TEiso: a pipeline for identification of transcripts isoforms in the context of transposable elements

This tutorial will describe how the workflow TEiso analyse RNA_seq to find the relation between TSS of each trancscripts (isoforms) and the closest transposable elements (TEs)

The workflow TEiso allows to do :

If you have any trouble to import your raw data into galaxy, or for your reads quality control, please go to the page NGS: reads quality control from the URGI dev team.

If you want to try it yourself, you can use simulated data from URGI download :

- 1.SRR070420_1.fastq.bz2
- 2.SRR070420_2.fastq.bz2
- 3.sfru_corn_OGS2.0.gff3
- 4-sfru.mais.corrected.3.1.fa
- 5-TEs.sfru.mais.corrected.3.1_AllCSS.gff3

Note : the data are already trimmed.

For more details on data format, see the slideshow on the format here

1) Mapping

TopHat is a program that aligns RNA-Seq reads to a genome in order to identify exon-exon splice junctions. To make mapping faster, The reference genome must first be "indexed" by bowtie-build. For more information see bowtie-bio.sourceforge.net manual and ccb.jhu.edu tophat manual.

Input format

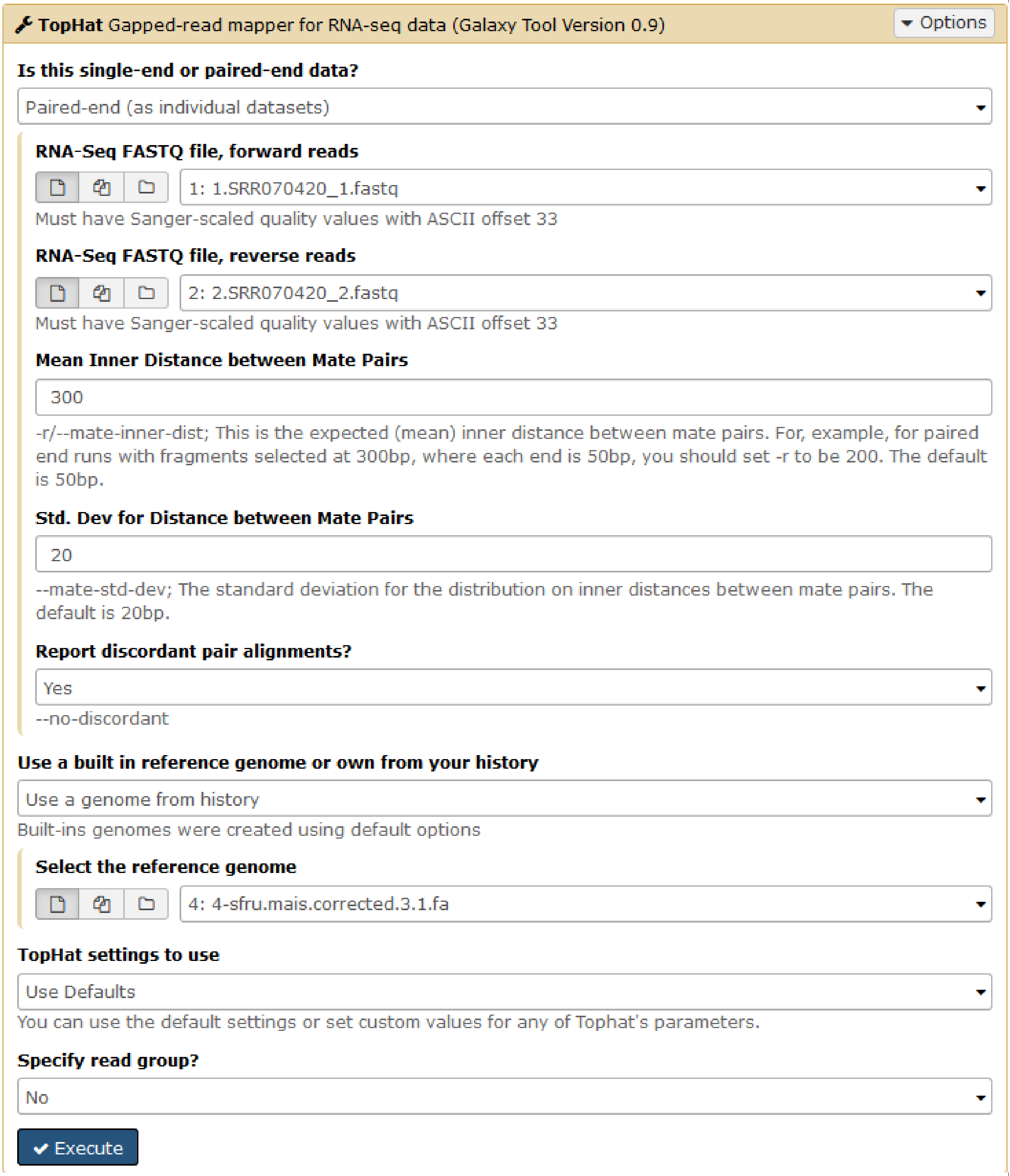
Tophat takes reads files in Sanger FASTQ format and a FASTA file with the sequence(s) of genome. Please note that you can access to your reference genome when you select « **use a genome from history** ».

