NGS pipeline from raw fastq to count data

Trimmomatic is a flexible read trimming tool for Illumina NGS data. It is used to trim and crop Illumina (FASTQ) data as well as to remove adapters. Below are the steps on how to use Trimmomatic for paired-end data:

1. **Download and Install Trimmomatic**:
   * Trimmomatic is a Java program, so you will need to have Java installed on your computer.
   * Download the latest Trimmomatic jar file from the Trimmomatic website.
2. **Prepare Adapter Sequence File**:
   * Trimmomatic requires a FASTA file of adapter sequences. This file is often provided with the Trimmomatic download or you can create your own based on the adapters used in your sequencing run.
3. **Run Trimmomatic**:
   * To run Trimmomatic, you need to specify the input files (paired-end data), the output files, and the desired trimming steps and parameters.
   * Here’s an example command for paired-end data trimming:
4. java -jar trimmomatic-0.39.jar PE \
5. -phred33 \
6. [input\_forward.fastq] [input\_reverse.fastq] \
7. [output\_forward\_paired.fastq] [output\_forward\_unpaired.fastq] \
8. [output\_reverse\_paired.fastq] [output\_reverse\_unpaired.fastq] \
9. ILLUMINACLIP:[adapter\_file.fa]:2:30:10 \
10. LEADING:3 \
11. TRAILING:3 \
12. SLIDINGWINDOW:4:15 \

MINLEN:36

java -jar trimmomatic-0.39.jar PE \

-phred33 \

[input\_forward.fastq] [input\_reverse.fastq] \

[output\_forward\_paired.fastq] [output\_forward\_unpaired.fastq] \

[output\_reverse\_paired.fastq] [output\_reverse\_unpaired.fastq] \

ILLUMINACLIP:[adapter\_file.fa]:2:30:10 \

LEADING:3 \

TRAILING:3 \

SLIDINGWINDOW:4:15 \

MINLEN:36

To test HISAT2 and ensure it's working correctly on your system, you can run a basic alignment using sample data. If you don't have your own FASTQ files ready for testing, you might need to download example data or use very minimal test data. Here's how you can perform a simple test:

### Step 1: Download Example Data (Optional)

If you need test data, here’s how you could download a small example dataset. Since I can't directly download files or access external databases, I'll provide you with a generic command to download a small FASTQ file from a public repository. Please replace the URL with a valid one that points to actual FASTQ files.

curl -O https://example.com/sample\_1.fastq

curl -O https://example.com/sample\_2.fastq curl -O https://example.com/sample\_1.fastq curl -O https://example.com/sample\_2.fastq

### Step 2: Create a Test Index

Before aligning, you need an index of the reference genome. For testing purposes, you can create a small, fake reference genome. Here's how to create a simple text file to simulate a very small reference genome:

1. **Create a Fake Reference Genome**:

echo ">chr1" > test\_ref.fa

echo "NNNNNNNNNNNNNNNNNNNN" >> test\_ref.fa >> test\_ref.fa

1. **Build HISAT2 Index**:

hisat2-build test\_ref.fa test\_ref\_index

hisat2-build test\_ref.fa test\_ref\_index

This command creates an index named **test\_ref\_index** based on your fake reference genome **test\_ref.fa**.

### Step 3: Run HISAT2

Now, you can run HISAT2 using your test data (or the example data you downloaded). If you're using paired-end data, your command might look like this:

hisat2 -x test\_ref\_index -1 sample\_1.fastq -2 sample\_2.fastq -S output.sam hisat2 -x test\_ref\_index -1 sample\_1.fastq -2 sample\_2.fastq -S output.sam

Replace **sample\_1.fastq** and **sample\_2.fastq** with the paths to your actual FASTQ files. If you're using single-end data, the command would instead look something like this:

hisat2 -x test\_ref\_index -U sample.fastq -S output.samhisat2 -x test\_ref\_index -U sample.fastq -S output.sam

### Step 4: Check the Output

After running HISAT2, you should have an output file named **output.sam**. You can check the contents of this file to confirm that HISAT2 ran successfully:

head output.sam output.sam

This command will display the first few lines of the SAM file, which should include a header section (lines starting with **@**) and possibly some alignments.

### Note:

This test is very basic and designed to simply verify that HISAT2 can run to completion. Real data and a complete reference genome are necessary for meaningful biological analysis. The example data and reference genome provided here are for demonstration purposes only and won't yield biologically relevant results.

With Samtools installed, you're ready to perform a wide range of tasks such as viewing, converting, and manipulating alignments. Here are a few basic commands to get started:

* **View a SAM/BAM File Header**:

samtools view -H your\_alignment\_file.bamsamtools view -H your\_alignment\_file.bam

* **Convert SAM to BAM**:

samtools view -bS your\_alignment\_file.sam > your\_alignment\_file.basamtools view -bS your\_alignment\_file.sam > your\_alignment\_file.bam

* **Sort a BAM File**:

samtools sort your\_alignment\_file.bam -o your\_sorted\_alignment\_file.bamsamtools your\_alignment\_file.bam -o your\_sorted\_alignment\_file.bam

* **Index a BAM File**:

samtools index your\_sorted\_alignment\_file.bam samtools index your\_sorted\_alignment\_file.bam

These commands are just the beginning of what you can do with Samtools. For more advanced usage and options, refer to the Samtools documentation or use **samtools --help** to explore the available commands and their parameters.

