interval [1/N, 1]). Using 100 will scale results to percentage values.

Parameters for Adjusting the Data Set Before Ranking

There are three additional parameters that can be used to constrain or adjust the data set before ranking:

- The *threshold* parameter sets the minimum for values in the data set. Any value below this will be increased to *threshold* before ranking.
- The *ceiling* parameter sets the maximum for values in the data set. Any value below this will be decreased to *ceiling* before ranking.
- The *shift* parameter gives an amount to adjust values in the data set. This *shift* adjustment value will be added to every value in the data set before ranking.

Note: If more than one of these are specified, the order of precedence is *threshold*, then *ceiling*, then *shift*.

These parameters are used to restrict or adjust the dynamic range of the data set, repositioning the data points (*shift*) or enforcing minimum (*threshold*) or maximum (*ceiling*) boundaries on their values. This might be necessary, for instance, with legacy microarray data sets, or if you suspect or know that your data is noisy in the high or low ranges. In the past, it was possible for certain microarray instruments to report negative intensity values and, in general, the scale of values for these was found to be noisy at the bottom of the range, causing issues in downstream analyses. Similar issues may also be seen at the upper end of the range.

References

R: A Language and Environment for Statistical Computing, Reference Index, Version 2.15.2 (2012-10-26), by The Core R Team.

Parameters

Name	Description
input file (required)	The dataset to be normalized in GCT or RES format.
output file name (required)	The name to be given to the output file (defaults to <input.file_basename>.NORM.<input.file_extension>)
scale to value (optional)	Result values will be scaled to this value by multiplication after normalization. Leaving this blank will give results scaled from 1 to N (where N is the number of rows).
threshold (optional)	This parameter is used to restrict or adjust the dynamic range of the data set, by setting a minimum threshold for values in the data set. Any data set value below this will be increased to the threshold value before normalization. This might be necessary, for instance, with legacy microarray data sets, or if you suspect or know that your data is noisy in the high or low ranges. If more than one of the *threshold*, *ceiling*, and *shift* parameters is specified, the order of precedence is *threshold*, then *ceiling,* then *shift*.
ceiling (optional)	This parameter is used to restrict or adjust the dynamic range of the data set, by setting a maximum ceiling for values in the data set. Any data set value above this will be decreased to the ceiling value before normalization. This might be necessary, for instance, with legacy microarray data sets, or if you suspect or know that your data is noisy in the high or low ranges. If more than one of the *threshold*, *ceiling*, and *shift* parameters is specified, the order of precedence is *threshold*, then *ceiling,* then *shift*.

shift (optional)	This parameter is used to restrict or adjust the dynamic range of the data set, by repositioning all values in the data set. The shift value will be added to all data set values before normalization. This might be necessary, for instance, with legacy microarray data sets, or if you suspect or know that your data is noisy in the high or low ranges. If more than one of the *threshold*, *ceiling*, and *shift* parameters is specified, the order of precedence is *threshold*, then *ceiling*, then *shift*.

Input Files

1. <input file>
   A data set in GCT or RES file format to be normalized by rank.

Output Files

1. <output file name>
   The resulting normalized ranked data set. The output format will match the input format, either GCT or RES. Any calls in a RES file will be maintained unchanged.

Requirements

The RankNormalize module requires R2.15.2 with the following packages:

- getopt_1.17
- optparse_0.9.5

The RankNormalize module uses R's built-in "rank" function.
These R packages will be automatically downloaded and installed when the module is installed.
R2.15.2 must be installed and configured independently; for more information, see the GenePattern Administrator's Guide:
http://www.broadinstitute.org/cancer/software/genepattern/gp_guides/administrators-guide/sections/r-versions.

Platform Dependencies

Module type:	Statistical Methods
CPU type:	Any
OS:	Any
Language:	R2.15.2

GenePattern Module Version Notes

Date	Version	Description
2/19/13	1	Initial version.