

RNA-Seq: GTF/GFF and Splice Aligner

BCB 5250 Introduction to Bioinformatics II

Spring 2020

Tae-Hyuk (Ted) Ahn

Department of Computer Science
Program of Bioinformatics and Computational Biology
Saint Louis University



SAINT LOUIS
UNIVERSITY™

— EST. 1818 —

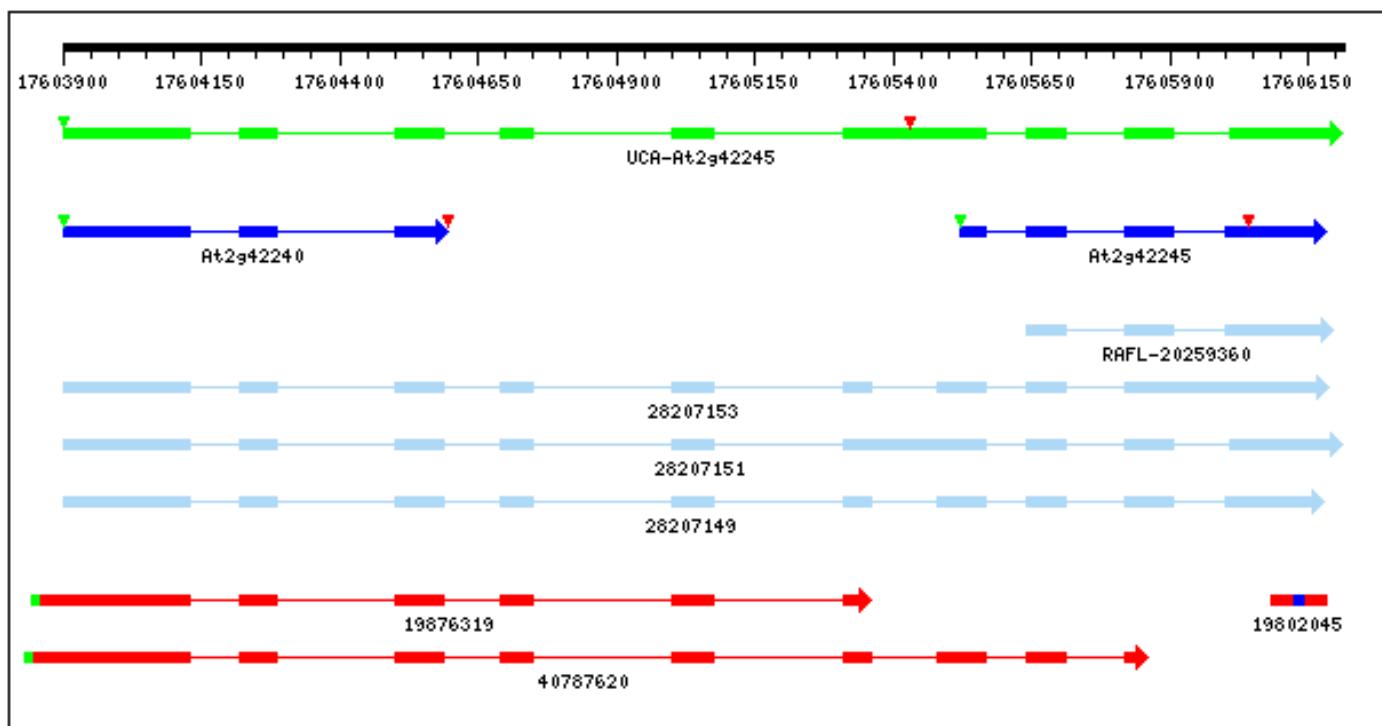
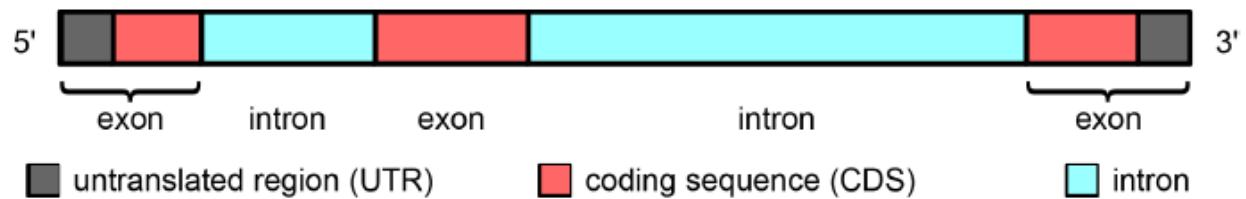
Genome sequence file: where to get?

