Coupled Analysis of Polymerase and Expression (CAPE) Users Guide

Version 1.0

Raymond Auerbach, Arif Harmanci, Joel Rozowsky, and Mark Gerstein
Gerstein Laboratory
Yale University

Note: The latest version of the Users Guide can always be found at http://cape.gersteinlab.org.

Table of Contents

Introduction

CAPE Is designed with two primary goals in mind. The first is to associate transcription factor binding site data from next-generation sequencing experiments such as ChIP-Seq with a set of genomic features such as transcription start sites, transcription end sites, etc. The second aim is to classify transcripts based on levels of transcription factor binding at the promoter vs. the expression level of the transcript. This tool is designed to work with many of the formats employed by large consortia such as NHGRI’s ENCODE and modENCODE consortia.

Program Requirements

CAPE is written in Java and requires Java version 1.6 or higher. The Java runtime environment (JRE) must be installed. If the JRE is not installed (i.e. typing “java –version” at the command line does not produce a version number or the version number is < 1.6), an updated version of the Java Runtime Environment can be downloaded from http://www.oracle.com/technetwork/java/index.html. Please see that website for instructions.

Obtaining the program

The CAPE programs are distributed as self-contained, executable java archives (a .jar file) and can be downloaded from http://cape.gersteinlab.org. If you wish to download the source code and associated first-party libraries for this tool, they can also be found at the same address. CAPE is distributed as-is as open-source software.

CAPE-analyze

Generating Transcription and Binding Reports using CAPE-analyze

Once the JRE is installed and the jar file is downloaded, the program can be run from a terminal window. From the command prompt, type “java –jar CAPE-analyze.jar <arguments>”. See the section “Command line arguments below for a detailed description of possible arguments. Long-form arguments are given in the format “--longFormOptionName=value” and short-form arguments are given in the format “-shortFormOptionName value”. For larger data sets, we recommend increasing the maximum amount of memory allocated to the Java heap to improve system performance. Depending upon your hardware configuration and the number of transcripts to be analyzed, the java heap size may be increased using Java’s “-Xmx” option (e.g. “-Xmx512M” will set the heap size to use 512 MB of RAM, “-Xmx1G” will use 1 GB of RAM, etc.). This option should appear between “java” and “-jar” in the example command given above.

CAPE-analyze Command-Line Arguments

Program Mode Arguments

One of the following arguments may be specified to select the mode of operation. If no argument is specified, CAPE will run in transcript mode by default.

Long-Form	Short-Form	Description
--binding	-b	Run in binding mode
--transcript	-t	Run in transcript mode (default)

If both arguments are specified, CAPE will exit with an error.

Binding Mode Arguments

The following arguments are required when running in binding mode:

Long-Form	Short-Form	Description
--signalfile	-s	Signal file in bigWig or bigBed format. Extension must be .bw, .bigWig, .bb, or .bigBed
--transcriptfile	-tf	File with transcript information. Must be GTF format
--peakfile	-p	Peak file. Extension must be .bed or .narrowPeak
--outputfile	-o	Filename to use for output

Additionally, the following optional parameters may be specified:

Long-Form	Short-Form	Description
--upstreampad	-up	The upstream overlap window in bp to use for feature association. Default=1000 bp.
--downstreampad	-dp	The downstream overlap window in bp to use for feature association. Default=1000 bp.

Transcript mode arguments

The following arguments are required when running in transcript mode:

Long-Form	Short-Form	Description
--signalfile	-s	Signal file in bigWig or bigBed format. Extension must be .bw, .bigWig, .bb, or .bigBed

--transcriptfile	-tf	File with transcript information. Must be in GTF format
--outputfile	-o	Filename to use for output

Additionally, the following optional parameters may be specified:

Long-Form	Short-Form	Description
--aggpad	-ap	The initial window size in bp on each side of a feature to use for the aggregation step. Default=1000 bp (note: this results in +/- 1000 bp from start)
--aggoverride	-ao	In transcript mode, the pad value to use on each side of a feature for aggregation. Value must be >= 50. Skips window size detection if set.
--rseqtoolsfile	-r	File created by the rSeqTools program mrfQuantifier containing expression values for all transcripts (the transcript ID appears in column 1 and the RPKM appears in column 2)
--cufflinksfile	-c	File created by cufflinks containing expression values for all transcripts (the transcript id appears in column 2 and the FPKM appears in column 10)
--GFFkey	-k	The feature type to use from the third column of a valid GFF or GTF file. Defaults to "transcript". Change if you are analyzing genes, etc.
--percentileoverride	-po	A comma-separated list of percentile cutoffs for low- and high-value binding and expression. Default=25,75
--expressionoverride	-eso	A comma separated list of expression value cutoffs for low- and high-value expression. Given in real units (RPKM, FPKM, etc). Masks the --percentileoverride option for expression.
--bindingoverride	-bso	A comma separated list of expression value cutoffs for low- and high-value expression. Given in real units (RPKM, FPKM, etc). Masks the -percentileoverride option for binding.

Description of Algorithm - Binding Mode

Given a transcript file, a signal file, and a peaks file, binding mode will identify the position within each peak where the signal is the highest. Using this summit position to represent the peak region, overlap analysis is then run against the transcript list and the closest TSS and TTS is reported as well as the distance to each. Distances are reported ina strand-specific manner where negative values correspond to the summit position being upstream of a feature.

Conversely, a positive distance value indicates that the summit occurs downstream of a feature. Each peak will be assigned an association of “TSS”, “TTS”, “Neither” or “Ambiguous” based on the distance to each feature. The values of --upstreampad and --downstream pad are used to assign a peak to a category. For example, the default value of 1000 bp means that a peak that falls within +/- 1000 bp of a TSS will be assigned a TSS association. If a peak falls halfway between a TSS and a TTS and both distances fall within the cutoff, a value of “ambiguous” is reported. Peaks falling outside the pad range will be reported as “Neither.” Note that for peak files in narrowPeak format, the summit position given in the file for each peak is used as the summit position. For files in bed format, the maximum signal in the region is determined and if this region spans multiple base pairs, the midpoint of the range is reported as the summit value.

In the event that there is more than one maximum value, the peak is reported as “multimodal” and an association is not calculated. Please note that binding mode is still under active development at this time and that the crux of CAPE-analyze is transcript mode.

Description of Algorithm - Transcript Mode

Given a transcript file, a signal file, and a file with expression values, the tool begins by aggregating the TF binding signal around TSSs given the window size of +/- the --aggPad argument (default: 1000 bp). The results from this aggregation are used to determine an ideal size for the promoter region as follows: for the aggregation values, the global maximum and minimum positions are determined. From the position of the global maximum, the closest positions where the value crosses below the value of (global minimum + ((global maximum - global minimum) * .10) are determined in both the upstream and downstream directions and the largest values used as the pad. For example, if the above criteria is met at -500 and +300 bp from the global maximum, then an “ideal” pad size of +/- 500 bp from the TSS is used. Using the ideal pad size to represent the promoter region, the average signal level in the promoter and the gene body are determined for each transcript and the ratio between these values is reported. Note that currently, there is no size restriction on transcripts. If a transcript is smaller than the ideal pad for a promoter, no signal over the transcript body is calculated and the ratio will be “N/A”. For gene expression, values are read directly from an expression file (typically in RPKM or FPKM) or can be read from the annotation file. Please see the description of input files below for more details.

Once both binding values and expression values are determined for each transcript, low- and high-value cutoffs are determined. This process occurs separately for binding and expression but the steps are the same. We will discuss in terms of binding in this example. Using only the binding signal values that are non-zero, we determine the values that correspond to the percentile cutoffs specified by --percentileoverride (default= 25th and 75th percentile). All transcripts with a binding value below the lower cutoff value will be deemed as “low binding,” any transcript with a binding value above the upper cutoff will be deemed “high binding.” Transcripts with binding values in the intermediate range are deemed “normal.” Alternatively, we expect there will be some cases where a researcher will want to use actual data values instead of percentiles to set the lower and upper cutoffs. These values can be set with the --

expressionoverride and --bindingoverride arguments (e.g. --bindingoverride=20,80 will set the low and high binding cutoffs to be below the 20th percentile and above the 80th percentile, respectively. These arguments will override the percentile cutoffs for their respective data types. After all transcripts are categorized, a list of transcripts that fall into each category is written to the file named in the --outputfile argument.