# Path to your FASTQ files (assuming a naming convention like sample1\_1.fastq and sample1\_2.fastq for paired-end files)

FASTQ\_DIR="/Users/seidmuhie/Documents/02052024\_RNAseqDataFolder/RNAseq\_pipeline/ arun\_fastq\_files"

# Path to save the trimmed FASTQ files

TRIMMED\_DIR="/path/to/trimmed\_arun\_fastq\_files"

mkdir -p "$TRIMMED\_DIR"

# Path to save the FastQC reports

FASTQC\_DIR="/path/to/fastqc\_reports"

mkdir -p "$FASTQC\_DIR"

# Path to the Trimmomatic adapter sequences file

ADAPTERS\_PATH="/path/to/trimmomatic/adapters/TruSeq3-PE.fa"

# Set variables

GENOME\_URL="http://example.com/ferret\_genome.fa"

GENOME\_FA="ferret\_genome.fa"

GENOME\_INDEX="ferret\_genome\_index"

ANNOTATION\_URL="http://example.com/ferret\_annotation.gff"

ANNOTATION\_FILE="ferret\_annotation.gff"

FASTQ\_DIR="/path/to/fastq\_files"

OUTPUT\_DIR="/path/to/output"

ALIGN\_DIR="${OUTPUT\_DIR}/alignments"

COUNTS\_DIR="${OUTPUT\_DIR}/counts"

"Poor Per tile sequence quality" – issues with the sequencing run for specific tiles or areas of the flow cell

Mitigation:

1. Assess the quality scores across all bases in all reads.
2. Determine if low-quality scores are isolated to specific tiles.
3. Consider re-sequencing if a significant portion of the data is affected.
4. If only a small portion is affected and it does not impact the overall dataset significantly, we might choose to proceed with caution, perhaps trimming or filtering out the low-quality reads or bases.

"Poor Per Base Sequence Content" – indicates an imbalance in the distribution of nucleotide frequencies across the length of reads. This can be due to sequencing bias, contamination, or library preparation issues.

Mitigation:

1. this may involve reviewing library preparation protocols,
2. considering bioinformatic adjustments (like trimming for specific adapters or contaminants), or
3. re-sequencing if the issue significantly impacts data quality or analysis outcomes.

Online link

<https://www.elucidata.io/blog/bulk-rna-sequencing-a-comparison-of-the-most-popular-tools-and-pipelines>

To perform an alignment using the HISAT2 pipeline for the Ferret genome, need to follow these general steps:

1. **Download the Ferret genome sequence**: Obtain the reference genome sequence for the Ferret. You can usually find this on genome databases or through specific Ferret genome projects.
2. **Indexing the genome**: Use HISAT2 to index the Ferret genome. This step is crucial for alignment.

*hisat2-build <ferret\_genome.fa> ferret\_index*

hisat2-build <ferret\_genome.fa> ferret\_index

Replace <ferret\_genome.fa> with the filename of the Ferret genome sequence.

1. **Quality control (optional)**: Perform quality control on your sequencing data if necessary, using tools like FastQC.
2. **Align reads to the indexed genome**: Use HISAT2 to align your sequencing reads to the indexed Ferret genome.

*hisat2 -x ferret\_index -U <reads.fastq> -S alignment.sam*

hisat2 -x ferret\_index -U <reads.fastq> -S alignment.sam

Replace <reads.fastq> with the filename of your sequencing reads.

1. **Convert SAM to BAM and sort**: Convert the SAM file to BAM format and sort it for further analysis.

*samtools view -bS alignment.sam | samtools sort -o alignment\_sorted.bam -*

samtools view -bS alignment.sam | samtools sort -o alignment\_sorted.bam -

1. **Index the sorted BAM file**: Index the sorted BAM file for downstream analysis.

*samtools index alignment\_sorted.bam*

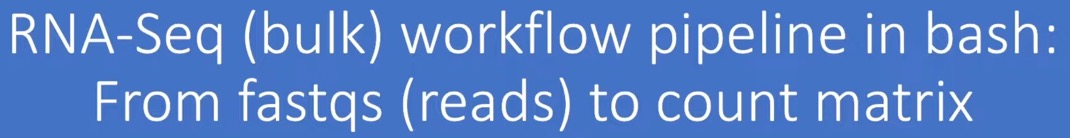
samtools index alignment\_sorted.bam

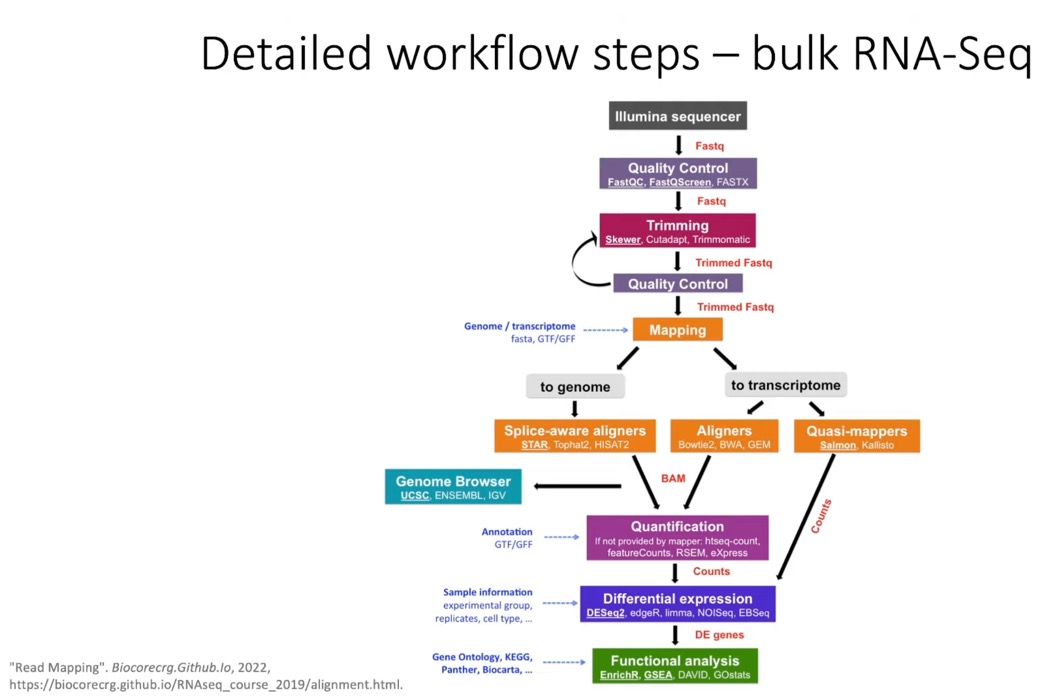
1. **Further analysis**: You can perform various downstream analyses such as variant calling, gene expression quantification, etc., depending on your research goals.

