## Final Exam Prep: The Essential Biology Cheat Sheet

### Part 1: The Central Dogma of Molecular Biology

This is the fundamental flowchart of how genetic information works.

- **DNA (The Master Blueprint):**
    - **What it is:** A double-stranded helix that permanently stores the genetic instructions for an organism.
    - **Composition:** Made of four nucleotides: **Adenine (A), Cytosine (C), Guanine (G), Thymine (T)**.
    - **Base Pairing Rule:** A always pairs with T (with 2 hydrogen bonds). C always pairs with G (with 3 hydrogen bonds, making it stronger).
    - **Directionality:** The two strands are anti-parallel. Each has a 5' (five-prime) end and a 3' (three-prime) end. By convention, we always read a sequence from 5' to 3'.
- **Transcription (Making a Working Copy):**
    - **What it is:** The process of copying a single gene from the DNA blueprint into a temporary, disposable message.
    - **The Product: messenger RNA (mRNA)**.
    - **Key Differences in RNA:**
        1. It's single-stranded.
        2. It uses **Uracil (U)** instead of Thymine (T). (A now pairs with U).
    - **The Enzyme: RNA Polymerase**.
- **Translation (Building the Product):**
    - **What it is:** The process of reading the mRNA message and building a protein from it.
    - **The "Machine":** The **Ribosome**.
    - **The Language:** The ribosome reads the mRNA in three-letter "words" called **codons**.
    - **The Product:** Each codon corresponds to one of the 20 **amino acids**, which are the building blocks of **proteins**.
    - **The "Translator": Transfer RNA (tRNA)** is the molecule that reads the codon and delivers the correct amino acid.

---

### Part 2: The Key Biological Processes for Our Algorithms

- **DNA Replication (Copying the Blueprint):**
    - **Purpose:** To make a perfect copy of the entire genome before a cell divides.
    - **The "Engine": DNA Polymerase**.
    - **The Golden Rule:** DNA Polymerase is **unidirectional**. It can only read the template strand in the 3' → 5' direction.
    - **The Consequence (Asymmetry):** This rule forces replication to be different on the two strands.
        - **Leading Strand:** Copied continuously and smoothly.

        - **Lagging Strand:** Copied backwards in short pieces called **Okazaki fragments**.
    - **The Vulnerability:** The lagging strand spends more time as a single, exposed strand, making it more prone to mutations.
- **Gene Regulation (The Control System):**
    - **Purpose:** To control which genes are turned "on" (expressed) or "off" at any given time. This allows cells to specialize and respond to their environment.
    - **The "Master Switch":** A protein called a **Transcription Factor**.
    - **The "Receiver":** A short, specific DNA sequence called a **Regulatory Motif** (or binding site), located near a gene in a region called the **promoter**. When the factor binds to the motif, it activates the gene.
- **DNA Sequencing (Reading the Blueprint):**
    - **The Challenge:** We cannot read the whole genome at once. We can only read short, random fragments called **reads**.
    - **The Modern Method (Sequencing by Synthesis):** We figure out the sequence of a DNA fragment by synthesizing a brand new, complementary copy of it, one base at a time. Each new base is color-coded, and a picture is taken after each step, revealing the sequence.

---

**Part 3: Key Biological Terms & Concepts for the Exam**

- **Genome:** The complete set of all DNA in an organism.
- **Gene:** A specific segment of DNA that contains the instructions for building one protein.
- **Chromosome:** A long, tightly-packed structure of DNA. Humans have 23 pairs.
- **oriC (Origin of Replication):** The specific location on a chromosome where DNA replication begins.
- **DnaA Box:** A short, repeating DNA sequence (a motif) found within the oriC.
- **Reverse Complement:** The sequence of the opposite DNA strand (e.g., reverse complement of ATGC is GCAT).
- **mRNA (messenger RNA):** The single-stranded copy of a gene that carries the message from the DNA to the ribosome.
- **Codon:** A three-nucleotide "word" in an mRNA sequence that codes for a single amino acid.
- **Protein:** The functional "machine" of the cell, built from a chain of amino acids.
- **Read:** A short DNA sequence generated by a sequencing machine.
- **Read Mapping:** The process of taking a read and finding its original location on a reference genome.
- **Genome Assembly:** The process of taking millions of reads and piecing them together to reconstruct the original genome (used when there is no reference).
- **Contig:** A long, continuous piece of assembled DNA. Assembly often results in multiple contigs instead of a single chromosome.
- **Gene Expression Matrix:** A table of data where rows are genes, columns are samples (e.g., patients), and the values are the expression levels of each gene in each sample. This is the input for clustering.