- Illumina iGenomes (recommended)
 - http://support.illumina.com/sequencing/sequencing_software/igenome.html
- Ensembl
 - <http://ensemblgenomes.org/info/access/ftp>
- NCBI genome
 - <http://www.ncbi.nlm.nih.gov/genome/>
- Organism specific databases/websites.

e!Ensembl



Genome annotation file



GFF/GTF File Format - Definition

- The GFF (General Feature Format) format
 - one line per feature
 - each containing 9 columns of data
 - plus optional track definition lines.
- GFF has many versions (GFF, GFF2, GFF3)
- GTF (General Transfer Format) identical to GFF2.
- Most spliced aligner supports both GTF and GFF3 (mostly GTF)

GTF/GTF2 format

9 columns:

```
<seqname> <source> <feature> <start> <end> <score> <strand> <frame> [attributes]
```

- seqname - name of the chromosome or scaffold
- source – program or database that generated this feature.
- feature – Examples: "CDS", "gene", "transcript", and "exon".
- start - The starting position of the feature in the sequence.
- end - The ending position of the feature (inclusive).
- score - A score between 0 and 1000.
- strand – '+' (forward) or '-' (reverse) or '.' (don't know/don't care).
- Frame – reading frame '0', '1' or '2'
- attribute – A semicolon-separated list of tag-value pairs, providing additional information about each feature.

Example of GTF2 format

```
AB000381 Twinscan CDS      380    401    .    +    0    gene_id "001"; transcript_id "001.1";
AB000381 Twinscan CDS      501    650    .    +    2    gene_id "001"; transcript_id "001.1";
AB000381 Twinscan CDS      700    707    .    +    2    gene_id "001"; transcript_id "001.1";
AB000381 Twinscan start_codon 380    382    .    +    0    gene_id "001"; transcript_id "001.1";
AB000381 Twinscan stop_codon 708    710    .    +    0    gene_id "001"; transcript_id "001.1";
```

A simple example with 3 translated exons. Order of rows is not important.

Some annotation sources (e.g. Ensembl) add the gene_name attribute

```
gene_id "ENSBTAG00000020601"; transcript_id "ENSBTAT00000027448"; gene_name "ZNF366";
```

<http://mblab.wustl.edu/GTF2.html>

Generic Feature Format Version 3 (GFF3)

