Introduction to Bio-Coding

Bio-coding, or coding for bioinformatics, involves using programming languages and tools to analyze and interpret biological data. As the field of bioinformatics grows, proficiency in bio-coding has become essential for researchers and professionals alike. Bio-coding typically involves the use of Linux, R, and Python—three powerful tools in the bioinformatics toolkit.

Why Linux for Bioinformatics?

Linux is an operating system similar to Windows or macOS, but it offers a higher degree of control over your system, such as managing RAM, processing power, and user access. Many bioinformatics tools and software are developed to run on Linux, largely because Linux is open-source, meaning the source code is freely available for anyone to modify. This open-source nature aligns with the collaborative spirit of the scientific community.

Linux (along with macOS) is heavily influenced by the Unix Unix-operating system. Unix is a powerful, multi user, multitasking operating system originally developed in the 1970s at AT&T's Bell Labs. Unix is known for its stability and efficiency, and is widely used in academic, business, and scientific computing environments.

Being both heavily influenced by Unix, Linux and macOS each incorporate Unix principles and structures in distinct ways. Here's a breakdown of their Unix heritage:

1. Linux

- **Unix-Like, Not Unix-Certified**: Linux was designed to be a Unix-like system, but it isn't directly based on original Unix code. Created by Linus Torvalds in 1991, Linux was developed as an open-source, Unix-inspired operating system kernel.
- **Modularity and Shell Commands**: Like Unix, Linux uses a modular design and includes a wide range of small, powerful command-line tools. It adheres closely to the Unix philosophy of "small, sharp tools" that can be combined to perform complex tasks.
- **File System and Permissions**: Linux adopts a Unix-like file system, with a hierarchical directory structure starting from the root directory (`/`). It also follows the same model of permissions and ownership for files, maintaining security and access control similar to Unix.
- **Shell Environment**: Linux uses the Bash shell by default (along with other shells like Zsh, Fish, etc.), which is derived from Unix shells. Bash scripting is nearly identical between Unix and Linux, allowing for similar commands, scripting, and automation.

2. macOS

- **Built on Unix**: macOS has a certified Unix foundation. It evolved from the NeXTSTEP operating system (developed by Steve Jobs' NeXT, which was based on the

Unix-derived BSD). When Apple acquired NeXT, this Unix-based foundation became the core of macOS.
- **POSIX Compliance**: macOS is POSIX-compliant, meaning it adheres to standardized Unix interfaces. This allows for Unix-certified functionality, making macOS a fully certified Unix OS. This compliance ensures compatibility and ease of use with many Unix tools and scripts.
- **File System and Permissions**: macOS uses a Unix-style file system structure and permissions system similar to Linux and traditional Unix systems. It recently transitioned to APFS (Apple File System), but the hierarchical structure and user permissions closely resemble those of Unix.
- **Terminal and Shell**: macOS provides a Terminal application where users can access a command-line interface with Unix commands. Until recently, macOS used Bash as its default shell, but it has since transitioned to Zsh. Both shells retain Unix-based functionality, making Unix command-line knowledge transferable to macOS.
- **Graphical and User-Friendly Interface**: While Unix traditionally lacked a graphical interface, macOS combines the Unix command-line environment with a highly user-friendly GUI (Graphical User Interface), making it approachable for general consumers while maintaining a Unix-based backend for developers and power users.

Unix Shell and Command Line Interface (CLI)

The shell is the command-line interpreter that enables users to interact with Unix by typing commands. Shells are like the machines' interpreters. Common Unix shells include:

- **Bash** (Bourne Again Shell)
- **Zsh** (Z Shell)
- **Tcsh** (an enhanced C shell)

Using the CLI, users can navigate the file system, manipulate files, and execute programs directly, making it a powerful environment for scripting and automation.

Advantages of Linux:

- **Control:** Gives users control over system resources.
- **Open Source:** Most bioinformatics tools are open source and developed for Linux.
- **Cost-Effective:** Linux is free, which makes it accessible to a wide range of users.

Understanding the Terminal and Bash

A **Terminal** is an interface that lets users interact directly with the operating system through text commands, without relying on a graphical user interface (GUI). In essence, it's a way to control a computer by typing commands instead of clicking icons or buttons.