Figure 1, illustrates that we used the default values for all options of Tophat.

Outputs

The tophat script produces a number of intermediate files, but in this pipeline, the intersting one is « **accepted_hits.bam** » which is a list of read alignments.

Figure 1



TopHat Gapped-read mapper for RNA-seq data (Galaxy Tool Version 0.9)

Is this single-end or paired-end data?

Paired-end (as individual datasets)

RNA-Seq FASTQ file, forward reads

1: 1.SRR070420_1.fastq

Must have Sanger-scaled quality values with ASCII offset 33

RNA-Seq FASTQ file, reverse reads

2: 2.SRR070420_2.fastq

Must have Sanger-scaled quality values with ASCII offset 33

Mean Inner Distance between Mate Pairs

300

-r/--mate-inner-dist; This is the expected (mean) inner distance between mate pairs. For, example, for paired end runs with fragments selected at 300bp, where each end is 50bp, you should set -r to be 200. The default is 50bp.

Std. Dev for Distance between Mate Pairs

20

--mate-std-dev; The standard deviation for the distribution on inner distances between mate pairs. The default is 20bp.

Report discordant pair alignments?

Yes

--no-discordant

Use a built in reference genome or own from your history

Use a genome from history

Built-ins genomes were created using default options

Select the reference genome

4: 4-sfru.mais.corrected.3.1.fa

TopHat settings to use

Use Defaults

You can use the default settings or set custom values for any of Tophat's parameters.

Specify read group?

No

2) Transcriptome assembly with Cufflinks

Cufflinks the program assembles transcriptomes from RNA-Seq data and quantifies their expression. One of Cufflinks' best features is that it can function as a reference-based transcriptome assembler. Cufflinks can use reference transcript annotation to guide assembly by using option *-g*. It can also identify novel transcripts in your sequencing data by examining their alignments to the genome. For more information see cole-trapnell-lab.github.io cufflinks tool documentation.

