

STAR manual

(2.3.0.1)

Alex Dobin (dobin@cshl.edu)

Feb 11, 2013

CONTENTS

1: Installation.....	3
1.1: Ubuntu 12.04.1 LTS.....	3
1.2: Red Hat Enterprise Linux 6.3, CentOS 6.2.....	3
1.3: SUSE Linux Enterprise Server 11.....	3
2: Before running STAR.....	3
3: Generating genomes.....	3
4: Generating genomes with annotations.....	4
5: Running mapping jobs.....	4
6: Crucial input parameters.....	5
6.1: The following parameters are crucial for the STAR mapping runs:.....	5
6.2: The following parameters control the filtering of the output alignments:.....	5
6.2.1: To filter by mapped length and alignment score:.....	5
6.2.2: To control the output of multi-mappers, use the following parameters:.....	6
6.2.3: To control the number of mismatches per read pair, use:.....	6
6.2.4: To filter out alignments containing non-canonical junctions use:.....	6
6.2.5: To set maximum intron size:.....	6
7: Loading genome into shared memory.....	7
8: Output.....	7
8.1: Log files.....	7
8.2: SAM alignments.....	8
8.2.1: Standard SAM attributes.....	8
8.2.2: Extra SAM attributes.....	8
8.2.3: XS SAM strand attribute for Cufflinks/Cuffdiff.....	8
8.3: Splice junctions.....	9
9: All input parameters.....	10

1: Installation

Unzip/tar *STAR_x.x.x.tgz* file into a directory of your choice *< STARsource >*, *cd < STARsource >* and run *make*. The source code will be compiled and the *STAR* executable will be generated. Standard GNU C++ distribution is required for compilation. The pre-compiled executable *STAR*, included in the source directory, should work on any x86_64 Linux machine.

The compilation of STAR was tested on the following Amazon EC2 servers, the following commands were used to install the necessary packages:

1.1: Ubuntu 12.04.1 LTS

```
sudo apt-get update
sudo apt-get install g++
sudo apt-get install make
```

1.2: Red Hat Enterprise Linux 6.3, CentOS 6.2

```
sudo yum update
sudo yum install make
sudo yum install gcc-c++
sudo yum install glibc-static
```

1.3: SUSE Linux Enterprise Server 11

```
sudo zypper update
sudo zypper in gcc gcc-c++
```

2: Before running STAR

You need the following items before you can run STAR:

1. *STAR* executable: see installation
2. Genome files: download standard genomes from *STAR* web-site, or generate your own genomes using (see Generating genomes section).
3. Sequencing *.fastq* files: standard *.fastq* files (e.g. Illumina *.fastq* output files) or *.fasta* files.

Each STAR run should be made from a fresh working directory. All the output files are stored in the working directory. The output files will be overwritten without a warning every time you run STAR.

3: Generating genomes

From a fresh directory, run

```
/pathToStarDir/STAR --runMode genomeGenerate --genomeDir /path/to/GenomeDir --genomeFastaFiles /path/to/genome/fastal /path/to/genome/fastal2 --runThreadN <n> ...
```

Reference sequences (henceforth called “chromosomes”) will be collected from the all the specified *.fasta* files. Generated genome files comprise binary genome sequence, suffix arrays, text chromosome names and lengths information. You can rename the chromosomes’ names in the *chrName.txt* keeping the order of the chromosomes in the file: the names from this file will be used in all output alignment files (such as *.sam*). The *tabs* are not allowed in chromosomes’ names, and *spaces* are not recommended. For each genome these files are stored in a separate directory */path/to/GenomeDir* that has to be supplied as a parameter to the alignment runs. You can use multiple threads by specifying *--runThreadN <n>*.

4: Generating genomes with annotations

STAR can use a splice junctions’ database to improve accuracy of the mapping. The annotations have to be supplied at the genome/suffix array generation step. The annotations can be supplied in the form of splice junctions’ loci or GTF (or GFF3) file.