Key Points About Terminals:

1. **Access to the Shell**: A terminal opens access to a “shell,” which is a program that interprets your commands. Each shell has unique features, but all allow basic file manipulation, program execution, and system management.
2. **Core Commands**: Terminals allow users to perform a wide range of operations, such as:
    - **Navigation**: Commands like `cd` (change directory) and `ls` (list files).
    - **File Management**: Commands like `cp` (copy), `mv` (move), and `rm` (remove).
    - **Permissions**: Commands like `chmod` and `chown` control access to files and directories.
3. **Automation and Scripting**: Terminals support shell scripting, allowing users to write scripts that automate repetitive tasks. This is especially powerful for developers, system administrators, and data scientists who need to execute sequences of commands quickly.
4. **Remote Access**: With tools like SSH (Secure Shell), terminals allow users to connect to and control remote systems securely, making it possible to manage servers and perform administrative tasks from anywhere.
5. **Efficiency and Control**: While terminals can have a learning curve, they provide precise control over the system and can be much faster than navigating through a GUI, especially for complex or repetitive tasks.

Terminals are foundational in Unix-like systems (like Linux and macOS) but are also available on Windows via PowerShell or Windows Subsystem for Linux (WSL), making them widely useful for power users and programmers across platforms.

Bash means Bourne Again Shell and it replaced the Bourne Shell in the Linux Project, which was the default shell for Unix Operating systems. A Shell makes it easy for the user to manage the operating system without knowing all of the inner workings and complexity of the operating system itself. Bash is currently the most popular shell scripting language for the Linux operating system. It’s also included in the MacOS and in Windows (when using subsystems for linux).

Key Points About Bash:

- **Shell vs. Operating System:** Bash is a shell (a command interpreter), while Linux is the operating system. Bash runs within Linux to interpret and execute user commands.
- **Scripting:** Bioinformaticians use Bash scripts to automate repetitive tasks, making it easier to manage large-scale data analysis.

Introduction to R and Python for Bio-Coding

While Linux provides the environment and Bash allows for command-line scripting, R and Python are the programming languages most commonly used in bioinformatics for data analysis and visualization.

- **R** is particularly strong in statistical analysis and data visualization.
- **Python** is known for its simplicity and versatility, making it ideal for scripting, data processing, and working with bioinformatics libraries like Biopython.

Combining Linux, R, and Python: The combination of these tools allows for powerful and flexible bio-coding, enabling researchers to manage, analyze, and visualize biological data effectively.

BIOINFORMATICS FILE FORMATS

Bioinformatics involves various file formats, each designed for specific data types and tasks. These formats can be broadly categorized based on their purposes and content types. Here's an overview of the most commonly used file formats every bioinformatician should know.

---

1. Sequence Data Formats

These formats are used to store raw or reference nucleotide and protein sequence data.

☐ FASTA

- **Purpose**: A text-based format for storing nucleotide (DNA/RNA) or protein sequences.
- **Extension**: `.fasta` or `.fa`
- **Content**: Each entry starts with a header line beginning with a `>` symbol, followed by the sequence data.
- **Use Cases**: FASTA files are used to store reference sequences, such as genomes, genes, or protein sequences. They are often used in sequence alignment, database searches, and phylogenetic analysis.

Example:

```
>sequence1

AGCTTAGCTAGCTACGATCG
```

☐ FASTQ

- **Purpose**: Text-based format representing both nucleotide sequences and quality scores from sequencing.
- **Extension**: `.fastq` or `.fq`
- **Content**: Each entry contains four lines:
    - Header line (`@` symbol and sequence identifier)
    - Sequence data
    - Plus line (`+` symbol)

    - Quality scores (ASCII-encoded for each nucleotide)
- **Use Cases**: FASTQ files are the standard format for raw sequencing data output from sequencers. The quality scores are essential for assessing the accuracy of the sequencing reads.

Phred Quality Scores in FASTQ files represent the accuracy of each base in a sequence read. They are encoded as ASCII characters (check https://www.ascii-code.com/) and indicate the probability that a given base is called incorrectly. Each score corresponds to a logarithmic probability value, where higher scores reflect greater confidence in the base call's accuracy.

For example, a Phred score of 20 means a 1 in 100 chance of error (99% accuracy), while a score of 30 indicates a 1 in 1000 chance of error (99.9% accuracy). These quality scores are essential for assessing the reliability of sequencing data before downstream analysis. ASCII characters represent **Phred Quality Scores** by assigning specific ASCII characters to each quality score. For instance, a Phred score of 20 is represented by the ASCII character **"5"** (decimal 53). Phred scores start at 0, which is represented by the character with ASCII value 33 (the "!" symbol) in the FASTQ format. To get the ASCII character for a Phred score, you simply add 33 to the score. This simple calculation reduces computational overhead during file encoding and decoding.

Example:

```
@sequence1

AGCTTAGCTAGCTACGATCG

+

!''*((((***+))%%%++)(****
```

---

2. Annotation and Feature Formats

These formats describe features or annotations in a genome.