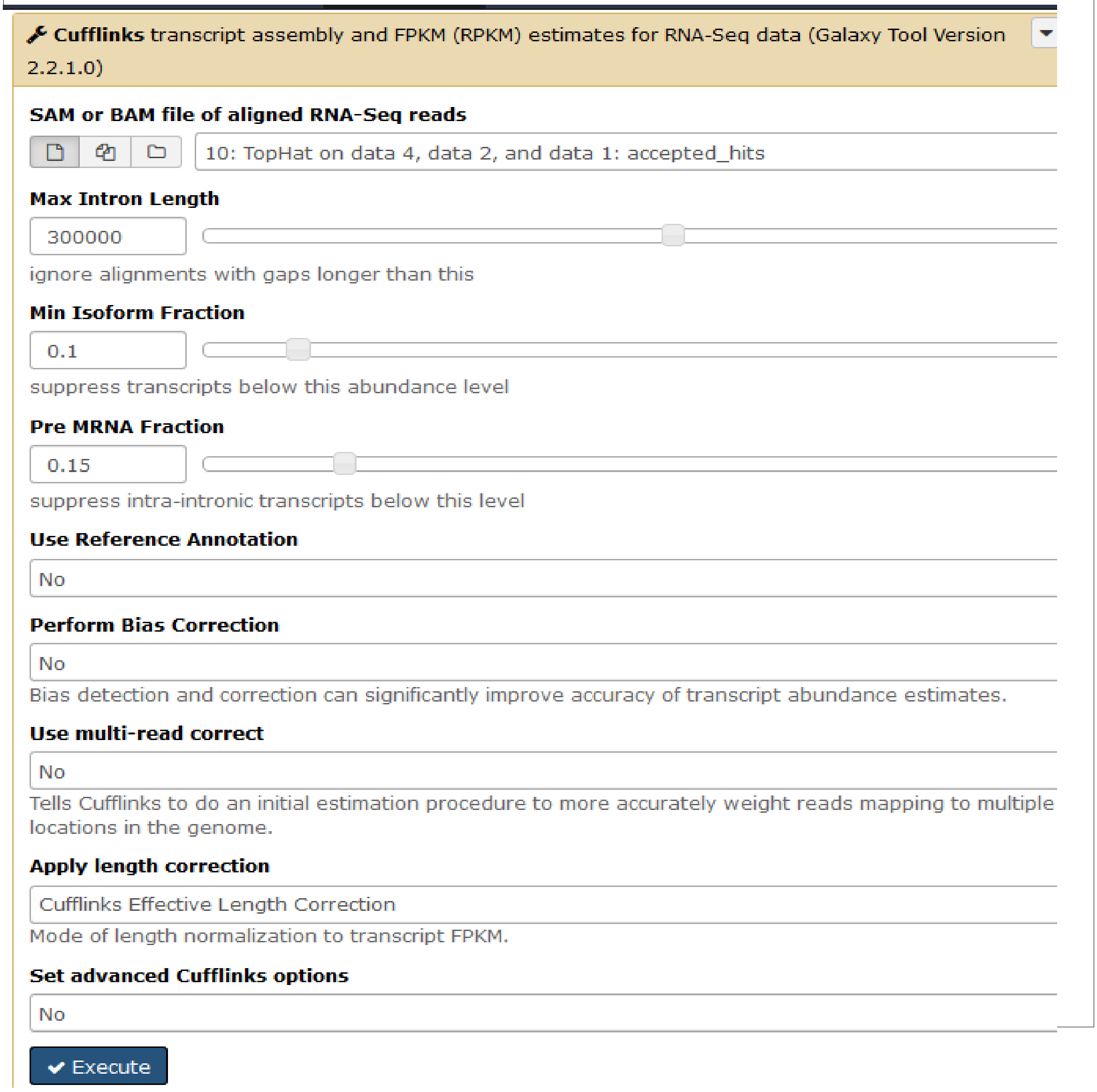
Input format

Cufflinks takes a list of read alignments in SAM or BAM format. In this pipeline, we use «accepted_hits.bam» which is generated by Tophat.
Please note that you should select « Use refrence annoaion as guide » to generate the novel transcripts from gene annotation dataset in GTF or GFF3 format (option *-g*).
Figure 2, illustrates that we used the default values for all options of Cufflinks.

Outputs

Cufflinks produces three output files, but in this pipeline, the intersting one is a GTF file « **transcript.gtf** » contains Cufflinks' assembled isoforms.

Figure 2



Cufflinks transcript assembly and FPKM (RPKM) estimates for RNA-Seq data (Galaxy Tool Version 2.2.1.0)

SAM or BAM file of aligned RNA-Seq reads

10: TopHat on data 4, data 2, and data 1: accepted_hits

Max Intron Length

300000

ignore alignments with gaps longer than this

Min Isoform Fraction

0.1

suppress transcripts below this abundance level

Pre MRNA Fraction

0.15

suppress intra-intronic transcripts below this level

Use Reference Annotation

No

Perform Bias Correction

No

Bias detection and correction can significantly improve accuracy of transcript abundance estimates.

Use multi-read correct

No

Tells Cufflinks to do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.

Apply length correction

Cufflinks Effective Length Correction

Mode of length normalization to transcript FPKM.

Set advanced Cufflinks options

No

3) Comparing the structure of assembled transcripts with Cuffcompare

After assembling a transcriptome from one or more samples, you’ll probably want to compare your assembly to known transcripts. Cuffcompare examines the structure of each isoforms (here, created transcript by cufflinks against a reference transcriptome) and then it generates a class code for each of them. see broadinstitute.org documentation on Cufflinks.cuffcompare.