Remember to adjust parameters and filenames according to your specific data and requirements. Additionally, ensure you have the necessary dependencies installed and available in your environment before running these commands.

<https://www.youtube.com/watch?v=lG11JjovJHE>

Taken from the above youtube channel





Quality check – FastQC

For alignment – HiSat2

Samtools for indexing

Counting – cufflinks

Differential analysis – BaySeq

HISat2-HTseq (pipeline for alignment and mapping) is the most efficient; and the baySeq is the most accurate (for differential genes)

## Shell scripts

are **text files that contain commands we want to run**.

And usually have the extension .sh

we can **run all operations** any time by typing **one single command**

Let’s write a shell script that will do two things:

1. Tell us our current working directory
2. List the contents of the directory

First open a new file using vim:

% vim listing.sh

Then type in the following lines in the listing.sh file:

echo "Your current working directory is:"

pwd

echo "These are the contents of this directory:"

ls -l

the echo commands are helpful in letting you know what came next

To run the new shell script, use the bash or sh command.

% sh listing.sh

Bash Variables

Let’s start with a simple variable that has a single number stored in it:

% num=25

How do we know that we actually created the bash variable? We can use the echo command to print to terminal:

% echo num

**when trying to retrieve the value stored in the variable, we explicitly use a $ in front of it**:

% echo $num

Sheband is a statement on the first line of a script that tells the computer which programme to run it with. In this case, you should need to have #!/bin/bash on line 1 of myscript.sh, which indicates that you intend to use the ‘bash’ program located in your ‘bin’ folder

#!/bin/bash

make the script ‘executable’ by running chmod +x myscript.sh on the terminal

This gives your computer permission to run it directly and so it can run with ./myscript.sh, which saves a few keystrokes.

A script can be run from another script using the source command, eg having source myscript.sh in a different script file inside the same folder as myscript.

Convention states that scripts that are sourced by other scripts should NOT have a shebang at the top.

Analyzing RNA sequencing (RNA-seq) data involves several steps to go from raw sequence reads to meaningful biological insights. Here's a detailed step-by-step pipeline for RNA-seq analysis:

* **1. Quality Control (QC)**:
  + Perform quality control checks on raw sequencing reads using tools like FastQC.
  + Identify any sequencing artifacts, overrepresented sequences, or adapter contamination that may affect downstream analysis.
* **2. Pre-processing**:
  + Trim low-quality bases and adapter sequences from raw reads using tools like Trimmomatic or Cutadapt.
  + Remove reads of low quality or short length that may adversely impact downstream analysis.
* **3. Alignment/Mapping**:
  + Map the pre-processed reads to a reference genome or transcriptome using alignment tools like STAR, HISAT2, or Bowtie.
  + Choose appropriate parameters for alignment, considering the type of data (e.g., stranded vs. unstranded) and the reference genome/transcriptome used.
* **4. Quantification**:
  + Quantify gene or transcript expression levels from aligned reads using tools like featureCounts, HTSeq, or Salmon.
  + Obtain count or abundance estimates for each gene or transcript in the dataset.
* **5. Differential Expression Analysis**:
  + Identify differentially expressed genes or transcripts between experimental conditions using statistical methods like DESeq2, edgeR, or limma-voom.
  + Perform normalization to account for library size differences and other technical biases.
  + Adjust for multiple testing to control the false discovery rate (FDR).
  + Determine significance thresholds (e.g., fold change and adjusted p-value cutoffs) based on study design and biological relevance.
* **6. Functional Analysis**:
  + Conduct functional enrichment analysis to elucidate the biological processes, pathways, and functions associated with differentially expressed genes.
  + Use tools like Gene Ontology (GO) enrichment analysis, Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway analysis, or Reactome pathway analysis.
  + Interpret the results to gain insights into the underlying biological mechanisms driving observed expression changes.
* **7. Visualization**:
  + Generate visualizations to explore and summarize the RNA-seq data and analysis results.
  + Create plots such as volcano plots, MA plots, heatmaps, and cluster dendrograms to visualize differential expression and clustering patterns.
  + Use tools like R's ggplot2, matplotlib, or specialized visualization packages for RNA-seq data.
* **8. Validation**:
  + Validate the findings of differential expression analysis using independent methods such as qRT-PCR or additional RNA-seq experiments.
  + Perform functional validation experiments to confirm the biological relevance of identified gene expression changes.
* **9. Integration with other Omics Data**:
  + Integrate RNA-seq data with other omics data (e.g., proteomics, metabolomics) to gain a more comprehensive understanding of biological processes and pathways.
  + Explore correlations or associations between gene expression changes and other molecular features.
* **10. Publication and Documentation**:
  + Document the RNA-seq analysis pipeline, including software versions, parameter settings, and data sources used.
  + Prepare figures, tables, and supplementary materials for publication or presentation.
  + Share analysis scripts, code, and processed data through repositories or supplementary materials to enable reproducibility and transparency.

By following this step-by-step pipeline, researchers can perform comprehensive RNA-seq analysis and gain valuable insights into gene expression dynamics and regulatory mechanisms in biological systems.

steps and tools to "Trim low-quality bases and adapter sequences from RNAseq fastq files "

Trimming low-quality bases and adapter sequences from RNA-seq FASTQ files is an essential preprocessing step to ensure the quality and accuracy of downstream analysis. Here are the steps and tools to accomplish this task:

* **1. Quality Control (QC)**:
  + Before trimming, assess the quality of the raw sequencing data using tools like FastQC or MultiQC.
  + FastQC provides detailed reports on sequence quality, GC content, sequence length distribution, overrepresented sequences, and adapter contamination.
* **2. Adapter Trimming**:
  + Identify and remove adapter sequences from the raw reads to prevent interference with downstream alignment and analysis.
  + Use adapter trimming tools such as Cutadapt, Trimmomatic, or bbduk (from BBMap suite).
  + Example command using Cutadapt    - -a

cutadapt -a ADAPTER\_SEQUENCE -o output.fastq input.fastq

* + Replace ADAPTER\_SEQUENCE with the actual adapter sequence used during library preparation.
* **3. Quality Trimming**:
  + Trim low-quality bases from the ends of reads to improve overall sequence quality.
  + Set quality thresholds (e.g., Phred score) below which bases are trimmed.
  + Use trimming tools like Trimmomatic, Fastp, or Trim Galore.
  + Example command using Trimmomatic

java -jar trimmomatic.jar SE -phred33 input.fastq output\_trimmed.fastq LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

MIN

* + This command trims bases with quality below 3 from the leading and trailing ends, performs sliding window trimming with a window size of 4 and average quality of 15, and discards reads shorter than 36 bases after trimming.
* **4. Optional: Length Filtering**:
  + Filter out reads that become too short after trimming.
  + Specify a minimum length threshold to retain only high-quality, sufficiently long reads.
  + Use the MINLEN parameter in Trimmomatic or similar options in other trimming tools.
* **5. Post-trimming QC**:
  + After trimming, perform QC checks on the processed reads using FastQC or MultiQC to ensure that adapter sequences and low-quality bases have been effectively removed.
  + Evaluate the distribution of sequence quality scores, GC content, and other metrics to confirm the improvement in data quality.
* **6. Alignment and Analysis**:
  + Use the trimmed and quality-filtered FASTQ files as input for alignment to the reference genome/transcriptome and downstream RNA-seq analysis.
  + Tools like STAR, HISAT2, or Bowtie2 can be used for alignment, followed by quantification using tools like featureCounts or HTSeq.
  + Perform differential gene expression analysis and functional annotation based on the aligned and quantified data.

By following these steps and utilizing appropriate tools for adapter trimming and quality filtering, researchers can preprocess RNA-seq FASTQ files effectively, ensuring high-quality data for downstream analysis and interpretation.

# Bulk RNA Sequencing: A Comparison of the Most Popular Tools and Pipelines

**ANJALI GUPTA**

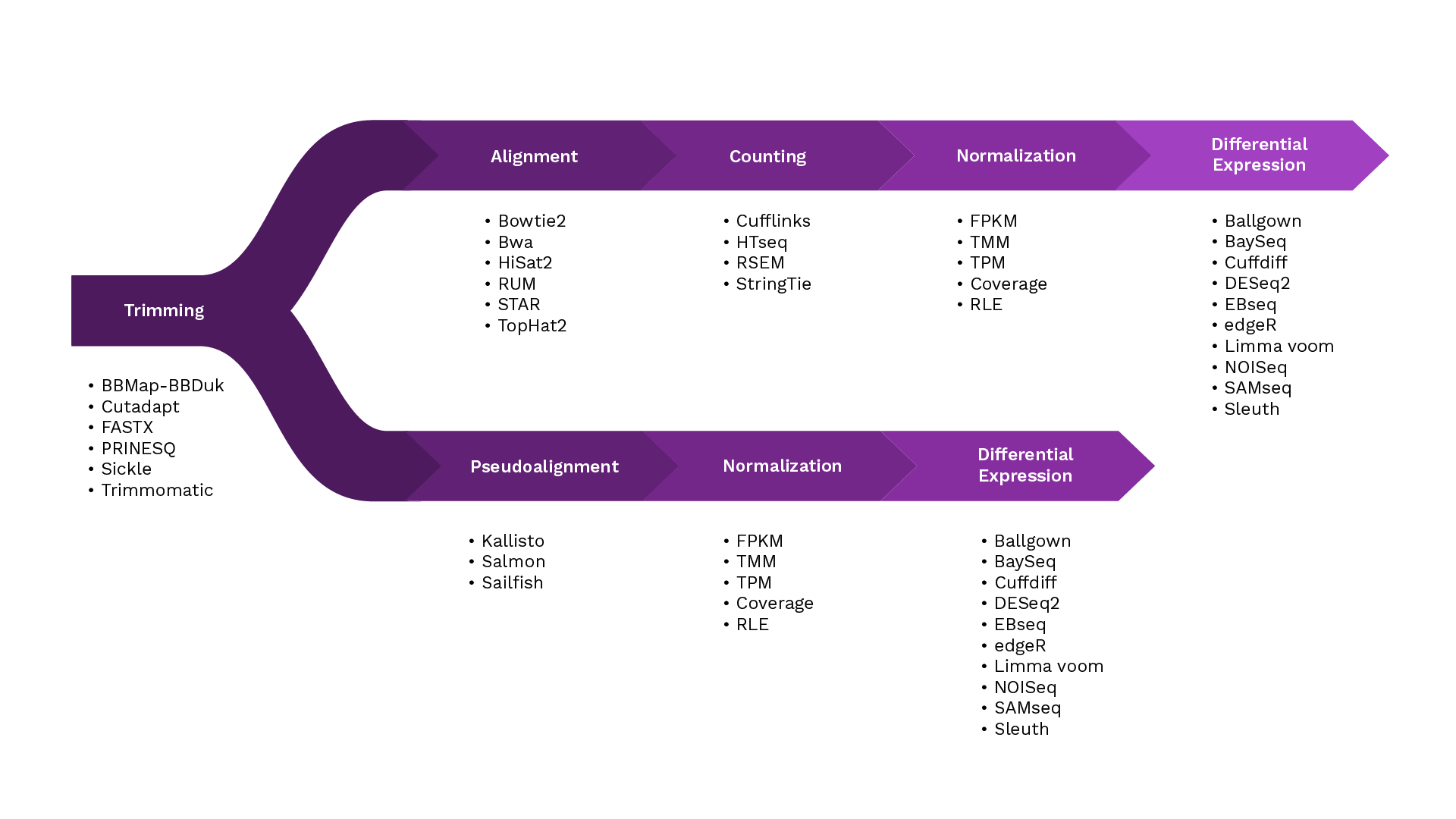
**FEBRUARY 10, 2023**

The association between **gene expression dynamics and biological function** has always been a subject of great interest in biology. With advances in technology, it has become possible to manipulate nucleic acids to quantify gene expression and interpret the significance biologically.