- ☐ **GenBank**
- **Purpose**: Richly annotated text format for nucleotide and protein sequences, developed by NCBI.
- **Extension**: `.gb` or `.gbk`
- **Content**: Sequence data, along with annotations such as gene names, coding regions, and protein products.
- **Use Cases**: Widely used for storing genome annotations and sharing sequence data with annotations.

Example:

```
LOCUS       SCU49845     5028 bp    DNA             PLN
21-JUN-1999

DEFINITION  Saccharomyces cerevisiae TCP1-beta gene, complete cds.

ORIGIN

      1 agcttacgct tacggtacgg tacgctgtag cgtagctagc gtagctgacg
```

- ☐ **GFF/GTF (General Feature Format/General Transfer Format)**
- **Purpose**: Text-based formats for describing genomic features (e.g., gene locations).
- **Extension**: `.gff`, `.gtf`
- **Content**: Contains tab-separated fields for sequence name, source, feature type, start, end, score, strand, and additional attributes.
- **Use Cases**: Used to represent gene structures and other genomic features; popular in genome annotation pipelines.

Example:

```
seq1  .  gene  1300  9000  .  +  .  ID=gene00001;Name=my_gene
```

---

3. Variant Calling Formats

These formats are used to represent sequence variations.

- ☐ **VCF (Variant Call Format)**
- **Purpose**: Text-based format for storing information about genetic variants (e.g., SNPs, indels).
- **Extension**: `.vcf`
- **Content**: Each entry includes fields for chromosome, position, ID, reference allele, alternate allele, quality score, filter, and additional variant information.
- **Use Cases**: Commonly used for reporting variants in population genomics, cancer genomics, and precision medicine studies.

Example:

```
#CHROM POS ID REF ALT QUAL FILTER INFO
```

```
chr1    8973 .  G    A    29.5 PASS  AC=1;AF=0.5
```

- [ ] **BCF (Binary Call Format)**

- **Purpose**: Compressed binary version of VCF.
- **Extension**: `.bcf`
- **Content**: Binary representation of VCF fields.
- **Use Cases**: Used to efficiently store large-scale variant data; often used with tools like `bcftools` for variant analysis.

---

4. Alignment Formats

These formats store alignment information, showing how sequence reads map to reference genomes.

- [ ] **SAM (Sequence Alignment Map)**

- **Purpose**: Text format for storing read alignments to reference sequences.
- **Extension**: `.sam`
- **Content**: Includes fields for read name, alignment position, quality scores, and additional alignment data.
- **Use Cases**: Standard format for storing sequence alignment data from short-read aligners.

Example:

```
read001  16  chr1  7  30  100M  *  0  0  AGCTTAGCTAGCTACGATCG  !
```

- [ ] **BAM (Binary Alignment Map)**

- **Purpose**: Compressed binary version of SAM.
- **Extension**: `.bam`
- **Content**: Same fields as SAM but stored in a binary format for space efficiency.
- **Use Cases**: Preferred format for large alignment files; typically used in downstream analysis pipelines (e.g., variant calling).
- **Tools**: `samtools` is commonly used to manipulate BAM files.

- [ ] **CRAM (Compressed Reference-Aligned Map)**

- **Purpose**: Compressed format similar to BAM, but optimized to reduce file sizes by referencing external genomes.
- **Extension**: `.cram`
- **Use Cases**: Useful for large datasets, especially when storage is a concern; requires access to the reference genome to decode.

---

5. Region/Coordinate Formats

These formats represent regions of interest in a genome.

☐ **BED (Browser Extensible Data)**

- **Purpose**: Text format for storing chromosomal coordinates.
- **Extension**: `.bed`
- **Content**: Tab-delimited fields specifying chromosome, start, end positions, and optional additional information.
- **Use Cases**: Used in genomic data analysis to represent intervals, regions, or features on a genome.

Example:

```
chr1  100  500  feature1  0  +
```

☐ **BigWig**

- **Purpose**: Binary format for storing continuous-valued data (e.g., signal tracks).
- **Use Cases**: Efficiently stores large data files for quick visualization in genome browsers.

---

6. Quality Control and Metadata Formats

☐ **FastQC**

- **Purpose**: Software tool that assesses quality of sequencing data in FASTQ files. Even though FASTQ comes with a quality score, we don't really know the implications of these scores. FastQC gives a more detailed insight of the sequenced data.
- **Content**: FastQC does not generate sequence data itself but analyzes the quality of the sequence data in FASTQ files. It produces a report summarizing various quality metrics, such as:
    - Per base sequence quality
    - Per sequence GC content
    - Sequence length distribution

    - Overrepresented sequences
    - Adapter content
- **Use Cases:** FastQC is used to assess the quality of sequencing data before proceeding with further analysis. It helps identify issues such as low-quality reads, adapter contamination, or uneven base composition.
  Output: FastQC generates an HTML report that provides visualizations and summaries of the quality metrics. These reports help researchers decide whether the sequencing data is of sufficient quality for downstream analysis or if further processing (e.g., trimming low-quality bases) is needed.
- **Output Extension**: (not a file format, but produces output files) - `.html` and `.zip`
  Example Output: FastQC might report:
    - A graph showing the quality score distribution across all bases in the sequences.
    - Warnings or failures for certain quality metrics, indicating potential issues with the data.