Class Codes

If you ran cuffcompare with the -r option, tracking rows will contain the following values.

```
Priority    Code      Description
---------------------------------
1           =          Match
2           c          Contained
3           j          New isoform
4           e          A single exon transcript overlapping a reference
                       exon and at least 10 bp of a reference intron,
                       indicating a possible pre-mRNA fragment.
5           i          A single exon transcript falling entirely with a
                       reference intron
6           r          Repeat. Currently determined by looking at the
                       reference sequence and applied to transcripts
                       where at least 50% of the bases are lower case
7           p          Possible polymerase run-on fragment
8           u          Unknown, intergenic transcript
9           o          Unknown, generic overlap with reference
10          .          (.tracking file only, indicates multiple
classifications)
```

Input format

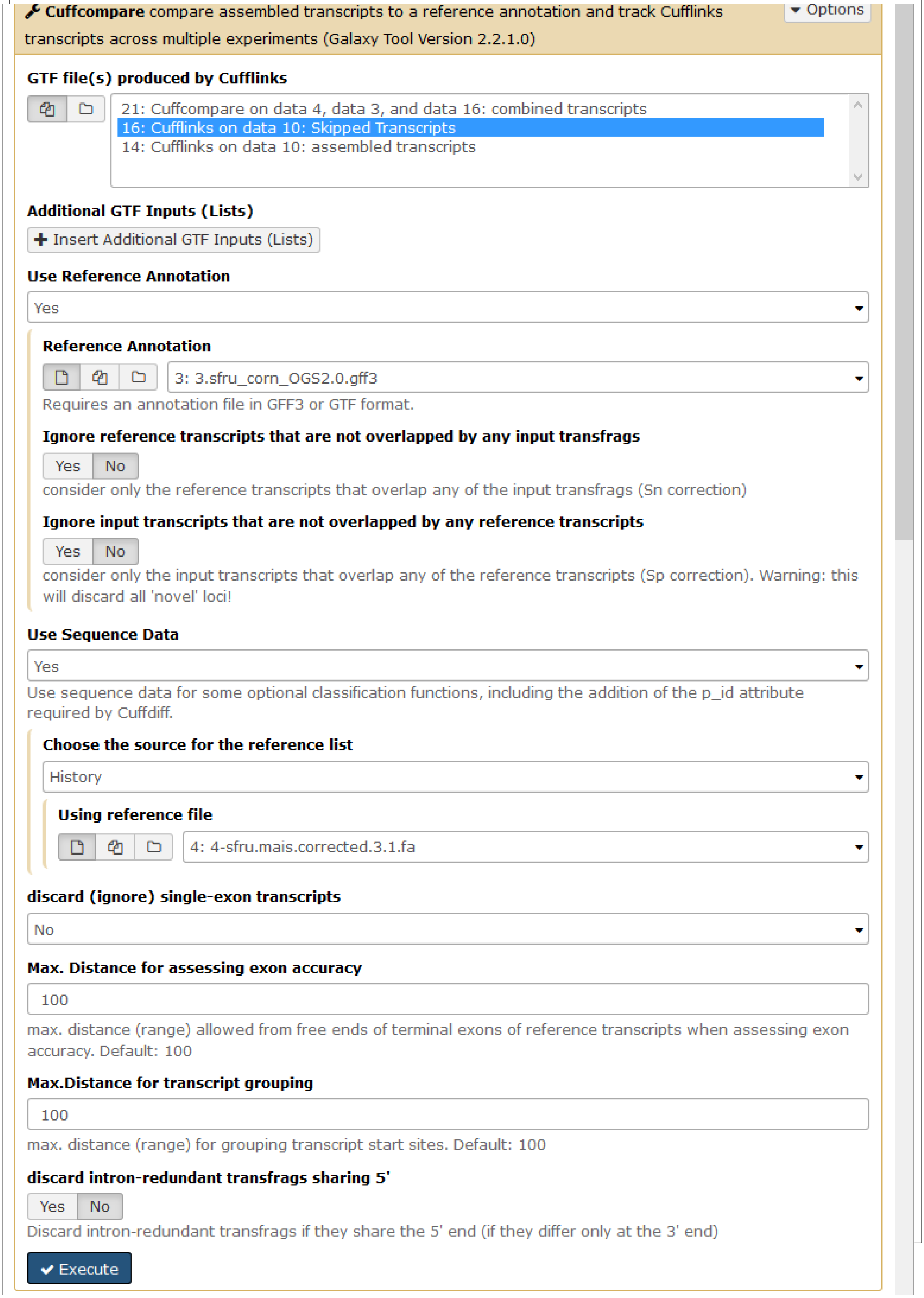
Cuffcompare takes a GTF file (here produced by Cufflinks) and a "reference" annotation.