**Bulk RNA sequencing** has changed the way researchers approach a problem. RNA sequencing can provide qualitative and quantitative analyses of the entire transcriptome of the targeted cells/tissues/organisms. The applications of the data generated by Bulk RNA sequencing are boundless. The data could be used to study differential gene expression in healthy versus cancerous cells or in immune profiling. This has accelerated drug discovery in unprecedented ways.

However, with the increase in the significance of RNA sequencing, there has been a subsequent increase in the number of tools and techniques available to analyze raw reads obtained. The users are more puzzled than ever as to what is the best way to analyze their sequencing data. In this blog, we have assimilated a list of tools commonly used at various steps and have compared them.

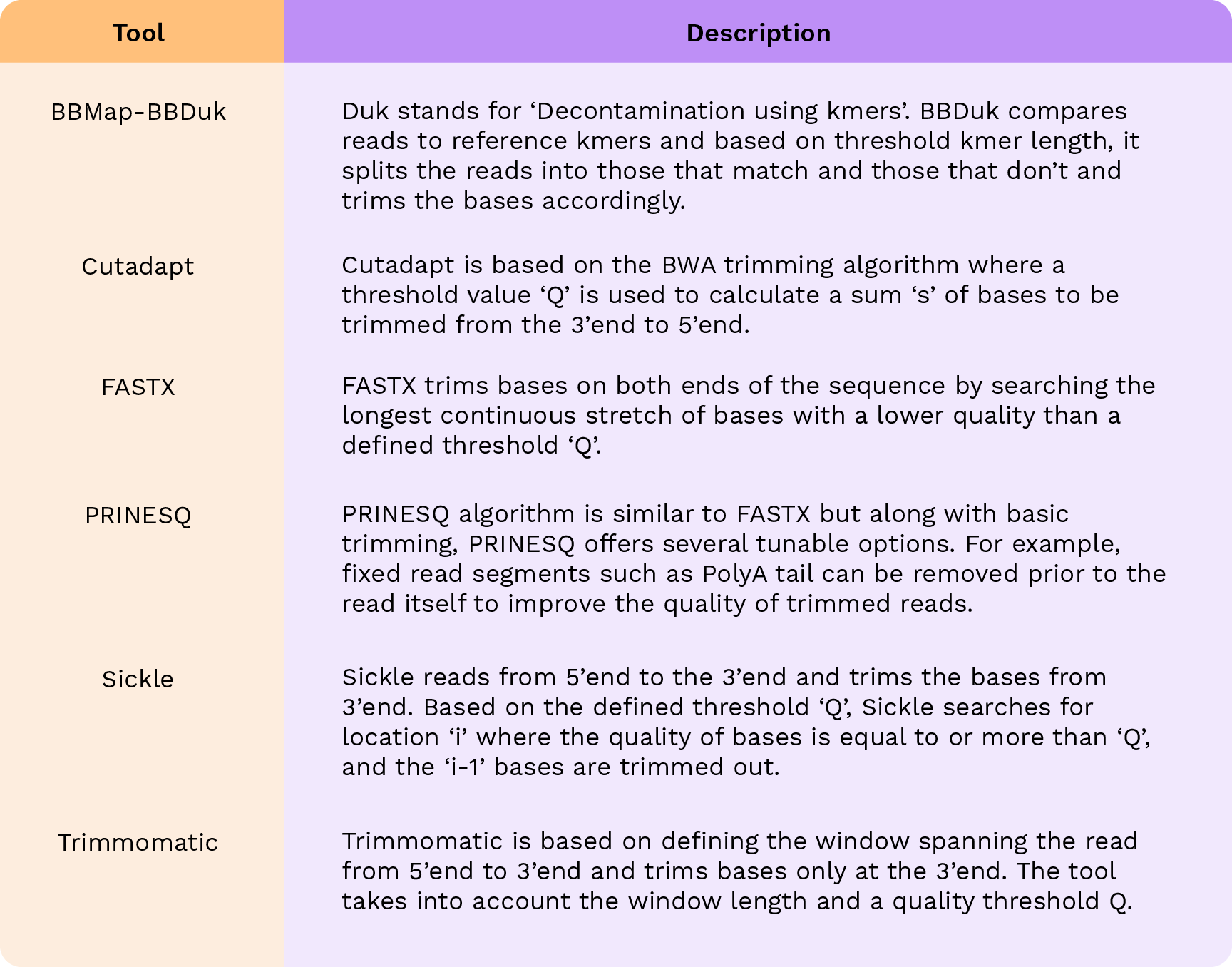
## Different Steps in the RNA Sequencing Analysis Pipeline



Following are the steps for analyzing RNAseq data -

### ****Step 1**** ****- Trimming/ Quality Check****

Trimming the raw data is crucial to eliminate the adaptor sequences and poor-quality nucleotides thereby increasing the rate of mapping reads. Trimming also increases the reliability of the downstream analysis while reducing computational requirements at the same time. But it should be done cautiously with carefully chosen trim length to prevent unwanted changes in gene expression and transcriptome assembly. Most sought-out[software](https://pubmed.ncbi.nlm.nih.gov/33184454/) used for [trimmings](https://pubmed.ncbi.nlm.nih.gov/24376861/) are

