Introduction to Applied Bioinformatics Workshop

Day 3

KAUST Bioinformatics Platform

10 Feb 2026

Recap: Day 01 & Day 02

Linux Commands

- Basic Commands: ls, cd, pwd, mkdir, rm, cp, mv
- File manipulation: cat, head, tail, less, wc
- Text processing: grep, awk, sed
- Data transfer: wget, curl, scp
- Pipes (|), redirection (>), append (>>), background (&)

Ibex Commands

- Resource: srun
- Module system: module load, module list, module avail

Bioinformatics Tools

- Data retrieval: sratoolkit, ncbi datasets
- Quality Assessment: FastQC, MultiQC
- FASTA/Q manipulation: seqkit
- Quality filtering: fastp

Data Formats

- FASTA, FASTQ, GTF

Day 3: Outline

Read Alignment

- Tools: STAR, samtools
- Data transfer: wget, curl, scp

Hands-on session: Read Alignment

Splice-aware alignment

- STAR, samtools

Sequence Alignment

- Tools: blastn, makeblastdb

Data Formats

- BAM file

Hands-on session: Read Alignment

Read Alignment to Reference Genomes

DNA-seq & RNA-seq Alignment

KAUST Bioinformatics Platform

10 Feb 2026

Session Objectives

- Understand what read alignment is and why it matters
- Learn how aligners work (indexing, seeding, extension)
- Know when to use BWA (DNA-seq) vs STAR (RNA-seq)
- Run alignments and interpret statistics
- Perform post-alignment processing

> Alignment is the bridge between raw sequencing reads and biological insight — choosing the right aligner and parameters is critical.

What is Read Alignment?

Mapping reads back to a reference genome

The Problem

You sequenced a sample and got millions of short reads (50–300 bp). Now what?

```
@SRR123456.1
ATCGATCGATCGATCGATCGATCGATCGATCGATCG
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

You need to figure out where in the genome each read came from.

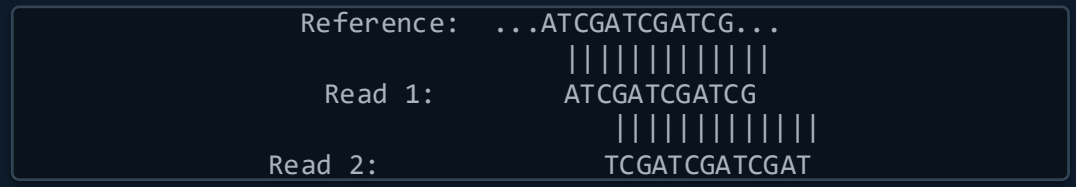
> This is fundamentally different from sequence alignment (BLAST, Needleman-Wunsch). Here we align billions of short reads against a 3-billion-bp reference — speed is everything.

Read Alignment: The Concept

The Task & Challenges

- Find where each read originated in the genome
- Billions of short reads to map
- 3 billion bp reference (human)
- Sequencing errors (~0.1–1%)
- Real biological variants (SNPs, indels)
- Repetitive regions (~50% of human genome)

Output: BAM file



- Genomic coordinates for each read
- CIGAR string (alignment details)
- Mapping quality score
- Mate pair information

How Aligners Work

Step 1: Index the Reference

- Pre-process reference genome once
- BWT / FM-index / hash tables

Step 2: Seed (Find Exact Matches)

- Break each read into short k-mers
- Look up exact matches in the index

Step 3: Extend

- Extend seeds into full alignments
- Allow mismatches and gaps

Step 4: Select

- Pick the best alignment(s)
- Calculate mapping quality (MAPQ)
- Handle multi-mappers

> The index-then-search strategy is what makes aligning billions of reads feasible. Without indexing, alignment would take weeks instead of hours.

Reference Genomes

What You Need

- FASTA file of the genome sequence
- GTF/GFF file of gene annotations
- Matching versions! (genome + annotation)

Where to Download

- Ensembl (recommended)
- UCSC Genome Browser
- NCBI / RefSeq
- GENCODE (for annotations)

Human Reference Assemblies

Assembly	Also Known As	Status
GRCh38	hg38	Current standard
GRCh37	hg19	Legacy (still used)
T2T-CHM13	—	Telomere-to-telomere

Watch out! Chromosome naming differs:
Ensembl: 1, 2, 3, X, Y
UCSC: chr1, chr2, chr3, chrX, chrY
Your FASTA and GTF must match!

DNA-seq Alignment: BWA-MEM2

The standard for DNA sequencing

BWA-MEM2: When to Use

Use Cases

- Whole genome sequencing (WGS)
- Whole exome sequencing (WES)
- Targeted gene panels
- ChIP-seq / ATAC-seq
- Any genomic DNA data

Features

- Fast and accurate
- Handles reads from 70 bp to long reads
- Gap-aware alignment
- Industry standard (GATK best practices)
- BWA-MEM2 is ~2x faster than BWA-MEM

BWA: Index and Align

Build the Index (once per reference)