Figure 3, illustrates that we used the default values for all options of Cuffcompare.

Outputs

Cufflinks produces four output files, but in this pipeline, the intersting one is a GTF file « transcripts.gtf.tmap» contains class code of each isoform according to reference.

Figure 3



Cuffcompare compare assembled transcripts to a reference annotation and track Cufflinks transcripts across multiple experiments (Galaxy Tool Version 2.2.1.0)

GTF file(s) produced by Cufflinks

- 21: Cuffcompare on data 4, data 3, and data 16: combined transcripts
- 16: Cufflinks on data 10: Skipped Transcripts
- 14: Cufflinks on data 10: assembled transcripts

Additional GTF Inputs (Lists)

Use Reference Annotation

Yes

Reference Annotation

3: 3.sfru_corn_OGS2.0.gff3

Requires an annotation file in GFF3 or GTF format.

Ignore reference transcripts that are not overlapped by any input transfrags

Yes [✓ No]

consider only the reference transcripts that overlap any of the input transfrags (Sn correction)

Ignore input transcripts that are not overlapped by any reference transcripts

Yes [✓ No]

consider only the input transcripts that overlap any of the reference transcripts (Sp correction). Warning: this will discard all 'novel' loci!

Use Sequence Data

Yes

Use sequence data for some optional classification functions, including the addition of the p_id attribute required by Cuffdiff.

Choose the source for the reference list

History

Using reference file

4: 4-sfru.mais.corrected.3.1.fa

discard (ignore) single-exon transcripts

No

Max. Distance for assessing exon accuracy

100

max. distance (range) allowed from free ends of terminal exons of reference transcripts when assessing exon accuracy. Default: 100

Max.Distance for transcript grouping

100

max. distance (range) for grouping transcript start sites. Default: 100

discard intron-redundant transfrags sharing 5'

Yes [✓ No]

Discard intron-redundant transfrags if they share the 5' end (if they differ only at the 3' end)

4) Converting transcriptome assembly to bed file

CufflinksGTFToBed is a python script to convert the transcriptome assembly of cufflinks (gtf file) to a bed file. This step parses a gtf file to extract all essential information for this pipeline and then writes output as bed file. Tracking columns of bed file will contain the following values.

```
Column      Description
---------------------------------
1           Name of the chromosome or scaffold of the transcript
2           Start position of the transcript
3           End position of the transcript
4           ID of the transcript
5           ID of the gene associated with the transcript
6           Strand of the transcript (defined as + (forward) or - (reverse).)
7           calculated FPKM of the transcript by Cufflinks
```

Input format

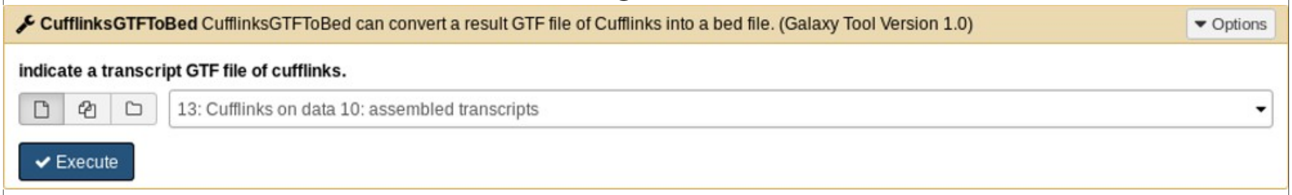
CufflinksGTFToBed takes a GTF file (here produced by Cufflinks).

Figure 4, illustrates a CufflinksGTFToBed schema.

Outputs

CufflinksGTFToBed produces a bed files .

Figure 4



CufflinksGTFToBed CufflinksGTFToBed can convert a result GTF file of Cufflinks into a bed file. (Galaxy Tool Version 1.0)

indicate a transcript GTF file of cufflinks.