☐ **BAM Index (BAI)**

- **Purpose**: Index file for BAM files.
- **Content**: Enables quick access to specific regions within BAM files.
- **Use Cases**: Required for efficient analysis and viewing of large BAM files in genome browsers or alignment viewers.

Understanding The Sequencing Process

What is Sequencing?

Sequencing is the process of determining the order of nucleotides (adenine, thymine, cytosine, and guanine) in DNA or RNA, or the order of amino acids in proteins. In genomics, DNA sequencing is used to understand the exact sequence of an organism's genome, giving insight into its genetic code. The sequence provides information about genes, regulatory regions, and other functional elements, helping researchers analyze how genes function, how mutations might lead to disease, and how organisms are related through evolution.

A **DNA library** is a collection of DNA fragments that have been prepared for sequencing or other molecular analyses. In the context of next-generation sequencing (NGS), a DNA library is created by fragmenting the genome or specific regions of interest and then adding specialized

adapter sequences to the ends of these fragments. These adapters are essential for binding the DNA to the sequencing platform and allow for amplification and sequencing of the fragments.

Here's a breakdown of the key steps involved in creating a DNA library:

1. **DNA Extraction and Fragmentation:**
    - The starting material is genomic DNA, which is extracted from cells.
    - The DNA is then fragmented into smaller pieces, typically ranging from 200 to 800 base pairs, either by mechanical shearing (e.g., sonication) or enzymatic digestion.
2. **Adapter Ligation:**
    - Specially designed short sequences called adapters are ligated (attached) to the ends of the fragmented DNA. These adapters enable the fragments to bind to the sequencing platform.
    - The adapters may also contain barcodes or indexes, allowing multiple samples to be pooled together in the same sequencing run (multiplexing).
3. **Amplification (Optional):**
    - In some sequencing protocols, the library is amplified using PCR to generate enough material for sequencing. This step can introduce some bias, so it's sometimes avoided in protocols like long-read sequencing.
4. **Size Selection and Cleanup:**
    - The library is typically size-selected to remove excessively large or small fragments. This ensures that the fragments are of the optimal length for the sequencing platform.
    - The DNA library is purified to remove any unligated adapters or unwanted fragments.
5. **Sequencing:**
    - Once the DNA library is prepared, it is loaded onto a sequencing platform where the actual sequencing occurs. The adapter sequences allow the fragments to bind to the platform and initiate sequencing reactions.

Some DNA material can be lost during the fragmentation process, but the extent of loss depends on the method used and how carefully the process is controlled. Fragmentation is a key step in preparing a DNA library for sequencing, but it can cause some sample loss due to the following reasons:

1. **Mechanical Shearing:**
    - When mechanical methods like sonication or nebulization are used to break DNA into smaller fragments, the process is somewhat random, and not all the DNA fragments may be of the desired size.
    - Some fragments may be too short or too long and are usually discarded during the size selection step.
2. **Enzymatic Fragmentation:**

    - Enzymatic digestion (using restriction enzymes or transposases) can lead to DNA loss if the enzymes don't cut the DNA uniformly or if fragments generated are too small or not suitable for the next steps.
3. **Cleanup Steps:**
    - During the fragmentation process, several cleanup and purification steps (e.g., using magnetic beads or gel electrophoresis) are performed to remove unwanted fragments and reagents. Some DNA material may be lost during these steps as well.
4. **Size Selection:**
    - After fragmentation, DNA is typically size-selected to ensure fragments are of the optimal length for sequencing. DNA fragments outside the desired size range are discarded, leading to loss of some material.
5. **Low Input DNA Samples:**
    - If the starting amount of DNA is low, some loss during fragmentation can make it challenging to recover enough DNA for downstream applications, which is why low-input kits or protocols are used in these cases.

While some DNA loss is inevitable, the process is optimized to minimize significant loss and ensure enough usable DNA remains for successful library preparation and sequencing.

Paired-End Sequencing

Paired-end sequencing is a sequencing technique where both ends of DNA fragments are sequenced, providing two reads for each fragment—one from each end. This technique is widely used in next-generation sequencing (NGS) and has significant advantages over single-end sequencing.

