ChromImpute User Manual (v1.0.5)

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

Overview

ChromImpute is Java software for large-scale systematic epigenome imputation. ChromImpute takes an existing compendium of epigenomic data and uses it to predict signal tracks for mark-sample combinations not experimentally mapped or to generate a potentially more robust version of data sets that have been mapped experimentally. ChromImpute bases its predictions on features from signal tracks of other marks that have been mapped in the target sample and the target mark in other samples with these features combined using an ensemble of regression trees.

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

java -mx4000M -jar ChromImpute.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.

ChromImpute has seven top level commands which then determine the required and optional set of parameters. The top level commands are briefly described here and a detailed description of each command, the required and optional parameters can be found in the remaining sections.

Convert – Converts signal tracks into binned signal resolution tracks.

ComputeGlobalDist – Computes the global distance based on correlation for each mark in each sample with the same mark in all other samples. Creates a file for each mark in each sample containing a ranked list of the globally nearest samples.

GenerateTrainData – This command takes a directory of converted data and global distances and generates a set of training data instances.

Train – This command trains regression trees based on the feature data produced by `GenerateTrainData`

Apply – This command applies the predictors generated in the `Train` command to generate the imputed data.

Eval – Compares the agreement between an observed and imputed data set.

ExportToChromHMM – Converts the signal into file formats that can either provided as input to ChromHMM’s BinarizeSignal or LearnModel commands

The usage for any of these commands can be obtained at the command line by typing at the command line

java -jar ChromImpute.jar *Command*

Note on parallelization: For imputing large compendiums of data it is advisable to run ChromImpute in parallel.

Convert

Description

Converts the observed signal into signal at a binned resolution.

Usage

```
Convert [-c chrom][-l convertsample][-m convertmark][-r resolution]
INPUTDIR inputinfofile chrominfofile CONVERTEDDIR

Note items in [] are optional
```

Required Parameters

`INPUTDIR` – The name of the directory containing the files to convert. The files should either be in BedGraph format (`.bedgraph`, `.bedgraph.gz`) or Wig format (`.wig`, or `.wig.gz`) and converts them to at a default resolution of 25bp signal, by averaging the signal at each base overlapping the bin.

`inputinfofile` – The file provides information on the input for the imputation. The first column is the sample, the second column is the mark, and the third is the file name. An optional fourth column can specify an output subdirectory for the converted data for that sample and mark combination. Here is an example without the optional fourth column:

```
sample1 mark1 fileA

sample1 mark2 fileB

sample2 mark1 fileC

sample2 mark2 fileD
```

`chrominfofile` – A two column tab delimited file with the first column being the chromosome and the second being the chromosome length of the chromosomes to include. The fetchChromSizes script available from the UCSC browser (http://hgdownload.cse.ucsc.edu/admin/exe/) specifying the desired assembly and redirecting the output to a text file and removing any unwanted chromosomes can be used to obtain this.

`CONVERTEDDIR` – The name of the directory where the converted input files should be written. Converted files are wig files one per chromosome with a browser header line in addition to the wig header line.

Optional Parameters

`-c chrom` – If this option is specified then only data from the specified chromosome is converted. The data should be present in a file with a prefix chrom_ followed by the file name given in the third column.

`-l convertsample` – If this option is present then only files for this sample are converted

`-m convertmark` – If this option is present then only files for this mark are converted

`-r resolution` – If this option is present then the resolution of the converted signal data will be changed to it. The default value is 25 base pairs.

Recommended Parallelization

For converting large compendiums it is advisable to parallelize over sample and mark combinations using the `–l convertsample` and `–m convertmark` options.

ComputeGlobalDist

Description

By default for each mark in each sample creates a file with a ranked listing of the globally nearest sample based on correlation of the mark in other samples.

Usage

```
java ChromImpute ComputeGlobalDist [-m mark] [-r resolution][-s sample
mark][-x extension] CONVERTEDDIR inputinfofile chrominfo DISTANCEDIR
```

Required Parameters

`CONVERTEDDIR` – the directory containing the converted data in the format produced by the `Convert` command. Note the converted wig files generated by the `Convert` and are assumed by this procedure to have two lines of header information.

`inputinfofile` – is the same file described in the `Convert` command

`chrominfo` – is the same file described in the `Convert` command

`DISTANCEDIR` – the directory where the output of distances based on correlations should be written

Optional Parameters

`-s sample mark` – Computes global correlations only relative to this mark in this sample. Can be used to parallelize this command, though for most purposes just parallelizing on marks through the -m option will be fast enough and require fewer CPUs.