```
module load bwa-mem2/2.2.1

bwa-mem2 index genome.fa
```

Align Paired-End Reads

```
bwa-mem2 mem -t 8 \
      genome.fa \
  sample_R1.fastq.gz sample_R2.fastq.gz \
| samtools sort -@ 4 -o sample.sorted.bam -

samtools index sample.sorted.bam
```

> BWA outputs SAM to stdout. Pipe directly to samtools for BAM conversion and sorting — avoids writing a large intermediate file.

DNA-seq: What the Alignment Looks Like

```
                    Reference: ATCGATCGATCGATCGATCGATCG
          Read 1:    ATCGATCGATCG............     (perfect match)
Read 2:    ..CGATCAATCGATCG........     (1 mismatch = SNP or error)
          Read 3:    ........ATCG--TCGATCGATC     (2 bp deletion)
          Read 4:    ............ATCGATCGATCG     (perfect match)
```

Key Points for DNA-seq

- Reads align contiguously to the reference (no splicing)
- Mismatches may be sequencing errors or real variants (SNPs)
- Gaps may be real indels or alignment artifacts
- Downstream: variant calling (GATK, DeepVariant) separates real variants from errors

RNA-seq Alignment: STAR

Spliced Transcripts Alignment to a Reference

The RNA-seq Challenge

Why Can't We Just Use BWA?

- RNA-seq reads come from mRNA (mature, spliced)
- A single read can span an exon-exon junction
- A regular aligner like BWA would fail to map or mis-align

What STAR Does

```
Genomic DNA:  [Exon1]---intron---[Exon2]

mRNA:         [Exon1][Exon2]

Read:             XXXXX|XXXXX
                  Exon1|Exon2
                 ^ splice junction
```

> Always use a splice-aware aligner for RNA-seq! Using BWA on RNA-seq data will lose junction-spanning reads and bias your results.

STAR: Features

Why STAR?

- Splice-aware alignment
- Very fast (faster than HISAT2 for most cases)
- High accuracy for junction detection
- Detects novel splice junctions
- Can output gene counts directly
- ENCODE / GATK recommended

Alternatives

Aligner	Speed	Memory	Notes
STAR	Very fast	~32 GB	Recommended
HISAT2	Fast	~8 GB	Lower memory
Salmon	Very fast	~1 GB	Pseudo-alignment (no BAM)

> STAR needs ~32 GB RAM for the human genome. If memory is limited, consider HISAT2 or Salmon.

STAR: Building the Index

```
module load star/2.7.10b

# Build genome index (run once per genome+annotation)
STAR --runMode genomeGenerate \
     --runThreadN 8 \
     --genomeDir ./star_index \
     --genomeFastaFiles genome.fa \
     --sjdbGTFfile annotation.gtf \
     --sjdbOverhang 99
```

Important Parameters

Parameter	Value	Meaning
--genomeDir	Directory path	Where to store index files
--genomeFastaFiles	genome.fa	Reference genome FASTA
--sjdbGTFfile	annotation.gtf	Gene annotation (known junctions)
--sjdbOverhang	read_length - 1	99 for 100bp, 149 for 150bp

STAR: Running the Alignment

