SHARPR (Systematic High-resolution Activation and Repression Profiling with Reporter-tiling) User Manual (v1.0.2)

Email any questions to Jason Ernst (jason.ernst@ucla.edu)

Overview

SHARPR is software for analyzing Massively Parallel Reporter Assay tiling designs allowing the identification at high resolution of activating and repressive nucleotides across many regulatory elements.

SHARPR can be run on any computer supporting Java 1.6 or later. SHARPR is executed from the command line with a command such as:

java -mx4000M -jar SHARPR.jar *Command* [commandoptions] commandparameters

where the 4000 specifies the amount of memory given to Java and could be adjusted based on the size of the data and the *Command* being executed. In some cases the memory flag could be omitted.

ExecuteAll – The command effectively executes the Normalize, ConvertTable, Infer, Combine, and Interpolate commands in series to go from raw counts of multiple overlapping tiles to inferred base resolution predictions of regulatory activity without printing any intermediary files.

Normalize – Takes the raw RNA and DNA counts returns a single normalized log ratio value for each unique reporter sequence.

ConvertTable – Converts the output of the Normalize command into a table format where each row corresponds to a unique region and each column different tile positions of the region.

Infer – This command takes as input the report value at each position and then infers an activity score at the resolution of the tiling step size.

Combine – This command produces a single output based on one or more experiments that should be considered as replicates and optionally multiple different parameter settings.

Interpolate – The command takes a file where the input is the inferred values at the tiling step size and the output has base resolution predictions

Enrich – This command enables computing the enrichment of base level activity inferences for an arbitrary set of coordinates.

AdjacentChanges – This command analyzes adjacent sequence tiles with multiple different bar-codes to identify those adjacent reporter sequences with a significant difference in their associated expression level.

ExecuteAll

Description

The command effectively executes the Normalize, ConvertTable, Infer, Combine, and Interpolate commands in series to go from the raw counts of multiple overlapping tiles to inferred base resolution predictions of regulatory activity without printing any intermediary files.

Usage

```
ExecuteAll [-c cutoff][-p pseudcountrna][-d pseudocountdna][-v1
varprior1][-v2 varprior2] rnafilelist dnafilelist npositionindex
tilelength stepsize numtilepos outfile
```

Required Parameters

`rnafilelist` – This parameter specifies a comma delimited list of the names of the files containing the RNA data. Each file will correspond to one experiment that will be treated as replicates. The format of these files are the first column contains an identifier for the reporter sequence followed by tab delimited columns for the RNA counts for each of the unique bar-codes for the sequence. The IDs should contain within it an integer 0-index value specifying which tile position it corresponds to delimited by '_'. All other values within the ID should uniquely identify the region. The first line is a header line with headers for each of the bar-code columns, but not the ID column. Below is a small example if there was two bar-codes per-unique reporter sequence. Note in some cases there might be only one bar-code column, but the header line is still expected.

```
          1     2
ID1_0 170   1328
ID1_1 194   483
ID2_0 222   244
```

`dnafilelist` - This parameter specifies the names of the files containing the DNA data corresponding to the RNA data in a comma delimited list. There should be one DNA file entry for each RNA file entry. The format of this file is the same as for the RNA data. The DNA file should match the RNA line by line in terms of the IDs of the reporter.

`npositionindex` - The 0-based index of the entry within the reporter ID which is delimited by '_' where the tile position is specified. For example if the reporter IDs are of the form ID1_0, ID1_1, ID_2,… then the `npositionindex` would be 1.

`tilelength` - This parameter specifies the number of base pairs one tile position covers.

`stepsize` - This parameter specifies the number of base pair offset between adjacent tile positions.

`numtilepos` - This parameter specifies the total number of unique reporter tile positions covering a given regulatory region.

`outfile` - This parameter specifies the output file where the final base level predictions should go.

Optional Parameters

`-c cutoff` – This parameter determines a threshold for the minimum number of DNA counts for the read-out corresponding to a specific bar-code to be retained. The pseudocount is not counted in meeting the filtering threshold. Default value is 20.

`-p pseudcountrna` – This is a pseudocount value added to all RNA counts. Default value is 1.

`-d pseudocountdna` – This is a pseudocount value added to all DNA counts. Default value is 1.

`-v1 varprior1` - This parameter specifies one variance prior. A larger value parameter might capture regulatory bases at higher resolution but also overfit, while a smaller value leads to an overall smoother fit but also may underfit. Default value is 1.

`-v2 varprior2` - This parameter specifies a second variance prior. If different than `varprior1` then the inferred value with a smaller absolute value is used except if the signs disagree in which case 0 is used. Default value is 50.

Normalize

Description

This command takes RNA-counts and DNA-counts for a MPRA experiment and returns a normalized $\log_2$ ratio for each unique sequence. In computing the normalized $\log_2$ ratio the RNA and DNA counts are first divided by their total counts in the experiment. If there are multiple bar-codes for the same sequence the median value is reported.