`-m mark` – Computes global correlations only for this mark.

`-r resolution` – Should match the resolution of the converted signal data used in the convert command. The default value is 25 base pairs.

`-x extension` – The extension to add to files when computing the global correlation from what is given in `inputinfoinfile`. The default is "`.wig.gz`" consistent with what the Convert command adds.

Recommended Parallelization

For computing global correlations for large compendiums it is advisable to parallelize over each mark using the `-m mark` option.

GenerateTrainData

Description

This command takes a directory of converted data and global correlations and generates a set of training data instances. The files generated are then sufficient to train predictors for any sample.

Usage

```
java ChromImpute GenerateTrainData [-a mintotalensemble][-b numbags]
[-c chrom][-d seed][-dnamethyl infofile directory header][-f
numsamples][-i incrementnarrow incrementwide][-k maxknn][-
methylavgchrom|-methylavggenome][-n knnwindow][-r resolution] [-
tieglobal] [-w windownarrow windowwide] CONVERTEDDIR DISTANCEDIR
inputinfofile chrominfo TRAINDATADIR mark
```

Required Parameters

`CONVERTEDDIR` – the directory with the converted data in the format produced by the `Convert` command. Note the converted wig files generated by the `Convert` and are assumed by this procedure to have two lines of header information.

`DISTANCEDIR` – the directory with the distance ranking from the `ComputeGlobalDist` command

`inputinfofile` – is the same file described in the `Convert` command

`chrominfo` – is the same file described in the `Convert` command

`TRAINDATADIR` – the directory in which to output the training data

`mark` – the name of the target mark to predict and for which training data will be generated

Optional Parameters

`-a mintotalensemble` – Requires the total number of regression trees used when predicting the mark in a sample in which it has not been mapped to be at least this number. The `numbags` per sample in which the mark was mapped is set to the larger of the smallest integer that makes it possible or the value of the `-b numbags` parameter. The same value for `numbags` is used even if the mark has been mapped and is thus not being used for the prediction. Should match what was used in `GenerateTrainData` and `Train`. The default value of this is 0.

`-b numbags` – Specifies the number of different sample bags to generate. The number used could be larger if the `-a mintotalensemble` option is also specified. Default is 1.

`-c chrom` – Only generates the subset of training data corresponding to positions on this chromosome. Training data is printed with the `chrom` prefix. Attribute information is only printed when executing the first chromosome. This is useful for parallelizing the command.

`-d seed –` If specified can be used to provide a random seed for selecting the locations to include in training.

`-dnamethyl infofile directory header –` The `dnamethyl` and the following three parameters should be specified if training to predict DNA methylation data. `infofile` is a two column tab delimited file for the DNA methylation data giving the chromosomes to include and then the file name. `directory` is the directory in which the DNA methylation files are located. These files are tab delimited, where the first column specifies the chromosome position and the following columns the DNA methylation values on a 0 to 1 scale in different samples. Values less than 0 are treated as missing. The file `header` gives the column header information for the DNA methylation data. The first column header is a filler specifying position and all the following columns should specify the sample of the corresponding DNA methylation data.

`-f numsamples –` Specifies the number of locations to be used for training. Default is 100,000.

`-i incrementnarrow incrementwide –` Specifies the bin increment for including same sample features. `incrementnarrow` pertains to positions from the target position up until `windownarrow. incrementwide` pertains to positions after `windownarrow` up until `windowwide.` Default value for `incrementnarrow` is 1 bin and for `incrementwide` is 20 bins.

`-k maxknn` – The maximum number of nearest neighbor cross sample features to generate for a specific distance metric. Default is 10.

`-methylavgchrom|-methylavggenome` – These flags can specify that missing DNA methylation from within an experiment conducted should either be filled in with the chromosome average DNA methylation (`–methylavgchrom`) or the genome average (`–methylavggenome`). By default the genomewide average is used unless the `–c` flag is specified and then the chromosome average is used.

`-n knnwindow –` Specifies the window around the target position in terms of the number bins to use in each direction computing the local Euclidean distance between the same mark in different sample. Default value is 20 bins.

`-r resolution –` Should match the resolution of the converted signal data used in the convert command. The default value is 25 base pairs.

