# 

# **A new generation of DNA hidden repeats detection algorithm and its application for isochore research**

# **Project Number: 25-2-R-8**

# **Capstone Project Phase A**

# 

# **Fatmeh Zoabi || Email: Fatmehzo3bi10@gmail.com**

# **Khalil Mansour || Email: Khalel.Mnsor@e.braude.ac.il**

# **Supervisor: Dr. Zakharia Frenkel**

# 

# **Keywords:**

# **Text segmentation, DNA sequence analysis,**

# **embedded repeats, GC-content Bayes,**

# **statistical estimation**

# 

**Contents**

**Abstract .....................................................................................................................2**

**1. Introduction ...........................................................................................................2**

**2. Project Background and Objective......................................................................2  
 2.1. DNA Repeats and Their Biological Significance..........................................2  
  2.1.1 Classification of DNA Repeats .......................................................................2**

### **2.1.2 Hidden or Low-Complexity Repeat Detection...............................................3**

### **2.2. Isochore Structures and Genomic Organization…………………………...5**

#### **2.2.1. Isochores and Their Significance.................................................................5**

#### **2.2.2. Isochore Detection Algorithm.......................................................................6**

**2.2.3. Isochores and Segments Alignment Test………………………....................6**

**2.3 Insights from the Research of Frenkel, Trifonov, and Collaborators…...6**

**2.4 Prior Study………………………………………………………………...……….7**

**2.4.1. Foundation in Previous Research Study…………………………….……...8**

**2.4.2. Experiment Data and Methodology of the Previous Work……………….8**

**2.5. Objectives of the Current Study………………………………………………....8**

**2.6. Background Conclusion………………………………………………….……….9**

**3. Our Proposed Algorithm……………………………………………………………….9**

**3.1. Source Work..........................................................................................................9**

**3.2. Connection to our project....................................................................................9**

**3.3. Updating Segmentation Algorithm...................................................................10**

**3.4. P-value incorporation.........................................................................................10**

**3.5. Algorithm Overview............................................................................................11**

**3.6 Summary Algorithm............................................................................................11**

**4. Project requirements...........................................................................................19**

**4.1. Functional Requirements..................................................................................19**

**4.2. Non-Functional Requirements..........................................................................20**

## **5. Description of the Research Process................................................................21**

### **5.1 Research Approach...........................................................................................21**

**5.2. Visual Summary of the Proposed Algorithm.................................................22**

### **6. Expected Challenges in Our Capstone Project……………………....................23**

**7. Project Conclusion..............................................................................................24**

### **7.1. Description of Algorithms / Testing Process ..................................................24**

### **7.2. Description of Project Success Metrics..........................................................25**

**8. Methodological Tools and Resources...............................................................25**

### **8.1 Research and Writing Tools...............................................................................25**

**8.2. Links to AI Tools Used and Prompts…………………………..……...................25**

**9. References…………………………….………………………………………………..27**

**Abstract**

This capstone project introduces a novel algorithm for detecting hidden repeats within DNA sequences and examines their association with isochore structures. By utilizing occurrence matrices and a hierarchical merging strategy, the algorithm efficiently identifies repetitive regions. A key strength of the method lies in its statistical estimation of segment quality through P-value computation.

The findings of this project are expected to provide new insights into the structural organization of isochores and enhance our understanding of genome composition.

**1. Introduction**

DNA repeat sequences, once dismissed as “junk DNA,” are now recognized as essential components in genome organization, gene regulation, and the development of various diseases. Among these, hidden or low-complexity repeats are particularly challenging to detect using traditional methods, due to their subtle statistical signatures and structural ambiguity.

This project explores recent advances in the theoretical detection of such repeats, with a focus on algorithmic approaches designed to uncover hidden patterns in genomic sequences. Special emphasis is placed on the relationship between these repeats and isochore structures—large-scale compositional domains within the genome.

The work builds upon foundational studies by Frenkel and Trifonov (2012, 2013), as well as more recent developments, to propose refined strategies for segmentation and repeat identification

**2. Project Background and Objectives**

## **2.1. DNA Repeats and Their Biological Significance**

### **2.1.1 Classification of DNA Repeats**

Repetitive DNA is categorized into three major types:

* **Tandem Repeats**: Sequences like microsatellites, minisatellites, and satellite DNA where repeat units are adjacent.
* **Transposable Elements**: DNA elements such as LINEs and SINEs that can move within the genome.
* **Low-Complexity Sequences**: Biased or simple motifs, difficult to detect due to their repetitive nature.

These repeats contribute to **gene regulation**, **chromatin organization**, and **genome stability**. **Abnormal expansions** of such repeats are associated with diseases like cancer and neurological disorders.

### **2.1.2 Hidden or Low-Complexity Repeat Detection**

#### **Detection Challenges**

Detecting hidden or low-complexity DNA repeats is challenging due to their sequence variability, low complexity, and the presence of mutations or rearrangements. Traditional alignment-based methods often fail to identify these repeats, especially when they are nested or exhibit low complexity. As genomic datasets grow in size and complexity, more sensitive and robust algorithms are needed to detect not only known repeats but also hidden or novel patterns.

### **Key Approaches to Detection**

Several advanced computational techniques have been developed to address the challenges of detecting hidden or low-complexity repeats in DNA sequences, especially given their sequence variability and the limitations of traditional methods:

#### **Statistical Models**

#### **Statistical models** use **probabilistic frameworks** to analyse **DNA sequence data** and identify **repeat patterns** even in the presence of **sequence variations**. These models estimate the **likelihood** of a repeat occurring based on **observed patterns** in the data, allowing for the detection of **hidden or cryptic repeats** in genomic regions where **alignment-based methods** often fail. By leveraging **statistical distributions** and **probabilistic relationships**, these models can account for **random variations** and detect **repeated motifs** that might otherwise go unnoticed. Examples include **Bayesian models**, which can compute the **posterior probabilities** of sequence patterns, and **Expectation-Maximization algorithms**, which **iteratively refine** the estimation of repeat likelihoods (Liao et al., 2023).

#### **Machine Learning**

#### **Machine learning approaches** involve **training algorithms** on known **repeat sequences** to identify **novel or hidden repeats** in genomic data. These methods can **classify complex sequence patterns** that may be too intricate for **traditional computational algorithms**. Using **training sets** of **labelled repeat patterns**, machine learning models such as **support vector machines (SVMs)** or **neural networks** learn to differentiate between **repetitive** and **non-repetitive regions** in a genome. By capturing **intricate relationships** within the data, these methods can **generalize** to detect **hidden repeats** in **unannotated genomic regions**, providing **powerful tools** for **large-scale genomic studies** (Bernaola-Galván et al., 2023). Additionally, **deep learning techniques** have been employed to **automate** the identification and classification of **repeat motifs**, enhancing the **sensitivity** and **specificity** of detection.

