

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” _____ 2022 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

на тему: Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях

Виконав : студент 4 курсу, групи ІВ-82
(шифр групи)

Дуплій Наталя Володимирівна

(прізвище, ім’я, по батькові)

(підпис)

Керівник асистент Кочура Ю.П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) професор, д.т.н., Сімоненко В. П.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2022 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 122 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
Сергій СТИРЕНКО

_____ (підпис)

“ ” _____ 2022 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Дуплій Наталі Володимирівни

1. Тема проєкту Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях

керівник проєкту асистент Кочура Ю.П.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 10 червня 2022 року №1033-с

2. Термін здачі студентом закінченого проєкту 11 червня 2022 р.

3. Вихідні дані до проєкту технічна документація, теоретичні дані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)

Розділ 1. Огляд існуючих рішень.

Розділ 2. Технології реалізації.

Розділ 3. Програмна реалізація.

Розділ 4. Дослідження та аналіз розробленої системи.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) структурна схема системи, функціональна схема (діаграма класів), алгоритм дій програмного забезпечення.

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В. П.		

7. Дата видачі завдання 2 травня 2022 року

Календарний план

№ п/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>10.12.2021-10.06.2022</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>02.05.2022-13.05.2022</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>16.05.2022-05.06.2022</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>20.05.2022-05.06.2022</i>	
5.	<i>Програмна реалізація системи</i>	<i>27.05.2022-05.06.2022</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>02.05.2022-05.06.2022</i>	
7.	<i>Захист програмного продукту</i>		
8.	<i>Передзахист</i>		
9.	<i>Захист</i>	<i>23.06.2022</i>	

Студент-дипломник _____ Наталя ДУПЛІЙ
(підпис)

Керівник проєкту _____ Юрій КОЧУРА
(підпис)

АНОТАЦІЯ

У даній роботі було розглянуто існуючі методи візуалізації нуклеотидних послідовностей. В результаті було розроблено веб-застосунок для візуального порівняння нуклеотидних послідовностей.

Ключові слова: візуалізація, нуклеотидні послідовності, візуальне порівняння.

ANNOTATION

In this project existing methods of nucleotide sequences visualization method were considered. As a result, web-application for visual comparison of nucleotide sequences was developed.

Key words: visualization, nucleotide sequences, visual comparison.

справки	Формат	Значення	Найменування	Кіл. листів	№ екземпля	Додаток
			Документація загальна			
			Знову розроблена			
	<i>A4</i>	<i>ІАЛЦ.467200.002 ТЗ</i>	Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях	3		
	<i>A4</i>	<i>ІАЛЦ.467200.003 ПЗ</i>	Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях	55		
	<i>A4</i>	<i>ІАЛЦ.467200.004 Д1</i>	Пояснювальна записка Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях Структурна схема	1		
	<i>A4</i>	<i>ІАЛЦ.4672008.005 Д2</i>	Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях Функціональна схема	1		
	<i>A4</i>	<i>ІАЛЦ.4672008.005 ДЗ</i>	Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях Алгоритм дій програмного забезпечення	1		

					<i>ІАЛЦ.467200.001 ОА</i>		
<i>Зм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп</i>	<i>Дата</i>			
<i>Розроб</i>		Дуплій Н.В.			<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перев</i>		Кочура Ю.П.				1	1
					<i>КПІ ім. Ігоря Сікорського ФІОТ ІВ-82</i>		

*Застосунок
візуального порівняння
нуклеотидних
послідовностей в
біомедичних
дослідженнях
Опис альбому*

ТЕХНІЧНЕ ЗАВДАННЯ
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «Застосунок візуального порівняння нуклеотидних послідовностей в
біомедичних дослідженнях»

Київ – 2022

ЗМІСТ

НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
ДЖЕРЕЛА РОЗРОБКИ.....	2
ТЕХНІЧНІ ВИМОГИ.....	3
Вимоги до розробленого продукту	3
Вимоги до програмного забезпечення	3
Вимоги до апаратної частини	3
ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.467200.002 ТЗ			
		№ докум.	Підпис	Дата				
Розробив	Герєга Б. Д.				Еволюційні алгоритми глобальної пошукової оптимізації Технічне завдання	Літ.	Аркуш	Аркушів
Перевірив	Волокита А. М.						1	3
Н. Контр.	Сімоненко В. П.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ПІ-73		
Затвердив								

1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання поширюється на розробку застосунку візуального порівняння нулеотидних послідовностей, а також на подальшу підтримку та вдосконалення розробленого застосунку.

Областю застосування цього застосунку є біомедичні дослідження, а саме дослідження ДНК і РНК.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки даної системи є завдання для виконання роботи кваліфікаційно-освітнього рівня «бакалавр інженерії програмного забезпечення», який був затверджений факультетом “Інформатики та обчислювальної техніки” кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

3 МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою та призначенням даної роботи є розробка застосунку візуального порівняння нуклеотидних послідовностей, що дозволить ефективно вирішувати дану задачу.

4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проекту є офіційні документації, публікації та статті в мережі Інтернет на дану тему, науково-технічна література.

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

Розроблена система має виконувати такі вимоги:

- Простий і інтуїтивно-зрозумілий інтерфейс системи.
- Надати можливість користувачам передавати вхідні дані для візуалізації та отримувати у відповідь результат.
- Надати можливість користувачам експорту результату.
- Надати вичерпну та зрозумілу документацію для розробленого продукту.

5.2. Вимоги до програмного забезпечення

- ОС Windows, Mac чи Linux.
- Visual Studio Code.

5.3. Вимоги до апаратної частини

- ЦП не менше ніж Intel® Celeron Processor N3350.
- RAM не менше ніж 4 ГБ.

6 ЕТАПИ РОЗРОБКИ

№ П/П	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>10.12.2021-10.06.2022</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>02.05.2022-13.05.2022</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>16.05.2022-05.06.2022</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>20.05.2022-05.06.2022</i>	
5.	<i>Програмна реалізація системи</i>	<i>27.05.2022-05.06.2022</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>02.05.2022-05.06.2022</i>	
7.	<i>Захист програмного продукту</i>		
8.	<i>Передзахист</i>		
9.	<i>Захист</i>	<i>23.06.2022</i>	

					ІАЛЦ.467200.002 ТЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Застосунок візуального порівняння нуклеотидних послідовностей в
біомедичних дослідженнях»

Київ – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП	4
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	5
1.1 Сервіси вирівнювання послідовностей як інструменти для порівняння нуклеотидних послідовностей	5
1.2 Представлення нуклеотидної послідовності у вигляді тривимірної кривої.....	6
1.3 Представлення нуклеотидної послідовності на нотному стані.....	8
1.4 Представлення нуклеотидної послідовності у вигляді монандра чи триандра	9
1.4.1 Triander	9
1.4.2 jsTriander	10
1.4.3 DNK Visualisation.....	11
1.5 Порівняння існуючих рішень	12
ВИСНОВОК ДО РОЗДІЛУ 1	15
РОЗДІЛ 2. ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ.....	17
2.1 Основні технології реалізації.....	17
2.2 Canvas і SVG.....	18
2.3 Елементи малювання	19
2.3.1 Line	19
2.3.2 Polyline	20
2.3.3 Path.....	20
2.3.4 Порівняння елементів.....	21
ВИСНОВОК ДО РОЗДІЛУ 2	23
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	24

					ІАЛЦ.467200.003 ПЗ						
Зм.	Арк.	№ докум.	Підпис	Дата	Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях Пояснювальна записка						
Розробив		Дуплій Н.В.							Літ.	Аркуш	Аркушів
Перевірив		Кочура Ю.П.								1	55
Реценз.									КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-83		
Н. Контр.		Сімоненко В.П.									
Затвердив											

3.1 HTML-файл.....	24
3.2 Вибір елементів	29
3.3 Візуалізація	30
3.4 Імпорт	33
3.5 Формат FASTA.....	34
3.5.1 Заміна описів на відсутність символів.....	34
3.5.2 Використання описів для розділу тексту на нуклеотидні послідовності	35
3.6 Повідомлення про помилки користувача	36
3.7 Експорт.....	37
3.8 CSS.....	38
3.8.1 Загальна стилізація	38
3.8.2 Стилiзація кнопок	39
ВИСНОВОК ДО РОЗДІЛУ 3	43
РОЗДІЛ 4 ДОСЛІДЖЕННЯ ТА АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ.....	44
4.1 Інтерфейс	44
4.2 Інструкція роботи користувача з системою	48
4.3 Перелік використовуваного програмного забезпечення	49
ВИСНОВОК ДО РОЗДІЛУ 4	50
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК 1	3
ДОДАТОК 2.....	5
ДОДАТОК 3.....	7

ПЕРЕЛІК СКОРОЧЕНЬ

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
MUSCLE	MUltiple Sequence Comparison by Log- Expectation
ДНК	Дезоксирибонуклеїнова кислота
РНК	Рибонуклеїнова кислота
SVG	Scalable Vector Graphics

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Тема генної інженерії є дуже актуальною на даний момент. Особливо актуальність зросла з появою інфекційної хвороби COVID-19, яка поширилася по всій Землі. Значна частина дослідників вивчає ДНК вірусу SARS-Cov-2, а технологія мРНК є основою деяких вакцин.

У дослідників виникає потреба порівнювати нуклеотидні послідовності, тому дуже важливо, щоб для цього були ефективні інструменти.

У зв'язку цим у цій дипломній роботі розроблено застосунок для візуального порівняння нуклеотидних послідовностей, щоб спростити аналіз людиною.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Відомих програм для візуального порівняння нуклеотидних послідовностей практично немає. Але можна знайти програми або для порівняння, або для візуалізації.

1.1 Сервіси вирівнювання послідовностей як інструменти для порівняння нуклеотидних послідовностей

Вирівнювання необхідне для порівняння послідовностей у зв'язку з результатами мутацій. Якби зміни обмежувалися точковими мутаціями, тобто, заміною одного нуклеотиду на інший, у цьому не було б потреби. Але є такі явища, як вставка та делеція, які змінюють довжину послідовності та зсув її частин. Без вирівнювання їх важко розпізнати, а різниця між послідовностями може здатися більшою, ніж вона є насправді.

Програми для вирівнювання послідовностей іноді позначають позиції, де нуклеотиди однакові для всіх послідовностей. Наприклад, у MUSCLE (Multiple Sequence Comparison by Log-Expectation)[1] вони позначаються символом зірочки, якщо обрати формат виводу ClustalW або ClustalW (strict) (рис 1.1).

```
CLUSTAL multiple sequence alignment by MUSCLE (3.8)

ENA|BAA20512|BAA20512.1   ATGAGTCTCTCTGATAAGGACAAGGCTGCTGTGAAAGCCCTATGGGCTAAGATCAGCCCC
ENA|CAA28435|CAA28435.1   ATGTCTCTGACCAGGACTGAGAGGACCATCATCCTGTCCCTGTGGAGCAAGATCTCCACA
ENA|CAA23748|CAA23748.1   ATGGTGCTGTCTCCTGCCGACAAGACCAACGTCAAGGCCGCCTGGGGTAAGGTCGGCGCG
ENA|CAA24095|CAA24095.1   ATGGTGCTCTCTGGGGAAGACAAAAGCAACATCAAGGCTGCCTGGGGGAAGATTGGTGGC
***  ** *          ** *          *      *      ***  *** *

ENA|BAA20512|BAA20512.1   AAAGCCGATGATATCGGCGCTGAAGCTCTCGGCAGAATGCTGACCGTCTACCCCTCAGACC
ENA|CAA28435|CAA28435.1   CAGGCAGACGTCATTGGCACCAGACCCCTGGAGAGGCTCTTCTCCTGCTACCCGCAGGCC
ENA|CAA23748|CAA23748.1   CACGCTGGCGAGTATGGTGCGGAGGCCCTGGAGAGGATGTTCTCTGCTTCCCCACCACC
ENA|CAA24095|CAA24095.1   CATGGTGCTGAATATGGAGCTGAAGCCCTGGAAAGGATGTTGCTAGCTTCCCCACCACC
* * * * *      ** * ** * ** * * * * *      * * * * *      ** * ** * **
```

Рисунок. 1.1 – Вивід програми MUSCLE у форматі ClustalW

У такого представлення є декілька проблем. По-перше, важко співвіднести відсутність зірочок з відповідними їм частинами послідовності (хоча й простіше, ніж намагатися знайти різницю самотійно). По-друге, довгі послідовності в форматі тексту майже неможливо аналізувати людиною.