`-tieglobal –` if the flag is present then ties for the nearest sample based on local distance are broken by the global distance. If the flag is not present an arbitrary selection is made.

`-w windownarrow windowwide –` Same sample features of other mark from target position are included at increments specified by `incrementnarrow` up until `windownarrow` and then from `incrementwide+windownarrow` to `windowwide` based on increments specified by `incrementwide.` Default value for `windownarrow` is 20 bins and for `windowwide` is 400 bins.

Recommended Parallelization

For generating training data for large compendiums it is advisable to parallelize over each chromosome through the `–c chrom` option.

Train

Description

This command trains regression trees based on the feature information generated in GenerateTrainData. If the sample mark combination was available in the compendium, then the feature information pertaining to it is not used.

Usage

```
java ChromImpute Train [-a mintotalensemble][-b numbags][-
sampleonly][-dnamethyl header][-g bagrequest][-k maxknn][-m
minnumpoints][-markonly][-p selectedmarks][-q samplerequest]
TRAINDATADIR inputinfofile PREDICTORDIR sample mark
```

Required Parameters

TRAINDATADIR – The directory containing the training data as generated by GenerateTrainData. The command first tries to load a file generated without a chrom prefix, and if not found uses the union of all files with a chrom prefix with the -c command.

inputinfofile – is the same file described in the Convert command

PREDICTORDIR – The directory to which the predictors should be written

sample – The sample for which predictors should be trained to predict

mark – The mark for which predictors should be trained to predict

Optional Parameters

-a mintotalensemble – Requires the total number of regression trees used when predicting the mark in a sample in which it has not been mapped to be at least this number. The numbags per sample in which the mark was mapped is set to the larger of the smallest integer that makes it possible or the value of the -b numbags parameter. The same value for numbags is used even if the mark has been mapped and is thus not being used for the prediction. Should match what was used in GenerateTrainData and Train. The default value of this is 0.

-b numbags – Specifies the number of different sample bags to train on. The number used could be larger if the -a mintotalensemble option is also specified. Should match what is specified in GenerateTrainData. Default value is 1.

-sampleonly – If this flag is present only features based on other marks in the same sample are used.

-dnamethyl header – If the target is DNA methylation information then this flag should be present with the same header file as given to GenerateTrainData.

`-g bagrequest –` If this flag is present only predictors corresponding to this bag index are trained, where bags are indexed starting from 0. Useful for parallelizing training.

`-k maxknn –` If this option is present specifies the maximum number of nearest neighbors to use as part of the features to the regression tree. It should be equal or less than the value provided to `GenerateTrainData.`

`-m minnumpoints –` This parameter specifies the minimum number of data points that needs to be associated with a leaf node of the regression tree. Default value is 20.

`-markonly –` If this flag is present only features based on the target mark in other samples is used.

`-p selectedmarks –` If this option is present only features that can be computed based on the marks specified in `selectedmarks` are used for training even if additional are present. Marks are delimited by a comma.

`-q samplerequest –` If this flag is present only predictors corresponding to this requested sample index are trained, where samples are indexed starting from 0. Useful for parallelizing training.

Recommended Parallelization

If training predictors for multiple sample-mark target combinations this command enforces the parallelization over those combinations. Additional parallelization can be done through the `–g bagrequest and –q samplerequest` options.

Apply

Description

This command applies the predictors generated in the `Train` command to generate the imputed data.

Usage

```
java ChromImpute Apply [-a mintotalensemble][-b numbags][-c chrom][-
coeffv][-sampleonly][-dnamethyl infofile directory header][-i
incrementnarrow incrementwide][-k maxknn][-markonly] [-
methylavggenome|-methylavgchrom][-n knnwindow][-noprintbrowserheader]
[-o outputfile][-p selectedmarks][-printonefile][-r resolution][-t
outputfile_coeffv][-targz targzipfile][-tieglobal][-w windownarrow
windowwide] CONVERTEDDIR DISTANCEDIR PREDICTORDIR inputinfofile
chrominfo OUTPUTIMPUTEDIR sample mark
```

Required Parameters

`CONVERTEDDIR` – the directory containing the converted data in the format produced by the `Convert` command

`DISTANCEDIR` – the directory containing the distance based global correlations output of the `ComputeGlobalDist` command

`PREDICTORDIR` – the directory containing the regression tree predictors that will be applied to generate the imputed data