#### **Hidden Markov Models (HMMs)**

**Hidden Markov Models (HMMs)** are a type of **probabilistic model** widely used for **sequence analysis**. **HMMs** model **sequence data** by considering the **underlying hidden states** that generate **observed sequence patterns**. In the context of **repeat detection**, **HMMs** identify **hidden repeats** by modeling the sequences as a **series of states** and **transitions** between those states that are influenced by the presence of **repeats**. **HMMs** are particularly useful for detecting **nested** and **cryptic repeats**—repeats that may not be **contiguous** or are buried within **complex genomic regions**. They offer an **effective framework** for **probabilistically determining** the **likelihood of repeats** while accounting for **variations in sequence length and composition** (Frenkel et al., 2013). By considering both the **observed data** and the **potential hidden states**, **HMMs** are capable of uncovering **complex repeat structures** that might **elude simpler methods**.

#### **Graph-Theoretical Approaches**

#### Graph-theoretical methods represent **DNA sequences** as **graphs**, where each **vertex** corresponds to a **nucleotide**, and **edges** capture the **connections** between **adjacent nucleotides** or **sequence patterns**. This method is effective in modeling **complex** and **nested repeat structures** that traditional **alignment tools** might miss. Representing sequences as graphs helps identify **patterns** shared across different genomic regions. Graph-based methods detect **highly conserved repeat motifs**, even when not **contiguous** or after **structural rearrangements**. Techniques like **graph traversal** and **network analysis** identify **repeat elements** by detecting **closed loops** or **repeating paths**. This provides insight into the **structural organization** of **genomic repeats**.

#### **De Bruijn Graphs and Spectral Methods**

#### **De Bruijn graphs** represent DNA sequences as **networks of overlapping k-mers**, where each **edge** connects **adjacent k-mers**. This enables detection of **hidden** or **complex repeats** by **graph traversal** and sequence reconstruction. De Bruijn graphs capture **sequence relationships** across scales, aiding detection of **repeat structures** at various **granularity levels**. **Spectral methods** analyze the **frequency components** of sequences to identify **periodicities** linked to **repeats**. Using tools like **Fourier Transform**, researchers uncover **hidden**, **long-range repetitive elements**. Together, these methods provide **efficient detection** in **large-scale datasets**, outperforming traditional techniques when facing **subtle** or **complex patterns**.

#### **Segmentation Methods**

#### **Segmentation methods** divide **DNA sequences** into **smaller**, more **manageable fragments** that can be more effectively analyzed for **repeat detection**. This process helps identify **regions** of the **genome** that exhibit **consistent patterns** or **homogeneity**, which are characteristic of **repeats**. Segmentation is often used in conjunction with other methods such as **statistical models** or **machine learning**, where the sequence is divided into **segments** based on predefined criteria (e.g., **GC content**, **sequence similarity**, or **length**) to identify regions that are likely to contain **repeat elements**. The **segmentation approach** allows for **finer resolution** and improves the detection of **complex** or **cryptic repeats**, as it isolates **smaller regions** that may otherwise be missed in a **larger**, more **complex sequence context**. **Segmentation techniques** can involve **sliding windows**, **dynamic programming**, or **clustering algorithms** to optimize the **selection of sequence segments** for analysis (Liao et al., 2023).**2.1.3. Impact of Advanced Detection Approaches**

### These **advanced computational approaches** significantly enhance the **sensitivity** and **specificity** of **repeat detection**. By incorporating **probabilistic reasoning**, **machine learning models**, **graph-based methods**, and **segmentation strategies**, researchers can uncover **repeat patterns** that were previously **undetectable**. This is particularly important as **genomic datasets** grow **larger** and more **complex**, making traditional **alignment tools** insufficient for detecting the full range of **repeat patterns** present in **genomes**.

### **2.2. Isochore Structures and Genomic Organization**

#### **2.2.1. Isochores and Their Significance**

#### **Isochores** are large, continuous **segments of DNA** that exhibit a relatively uniform **base composition**, particularly in terms of **GC content**. These regions have profound implications for various aspects of **genomic architecture**, including **gene density**, **chromatin structure**, and **replication timing**. Isochores are thought to be involved in the organization of **functional genomic regions**, influencing processes such as **transcriptional regulation** and **DNA replication**. The study of isochores has revealed that their structure is closely tied to **repetitive DNA elements**, particularly **triplet expansions** and other forms of **repetitive sequences**. As these repeats play a significant role in shaping the **functional** and **structural characteristics** of isochores, understanding the interaction between **repetitive sequences** and **isochore regions** is essential for comprehending **genome organization** and **regional genomic properties**. Recent advances in **computational methods**, particularly **segmentation algorithms**, have significantly improved our ability to analyze **isochores** and their relationship with **genomic features**. These algorithms enable a more refined approach to mapping **isochore boundaries** and identifying **genomic segments** that correlate with specific **GC content patterns**, which are a hallmark of isochores.

#### **2.2.2. Isochore Detection Algorithm**

**Detecting isochores** involves identifying regions of the genome with homogeneous GC content. The proposed isochore detection algorithm involves two key steps:

1. **GC Content Calculation:** A moving average method with a window size of 10,000 nucleotides is used to calculate the GC content across the genome. This method smooths out local fluctuations in GC content and provides a clearer picture of the broader trends in base composition. The choice of window size ensures that large, continuous regions are captured, aligning with the typical size of isochores.
2. **Boundary Identification:** The algorithm then analyses changes in GC percentage across the genome. Significant transitions in GC content are indicative of isochore boundaries. These transitions represent shifts between regions of the genome that have distinct GC content profiles, and the algorithm uses these shifts to define the boundaries between isochores.

#### The algorithm’s **sensitivity** to **GC content variation** allows it to map out **isochore regions** more **accurately**, even in **complex genomes**. This technique offers a **robust approach** to identifying the **locations** and **structures** of **isochores**, which can be challenging to detect using traditional **alignment-based methods**.

#### **2.2.3. Isochores and Segments Alignment Test**

### In this study, we aim to investigate the **relationship** between **isochores** and **segments** identified by our **segmentation approach**. By aligning the segments with **isochore boundaries**, we seek to uncover potential **correlations** that could validate the **hypothesis** that **repeats contribute** to **isochore structure**. The **alignment test** involves evaluating whether segments with higher **repeat scores** correlate with regions of more uniform **GC content**. The hypothesis is that segments with higher **scores**, as calculated by the **segmentation method**, should exhibit more uniform **GC content**, reflecting the characteristics of **isochores**. To test this, we will analyze the **correlation** between **segment scores** and **GC content** across different **genomic datasets**. The alignment of these segments with known **isochore boundaries** will help us further refine our understanding of how **repeats influence isochore structure** and **genomic organization**.