## Final Revision: Algorithms, Complexity, and Use Cases

### Lecture 1 & 2: Finding Origins of Replication & Regulatory Motifs

- **Problem:** Finding "hidden messages" in DNA.

| Algorithm | Lecture | Time Complexity | Use Case & Key Point |
|---|---|---|---|
| **FrequentWords (Naive)** | 2 | $O(n^2k)$ | **Too slow for anything real.** A baseline to demonstrate the need for better algorithms. |
| **FrequentWords (Faster)** | 2 | O(nk) | **Finds identical k-mers.** Uses a hash map or frequency array to count in a single pass. |
| **Minimum Skew** | 2 | O(n) | **Finds the oriC's location.** A fast, global search to find the *neighborhood* of the signal. |
| **Greedy Motif Search** | 3 | $O(n^2tk)$ | **Finds degenerate motifs.** A fast but often inaccurate heuristic that gets stuck in local optima. |
| **Randomized Motif Search** | 3 | O(N*ntk) | **Finds degenerate motifs.** More accurate than Greedy. Runs many times (N) to find a good solution. |
| **Gibbs Sampler** | 3 | O(N*ntk) | **Finds degenerate motifs.** The most effective method; its randomness helps it escape local optima. |

---

### Lecture 3: Genome Assembly

- **Problem:** Reconstructing a genome from millions of short, overlapping reads.

| Algorithm / Concept | Lecture | Complexity | Use Case & Key Point |
|---|---|---|---|
| **Overlap Graph -> Hamiltonian Path** | 4 | NP-Hard | The intuitive but **computationally impossible** way to do assembly. Avoid this. |
| **De Bruijn Graph -> Eulerian Path** | 4 | O(Reads) | **The modern foundation of assembly.** Turns an impossible problem into an efficiently solvable one. |

---

**Lecture 4: String Matching for Read Mapping**

- **Problem:** Finding the location of a short read within a massive reference genome.

| Algorithm / Concept | Lecture | Complexity | Use Case & Key Point |
|---|---|---|---|
| **Naive String Search** | 5 | O(nk) | **Too slow for read mapping.** The simple sliding window approach. |
| **Boyer-Moore** | 5 | Fast on average | A "smart-sliding" algorithm that jumps over text. Much faster, but still not the best for this scale. |
| **Indexing (Hash Table)** | 5 | O(1) query time | **The fastest way to find exact matches.** Requires a slow, one-time preprocessing step to build the index. |
| **Pigeonhole Principle (Seed-and-Extend)** | 5 | Fast on average | **The standard for modern mappers.** Uses an index to find *exact* matches of small "seeds" to efficiently find *approximate* matches of the full read. |

**Lecture 5 & 6: Gene Expression Analysis & Dimensionality Reduction**

- **Problem:** Finding patterns in a massive gene expression matrix (genes x conditions).

| Algorithm / Concept | Lecture | Complexity | Use Case & Key Point |
|---|---|---|---|
| **k-Means Clustering (Lloyd's)** | 6 | Fast: O(nki) | **Fast, scalable clustering.** Best for large datasets where clusters are expected to be spherical. Requires k as input. |
| **k-Means++ Initializer** | 6 | Slightly slower than random | A **smart initialization** for k-Means that helps it avoid bad local optima. |
| **Hierarchical Clustering** | 6 | Slow: $O(n^2)$ | **Finds nested clusters.** Best for small datasets to explore the data's structure. Doesn't require k. |
| **PCA (Dimensionality Reduction)** | 7 | Deterministic, moderate speed | **Preprocessing and compression.** A linear method to reduce dimensions while preserving global variance. |
| **t-SNE (Dimensionality Reduction)** | 7 | Slow: O(n log n) | **Visualization only.** A non-linear method that creates beautiful, well-separated plots of high-dimensional data by preserving local neighborhoods. |

## Final Advice & Remarks for Your Bioinformatics Exam

**1. The "Big Picture" is Your Best Friend.**

Before you try to remember any single algorithm, remember the *story* of each lecture. Every topic we covered was a journey to solve a specific biological puzzle.