The splice junction loci are specified with

--sjdbFileChrStartEnd <FileName>: the file with annotated introns loci in a four-column format:

```
Chr \tab\ Start \tab\ End \tab\ Strand(+or-)
```

where Start and End are first and last bases of the introns (1-based chromosome coordinates).

The GTF file can be specified with *--sjdbGTFfile <FileName>*.

For GFF3 also specify *--sjdbGTFtagExonParentTranscript Parent*.

Another parameter that has to be specified at the genome generation step is

--sjdbOverhang <N>: the length of the "overhang" on each side of a splice junctions. Ideally it should be equal to (MateLength - 1).

At the mapping stage, *--sjdbScore <N>* (=2 by default) provides extra alignment score for alignments that cross database junctions. If this score is positive, it will bias the alignment toward annotated junctions.

5: Running mapping jobs

From a fresh working directory run:

```
/pathToStarDir/STAR --genomeDir /path/to/GenomeDir --readFilesIn /path/to/read1 [/path/to/read2] --runThreadN <n> --<inputParameterName> <input parameter value(s)> ...
```

All input parameters are entered as *--<inputParameterName> <input parameter value(s)>*. If a parameter allows several values, the values are separated by spaces. For advanced usage, parameters can also be input from config files *Parameters1.in* and *Parameters2.in* in the working directory. In these files each parameter

with its values occupies one line without preceding dashes (--). There are four levels of input parameters, *Default*, *Parameters1.in*, *Parameters2.in*, *Command line*; each level overrides the previous ones.

6: Crucial input parameters

The default parameters are typical for mapping 2x76 or 2x101 Illumina reads to the human genome. Brief description of all parameters can be found at the end of this manual. For each STAR version, the most up-to-date information about all STAR parameters can be found in the parametersDefault file in the STAR source directory. The parameters in the parametersDefault file are grouped by function.

Special attention has to be paid to parameters that start with 'out*', as they control the STAR output. In particular, outFilter* parameters control the filtering of alignments which you might want to tweak to your needs.

Annotations (splice junction database) are controlled by sjdb* options at the genome generation step (which is controlled by genome* parameters).

Output of "chimeric" alignments is controlled by chim* parameters.

Tweaking score*, align*, seed*, win* parameters, which requires understanding of the STAR alignment algorithm, is recommended only for advanced users.

6.1: The following parameters are crucial for the STAR mapping runs:

PARAMETER	DEFAULT
NAME	VALUE
genomeDir	GenomeDir/ string: path to the directory where genome files are stored
readFilesIn	Read1 Read2 string(s): paths to .fastq or .fasta files that contain input read1 (and, if needed, read2)
runThreadN	1 int>0: the number of threads to use

6.2: The following parameters control the filtering of the output alignments:

6.2.1: To filter by mapped length and alignment score:

outFilterScoreMin	0 int: alignment will be output only if its score is higher than this value
outFilterScoreMinOverLread	0.66

float: outFilterScoreMin normalized to read length (sum of mates' lengths for paired-end reads)

outFilterMatchNmin 0

int: alignment will be output only if the number of matched bases is higher than this value

outFilterMatchNminOverLread 0.66

float: outFilterMatchNmin normalized to read length (sum of mates' lengths for paired-end reads)

6.2.2: To control the output of multi-mappers, use the following parameters:

outFilterMultimapScoreRange 1

int: the score range below the maximum score for multimapping alignments

outFilterMultimapNmax 10

int: read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output

6.2.3: To control the number of mismatches per read pair, use:

outFilterMismatchNmax 10

int: alignment will be output only if it has fewer mismatches than this value

outFilterMismatchNoverLmax 0.3

int: alignment will be output only if its ratio of mismatches to mapped length is less than this value

6.2.4: To filter out alignments containing non-canonical junctions use:

outFilterIntronMotifs None

string: filter alignment using their motifs

None : no filtering

RemoveNoncanonical : filter out

alignments that contain non-canonical junctions

RemoveNoncanonicalUnannotated : filter

out alignments that contain non-canonical unannotated junctions when using annotated splice junctions database. The annotated non-canonical junctions will be kept.