Key Features:

1. **Dual Read Generation**:
    - Both ends of the DNA fragment are sequenced, producing two reads (forward and reverse) for each fragment.
    - The length of these reads depends on the sequencing technology and platform used (e.g., 150 base pairs per read on Illumina platforms).
2. **Enhanced Mapping Accuracy**:
    - Since reads come from both ends of a fragment, they can be aligned more accurately to a reference genome, improving the detection of structural variations, indels, and duplications.
3. **Longer Overall Coverage**:
    - While each individual read might not cover the entire fragment, the combined information from both ends can span longer genomic regions.
    - This feature is especially useful in resolving repetitive regions or difficult-to-sequence regions.
4. **Applications**:

    - **De Novo Genome Assembly**: Helps in assembling genomes without a reference by providing greater confidence in mapping.
    - **Structural Variation Detection**: Enables better identification of insertions, deletions, duplications, and translocations.
    - **RNA Sequencing**: Useful in transcriptomics for identifying splice variants and quantifying gene expression.

Advantages:

- **Higher Accuracy**: Paired-end reads help resolve ambiguities, especially in repetitive sequences, by confirming the location of fragments.
- **More Information**: Provides more data for genome assembly and variant detection by reading from both ends.
- **Effective with Low-Quality DNA**: Useful for samples where the DNA may be fragmented.

Limitations:

- **Higher Cost**: Since it generates twice the amount of data, paired-end sequencing can be more expensive than single-end sequencing.
- **Increased Data Handling**: Requires more computational resources to process the additional reads and perform alignments.

Other Types of Sequencing

1. **Single-End Sequencing**:
    - **Description**: Only one end of the DNA fragment is sequenced.
    - **Applications**: Used in applications where sequence alignment is straightforward and where structural variants or repeat regions aren't the focus.
    - **Advantages**: Cheaper and simpler than paired-end sequencing with fewer data processing requirements.
    - **Limitations**: Less accurate for complex regions of the genome; harder to detect structural variants.
2. **Mate-Pair Sequencing**:
    - **Description**: A type of sequencing where the ends of long DNA fragments (typically >2kb) are sequenced. The large insert size allows distant regions of the genome to be linked.
    - **Applications**: Useful for genome assembly, resolving large structural variations, and sequencing repetitive regions.
    - **Advantages**: Can span large structural variations and distant regions that are difficult to resolve with paired-end sequencing.
    - **Limitations**: More complex library preparation and data analysis.
3. **Whole Genome Sequencing (WGS)**:

- **Description**: A comprehensive approach that sequences the entire genome of an organism.
- **Applications**: Used in genetic research, evolutionary biology, clinical genomics, and personalized medicine.
- **Advantages**: Provides a complete genetic map, allowing detection of all variants, including SNPs, indels, and structural variations.
- **Limitations**: Requires high coverage for accurate variant detection, making it costly and data-intensive.

4. **Exome Sequencing (WES)**:
   - **Description**: Focuses on sequencing the coding regions of the genome (exons), which represent about 1-2% of the genome.
   - **Applications**: Frequently used in clinical genomics for identifying disease-causing mutations in protein-coding regions.
   - **Advantages**: Less expensive and data-intensive than WGS, but still highly informative for identifying mutations in exons.
   - **Limitations**: Misses variations in non-coding regions that could still be biologically significant.
5. **Targeted Sequencing**:
   - **Description**: Sequencing a specific subset of genes or genomic regions, such as genes associated with a particular disease.
   - **Applications**: Cancer genomics, pathogen sequencing, or any scenario where only specific genes are of interest.
   - **Advantages**: More cost-effective and data-efficient than WGS or WES when only a few regions need to be sequenced.
   - **Limitations**: Limited to preselected regions, so novel variants outside the target may be missed.
6. **RNA Sequencing (RNA-Seq)**:
   - **Description**: Sequencing of RNA transcripts to study gene expression, alternative splicing, and transcriptome assembly.
   - **Applications**: Widely used in transcriptomics, differential gene expression studies, and identification of splice variants.
   - **Advantages**: Provides quantitative and qualitative insights into the transcriptome.
   - **Limitations**: Requires careful preparation of RNA libraries and has biases due to differences in transcript abundance.
7. **Nanopore Sequencing**:
   - **Description**: A long-read sequencing technology that passes a DNA or RNA molecule through a nanopore and measures changes in electrical current to determine the sequence.
   - **Applications**: De novo genome assembly, metagenomics, and sequencing of long repetitive regions.
   - **Advantages**: Capable of producing very long reads, making it easier to resolve complex regions of the genome.

