RNA-Seq: Quantification

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

RNA-Seq Data Analysis Protocol

- Quality Control
- Read mapping or alignment
- Quantification
- Differential gene expression analysis

Software for Quantification

Alignment-based

- HTSeq
- Cufflinks2
- StringTie
- RSEM

Alignment-free (Alignment-independent)

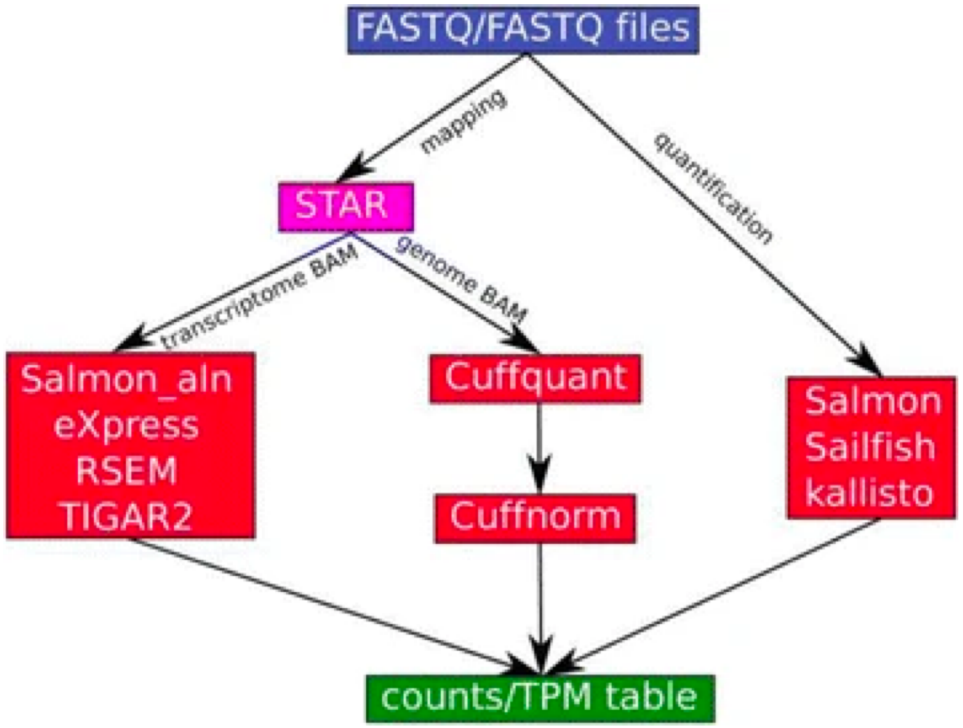
Pseudo-alignment

A classification based RNA-Seq

(Instead of using a genome as a reference, each read is classified as assigned a single transcript.)

- Sailfish
- Salmon
- Kallisto

Possible Workflow



- Workflow for transcript isoform quantification. Sequencing reads were either mapped by STAR aligner or directly fed into alignment-free methods, Salmon, Sailfish or Kallisto. The transcriptome BAM files were quantified by Salmon_aln, eXpress, RSEM or TIGAR2. The genome BAM files were quantified by Cuffquant and then Cuffnorm from the Cufflinks package. The results are summarized into counts and TPM tables for comparison

https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-017-4002-1

Research article | Open Access | Published: 07 August 2017

Evaluation and comparison of computational tools for RNA-seq isoform quantification

Chi Zhang, Baohong Zhang, Lih-Ling Lin & Shanrong Zhao

BMC Genomics **18**, Article number: 583 (2017) | Cite this article

18k Accesses | **30** Citations | **55** Altmetric | Metrics

Counting reads per genes with HTSeq

- Given a BAM file and a list of gene locations, counts how many reads map to each gene.
  - A gene is considered as the union of all its exons.
  - Reads can be counted also per exons.
- Locations need to be supplied in GTF file
  - Note that GTF and BAM must use the same chromosome naming
- Multimapping reads and ambiguous reads are not counted
- 3 modes to handle reads which overlap several genes
  - Union (default), Intersection-strict, Intersection-nonempty
- Attention: was your data made with stranded protocol?
  - You need to select the right counting mode!