Input File Formats

CAPE-analyze supports a variety of different input formats for peak files, signal files, and expression files.

Peak Files

Peak files are used in binding mode and can be provided in three or four column, tab-delimited bed format. The first column corresponds to the chromosome, the second column gives the start position, the third column gives the end position, and the optional fourth column is a single character denoting the strand (“+” or “-”). If strand is omitted, all features will be considered fo be on the forward strand. Please note that bed files must end in the extension “.bed”. For more information about bed format, please see http://genome.ucsc.edu/FAQ/FAQformat#format1. For example:

```
chr1    500     1000    +
chr2    100     300     -
```

Alternatively, peak files can also be provided in ENCODE narrowPeak format to allow for direct download from the ENCODE data repository at UCSC. For a description of the ENCODE narrowPeak format, please see http://genome.ucsc.edu/FAQ/FAQformat#format12. ENCODE narrowPeak files must end in the extension “.narrowPeak”.

Signal Files

Signal files must be provided in bigWig or bigBed formats. These files are indexed, allow for random access, and reduce the storage requirements for large signal files such as those produced from whole-genome analyses in human cells. Other formats such as bedGraph and wiggle can be easily converted to bigBed or bigWig formats using Jim Kent’s toolkit at UCSC (binaries available at http://hgdownload.cse.ucsc.edu/admin/exe/). Please see the documentation in the source distribution of Jim Kent’s utilities for more information. The relevant programs are bedGraphToBigWig, wigToBigWig, and bedToBigBed. Alternatively, Galaxy may also be used to convert other signal file formats to bigWig and bigBed (http://galaxy.psu.edu/). For more information about the bigWig and bigBed formats, please see http://genome.ucsc.edu/FAQ/FAQformat#format1.5 and http://genome.ucsc.edu/FAQ/FAQformat#format6.1.

Transcript Files

Transcript files are accepted in either GTF (preferred) or GFF3 files. A full description of these files formats can be found at http://genome.ucsc.edu/FAQ/FAQformat#format4 (GTF) and http://genome.ucsc.edu/FAQ/FAQformat#format3 (GFF3). An example in GTF format is given below:

```
II	modENCODE_TX	gene	8651057	8658766	.	+	.	RPKM "109.609215"; gene_id "pyr-1"
II	modENCODE_TX	gene	5399522	5405988	.	+	.	RPKM "94.070177"; gene_id "mog-5"
II	modENCODE_TX	gene	13670567	13694711	.	-	RPKM "61.110324"; gene_id "Y48E1A.1"
```

To illustrate some of the features of CAPE-analyze, let's use the above snippet as an example. This GTF file targets genes instead of transcripts. The default behavior of CAPE-analyze is to look for transcripts, but this can be changed by setting the "--GFFkey=gene" parameter at the command line. This will tell CAPE-analyze to look for genes and to key off the gene_id value given in the ninth column.

In both GFF3 and GTF, the ninth column is a "catch-all" column where the id and, occasionally, gene expression data will appear. Running in default mode, CAPE-analyze will look for records with "transcript" in the third column and "transcript_id" in the ninth column. If the "--GFFkey=gene" option is set, CAPE-analyze will instead analyze records with "gene" in the third column and "gene_id" in the ninth column.

Also shown above is the expression value given in the ninth column ("RPKM"). CAPE-analyze will use this field if no other expression files are provided. CAPE-analyze is designed to look for RPKM or FPKM tags in the ninth column. Multiple instances of these tags can exist (e.g. RPKM1 and RPKM2 in the case of multiple replicates being described in the same file). In these cases, all values will be averaged to calculate the overall expression value to be used for the transcript.