13: Cufflinks on data 10: assembled transcripts

5) Converting transposable elements annotation file to bed file

GFFToBed is a python script to convert a gff file of the transposable elements (TEs) to a bed file. This step parses a gff file to extract all essential information for this pipeline and then writes output as bed file. Tracking columns of bed file will contain the following values.

```
Column      Description
---------------------------------
1           Name of the chromosome or scaffold of the TE
2           Start position of the TE
3           End position of the TE
4           ID of the TE
5           Target of the TE (here result of REPET)
6           Strand of the TE (defined as + (forward) or – (reverse).)
```

Input format

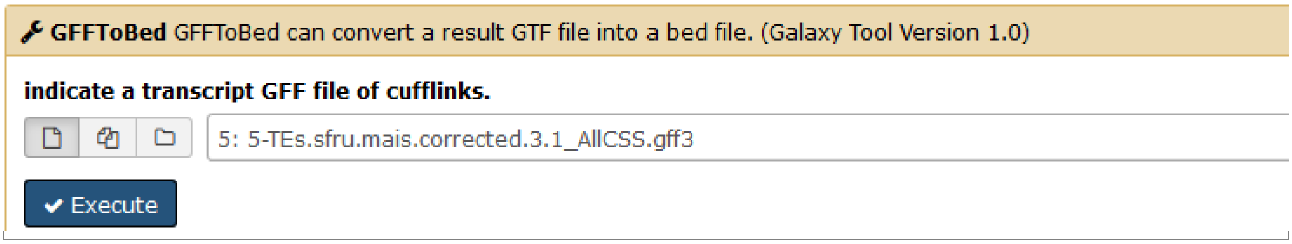
GFFToBed takes a GFF file (here produced by Cufflinks).

Figure 5 illustrates a GFFToBed schema.

Outputs

GFFToBed produces a bed files .

Figure 5



GFFToBed GFFToBed can convert a result GTF file into a bed file. (Galaxy Tool Version 1.0)

indicate a transcript GFF file of cufflinks.

5: 5-TEs.sfru.mais.corrected.3.1_AllCSS.gff3

6) Taking the closest transposable elements to transcriptomes with bedtools

Bedtools closest is a tool to find the closest locations between two bed files. TEiso goal is to find the nearest transposable elements (TEs) to each transcriptomes by Bedtools closest. As we explained above, two python scripts CufflinksGTFToBed and GFFToBed creat the bed files with corresponding to transcriptome assembly of cufflinks (step 4) and transposable elements (step 5) . Here, bedtools closest uses these two bed files to find the clossest TEs to each transcriptome (isoforms) and also calculates the distance between them. The output of bedtools closest is also into bed file. Tracking columns of final bed file will contain the following values.

```
Column      Description
---------------------------------
1           Name of the chromosome or scaffold of the transcript
2           Start position of the transcript
3           End position of the transcript
4           ID of the transcript
5           ID of the gene associated with the transcript
6           Strand of the transcript (defined as + (forward) or - (reverse).)
7           calculated FPKM of the transcript by Cufflinks
8           name of the chromosome or scaffold of the closest TE
9           Start position of the closest TE
10          End position of the closest TE
11          ID of the closest TE
12          Target of the closest TE (here result of REPET)
13          Strand of the closest TE (defined as + (forward) or - (reverse).)
14          Distance between closest TE and transcript.The reported distance for
overlapping/including features will be 0.
```