Якщо обрати формат HTML, позиції, де нуклеотиди однакові для всіх послідовностей, виділені кольором (рис 1.2). Таке представлення вирішує першу проблему, але не другу.

```

ENA | BAA20512 | BAA20512.1 | ATGagtCTCTCTgatAAgGACAAgGctgctgTgAAaGCCctTaTGGGctAAGATCaGcCcc
ENA | CAA28435 | CAA28435.1 | ATGtctCTgaCcagGActGAgGACCAcATCtCctGtCCctgTGGaGcAAGATctcCaCa
ENA | CAA23748 | CAA23748.1 | ATGgtGCTgTCTcctgccGACAAGACCAACgTCAAGGCCgCCTGGGGtAAGgTCGGCgCg
ENA | CAA24095 | CAA24095.1 | ATGgtGCTCTCTggGgAaGACAaAgCAACATCAAGGctgCCTGGGGgAAGATtGGtgGc

ENA | BAA20512 | BAA20512.1 | aAaGcCgatGAtAtcGGCGctGAAGctCTcGgcAGaATGcTgaCcgctCTACCCctcAgACC
ENA | CAA28435 | CAA28435.1 | CAgGCaGaCGtcAtTGGCaCcGAgaCCCTGGAgAGGcTtTctCctgCTACCCgcAgggCC
ENA | CAA23748 | CAA23748.1 | CAcGCTGgCGAgTATGGtGCgAgGCCCTGGAAGGATGTTcctgtcCTtCCCCAcCACC
ENA | CAA24095 | CAA24095.1 | CAtgTGctGAatATGgaGctGAAGCCCTGGAaAGGATGTTtgCtagCTtCCCCAcCACC

```

Рисунок. 1.2 – Вивід програми MUSCLE у форматі HTML

1.2 Представлення нуклеотидної послідовності у вигляді тривимірної кривої

У 1983 році було опубліковано статтю, у якій запропоновано представлення нуклеотидної послідовності у вигляді тривимірної кривої, яка також називається «Н-кривою» (рис 1.3).[2]

«Н-крива» складається з векторів, кожен з яких представляє нуклеотид (рис. 1.4). Вектор для А напрямлений за напрямом осі x, за напрямом осі y, проти напрямку осі z. Вектор для Т напрямлений за напрямом x, проти напрямку y, проти напрямку z. Вектор для С напрямлений проти x, проти напрямку y, проти напрямку z. Вектор для G напрямлений проти напрямку x, за напрямом y, проти напрямку z.

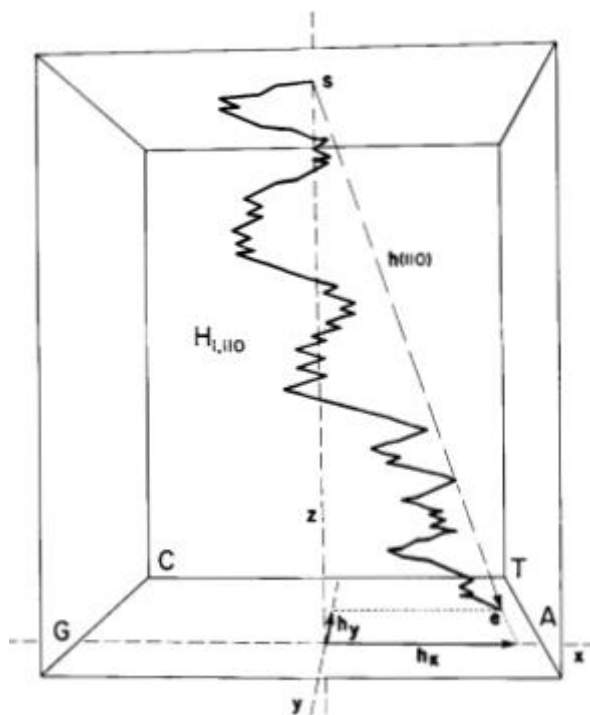


Рисунок 1.3 – Приклад «Н-кривої»[2]

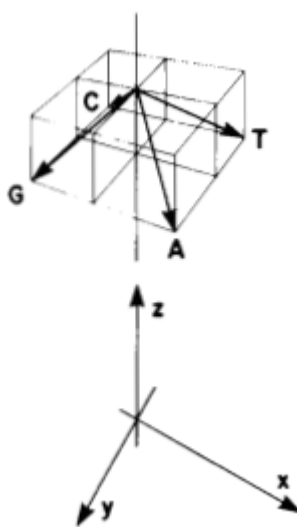


Рисунок 1.4 – Вектори, з яких складається «Н-крива»[2]

Оригінальна програма написана на Fortran. Але з тексту статті не зрозуміло, чи була вона у вільному доступі.

1.3 Представлення нуклеотидної послідовності на нотному стані

У 1986 році було опубліковано статтю про метод представлення на нотному стані.[5] ДНК представлена 4 інтерліній, відповідно G, A, T, C (Рис. 1.5). Якщо певний нуклеотид є на певній позиції, вона позначається на відповідній інтерлінії певним символом.

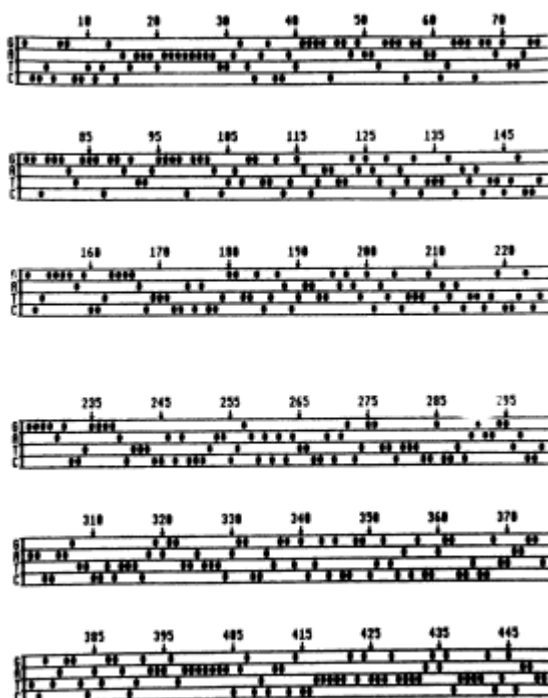


Рисунок 1.5 – приклад представлення ДНК цим методом[5]

Цей метод допомагає розпізнавати окремі патерни, але незручний для великих послідовностей.[4]

У цій статті також описано спосіб візуалізації порівняння нуклеотидних послідовностей. У ньому дві послідовності зображуються на одній інтерлінії, але різними символами.[5]

Програма є, але призначена для Acorn BBC Model B computer. Вихідний код включено в статтю про цей метод.[5]

1.4 Представлення нуклеотидної послідовності у вигляді монандра чи триандра

Обхід ДНК є двовимірною альтернативою «Н-кривих». Існують різні варіації: з різними довжинами векторів, з поділом на три криві і т.д.

Цей спосіб зручний для представлення довгих послідовностей.

На даний момент написано кілька програм, які візуалізують ДНК і РНК у вигляді триандрів та монандрів.

1.4.1 Triander

Програму можна завантажити на сайті Інституту клітинної біології та генетичної інженерії НАН України.

Там же можна завантажити вихідний код програми.

Програма написана на Lazarus.

Можна завантажити будь-який текстовий формат, але парсинг інформації обмежений. Програма розуміє рядки хедерів (або описів) формату fasta та не використовує цей текст у візуалізації, але сприймає всі інші рядки як одну послідовність, навіть якщо вони розділені хедером.

Є можливість експортувати результат у формат svg.

Програма малює триандри та монандри.

Налаштування Magnitude відповідає за відносну довжину векторів відносно одне одного. За Equal усі вектори мають одну довжину. При Natural довжина векторів така: C – 4, G – 3, T – 2, A – 1.[4]

За довжину одиничного відрізка відповідає налаштування Scale.

Частини триандра відрізняються одне від одного товщиною лінії, що добре при відсутності можливості розрізняти кольори.

					ІАЛЦ.467200.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

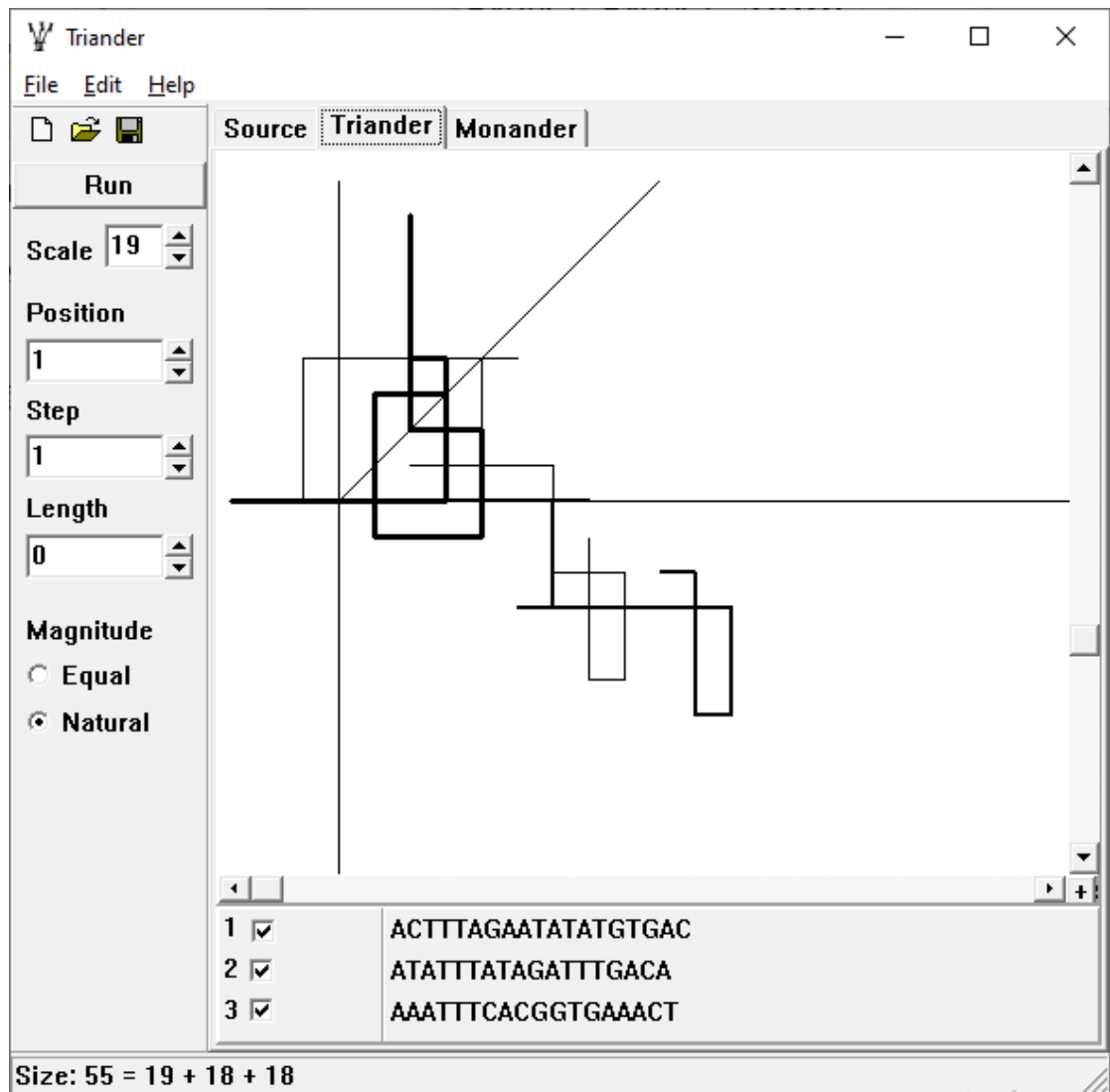


Рисунок 1.6 – приклад роботи програми Triander

Є можливість сховати частини тріандру.