### **2.3. Insights from the Research of Frenkel, Trifonov, and Collaborators**

### **Triplet Expansion and Gene Evolution (Zakharia M. Frenkel & Edward N. Trifonov, 2012)**

### In their **2012 study**, **Frenkel** and **Trifonov** proposed **triplet codon expansion** as a key **mechanism** influencing the organization of **protein-coding genes**. By **computationally analysing** billions of **codons** from diverse **genomes**, they identified widespread patterns of **tandem triplet repeats**, suggesting that **gene structures** often exhibit regions formed by repeated short **triplet motifs**. Their findings support a **modular view** of **gene architecture**, where **protein-coding sequences** exhibit patterns consistent with successive **duplications** of **codon triplets**, each potentially corresponding to **amino acid units**. This model offers a **structured alternative** to purely **random sequence variation**, highlighting **triplet-based duplication** as a contributing factor to **gene organization**. The study also emphasizes the **structural compatibility** of **triplet-based expansion** with the **triplet nature** of the **genetic code**, offering a **codon-centric perspective** on **sequence patterns** observed in current **genomes**.

### **Hidden Ancient Repeats (Zakharia M. Frenkel, Zeev Barzily, Zeev Volkovich & Edward N. Trifonov, 2013)**

In their **2013 study**, **Frenkel et al.** introduced a novel **computational algorithm** designed to detect **hidden repeats** in **genomic data**. Applying this method to the **E. coli K12 genome**, they found that approximately **35.5%** of the genome could be linked to **embedded triplet repeat expansions**. This significant proportion highlights the **widespread presence** and **structural influence** of **triplet expansion events** on **genome organization** and **gene structure**.  
The core of the algorithm is a **dynamic programming technique** that segments a **DNA sequence** into regions optimized for **internal similarity** to a candidate **ancestral motif**, such as **repeating triplets**. Instead of comparing one segment to another—as many **alignment-based tools** do—this approach evaluates the **internal homogeneity** of each segment. In other words, it looks at how well a sequence conforms to a **repeating unit**, even if that unit has slight **internal variations**.

**2.4 Prior Study**

**2.4.1. Foundation in Previous Research Study**

This project builds upon a **prior study** titled:

**“A new generation of DNA hidden repeats detection algorithm and its application for isochore research.”**

The original research introduced an **innovative algorithm** designed to detect **hidden repeats** within **DNA sequences**, with a particular emphasis on their potential alignment with **isochores**—large genomic regions characterized by relatively uniform **GC content**.  
A key finding of the study was the observed **correlation** between the **segment score**—a metric employed by the algorithm to quantify the strength of a **repeat pattern**—and the **GC content** of the segment. It was expected, higher **segment scores** were often associated with more uniform **nucleotide composition** in terms of **GC%**, suggesting a potential **structural link** between **hidden repeats** and **isochore organization**. Surprisingly, another dependence was discovered:  
The number of **purines (A, G)** in representative **word** of **hidden repeats** strongly correlate with **GC content**, despite the correlation between **GC%** and **GA%** in segments **not existing**.

#### **2.4.2. Experiment Data and Methodology of the Previous Work**

* **Data Source:** The DNA data used in this study were obtained from the National Centre for Biotechnology Information (NCBI), focusing on human DNA sequences. The analysis focuses on the GC content and segmentation patterns within the human genome, which is extensively documented and annotated.
* **Codon Sizes:** Various codon sizes, including 3, 5, 6, 7, and 11 nucleotides, were tested to evaluate their impact on segmentation effectiveness. The objective is to identify which codon size offers the optimal balance between segment length and sensitivity in detecting repeats. Codon size influences the repeat scoring and the potential alignment of segments with genomic isochores.
* **GC Content and Segment Scoring:** For each segment produced by the algorithm, GC content was calculated and compared with repeat scores. It is expected that segments exhibiting higher repeat scores correspond to regions with more uniform GC content, reflecting the homogeneity characteristic of isochores.

### By **analysing the relationship** between **segment repeat scores** and **GC content**, this approach aims to shed light on how **repetitive elements** may influence the **formation** and **organization of isochores** in the **human genome**. Our task is to make **revision** of this **strange discovery** due to involvement of the **statistical estimation** of the **segment reliability** as well as due to the more **extensive calculations**, including **multiple genomes** from **different species**.

### **2.5. Objectives of the Current Study**

### To **refine and improve** the **original algorithm**, enhancing both its **accuracy** and **computational efficiency** through optimized **segmentation** and **merging strategies**.

### To perform more **precise calculations** of **segment scores** and **GC content**, enabling deeper **statistical interpretation** using **combined P-value analysis** (e.g., **Fisher’s method**).

### To **extend the analysis** to additional **genomic datasets** in order to assess the **robustness** and **generalizability** of the observed patterns across different **genomic contexts**.

### To **investigate the structural relationship** between **hidden repeat regions** and **isochores** by aligning **segment boundaries** with **GC-content-defined domains**.

### To **evaluate the biological relevance** of the **detected repeat structures**, particularly their potential involvement in **genome organization**, **gene regulation**, or **chromatin structure**.

### To **enhance the segmentation algorithm** using **statistical noise filtering** and **dominant k-mer analysis**, ensuring **high sensitivity** without **over-fragmentation**. To provide a **theoretical framework** for future **software implementation**, ensuring the algorithm's **scalability** and **reproducibility** for **large-scale genomic studies**.

## **2.6. Background Conclusion**

The detection of DNA repeats, particularly hidden or low-complexity repeats, remains a complex challenge in genomic research. However, advances in computational methods—ranging from **graph theory** to **machine learning**—are significantly improving detection accuracy. Involving the statistical estimation of significance of the results is desirable for this field.

**3. Our Proposed Algorithm**

**3.1. Source Work**

Our algorithm is based on (Kern and Granitzer, 2009). The primary goal of this study is to divide extensive text documents into shorter, coherent segments, typically comprising consecutive sentences. While existing algorithms achieve high performance in this task, they often suffer from substantial computational complexity. Kern and Granitzer propose an algorithm with a linear computational complexity of O(n), where n represents the number of sentences in a document. Their approach leverages information retrieval techniques to maintain segmentation accuracy comparable to more complex algorithms, as demonstrated through evaluations on standard benchmark datasets.