Input format

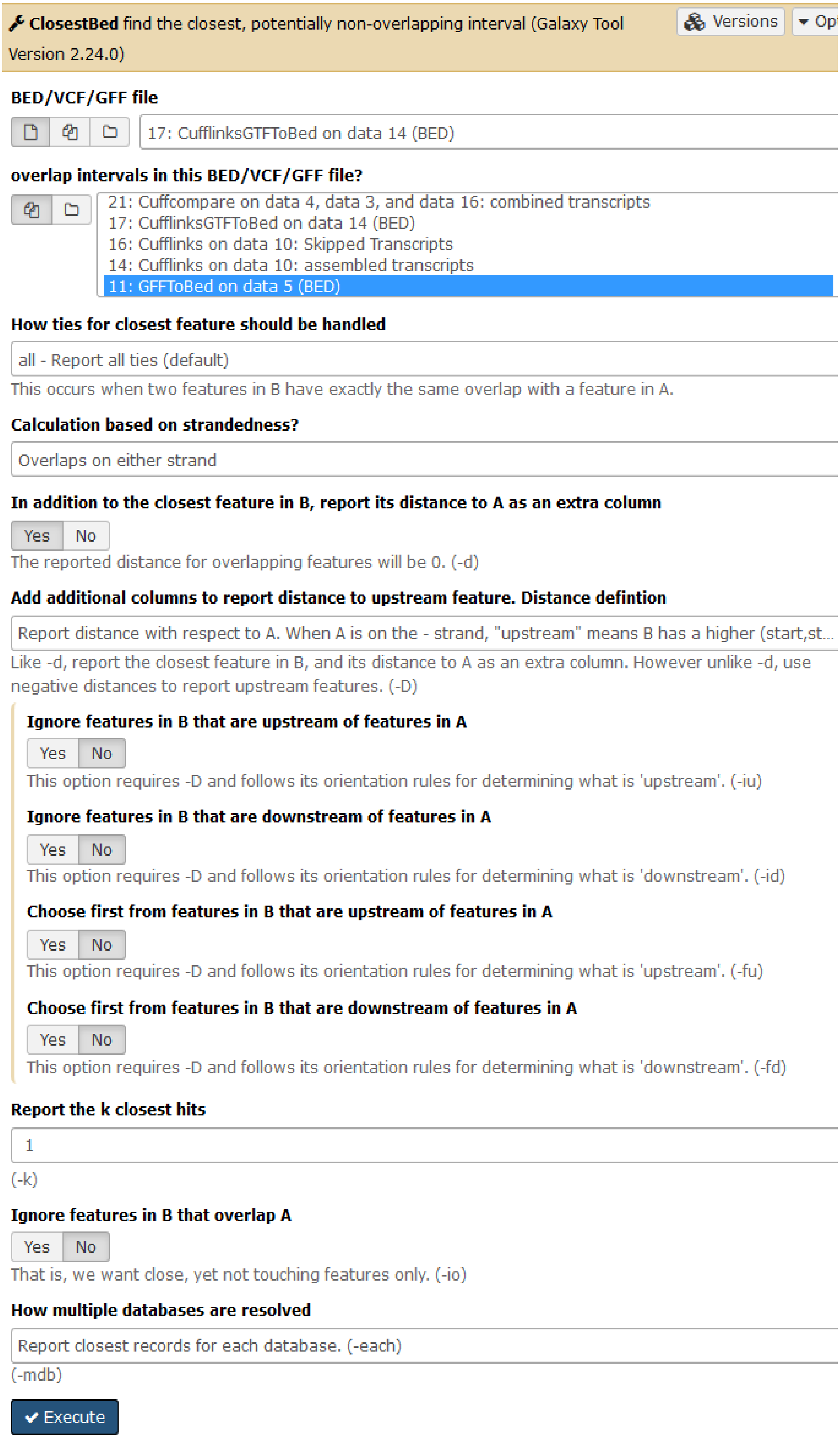
Bedtools closest takes two BED files. In this pipeline, the first bed file is for transcriptomes (produced by CufflinksGTFToBed) and the second one is for TEs (produced by GFFToBed).

Figure 6, illustrates that we used the default values for all options of Bedtools closest.

Outputs

Bedtools closest produces a output file in format Bed.

Figure 6



ClosestBed find the closest, potentially non-overlapping interval (Galaxy Tool Version 2.24.0)

BED/VCF/GFF file

17: CufflinksGTFToBed on data 14 (BED)

overlap intervals in this BED/VCF/GFF file?

21: Cuffcompare on data 4, data 3, and data 16: combined transcripts
17: CufflinksGTFToBed on data 14 (BED)
16: Cufflinks on data 10: Skipped Transcripts
14: Cufflinks on data 10: assembled transcripts
[✓ 11: GFFToBed on data 5 (BED)]

How ties for closest feature should be handled

all - Report all ties (default)

This occurs when two features in B have exactly the same overlap with a feature in A.

Calculation based on strandedness?

Overlaps on either strand

In addition to the closest feature in B, report its distance to A as an extra column

[✓ Yes] No

The reported distance for overlapping features will be 0. (-d)

Add additional columns to report distance to upstream feature. Distance defintion

Report distance with respect to A. When A is on the - strand, "upstream" means B has a higher (start,st...