6.2.5: To set maximum intron size:

`--alignIntronMax` is the max intron size for a splice that occurs inside each mate, i.e. the one recorded as xxxN in the .sam CIGAR.

`-- alignMatesGapMax` is the maximum genomic gap between the mates, i.e. inner distance between the right end of the left mate and left side of the right mate. Note that this gap is not truly an intron, and it could contain several introns. If the (insert-2*mate) of the library is not very large (< typical exon size), you can assume that there is no more than one intron in the unsequenced RNA between the mates, and so you can use `alignMatesGapMax=alignIntronMax` However, for longer reads it might be necessary to use `alignMatesGapMax>alignIntronMax` to allow more than one intron in the genomic gap between the mates.

7: Loading genome into shared memory

The `--genomeLoad` parameter controls how the genome is loaded into memory. If `genomeLoad=LoadAndKeep`, STAR loads the genome as a standard Linux shared memory piece. Before loading the genome, STAR will check if the genome has already been loaded into the shared memory. The genomes are identified by their unique directory paths. If the genome has not been loaded, STAR job will load it and will keep it in memory even after STAR job itself finishes. The genome will be shared with all the other STAR instances. You can remove the genome from the shared memory running STAR with `genomeLoad=Remove`. The shared memory piece will be physically removed only after all STAR jobs attached to it complete. If `genomeLoad=LoadAndRemove`, STAR will load genome in the shared memory, and mark it for removal, so that the genome will be removed from the shared memory once all STAR jobs using it exit. If `genomeLoad=LoadAndExit`, STAR will load genome in the shared memory, and immediately exit without performing any alignment, keeping the genome loaded in the shared memory for the future runs.

If you need to check or remove shared memory pieces manually, use the standard Linux command `ipcs` and `ipcrm`. If the genome residing in shared memory is not used for a long time it may get paged out of RAM which will slow down STAR runs considerably. It is strongly recommended to regularly re-load (i.e. remove and load again) the shared memory genomes.

If `genomeLoad=NoSharedMemory`, shared memory is not used. This option is recommended if the shared memory is not configured properly on your server.

Many standard Linux distributions do not allow large enough shared memory blocks. You can fix this issue if you have root privileges, or ask you system administrator to do it. To enable the shared memory modify or add the following lines to `/etc/sysctl.conf`:

```
kernel.shmmax = 31000000000
```

```
kernel.shmall = 31000000000
```

Then run:

```
> /sbin/sysctl -p
```

This will increase the allowed shared memory blocks to ~31GB, enough for human or mouse genome.

8: Output

8.1: Log files

`Log.out` - log files with some information about the run recorded.

Log.progress.out - job progress with the number of processed reads, % of mapped reads etc., updated every ~1 minute

Log.final.out – final mapping statistics

8.2: SAM alignments

Aligned.out.sam - alignments in standard SAM format.

The number of loci N_{map} a read maps to (multi-mapping) is given by NH:i: field.

The mapping quality MAPQ (column 5) is 255 for uniquely mapping reads, and $int(-10*\log_{10}(1-1/N_{map}))$ for multi-mapping reads. This scheme is same as the one used by Tophat and is compatible with Cufflinks.

For multi-mappers, all alignments except one are marked with 0x100 (secondary alignment) in the FLAG column 2. The un-marked alignment is either the best one (i.e. highest scoring), or is randomly selected from the alignments of equal quality.