Прокрутка відбувається за допомогою смуг прокручування.

Іноколи виникають проблеми з відображенням, але вони вирішуються при натисненні кнопки Run.

1.4.2 jsTriander

Ця програма є спрощеною on-line версією програми Triander.

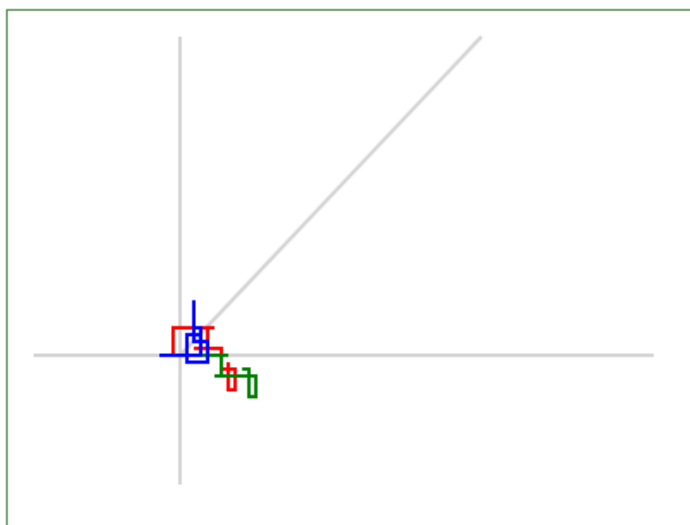
Програма написана на JavaScript.

Програма не підтримує читання файлів. Якщо дати програмі завдання візуалізувати набір символів, які не входять до набору A, C, T, G, U, сторінка перестане працювати.

Є можливість отримати текст для експорту у svg.

Програма малює лише триандри. Немає ніяких налаштувань. Довжини векторів такі ж, як Natural Magnitude у програмі Triander.

jsTriander



AAACATAAATTTTTATTGACATAAACTGGAAGTTTATGTTAGGATAAGCCAATC

Рисунок 1.7 – приклад роботи програми jsTriander

Частини триандра відрізняються кольором. При знебарвленні червоний та синій неможливо відрізнити одне від одного.

1.4.3 DNK Visualisation

Ця програма є веб-сторінкою.

Можна завантажити будь-який текстовий формат, але він буде сприйматися як одна послідовність.

Є можливість експортувати у формат svg.

Програма малює моноандри та триандри.

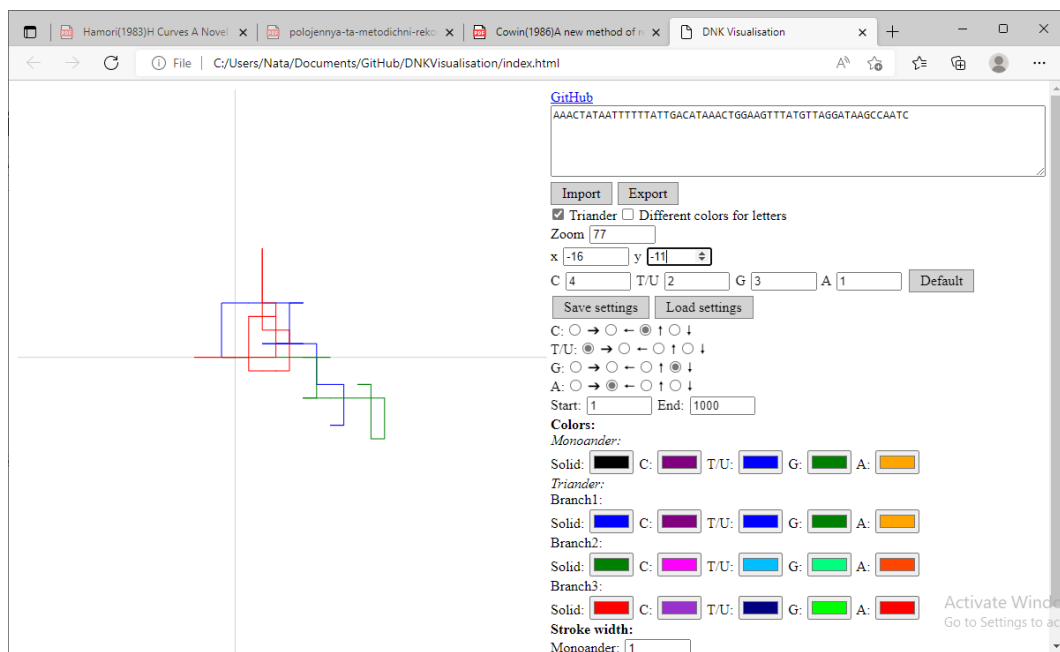


Рисунок 1.8 – приклад роботи програми DNKVisualisation

Можна налаштувати довжини та напрями векторів, кольори гілок тріандра та моноандра, а також їх товщину.

Прокрутка виконується за допомогою зміни налаштувань x та y.

1.5 Порівняння існуючих рішень

У таблиці 1.1 будуть описані переваги та недоліки методів представлення нуклеотидних послідовностей. Також будуть описані переваги та недоліки застосунків, які реалізують метод представлення тріандром/монандром. Сервіси вирівнювання послідовностей порівнювати між собою не має сенсу, а готових програм для інших методів по суті немає.

Таблиця 1.1. – Переваги та недоліки існуючих рішень

Рішення	Переваги	Недоліки
Сервіси вирівнювання послідовностей	1) Виділяють однакові нуклеотиди 2) Вирівнюють послідовності	1) Незручне порівняння

Продовження таблиці 1.1

Рішення	Переваги	Недоліки
Н-крива	1) Зручність представлення для людського ока	1) Програми немає у вільному доступі 2) Не передбачено порівняння 3) Складність проєкції на двовимірний простір
Нотний стан	1) Зручність представлення для людського ока та комп'ютера	1) Незручність аналізу довгих послідовностей 2) Необхідність компілювати програму
Монандр/тріандр	1) Зручність представлення для людського ока 2) Зручність аналізу довгих послідовностей	1) Не передбачено порівняння
<ul style="list-style-type: none"> • Triander 	1) Гілки добре розпізнаються без кольорів	1) Незначні проблеми з відображенням 2) Необхідність завантажувати на комп'ютер
<ul style="list-style-type: none"> • jsTriander 	1) Не вимагає завантаження на комп'ютер	1) Знебарвлення позбавляє можливості розрізняти гілки 2) Перестає працювати при неправильних вхідних даних

Кінець таблиці 1.1

Рішення	Переваги	Недоліки
<ul style="list-style-type: none"> DNK Visualisation 	1) Гілки добре розпізнаються навіть без кольорів (якщо налаштувати товщину)	1) Незручна прокрутка

За таблицею 1.1 можна зробити висновок, що монандри/тріандри найбільш зручні для аналізу нуклеотидних послідовностей. Також при розробці варто звернути увагу на розпізнавання без кольорів, роботу програми при неправильних вхідних даних і прокрутку, щоб уникнути проблем однієї з існуючих реалізацій тріандра.

ВИСНОВОК ДО РОЗДІЛУ 1

У першому розділі розглянуто існуючі рішення для порівняння та візуалізації нуклеотидних послідовностей.

Прикладів візуального порівняння майже не має, тому порівнюються в першу чергу способи представлення послідовностей та програми, які їх реалізують. Єдиним прикладом поєднання нетекстового представлення та порівняння є модифікація методу представлення на нотному стані, але розгляду лише неї недостатньо для огляду існуючих рішень.

Розглянуто суть сервісів вирівнювання послідовностей та їхнє використання для порівняння нуклеотидних послідовностей. Хоча деякі з цих сервісів позначають однакові нуклеотиди на позиції, але аналіз тексту незручний для людини.

Розглянуто представлення нуклеотидної послідовності у вигляді тривимірної кривої, на нотному стані та у вигляді монандра чи триандра (за допомогою обходу ДНК). Усі ці методи простіші для аналізу людиною, ніж текстове представлення. Але тривимірну криву проблематично проектувати на двовимірний простір (через що і з'явилися двовимірні представлення), а представлення на нотному стані погано підходить для аналізу довгих послідовностей.

Вирішено взяти за основу візуалізації обхід ДНК, який є двовимірною модифікацією Н-кривої.

Огляд існуючих рішень показав, що задача є актуальною, бо готових програм, які б її вирішували, немає. Огляд існуючих програм для візуалізації (у яких не візуального порівняння) допоміг зрозуміти, на що варто звернути увагу при створенні застосунку. У даному випадку, на якість візуалізації результатів

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

при відсутності кольору та відповідь програми на неочікувані символи у завантажених файлах.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

РОЗДІЛ 2. ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ

2.1 Основні технології реалізації

Щоб визначитися з технологіями реалізації, треба спочатку визначитися, на яких платформах буде працювати програма.

За планом з застосунком можна буде працювати онлайн або завантажити його на комп'ютер та працювати з ним без підключення до інтернету. Отже, застосунок буде веб-сторінкою, а виконання програми буде на клієнтській частині, щоб не було залежності від серверу.

Для розмітки використовується мова HTML.

Для стилів використовується CSS.

Є кілька способів застосувати CSS до елементів[13]:

- внутрішній – CSS усередині тегу style,
- зовнішній – CSS у окремому файлі,
- в лінію – CSS у атрибуті style тегу, до елемента якого застосовується стиль.

У цьому дипломному проєкті буде використовуватися зовнішній спосіб застосувати CSS до елементів, щоб HTML-файл не був перегружений і відділити розмітку від стилю.

Було створено кілька мов програмування, які призначалися для написання поведінки веб-сторінки на стороні клієнта. Серед них Jscript[10] або VBScript[11]. Але не всі з них практично використовувати (наприклад, Jscript і VBScript більше не підтримуються Microsoft і за замовчуванням вимкнені у деяких новіших версіях Internet Explorer).

При виборі важливо знати, які мови програмування підтримуються популярними браузерами. Якщо обрати технологію, яка підтримується якимось

					ІАЛЦ.467200.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

рідкісним браузером, у кращому випадку користувачам буде незручно його встановлювати, а в гіршому вони подумують, що програма не працює або підуть шукати альтернативи.

Як мова для поведінки буде використовуватися мова JavaScript, бо вона призначена для виконання клієнтської частини та підтримується більшістю браузерів, на відміну від інших мов програмування.

Для мови JavaScript є додаткові бібліотеки та фреймворки, але вони не будуть використовуватися для написання цієї програми.

Ці технології зазвичай використовуються для проєктів, які складно зробити стандартними засобами, але реалізація на чистому JavaScript достатньо проста, тому немає сенсу шукати бібліотеки чи фреймворки для спрощення.

До того ж, у бібліотеки та фреймворків є інші недоліки. Серед них виділимо збільшення місця, яке займає проєкт, та потенційні проблеми з безпекою, які важче контролювати, ніж якби код обмежувався власним.[6]

2.2 Canvas і SVG

Для малювання на частому JavaScript поширені Canvas і SVG. Тому необхідно розглянути різницю між ними для того, щоб визначити, яку з цих технологій використати.

За допомогою Canvas малюється растрова графіка, хоча часто для цього використовуються форми. Растрова графіка погано масштабується, тому Canvas погано підходить для графіки, для якої передбачено зміну розміру. Але Canvas краще працює з більшою кількістю об'єктів.[8]

За допомогою SVG малюється векторна графіка. Векторна графіка добре масштабується, тому SVG добре підходить для графіки, для якої передбачено зміну розміру. Але SVG краще працює з меншою кількістю об'єктів.[8]

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

Прийнято рішення використати SVG, бо ця технологія допомагає створювати векторну графіку, яка у нашому випадку більш корисна, ніж растрова. Більшість переваг Canvas у цьому проєкті неважлива, враховуючи, що багато об'єктів використовуватися не буде. Пояснення цьому буде пізніше, під час розгляду елементів малювання.

2.3 Елементи малювання

При використанні SVG є кілька способів намалювати ламані лінії з розривами. Розглянемо різні об'єкти для малювання, щоб визначити, який з них краще нам підходить.

2.3.1 Line

Елемент line малює пряму. Основні атрибути:

- $x1$ – x-координата початку прямої,
- $y1$ – y-координата початку прямої,
- $x2$ – x-координата кінця прямої,
- $y2$ – y-координата кінця прямої.