> *Important:* The default for strandedness is *yes*. If your RNA-Seq data has not been made with a strand-specific protocol, this causes half of the reads to be lost. Hence, make sure to set the option `--stranded=no` unless you have strand-specific data!

htseq-count

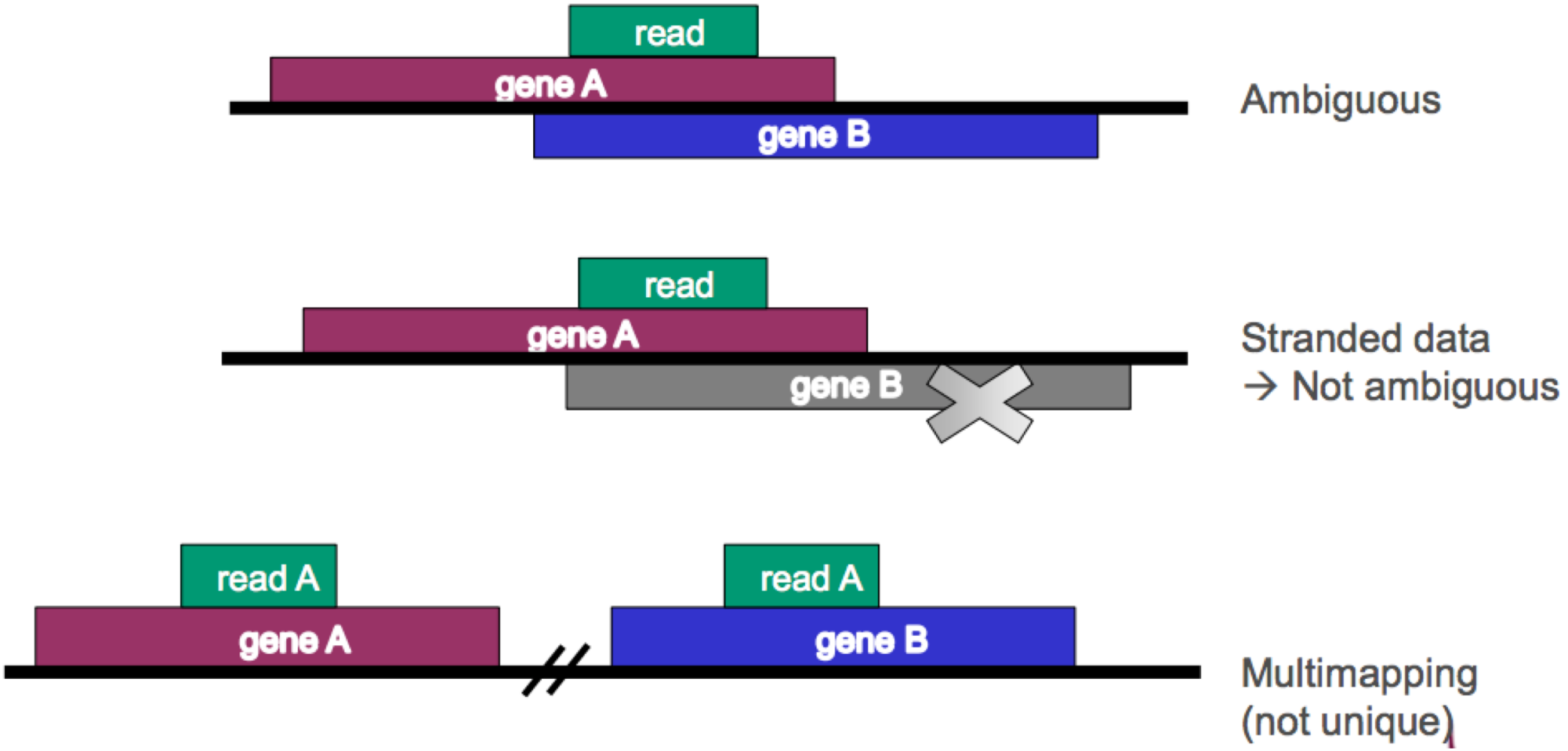
- The script *htseq-count* is a tool for RNA-Seq data analysis: Given a SAM/BAM file and a GTF or GFF file with gene models, it counts for each gene how many aligned reads overlap its exons.
- These counts can then be used for gene-level differential expression analyses using methods such as *DESeq2* or *edgeR*.