8.2.1: Standard SAM attributes

With default `--outSAMattributes Standard` option the following SAM attributes will be generated:

Column 12: NH: number of loci a read (pair) maps to

Column 13: IH: alignment index for all alignments of a read

Column 14: aS: alignment score

Column 15: nM: number of mismatches (does not include indels)

8.2.2: Extra SAM attributes

If `--outSAMattributes All` option is used, the following additional attributes will be output:

Column 16: jM:B:c, M1, M2, ... Intron motifs for all junctions (i.e. N in CIGAR): 0: non-canonical; 1: GT/AG, 2: CT/AC, 3: GC/AG, 4: CT/GC, 5: AT/AC, 6: GT/AT. If splice junctions database is used, and a junction is annotated, 20 is added to its motif value.

Column 17: jI:B:I, Start1, End1, Start2, End2, ... Start and End of introns for all junctions (1-based)

Note, that samtools 0.1.18 or later have to be used with these extra attributes.

8.2.3: XS SAM strand attribute for Cufflinks/Cuffdiff

If you have **un-stranded RNA-seq** data, and wish to run Cufflinks/Cuffdiff on STAR alignments, you will need to run STAR with `--outSAMstrandField intronMotif` option, which will generate the XS strand attribute for all alignments that contain splice junctions. The spliced alignments that have undefined strand (i.e. containing only non-canonical junctions) will be suppressed.

If you have **stranded RNA-seq data**, you do not need to use any specific STAR options. Instead, you need to run Cufflinks with the library option `--library-type` options. For example, `cufflinks <...> -library-type fr-firststrand` should be used for the “standard” dUTP protocol. This option has to be used only for Cufflinks runs and **not** for STAR runs.

It is recommended to remove the non-canonical junctions for Cufflinks runs using –

```
--outFilterIntronMotifs RemoveNoncanonical
```

filter out alignments that contain non-canonical junctions

OR

```
--outFilterIntronMotifs RemoveNoncanonicalUnannotated
```

filter out alignments that contain non-canonical unannotated junctions when using annotated splice junctions database. The annotated non-canonical junctions will be kept.

8.3: Splice junctions

SJ.out.tab - high confidence collapsed splice junctions in tab-delimited format. Only junctions supported by uniquely mapping reads are reported.

Column 1: chromosome

Column 2: first base of the intron (1-based)

Column 3: last base of the intron (1-based)

Column 4: strand

Column 5: intron motif: 0: non-canonical; 1: GT/AG, 2: CT/AC, 3: GC/AG, 4: CT/GC, 5: AT/AC, 6: GT/AT

Column 6: 0: unannotated, 1: annotated (only if splice junctions database is used)

Column 7: number of uniquely mapping reads crossing the junction

Column 8: number of multi-mapping reads crossing the junction

Column 9: maximum spliced alignment overhang

The filtering for this output file is controlled by the following parameters (see *parametersDefault* for description) `--outSJfilterReads`, `--outSJfilterOverhangMin`, `--outSJfilterCountUniqueMin`, `--outSJfilterCountTotalMin`, `--outSJfilterDistToOtherSJmin`, `--outSJfilterIntronMaxVsReadN`

9: All input parameters

The most up-to-date input parameters and their defaults can be found in the *parametersDefault* file in the STAR source directory.

PARAMETERS

parametersFiles -
 string: name of a user-defined parameters file, "-": none. Can only be defined on the command line.

RUN PARAMETERS

runMode alignReads
 string: type of the run: alignReads ... map reads
 genomeGenerate ... generate genome files

runThreadN 1
 int: number of threads to run STAR

GENOME PARAMETERS

genomeDir ./GenomeDir/
 string: path to the directory where genome files are stored (if runMode!=generateGenome) or will be generated (if runMode==generateGenome)

genomeLoad NoSharedMemory
 mode of shared memory usage for the genome files
 string: LoadAndKeep ... load genome into shared and keep it in memory after run
 LoadAndRemove ... load genome into shared but remove it after run
 LoadAndExit ... load genome into shared memory and exit, keeping the genome in memory for future runs

Remove ... do not map anything, just
remove loaded genome from memory

NoSharedMemory ... do not use shared memory,
each job will have its own private copy of the genome