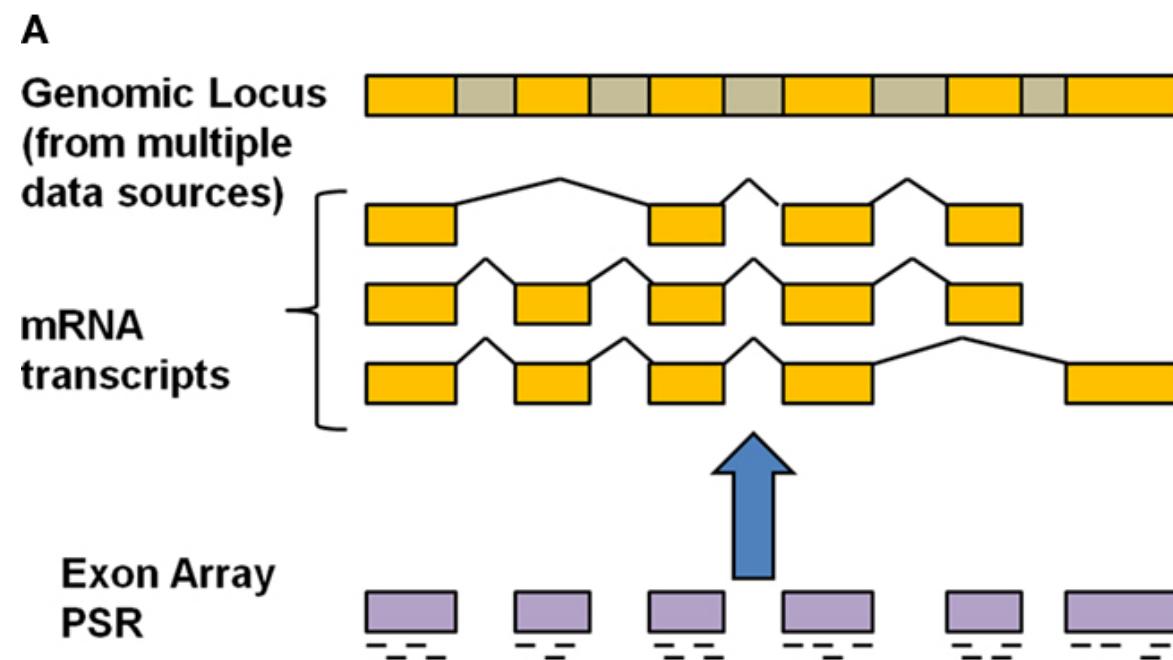
```
##gff-version 3.2.1
##sequence-region ctg123 1 1497228
ctg123 . gene    1000 9000 . + . ID=gene00001;Name=EDEN
ctg123 . TF_binding_site 1000 1012 . + . ID=tfbs00001;Parent=gene00001
ctg123 . mRNA    1050 9000 . + . ID=mRNA00001;Parent=gene00001;Name=EDEN
ctg123 . mRNA    1050 9000 . + . ID=mRNA00002;Parent=gene00001;Name=EDEN
ctg123 . mRNA    1300 9000 . + . ID=mRNA00003;Parent=gene00001;Name=EDEN
ctg123 . exon    1300 1500 . + . ID=exon00001;Parent=mRNA00003
ctg123 . exon    1050 1500 . + . ID=exon00002;Parent=mRNA00001,mRNA00002
ctg123 . exon    3000 3902 . + . ID=exon00003;Parent=mRNA00001,mRNA00002
ctg123 . exon    5000 5500 . + . ID=exon00004;Parent=mRNA00001,mRNA00002
ctg123 . exon    7000 9000 . + . ID=exon00005;Parent=mRNA00001,mRNA00002
ctg123 . CDS     1201 1500 . + 0 ID=cds00001;Parent=mRNA00001;Name=edEN
ctg123 . CDS     3000 3902 . + 0 ID=cds00001;Parent=mRNA00001;Name=edEN
ctg123 . CDS     5000 5500 . + 0 ID=cds00001;Parent=mRNA00001;Name=edEN
ctg123 . CDS     7000 7600 . + 0 ID=cds00001;Parent=mRNA00001;Name=edEN
ctg123 . CDS     1201 1500 . + 0 ID=cds00002;Parent=mRNA00002;Name=edEN
ctg123 . CDS     5000 5500 . + 0 ID=cds00002;Parent=mRNA00002;Name=edEN
ctg123 . CDS     7000 7600 . + 0 ID=cds00002;Parent=mRNA00002;Name=edEN
ctg123 . CDS     3301 3902 . + 0 ID=cds00003;Parent=mRNA00003;Name=edEN
ctg123 . CDS     5000 5500 . + 1 ID=cds00003;Parent=mRNA00003;Name=edEN
ctg123 . CDS     7000 7600 . + 1 ID=cds00003;Parent=mRNA00003;Name=edEN
ctg123 . CDS     3391 3902 . + 0 ID=cds00004;Parent=mRNA00003;Name=edEN
ctg123 . CDS     5000 5500 . + 1 ID=cds00004;Parent=mRNA00003;Name=edEN
ctg123 . CDS     7000 7600 . + 1 ID=cds00004;Parent=mRNA00003;Name=edEN
```