Expression Files

CAPE-analyze accepts expression values in any of three possible formats:

- rSeqTools - A tab-delimited file where the transcript ID occurs in column 1 and the expression value occurs in column 2.

- cufflinks - A tab-delimited file where the transcript ID occurs in column 2 and the expression value occurs in column 10. Note that users of Cufflinks 2.0+ should use the GTF option below if a final GTF file is produced.

- GTF - a GTF or GFF file with one or more of the following tags present in the final field for each transcript: RPKM or FPKM. In cases where tags are present for multiple replicates (ex: RPKM1 and RPKM2), the average expression value will be used.

For rSeqTools and cufflinks files, the file should be specified on the command-line using the "--expressionfile" option. For GTF format files, one must only specify the GTF file using the "--signalfile" option (the "--expressionfile" option should be omitted).

NOTE: Using a cufflinks file or an rSeqTools file will override any expression values given in the GTF file. In transcript mode, the program will exit with an error if expression values are omitted completely. Binding mode will still function, but expression values will be omitted from the final reports.

CAPE-analyze Output File

CAPE-analyze will produce a tab-delimited text file that can be easily viewed in Excel or a text editor of your choice. A snippet of a CAPE-analyze report for *C. elegans* follows:

```
#Transcript File: /emb_study/worm/total-RNA_20_degree_celsius_N2_Early_Embryos_RNA-seq.gtf
#Signal File: /emb_study/worm/2435_Snyder_N2_POLII_eemb_combined.bw
#Mode: transcript
#Promoter Binding Low Percentile Threshold: 25.0 (Binding <= 12.771
#Promoter Binding High Percentile Threshold: 75.0 (Binding >= 27.305
#Expression Low Percentile Threshold: 25.0 (Expression <= 43.015
#Expression High Percentile Threshold: 75.0 (Expression >= 80.253

#A. Transcripts with low promoter binding and high expression: 32
#B. Transcripts with high promoter binding and low expression: 45
#C. Transcripts with no promoter binding and no expression: 0
#D. Transcripts with no promoter binding: 6
#E. Transcripts with no expression: 0
#F. "Normal" transcripts: 937

#Category       Transcript ID       Chromosome      Start     End       Strand    PromoterSignal    BodySignal
        Ratio (Stalling Index)      Expression Value
A       ZK858.6  I          9138260 9145625  -      8.889     24.151    0.368     88.639
A       rpt-5    I          5741868 5743576  -      12.163    22.264    0.546     106.276
A       sup-17   I          8792767 8797287  +      12.737    19.403    0.656     90.673
A       ngp-1    I          8394722 8398622  +      12.222    17.766    0.688     86.28
…
…
…
```

The top of the report shows the files used for analysis as well as the parameters and cutoffs used by CAPE-analyze to categorize transcripts (or in this particular analysis, genes). This is followed by a breakdown of the six classifications used by CAPE-analyze as well as summary counts. The table then shows the values calculated for each gene or transcript as well as identifying characteristics such as chromosome, position, and strand. Raw values for promoter and body signals are also provided as well as the expression value from the expression file. The stalling index is the ratio of signal in the promoter vs the gene body. For genes that are smaller than the average window size being used for aggregation, the bodySignal value will be 0 and the Stalling Index will be "NA".

CAPE-compare

Comparing Results Using CAPE-compare

Often, researchers will want to compare transcripts from different samples or organisms to identify changes between ortholog groups or compare multiple samples from the same organism (e.g. healthy vs diseased cells, replicates, etc). The CAPE-compare tool takes two or more reports generated using the transcript mode of CAPE-analyze as well as an optional tab-delimited ortholog file and generates a tabular file combining the results into a single file. This file can easily be viewed in Excel or any text editor. Additionally, if two, three, or four reports are specified as input, an HTML file summarizing the overlaps within each transcription/binding category is also produced in addition to a ready-to-run R script that will produce Venn diagrams comparing samples across each category and save them as TIFF files. Please note that the free "VennDiagram" R package must be installed within R to use the script produced by CAPE-compare (see Acknowledgement of External Libraries section). We envision CAPE-compare to be useful in two different scenarios: comparing between organisms and comparing within an organism.