GENOME GENERATION PARAMETERS

genomeFastaFiles -

fasta files with genomic sequence for genome files generation, only
used if runMode==genomeGenerate

string(s): path(s) to the files from the working directory (separated
by spaces)

genomeChrBinNbits 18

int: =log2(chrBin), where chrBin is the size of the bins for genome
storage: each chromosome will occupy an integer number of bins

genomeSAindexNbases 14

int: length (bases) of the SA pre-indexing string. Typically between
10 and 15. Longer strings will use much more memory, but allow faster
searches.

genomeSAsparseD 1

int>0: suffix array sparsity, i.e. distance between indices: use
bigger numbers to decrease needed RAM at the cost of mapping speed
reduction

READ PARAMETERS

readFilesIn Read1 Read2

string(s): paths to files that contain input read1 (and, if needed,
read2)

readFilesCommand -

string(s): command line to execute for each of the input file. This command should generate FASTA or FASTQ text and send it to stdout

For example: zcat - to uncompress .gz files, bzip2 - to uncompress .bz2 files, etc.

readMatesLengthsIn NotEqual

string: Equal/NotEqual - lengths of names, sequences, qualities for both mates are the same / not the same. NotEqual is safe in all situations.

clip3pNbases 0

int(s): number(s) of bases to clip from 3p of each mate. If one value is given, it will be assumed the same for both mates.

clip5pNbases 0

int(s): number(s) of bases to clip from 5p of each mate. If one value is given, it will be assumed the same for both mates.

clip3pAdapterSeq -

string(s): adapter sequences to clip from 3p of each mate. If one value is given, it will be assumed the same for both mates.

clip3pAdapterMMp 0.1

double(s): max proportion of mismatches for 3p adapter clipping for each mate. If one value is given, it will be assumed the same for both mates.

clip3pAfterAdapterNbases 0

int(s): number of bases to clip from 3p of each mate after the adapter clipping. If one value is given, it will be assumed the same for both mates.

LIMITS

limitGenomeGenerateRAM 31000000000

```

    int>0: maximum available RAM (bytes) for genome generation

limitIObufferSize                150000000
    int>0: max available buffers size (bytes) for input/output, per
thread

### OUTPUT
outFileNamePrefix                ./
    string: output files name prefix (including full or relative path).
Can only be defined on the command line.

outStd                            Log
    string: which output will be directed to stdout (standard out)
                Log : log messages
                SAM : alignments in .sam format (which
normally are output to Aligned.out.sam file), normal standard output will
go into Log.std.out

outSAMmode                        Full
    string: mode of SAM output  None : no SAM output
                Full : full SAM output

outSAMstrandField                 None
    string: Cufflinks-like strand field flag  None : not used
                intronMotif : strand
derived from the intron motif. Reads with inconsistent and/or non-
canonical introns are filtered out.

outSAMattributes                  Standard
    string: which SAM attributes to output?
                Standard : NH, HI, AS, nM attributes
                All
                None

outSAMunmapped                    None

```

```

    string: output of unmapped reads in the SAM format
                None      : no output
                Within    : output unmapped reads within
the main SAM file (i.e. Aligned.out.sam)

outReadsUnmapped      None
    string: output of unmapped reads (besides SAM)
                None      : no output
                Fastx     : output in separate fasta/fastq
files, Unmapped.out.mate1/2

outQSconversionAdd    0
    int: add this number to the quality score (e.g. to convert from
Illumina to Sanger, use -31)

### OUTPUT FILTERING
outFilterType          Normal
    string: type of filtering
                Normal: standard filtering using only
current alignment
                BySJout: keep only those reads that
contain junctions that passed filtering into SJ.out.tab

outFilterMultimapScoreRange  1
    int: the score range below the maximum score for multimapping
alignments

outFilterMultimapNmax    10
    int: read alignments will be output only if the read maps fewer than
this value, otherwise no alignments will be output

outFilterMismatchNmax    10
    int: alignment will be output only if it has fewer mismatches than
this value