GFF3 adds parent feature

<http://www.sequenceontology.org/gff3.shtml>

GFF2 vs GFF3

- GFF2
 - two-level hierarchies *transcript* → *exon*
- GFF3
 - three-level hierarchy of *gene* → *transcript* → *exon*



<http://gmod.org/wiki/GFF2>

Conversion GFF3 To GTF

- Optional
- Use `gffread` (comes with the Cufflinks software suite)

```
$ gffread my.gff3 -T -o my.gtf
```

- See `gffread -h` for more information

Download Prepare GTF/GFF file

- Download it from Illumina's iGenomes project (for model species)
 - http://support.illumina.com/sequencing/sequencing_software/igenome.html

Or

- Download gff3 files from genome database,
 - Ensembl genome
 - <http://ensemblgenomes.org/>

Splice Aligner

- The alignment process consists of choosing an appropriate reference genome to map our reads against and performing the read alignment using one of several splice-aware alignment tools such as [STAR](#) or [HISAT2](#).
- The choice of aligner is often a personal preference and also dependent on the computational resources that are available to you.

One Critique from a Student

A critique of this paper, which must do more with the field of bioinformatics itself, is that there are so many methods and tools available to basically do the same thing. How does one reconcile this? What I usually do is to see which tools have been highly cited. This gives me a good baseline as to whether I can trust the output of a program. I also like to use well-documented tools. Using a ‘black box’ is something I really don’t like to do, because I have no way of analyzing the results or understanding how the results were derived. I can only use another similar tool, and compare those results, but then, I might as well have used the other tool to begin with. In sum, I think this field, while very mature, is still exciting and growing. Especially with the advent of third-generation sequencing.

HISAT2 V.S. STAR

- <https://www.biostars.org/p/288726/>



11

The main benefit of hisat2 is that it uses fewer resources than STAR and that it can better handle known SNPs if you make the aligner aware of them. Aside from that, I essentially always get better results from STAR, which is why we use it in our standard pipelines instead of hisat2 (this also bears out in published comparisons).

[ADD COMMENT](#) • [link](#)

written 2.2 years ago by [Devon Ryan](#) ♦ 94k



2.2 years ago by
[Devon Ryan](#) ♦ 94k
Freiburg, Germany



thanks! I will then stick to STAR and DESEP2 method. One more question is if i use DESEQ2, but i actually want to know the abundance of my transcript, in this case would you still recommend using RPKM from Cuffdiff? or other methods?

[ADD REPLY](#) • [link](#)

written 2.2 years ago by [langya](#) • 60



You can get normalized counts from DESeq2, so either use them directly or convert them to FPKMs or just divide by transcript length if you want some sort of length-normalized value.

[ADD REPLY](#) • [link](#)

written 2.2 years ago by [Devon Ryan](#) ♦ 94k



For reference, here is a published comparison: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/>

[ADD REPLY](#) • [link](#)

written 11 months ago by [takeshi](#) • 0



- Based on their latest respective releases, do you still consider STAR better than HISAT2?
- What are the qualifications behind this assertion, are they different from before?

[ADD REPLY](#) • [link](#)

written 7 months ago by [Anand Rao](#) • 250



- yes
- At least STAR has added a plethora of new features

1

[ADD REPLY](#) • [link](#)

written 7 months ago by [Devon Ryan](#) ♦ 94k

Another Question from a Student

I do not fully understand how to resolve the difficulty posed by various splicing alternative and the resultant isoforms when using short Illumina reads. If the reads are not long enough to span across various exon boundaries, how can such splicing be deduced? PE reads may help eliminate some ambiguity, but I don't understand how PE reads will map back to the transcriptome with alternative splicings.

Also, with regard to single-cell, I would be interested to know how much variation exists in the RNA content among different cells of a single tissue sample.