```
STAR --runThreadN 8 \
     --genomeDir ./star_index \
     --readFilesIn sample_R1.fastq.gz sample_R2.fastq.gz \
     --readFilesCommand zcat \
     --outFileNamePrefix sample_ \
     --outSAMtype BAM SortedByCoordinate \
     --outSAMunmapped Within \
     --quantMode GeneCounts
```

Key Outputs

File	Contents
sample_Aligned.sortedByCoord.out.bam	Aligned reads (sorted BAM)
sample_Log.final.out	Alignment statistics
sample_ReadsPerGene.out.tab	Gene-level read counts
sample_SJ.out.tab	Detected splice junctions

STAR Log: What Good Looks Like

```
                                 Number of input reads |  20000000
                      Uniquely mapped reads number     |  18000000
                           Uniquely mapped reads %     |  90.00%
                 Number of reads mapped to multiple loci |  1500000
                      % of reads mapped to multiple loci |  7.50%
                                 % of reads unmapped   |  2.50%
```

Good Alignment

- Unique mapping > 70%
- Multi-mapping < 20%
- Unmapped < 10%

Warning Signs

- Low unique mapping → contamination? wrong reference?
- High unmapped → wrong organism? degraded RNA?
- High multi-map → repetitive sequences, rRNA contamination

DNA-seq vs RNA-seq: Key Differences

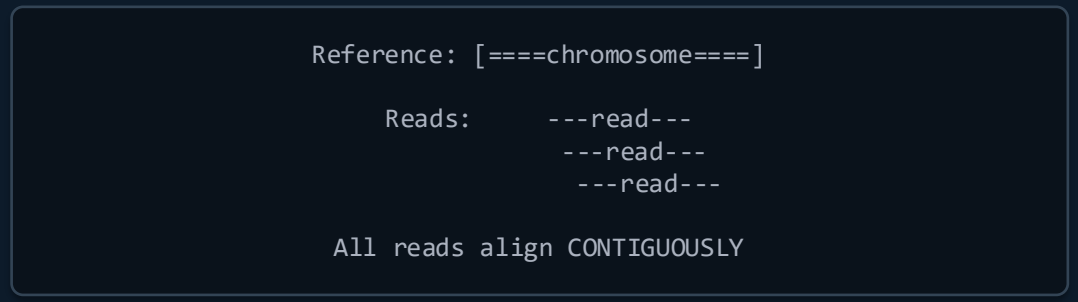
Choosing the Right Aligner

Feature	DNA-seq (BWA)	RNA-seq (STAR)
Input data	Genomic DNA fragments	mRNA / cDNA
Splicing	No — reads align contiguously	Yes — reads span exon junctions
Aligner	BWA-MEM2	STAR (or HISAT2)
Annotation needed?	No (for alignment)	Yes (GTF for known junctions)
Memory	~8 GB	~32 GB (STAR)
Downstream	Variant calling (GATK)	Quantification → DE (DESeq2)

> The #1 beginner mistake: Using BWA for RNA-seq or STAR for DNA variant calling. Match the aligner to your data type!

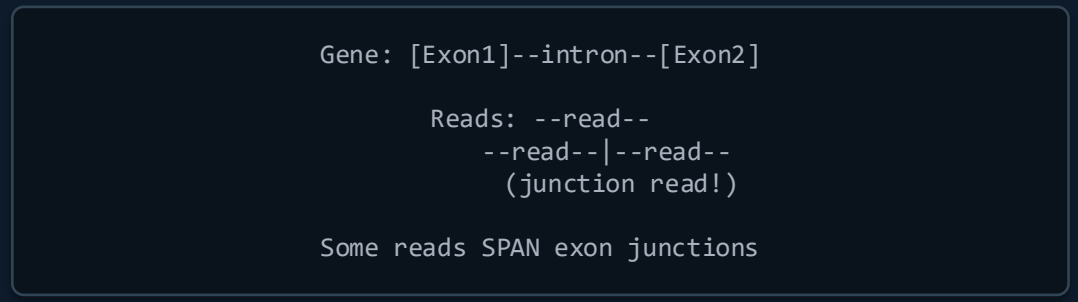
Visual Comparison

DNA-seq Alignment



Goal: Find variants (SNPs, indels, SVs)

RNA-seq Alignment



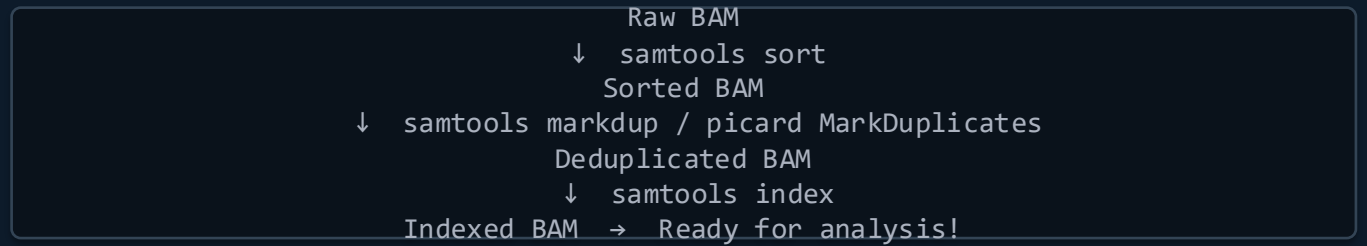
Goal: Quantify gene expression

Post-Alignment Processing

Getting your BAM analysis-ready

The Post-Alignment Pipeline

Raw BAM → Analysis-Ready BAM



Why Each Step Matters

Step	Why	Tool
Sort	Most tools require coordinate-sorted BAMs	samtools sort
Mark duplicates	PCR duplicates inflate signals	samtools markdup / Picard
Index	Enables random access (e.g., view chr1:1-2000)	samtools index

Post-Alignment Commands