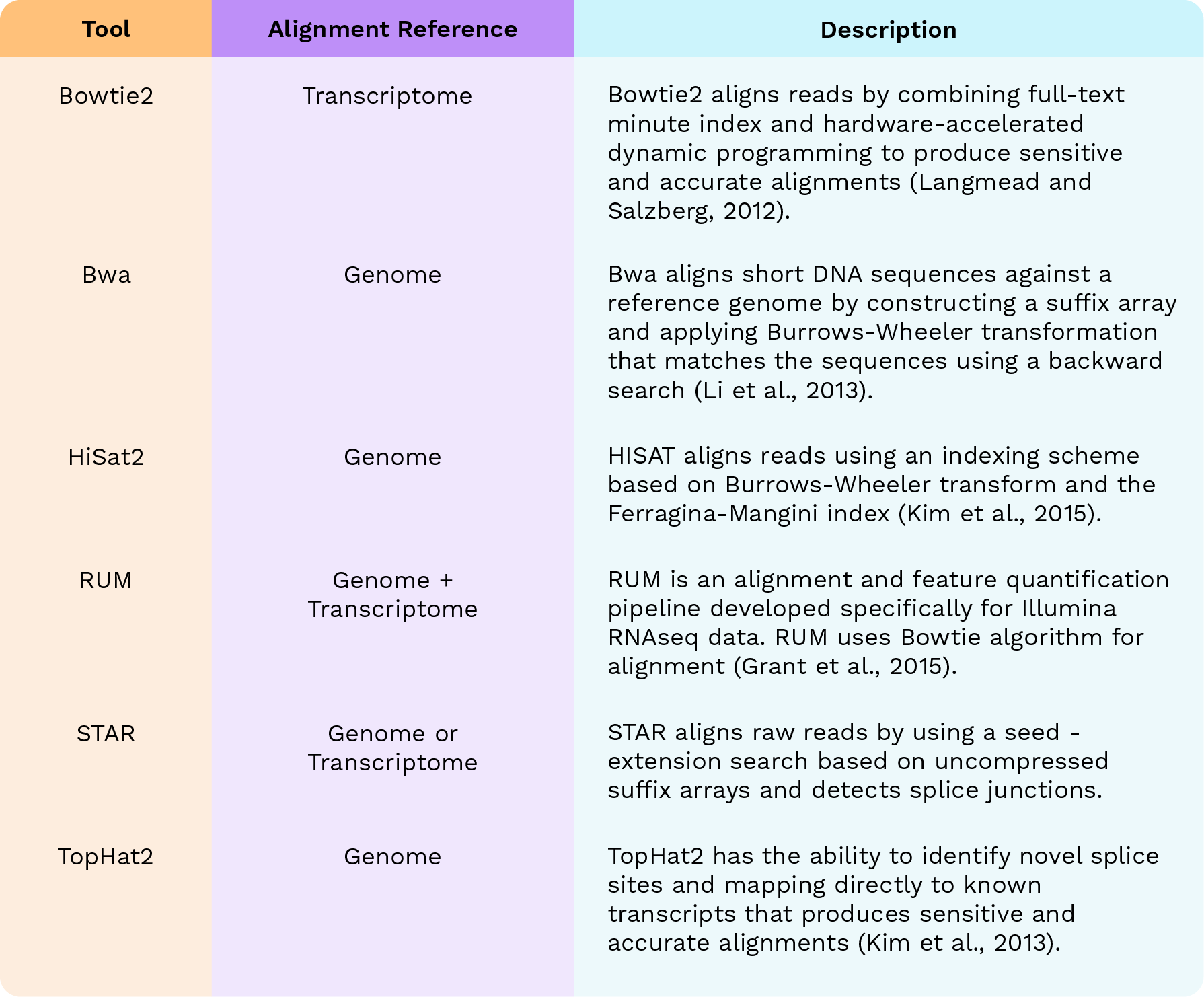


A whole host of tools are available for trimming however, various studies show that there is no generic answer to ‘what is the best trimming tool’. The choice of tool depends on the type of dataset downstream analysis and user-defined parameters that need to be taken into consideration. For example, setting the main threshold parameter too high could reduce the size of surviving dataset while setting it too low may render the trimming exercise futile.

(this does not matter with regard to compute resource – select the tool which can lead more cleaner read but optimize not to cut too much – check size to filter out) – find optimal size cut-off (search for best practice).

### ****Step 2 - Alignment****‍

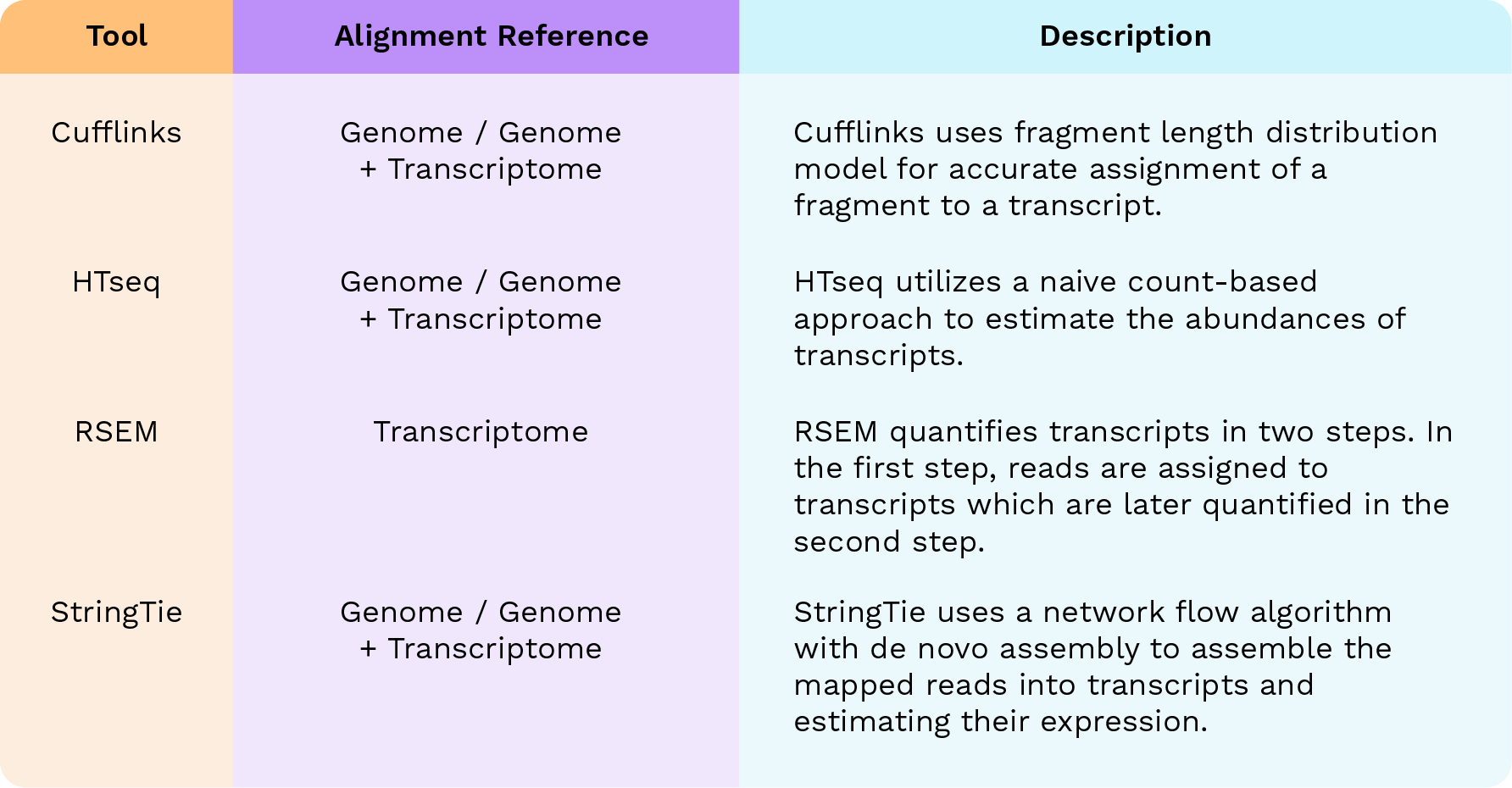
Aligning the reads to a reference genome/transcriptome is the second step in the RNAseq pipeline. The tools popular for alignment are-