Usage

```
Normalize [-c cutoff][-p pseudcountrna][-d pseudocountdna][-n
noavgoutfile] rnafile dnafile outfile
```

Note items in [] are optional

Required Parameters

`rnafile` – This parameter specifies the name of the file containing the RNA data from one experiment. The format of this file is the first column contains an identifier for the reporter sequence followed by tab delimited columns for the RNA counts for each of the unique bar-codes for the sequence. The IDs should contain within it an integer 0-index value specifying which tile position it corresponds to delimited by '_'. All other values within the ID should uniquely identify the region. The first line is a header line with headers for each of the bar-code columns, but not the ID column. Below is a small example if there was two bar-codes per-unique reporter sequence. Note in some cases there might be only one bar-code column, but the header line is still expected.

```
          1     2
ID1_0 170   1328
ID1_1 194   483
ID2_0 222   244
```

`dnafile` - This parameter specifies the name of the file containing the DNA data corresponding to the RNA data. The format of this file is the same as for the RNA data. The DNA file should match the RNA line by line in terms of the IDs of the reporter.

`outfile` – This parameter specifies the file where the ratios should be outputted. The format of the output is three columns. The first contains the sequence ID. The second column contains the normalized ratio. The third column contains the number of bar-codes this is based off, which excludes those that did not meet the DNA cutoff. There is no header line in the output.

Optional Parameters

`-c cutoff` – This parameter determines a threshold for the minimum number of DNA counts for the read-out corresponding to a specific bar-code to be retained. The pseudocount is not counted in meeting the filtering threshold. Default value is 20.

`-p pseudcountrna` – This is a pseudocount value added to all RNA counts. Default value is 1.

`-d pseudocountdna` – This is a pseudocount value added to all DNA counts. Default value is 1.

`-n noavgoutfile` – If this option is present, then it specifies a file to output the log-ratio for each bar-code leaving empty those entries for which the denominator did not meet the minimum count threshold. The first column contains the reporter ID, and the following columns contain the log-ratio values. There is no header line in the output.

ConvertTable

Description

The command takes as input data produced by the normalize command and converts it into a table format where each row corresponds to one region. In the output the first column is the ID and the remaining columns corresponds to the normalized value for each tab position or an empty value if the tile was not included. Optionally the values can be recentered by either the overall output means or just the mean value at the two end positions. There is no header line in the output.

Usage

```
ConvertTable [-recentermean][-recenterends] convertinputfile
convertoutputfile npositionindex
```

Required Parameters

`convertinputfile` – The input file with the normalized data that should be converted into table format

`convertoutputfile` – The output file where the data in table format should be written. The output is a tab delimited file where the first column is the region ID, which corresponds to the reporter IDs without the part specifying the specific tile position. The remaining columns have the normalized $\log_2$ ratios at each tile position. If there were not any observations that met the minimum count threshold for a specific position for a region that entry is empty.

`npositionindex` – The 0-based index of the entry within the reporter ID which is delimited by '_' where the tile position is specified. For example if the reporter IDs are of the form ID1_0, ID1_1, ID_2,… then the `npositionindex` would be 1.

Optional Parameters

`-recentermean` – If this flag is present, then the overall mean is subtracted from all values.

`-recenterends` – If this flag is present, then the mean of all values in the two end positions is subtracted from all values.

Infer

Description

This command takes the normalized input data of the reporter values in table format as produced by ConvertTable and produces inferred regulatory activity values for each interval of size `stepsize`. The assumption is made that the `tilelength` is divisible by the `stepsize`. If not satisfied directly, the assumption can still be satisfied by reducing the `stepsize` so it does divide `tilelength` and treating specific tile positions as missing.

Usage

```
Infer [-nostandardize] inputtablefile outputfile varprior tilelength
stepsize numtilepos
```

Required Parameters

`inputtablefile` – This parameter specifies the input file. The input file has the normalized data in table format as produced from the ConvertTable command.

`outputfile` - This parameter specifies the output file where the inferred values should be written. The output format is such that the region identifiers are in the first column and the remaining columns are the inferred values at each interval of length `stepsize`.

`varprior` - This parameter specifies the variance prior. A larger value parameter might capture regulatory bases at higher resolution but also overfit, while a smaller value leads to an overall smoother fit but also may underfit.

`tilelength` - This parameter specifies the number of base pairs one tile position covers.

`stepsize` - This parameter specifies the number of base pair offset between adjacent tile positions.

`numtilepos` - This parameter specifies the total number of unique reporter tile positions covering a given regulatory region.

Optional Parameters

`-nostandardize` – By default the output is standardized by subtracting the mean and dividing by the standard deviation of all the inferred values. If this flag is present then the inferred values without this standardization is returned.