Отже, намалювати ламану можна за допомогою великої кількості елементів line.

SVG краще працює з меншою кількістю об'єктів, і велика кількість об'єктів line набагато менш ефективна, ніж кілька polyline чи path. Це підтвердилося під час роботи над іншим проєктом, у якому було вирішено відмовитися від line на користь path. Але більш точні результати будуть наведені у пункті 2.3.4.

					ІАЛЦ.467200.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

2.3.2 Polyline

Елемент `polyline` малює ламану, яка складається з кількох відрізків. Основний атрибут – `points`. Він складається з набору точок, між якими будуть проведені лінії. Координати різних точок розділені пробілами, а `x`-координата та `y`-координата – комою.

Теоретично, він має бути ефективнішим за `line`, бо вимагає меншої кількості об'єктів для такого ж результату.

Якби у нас не було передбачено розриви, вірогідно, кількох `polyline` було б достатньо. Але ми маємо передбачати гепи, які додаються до послідовностей для їхнього вирівнювання. Якщо ламані будуть розриватися, доведеться створювати новий елемент `polyline` кожен раз, коли з'являється розрив. Отже, варто звернути увагу на наступний елемент.

2.3.3 Path

Елемент `path` малює шлях, який може бути розірваним. Цей елемент є найбільш універсальних із перелічених у цьому пункті. Усі базові форми можуть бути створені цим елементом.[12]

Основний атрибут – `d`, у якому прописано шлях. Тут не будуть розглядатися усі команди шляху, тому що не всі з них нам знадобляться. Нам потріб-но лише дві команди: перемістити до (`MoveTo`) і провести лінію до (`LineTo`).

Елемент `path` за замовчуванням також має заливку всередині намальованого шляху, якщо її не прибрати за допомогою атрибуту.

Теоретично, елемент `path` ефективніший за `line` та `polyline`, бо він дозволяє використати мінімальну кількість об'єктів для досягнення нашої мети. Але варто перевірити на практиці швидкість виконання при використанні різних елементів для малювання.

					ІАЛЦ.467200.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

2.3.4 Порівняння елементів

Для того, щоб перевірити теорію, напишемо програму, яка малює один і той самий набір ліній трьома різними способами: за допомогою елемента `line`, за допомогою елемента `polyline` та за допомогою елемента `path`.

Сама програма для порівняння детально розглядатися не буде, бо вона не є частиною застосунку, а лише способом перевірити ефективність технологій.

```
let svg = document.getElementById('svg');

function drawPath(path, color, width){
  let element = document.createElementNS('http://www.w3.org/2000/svg', 'path');
  element.setAttribute('d', path);
  element.setAttribute('stroke', color);
  element.setAttribute('stroke-width', width);
  element.setAttribute('fill-opacity', 0);
  element.setAttribute('stroke-linecap', 'round');
  svg.appendChild(element);
}

function drawPolyline(points, color, width){
  let element = document.createElementNS('http://www.w3.org/2000/svg', 'polyline');
  element.setAttribute('points', points);
  element.setAttribute('stroke', color);
  element.setAttribute('stroke-width', width);
  element.setAttribute('fill-opacity', 0);
  element.setAttribute('stroke-linecap', 'round');
  svg.appendChild(element);
}
```

Рис. 2.1 – Функції створення елементів `line` і `polyline`.

```
function drawLine(x1, y1, x2, y2, color, width){
  let element = document.createElementNS('http://www.w3.org/2000/svg', 'line');
  element.setAttribute('x1', x1);
  element.setAttribute('y1', y1);
  element.setAttribute('x2', x2);
  element.setAttribute('y2', y2);
  element.setAttribute('stroke', color);
  element.setAttribute('stroke-width', width);
  element.setAttribute('fill-opacity', 0);
  element.setAttribute('stroke-linecap', 'round');
  svg.appendChild(element);
}
```

Рис. 2.2 – Функція створення елемента `path`.

```

console.time('Lines');
for (let i = 0; i < 1000; i++){
    drawLine(i * 10, 0, (i + 1) * 10, 10, 'black', 1);
}
console.timeEnd('Lines');

console.time('Polyline');
for (let i = 0; i < 1000; i++){
    drawPolyline((i * 10) + ',0 ' + ((i + 1) * 10) + ',10', 'black', 1);
}
console.timeEnd('Polyline');

console.time('Path');
path = '';
for (let i = 0; i < 1000; i++){
    path += 'M ' + (i * 10) + ',0\nL ' + ((i + 1) * 10) + ',10\n';
}
drawPath(path, 'black', 1);
console.timeEnd('Path');

```

Рис. 2.3 – Код малювання 1000 ліній.

Метод `console.time` запускає секундомір з певною назвою (у нашому випадку – `Lines`, `Polyline` та `Path`). Метод `console.timeEnd` зупиняє секундомір з певною назвою та виводить результат у консоль.

```

Lines: 95.30517578125 ms main.js:40
Polyline: 104.655029296875 ms main.js:46
Path: 1.466064453125 ms main.js:55
>

```

Рис. 2.4

Як ми можемо побачити з результатів тестування, теорія справдилася: використання одного елемента `path` набагато ефективніше по часу виконання, ніж використання тисячі елементів `line` чи тисячі елементів `polyline` для того самого результату.

ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі розглянуто та обрано технології реалізації застосунку, зважаючи на те, що його вирішено зробити веб-сторінкою, яка може не лише бути частиною веб-сайту в Інтернеті, а й працювати оффлайн лише за наявності браузера.

При дослідженні основних технологій виявилось, що вибір невеликий. Обрано мову розмітки HTML, мову стилізації CSS і скриптову мову JavaScript.

Вирішено, що додаткові бібліотеки та фреймворки не потрібні, бо для вирішення задачі вигідніше використовувати стандартні засоби.

Між Canvas і SVG було обрано SVG, який призначений для векторної графіки, яка добре масштабується.

Для малювання набору ліній було обрано path через меншу необхідну кількість об'єктів і, як наслідок, більшу швидкодію. Його ефективність у порівнянні з line і polyline доведено за допомогою спеціально написаної програми, яка малює 1000 ліній, не з'єднаних між собою.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 HTML-файл

Для роботи веб-застосунків необхідно створити хоча б один HTML-файл. Тому в першу чергу ми розглянемо, як побудовано цей файл (рис. 3.1, рис. 3.2).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>VisualComparison</title>
  </head>
  <body>
    <label for="import1" class="button">
      DNA1
    </label>
    <input type="file" id="import1">
    <span id="seq1">There are no sequence 1</span>
    <br>
    <label for="import2" class="button">
      DNA2
    </label>
    <input type="file" id="import2">
    <span id="seq2">There are no sequence 2</span>
    <br>
    <input type="checkbox" id="natural">
    <label for="natural">
      Natural
    </label>
    <input type="color" id="color1" value="#FF0000">
    <input type="color" id="color2" value="#0000FF">
    <br>
    <input type="button" id="run" value="Compare" class="button">
    <input type="button" id="export" value="Export" class="button">
    <div>
      <svg xmlns="http://www.w3.org/2000/svg" id="comparison"></svg>
    </div>
```

Рис 3.1 – Початок файлу index.html