    - **Limitations**: Lower accuracy compared to short-read technologies like Illumina, although improvements are ongoing.
8. **Third-Generation Sequencing (e.g., PacBio, Oxford Nanopore)**:
    - **Description**: Refers to single-molecule, real-time (SMRT) sequencing technologies that produce long reads and sequence entire molecules.
    - **Applications**: Ideal for assembling complex genomes, resolving structural variants, and sequencing full-length RNA transcripts.
    - **Advantages**: Generates much longer reads than second-generation technologies, simplifying assembly and variant detection.
    - **Limitations**: Higher error rates and higher costs, though error correction techniques are improving.

Comparison Summary:

Sequencing Type	Read Length	Accuracy	Cost	Applications
Paired-End Sequencing	Short (e.g., 150 bp)	High	Moderate to High	De novo assembly, structural variant detection
Single-End Sequencing	Short (e.g., 150 bp)	Moderate	Low	Simple alignment, gene expression studies
Mate-Pair Sequencing	Long (2-5 kb fragments)	Moderate to High	High	Long-range assembly, structural variants
Whole Genome Sequencing	Short	High	High	Full genome analysis
Exome Sequencing	Short	High	Moderate	Disease gene discovery
Targeted Sequencing	Short	High	Low to Moderate	Specific gene mutation analysis
RNA Sequencing (RNA-Seq)	Short to Long	High	Moderate	Gene expression, transcriptome studies
Nanopore Sequencing	Long (up to megabases)	Moderate	High	Long-read sequencing, de novo assembly

Each sequencing type has its specific applications depending on the desired level of resolution, accuracy, and the biological question being addressed.

Next-Generation Sequencing

Next-Generation Sequencing (NGS) refers to a suite of advanced technologies that enable the rapid and high-throughput sequencing of DNA or RNA. Unlike earlier sequencing methods, NGS can sequence millions to billions of nucleotides in parallel, making it possible to sequence entire genomes, transcriptomes, or specific regions of interest much faster and at a lower cost than previous methods.

Key Features of NGS:

1. **High Throughput:**
    - NGS platforms can generate massive amounts of data in a single run, sequencing multiple samples simultaneously. This high throughput allows for comprehensive studies, such as whole-genome sequencing (WGS) or transcriptome analysis.
2. **Parallel Sequencing:**
    - NGS technologies read millions of DNA fragments in parallel, significantly increasing speed and efficiency compared to traditional methods like Sanger sequencing, which sequences one DNA fragment at a time.
3. **Scalability:**
    - NGS is highly scalable, meaning it can be used for small-scale projects like sequencing a few genes or large-scale projects like sequencing entire genomes or populations.
4. **Cost-Effective:**
    - The cost of sequencing has dramatically decreased with the advent of NGS, making it accessible for a wide range of research and clinical applications.

How NGS Works:

1. **Library Preparation:**
    - DNA or RNA samples are fragmented into smaller pieces, and adapters are added to both ends of each fragment. These adapters allow the fragments to be amplified and sequenced.
2. **Amplification:**
    - The prepared library is amplified using PCR (Polymerase Chain Reaction) to generate multiple copies of each fragment. Some NGS platforms use other amplification methods.
3. **Sequencing:**
    - The amplified DNA fragments are sequenced in parallel, with the NGS platform reading the sequence of nucleotides (A, T, C, G) in each fragment.
4. **Data Analysis:**
    - The raw sequence data generated by the NGS platform is processed and aligned to a reference genome (if available) or assembled de novo. Bioinformatics tools

are used to analyze the sequences, identify variants, detect gene expression levels, or study epigenetic modifications.

Applications of NGS:

1. **Whole-Genome Sequencing (WGS):**
    - Sequencing the entire genome of an organism to study genetic variations, mutations, and structural changes.
2. **Targeted Sequencing:**
    - Focusing on specific regions of the genome, such as exons or specific genes, to identify mutations or variants related to diseases.
3. **RNA Sequencing (RNA-Seq):**
    - Sequencing the transcriptome to study gene expression, identify novel transcripts, and understand the regulatory mechanisms of gene expression.
4. **Epigenetic Studies:**
    - Analyzing DNA methylation patterns and other epigenetic modifications to understand gene regulation and the impact of the environment on gene expression.
5. **Metagenomics:**
    - Sequencing the DNA of microbial communities in a given environment to study biodiversity, identify species, and understand ecological interactions.
6. **Cancer Genomics:**
    - Sequencing tumor DNA to identify mutations, study clonal evolution, and guide personalized cancer treatment.

NGS Platforms:

Some of the most widely used NGS platforms include:

- **Illumina**: Known for short-read sequencing with high accuracy.
- **PacBio**: Provides long-read sequencing, useful for resolving complex regions of the genome.
- **Oxford Nanopore**: Offers portable sequencing devices and long-read sequencing capabilities.