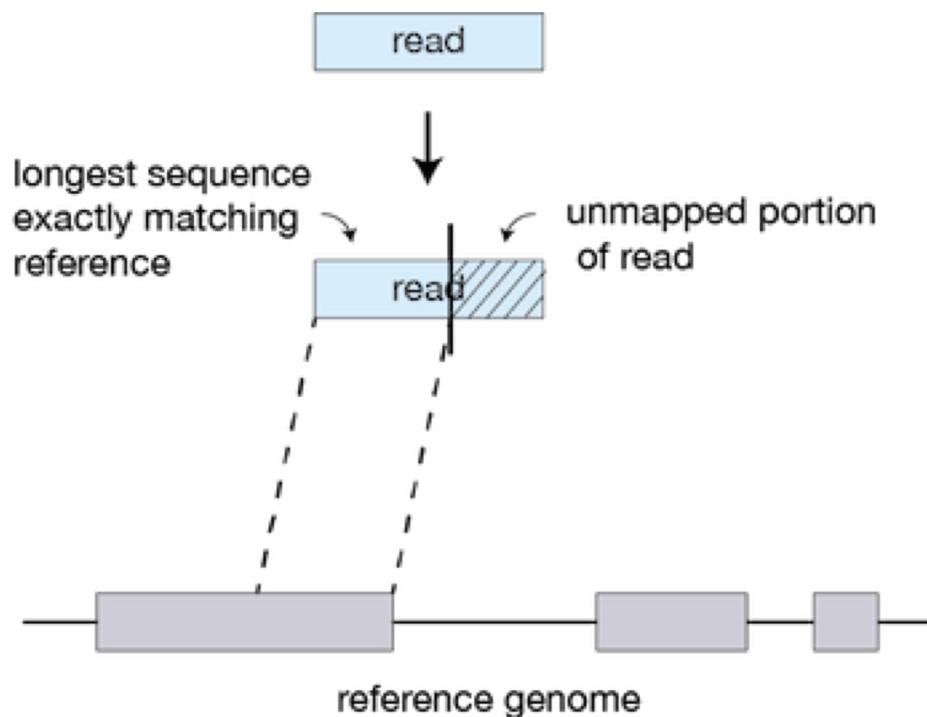
STAR Alignment Strategy

STAR is shown to have high accuracy and outperforms other aligners by more than a factor of 50 in mapping speed, but it is memory intensive. The algorithm achieves this highly efficient mapping by performing a two-step process:

- Seed searching
- Clustering, stitching, and scoring

Seed searching

- For every read that STAR aligns, STAR will search for the longest sequence that exactly matches one or more locations on the reference genome. These longest matching sequences are called the Maximal Mappable Prefixes (MMPS):