BWA had the highest alignment rate (percentage of sequenced reads that were successfully mapped to reference genome) and the most coverage among all the tools. HiSat2 was the fastest aligner among all the tools. STAR and HiSat2 perform slightly better in [aligning](https://pubmed.ncbi.nlm.nih.gov/24204709/) the [unmapped](https://pubmed.ncbi.nlm.nih.gov/33184454/) reads.**‍**

### ****Step 3 - Counting/Quantification****

**‍**After the reads have been mapped, they are assigned to a gene or transcript in a process called counting/quantification. This step quantifies the number of transcripts that will later be used to compare case versus control. The most commonly used tools are-



When compared for the best tool, Cufflinks and RSEM were ranked at the top followed by HTseq and [StringTie-based](https://pubmed.ncbi.nlm.nih.gov/33184454/) pipelines.

Port the older pipeline and do efficiency testing to check this (parallel vs sequential computation)

### ****Step 4 - Normalization****‍

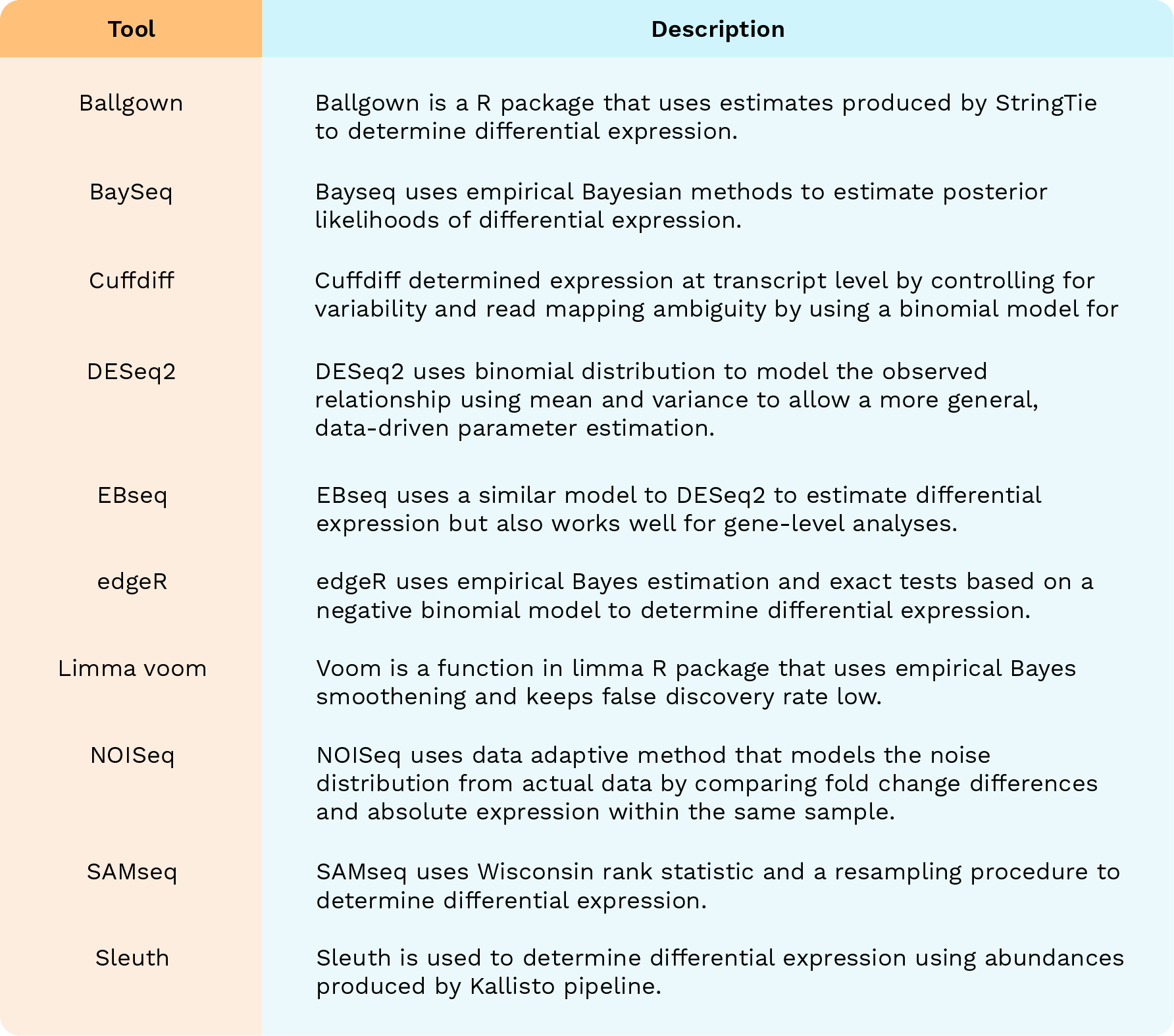
After the counting step, quantified transcripts undergo a normalization procedure to remove sequencing bias. Each normalization technique represents different gene expression values - Fragments per Kilobase of Mapped reads (FPKM), Transcripts per Million (TPM), Trimmed Mean of M values (TMM from edgeR), and Relative Log Expression (RLE from DESeq2), upper quartile (UQ), coverage (cov), estimated counts (est\_counts) and effective counts (eff\_counts). Researchers evaluated various normalization methods and found that the pipelines using TMM performed best followed by RLE, TPM, and FPKM.