Impact of NGS:

Next-Generation Sequencing has revolutionized genomics and molecular biology, enabling a deeper understanding of genetics, disease mechanisms, and evolutionary biology. It has broad applications in research, clinical diagnostics, agriculture, forensics, and personalized medicine, among other fields.

Working with Bioinformatic Databases

Bioinformatics databases are indispensable resources that store and organize extensive data, including nucleotide sequences, protein structures, and functional annotations, aiding researchers in a variety of analyses. Two prominent organizations, the **National Center for Biotechnology Information (NCBI)** and the **European Bioinformatics Institute (EBI)**, house a wide range of databases covering various biological data. Here's an overview of essential bioinformatic databases:

Major Bioinformatics Database Providers

1. NCBI Databases

- **Total Databases**: Over 30 specialized databases.
- **Notable Databases**:
    - **GenBank**: A comprehensive nucleotide sequence repository.
    - **RefSeq**: A curated database of reference sequences for genes and proteins across species.
    - **BLAST**: A suite of tools for sequence alignment and similarity search.
    - **Gene**: Provides gene-specific data across organisms.
    - **PubMed**: A database of biomedical literature.
    - **dbSNP**: Information on single nucleotide polymorphisms (SNPs) and small genetic variations.
    - **ClinVar**: Variants linked to medical conditions.
    - **Protein**: Protein sequence and functional data.
    - **dbVar** and **dbGaP**: Databases focused on structural variations and genotypic/phenotypic associations.
- **Scope**: NCBI databases cover a comprehensive range of genomic, gene function, variant, literature, and pathway information, with a primary focus on human and model organism genomics.

2. EBI Databases

- **Total Databases**: Over 40 main databases, often integrating multiple data types.
- **Notable Databases**:
    - **Ensembl**: Genome annotations for vertebrates and model organisms.
    - **UniProt**: Protein sequences and functional annotations (in collaboration with SIB and PIR).
    - **ArrayExpress**: Functional genomics data from experiments.
    - **EMBL-EBI Genomes**: Nucleotide sequence data.
    - **Reactome**: Pathways and molecular interaction networks.
    - **ChEMBL**: Bioactive compounds and their properties.

    - **InterPro**: Protein signatures and functional data.
    - **PRIDE**: A repository for proteomics data.
    - **ENA**: The European Nucleotide Archive, housing sequence data.
- **Scope**: EBI databases encompass a broad range of data, including genome, transcriptome, proteome, structural, chemical biology, and pathway information.

Essential Databases in Bioinformatics

Genomic Databases

1. **RefSeq (Reference Sequence Database)**
    - **Provider**: NCBI
    - **Purpose**: Offers a curated, non-redundant set of reference sequences for genomes, genes, and proteins.
    - **Use Cases**: Standard reference in clinical genomics, variant analysis, and comparative studies.
2. **Ensembl**
    - **Provider**: EBI and Wellcome Sanger Institute
    - **Purpose**: Annotates genomes with a focus on vertebrates, supporting comparative and functional genomics.
    - **Use Cases**: Useful for evolutionary studies and comparative genomics.
3. **UCSC Genome Browser**
    - **Provider**: University of California, Santa Cruz
    - **Purpose**: Provides an interactive genome browser focused on human and model organisms.
    - **Use Cases**: Widely used in cancer genomics, structural annotation, and sequence visualization.

Variant and Mutation Databases

4. **dbSNP**
    - **Provider**: NCBI
    - **Purpose**: Repository for SNPs and small genetic variations.
    - **Use Cases**: Key for SNP analysis and population genomics research.
5. **ClinVar**
    - **Provider**: NCBI
    - **Purpose**: Provides clinically relevant genetic variants associated with disease.
    - **Use Cases**: Critical in clinical genomics for understanding the role of variants in disease.

Protein and Functional Annotation Databases

6. **UniProt**

- **Provider**: EBI (in collaboration with SIB and PIR)
- **Purpose**: Comprehensive resource for protein sequences and functional data.
- **Notable Databases**:
  - **Swiss-Prot**: Curated protein data.
  - **TrEMBL**: Translated protein sequences.
- **Use Cases**: Widely used in protein function analysis, structure studies, and interaction networks.

7. **Pfam**
   - **Provider**: EBI
   - **Purpose**: Database of protein families and domains based on conserved sequences.
   - **Use Cases**: For understanding protein domains and identifying protein family memberships.