`inputinfofile` – is the same file described in the `Convert` command

`chrominfo` – is the same file described in the `Convert` command

`OUTPUTIMPUTEDIR` – The directory where the imputed files should be written

`sample` – The sample for which the imputation should be done

`mark` – The mark for which the imputation should be done

Optional Parameters

`-a mintotalensemble` – Requires the total number of regression trees used when predicting the mark in a sample in which it has not been mapped to be at least this number. The `numbags` per sample in which the mark was mapped is set to the larger of the smallest integer that makes it possible or the value of the `-b numbags` parameter. The same value for `numbags` is used even if the mark has been mapped and is thus not being used for the prediction. Should match what was used in `GenerateTrainData` and `Train`. The default value of this is 0.

`-b numbags` – The number of bags the classifiers was requested to be trained on. The number used could be larger if the `-a mintotalensemble` option was also specified. Should match what was used in `GenerateTrainData` and `Train`. The default value of this is 1.

-c chrom – If this flag is present then predictions are made for chromosome chrom

-coeffv – If this flag is present then the coefficient of variation of the different predictors is outputted.

-sampleonly – Same option as described in Train and should match value from Train.

-dnamethyl infofile directory header – Same options as described in GenerateTrainData and should match values from this command.

-i incrementnarrow incrementwide – Same option as described in GenerateTrainData and should match value

-k maxknn – Same option as described in Train and should match value from Train.

-markonly – Same option as described in Train and should match value from Train.

-methylavgchrom|-methylavggenome – These flags can specify that missing DNA methylation from within an experiment conducted should either be filled in with the chromosome average DNA methlation (-methylavgchrom) or the genome average (-methylavggenome). By default the geneomewide average unless the -c flag is specified and then the chromosome average is used.

-n knnwindow – Same option as described in GenerateTrainData and should match value.

-noprintbrowserheader – If this flag present then suppresses the printing of the browser header line which should not be present if converting the files later to BigWig format.

-o outputfile – The name of the outputfile to produce without the .gz extension and chromosome prefix. If not provided the default is impute_sample_mark.wig

-p selectedmarks – Same option as described in Train and should match value from Train.

-printonefile – if the flag is present prints all the chromosome in one file. Default is each chromosome is written to a separate file prefixed by the chromosome name followed by an underscore.

-r resolution – Same option as described in GenerateTrainData and should match value.

-tieglobal – if the flag is present then ties for the nearest sample based on local distance are broken by the global distance. If the flag is not present an arbitrary selection is made.

-t outputfile_coeffv – if the flag is present then the coefficient of variation output without the .gz extension and chromosome prefix. If not provided the default is impute_sample_mark_coeffv.wig

-targz targzfile – If this option is specified then the predictors are read from a .tar.gz file with the name targzfile

`-w windownarrow windowwide –` Same option as described in `GenerateTrainData` and should match value

Recommended Parallelization

If making predictions large for multiple sample-mark target combinations this command enforces the parallelization over those combinations. Additionally it is also recommended to parallelize over target chromosomes through the `–c chrom` option.

Note about Conversion to BigWig: If converting to BigWig using the program wigToBigWig, then the option '-clip' needs to be added since the last 25-bp bin is only partially contained in the chromosome.

Eval

Description

This command compares observed data generated by Convert to genome-wide imputed data predictions. It outputs (1) the fraction of the observed top percent1 locations in the imputed top percent1 locations, (2) the fraction of the imputed top percent1 in the observed top percent2, (3) the fraction of the observed top percent1 in the imputed top percent2, (4) the Pearson correlation between the observed and imputed data, (5) the area under the ROC for predicting the top percent1 imputed signal with the full range of observed signal, (6) the area under the ROC for predicting the top percent1 observed signal with the full range of imputed signal, (7) the Pearson correlation after clipping values at that clipping threshold.

Usage

```
java ChromImpute Eval [-c clipthresh][-f peakevalfile][-
noprintbrowserheader][-o outfile][-p percent1 percent2][-printonefile]
CONVERTEDDIR ConvertedFile IMPUTEDIR ImputeFile chrominfo
```

Required Parameters

`CONVERTEDDIR` – the directory containing the converted data

`ConvertedFile` – the name of the converted data files to compare to excluding the 'chr_' prefix

`IMPUTEDIR` – the directory containing the imputed data to compare to with each chromosome in a separate file

`ImputeFile` – the name of the imputed data files to compare to excluding the 'chr_' prefix

`chrominfo` – a chromosome info file as described with the `Convert` command containing the names of the chromosomes to evaluate

Optional Parameters

`-c clipthresh` – if present this specifies a non-default value for the clipthresh. The default value is 500. Value above this threshold will be set to this value before computing the correlation based on clipping, which can reduce the impact of outliers.