```
<link rel="stylesheet" href="style.css">
<script src="main.js"></script>
</body>
</html>
```

Рис. 3.2 – Кінець файлу index.html

Створення цього файлу ще не робить програму здатною вирішувати поставлену перед нею задачу, але ми вже можемо побачити приблизний вигляд програми, не стилізований файлом.

Розглянемо детальніше компоненти.

Тег `<!DOCTYPE html>` необхідний для гарантії, що браузер буде дотримуватися специфікацій, коли це можливо[7]. Його відсутність впливає на відображення цієї сторінки.

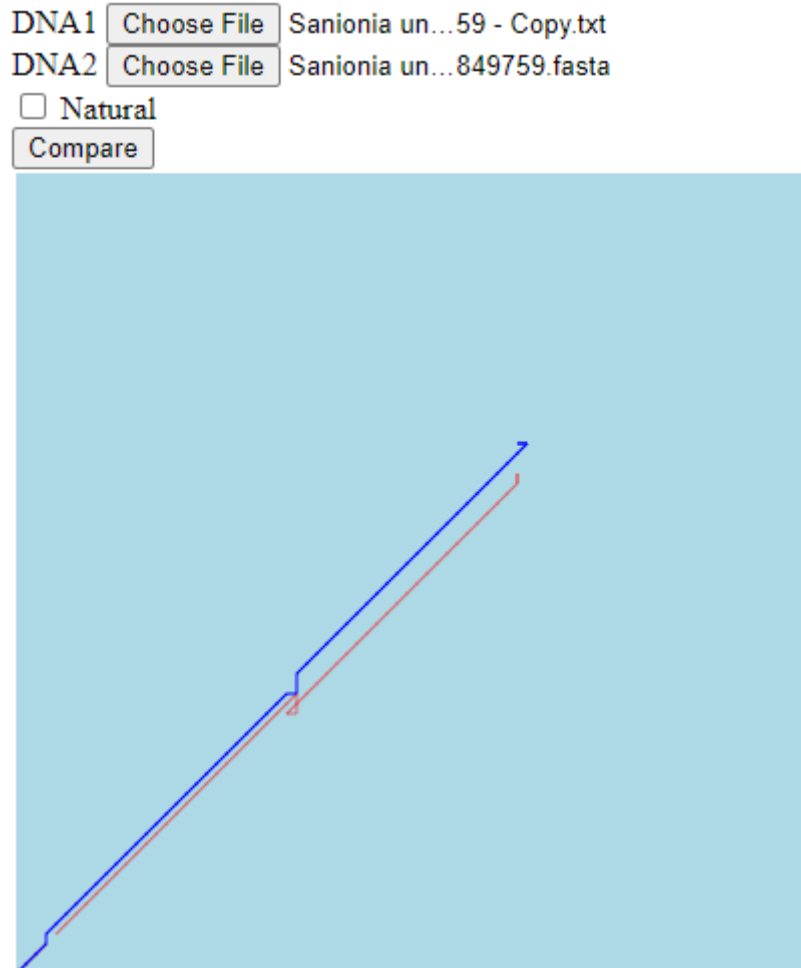


Рис. 3.3 – Приклад відображення з `<!DOCTYPE html>`

DNA1 Choose File No file chosen
DNA2 Choose File No file chosen
 Natural
Compare

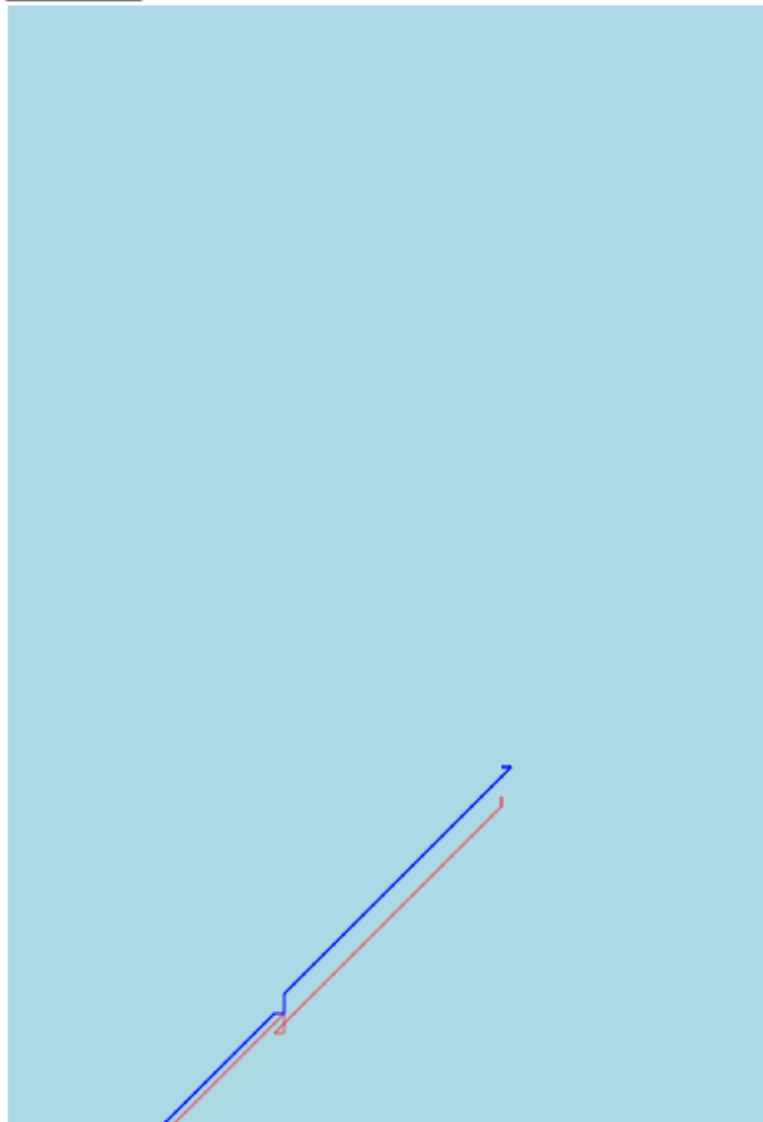


Рис. 3.4 – Приклад відображення без <!DOCTYPE html>

Тег html є корневим тегом HTML-документу, усередині якого міститься весь зміст веб-сторінки та додаткова інформація для браузера, наприклад, заголовки або кодування.

Усередині тегу title міститься заголовок, який відображається в браузері (рис. 3.5).

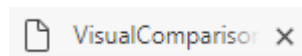


Рис. 3.5 – Заголовок у браузері

Тег body містить безпосередньо зміст веб-сторінки.

У цьому проєкті використовуються три види тегу input: button, file і color.

Тип button використовується для звичайних кнопок. Сам по собі він не має ніякої поведінки, але її можна прив'язати за допомогою скриптової мови. Інші використані типи input є спеціалізованими.

Тип file використовується для кнопок завантаження файлів. При натисненні на нього викликається діалог вибору файлу.

Тип color використовується для кнопок вибору кольору. При натисненні на нього викликається діалог вибору кольору, який залежить від браузера.

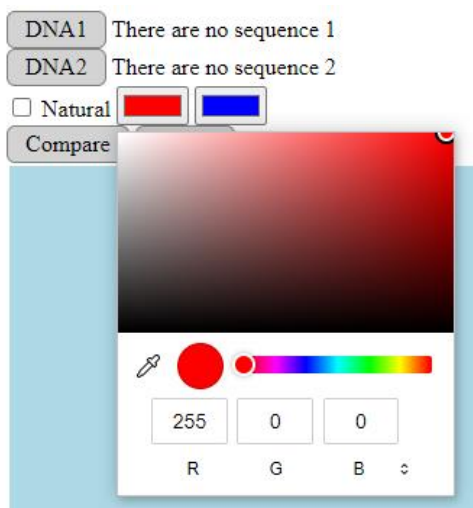


Рис. 3.6 – Діалог вибору кольору в Microsoft Edge

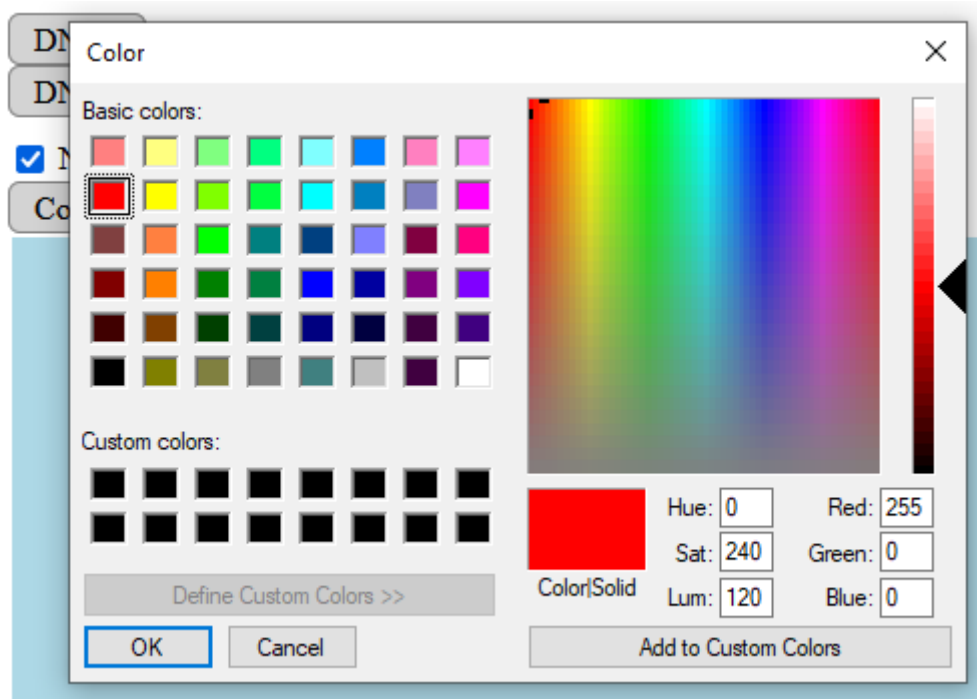


Рис. 3.7 – Діалог вибору кольору в Mozilla Firefox

Тег `label` є підписом до елемента веб-сторінки.

Тег `span` є строковим контейнером без ефектів і використовується для стилізації за допомогою CSS. У даному випадку змістом цих елементів будуть завантажені нуклеотидні послідовності, щоб користувачі могли бачити, чи завантажили вони файл з послідовністю.

Тег `div` є блочним контейнером без ефектів і також використовується для стилізації за допомогою CSS.

Тег `svg` буде містити результат візуалізації, але малювати його буде JavaScript.

Тег `link` у даному випадку визначає, який файл CSS буде використовуватися для стилізації веб-сторінки.

Тег `script` підключає виконуваний JavaScript код з файлу або з тексту всередині цього тегу.

Деякі елементи мають атрибут `«id»` - ідентифікатор, за яким буде отримано посилання на елемент. Детальніше це буде розглянуто у наступному пункті.

Також деякі елементи мають атрибут «class». Він буде використовуватися для більш зручного застосування стилів до елементів, створених різними тегами.

3.2 Вибір елементів

Для того, щоб працювати з елементами HTML, необхідно мати посилання на них (рис. 3.8). Тому ми почнемо розгляд нашого JavaScript-файлу саме з отримання цих посилань.

```
//DEFINITIONS
let svg = document.getElementById('comparison');
let oneVectorLength = 5;

let importInput1 = document.getElementById('import1');
let importInput2 = document.getElementById('import2');

let naturalCheckbox = document.getElementById('natural');

let compareButton = document.getElementById('run');
let exportButton = document.getElementById('export');

let sequenceText1 = document.getElementById('seq1');
let sequenceText2 = document.getElementById('seq2');

let colorPicker1 = document.getElementById('color1');
let colorPicker2 = document.getElementById('color2');
```

Рис. 3.8 – Вибір елементів

Є кілька способів отримати посилання[9]:

- за класом (`Document.getElementsByClassName(String className)`),
- за тегом (`Document.getElementsByTagName(String tagName)`),
- за тегом у просторі імен (`Document.getElementsByTagNameNS(String namespace, String tagName)`),
- за ідентифікатором (`Document.getElementById(String id)`),
- за іменем (`Document.getElementsByName(String name)`).

Отримувати посилання за тегом не варто, бо ми працюємо з конкретними елементами, а не всіма елементами з певним тегом водночас. Серед них можна обрати елемент з певним ідентифікатором, але в цьому немає потреби, бо можна отримати посилання за ідентифікатором. За цієї ж причини немає сенсу отримувати посилання за тегом у просторі імен.

Ідентифікатор унікальний для елемента, тому отримувати посилання будемо за ним. За тієї ж причини використовується саме атрибут `id`, а не `name`.

У цьому проєкті необхідно отримати посилання на такі елементи:

- `svg`-елемент, у якому буде відбуватися вся візуалізація,
- кнопки завантаження файлів,
- флажок «Natural»,
- кнопку для запуску візуалізації,
- текст, який містить завантажені нуклеотидні послідовності,
- кнопки для вибору кольору.

3.3 Візуалізація

Візуалізація є однією з найважливіших частин цього проєкту, тому необхідно в першу чергу реалізувати її.

```
function partOfPath(x1, y1, x2, y2){
  return 'M ' + (x1 * oneVectorLength) + ', ' + (y1 * oneVectorLength) +
  '\n L ' + (x2 * oneVectorLength) + ', ' + (y2 * oneVectorLength) + '\n';
}
```

Рис. 3.9 – Функція формування частини шляху

Ця функція створена для того, щоб спростити формування частини шляху та редагування цієї частини коду. `M` позначає рух, `L` – малювання лінії від теперішніх координат до майбутніх координат.

```

let letters = {
  'A': {x: -1, y: 0, length: 1},
  'C': {x: 0, y: -1, length: 4},
  'T': {x: 1, y: 0, length: 2},
  'U': {x: 1, y: 0, length: 2},
  'G': {x: 0, y: 1, length: 3}
};

```

Рис. 3.10 – Визначення значень для різних літер

Для кожного символу є такі значення:

- x – напрям по осі x,
- y – напрям по осі y,

length – довжина відрізка при налаштуванні Natural.

```

function drawComparison(){
  svg.innerHTML = "";
  let path1 = "";
  let path2 = "";
  let x1 = 0;
  let y1 = svg.getBoundingClientRect().height / oneVectorLength;
  let x2 = 0;
  let y2 = svg.getBoundingClientRect().height / oneVectorLength;
  let length = Math.min(XNA[0].length, XNA[1].length);
  let weirdSymbols = false;
  for (let i = 0; i < length; i++){
    if (XNA[0][i] == XNA[1][i] && letters[XNA[0][i]]){
      path1 += partOfPath(x1, y1, x1 + 1, y1 - 1);
      path2 += partOfPath(x2, y2, x2 + 1, y2 - 1);
      x1 += 1;
      y1 -= 1;
      x2 += 1;
      y2 -= 1;
    } else {
      if (letters[XNA[0][i]]){
        l = 1;
        if (naturalCheckbox.checked == true){
          l = letters[XNA[0][i]].length;
        }
        path1 += partOfPath(x1, y1, x1 + letters[XNA[0][i]].x * l, y1 + letters[XNA[0][i]].y * l);
        x1 += letters[XNA[0][i]].x * l;
        y1 += letters[XNA[0][i]].y * l;
      }
    }
  }
}

```

Рис. 3.11 – Перша частина функції візуалізації порівняння

```

else if (XNA[0][i] == "-"){
    path1 += 'M ' + (x1 + oneVectorLength) + ', ' + (y1 - oneVectorLength) + "\n";
    x1 += 1;
    y1 -= 1;
}
else if (!weirdSymbols) {
    weirdSymbols = true;
    alert('Symbol ' + XNA[0][i] + " can't be processed");
}
if (letters[XNA[1][i]]){
    l = 1;
    if (naturalCheckbox.checked == true){
        l = letters[XNA[1][i]].length;
    }
    path2 += partOfPath(x2, y2, x2 + letters[XNA[1][i]].x * l, y2 + letters[XNA[1][i]].y * l);
    x2 += letters[XNA[1][i]].x * l;
    y2 += letters[XNA[1][i]].y * l;
}
else if (XNA[1][i] == "-"){
    path2 += 'M ' + (x2 + oneVectorLength) + ', ' + (y2 - oneVectorLength) + "\n";
    x2 += 1;
    y2 -= 1;
}
else if (!weirdSymbols) {
    weirdSymbols = true;
    alert('Symbol ' + XNA[1][i] + " can't be processed");
}
}
}

```

Рис. 3.12 – Друга частина функції візуалізації порівняння

```

path1 += 'z';
path2 += 'z';
drawPath(path1, color1, 0.5);
drawPath(path2, color2, 1);
}

```

Рис. 3.13 – Третя частина функції візуалізації порівняння

Формування шляхів виконується посимвольно у функції drawComparison, яка запускається при натисненні на кнопку «Compare». Якщо символи співпадають, малюється діагональна лінія. Інакше малюється лінія, яка відповідає символу. Довжина лінії залежить від символу та налаштування «Natural». Значення Natural для кожного символу наведені вище, у рис. 3.10. Символ гепу («->») не входить у список, бо при його наявності потрібно не малювати лінію, а робити пропуск за допомогою M.

Коли шлях сформовано, він подається у функцію drawPath разом з кольором і товщиною шляху. Кольори, якими малюються шляхи, задаються користувачами.