Combine

Description

This command combines inferred values based on one more experiments that should be treated as replicates and optionally based on two different parameter settings.

Usage

```
Combine [-c fileset2] fileset1 outputfile
```

Required Parameters

`fileset1` – A comma delimited list of files that should be considered replicates and their values averaged. These different files should be based on the same variance prior. The format of the file is the first column contains the region ID and each remaining column contains the inferred reporter values. The columns are tab delimited and there is no header line.

`outputfile` – A single output file where the combined values should be written. The format of the output is the same format as the individual input files. If only `fileset1` is specified the combined values are determined based on taking the average values from `fileset1`. If `fileset2` is also specified the combined value is determined by taking the average value from each set and then using the one with the minimum absolute value if they agree on the sign and 0 otherwise.

Optional Parameters

`-c fileset2` – This specifies another set of inferred values for experiments that should be considered replicates. These files should be based on the same variance prior, but a different one than for `fileset1`.

Interpolate

Description

The command takes a file where the input is the inferred values at the interval specified by `stepsize` and the output has base resolution predictions determined between piecewise linear interpolation between adjacent intervals.

Usage

```
Interpolate interpolateinputfile interpolateoutputfile stepsize
```

Required Parameters

`interpolateinputfile` – This parameter specifies the input file that has the inferred values at `stepsize` resolution for which a base resolution interpolation should be made. The format of the file is the first column contains the region IDs. The remaining columns correspond to the inferred values at each `stepsize` interval. Each row corresponds to one region. The columns are tab delimited. There is no header line.

`interpolateoutputfile` – This parameter specifies the output file where the interpolated values should be written. The output file has the same format as the input file except there will be `stepsize` more columns of values.

`stepsize` – This parameter specifies the number of base offset between adjacent tiles and should match what is specified by the `Infer` command.

Enrich

Description

This command enables computing the enrichment of base level activity inferences for an arbitrary set of coordinates. The enrichment is shown based on different bins of activity levels. The bins can be determined to have an equal number of bases or based on the activity level.

Usage

```
Enrich [-b basetooverlap][-e mincountextreme][-f featureoutput][-
maxabsenrich][-n numoverlapbins][-s summaryoutput][-valuebins][-v
value] reportercoordinates overlapcoordinates basepredictions
overlapoutput
```

Required Parameters

`reportercoordinates` – This parameter specifies the file containing the coordinates to which each reporter corresponds. The first column has the region ID, the second column the chromosome of the reporter ID, and the third and fourth columns the start and end coordinates of the sequence. The coordinates are 0-based with the start inclusive and the end exclusive.

`overlapcoordinates` – This parameter specifies a file containing the coordinates that should be overlapped for enrichment. The format of this file is a three column file in bed format with the first column the chromosome, the second and third the start and coordinates. The coordinates are 0-based with the start inclusive and the end exclusive.

`basepredictions` – This parameter specifies a file containing activity level predictions at each base within the reporter sequence interval. The format of the file is each row corresponds to a region where its ID is in the first column and the remaining columns have the base level predictions. This is the format that is produced by the `ExecuteAll` and `Interpolate` commands.

`overlapoutput` – This parameter specifies a file where the output should be written. If the bins were determined based on percentiles to represent an equal number of bases then the columns are as follows: first column is the bin index, then the percentile of the bin (with the highest activity assigned the lowest percentile), then the average activity level of points assigned to the bin, the fraction of bases in the bin that overlap the coordinate set, then the overall hit fraction, then the center base hit fraction, then the cumulative fold enrichment relative to the overall hit fraction when ranking based on the most activating bases, then the cumulative fold enrichment relative to the overall hit fraction when ranking based on the most repressive bases.

Optional Parameters

`-b basetooverlap` - If this parameter is specified then the overlap enrichment is computed based on a specific base position in the reporter sequence where the base positions are 0 indexed relative to the first.

`-e mincountextreme` - If the `valuebins` parameter is specified this parameter determines the minimum number of points that should fall in the bin with the largest and separately the smallest (or most negative) value. If the `valuebins` parameter is not specified, then this parameter is not relevant. Default value is 500.

`-f featureoutput` – If this parameter is specified then `featureoutput` is the name of a file where the features should be printed. The first column is the reported IDs and the remaining columns correspond to each base the reporter covers. There is a value of 1 if the reporter overlaps a coordinate in the `overlapcoordinates` file at that base and a 0 otherwise.

`-n numoverlapbins` – In computing overlaps if the `valuebins` then bins for the enrichment are formed to have an equal number of bases where the number of bins is determined by this parameter. If the `valuebins` parameter is specified, then this parameter is not relevant. Default value is 5000 if `maxabsenrich` is not specified, and 200 if it is specified.