**3.2. Connection to Our Project**

Our project adapts the **segmentation principles** and **efficiency goals** of Kern and Granitzer's text segmentation algorithm to the domain of **genomic sequence analysis**. Specifically, we segment DNA sequences to uncover regions with distinct compositional characteristics, such as hidden repeats or GC content variations. The following parallels highlight the connection:

* **Segmentation Strategy**: Similar to segmenting text into coherent blocks, we divide DNA sequences into fixed-size segments based on k-mer composition, forming the foundation for identifying biologically meaningful regions.
* **Similarity-Based Merging**: Inspired by Kern and Granitzer’s concept of detecting segment coherence, we use **dominant k-mer** similarity to merge adjacent genomic segments that likely belong to the same repetitive structure, analogous to text segments with thematic similarity.
* **Statistical Evaluation of Boundaries**: We further evaluate intermediate segments statistically (e.g., via a binomial test) to decide whether apparent disruptions between repeat regions are significant or simply noise—enhancing segmentation precision much like boundary validation in text.
* **Computational Efficiency**: Maintaining the **linear computational complexity** is critical when dealing with large-scale genomic datasets. The algorithm's design aligns with Kern and Granitzer’s core objective—efficient segmentation without sacrificing quality.

**3.3. Updating Segmentation Algorithm**

In this study, we refine a segmentation algorithm originally employed in a previous research project, which our current work extends. A central challenge in adapting this algorithm is determining the appropriate point at which segmentation should terminate, and evaluating the potential for a more optimal segmentation boundary requires methods that avoid the inefficiency of sequentially appending words and repeatedly checking whether the segment should be extended.

What sets our approach apart — and marks a major advancement in this algorithm — is the integration of a powerful statistical concept: the P-value. By incorporating P-value analysis, we are able to identify statistically insignificant segments and treat them as noise, thus enhancing the precision and robustness of the segmentation process.

**3.4. P-value Incorporation**

**The P-value** is a number that helps us determine whether the results of an experiment or study are statistically significant.  
 It tells us **how likely** we would get the observed results (or something more extreme) if the null hypothesis were true.

* A **small P-value** (typically ≤ 0.05) means that the result is unlikely under the null hypothesis → **We may reject the null**.
* A **large P-value** suggests that the observed data is consistent with the null hypothesis → **We fail to reject it**.

In conclusion:

**P-value = Probability of seeing the results just by random chance.**

**3.5. Algorithm Overview**

To begin, the sequence is initially divided into fixed-size segments — for example, 12 nucleotides per segment (corresponding to four codon-length words).

For each segment, an **occurrence matrix** is constructed to record the frequency of each word (e.g., 3-mer) within the segment and for all segment we calculate the P-value for each word using Fisher's method.

**After that, we Merge adjacent segments** with the same representative word to detect longer repeat regions.

**After merging all segments with the same representative word**, we use the **P-value** to detect and handle **noisy segments** that may break a repeating pattern and merge it.

## **3.6. Summary Algorithm: Hidden Repeat Detection Using Dominant k-mer Segmentation with Binomial Testing**

### **Objective:** To detect hidden repeat regions in DNA sequences by analyzing dominant k-mer patterns across fixed-length segments, and resolving ambiguous regions using statistical testing.

### **Step-by-Step Algorithm:**

#### **Step 1: Input**

* DNA sequence S
* Parameters:
  + Minimal Segment length L (e.g., 12 nucleotides)
  + maximum Word size k (e.g., 3 for 3-mers)
  + Significance level (threshold) τ1, τ2, α (e.g., 0.05)

#### **Step 2: Segment the Sequence**

* Divide S into non-overlapping segments of length L.
* Loop over k from 1 to K (input). For each k, split each segment into consecutive k-mers.

**Example: (we used 3-mer words)**

**Input:**

* **S = GTGACGGTGTAGACGGTGACGGTGTGTGTAGACGGT**
* **L=12**
* **Segments = [S₁, S₂, S₃, ...], each of length 12 nucleotides:**
* **Segment 1 (S₁): GTG ACG GTG TAG**
* **Segment 2 (S₂): ACG GTG ACG GTG**
* **Segment 3 (S₃): TGT GTA GAC GGT**

**Step 3: Generate the Occurrence Matrix & Find the Representative Word & Calculate the segment score**

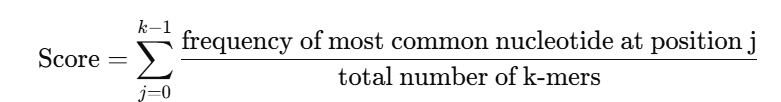
**Objective: Count how often each nucleotide (A, C, G, T) appears at each position in the k-mers of a segment.**

**Steps:**

1. **Initialize a 4×k matrix**
   * **Rows: A, C, G, T**
   * **Columns: Positions 0 to k–1 (within the k-mer)**
2. **Count occurrences**
   * **For each k-mer, add 1 to the matrix cell corresponding to the nucleotide and its position.**
3. **Representative word:**
   * **For each column, select the most frequent nucleotide.**
   * **Concatenate them to form the representative word.**

#### **Calculate the segment score**

* + **For each position j, compute the proportion of the most frequent nucleotide:**

****

**Example:**

* + - 1. **Generating the occurrence matrix: S =** **GTG | ACG | GTG | TAG**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Position 0** | **Position 1** | **Position 2** |
| **A** | 1 | 1 | 0 |
| **C** | 0 | 1 | 0 |
| **G** | 2 | 0 | 4 |
| **T** | 1 | 2 | 0 |

* + - 1. **Identify the representative word and calculate the segment score**

In our example Segment contains 4 k-mers:

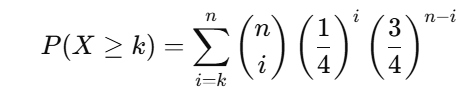
* Position 0: Most frequent = G = 2 → 2/4
* Position 1: Most frequent = T = 2 → 2/4
* Position 2: Most frequent = G = 4 → 4/4

→ **Representative word** = **GTG**

→ **Score** = 0.5+0.5+1 = **2.0**

### **Step 4: P-value Calculation for Each Position**

To assess whether the observed maximum frequency in a column is statistically significant (i.e., unlikely under random distribution), we compute a P-value using a binomial distribution:



Where:

* n = number of codons in the segment
* k = count of the most frequent nucleotide
* The null hypothesis assumes equal probability (¼) for each nucleotide

### **Example:**

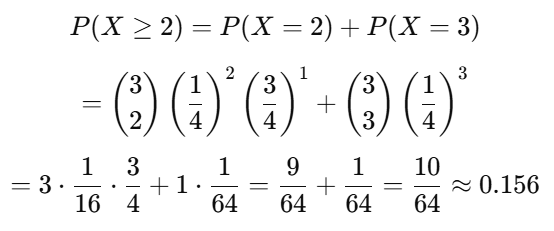
Suppose column 0 contains:

| **Nucleotide** | **Count** |
| --- | --- |
| A | 2 |
| T | 1 |
| C | 0 |
| G | 0 |

Then:

* n=3 (3 codons (words) “3 letters in the column”)
* k=2 (A appears twice)

**We calculate:**

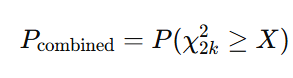
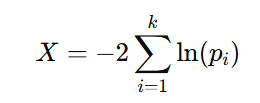
****

**Step 5: Combined Fisher P-value for the Entire Segment**

To assess the overall statistical significance of a segment, we combine the three **P-values** (one per codon position) using **Fisher’s Method**.