```
function drawPath(path, color, width){
  let element = document.createElementNS('http://www.w3.org/2000/svg', 'path');
  element.setAttribute('d', path);
  element.setAttribute('stroke', color);
  element.setAttribute('stroke-width', width);
  element.setAttribute('stroke-linecap', 'round');
  svg.appendChild(element);
}
```

Рис. 3.14 – Функція малювання шляху.

Ця функція написана для спрощення додавання шляху на полотно. У новоствореного елемента є такі атрибути:

- d – сам шлях,
- stroke – колір обводки,
- stroke-width – товщина обводки,
- stroke-linecap - впливає на кінці відрізків, з яких складається path (у даному випадку до кінців відрізків буде додано заокруглення значенням «round»).

3.4 Імпорт

```
importInput1.addEventListener("change", () => {
  let file = importInput1.files[0];
  let reader = new FileReader();
  reader.readAsText(file);
  reader.onload = function() {
    read_result = reader.result;
    XNA[0] = read_result;
    reg = />.*</i>;
    XNA[0] = XNA[0].split(reg).filter(function(value){return value != ''})[0];
    XNA[0] = XNA[0].replaceAll('\r', '');
    XNA[0] = XNA[0].replaceAll('\n', '');
    sequenceText1.textContent = XNA[0];
  };
});
```

Рис. 3.15 – Події кнопок імпорту

У кінці обробки тексту послідовності від записується у елементи sequenceText1 і sequenceText2 відповідно.

3.5 Формат FASTA

До цього розробка програми велася з розрахунком на те, що у текстовому файлі не буде нічого, крім нуклеотидної послідовності. Але часто використовуються інші формати, які бажано правильно обробляти. Один з таких форматів – FASTA, який також містить ідентифікатори та іноді коментарі.

Описи нуклеотидних послідовностей починаються зі символу «>» (більше). Описи не є частинами нуклеотидних послідовностей, тому можна їх просто ігнорувати.

3.5.1 Заміна описів на відсутність символів

Для обробки описів нам знадобляться регулярні вирази.

```
reg = />.*\/ig;
DNA1 = DNA1.replaceAll(reg, '');
DNA1 = DNA1.replaceAll('\r', '');
DNA1 = DNA1.replaceAll('\n', '');
```

Рис. 3.16 – Приклад використання глобальної заміни та регулярного виразу

Є кілька способів створити регулярний вираз через JavaScript:

- з використанням літералу регулярного виразу,
- викликаючи функцію конструктора об'єкту RegExp.

У нашому випадку має сенс використовувати літерал регулярного виразу, бо регулярний вираз сталий, а використання літералу збільшить продуктивність у порівнянні з викликом функції конструктора.[15]

Регулярний вираз для прибирання описів – «/>.*\/ig».

					ІАЛЦ.467200.003 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

Сам регулярний вираз міститься між символами «/». Символ «g» у кінці є прапорцем глобального пошуку, тобто, будуть знайдені усі підрядки, які відповідають регулярному виразу.

Регулярний вираз записаний як «>.*», де «.» - будь-який символ, а «*» - будь-яка кількість попереднього символу підряд, включаючи його відсутність. Отже, будуть замінені усі рядки, які починаються з символу «>» і мають після нього будь-яку кількість символів.

Для заміни регулярного виразу на набір символів підходить як метод `replaceAll`, так і метод `replace`. Але є причина, чому вирішено використати саме `replaceAll`. Якщо використати регулярний вираз без прапорця глобального пошуку як аргумент методу `replaceAll`, консоль видасть помилку. Якщо використати такий вираз як аргумент методу `replace`, метод замінить перший набір символів, який підходить. Отже, метод `replaceAll` обраний для того, щоб було легше згадати про забутий прапорець.

Крім того, щоб прибрати описи, варто також прибрати спецсимволи «\r» і «\n», тому для них теж зроблена глобальна заміна.

3.5.2 Використання описів для розділу тексту на нуклеотидні послідовності

Але в звичайному ігноруванні описів є один недолік: у файлах формату FASTA може бути більше однієї нуклеотидної послідовності, і вони розділені описами, тому якщо просто прибрати описи, нуклеотидні послідовності будуть сприйматися як одна, що робить роботу програми некоректною. Отже, потрібно використовувати описи як набір символів, який розділяє нуклеотидні послідовності, навіть якщо в реалізації програми буде використовуватися лише перша послідовність.

Для цього ми замінимо метод `replaceAll` на метод `split`, який теж приймає регулярні вирази як аргумент (рис. 3.17). Для його роботи необов'язково, щоб

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

регулярний вираз мав прапорець глобального пошуку, але наявність цього прапорця не нашкодить, і прибирати його не будемо.

```
importInput1.addEventListener("change", () => {
  let file = importInput1.files[0];
  let reader = new FileReader();
  reader.readAsText(file);
  reader.onload = function() {
    read_result = reader.result;
    XNA[0] = read_result;
    reg = />.*\/i;
    XNA[0] = XNA[0].split(reg).filter(function(value){return value != ''})[0];
    XNA[0] = XNA[0].replaceAll('\r', '');
    XNA[0] = XNA[0].replaceAll('\n', '');
  };
});
```

Рис. 3.17 – Вибір першої нуклеотидної послідовності з файлу.

Спочатку вміст файлу розділяється за регулярним виразом. Після цього за допомогою методу `filter` видаляються рядки без символів. У змінну записується лише перший елемент списку, щоб спростити реалізацію.

3.6 Повідомлення про помилки користувача

Якщо користувачем завантажений файл з неочікуваним змістом, необхідно про це повідомити. До того ж, треба повідомити про це таким чином, щоб було зрозуміло, у чому саме проблема, і користувачу не довелося витратити забагато часу на те, щоб знайти причину некоректної роботи застосунку.

Для повідомлень від веб-сторінок використовується функція `alert()`.

У змінній `weirdSymbols` зберігається значення про те, чи є неочікувані символи у нуклеотидній послідовності. Вона потрібна для того, щоб сторінка не видавала тисячі повідомлень про неочікувані символи.

```
else if (!weirdSymbols) {
  weirdSymbols = true;
  alert('Symbol ' + XNA[0][i] + " can't be processed");
}
```

Рис. 3.18 – Повідомлення про неочікуваний символ

Значення змінної `weirdSymbols` змінюється на `true`, а веб-сторінка показує повідомлення про перший неочікуваний символ.

3.7 Експорт

Користувачу може бути потрібно зберегти результати візуалізації у файл. Тому варто зробити експорт результатів у SVG (рис. 3.19).

```
exportButton.addEventListener("click", () => {
  let svgData = svg.outerHTML;
  let svgBlob = new Blob([svgData], {type:"image/svg+xml"});
  let svgUrl = URL.createObjectURL(svgBlob);
  let downloadLink = document.createElement("a");
  downloadLink.href = svgUrl;
  downloadLink.download = "xna_visual_compare.svg";
  document.body.appendChild(downloadLink);
  downloadLink.click();
  document.body.removeChild(downloadLink);
});
```

Рисунок 3.19 – Експорт SVG-файлу[16]

Використаний код оснований на відповіді зі StackOverflow від DaveTheScientist.

Експорт тут відбувається за таким алгоритмом:

- у змінну `svgData` записується зміст полотна,
- створюється об'єкт типу `Blob`, зі змістом `svgData` і типом MIME `image/svg+xml`,
- створюється URL-посилання з об'єкту типу `Blob`,
- створюється елемент з тегом «а»,
- новоствореному елементу задається атрибут посилання,
- новоствореному елементу задається назва файлу, який буде завантажено,
- до документу додається новостворений елемент,
- імітується клік на новостворений елемент,

- з документу прибирається новостворений елемент.

3.8 CSS

Файл CSS потрібен для стилізації сторінки. Його наявність не обов'язкова для працездатності застосунку, але корисна для його оформлення.

3.8.1 Загальна стилізація

```
svg {  
  min-width: 400px;  
  min-height: 400px;  
  width: 100%;  
  height: 100%;  
  background-color: lightblue;  
}  
div {  
  padding: 2px;  
}
```

Рис. 3.20 – Налаштування елементів svg і div

У цьому файлі визначено, що:

- мінімальна ширина полотна – 400 пікселів,
- мінімальна висота полотна – 400 пікселів,
- ширина полотна – 100%,
- висота полотна – 100%,
- колір фону полотна – світло-синій,
- відступ від країв – 2 пікселі.

3.8.2 Стилiзацiя кнопок

Стилiзацiя також допомагає повнiстю переробити вигляд деяких елементiв (у даному випадку – вигляд елементу «input» з типом «file»). Частина файлу, яка за це відповідає, наведена в рисунку 3.22.