```
# Sort (if not already sorted)
samtools sort -@ 4 -o sorted.bam aligned.bam

# Mark duplicates
samtools markdup sorted.bam marked.bam

# Index the final BAM
samtools index marked.bam
```

One-Liner Pipeline (BWA example)

```
bwa-mem2 mem -t 8 ref.fa R1.fq.gz R2.fq.gz \
    | samtools sort -@ 4 \
    | samtools markdup - - \
    | samtools view -b -o final.bam

samtools index final.bam
```

Alignment Quality Metrics

```
# Quick summary
samtools flagstat sample.bam

# Detailed statistics
samtools stats sample.bam > stats.txt
grep "^SN" stats.txt
```

What to Check

- % mapped (should be > 90%)
- % properly paired (> 85% for PE)
- % duplicates (varies by library prep)
- Insert size distribution
- Error rate (should be < 1%)

Troubleshooting

Problem	Possible Cause
Low mapping rate	Wrong reference genome
Low proper pairs	Library preparation issue
High error rate	Sequencing problem
Unexpected insert size	Contamination
Very high duplicates	Low library complexity

MultiQC: See Everything at Once

What MultiQC Aggregates

- FastQC reports (pre- and post-trimming)
- STAR / BWA alignment logs
- samtools flagstat / stats
- Picard duplicate metrics
- featureCounts summary
- 100+ supported tools

```
# Run in your project directory
multiqc .