Like -d, report the closest feature in B, and its distance to A as an extra column. However unlike -d, use negative distances to report upstream features. (-D)

Ignore features in B that are upstream of features in A

Yes [✓ No]

This option requires -D and follows its orientation rules for determining what is 'upstream'. (-iu)

Ignore features in B that are downstream of features in A

Yes [✓ No]

This option requires -D and follows its orientation rules for determining what is 'downstream'. (-id)

Choose first from features in B that are upstream of features in A

Yes [✓ No]

This option requires -D and follows its orientation rules for determining what is 'upstream'. (-fu)

Choose first from features in B that are downstream of features in A

Yes [✓ No]

This option requires -D and follows its orientation rules for determining what is 'downstream'. (-fd)

Report the k closest hits

1

(-k)

Ignore features in B that overlap A

Yes [✓ No]

That is, we want close, yet not touching features only. (-io)

How multiple databases are resolved

Report closest records for each database. (-each)

(-mdb)

7) Finding cases where TSS are closest to TEs

TEiso goal is to find the nearest transposable elements (TEs) to the TSS of each transcriptomes (isoforms). ClosestToStartSite is a python script to parses the results of bedtools closest (step 6) and extract the cases where TEs are close/overlap to/withTSS of each isoforms. The output of this step is a bed file. Tracking columns of final bed file will contain the following values.

```
Column      Description
---------------------------------
1           Name of the chromosome or scaffold of the transcript
2           Start position of the transcript
3           End position of the transcript
4           ID of the transcript
5           ID of the gene associated with the transcript
6           Strand of the transcript (defined as + (forward) or - (reverse).)
7           calculated FPKM of the transcript by Cufflinks
8           name of the chromosome or scaffold of the closest TE
9           Start position of the closest TE
10          End position of the closest TE
11          ID of the closest TE
12          Target of the closest TE (here result of REPET)
13          Strand of the closest TE (defined as + (forward) or - (reverse).)
14          Distance between closest TE and transcript.
            For Case TE overlap transcript: distance is region overlap
15          Discription of the TE's position according to the TSS
            TE_near_TSS ,  TE_overlap_TSS, TE-inclus-gene, gene-inclus-TE
```

Input format

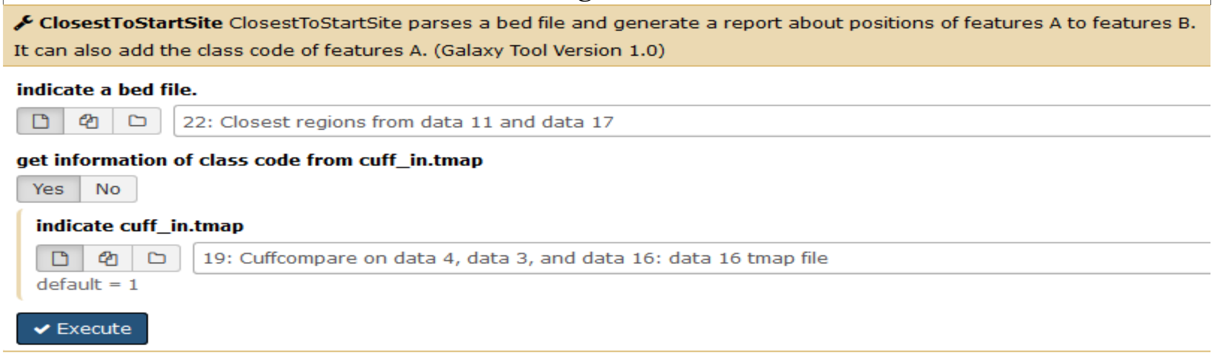
ClosestToStartSite takes a BED file which is produced by Bedtools closest. You can keep the information of class code of isoforms by choosing « get information of class code » and give a GTF file « transcripts.gtf.tmap» (produced by Cuffcompare).

Figure 7, illustrates a ClosestToStartSite schema.

Outputs

ClosestToStartSite produces an output file in format Bed.

Figure 7



ClosestToStartSite ClosestToStartSite parses a bed file and generate a report about positions of features A to features B. It can also add the class code of features A. (Galaxy Tool Version 1.0)

indicate a bed file.

22: Closest regions from data 11 and data 17

get information of class code from cuff_in.tmap

indicate cuff_in.tmap

19: Cuffcompare on data 4, data 3, and data 16: data 16 tmap file

default = 1