htseq-count

The three overlap resolution modes of htseq-count work as follows. For each position *i* in the read, a set *S(i)* is defined as the set of all features overlapping position *i*. Then, consider the set *S*, which is (with *i* running through all position within the read or a read pair)

- the union of all the sets *S(i)* for mode union. This mode is recommended for most use cases.
- the intersection of all the sets *S(i)* for mode intersection-strict.
- the intersection of all non-empty sets *S(i)* for mode intersection-nonempty.

Not unique or ambiguous?



https://chipster.csc.fi/material/rna/RNAseqDataAnalysis2019.pdf

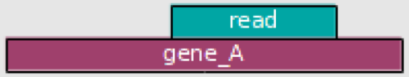
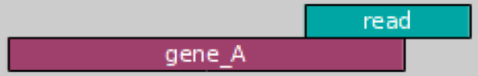
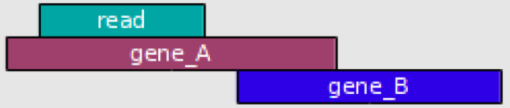
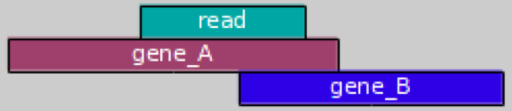
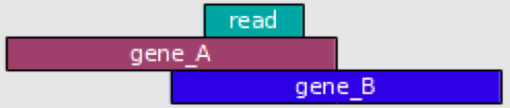
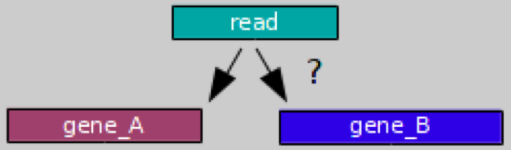
- https://htseq.readthedocs.io/en/release_0.11.1/count.html

If *S* contains precisely one feature, the read (or read pair) is counted for this feature. If *S* is empty, the read (or read pair) is counted as `no_feature`. If *S* contains more than one feature, `htseq-count` behaves differently based on the `--nonunique` option:

- `--nonunique none` (default): the read (or read pair) is counted as `ambiguous` and not counted for any features. Also, if the read (or read pair) aligns to more than one location in the reference, it is scored as `alignment_not_unique`.
- `--nonunique all`: the read (or read pair) is counted as `ambiguous` and is also counted in all features to which it was assigned. Also, if the read (or read pair) aligns to more than one location in the reference, it is scored as `alignment_not_unique` and also separately for each location.

Notice that when using `--nonunique all` the sum of all counts will not be equal to the number of reads (or read pairs), because those with multiple alignments or overlaps get scored multiple times.

The following figure illustrates the effect of these three modes and the `--nonunique` option:

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

htseq-count

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4287950/

We distribute two stand-alone scripts with HTSeq, which can be used from the shell command line, without any Python knowledge, and also illustrate potential applications of the HTSeq framework. The script *htseq-qa* is a simple tool for initial quality assessment of sequencing runs. It produces plots that summarize the nucleotide compositions of the positions in the read and the base-call qualities.