`-s summaryoutput` – If this parameter is specified then a file `summaryoutput` will be created which has summary statistics related to the overlap with positions of the coordinate set. The output in this file includes the Area under a ROC Curve (AUC) at recovering coordinate set bases when ranking based on most activating, as well as the AUC up to false positive rates of 5%, 1%, 0.5%. Also given is the same statistics based on ranking bases in terms of the most repressive activity score and the absolute value of the activity score. Additional statistics include the average absolute activity score for all bases considered and just those that overlap the coordinate set. The fraction of bases the coordinate set overlaps that were considered is also reported. If the `maxabsenrich` flag is present then this is restricted to those bases that have the maximum absolute value within a region. A similar set of statistics is reported for just the center base.

`-maxabsenrich` - If this parameter is specified then only the single position with the maximum absolute regulation score is used for each region in computing the enrichments.

`-valuebins` - If this parameter is specified then enrichments are shown where the activity score has been binned based on the nearest multiple of the `value` parameter. The most extreme bins contain all values less than and greater than them, and are the determined by the `mincountextreme` parameter.

`-v value` - If the `valuebins` parameter is specified, then the bins are determined based on rounding the activity score to the nearest multiple this parameter. If the `valuebins` parameter is not specified, then this parameter is not relevant. Default value is 0.5.

AdjacentChanges

Description

This command analyzes adjacent sequence tiles with multiple different bar-codes to identify those adjacent reporter sequences with a significant difference in their associated expression level based on a Mann-Whitney U-test, which allows high resolution isolation of sequences either activating or repressing gene expression.

Usage

```
AdjacentChanges [-e extension] datafiles stepsize numpositions
sequences coordinates adjacentoutfile
```

Required Parameters

`datafiles` – This parameter is a comma delimited list of files which should be considered as replicate experiments. The format of this file is each row corresponds to a unique reporter sequence. Rows are assumed to be ordered such that adjacent tiles appear on consecutive rows. It is also assumed there is row for every position in each region even if all values were missing. The first column has a unique identifier for the reporter sequence. The following columns are tab delimited and have the normalized log-ratios for the individual bar-codes. One can produce such a file by running the `Normalize` command and providing a `-n` option with the name of the file for which the non-averaged reporter values should be outputted. Extra tabs between values which might represent missing measurements can be included and will be ignored.

`stepsize` – The number of base pairs between the start of adjacent tile positions.

`numpositions` – The number of unique positions of reporter tiles covering a region.

`sequences` – This parameter specifies the name of a file with the DNA-sequence information for the individual reporter sequences. The format of this file is two columns. The first column contains the reporter identifier. The second column contains the DNA-sequence corresponding to the reporter.

`coordinates` – This parameter specifies the name of a file with the coordinates. The format of this file is four columns. The first column is the identifier for the reporter sequence. The second column is the chromosome. The third and fourth columns are the start and end coordinates, following the bed convention of being 0-based with the start coordinate inclusive and the end coordinate exclusive.

`adjacentoutfile` – This parameter specifies the name of the file where output on the analysis of adjacent tiles should be written. The first line of the output file is a header line. The remaining lines correspond to every adjacent pair of tiles. The first two columns give the IDs of the adjacent reporter tiles. This is followed by a p-value and FDR for the two reporters having different associated expression levels. The next column has the unique sequence to the activating tile in the pair extended by the `extension` amount, followed by the same but for the repressive tile. The first tile is the more

activating tiling and the second is the repressive tiling if the p-value for the second tile having lower expression than the first tile has a lower or equal p-value than the p-value for the first tile having lower expression than the second tile, otherwise the second tile is the activating tiling and the first is the repressive one. The next three columns are the coordinates of the activating sequence including extension. The coordinates are in bed format being 0-based, first coordinate inclusive and second exclusive. The next three columns are the same, but for the repressive sequence. The next set of columns report the p-value for reporter 2 having lower expression than reporter 1 in each individual experiment. The next set of experiments report the p-value for reporter 1 having lower expression than reporter 2 in each individual experiment. The next column reports a p-value for reporter 2 having lower expression than reporter 1 based on combining p-values with Fisher's method. The next is the same as the previous column except for reporter 1 having lower expression than reporter 2. The next set of columns reporter the median expression for reporter 1 in each replicate experiment. The next is the same except for reporter 2.

Optional Parameters

`-e extension` – This parameter specifies the number of base pairs into the overlapping regions the region of difference should be extended. This is useful for instance to capture the full regulatory motif that might only partly be contained in the non-overlapping bases. Default value is 10.

Acknowledgements

SHARPR also makes use of The Apache Commons Mathematics Library (v3.3). Funding for development of SHARPR provided by NIH grants R01ES024995, U01HG007912, U01MH105578, R01HG006785, R01GM113708, U01HG007610, R01HG004037, U54HG006991, U41HG007000, an NSF CAREER Award #1254200, and an Alfred P. Sloan Fellowship.