STAR Alignment - Comprehensive Reference Guide

Overview

STAR (Spliced Transcripts Alignment to a Reference) is an ultrafast, splice-aware aligner designed for RNA-seq data. It maps reads to a reference genome while accurately detecting splice junctions.

Website: https://github.com/alexdobin/STAR
Publication: Dobin et al. Bioinformatics 2013
Current Version: 2.7.10a+

Applicable to: RNA-seq, small RNA-seq, long-read RNA-seq, circRNA detection, fusion gene detection

---

What STAR Does

Core Functions

1. **Read Mapping**: Aligns reads to genome with mismatches and gaps
2. **Splice Detection**: Identifies exon-exon junctions (splice sites)
3. **Novel Junction Discovery**: Finds junctions not in annotation
4. **Gene Quantification**: Counts reads per gene
5. **Junction Quantification**: Counts reads supporting each junction

STAR vs Other Aligners

Feature	STAR	HISAT2	TopHat2
Speed	Very fast	Fast	Slow (deprecated)
Memory	High (30GB)	Low (8GB)	Moderate
Accuracy	Excellent	Excellent	Good
Novel junctions	Two-pass mode	Yes	Yes
Gene counts	Built-in	Via HTSeq	Via HTSeq
Long reads	Excellent	Good	Poor
Maintenance	Active	Active	Deprecated

Basic Alignment Command

```
STAR \
    --genomeDir star_index \
    --readFilesIn R1.fastq.gz R2.fastq.gz \
    --readFilesCommand zcat \
    --outFileNamePrefix sample1. \
    --outSAMtype BAM SortedByCoordinate \
    --runThreadN 8
```

Command Breakdown

```
STAR \                                    # STAR aligner program
    --genomeDir star_index \              # Path to STAR index directory
    --readFilesIn R1.fq.gz R2.fq.gz \     # Input FASTQ files (PE shown)
    --readFilesCommand zcat \             # Decompress .gz on-the-fly
    --outFileNamePrefix sample1. \        # Prefix for all output files
    --outSAMtype BAM SortedByCoordinate \ # Output sorted BAM
    --runThreadN 8                        # Use 8 CPU cores
```

Single-end data:

```
--readFilesIn R1.fastq.gz    # Only one file
```

Paired-end data:

```
--readFilesIn R1.fastq.gz R2.fastq.gz    # Two files: R1 and R2
```

Key Parameters Explained

Input/Output Parameters

--readFilesCommand **Purpose**: Specifies how to read/decompress input files

Options:

```
--readFilesCommand zcat           # For .gz files (most common)
--readFilesCommand bzcat          # For .bz2 files
--readFilesCommand gunzip -c      # Alternative for .gz
--readFilesCommand cat            # For uncompressed files
```

Why use compression:

- Saves disk space (5-10x smaller)
- Only ~5% slower than uncompressed
- No need to pre-decompress

--outSAMtype **Purpose**: Output format and sorting

Options:

```
--outSAMtype SAM                         # SAM format (text, large)
--outSAMtype BAM Unsorted                # BAM format (binary, unsorted)
--outSAMtype BAM SortedByCoordinate      # BAM sorted by position (RECOMMENDED)
```

Format comparison:

Format	Size	Speed	Usability
SAM	10-20GB	Slow	Human-readable
BAM Unsorted	2-5GB	Fast	Needs sorting
BAM Sorted	2-5GB	Moderate	Ready for IGV/RSeQC

Recommendation: Always use `BAM SortedByCoordinate`

--outFileNamePrefix **Purpose**: Prefix for all output filenames

Example:

```
--outFileNamePrefix sample1.

# Generates:
sample1.Aligned.sortedByCoord.out.bam
sample1.Log.final.out
sample1.SJ.out.tab
sample1.ReadsPerGene.out.tab
```

Best practices:

- Use sample ID as prefix
- Include trailing dot (.)
- Avoid spaces or special characters

Two-Pass Mode

--twopassMode Basic **Purpose**: Improves mapping by discovering novel junctions

How it works:

```
First Pass:
  1. Align all reads
  2. Detect novel splice junctions
  3. Filter junctions by support

Second Pass:
  1. Add novel junctions to database
```

```
2. Re-align ALL reads with expanded database
3. Improved sensitivity for junction-spanning reads
```