# Opens: multiqc_report.html
```

> Run MultiQC after each pipeline step to track quality throughout. Compare across samples to quickly spot outliers.

Key Takeaways

Alignment Basics

- Reference genome + index required before alignment
- Aligners use index → seed → extend → select strategy
- Always check that genome and annotation versions match

Choose the Right Aligner

- BWA-MEM2 for DNA-seq (WGS, WES, ChIP-seq)
- STAR for RNA-seq (splice-aware, needs ~32 GB RAM)
- Never use BWA for RNA-seq or STAR for DNA variant calling

Post-Processing

- Sort → Mark duplicates → Index
- Check metrics: mapping rate, proper pairs, duplicates
- Use MultiQC to aggregate and compare across samples

Hands-on Lab

Sequence Alignment Fundamentals

Pairwise Alignment, MSA & BLAST

KAUST Bioinformatics Platform

10 Feb 2026

Session Objectives

- Understand why we align sequences
- Learn pairwise alignment concepts (global vs local)
- Understand scoring matrices and gap penalties
- Perform multiple sequence alignments
- Run and interpret BLAST searches

> Key Takeaway: Alignment is the most fundamental operation in bioinformatics — nearly every analysis starts here.

Part 1: Pairwise Alignment

Comparing two sequences

Why Align Sequences?

Key Questions

- Are these sequences related?
- What is the function of this gene?
- How have sequences evolved?
- Where are the conserved regions?

Applications

- Homology detection
- Functional annotation
- Phylogenetic analysis
- Structure prediction
- Variant identification
- Primer design

> Similar sequences often have similar functions. Alignment helps us find and quantify that similarity.

What is Sequence Alignment?

Arranging sequences to identify similar regions

```
Sequence 1:  A T G C T A G - C T A
             | |   | | | |   | | |
Sequence 2:  A T - C T A G G C T A
```

Alignment Elements

Element	Symbol	Meaning
Match	|	Same character at same position
Mismatch		Different characters aligned
Gap	-	Insertion or deletion (indel)

> Goal: Find the arrangement that maximizes similarity (or minimizes differences).

Types of Pairwise Alignment

Global Alignment

- Align entire sequences end-to-end
- Best for similar-length sequences
- Algorithm: Needleman-Wunsch (1970)

Use when: Comparing full genes or proteins of the same family

Local Alignment

- Find the best matching sub-region
- Best for partial matches or domain search
- Algorithm: Smith-Waterman (1981)

Use when: Finding domains, conserved motifs, database searches

Global vs Local — When to Use Which?

Scenario	Type	Why
Compare two hemoglobin sequences	Global	Similar length, same family
Find a known domain in a new protein	Local	Only part matches
Check if two full genomes are related	Global	End-to-end comparison
Search a database for similar sequences	Local	Partial matches expected
Identify a shared motif	Local	Short conserved region

> Choosing the wrong type can give misleading results! A global alignment of very different-length sequences will force unnecessary gaps.

Alignment Scoring

How do we know which alignment is "best"?

Simple Scoring Scheme

Event	Score
Match	+1
Mismatch	-1
Gap	-2