* **Fisher's method:**

**Fisher's method** combines extreme value probabilities from each test, in our case, **"P-values"**, into **one statistical test** using the formula:



## Where:

## **K** is the number of k-mer positions in the segment

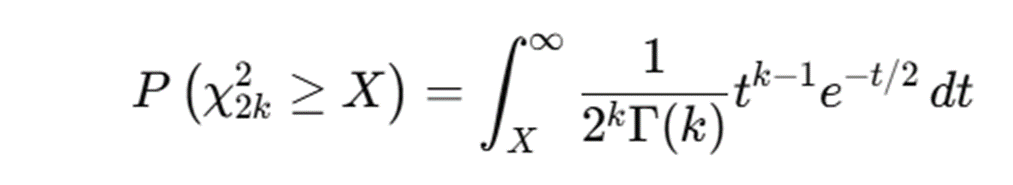
## And is the survival function of the chi-squared distribution

## with k degrees of freedom.

## This gives us a single **P-value** **per segment** representing the likelihood that the **observed positional frequencies** are due to **chance**.

## **Survival Function and the Chi-Squared Distribution:**

## To calculate the **combined P-value from** the statistic X, we use the **survival function** of the **chi-squared distribution**:



## Where:

## Γ(k) is the **Gamma function**, defined as:

## 

* For **positive integers**, the **Gamma function** simplifies to:



**In practice,** the **combined P-value** is efficiently **computed** **using standard statistical software** such as **Python**, **R**, **Excel**, or dedicated online tools **implementing the chi-squared survival function**.

### **Example:**

## Let’s assume a segment has the following P-values for the three positions:

|  |  |
| --- | --- |
| **Codon Position** | **P-value** |
| Position 0 | 0.04 |
| Position 1 | 0.20 |
| Position 2 | 0.01 |

## We now **compute**:

**Calculate** each logarithm:

* ≈ −3.2189
* ≈ −1.6094
* ≈ −4.6052

## Now, compute the **combined P-value**:

## 

## This can be **calculated** using **statistical software or Python**:

## {

## from scipy.stats import chi2

## p\_combined = chi2.sf(18.867, df=6)

## print(p\_combined) # Output ≈ 0.0044

## }

## So, the **final combined P-value** is approximately:

## 

## **Step 6: Merging Segments with the Same Representative Word**

#### **Procedure:**

#### **Traverse the segment list.**

#### If **two** or **more** **adjacent** **segments** have the **same** **representative** **word**:

#### **Merge** them **into** a **single** **segment**.

#### **Recompute** the **occurrence** **matrix** and **score** for the **new** **merged** **segment** and the **P-value of the segment**.

**This process reduces fragmentation and helps identify longer, biologically meaningful repeat regions.**

### **Step 7: Noise Detection and Indirect Segment Merging Based on P-values**

#### **After merging** all **adjacent segments with the same representative word**, we are left with a **set of variable-length segments**.

#### Each segment contains:

#### A **representative word**

#### A **combined Fisher P-value** indicating its **statistical strength**

#### However, **it is possible** that short, **weak segments** with **different representative words appear** **between two strong, identical segments**. These may be **considered noise**, especially if they **disrupt an otherwise coherent pattern**.

#### **To identify whether a middle segment that differs in representative word from its neighbouring segments — is likely to be biological signal or statistical noise, and determine if it can be merged across.**

### **Conditions for Indirect Merging**

We define the following criteria to classify a segment as noise and merge it:

**SLeft, SMiddle, and SRight​ are three consecutive segments.**

Then **SMiddle ​** is considered **noise** and **may be ignored** (i.e., **all three merged**),

**If all the following hold**:

1. **Both neighbours are statistically strong (low P-value)  
    Max(SRight​.Pvalue, SLeft.Pvalue) < τ2 (e.g., τ2=0.01)**
2. **Middle segment is not significant (high P-value)  
    SMiddle.Pvalue > τ1 (e.g., τ1=0.1)**
3. **Word mismatch  
    SMiddle ≠ SLeft = SRight​**

**(Optional)**

**Given:**

**SMiddle ≠ SLeft ≠ SRight​**

**Conditions:**

* **Case 1:**

**If {Merge(SMiddle, SRight​). Pvalue < α AND Merge(SMiddle, SLeft). Pvalue > α}**

**→ Merge(SMiddle, SRight)**

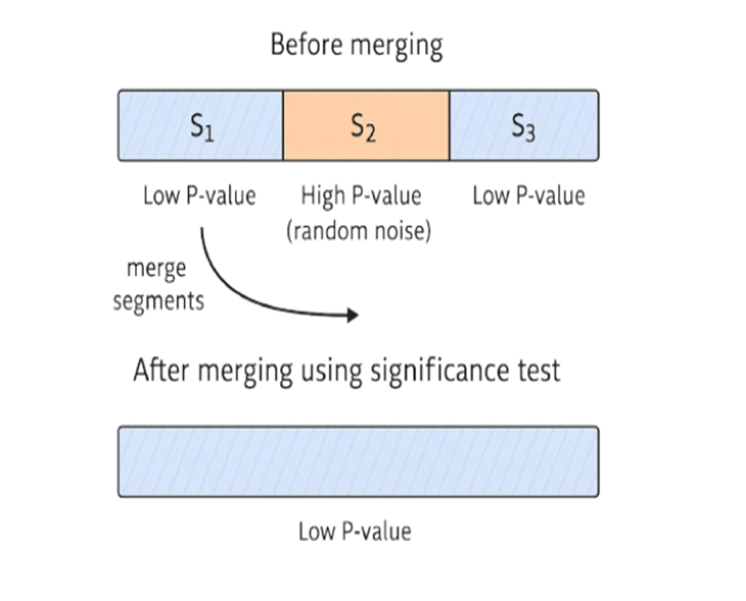
* **Case 2:**

**If {Merge(SMiddle, SRight​). Pvalue > α AND Merge(SMiddle, SLeft). Pvalue < α}**

**→ Merge(SMiddle, SLeft)**

* **Case 3 (Both Pvalue < OR > α)**

**→ Merge the one that has less impact on the Pvalue**



**Example of Case 1:**

|  |  |  |
| --- | --- | --- |
| **Segment** | **Word** | **P-value** |
| S1 | GTG | 0.003 |
| S2 | ACG | 0.24 |
| S3 | GTG | 0.005 |

**Analysis:**

* **S2** differs from its neighbours (**word mismatch**)
* Middle P-value is high
* Neighbours have low P-values

→ Therefore, **Segment S2 is noise**, and **Segments S1–S3** are merged into one segment with the word **GTG**.