Benefits:

- 2-5% improvement in mapping rate
- Discovers unannotated isoforms
- Better for incomplete annotations
- Essential for non-model organisms

Trade-offs:

- ~30% slower (50 min vs 35 min)
- Uses more disk space temporarily
- More memory for junction database

When to use:

- Standard RNA-seq analysis
- Incomplete genome annotations
- Novel isoform discovery
- Speed is critical (quick QC)
- Very well-annotated genomes only

Gene Quantification

--quantMode GeneCounts **Purpose**: Generate gene-level read counts

Output: `ReadsPerGene.out.tab` with 4 columns:

```
gene_id          unstranded   sense   antisense
ENSG00000000001  1000         1050    50
ENSG00000000002  500          25      480
```

Column explanations:

Column	Description	When to Use
1	Gene ID	-
2	Unstranded	Unstranded libraries
3	Sense strand	Stranded (dUTP, Illumina TruSeq)
4	Antisense strand	Reverse-stranded protocols

How to choose column:

1. **Check your library prep protocol**:
    - Unstranded → Column 2
    - Stranded (dUTP) → Column 3

    - Reverse-stranded → Column 4

2. **If unsure, check distribution**:

   ```
   # Most reads should be in ONE column (not split)
   head -n 20 sample.ReadsPerGene.out.tab
   ```

3. **Or use MultiQC** - shows library type detection

Use in R/DESeq2:

```
# Read counts
counts <- read.table("ReadsPerGene.out.tab", skip=4, row.names=1)
# Column 2 for unstranded, 3 for stranded
gene_counts <- counts[, 2]  # or counts[, 3] for stranded
```

Splice Junction Parameters

--sjdbOverhang **Purpose**: Maximum overhang for annotated junctions

Formula: `ReadLength - 1`

Examples:

```
50bp reads  → --sjdbOverhang 49
75bp reads  → --sjdbOverhang 74
100bp reads → --sjdbOverhang 99
150bp reads → --sjdbOverhang 149
```

Note: This should match value used during indexing!

Universal value: 100 works well for 75-150bp reads

Multi-Mapping Parameters

--outFilterMultimapNmax **Purpose**: Maximum number of loci a read can map to

Default: 10

Behavior:

- Reads mapping to 10 loci: Kept (marked as multi-mappers)
- Reads mapping to >10 loci: Discarded

Why allow multi-mappers?:

Scenario	Multi-mapping Reads	Action
Gene families	HOX genes, immunoglobulins	Keep (10-20)
Repetitive elements	LINEs, SINEs	May discard

Scenario	Multi-mapping Reads	Action
Recent duplications	Paralogous genes	Keep
rRNA contamination	Very high mapping	May indicate problem

Recommendations:

```
# Gene expression (standard)
--outFilterMultimapNmax 10

# More permissive (gene families)
--outFilterMultimapNmax 20

# Unique-only (variant calling)
--outFilterMultimapNmax 1

# Very permissive (repetitive elements)
--outFilterMultimapNmax 100
```

Trade-offs:

- Too low (1): Lose 10-20% of gene family data
- Too high (>100): Include spurious alignments

Mismatch Filtering

--outFilterMismatchNmax **Purpose**: Maximum total mismatches allowed

Common settings:

```
--outFilterMismatchNmax 999    # Effectively unlimited (use percentage filter)
--outFilterMismatchNmax 10     # Hard limit of 10 mismatches
```

Recommendation: Use 999 and rely on percentage filter (below)

--outFilterMismatchNoverReadLmax **Purpose**: Maximum mismatches as fraction of read length

Formula: `mismatches / read_length  threshold`

Examples:

```
--outFilterMismatchNoverReadLmax 0.04    # 4% (default, recommended)
--outFilterMismatchNoverReadLmax 0.06    # 6% (more permissive)
--outFilterMismatchNoverReadLmax 0.02    # 2% (stringent)
```

Calculation for 0.04:

Read Length	Max Mismatches
50bp	2
75bp	3
100bp	4
150bp	6
250bp	10

Why percentage-based?:

- Accounts for read length differences
- Longer reads naturally have more errors
- Fair comparison across read lengths

--outFilterType BySJout **Purpose**: Filters reads based on splice junction confidence

How it works:

- Keeps reads with junctions that are:
  - **Annotated** (in GTF), OR
  - **Well-supported** (multiple reads)
- Discards reads with low-confidence novel junctions

Benefits:

- Reduces false novel junctions from errors
- Improves junction call quality

Trade-offs:

- May miss genuine rare splice variants
- More conservative

When to use:

- Gene-level expression analysis
- Standard RNA-seq QC
- Novel isoform discovery
- Rare splice variant detection

Alignment Strategy Parameters

--alignEndsType **Purpose**: How to handle read ends

Options:

Option	Behavior	Use Case
Local	Soft-clip poorly matching ends	RNA-seq (default)
EndToEnd	Require full read alignment	DNA-seq, stringent
Extend5pOfRead1	Special for specific protocols	Rare

Local soft-clipping example:

```
Read:     ACTGACTGACTGACTGNNNN
          ||||||||||||||||
Genome:   ACTGACTGACTGACTG----
Alignment: 16M4S (16 matched, 4 soft-clipped)
```

Why soft-clip?:

- Adapters at read ends
- Low-quality bases
- Non-genomic sequences

Recommendation: Use `Local` for RNA-seq

--alignIntronMin / --alignIntronMax **Purpose**: Valid intron size range

Defaults:

```
--alignIntronMin 21          # Minimum gap to call intron
--alignIntronMax 1000000     # Maximum intron size (1Mb)
```

Biological context:

Organism	Typical Range	Largest Intron
Human	100bp - 100kb	~800kb (DMD gene)
Mouse	100bp - 100kb	~500kb
Drosophila	50bp - 10kb	~100kb
C. elegans	50bp - 5kb	~20kb
Yeast	N/A (rare)	<500bp

Why filter by size?:

- Gaps <20bp: Likely deletions, not introns
- Gaps >1Mb: Likely misalignments across chromosomes

Custom settings:

```
# Yeast (small introns)
--alignIntronMin 10 --alignIntronMax 5000

# Human (allow large introns)
--alignIntronMin 20 --alignIntronMax 1000000

# Compact genome (C. elegans)
--alignIntronMin 20 --alignIntronMax 50000
```

--alignMatesGapMax **Purpose**: Maximum distance between paired-end mates

Default: 1000000 (1Mb, matches --alignIntronMax)

Use case: RNA-seq with large introns

For DNA-seq: Much smaller (~1000bp for 500bp fragments)

Unmapped Reads

--outSAMunmapped **Purpose**: What to do with unmapped reads

Options:

Option	Behavior	BAM Size	Use Case
None	Discard unmapped	Smaller	Save space
Within	Include in BAM	Larger	Troubleshooting
Within KeepPairs	Keep both mates if one unmapped	Larger	PE analysis

Recommendation: `Within` for troubleshooting, `None` for production

Output Files

1. Aligned.sortedByCoord.out.bam

Description: Main alignment file

Format: Binary BAM, sorted by genomic coordinates

Contents:

- Aligned reads with mapping positions
- CIGAR strings (alignment pattern)
- Mapping quality scores

- SAM flags (PE info, strand, etc.)

Size: 2-10GB per sample (human)

Uses:

- Visualization in IGV
- QC analysis (RSeQC, Qualimap)
- Variant calling
- Coverage analysis

Viewing:

```
# View header
samtools view -H sample.bam

# View first 10 alignments
samtools view sample.bam | head -n 10

# Count total reads
samtools view -c sample.bam

# Count mapped reads
samtools view -c -F 4 sample.bam
```

---

2. ReadsPerGene.out.tab

Description: Gene-level read counts

Format: Tab-separated, 4 columns

Structure:

```
N_unmapped          1000000  1000000  1000000
N_multimapping      500000   500000   500000
N_noFeature         300000   300000   300000
N_ambiguous         100000   100000   100000
ENSG00000000001     1000     1050     50
ENSG00000000002     500      25       480
```