```
.button {  
  background-color: lightgray;  
  border: 1px solid gray;  
  border-radius: 6px;  
  display: inline-block;  
  padding: 3px 12px;  
  cursor: pointer;  
  font-size: inherit;  
  font-family: inherit;  
}
```

Рис. 3.22 – Змiна зовнiшнього вигляду кнопки завантаження файлу

Для необхідної стилiзацiї змiнюється вiдображення таких елементiв:

- елементiв «input»,
- елементiв з класом button (а саме елементiв «label» i елементу кнопки «Compare»).

Для того, щоб прибрати стандартне вiдображення кнопки завантаження файлу, параметр display встановлено у значення none. Але вiдсутнiсть вiдображення кнопки впливає не лише графiчне представлення, а й на функцiонал: тепер для завантаження файлу можна та треба натиснути на пiдпис до цiєї кнопки.

Тепер розглянемо параметри, якi застосовуються до елементiв з класом «button»:

- background-color – колiр фону, який малюється позаду змiсту елементу,
- border – налаштування обводки, яке складається з таких елементiв:
 - 1px – товщина обводки 1 пiксель,
 - solid – обводка нерозривною лiнiєю,

- gray – сірий колір обводки.
- border-radius – наскільки закругленими будуть кути в прямокутника, який малюється позаду змісту елемента,
- display – властивість типу відображення (у даному випадку «inline-block» - блок усередині тексту),
- padding – значення відступу (перше значення – відступ знизу і зверху, друге значення – відступ зліва і справа),
- cursor – як буде змінюватися графіка курсору при наведенні на елемент (pointer – іконка як при наведенні на посилання),
- font-size – якого розміру буде шрифт,
- font-family – якого сімейства буде шрифт.

У цьому файлі CSS використано назви кольорів. Їхня перевага в тому, що по назвам легше зрозуміти, що це за колір, ніж по коду (хоча деякі коди легко запам'ятати, наприклад, код для білого (#ffffff), червоного (#ff0000) і подібних).

Але у назв кольорів є недоліки. У стандарті CSS зазначено, що стандартизовані кольори не є хорошими: наприклад, стандартизований темно-сірий (darkgray) насправді світліше за стандартизований сірий (gray), а використання деяких назв може бути сприйнятим як образа.[14]

У використаних у цьому файлі кольорів немає проблем: назви сірий (gray) і світло-сірий (lightgray) відповідають реальності, бо стандартизований світло-сірий світліше стандартизованого сірого.

Порівнюючи рис. 3.23 і рис. 3.24, можна побачити, наскільки використання CSS може змінити зовнішній вигляд окремих елементів. А порівнюючи рис. 3.25 і рис. 3.26, можна побачити, наскільки може бути зміненим зовнішній вигляд веб-сторінок загалом.

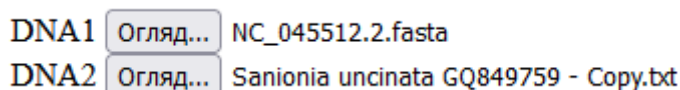


Рис. 3.23 – Кнопки завантаження файлів без CSS.

DNA1
DNA2

Рис. 3.24 – Кнопки завантаження файлів з CSS.

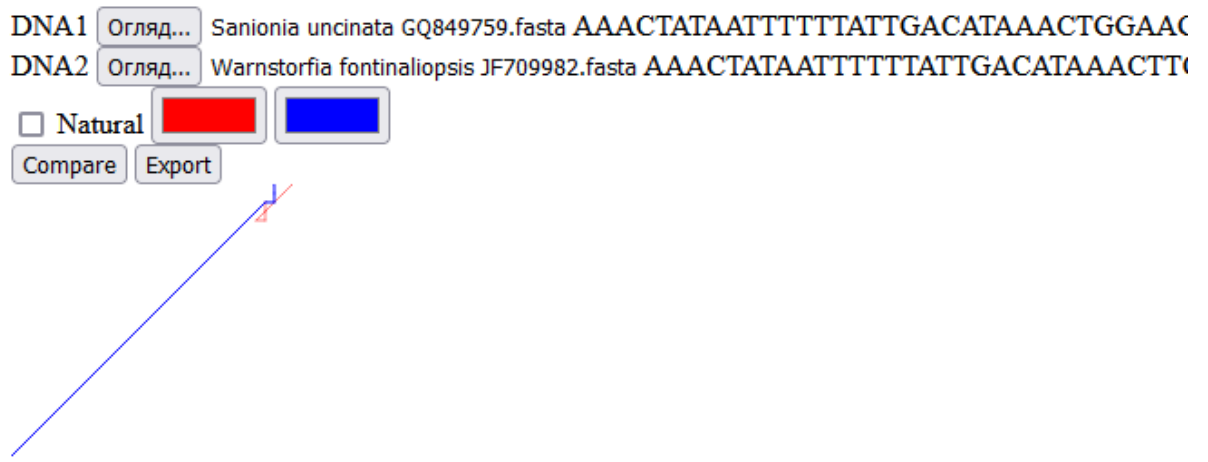


Рис. 3.25 – Зовнішній вигляд веб-сторінки без використання файлу CSS

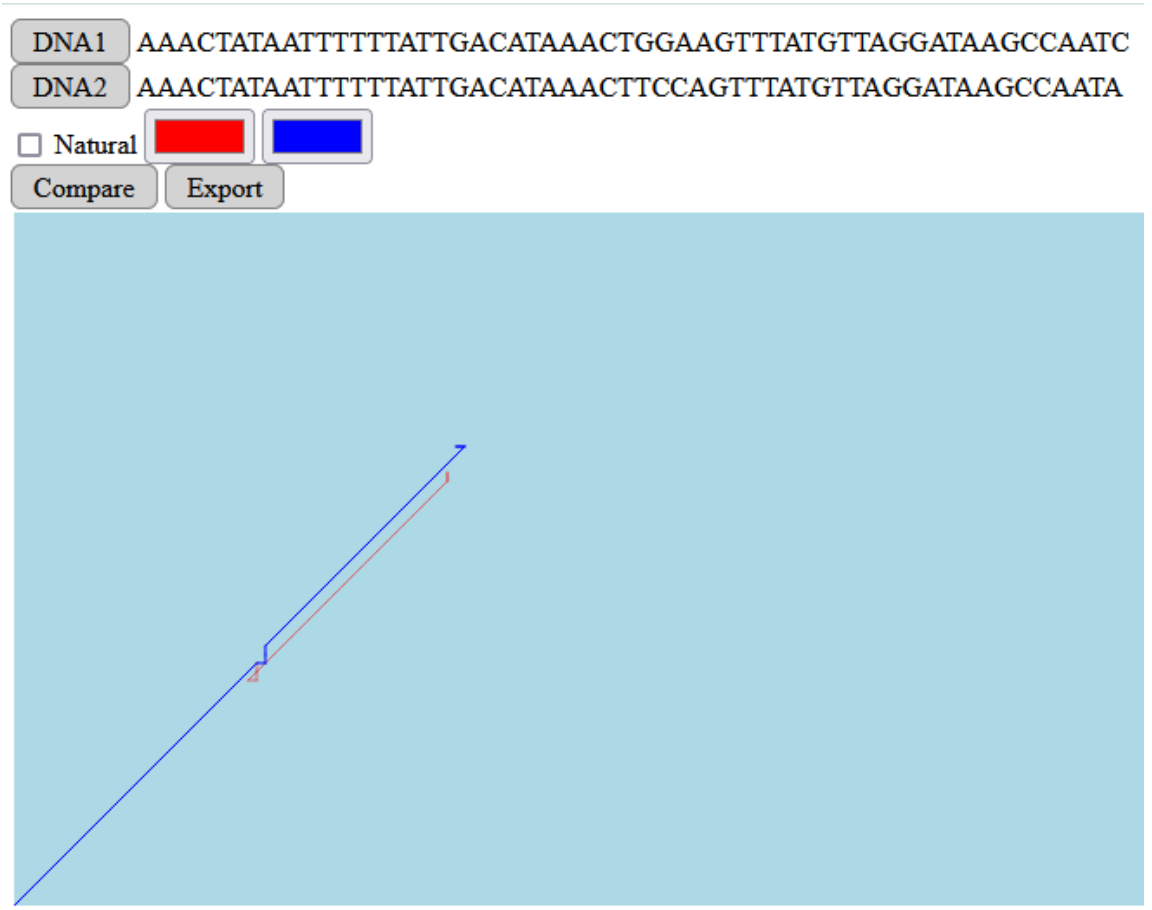


Рис. 3.26 – Зовнішній вигляд веб-сторінки з використанням CSS

Таким чином можна зробити зовнішній вигляд сторінки більш привабливим для користувачів, ніж зовнішній вигляд неоформленої сторінки.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

ВИСНОВОК ДО РОЗДІЛУ 3

У третьому розділі описано процес розробки застосунку для візуального порівняння нуклеотидних послідовностей.

Показано:

- склад HTML-файлу, який буде запущено,
- принцип вибору елементів,
- алгоритм візуалізації та обробки неочікуваних символів,
- імпорт файлів для введення початкових даних,
- алгоритм роботи з форматом FASTA,
- склад файлу CSS і його вплив на сторінку.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

РОЗДІЛ 4. ДЕМОНСТРАЦІЯ РОБОЧИХ ПРОЦЕСІВ

4.1 Інтерфейс

Програма містить:

- полотно для візуалізації,
- кнопки для вибору файлів «Choose File»,
- текст відображення текстового представлення нуклеотидних послідовностей,
- налаштування «Natural» для відрізків різного розміру,
- налаштування кольорів ліній,
- кнопка «Compare» для запуску процесу візуалізації.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

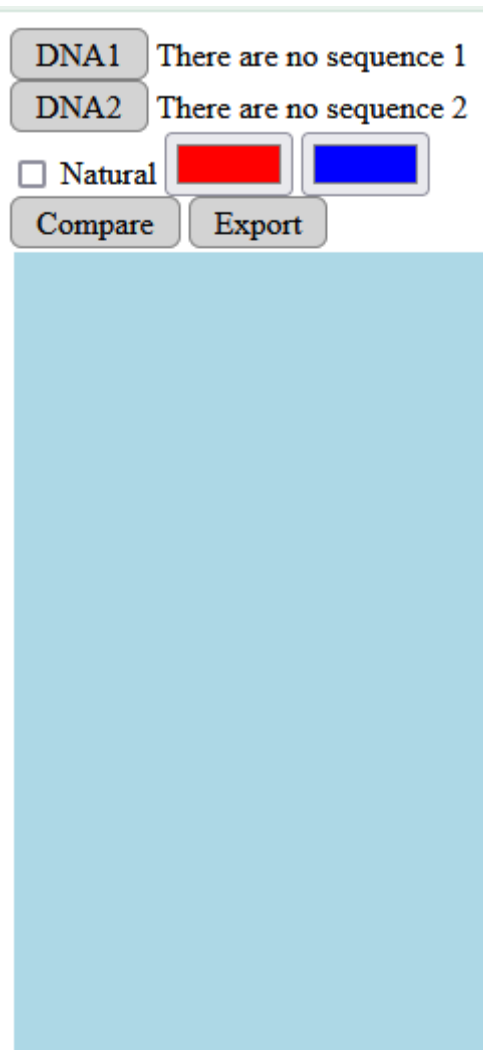


Рис. 4.1 – Застосунок на початку роботи

DNA1 AAACTATAATTTTTATTGACATAAACTGGAAGTTTATGTTAGGATAAGCCAATC
 DNA2 AAACTATAATTTTTATTGACATAAACTTCCAGTTTATGTTAGGATAAGCCAATA

Natural

Compare Export




Рис. 4.2 – Демонстрація робочого процесу

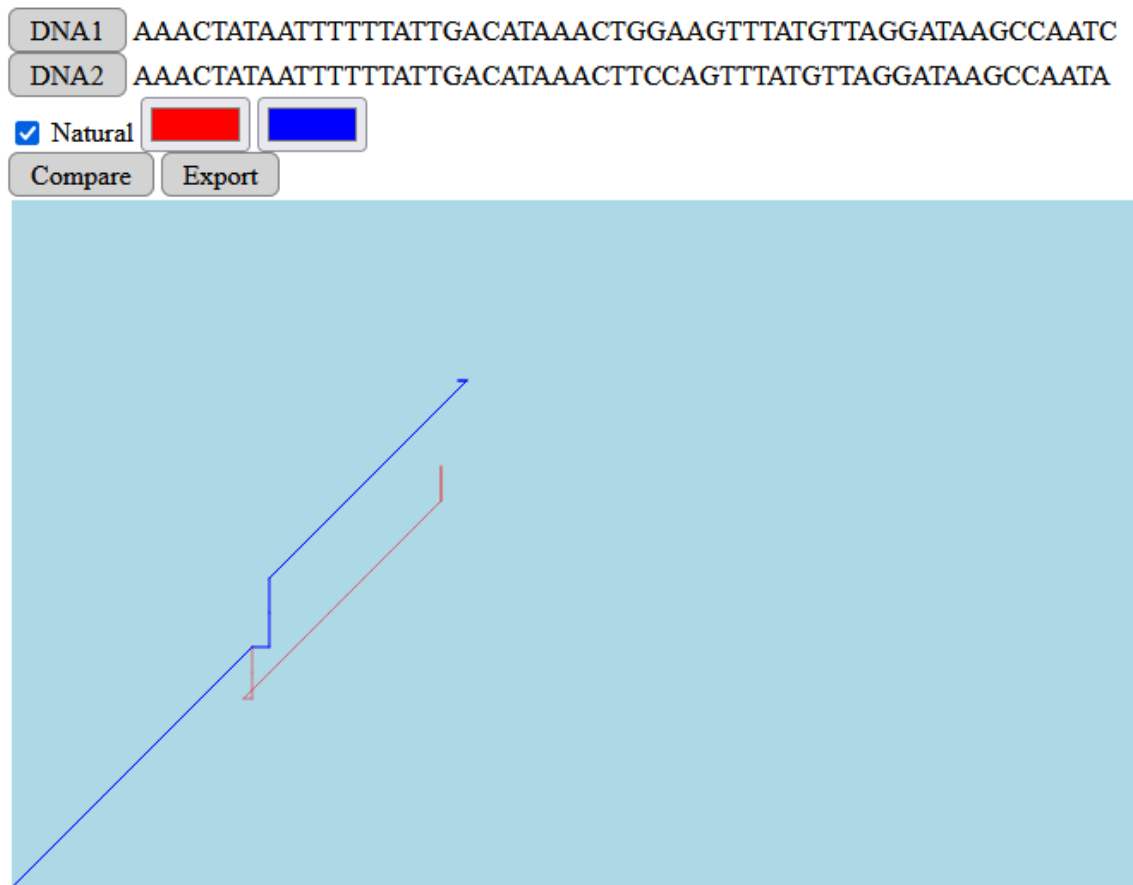


Рис. 4.3 – Демонстрація робочого процесу з параметром «Natural»

Якщо буде завантажено файл з символами, які не підтримуються, сторінка виведе повідомлення про те, що певний символ не може бути оброблено. (рис. 4.2).

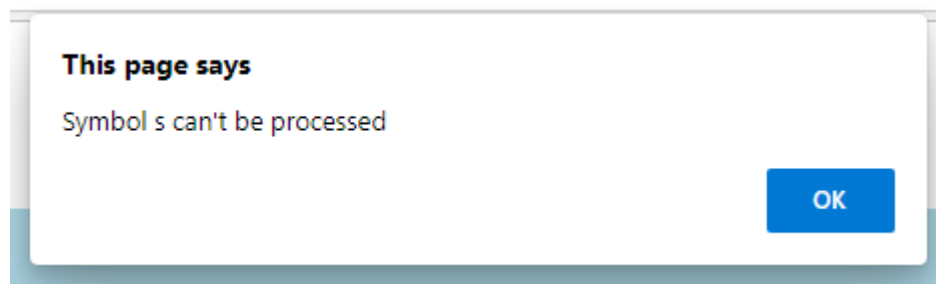


Рис. 4.3 – Демонстрація повідомлення про помилку

4.2 Інструкція роботи користувача з системою

Якщо файли програми завантажено на комп'ютер, щоб почати роботу програми, необхідно відкрити файл index.html у браузері. Якщо програма є частиною сайту, потрібно перейти на сторінку програми. На даний момент онлайн версії немає, але її дозволено додати на свій сайт.

Програма не буде працювати, якщо браузер не підтримує JavaScript або відсутній дозвіл на його використання. Інструкції щодо надання дозволу використання JavaScript залежать від браузера.

Файл має бути текстовим і може бути з таким вмістом:

- лише нуклеотидна послідовність,
- нуклеотидні послідовності з описами до кожного з них у форматі FASTA (але буде використано лише першу нуклеотидну послідовність).

Доступні символи: A, C, G, T, U, -. Символи для позначення позиції, на якій може бути не тільки один варіант нуклеотиду, не підтримуються у цій програмі. Наявність символів, яких немає у вище наведеному списку, може негативно вплинути на коректність результату. Але про такі символи буде попереджено веб-сторінкою, якщо вони виявляться у файлі.

Для кращого результату послідовності, які порівнюються, мають бути вирівняні. Для цього можна використати інші програми, наприклад, MUSCLE, описана в розділі 1.

Для загрузки файлу необхідно натиснути кнопку «Choose File» біля відповідного номеру послідовності.

Перед порівнянням впевніться, що увімкнено чи вимкнено «Natural», у залежності від того, чи потрібні вам відрізки різної довжини для різних нукле-

					ІАЛЦ.467200.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

отидів. Значення довжин при увімкненому налаштуванні «Natural»: С – 4, G – 3, T – 2, A – 1.

Коли файли завантажені, необхідно натиснути кнопку «Compare» для візуалізації результату.

4.3 Перелік використововуваного програмного забезпечення

Програму було розроблено на операційній системі Windows 10.

Як середовище розробки програмного коду була використана програма Visual Studio Code.

Використано мову програмування JavaScript для написання поведінки програми.

Для розмітки сторінки було використано HTML.

Для оформлення сторінки було використано CSS.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

ВИСНОВОК ДО РОЗДІЛУ 4

У четвертому розділі показано готову на даний момент версію застосунку, показано інтерфейс користувача, надано інструкцію з використання, перераховано використане програмне забезпечення.

Ознайомитись детальніше з програмною реалізацією проєкту можна за наступним посиланням:

<https://github.com/WhitePaperChan/VisualDNAComparison>