```

```

outFilterMismatchNoverLmax      0.3
    int: alignment will be output only if its ratio of mismatches to
mapped length is less than this value

outFilterScoreMin                0
    int: alignment will be output only if its score is higher than this
value

outFilterScoreMinOverLread      0.66
    float: outFilterScoreMin normalized to read length (sum of mates'
lengths for paired-end reads)

outFilterMatchNmin              0
    int: alignment will be output only if the number of matched bases is
higher than this value

outFilterMatchNminOverLread     0.66
    float: outFilterMatchNmin normalized to read length (sum of mates'
lengths for paired-end reads)

outFilterIntronMotifs           None
    string: filter alignment using their motifs
                None : no filtering
                RemoveNoncanonical : filter out
alignments that contain non-canonical junctions
                RemoveNoncanonicalUnannotated : filter
out alignments that contain non-canonical unannotated junctions when
using annotated splice junctions database. The annotated non-canonical
junctions will be kept.

### OUTPUT FILTERING: SPLICE JUNCTIONS
outSJfilterReads                All

```

string: which reads to consider for collapsed splice junctions output
All: all reads, unique- and multi-mappers
Unique: uniquely mapping reads only

outSJfilterOverhangMin 30 12 12 12

4*int: minimum overhang length for splice junctions on both sides
for: (1) non-canonical motifs, (2) GT/AG motif, (3) GC/AG motif, (4) AT/
AC motif. -1 means no output for that motif

does not apply to annotated junctions

outSJfilterCountUniqueMin 3 1 1 1

4*int: minimum uniquely mapping read count per junction for: (1) non-
canonical motifs, (2) GT/AG motif, (3) GC/AG motif, (4) AT/AC motif. -1
means no output for that motif

Junctions are output if one of

outSJfilterCountUniqueMin OR outSJfilterCountTotalMin conditions are
satisfied

does not apply to annotated junctions

outSJfilterCountTotalMin 3 1 1 1

4*int: minimum total (multi-mapping+unique) read count per junction
for: (1) non-canonical motifs, (2) GT/AG motif, (3) GC/AG motif, (4) AT/
AC motif. -1 means no output for that motif

Junctions are output if one of

outSJfilterCountUniqueMin OR outSJfilterCountTotalMin conditions are
satisfied

does not apply to annotated junctions

outSJfilterDistToOtherSJmin 10 0 5 10

4*int>=0: minimum allowed distance to other junctions' donor/acceptor

does not apply to annotated junctions

outSJfilterIntronMaxVsReadN 50000 100000 200000

N*int>=0: maximum gap allowed for junctions supported by 1,2,3...N
reads

i.e. by default junctions supported by 1
read can have gaps ≤ 50000 b, by 2 reads: ≤ 100000 b, by 3 reads: ≤ 200000 .
by ≥ 4 reads any gap $\leq \text{alignIntronMax}$

does not apply to annotated junctions

SCORING

scoreGap 0

gap open penalty

scoreGapNoncan -8

non-canonical gap open penalty (in addition to scoreGap)

scoreGapGCAG -4

GCAG gap open penalty (in addition to scoreGap)

scoreGapATAC -8

ATAC gap open penalty (in addition to scoreGap)

scoreGenomicLengthLog2scale -0.25

extra score logarithmically scaled with genomic length of the
alignment: $-\text{scoreGenomicLengthLog2scale} \cdot \log_2(\text{genomicLength})$

scoreDelOpen -2

deletion open penalty

scoreDelBase -2

deletion extension penalty per base (in addition to scoreDelOpen)