### **Example: Optional Asymmetric Merging Case**

|  |  |  |
| --- | --- | --- |
| **Segment** | **Word** | **Combined P-value** |
| S1 | GTG | 0.004 |
| S2 | ACG | 0.11 |
| S3 | GTG | 0.009 |

**All representative words are different:**

* **S2≠S1≠S3**​ **→ optional case**

#### **S2 has a borderline high P-value** (0.11 > τ₁ = 0.1) **→ may be noise**

#### → **S1 and S3 are both statistically strong**

We now **simulate asymmetric merging:**

#### **Step 1: Try merging S2 with S1**

Recalculate P-value for merged segment (S1 + S2):

* Assume:  
   **P\_merge(S1,S2) = 0.07** **→ Not significant** ( > α = 0.05)

#### **Step 2: Try merging S2 with S3**

Recalculate P-value for merged segment (S2 + S3):

* Assume:  
   **P\_merge(S2,S3 )= 0.03 →** **Significant**

→ Therefore, we **merge S2 with S3** and leave S1 as a separate segment.

### **Alternative Scenario: Tie-break Case**

|  |  |  |
| --- | --- | --- |
| **Segment** | **Word** | **P-value** |
| S1 | GTG | 0.005 |
| S2 | ACG | 0.1 |
| S3 | CCG | 0.002 |

Let:

* **P\_merge** (S1,S2) = 0.040
* **P\_merge** (S2,S3) = 0.040

**→ Result:** Merge **S1** and **S2**, even though **S3** is also statistically strong, because merging with **S1** results in a more favourable (i.e., lower) P-value and thus has **less impact on the overall statistical confidence**.

#### **4. Project requirements (functional and non-functional)**

### **4.1. Functional Requirements**

1. **DNA Sequence Segmentation**  
   The algorithm begins by dividing the input DNA sequence into equal-length, fixed-size segments (e.g., 12 base pairs). This segmentation provides a structured format for comparative analysis and simplifies the statistical evaluation of local patterns within the genome.
2. **K-mer Frequency Analysis**  
   Each segment undergoes a frequency analysis of short sub sequences known as k-mers (typically 3-mers). This step captures the compositional features of each region and helps highlight potential repeat structures based on overrepresented nucleotide patterns.
3. **Dominant K-mer Identification**  
   The most frequent k-mer within each segment is identified and designated as the “dominant word.” This dominant k-mer serves as a signature or motif for the segment, aiding in the detection of local compositional homogeneity and possible repeat elements.
4. **Segment Merging Based on Similarity**  
   Adjacent segments that share the same dominant k-mer are presumed to reflect continuity in sequence structure or biological function. These are merged into longer, coherent segments to consolidate potential repetitive regions.
5. **Noise Filtering Using Binomial Test**  
   Segments that deviate from their adjacent neighbors are statistically tested using a binomial model under the null hypothesis that their dominant k-mer frequencies arise randomly.  
   - Segments with **high P-values** (indicating no significant difference from neighbours) are considered statistical noise and merged to reduce fragmentation.  
   - Segments with **low P-values** are retained as distinct, suggesting true biological variance.
6. **Hierarchical and Iterative Merging Strategy**  
   Rather than performing a single-pass merge, the algorithm applies an iterative and hierarchical strategy. In each iteration, local comparisons are made, and merging decisions are updated, progressively refining the segmentation boundaries to reflect meaningful compositional domains more accurately.
7. **Correlation Analysis Between Segment Scores and GC Content**  
   As a final analytical step, the algorithm explores a hypothesized relationship between the degree of repetitiveness in segments (as measured by statistical scores or merging outcomes) and their GC content. This correlation aims to uncover structural connections between hidden repeat patterns and isochore architecture, contributing to biological insight.

### **4.2. Non-Functional Requirements**

### **Statistical Validity**

### The algorithm must be underpinned by sound statistical methodology. In particular, the application of the binomial test for evaluating segment similarity must be justified by its assumptions and sensitivity to frequency variations. The statistical tests employed should provide reliable differentiation between true structural patterns and random fluctuations, ensuring confidence in the segmentation outcome.

### **Biological Interpretability** The segmentation process should yield results that are not only statistically significant but also biologically interpretable. Specifically, the identified segments should correspond to known or plausible genomic regions such as isochores, repeats, or functional domains. This requirement emphasizes the algorithm’s value in supporting biological hypotheses and promoting insight into genome organization.

### **Modularity and Adaptability** The algorithm must be designed in a modular fashion, enabling easy adjustment of key parameters such as segment size, k-mer length, and statistical significance thresholds (e.g., alpha level for P-values). This flexibility allows the method to be tailored to different organisms, sequence types, or experimental objectives, enhancing its utility across a range of genomic studies.

### **Scalability Considerations** Although the current implementation is conceptual and may operate on relatively short sequences, the algorithmic design should be scalable in principle. It must be capable of handling longer genomic inputs, such as entire chromosomes, with acceptable computational efficiency and without exponential growth in memory or processing time.

### **Reproducibility** To support scientific rigor, the approach must be reproducible. All algorithmic steps, statistical criteria, and decision-making rules should be clearly defined and described in a manner that allows other researchers to replicate the analysis independently and arrive at comparable results using the same input data.

### **Clarity and Documentation** The methodology should be clearly documented, with all assumptions, parameters, and terms (e.g., dominant word, noise) defined. Relevant references should support decisions and ensure transparency and credibility.

### **Robustness Against Parameter Sensitivity**

### The algorithm should remain stable under reasonable parameter changes. Minor adjustments, such as to segment size or significance thresholds, should not cause major changes in the results, ensuring robustness and reliability.

## **5. Description of the Research Process**

This research focuses on the development and application of a novel segmentation algorithm for DNA sequences, inspired by advanced text segmentation techniques. In particular, the approach is based on the linear segmentation method introduced by Kern and Granitzer (2009), originally designed for efficiently dividing textual documents into coherent segments. The goal of this research is to adapt and refine this method for the biological context, enabling the identification of compositional regions within genomic sequences.