First 4 lines (statistics):

- `N_unmapped`: Reads that didn't map
- `N_multimapping`: Multi-mapping reads
- `N_noFeature`: Mapped but not to any gene
- `N_ambiguous`: Mapped to multiple genes

Remaining lines: Gene counts

Use in differential expression:

```r
# Read data (skip statistics)
counts <- read.table("sample.ReadsPerGene.out.tab", skip=4)
colnames(counts) <- c("gene_id", "unstranded", "sense", "antisense")

# Extract column based on library type
final_counts <- counts[, "sense"]  # For stranded libraries
```

---

3. SJ.out.tab

Description: Splice junction table

Format: Tab-separated, 9 columns

Columns:

1. **Chromosome**: chr1, chr2, etc.
2. **Intron start** (1-based): First base of intron
3. **Intron end** (1-based): Last base of intron
4. **Strand**: 0 (undefined), 1 (+), 2 (-)
5. **Intron motif**:
    - 0: non-canonical
    - 1: GT/AG
    - 2: CT/AC
    - 3: GC/AG
    - 4: CT/GC
    - 5: AT/AC
    - 6: GT/AT
6. **Annotated**: 0 (novel), 1 (in GTF)
7. **Unique reads**: Count of uniquely mapping reads
8. **Multi-mapping reads**: Count of multi-mappers
9. **Maximum overhang**: Longest anchoring sequence

Example:

```
chr1  1000  2000  1  1  1  50   5   25
chr1  3000  4000  2  1  0  10   0   20
```

Interpretation:

- First junction: Annotated GT/AG on + strand, 50 unique reads
- Second junction: Novel GT/AG on - strand, 10 unique reads

Uses:

- Novel junction discovery
- Splice variant analysis
- Fusion gene detection
- Alternative splicing quantification

4. Log.final.out

Description: Alignment statistics summary

Format: Human-readable text

Key metrics:

```
                               Started job on |   Jan 22 10:00:00
                           Started mapping on |   Jan 22 10:05:00
                                  Finished on |   Jan 22 10:35:00
     Mapping speed, Million of reads per hour |   120.00

                        Number of input reads |   50000000
                    Average input read length |   101
                                  UNIQUE READS:
                 Uniquely mapped reads number |   40000000
                      Uniquely mapped reads % |   80.00%
                        Average mapped length |   100.50
                     Number of splices: Total |   15000000
                  Number of splices: Annotated |   14000000
                     Number of splices: GT/AG |   14500000
                     Number of splices: GC/AG |   400000
                     Number of splices: AT/AC |   50000
             Number of splices: Non-canonical |   50000
                    Mismatch rate per base, % |   0.30%
                       Deletion rate per base |   0.01%
                      Deletion average length |   1.50
                      Insertion rate per base |   0.01%
                     Insertion average length |   1.40
                           MULTI-MAPPING READS:
      Number of reads mapped to multiple loci |   8000000
           % of reads mapped to multiple loci |   16.00%
      Number of reads mapped to too many loci |   500000
           % of reads mapped to too many loci |   1.00%
                                UNMAPPED READS:
Number of reads unmapped: too many mismatches |   1000000
     % of reads unmapped: too many mismatches |   2.00%
          Number of reads unmapped: too short |   400000
               % of reads unmapped: too short |   0.80%
              Number of reads unmapped: other |   100000
                   % of reads unmapped: other |   0.20%
```

Quality thresholds:

Metric	Excellent	Good	Acceptable	Poor
Uniquely mapped	>80%	70-80%	60-70%	<60%
Multi-mapping	<10%	10-20%	20-30%	>30%
Unmapped	<10%	10-20%	20-30%	>30%
Mismatch rate	<0.5%	0.5-1%	1-2%	>2%

Resource Requirements

Memory (RAM)

Component	Requirement
STAR index in RAM	30GB (human)
BAM sorting	5-10GB
Buffer	5-10GB
Total	40-50GB

Scaling:

- Mouse: 25-30GB
- Drosophila: 5-8GB
- Yeast: 2-3GB

Memory optimization:

```
# Limit BAM sorting memory
--limitBAMsortRAM 10000000000    # 10GB

# Reduce sorting threads
--outBAMsortingThreadN 2

# Don't load index into shared memory
--genomeLoad NoSharedMemory
```

CPU Cores

Scaling:

- 4 cores: Baseline
- 8 cores: 1.7x faster
- 16 cores: 2.5x faster
- 32 cores: 3x faster (diminishing returns)

Recommendation: 8-16 cores optimal

Disk Space

Per sample (human):

- BAM file: 5-10GB
- Temporary files: 10-20GB
- Total: 15-30GB

Temporary space: 2-3x final BAM size

Time

Sample Size	8 cores	16 cores
20M reads	10 min	6 min
50M reads	25 min	15 min
100M reads	50 min	30 min

Troubleshooting

Low Mapping Rate (<50%)

Possible causes:

1. **Wrong reference genome**

   ```
   # Check index species
   cat star_index/genomeParameters.txt | grep genomeFastaFiles
   ```

2. **Contamination**

   - Run FastQ Screen
   - Check for bacterial/adapter sequences

3. **Degraded RNA**

   - Check RIN scores
   - Run FastQC for quality

4. **Wrong library type**

   - Ensure RNA-seq, not DNA-seq
   - Check protocol

High Multi-Mapping (>30%)

Causes:

1. **rRNA contamination**

   ```
   # Check for ribosomal RNA
   # High counts on rRNA genes
   ```

2. **Low complexity**

   - Check library prep
   - May need deeper sequencing

Solutions:

- Better rRNA depletion
- Check poly-A selection efficiency

Out of Memory

Error: ”EXITING: fatal error trying to allocate genome arrays”

Solutions:

1. **Increase RAM allocation**

2. **Reduce BAM sorting memory**

   ```
   --limitBAMsortRAM 5000000000  # 5GB
   ```

3. **Reduce sorting threads**

   ```
   --outBAMsortingThreadN 1
   ```

4. **Don't share genome**

   ```
   --genomeLoad NoSharedMemory
   ```

Slow Performance

If taking >1 hour per sample:

1. **Increase CPU cores**

   ```
   --runThreadN 16
   ```

2. **Use SSD storage**

   - Move working directory to SSD

3. **Disable two-pass mode** (if not needed)

   ```
   --twopassMode None
   ```

4. **Check I/O bottlenecks**

   ```
   iostat -x 1
   ```

Best Practices

Standard RNA-seq

```
STAR \
    --genomeDir star_index \
    --readFilesIn R1.fq.gz R2.fq.gz \
    --readFilesCommand zcat \
    --outFileNamePrefix sample. \
    --outSAMtype BAM SortedByCoordinate \
    --quantMode GeneCounts \
    --twopassMode Basic \
    --runThreadN 8
```

Fusion Detection

```
STAR \
    --genomeDir star_index \
    --readFilesIn R1.fq.gz R2.fq.gz \
    --readFilesCommand zcat \
    --outFileNamePrefix sample. \
    --outSAMtype BAM SortedByCoordinate \
    --chimSegmentMin 20 \
    --chimJunctionOverhangMin 20 \
    --chimOutType WithinBAM \
    --runThreadN 8
```

CircRNA Detection

```
STAR \
    --genomeDir star_index \
    --readFilesIn R1.fq.gz R2.fq.gz \
    --readFilesCommand zcat \
    --outFileNamePrefix sample. \
    --outSAMtype BAM SortedByCoordinate \
    --chimSegmentMin 20 \
    --chimJunctionOverhangMin 20 \
    --chimOutType Junctions \
```

```
    --chimScoreMin 1 \
    --runThreadN 8
```

Related Documentation

- **STAR Index**: `docs/star_index.md`
- **Gene Counting**: `docs/feature_counts.md`
- **Differential Expression**: `docs/deseq2.md`
- **STAR Manual**: https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

---

Document Version: 2.0
Last Updated: January 2026
STAR Version: 2.7.10a+
Applicable to: All RNA-seq applications requiring genome alignment