https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html

Seed searching

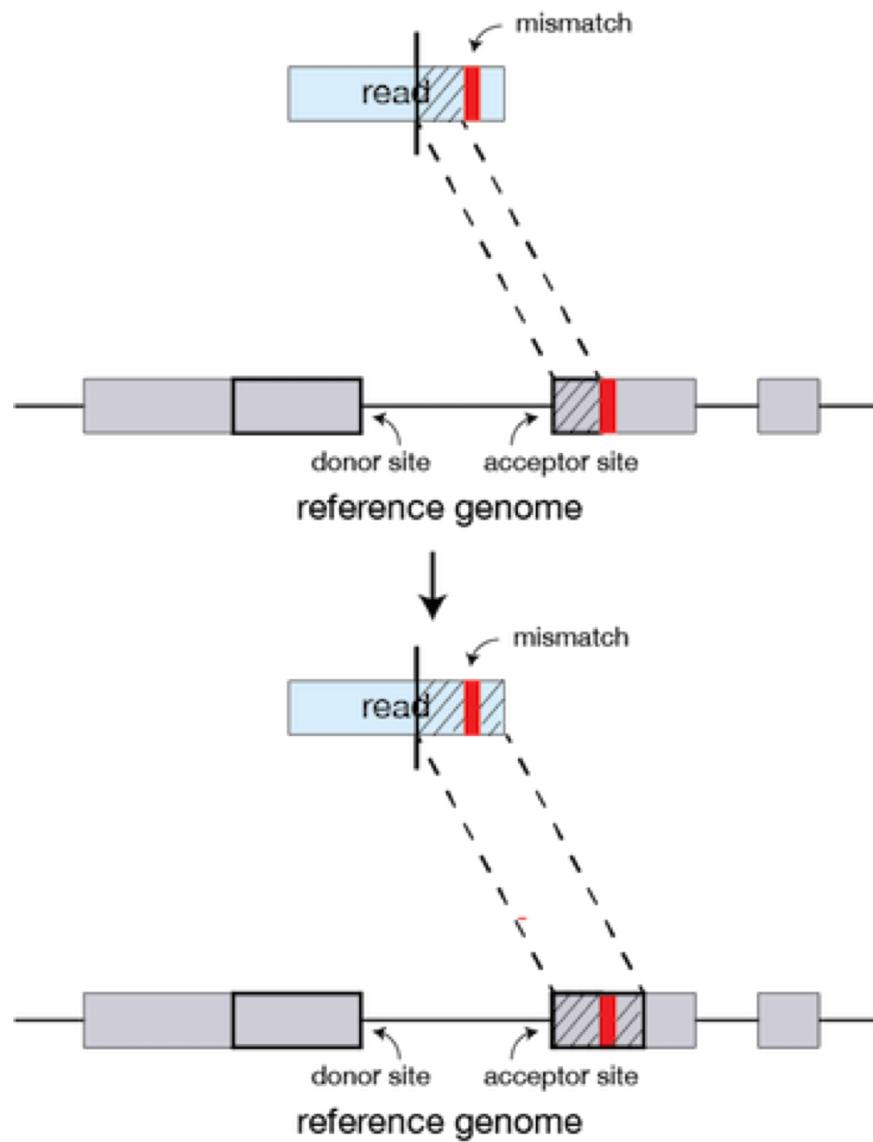
- The different parts of the read that are mapped separately are called ‘seeds’. So the first MMP that is mapped to the genome is called seed1.
- STAR will then search again for only the unmapped portion of the read to find the next longest sequence that exactly matches the reference genome, or the next MMP, which will be seed2.



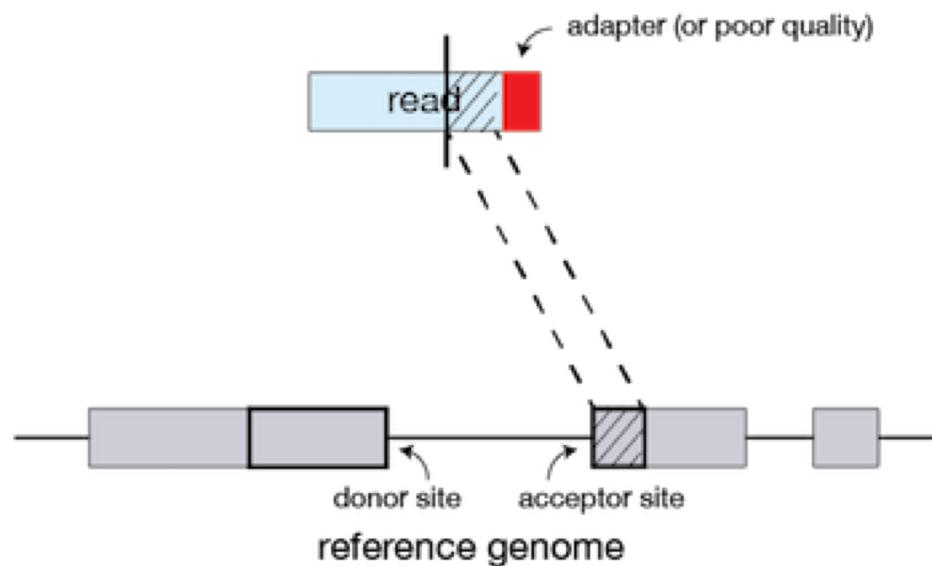
This sequential searching of only the unmapped portions of reads underlies the efficiency of the STAR algorithm. STAR uses an uncompressed suffix array (SA) to efficiently search for the MMPs, this allows for quick searching against even the largest reference genomes. Other slower aligners use algorithms that often search for the entire read sequence before splitting reads and performing iterative rounds of mapping.