Gene Ontology (GO) and Pathway Databases

8. **Gene Ontology (GO)**
   - **Purpose**: Provides controlled vocabulary for describing gene functions, biological processes, and cellular components.
   - **Use Cases**: Essential in functional genomics and pathway enrichment analysis.
9. **Reactome**
   - **Provider**: EBI
   - **Purpose**: Curated pathway database covering molecular interactions and reaction networks.
   - **Use Cases**: Useful in pathway analysis, drug discovery, and understanding cellular processes.

Expression and Structural Databases

10. **GEO (Gene Expression Omnibus)**
    - **Provider**: NCBI
    - **Purpose**: Repository for gene expression and other high-throughput functional genomic data.
    - **Use Cases**: Useful for differential expression analysis and gene regulation studies.
11. **PDB (Protein Data Bank)**
    - **Purpose**: Repository of 3D structures of proteins, nucleic acids, and molecular complexes.
    - **Use Cases**: Essential in structural biology, drug design, and protein-ligand studies.

Other Major Databases

- **DDBJ (DNA Data Bank of Japan)**: Part of the INSDC, sharing sequence data with GenBank and EMBL-EBI's ENA.
- **JGI Genome Portal**: Hosts sequencing data from plants, microbes, and metagenomes, developed by the Joint Genome Institute.

Overview of RNA-seq

RNA-Sequencing (RNA-Seq) is a powerful tool used to study the transcriptome of an organism. It provides a comprehensive view of gene expression and allows for the identification of novel transcripts, alternative splicing events, and post-transcriptional modifications. RNA-Seq has revolutionized the field of transcriptomics and has become a widely used technique in molecular biology research.

A novel transcript is simply a transcript that has not been observed before. If the transcript structure you observe is not currently represented in RefSeq, Ensembl, or UCSC, there is a good chance it might be novel.

To start the RNA-Seq process:

- **RNA Isolation:** The first step is to isolate the RNA from a sample. This can be done using a variety of methods, such as TRIzol extraction or RNA purification kits.
- **cDNA Synthesis:** The isolated RNA is then converted into cDNA (a complementary DNA copy of the RNA). This is done using a reverse transcriptase enzyme, which synthesizes cDNA from the RNA template. This is because the RNA is naturally not stable and can be degraded easily.
- **Library Preparation:** The cDNA is then fragmented into smaller pieces, and adapters are ligated to the ends of the fragments. The adapters are short sequences of DNA that allow the fragments to be attached to a sequencing platform.

  Genomic DNA ⇒ DNA Fragments ⇒ Unpolished DNA Fragments ⇒ Polishing of DNA Fragments & Addition of A Bases ⇒ PCR (Optional) ⇒ Sequencing

- **Sequencing:** The fragmented DNA is then sequenced using a Next-Generation sequencing platform. NGS platforms can sequence millions or billions of DNA or RNA molecules in a single run.
- **Data Analysis**: The resulting reads are now aligned to a reference genome or reference transcriptome. The alignment of the reads to the reference genome allows the expression levels of genes and transcripts to be calculated, and these expression levels can then be visualised and analysed using a variety of bioinformatic software tools.

Align Reads Identify Variants

FASTQ ==========> BAM ==========> VCF

There are 2-3 things to mainly consider when conducting RNA-Seq analysis:

1. The quality of the RNA file has to be very good.
2. The number of samples used for the analysis, because the statistics involved in RNA-Seq analysis requires at least 3-4 samples (either wetlab samples or in-silico samples) for accuracy of results.
3. The coverage of the samples.

Uses of RNA-Seq

- It can be used to measure the expression levels of thousands or even millions of genes in a single experiment.
- It can be used to detect novel transcripts and study the structure of transcripts
- It can be used to identify genes that are differentially expressed between two or more conditions.
- It can be used to study the transcriptome of individual cells, which can provide insights into the heterogeneity of cell populations. We have the single cell RNA-Seq techniques, as well as the single nucleus RNA-Seq techniques.

You'll analyze RNA-seq data using Bash scripting, starting with simple commands and building towards more complex workflows.

Windows Subsystem for Linux (WSL).

Setting Up Linux on Windows

If you are using Windows and want to leverage Linux, you can use the "Windows Subsystem for Linux (WSL)" to run Linux directly on your Windows machine.

Steps to Set Up Linux on Windows:

1. **Turn on WSL:**
    - Go to the Windows search bar and type "Turn Windows features on or off."
    - Check the box for "Windows Subsystem for Linux."

    - Restart your system.
2. **Install Ubuntu:**
    - After restarting, go to the Microsoft Store.
    - Search for and install **Ubuntu 22.04.3**, a popular Linux distribution.

Once installed, you can access Ubuntu (and its terminal) from the Start menu **(run as an administrator)**, allowing you to run Linux commands on your Windows system.

wsl --set-default-version 2

https://aka.ms/wsl2kernel