scoreInsOpen -2

insertion open penalty

scoreInsBase -2

insertion extension penalty per base (in addition to scoreInsOpen)

```

scoreStitchSJshift          1
    maximum score reduction while searching for SJ boundaries inthe
    stitching step

### ALIGNMENT and SEED PARAMETERS

seedSearchStartLmax         50
    int>0: defines the search start point through the read - the read is
    split into pieces no longer than this value

seedSearchStartLmaxOverLread  1.0
    float: seedSearchStartLmax normalized to read length (sum of mates'
    lengths for paired-end reads)

seedSearchLmax              0
    int>=0: defines the maximum length of the seeds, if =0 max seed
    lengthis infinite

seedMultimapNmax            10000
    int>0: only pieces that map fewer than this value are utilized in the
    stitching procedure

seedPerReadNmax             1000
    int>0: max number of seeds per read

seedPerWindowNmax           50
    int>0: max number of seeds per window

seedNoneLociPerWindow       10
    int>0: max number of one seed loci per window

alignIntronMin              21
    minimum intron size: genomic gap is considered intron if its
    length>=alignIntronMin, otherwise it's considered Deletion

```

```

alignIntronMax          0
    maximum intron size, if 0, max intron size will be determined by
    (2^winBinNbits)*winAnchorDistNbins

alignMatesGapMax        0
    maximum gap between two mates, if 0, max intron gap will be
    determined by (2^winBinNbits)*winAnchorDistNbins

alignSJoverhangMin      5
    int>0: minimum overhang (i.e. block size) for spliced alignments

alignSJDBoverhangMin    3
    int>0: minimum overhang (i.e. block size) for annotated (sjdb)
    spliced alignments

alignSplicedMateMapLmin  0
    int>0: minimum mapped length for a read mate that is spliced

alignSplicedMateMapLminOverLmate 0.66
    float>0: alignSplicedMateMapLmin normalized to mate length

alignWindowsPerReadNmax 10000
    int>0: max number of windows per read

alignTranscriptsPerWindowNmax 100
    int>0: max number of transcripts per window

alignTranscriptsPerReadNmax 10000
    max number of different alignments per read to consider

### SPLICE JUNCTIONS DATABASE PARAMETERS
sjdbFileChrStartEnd      -

```

string: path to the file with genomic coordinates (chr <tab> start <tab> end <tab> strand) for the introns

sjdbGTFfile -
string: path to the GTF file with annotations

sjdbGTFchrPrefix -
string: prefix for chromosome names in a GTF file (e.g. 'chr' for using ENSMEBL annotations with UCSC geneomes)

sjdbGTFfeatureExon exon
string: feature type in GTF file to be used as exons for building transcripts

sjdbGTFtagExonParentTranscript transcript_id
string: tag name to be used as exons' parents for building transcripts

sjdbOverhang 0
int>=0: length of the donor/acceptor sequence on each side of the junctions, ideally = (mate_length - 1)
if =0, splice junction database is not used

sjdbScore 2
int: extra alignment score for alignmets that cross database junctions

WINDOWS, ANCHORS, BINNING

winAnchorMultimapNmax 50
int>0: max number of loci anchors are allowed to map to

winBinNbits 16
int>0: =log2(winBin), where winBin is the size of the bin for the windows/clustering, each window will occupy an integer number of bins.

```

winAnchorDistNbins          9
    int>0: max number of bins between two anchors that allows aggregation
of anchors into one window

winFlankNbins              4
    int>0: log2(winFlank), where win Flank is the size of the left and
right flanking regions for each window

### CHIMERIC ALIGNMENTS

chimSegmentMin             0
    int>0: minimum length of chimeric segment length, if ==0, no chimeric
output

chimScoreMin               0
    int>0: minimum total (summed) score of the chimeric segments

chimScoreDropMax           20
    int>0: max drop (difference) of chimeric score (the sum of scores of
all chimeric segments) from the read length

chimScoreSeparation        10
    int>0: minimum difference (separation) between the best chimeric
score and the next one

chimScoreJunctionNonGTAG   -1
    int: penalty for a non-GT/AG chimeric junction

chimJunctionOverhangMin    20
    int>0: minimum overhang for a chimeric junction

```