Comparing Results Between Organisms

For comparing reports generated between different organisms or annotation sources (e.g. worm, fly, and human), an ortholog file must also be provided to CAPE-compare and specified on the command line with the “--orthologs” option. The ortholog file is a tab-delimited text file with the number of columns equal to the number of organisms being compared where each column contains the gene or transcript IDs used in the report Each row represents an ortholog set. For example:

```
humanOrtholog1      flyOrtholog1    wormOrtholog1
humanOrtholog2      flyOrtholog2    wormOrtholog2
humanOrtholog3      flyOrtholog3    wormOrtholog3
...
...
...
etc...
```

Only transcripts for which a complete list of orthologs exists should be provided (e.g. if three samples are being used, a value must exist in all three columns in each row).

When comparing different organisms, the following should be noted:

1. The IDs in the ortholog file must exactly match those in the report file.
2. The order of the filenames given to the --input option must match the order of the columns in the ortholog file (e.g. for the above example, the --input option must be “--input=humanReport.txt,flyReport.txt,wormReport.txt”).
3. Only transcripts specified in the orthologs file will be written to the output files and considered in the output files.
4. When comparing two, three, or four CAPE-analyze reports, the HTML files and R script for Venn diagrams will be produced. In these files, only transcripts for which data exists for all samples being compared will be used. In the raw data output from CAPE-analyze, all transcripts will be listed with those missing data being classified as “NA”.
5. Venn diagrams will only be produced for categories containing at least one data point.

Comparing Reports from the Same Organism

Researchers may also wish to compare CAPE reports from the same organism generated under different experimental conditions, as different experimental replicates, etc. In this case, an ortholog file is not necessary and CAPE-analyze will assume that the reports share the same set of transcript IDs (for cases where transcript IDs may differ, such as when using different annotation versions, use the “Comparing Reports Between Organisms” workflow described above). Alternatively, an ortholog file can also be specified when a researcher wishes to limit the subset of transcripts being examined. In this case, the “Comparing Reports Between Organisms” workflow should be used. A tab-delimited text file or raw data generated from combining the reports will be generated. An HTML summary file and an R script to generate associated Venn diagrams will also be generated when comparing two, three, or four CAPE-analyze reports. These files are as described above.

Running CAPE-compare

CAPE-compare is run in the same manner as CAPE-analyze. From the command-line:

```
java -jar CAPE-compare.jar <argument list>
```

CAPE-compare Command Line Arguments

The following arguments are required for CAPE-compare:

Long-Form	Short-Form	Description
--input	-i	Comma-delimited list of CAPE reports to compare
--prefix	-p	the prefix to use for output files

Additionally, the following arguments may also be specified:

Long-Form	Short-Form	Description
--labels	-l	comma-delimited list of labels to use in the reports. Order must correspond to the order given for the --input option. File names will be used as labels if not specified
--orthologs	-o	the ortholog file to use

CAPE-compare File Formats

Input Files

Input files are in the output format produced by CAPE-analyze and should be specified as a comma-delimited list at the command line using hte input option. For example, "--input=file1.txt,file2.txt".

Labels

We recommend giving each file a descriptive label, as it will make the reports easier to read. These labels should be provided in a comma-delimited list with the "--labels" parameter. For example: "--labels=worm,fly". Please note that the order of the labels must coincide with the order of the input files passed to the --input argument.

Output Files

All output files will contain the prefix specified by the “--prefix” argument. For example, for “--prefix=testOutput” with three input files, CAPE will produce the output files testOutput.txt, testOutput.html, and testOutput.r.

Ortholog File

The ortholog file is described in the section “Comparing Results Between Organisms” above and is optional.