#### ****Pseudoalignment****-

Apart from the above-mentioned steps, there is an alternative pipeline called quantification by Pseudoalignment. The process where all three steps - alignment, counting and normalization are performed in a single step is called pseudoalignment. Commonly used pseudo aligners are - Kallisto, Salmon and Sailfish. When compared, all three methods showed similar performance in terms of [precision and accuracy](https://pubmed.ncbi.nlm.nih.gov/33184454/).

#### ****Differential Expression****-

The final step in the RNAseq analysis pipeline is comparing the normalized transcript counts in case versus control to get the differential expression of genes. Since this is the most crucial step, there are several tools and techniques developed for DE analysis.



When compared for detection ability amongst these tools, Cuffdiff generated the least number of differentially expressed genes while SAMseq generated the most number of differentially expressed genes. When compared for accuracy, limma trend, limma voom and baySeq turned out to be the most accurate. Overall, for 16 different parameters, baySeq turned out to be the best tool for analysis followed by edgeR, limma trend, and [limma voom](https://pubmed.ncbi.nlm.nih.gov/33184454/).

HISat2-HTseq (pipeline for alignment and mapping) is the most efficient; and the baySeq is the most accurate (for differential genes)

Due to its efficiency, I will use the HISat2-HTseq (for efficiency and economical compute resource/time)

### ****Overall Pipeline Comparison****

Six RNAseq pipelines [were compared](https://pubmed.ncbi.nlm.nih.gov/35337265/)-



Most tools and pipelines available are comparable with each other and produce similar results. Researchers can choose tools based on the available computing resources, the research objectives, and the gene expression values. It would be wise to utilize multiple procedures/pipelines to obtain the most reliable fold change and the number of differentially expressed genes.

Already picked more efficient pair for the first part (HISat2-HTseq pipeline) – limited compute resource

HISat2-cufflink-baySeq

### Bulk RNA Sequencing Data on Polly

[Polly](https://www.elucidata.io/blog/visualization-of-bulk-rna-seq-data-using-phantasus#:~:text=Bulk%20RNASeq%20OmixAtlas.-,Polly,-hosts%20the%20world%E2%80%99s) is a data-centric ML Ops platform that hosts OmixAtlas. OmixAtlas is a collection of several millions of datasets that consists of carefully curated biomolecular data. Each dataset ingested into Polly OmixAtlas is efficiently curated to make the dataset **analysis ready**. Curated datasets also make it easier to **query, find, and search** datasets.

Bulk RNA sequencing OmixAtlas on Polly provides curated datasets with several metadata such as cell type, cell line, disease, drug, tissue, and organism. All datasets ingested go through 2 steps -

1. **Data engineering**- This is done to transform the data into one consistent data schema.
2. **Metadata Harmonization** - This is done to tag each sample and data into a uniform ontology.

[**Reach out**](https://www.elucidata.io/schedule-a-meeting)**to us to learn more about how to accelerate your research!**

‍

**References:**

* Borozan, I., Watt, S. N., & Ferretti, V. (2013). Evaluation of alignment algorithms for discovery and identification of pathogens using RNA-Seq. PloS one, 8(10), e76935.
* Corchete, L. A., Rojas, E. A., Alonso-López, D., De Las Rivas, J., Gutiérrez, N. C., & Burguillo, F. J. (2020). Systematic comparison and assessment of RNA-seq procedures for gene expression quantitative analysis. Scientific reports, 10(1), 19737.
* Del Fabbro, C., Scalabrin, S., Morgante, M., & Giorgi, F. M. (2013). An extensive evaluation of read trimming effects on Illumina NGS data analysis. PloS one, 8(12), e85024.
* Grant, G. R., Farkas, M. H., Pizarro, A. D., Lahens, N. F., Schug, J., Brunk, B. P., Stoeckert, C. J., Hogenesch, J. B., & Pierce, E. A. (2011). Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). Bioinformatics (Oxford, England), 27(18), 2518–2528.
* Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., & Salzberg, S. L. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome biology, 14(4), R36.
* Kim, D., Langmead, B., & Salzberg, S. L. (2015). HISAT: a fast spliced aligner with low memory requirements. Nature methods, 12(4), 357–360.
* Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature methods, 9(4), 357–359.
* Liu, X., Zhao, J., Xue, L., Zhao, T., Ding, W., Han, Y., & Ye, H. (2022). A comparison of transcriptome analysis methods with reference genome. BMC genomics, 23(1), 232.
* Musich, R., Cadle-Davidson, L., & Osier, M. V. (2021). Comparison of Short-Read Sequence Aligners Indicates Strengths and Weaknesses for Biologists to Consider. Frontiers in plant science, 12, 657240.

Email from Nela:

Please see the attached files. There you can find all the indexes that we used for sample prep. In the CPT\_Run2\_NextSeq2000\_AAC73CNHV there are 3 samples highlighted with pink, they are the 3 samples that I need to redo. Also, attached to this email, it is the Arun seq data information.