```
          ATGC-TA
          |||||.|
    ATGCGTA   Score: 5(+1)+0(-1)+1(-2) = 3
```

Why Simple Isn't Enough

- Not all mismatches are equal!
- A → G (purine → purine) is more likely
- A → C (purine → pyrimidine) is less likely
- We need substitution matrices that encode biological likelihood

Substitution Matrices

For DNA

- Simple match/mismatch often sufficient
- Transitions (A↔G, C↔T) more common
- Transversions (purine↔pyrimidine) less common

For Proteins: BLOSUM

- Based on observed substitutions in conserved protein blocks
- BLOSUM62 is the most widely used (default in BLAST)

BLOSUM62 (partial)

	A	R	N	D	C
A	4	-1	-2	-2	0
R	-1	5	0	-2	-3
N	-2	0	6	1	-3
D	-2	-2	1	6	-3
C	0	-3	-3	-3	9

- Positive = favorable substitution
- Negative = unfavorable
- Diagonal = match scores (C→C = 9!)

Choosing the Right BLOSUM Matrix

Matrix	Sequences	Use Case
BLOSUM80	Very similar (>80% identity)	Close homologs
BLOSUM62	Moderate similarity (default)	General purpose
BLOSUM45	Distantly related	Remote homologs

> Higher BLOSUM number = designed for more similar sequences. This is the opposite of PAM matrices, where higher number = more distant.

Gap Penalties

Gaps represent insertions/deletions — they should be penalized

Linear Gap Penalty

- Penalty = g × n
- g = penalty per gap position, n = gap length
- Simple but not biologically realistic

Affine Gap Penalty

- Penalty = d + e × (n-1)
- d = gap open penalty (expensive!), e = gap extension (cheap)
- More realistic: one mutation can insert multiple bases

> Typical values — Gap open: -10 to -12 | Gap extension: -1 to -2. Opening a gap is expensive; extending it is much cheaper.

Dynamic Programming: How Alignment Works

The Algorithm (Conceptual)

- 1. Create a matrix (seq1 × seq2)
- 2. Fill cells with optimal scores
- 3. Traceback to find the best alignment

At Each Cell, Choose the Best:

- Diagonal → match/mismatch
- From above → gap in sequence 1
- From left → gap in sequence 2

Example: Align AGG to ATG

	-	A	T	G
-	0	-2	-4	-6
A	-2	1	-1	-3
G	-4	-1	0	0
G	-6	-3	-2	1

> Complexity: O(mn) — fine for pairs, but too slow for database searches. That's why we need BLAST!

Part 2: Multiple Sequence Alignment

Aligning more than two sequences

Why Multiple Sequence Alignment?

What MSA Reveals

- Conserved regions → functionally important
- Variable regions → less constrained
- Evolutionary patterns across species
- Functional domains shared by protein families

Applications

- Build phylogenetic trees
- Predict protein structure/function
- Identify functional domains
- Design primers for conserved regions
- Detect co-evolving positions

> MSA is the foundation for phylogenetics — you cannot build a reliable tree without a good alignment!

MSA Example: Hemoglobin Alpha Chain

```
Human      MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSH
Chimp      MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSH
Mouse      MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSH
Chicken    MVLSAADKNNVKGIFTKIAGHAEEYGAETLERMFTTYPPTKTYFPHFDLSH
           ***:.:**: *:. :  * .*.*.***:****** :*:**********:**
```

Conservation Symbols (ClustalW format)

Symbol	Meaning
*	Fully conserved — identical in all sequences
:	Strongly similar — conservative substitution
.	Weakly similar — semi-conservative substitution
(space)	Not conserved

MSA Tools

Tool	Method	Speed	Best For
ClustalW	Progressive	Slow	Historical reference
Clustal Omega	HMM-based	Fast	Large datasets
MUSCLE	Iterative	Medium	High accuracy
MAFFT	Various strategies	Fast	General use
T-Coffee	Consistency-based	Slow	Maximum accuracy

> Start with Clustal Omega or MAFFT for most tasks. Use MUSCLE when accuracy is critical. All are available both online and as command-line tools.

Evaluating MSA Quality

Good Signs

- Clear conserved blocks visible
- Gaps cluster at ends or loop regions
- Known motifs and domains are preserved
- Alignment is biologically meaningful

Warning Signs

- Too many scattered gaps
- Low overall conservation
- Gaps within known functional domains
- Alignment shifts unexpectedly

> Alignment algorithms always produce an alignment, even if the sequences are completely unrelated! Always check that the result makes biological sense.

Part 3: BLAST

Basic Local Alignment Search Tool

What is BLAST?

Purpose

- Search a sequence against a database
- Find similar sequences rapidly
- Uses a heuristic algorithm (not guaranteed optimal)
- The most used bioinformatics tool — billions of searches/year

How It Works (Simplified)

- 1. Break query into short words (seeds)
- 2. Find exact matches in the database
- 3. Extend matches in both directions
- 4. Keep high-scoring alignments
- 5. Report results with statistics

> Smith-Waterman is optimal but too slow for large databases. BLAST trades a little sensitivity for massive speed.

BLAST Programs

Program	Query	Database	Use Case
blastn	DNA	DNA	Find similar nucleotide sequences
blastp	Protein	Protein	Find similar proteins
blastx	DNA → translated	Protein	What does this DNA encode?
tblastn	Protein	DNA → translated	Find a gene in a genome
tblastx	DNA → translated	DNA → translated	Compare at protein level

> Most common: blastn ("Is my DNA in the database?"), blastp ("What proteins are similar?"), blastx ("What protein does this unknown DNA encode?")

Understanding BLAST Output

Key Statistics

- **Bit Score** — alignment quality (higher = better)
- **E-value** — expected hits by chance (lower = better)
- **% Identity** — matching positions / alignment length
- **Query Coverage** — % of query that aligned

E-value Interpretation

E-value	Interpretation
< 1e-50	Identical or nearly identical
< 1e-10	Definitely homologous
< 0.01	Probably related
> 1	Likely a random match

> E-value = the number of hits you'd expect by chance given the database size. An E-value of 1e-30 means you'd need to search 1e30 random databases to see this hit once by luck.

BLAST Tabular Output

For automated analysis, use -outfmt 6 (tabular):