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

ВИСНОВКИ

Розроблено веб-застосунок для візуального порівняння нуклеотидних послідовностей. Результати візуалізації краще допомагають розпізнати патерни вставлених частин нуклеотидних послідовностей, ніж порівняння вирівняних послідовностей чи порівняння зображень візуалізованих послідовностей.

Застосунок вирішено зробити веб-сторінкою, яка може бути частиною сайти чи працювати без інтернету.

Розроблений веб-застосунок є унікальним, бо на даний момент практично неможливо знайти його повноцінний аналог чи програму для будь-якого візуального порівняння. Найближчим до цього є програма, яка реалізує представлення нуклеотидної послідовності на нотному стані, але вона не скопійована та без редагування коду не може бути ефективно використана для порівняння. Саме існування цього застосунку є його перевагою.

Під час написання дипломної роботи було проаналізовано існуючі методи візуалізації. Це зроблено для того, щоб визначити найбільш підходящий метод візуалізації нуклеотидних послідовностей.

Також було проаналізовано існуючі програми для візуалізації. Досліджувалися переваги та недоліки наявного програмного забезпечення для того, щоб знати, на що варто звернути увагу під час розробки власного застосунку та отримати ідеї для додаткових налаштувань.

Хоча програма схожа на написана раніше цією студенткою програму DNK Visualisation, яка згадується у розділі 1, ця програма в першу чергу вирізняється тим, що змінює візуалізацію заради простішого аналізу подібних нуклеотидних послідовностей.

					ІАЛЦ.467200.003 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

До того ж, цей застосунок, на відміну від вищезгаданого проєкту, підтримує формат FASTA, хоча й використовує лише першу послідовність з файлу такого формату.

Програма не виконує вирівнювання нуклеотидних послідовностей, тому для найбільш коректних результатів необхідно загрузати вже вирівняні послідовності. Вирівнювання нуклеотидних послідовностей є окремою задачею зі своїми труднощами, і вже є достатньо застосунків у вільному доступі, які цю задачу виконують. Тому й було вирішено обмежитися візуалізацією вже вирівняних нуклеотидних послідовностей.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Madeira F, Pearce M, Tivey ARN, Basutkar P, Lee J, Edbali O, Madhusoodanan N, Kolesnikov A, Lopez R. Search and sequence analysis tools services from EMBL-EBI in 2022. *Nucleic Acids Res.* 2022 Apr gkac240. doi:10.1093/nar/gkac240. PMID: 35412617.
2. Hamori E, Ruskin J. H curves, a novel method of representation of nucleotide series especially suited for long DNA sequences. *J Biol Chem.* 1983 Jan 25;258(2):1318-27. PMID: 6822501.
3. Duplij D, Duplij S. DNA sequence representation by trianders and determinative degree of nucleotides. *J Zhejiang Univ Sci B.* 2005;6(8):743-755. doi:10.1631/jzus.2005.B0743
4. Дуплій В.П., Дуплій С.А. Triander – новая программа для визуального анализа нуклеотидных последовательностей // Фактори експериментальної еволюції організмів: Зб. наук. пр. / Національна академія наук України, Інститут молекулярної біології і генетики, Укр. Т-во генетиків і селекціонерів ім. М.І. Вавилова; редкол. / В. А. Кунах (голов. ред.) [та ін.] – К.: Укр. т-во генетиків і селекціонерів ім. М.І. Вавилова, 2015. – Т. 17. – С. 51-54
5. Cowin JE, Jellis CH, Rickwood D. A new method of representing DNA sequences which combines ease of visual analysis with machine readability. *Nucleic Acids Res.* 1986 Jan 10;14(1):509-15. doi: 10.1093/nar/14.1.509. PMID: 3003680; PMCID: PMC339435.
6. Using a JavaScript library versus building the functionality yourself [Електронний ресурс] – Режим доступу до ресурсу: <https://www.codereadability.com/javascript-library-vs-building-yourself/>

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

7. Doctype - MDN Web Docs Glossary: Definitions of Web-related terms: MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/Doctype>
8. joshikashyap399 (n.d.). *Difference between SVG and HTML 5 Canvas* [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/difference-between-svg-and-html-5-canvas/>
9. Document - Интерфейсы веб API: MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/API/Document>
10. JScript (ECMAScript3) [Електронний ресурс] – Режим доступу до ресурсу: [https://docs.microsoft.com/en-us/previous-versions/hbxc2t98\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/hbxc2t98(v=vs.85))
11. VBScript [Електронний ресурс] – Режим доступу до ресурсу: [https://docs.microsoft.com/en-us/previous-versions/t0aew7h6\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/t0aew7h6(v=vs.85))
12. <path> - SVG: Scalable Vector Graphics: MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/SVG/Element/path>
13. Types of CSS: Inline, External and Internal Definitions and Differences Explained [Електронний ресурс] – Режим доступу до ресурсу: https://www.hostinger.com/tutorials/difference-between-inline-external-and-internal-css#The_Difference_Between_Inline_External_and_Internal_CSS_Styles
14. CSS Color Module Level 4 [Електронний ресурс] – Режим доступу до ресурсу: <https://drafts.csswg.org/css-color/#named-colors>
15. Regular expressions - JavaScript: MDN [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

16. How do I save/export an SVG file after creating an SVG with D3.js (IE, safari and chrome)? [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/questions/23218174/how-do-i-save-export-an-svg-file-after-creating-an-svg-with-d3-js-ie-safari-an>

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

ДОДАТОК 1

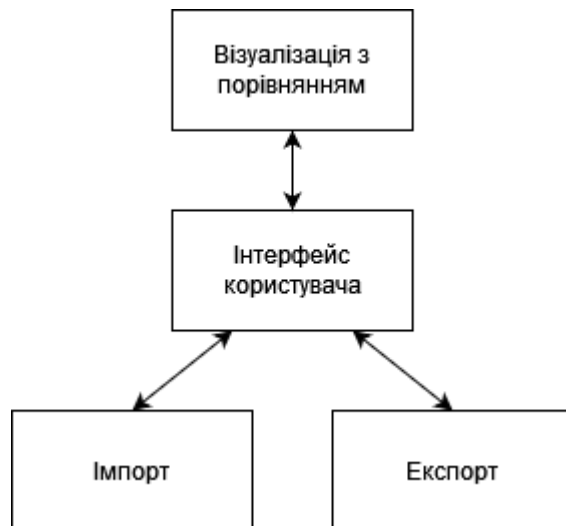
Застосунок візуального порівняння нуклеотидних
послідовностей в біомедичних дослідженнях

Структурна схема системи

ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2022 р



					ІАЛЦ.467200.004 Д1			
		№ докум.	Підпис	Дата	Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях Структурна схема	Літ.	Аркуш	Аркушів
Розробив	Дуплій Н.В.						1	1
Перевірив	Кочура Ю.П.					КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-82		
Н. Контр.	Сімоненко В. П.							
Затвердив								

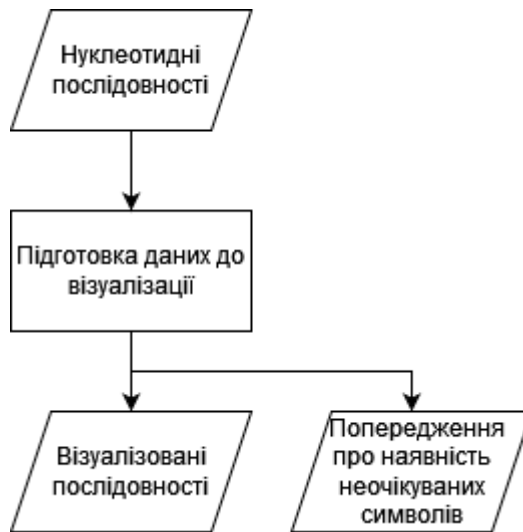
ДОДАТОК 2

Застосунок візуального порівняння нуклеотидних
послідовностей в біомедичних дослідженнях

Функціональна схема
ІАЛЦ.467200.006 Д2

Аркушів 1

Київ 2022 р



					ІАЛЦ.467200.006 Д2		
		№ докум.	Підпис	Дата			
Розробив	Дуплій Н.В.				Літ.	Аркуш	Аркушів
Перевірив	Кочура Ю.П.					1	1
Н. Контр.	Сімоненко В. П.				КПІ ім. Ігоря		
Затвердив					Сікорського, ФІОТ, ІВ-82		
					Застосунок візуального порівняння нуклеотидних послідовностей в біомедичних дослідженнях Функціональна схема (схема даних)		

ДОДАТОК 3

Застосунок візуального порівняння нуклеотидних
послідовностей в біомедичних дослідженнях

Алгоритм дій програмного забезпечення

ІАЛЦ.467200.006 ДЗ

Аркушів 1

Київ 2022 р



					ІАЛЦ.467200.006 ДЗ		
		№ докум.	Підпис	Дата			
Розробив	Дуплій Н.В.				Літ.	Аркуш	Аркушів
Перевірив	Кочура Ю.П.					1	1
Н. Контр.	Сімоненко В. П.				КПІ ім. Ігоря		
Затвердив					Сікорського, ФІОТ, ІВ-82		

Застосунок візуального порівняння
нуклеотидних послідовностей в
біомедичних дослідженнях
Алгоритм дій програм-
ного забезпечення