The script *htseq-count* is a tool for RNA-Seq data analysis: Given a SAM/BAM file and a GTF or GFF file with gene models, it counts for each gene how many aligned reads overlap its exons. These counts can then be used for gene-level differential expression analyses using methods such as *DESeq2* (Love *et al.*, 2014) or *edgeR* (Robinson *et al.*, 2010). As the script is designed specifically for *differential* expression analysis, only reads mapping unambiguously to a single gene are counted, whereas reads aligned to multiple positions or overlapping with more than one gene are discarded. To see why this is desirable, consider two genes with some sequence similarity, one of which is differentially expressed while the other one is not. A read that maps to both genes equally well should be discarded, because if it were counted for both genes, the extra reads from the differentially expressed gene may cause the other gene to be wrongly called differentially expressed, too. Another design choice made with the downstream application of differential expression testing in mind is to count fragments, not reads, in case of paired-end data. This is because the two mates originating from the same fragment provide only evidence for one cDNA fragment and should hence be counted only once.

As the *htseq-count* script has found widespread use over the past 3 years, we note that we recently replaced it with an overhauled version, which now allows processing paired-end data without the need to sort the SAM/BAM file by read name first. See the documentation for a list of all changes to the original version.

Lap Recap: Build Star Index

- Download STAR aligner
- Check the STAR manual
- Map all reads to the reference
  - Build the index
    - Recommendation:
    - $ mkdir star_indexes
    - $ STAR --runThreadN 12 --runMode genomeGenerate --genomeDir star_indexes --genomeFastaFiles genome/chrX.fa --sjdbGTFfile genes/chrX.gtf --sjdbOverhang 74 --genomeSAindexNbases 12)
  - Create a shell script to align all reads to once
    - Recommendation:
    - $ STAR --genomeDir star_indexes --runThreadN 6 --readFilesIn XXX.fq --outFileNamePrefix star_align/XXX --outSAMtype BAM SortedByCoordinate --outSAMunmapped Within --outSAMattributes Standard

Lap Recap: Base Shell Example

```
#!/bin/bash
dir="chrX_data"
for entry in $dir"/samples/"*1.fastq.gz
do
   name=$(echo $entry | egrep -o "ERR[^_]+")
   hisat2 -p 8 --dta -x "$dir"/indexes/chrX_tran -1 "$dir"/samples/"$name"_chrX_1.fastq.gz -
2 "$dir"/samples/"$name"_chrX_2.fastq.gz -S "$dir"/map/"$name"_chrX.sam
done
```

run_star.sh

```
[ahnt@hopper:~/Course/bcb5250/2020S/labs/rna_seq/chrX_data]$ cat run_star.sh
#!/bin/bash
dir=/faculty/ahnt/Course/bcb5250/2020S/labs/rna_seq/chrX_data
files=$dir/samples/*1.fastq.gz
for entry in $files
do
    basename=$(echo $entry|egrep -o "ERR[^_]+")
    STAR --genomeDir star_indexes --runThreadN 12 --readFilesCommand zcat --readFilesIn
"$dir"/samples/"$basename"_chrX_1.fastq.gz "$dir"/samples/"$basename"_chrX_2.fastq.gz --
outFileNamePrefix "$dir"/star_align/"$basename"_chrX_ --outSAMtype BAM SortedByCoordinat
e --outSAMunmapped Within --outSAMattributes Standard
done
```

run htseq-count

- One example to run:
  - Go to the star_align directory
  - Run as below:

$ htseq-count -f bam ERR188044_chrX_Aligned.sortedByCoord.out.bam ../genes/chrX.gtf > ERR188044_chrX_htseq.out

```
NR_125730   0
NR_126059   72
NR_126093   0
NR_126094   46
NR_126350   0
NR_126557   1
NR_126564   0
NR_130172   0
NR_130179   0
NR_130469   0
NR_130638   0
NR_130730   2
NR_130733   39
NR_130768   0
NR_130770   4
NR_131204   0
NR_131236   3
NR_131237   0
NR_131238   0
NR_131250   0
NR_131982   4
NR_132337   0
NR_132647   0
NR_132964   0
NR_133641   0
__no_feature    1448685
__ambiguous 446972
__too_low_aQual 0
__not_aligned   24119
__alignment_not_unique  30798
```

run htseq-count

Make your own script to run htseq-count with all BAM files