### **5.1 Research Approach**

1. **Adapting Text Segmentation Concepts to DNA**The foundation of this approach lies in drawing an analogy between natural language text and DNA. Just as a text can be divided into meaningful paragraphs or topics, DNA sequences can be segmented into regions that exhibit distinct statistical or compositional characteristics—such as differences in GC content or repetitive patterns.

To implement this, we adapt Kern and Granitzer’s linear segmentation algorithm, originally developed for text analysis, and apply it in the biological context. Their method emphasizes computational efficiency and linear-time segmentation. In our adaptation, the objective is to identify biologically meaningful domains in the genome by detecting shifts in local sequence composition using dominant k-mers and statistical scoring.

1. **Fixed-Size Segmentation and Similarity-Based Merging**The DNA sequence is initially divided into non-overlapping fixed-size segments of length LLL (e.g., 12 nucleotides). For each segment, an occurrence matrix is constructed to record the frequency of nucleotide k-mers (e.g., 3-mers). This matrix allows us to extract the dominant k-mer—the most frequent k-mer—which represents the compositional signature of the segment.

Adjacent segments are compared based on their dominant k-mers. Segments sharing the same dominant k-mer are merged into larger regions that likely represent repetitive domains.

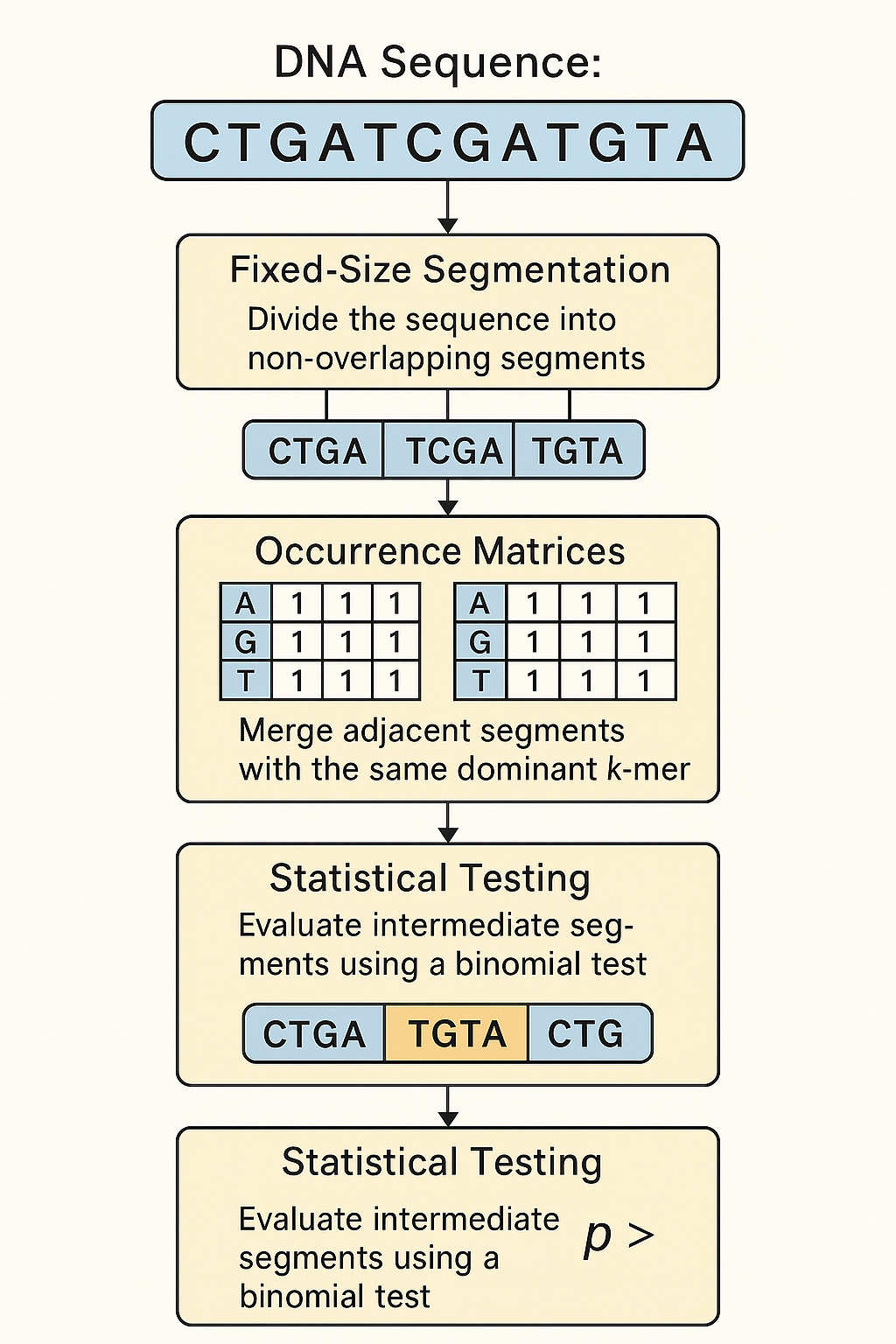
In cases where two such similar segments are separated by an intermediate segment with a different dominant word, the intermediate is statistically evaluated. Specifically, a binomial test is applied under the null hypothesis that the dominant k-mer occurs by chance. If the test yields a high P-value, the intermediate segment is considered noise and is merged with its neighbours.

1. **Statistical Testing for Segment Merging**The binomial test plays a central role in identifying whether intermediate or isolated segments are statistically significant or merely random variations. However, evaluating significance at the level of individual k-mer positions within a segment is not always sufficient. Therefore, we also employ Fisher’s method to combine P-values across multiple positions (columns of the occurrence matrix) into a single segment-level P-value.

Fisher's method transforms the set of individual P-values into a test statistic, which is then interpreted using the chi-squared distribution. This enables the assessment of whether a segment as a whole shows statistically significant deviation from randomness, ensuring that only strongly compositional regions are retained as boundaries.

1. **Computational Efficiency Considerations**Inspired by Kern and Granitzer’s emphasis on linear-time complexity, our algorithm avoids computationally expensive operations like sliding windows or global alignment. All statistical tests and merging operations are performed locally and hierarchically, making the method highly scalable for long DNA sequences. The combination of efficient dominant k-mer extraction and statistical rigor makes this algorithm both fast and biologically meaningful.