```
blastp -query protein.fasta -db nr -outfmt 6 -evalue 1e-5
```

```
# qseqid  sseqid    %id    len    mm   gaps  qstart qend  sstart send  evalue    bitscore
  seq1      gi|123    99.5   200    1    0     1      200   1      200   1e-100    380
  seq1      gi|456    85.0   180    27   0     10     189   5      184   1e-50     200
  seq1      gi|789    60.0   150    60   5     25     170   100    245   1e-10     80
```

> Tabular format is easy to parse: awk '$11 < 1e-10' results.txt # Filter by E-value

Common BLAST Pitfalls

Over-interpreting

- High identity ≠ same function
- Top hit ≠ correct annotation
- Similarity ≠ homology
- Short high-identity hits can be random

Under-interpreting

- Don't dismiss low-identity hits
- Check alignment length and coverage
- Consider biological context
- Verify with other evidence

> BLAST finds similarity, not homology! Homology is a biological conclusion about shared ancestry. Similarity is just a measurement.

BLAST Best Practices

- 1. Choose the right program — blastn for DNA, blastp for protein, blastx if unsure
- 2. Select appropriate database — SwissProt for quality, nr for comprehensiveness
- 3. Check E-values — don't trust hits with E > 0.01
- 4. Look at coverage — full-length match vs domain only?
- 5. Examine the alignment — where do mismatches/gaps occur?
- 6. Consider the biology — does the hit make sense for your organism/context?
- 7. Don't stop at the top hit — check multiple hits for consistency

Key Takeaways

Pairwise Alignment

- Global (Needleman-Wunsch): full-length, end-to-end
- Local (Smith-Waterman): best matching sub-region
- Scoring: substitution matrices (BLOSUM62) + gap penalties

Multiple Sequence Alignment

- Foundation for phylogenetics and conservation analysis
- Tools: Clustal Omega, MUSCLE, MAFFT
- Always inspect results — algorithms can't tell you if biology makes sense

BLAST

- Fast heuristic for database searches
- E-value: statistical significance (lower = more significant)
- Bit score: alignment quality (higher = better)
- Similarity ≠ homology!

Hands-on Lab

Workshop Conclusion

Introduction to Applied Bioinformatics

KAUST Academy 2026

THE BIOINFORMATICS PLATFORM

What is Bioinformatics?

Bioinformatics is the application of computational tools and techniques to understand biological data.

> It sits at the intersection of biology, computer science, and statistics — turning raw sequencing data into biological insight.

In this workshop you learned to:

- Work on a Linux command line
- Genomic File Formats
- Retrieve data from public repositories
- Assess and improve data quality
- Align reads to a reference genome
- Search sequences with BLAST

What We Covered

LAB 1

Linux Basics

Navigation, file manipulation, pipes, and working on Ibex

LAB 2

Genomic File Formats

FASTA, FASTQ, GFF/GTF, BED, SAM/BAM

LAB 3

Public Data Retrieval

NCBI SRA, GEO, sra-tools, and the NCBI datasets CLI

LAB 4

Quality Control

FastQC for assessment, fastp for trimming & filtering

LAB 5

Genome Alignment

STAR index, read alignment, samtools, BAM & IGV

LAB 6

BLAST Search

blastn, blastp, nt/nr databases, filtering & local databases

Linux Command Line

Why it matters

- Most bioinformatics tools are command-line only
- HPC clusters like Ibex require terminal access
- Scripts enable reproducible and scalable analyses

Key skills you gained

- **cd, ls, mkdir, cp, mv, less, wc, head, tail** — navigation & file management
- **grep, nano** — text processing
- **pipes (|)** — chaining commands together