https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html

Best Mapping

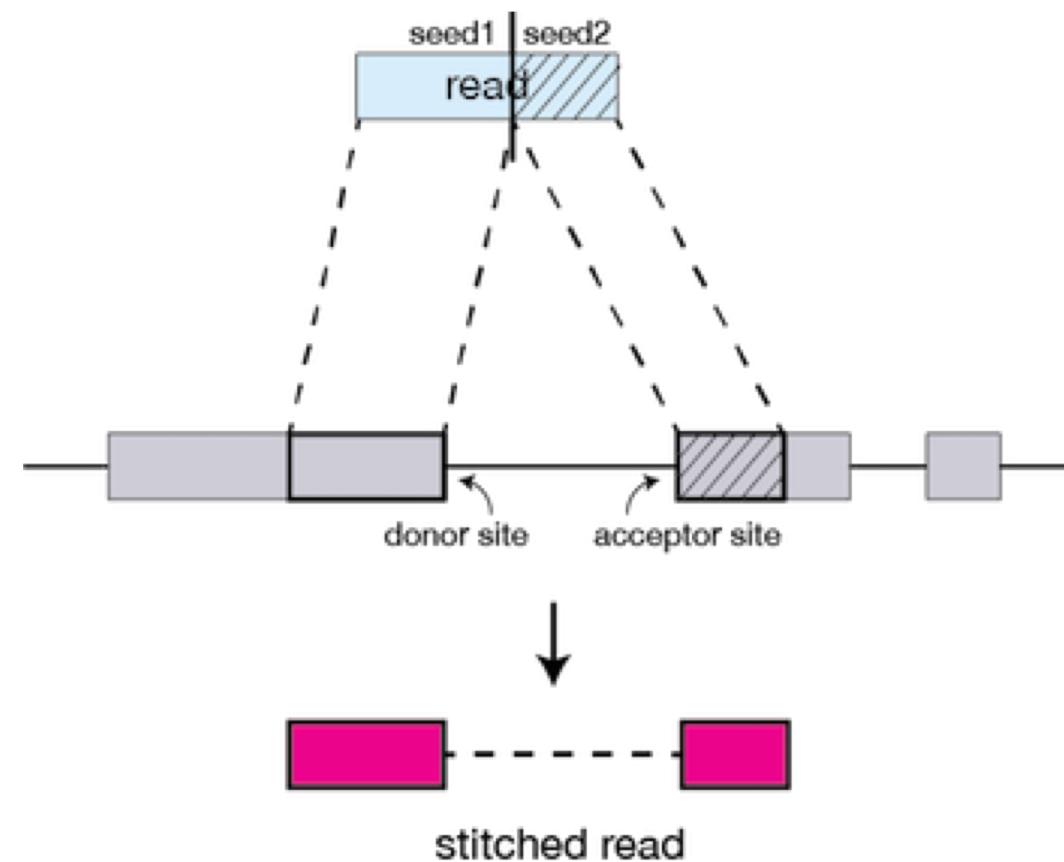


- If STAR does not find an exact matching sequence for each part of the read due to mismatches or indels, the previous MMPs will be extended.
- If extension does not give a good alignment, then the poor quality or adapter sequence (or other contaminating sequence) will be soft clipped.



https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html

Clustering, stitching, and scoring



- The separate seeds are stitched together to create a complete read by first clustering the seeds together based on proximity to a set of 'anchor' seeds, or seeds that are not multi-mapping.
- Then the seeds are stitched together based on the best alignment for the read (scoring based on mismatches, indels, gaps, etc.).

https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html