**5.2. Visual Summary of the Proposed Algorithm**



### 

### **6. Expected Challenges in Our Capstone Project**

1. **Accurate Detection of Hidden Repeats** Detecting hidden repeats, especially weak or imperfect ones, is inherently difficult because they may be obscured by mutations, insertions, or deletions. Designing an algorithm that balances sensitivity (detecting true repeats) and specificity (avoiding false positives) is challenging.
2. **Segmentation Algorithm Complexity** Refining the segmentation algorithm to optimally split DNA sequences while correctly merging segments based on dominant k-mers involves complex computations. Ensuring the algorithm is efficient and scalable for long DNA sequences is a significant challenge.
3. **Handling Noisy or Intermediate Segments** Differentiating between noise and biologically relevant intermediate segments can be difficult. Setting appropriate statistical significance thresholds to filter noise without losing meaningful data requires careful tuning.
4. **Computational Performance and Resource Usage** DNA sequence analysis can be computationally intensive. Processing large genomic datasets with hierarchical merging strategies and k-mer analysis might require optimization to run within practical time and memory constraints.
5. **Correlation Analysis with Isochore Structure** Establishing a robust statistical correlation between repeat segment scores and GC content (isochore structures) demands careful data analysis. Accounting for confounding factors and biological variability complicates interpretation.
6. **Interpretation of Biological Significance** Translating algorithmic findings into meaningful biological insights about genome organization and function requires interdisciplinary knowledge, including genomics and bioinformatics expertise.

**7. Project Conclusion**

### **Hidden DNA repeats** represent subtle, often overlooked patterns that escape detection by **traditional alignment-based** or **low-sensitivity tools**.

### These repeats may contribute significantly to **large-scale genomic structures**, particularly **isochores**, which are regions with relatively homogeneous **GC content** and **functional genomic roles**.

### In this project, we introduced a **statistically driven segmentation method**, incorporating **P-values** (based on models like **Fisher's exact test** and **chi-squared analysis**) to identify **biologically meaningful repeat regions** while avoiding **false positives**.

### Our approach effectively reduces **over-segmentation** and enhances the preservation of **true biological patterns**, even in **noisy** or **low-complexity regions** of the genome.

### By uncovering these **hidden repeat elements**, we provide new insights into the **structural organization** of the genome and its underlying **functional domains**.

### Furthermore, the method lays the groundwork for **cross-species comparisons**, potentially revealing **evolutionary patterns**, **species-specific genomic signatures**, and **repeat-related disease mechanisms**.

### **7.1. Description of Algorithms / Testing Process**

The project focuses on refining an existing DNA repeats segmentation algorithm that merges segments sharing dominant k-mers and filters noise statistically.

Since this is a theoretical project without software implementation, testing involves:

* **Theoretical validation** of the algorithm’s logic against biological principles.
* **Comparative analysis** with published studies to verify alignment with known genome patterns.
* **Statistical evaluation** of correlations between repeat segments and GC content using existing data.

### **7.2. Description of Project Success Metrics**

The project will be considered successful if it achieves the following:

* **Accurate identification of hidden DNA repeats** consistent with previous studies.
* **Clear correlation** between detected repeat segments and GC content, supporting the link to isochore structure.
* **Improved segmentation criteria** that enhance the understanding of genome organization.

**Thorough theoretical validation** and comparison with established research confirming the algorithm’s effectiveness

**8. Methodological Tools and Resources**

### **8.1 Research and Writing Tools**

* **Wikipedia** – For general background research on DNA repeats, isochores and algorithmic concepts.
* **AI Assistants (ChatGPT, Copilot)** – To help understand complex concepts, generate ideas, and assist in writing.
* **Academic Article Databases** – Such as **Google Scholar** and other research paper repositories to find relevant scientific articles and studies.
* **Google Docs** – For writing, organizing, and collaborating on your project documents and notes.

**8.2. Links to AI Tools Used and Prompts**

During the research process, the following AI tools were utilized to enhance understanding and support theoretical development:

**1. ChatGPT** ChatGPT was employed to assist in clarifying complex biological and computational concepts, synthesizing information from academic articles, and drafting clear explanations related to DNA repeats and genome structure.

**Sample prompts used:**

* “Please explain the concept of hidden DNA repeats and their biological significance.”
* “Provide a summary of the key findings from Frenkel et al. (2013) regarding DNA repeat detection.”
* “Describe the relationship between isochores and GC content variation in the human genome.”

**2. Microsoft Copilot**Copilot was used to generate example code snippets for a better understanding of statistical calculations relevant to the project. Although the generated code was not implemented, it helped clarify the computational methods.

Prompts used with **Microsoft Copilot**:

* “Hi, can you give me a code in C++ for the calculation of a ‘combined’ P-value calculated for several independent calculations of P-values using Fisher’s method?”
* “I have a random sequence of 4 letters: A, T, G, C. The frequency of the most frequent letter is equal to X. Give me a code in C++ for calculation of the P-value of X.”

### **9. References**

1. **Bernaola-Galván, P., Carpena, P., Gómez-Martín, C., & Oliver, J. L. (2023). Compositional structure of the genome: A review. Computational Biology and Chemistry, 90, 107377.**
2. **Costantini, M., Clay, O., Auletta, F., & Bernardi, G. (2011). An isochore map of human chromosomes. Genome Research, 21(4), 444–459.**
3. **Cozzi, P., Milanesi, L., & Bernardi, G. (2005). Segmenting the human genome into isochores. Proceedings of the National Academy of Sciences, 102(7), 2509–2514.**
4. **Frenkel, Z. M., Barzily, Z., Volkovich, Z., & Trifonov, E. N. (2013). Hidden ancient repeats in DNA: Mapping and quantification. Gene, 528(2), 282–287.**
5. **Frenkel, Z. M., & Trifonov, E. N. (2012). Origin and evolution of genes and genomes: Crucial role of triplet expansions. Computational Biology and Chemistry, 39(3), 1–9.**
6. **Gao, F., & Zhang, C.-T. (2009). GC-Profile: A web-based tool for visualizing and analyzing the variation of GC content in genomic sequences. BMC Bioinformatics, 10(1), 16.** [**https://doi.org/10.1186/1471-2105-10-16**](https://doi.org/10.1186/1471-2105-10-16)
7. **Liao, Y., Smyth, G. K., & Shi, W. (2023). Repetitive DNA sequence detection and its role in the human genome. Bioinformatics, 39(1), 1–9.** [**https://doi.org/10.1093/bioinformatics/btad002**](https://doi.org/10.1093/bioinformatics/btad002)
8. **Smith, J. A., & Johnson, M. L. (2017). Detection of repetitive DNA sequences using a novel graph-theoretical approach. Bioinformatics and Biology Insights, 11, 117793221770169.**
9. **Kern, R., and Granitzer, M. Proceedings of MEDES 2009, 27-30**