Output FIle Examples

Raw Data Output

Regardless of the number of CAPE-analyze input reports to compare, the raw data text file will be produced. This file is tab-delimited and an example follows:

```
#A. Transcripts with low promoter binding and high expression
#B. Transcripts with high promoter binding and low expression
#C. Transcripts with no promoter binding and no expression
#D. Transcripts with no promoter binding
#E. Transcripts with no expression
#F. "Normal" transcripts

#Worm Feature   Fly Feature        Human Feature     Worm State        Fly StateHuman State
dhc-1     FBgn0261797     ENSG00000197102          F          F          NA
prp-8     FBgn0033688     ENSG00000174231          F          F          NA
ama-1     FBgn0003277     ENSG00000181222          F          F          NA
sma-1     FBgn0004167     ENSG00000137877          F          F          F
…
…
…
```

In the above example, some genes were not able to be analyzed by CAPE-analyze (in this case, the quality data for some transcripts were subpar and as a result, these genes were omitted from the human GTF annotation file). Genes and transcripts for which data are missing will still be shown in the raw data text file but will be given the category “NA”. However, the entire record for the corresponding ortholog set will be omitted from both the html files and the Venn diagram calculations, should these be generated.

HTML Summary File

In cases where two, three, or four CAPE-analyze reports are being compared, a summary file in HTML format will be produced. This file will contain a breakdown of transcript counts by category and sample combination as well as a list of the orthologous features comprising each category. The HTML file links the numbers in the summary tables to the corresponding feature list. A sample can be viewed as part of the Use Case document available at http://cape.gersteinlab.org.

R Script for Venn Diagrams

In cases where two, three, or four CAPE-analyze reports are being compared, a ready-to-run R script will be produced. When run, this script will create Venn diagrams for each category, showing the distribution of orthologs across sample combinations (e.g. Worm Only, Worm and Fly Only, etc). Please note that the R script requires that the free VennDiagram

R package be installed (http://cran.r-project.org/web/packages/VennDiagram/index.html). Sample Venn diagrams can be viewed as part of the Use Case document available at http://cape.gersteinlab.org.

Acknowledgement of External Libraries

CAPE-analyze makes use of the following external, publicly-available libraries: Google Guava, Apache Commons Math, Apache Commons CLI, and the Broad Institute's IGV BigFile. CAPE-compare utilizes the Apache Commons CLI and charts4j external libraries. These libraries are packaged as part of the respective executable jar file in their original, unaltered forms.

For more information on each library, please see the following links:
Broad IGV BigFile: http://http://code.google.com/p/bigwig/
Apache Commons Math: http://commons.apache.org/math/
Apache Commons CLI: http://commons.apache.org/cli/
Google Guava: http://code.google.com/p/guava-libraries/

Additionally, the following free R library is required to run the Venn diagram script file generated by CAPE-compare:
VennDiagram: http://cran.r-project.org/web/packages/VennDiagram/index.htm

Frequently Asked Questions:

Q. In CAPE-analyze, I don't know what percentile I want to use but I do know which specific cutoff values I want to use for my expression and binding cutoffs. Can I specify raw values?
A. You bet! We designed CAPE-analyze to allow for either percentiles or raw values to be used. If you want to define the low RPKM cutoff at <= 1 and the high RPKM cutoff at >= 5, for example, the following argument should be used at the command line: "--expressionoverride=1,5". The "--bindingoverride" option can be used to set binding cutoffs in the same manner. Please note that the cutoffs should be given in ascending order. The CAPE-analyze report will reflect your chosen cutoffs in the file header.

Q. CAPE-analyze is giving me very strange results for the ideal window size determined by aggregation. What should I do?
A. Although we have tried to tune our algorithm to choose an ideal window size, sometimes a larger window than expected can be selected if ChIP-Seq signal is particularly broad or if the annotation is not specific (e.g. in worm, for example, there are splice leaders that are not always removed from an annotation. In these cases the actual RNAPII signal may be offset from the annotation start site). For these cases, we suggest setting the "--aggoverride" option to a reasonable setting for your organism. This will skip the initial aggregation step and use your value to define promoter size. For example, "--aggoverride=500" will set the promoter region as +/- 500 bp from the start sites in the annotation file.
Q. Are there any sample runs available?
A. Yes. Sample runs with links to data, logs, and program output are included in the Use Case document available at http://cape.gersteinlab.org. Use cases are given for both transcript and binding modes.