`-f peakevalfile` – if present the recovery of peaks specified in a three column bedformat is evaluated and the `CONVERTEDDIR` and `ConvertedFile` entries should still be provided but are ignored.

`-noprintbrowserheader` – flag should be present if was present when using the `Apply` command.

`-o outfile` – If present then the output is written to the file `outfile` instead of being printed to the terminal.

`-p percent1 percent2 –` Gives lower and upper percentages to use in evaluation. Default is percent1 is 1% and percent2 is 5%.

`-printonefile –` flag should be present if was present when using the `Apply` command.

ExportToChromHMM

Description

This command converts multiple signal files in the form that ChromImpute generates in the `Apply` step to a form that can be used by ChromHMM. Note that the conversion of ChromImpute's DNA methylation files is not supported. The converted form for ChromHMM can either be binarized data that can directly be used ChromHMM's `LearnModel` command or as signal data that can be provided to ChromHMM's `BinarizeSignal` command, as discussed in the use of the '`-g signalthresh`' option below.

Usage

```
java ChromImpute ExportToChromHMM [-b chromhmmbinsize][-g
signalthresh][-partial][-r resolution][-usenames] CHROMIMPUTEDIR
inputinfofile chrominfofile CHROMHMMDIR
```

Required Parameters

`CHROMIMPUTEDIR` – The directory where the signal files from ChromImpute that should be converted to a form for use in ChromHMM are present.

`inputinfofile` – This specifies a file of the same format as `inputinfofile` in the `Convert` command, though the third column containing filenames is optional and ignored unless the `-usenames option` is provided in which case it is required. A set of converted files is made for each cell type present in the first column of `inputinfofile` with each file containing all the marks present anywhere in the second column of `inputinfofile`. As explained in the `–usenames` option description, by default files for each cell type and mark combination are assumed to present in the `CHROMIMPUTEDIR` under the default naming of the `Apply` command, but if `–usenames` option is provided then different file names can be specified in the third column.

`chrominfofile` – is the same file described in the `Convert` command. Output files will be generated for each chromosome specified in this file.

`CHROMHMMDIR` – The directory where the converted files for ChromHMM should be written.

Optional Parameters

`-b chromhmmbinsize` – This specifies the bin size that will be used with ChromHMM. The default is ChromHMM's default bin size which is 200bp bins. The ChromHMM bin size must be evenly divisible by the ChromImpute resolution.

`-g signalthresh` – If this flag is present, then binarized files that can be used directly with ChromHMM's `LearnModel` command are produced. For a given mark and a given bin, the average of all the ChromImpute values for that mark in that bin are averaged. Values that are equal to or greater than `signalthresh` receive a binarized value of '1' while signal values less than the threshold. If this flag is not present, then signal values that can be used with ChromHMM's BinarizeSignal command are generated. The signal values are the bin averaged signal values. Using this flag can make sense if a uniform threshold is meaningful across different marks, for instance if the signal values represent fold enrichments or –log p-values. If this flag is not used, by default ChromHMM's BinarizeSignal command assumes signal values represent counts.

`-partial` – If this flag is present, then a line for a partial bin at the end that does not span a full `chromhmmbinsize` is still included otherwise such a line is omitted. Note that if this flag is included, to prevent the ChromHMM `LearnModel` command from producing segmentation intervals that goes past the end of the chromosome, the option '`-l chromosomelengthfile`' would need to be provided to ChromHMM.

`-r resolution` – This specifies the ChromImpute resolution and is the same option as described in `GenerateTrainData` and `Apply` and should match value. Default value is 25bp.

`-usenames` – If this flag is present then the file names in the third column of `inputinfofile` are used, otherwise it is assumed files are named in the default format of the ChromImpute `Apply` command '`CHROMOSOME_impute_CELL_MARK.wig.gz`'. Note that if this option is specified an entry must be present in `inputinfofile` for every possible cell type and mark combination among cell types and marks appearing in the file at least once. If the option is not specified, a file for every cell and mark combination under the default naming needs to exist, but does not need to be explicitly given in `inputinfofile`.