- **module load, scp** — accessing software on Ibex

> The command line is your universal interface to bioinformatics. Every tool you used in this workshop — from sra-tools to STAR to BLAST — was run from the terminal.

Public Data Repositories

Where the data lives

- **NCBI SRA** — raw sequencing reads
- **NCBI GEO** — expression datasets
- **NCBI Refseq | Genbank** — genomes & annotations
- **NCBI datasets** — modern CLI for genomes

Tools you used

- **prefetch + fasterq-dump** — download & convert SRA data
- **datasets** — download genomes, proteins, CDS
- **wget** — download from Ensembl FTP
- **seqkit** — summarize & subsample

> Millions of publicly available datasets are just one command away. Always check the metadata (organism, library layout, read length) before downloading.

Quality Control: FastQC & fastp

FastQC — Assessment

- Per-base sequence quality
- GC content distribution
- Adapter contamination
- Overrepresented sequences
- Duplication levels

fastp — Processing

- Adapter auto-detection & removal
- Quality trimming (Q20, Q30)
- Minimum length filtering
- Comprehensive HTML report
- Fast multi-threaded processing

> Garbage in, garbage out. Quality control is not optional — it is the foundation of every reliable bioinformatics analysis.

Genome Alignment: STAR

The alignment workflow

- **Build index** — one-time step per genome
- **Align reads** — map FASTQ to reference
- **Inspect BAM** — samtools flagstat, view, index
- **Visualize** — IGV for interactive exploration

Key metrics

- **Uniquely mapped %** — should be >80%
- **Multi-mapped %** — should be <10%
- **Unmapped %** — should be <5%
- **Properly paired %** — for paired-end data

> Alignment connects your raw reads to their genomic context. The BAM file is the starting point for variant calling, expression counting, and many other analyses.

Sequence Search: BLAST

What BLAST does

- Finds similar sequences in large databases
- Identifies homologs across species
- Annotates unknown sequences
- Supports nucleotide & protein searches

What you learned

- **blastn / blastp** — nucleotide & protein search
- **Output format 6** — tabular, parseable results
- **E-value, % identity, bit score** — assessing hits
- **makeblastdb** — building local databases

> BLAST is often the first thing a bioinformatician reaches for when facing an unknown sequence. Command-line BLAST gives you full control and scalability over the web interface.

The Bioinformatics Workflow

Putting it all together — from raw data to biological insight:

Every step builds on the previous one. Understanding this workflow gives you the foundation to tackle RNA-seq, variant calling, metagenomics, and many other applications.

Where to Go from Here

Continue learning

- **RNA-seq** — differential expression with DESeq2
- **Variant calling** — GATK, bcftools
- **Metagenomics** — Kraken2, MetaPhlAn
- **Genome assembly** — SPAdes, Flye
- **R / Python** — for data analysis & visualization

Useful resources

- **Ibex documentation** — KAUST HPC resources
- **Tool --help** — read tool manuals
- **AI Solutions** — ChatGPT, Claude, etc
- **KAUST Bioinformatics Platform** — we are here to help!

We Want Your Feedback!

Your feedback helps us improve future workshops.
Please take 2 minutes to share your thoughts.

Scan the QR code above or click the link below:

Take the Survey

Thank You!

We hope you enjoyed the workshop and feel confident taking your first steps in bioinformatics.

The Bioinformatics Platform — KAUST

KAUST Academy 2026

THE BIOINFORMATICS PLATFORM