**Overview**

CAFE is a high-performance pipeline for transcriptome assembly. CAFE predicts the directionality of unstranded RNA-seq reads using a *k*MC (*k*-order Markov Chain) model and generates RPDs (reads with a predicted direction), increasing assembly sensitivity and specificity. Furthermore, full-length transcripts can be obtained by updating exon-junctions with exon-junction reads and calculating maximum entropy scores from putative splicing signals. To improve annotations at transcript boundaries, transcription start sites (TSSs), determined from CAGE-seq, and cleavage and polyadenylation sites (CPSs), from 3P-seq, were incorporated into relevant transcripts. CAFE should not only help to build comprehensive, precise transcriptome maps from complex genomes but also expand the universe of non-coding genomes.

**Requirement**CAFE was developed in a Linux environment (CentOS 6.8). Cufflinks, SAMtools, Perl, and Python are required to run this program.

**Installing CAFE**

CAFE can be downloaded as a source package from Supplemental Materials and from our website (<http://big.hanyang.ac.kr/CAFE)>.

To install CAFE from the source package, follow these steps:

1. Decompress the downloaded CAFE source archive in your local directory, e.g.:  
   cd ~/programs/  
   tar xvzf ~/programs/CAFE.Version.tar.gz  
     
   A directory named CAFE.Version (where Version is a numeric version of the program) will be created in the current directory.
2. Download a reference genome from the UCSC Genome Browser (<http://genome.ucsc.edu/index.html)> and move it to the corresponding location in the program, e.g.:  
   mv hg19.fa ~/programs/CAFE.Version/data/fasta/human/hg19/. or  
   mv mm9.fa ~/programs/ CAFE.Version/data/fasta/mouse/mm9/.  
     
   The directions ‘~/programs/CAFE.Version/data/fasta/human/hg19 or mm9’ will be automatically generated as you decompress CAFE. The reference genome should be the same one that was used in RNA-seq mapping.

**Running CAFE**  
Run CAFE from the command line like this:

python cafe.py -g <species> -v <assembly> -c <chromosome> -t <type> -u <unstranded.bam> -s <stranded.bam> -o <outputdir>

The main input of the program is a BAM file with mapped RNA-seq reads, sorted by their genomic location.

The following optional parameters can be specified when running CAFE:

-h/--help Prints help message and exits

-g <species> Species used for assembly (either ‘*human*’ or ‘*mouse*’)

-v <assembly> Assembly version of reference genome (e.g.: ‘*hg19*’ or ‘*mm9*’)

-c <chromosome> Sets the chromosome for assembly (e.g.: *‘all*’ or ‘*chr1*’)

-t <type> Type of input BAM files. If you want to perform co-assembly with unstranded and stranded RNA-seq reads, specify ‘*-t npsp*’. (*npsp*: co-assembly of unstranded and stranded reads,  
*np*: assembly of unstranded reads alone,  
*sp*: assembly of stranded reads alone)

-u <unstranded.bam> Input BAM file with unstranded RNA-seq reads for assembly (not compatible with ‘*-t sp*’)

-s <stranded.bam> Input BAM file with stranded RNA-seq reads for assembly

(not compatible with ‘*-t np’*)

-o <outputdir> Sets the output directory where CAFE will write all result files

More detailed parameters can be modified by changing the *config* file. However, this is not recommended since these parameters were already optimized.

~/programs/ CAFE.Version/config.py

chrs Sets the chromosome flags in reference genome

intronmin Minimum length of intron to be considered in exon-junction update

intronmax Maximum length of intron to be considered in exon-junction update

sampling\_number The number of samplings used to construct the *k*-ordered Markov chain model

rpds\_type A model to be used when predicting the direction of unstranded RNA-seq reads  
(hmm: *k*-order Markov Chain, dis: majority voting method)

rpds\_order The *k* value used to predict the directionality of unstranded RNA-seq reads in the *k*-order Markov Chain (1~5)

exj\_type Method to update exon-junctions  
 (read: use exon-junction reads, signal: use maximum entropy scores from putative splicing signals, all: use both to update exon-junctions)

exj\_cutoff A cutoff value for maximum entropy scores in exon-junction update

tss\_type Criteria for choosing CAGE-seq tag(s)

(all: assign all, major: assign the most abundant, near: assign

the proximal CAGE-seq tag(s) to a corresponding transcript to update the 5’ end)

fpseqInterval Length of interval upstream from a 5’ end to be considered when updating the TSS using CAGE-seq tags

s

fpseqThreshold A threshold for the fraction of CAGE-seq tags in the sum of all tags assigned to a transcript; those falling below this threshold will be ignored in the subsequent TSS updating process

cps\_type Criteria for choosing 3P-seq tag(s)

(all: assign all, major: assign the most abundant, near: assign

the proximal 3P-seq tag(s) to a corresponding transcript to update the 3’ end)

tpseqInterval Length of interval downstream from a 3’ end to be considered when updating the CPSs using 3P-seq tags

tpseqThreshold A threshold for the fraction of 3P-seq tags in the sum of all tags assigned to a transcript; those falling below this threshold will be ignored in the subsequent CPS updating process

**Input files**

CAFE takes input(s) in a binary BAM format that is sorted by genomic coordination. BAM file(s) contain spliced read alignments and can be directly produced by spliced alignment programs. When the input BAM file contains unstranded RNA-seq reads, CAFE converts them to RPDs (reads with a predicted direction) first before it proceeds to an assembly step. In the case of stranded RNA-seq reads, CAFE enters them into the assembly step without RPD conversion.

**Output files**

Main output files

1. A RPD (read with a predicted direction) BAM file generated by predicting the direction of unstranded RNA-seq reads.

The BAM file is stored in:

~/programs/CAFE.Version/<outputdir>/rpds/

1. Transfrags assembled using Cufflinks with RPDs and stranded RNA-seq reads are reported in a GTF format.

The GTF file is stored in:  
~/programs/CAFE.Version/<outputdir>/gtf/rpds or stranded

1. A merged GTF file from assembled transfrags with RPDs and stranded RNA-seq reads.

The GTF file is stored in:  
~/programs/CAFE.Version/<outputdir>/gtf/combine/

1. Exon-junction, TSS, and CPS updated transcripts from the merged GTF files.  
   The exon-junction updated GTF file is stored in:  
   ~/programs/CAFE.Version/<outputdir>/gtf/exj/

The TSS updated GTF file is stored in:

~/programs/CAFE.Version/<outputdir>/gtf/tss/

The CPS updated GTF file is stored in:

~/programs/CAFE.Version/<outputdir>/gtf/cps/

1. The final GTF file with updated transcripts of CAFE after all processes.

The GTF file is stored in:

~/programs/CAFE.Version/<outputdir>/filter/

Please read the CAFE manuscript for detailed processes and/or methods.

**Example**

The CAFE source package provides exercise files to follow the CAFE pipeline.~/programs/CAFE.Version/data/example/The ‘example’ directory contains BAM files with unstranded (GEO accession numbers: GSM591659 and GSM591682) and stranded (GEO accession numbers: GSM546921, GSM546927, GSM591670, and GSM591671) RNA-seq reads sequenced from HeLa-S3 cells.

Run CAFE from the command line as below:

python cafe.py -g human -v hg19 -c chr22 -t npsp

-u ./data/example/unstranded/hela\_chr22.bam

-s ./data/example/stranded/hela\_chr22.bam -o ./results/

CAFE's progress as it runs will be printed at each step as standard output. All result files produced by CAFE will be stored in the ‘results’ directory.