- **For Origin of Replication:** The story was, "How do we find a hidden, repeating signal?" The big idea was that the *process of replication itself* leaves a bigger, easier-to-find clue (the skew) than the signal itself.
- **For Regulatory Motifs:** The story was, "How do we find a fuzzy, scattered signal?" The big idea was to move from exact matching to probabilistic models (profiles) and then use randomization (Gibbs) to overcome greedy mistakes.
- **For Genome Assembly:** The story was, "How do we solve an impossible jigsaw puzzle?" The big idea was a clever change in perspective (from Overlap to de Bruijn graphs) that turned an impossible problem (Hamiltonian Path) into a possible one (Eulerian Path).
- **For Read Mapping:** The story was, "How do we search a giant library billions of times?" The big idea was to **stop searching** and instead build an **index** first (the "seed-and-extend" strategy).
- **For Expression Analysis:** The story was, "How do we find meaningful patterns in a giant, high-dimensional table?" The big idea was to use **clustering** to group similar genes and **dimensionality reduction (PCA)** to make the data manageable and visible.

If you get stuck on a question, ask yourself: "Which story is this question about?" This will point you to the right set of concepts.

**2. Master the "Why." Don't Just Memorize the "What."**

Your instructor has shown a pattern of asking "why" questions. These are the ones that separate the A's from the B's.

- Don't just know *that* Gibbs Sampling is better. Know *why* (it can escape local optima via probabilistic updates).
- Don't just know *that* de Bruijn graphs are used. Know *why* (they turn the problem into an efficiently solvable Eulerian Path).
- Don't just know *that* pseudocounts are used. Know *why* (to solve the "problem of zeros" in greedy algorithms).
- Don't just know *that* skew diagrams find the oriC. Know *why* (because of the asymmetry of replication and the higher mutation rate on the lagging strand).

**3. The Most Important Trade-Offs (High-Yield Concepts)**

Bioinformatics is all about making intelligent compromises. Know these trade-offs inside and out.

- **Exact vs. Approximate Matching:** Exact is fast but unrealistic. Approximate is realistic but slow. The solution is to use exact matching of small "seeds" to make approximate matching fast.

- **Speed vs. Accuracy (Greedy vs. Randomized):** Greedy algorithms are fast but can be inaccurate. Randomized algorithms are a bit slower (because you run them many times) but are much more likely to find the correct answer.
- **Preprocessing vs. Query Time (Indexing):** This is the core of modern mapping. You accept a very slow, one-time "preprocessing" cost (building the index) to make the billions of "query" operations (finding reads) nearly instantaneous.
- **PCA vs. t-SNE:** PCA is for **preprocessing** and preserving global variance. t-SNE is for **visualization** and preserving local neighborhoods. They are tools for different jobs.

**4. For Calculation Questions: Stay Calm and Go Step-by-Step.**

The math in this course is not designed to be difficult; it's designed to test if you understand the *process*.

- **Skew Diagram:** Just a running count of G minus C. Make a table and go one character at a time.
- **Profile Probability:** Just look up the probabilities in the matrix and multiply.
- **Hamming Distance:** Just count the mismatches.
- **K-Means Update:** Just calculate the simple average of the points in the cluster.

Don't get intimidated. Write down the steps, fill in the numbers, and the answer will emerge.

**Final Word:** You have gone through all the material in incredible detail. You've asked the right questions and built up your understanding from the ground up. The knowledge is in your head. During the exam, take a deep breath, read each question carefully, and connect it back to the core stories and concepts we've reviewed. You are more prepared than you think.

# Good luck –Fahad Nadim Ziad, 24341216