

Міністерство освіти та науки України  
Національний університет водного господарства та  
природокористування  
Кафедра екології, технології захисту навколишнього середовища та  
лісового господарства

**03-06-169М**

### МЕТОДИЧНІ ВКАЗІВКИ

до виконання практичних робіт з навчальної дисципліни «Основи  
біоінформатики» для здобувачів вищої освіти першого  
(бакалаврського) рівня за освітньою програмою «Біотехнології,  
біоробототехніка та біоенергетика» спеціальності 162  
«Біотехнології та біоінженерія» денної форми навчання

Рекомендовано  
науково-методичною радою з якості  
НП будівництва та архітектури  
Протокол № 5 від 11 лютого 2025 р.

Рівне – 2025

Методичні вказівки до виконання практичних робіт з навчальної дисципліни «Основи біоінформатики» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньою програмою «Біотехнології, біоробототехніка та біоенергетика» спеціальності 162 «Біотехнології та біоінженерія» денної форми навчання [Електронне видання] / Буднік З. М. – Рівне : НУВГП, 2025. – 26 с.

Укладачі: Буднік З. М. – кандидат сільськогосподарських наук, доцент кафедри екології, технології захисту навколишнього середовища та лісового господарства/

Відповідальний за випуск: Мартинов С. Ю., доктор технічних наук, професор, завідувач кафедри водопостачання, водовідведення та бурової справи/

Керівник групи забезпечення освітньо-професійної програми першого (освітньо-професійного) рівня за освітньою програмою «Біотехнології, біоробототехніка та біоенергетика» спеціальності 162 «Біотехнології та біоінженерія» к.т.н., доцент Грицина О. О.

© З. М. Буднік, 2025  
© Національний університет  
водного господарства та  
природокористування, 2025

## ЗМІСТ

ВСТУП.....	3
<i>ПРАКТИЧНА РОБОТА № 1. ВИРІВНЮВАННЯ БІЛКОВИХ ПОСЛІДОВНОСТЕЙ</i> .....	5
<i>ПРАКТИЧНА РОБОТА № 2. ВИВЧЕННЯ ПРОСТОРОВОЇ СТРУКТУРИ МАКРОМОЛЕКУЛ</i> .....	7
<i>ПРАКТИЧНА РОБОТА № 3. АНАЛІЗ БІЛКОВИХ ТА НУКЛЕОТИДНИХ ПОСЛІДОВНОСТЕЙ</i> .....	10
<i>ПРАКТИЧНА РОБОТА №4. ВИРІВНЮВАННЯ АМІНОКИСЛОТНИХ ПОСЛІДОВНОСТЕЙ. АЛГОРИТМ ГЛОБАЛЬНОГО ВИРІВНЮВАННЯ НІДЛМАНА – ВУНША</i> .....	13
<i>ПРАКТИЧНА РОБОТА №5. МОДИФІКОВАНІ АЛГОРИТМИ ПАРНОГО ВИРІВНЮВАННЯ</i> .....	16
<i>ПРАКТИЧНА РОБОТА №6. БАНК ДАНИХ БІЛКОВИХ ПОСЛІДОВНОСТЕЙ UNIPROT</i> .....	20
<i>ПРАКТИЧНА РОБОТА №7. ВИЗНАЧЕННЯ РЕДАКЦІЙНОЇ ВІДСТАНИ МІЖ БІОЛОГІЧНИМИ ПОСЛІДОВНОСТЯМИ. АЛГОРИТМ ВАГНЕРА–ФІШЕРА. ВИРІВНЮВАННЯ БІОЛОГІЧНИХ ПОСЛІДОВНОСТЕЙ З ЛІНІЙНОЮ ПАМ'ЯТТЮ. АЛГОРИТМ МІЛЛЕРА-МАЙЄРСА</i> .....	22
ЛІТЕРАТУРА.....	26

## ВСТУП

Сучасна біотехнологія та біоінженерія неможливі без використання методів біоінформатики, яка поєднує знання біології, інформатики, математики та статистики. Біоінформатика відіграє ключову роль у вивченні структури та функцій біологічних молекул, аналізі геномів, протеомів, метаболомів, а також у моделюванні біологічних процесів.

Методичні вказівки містять теоретичні основи, практичні завдання та методичні рекомендації щодо використання сучасних біоінформатичних інструментів і баз даних.

Метою цих методичних вказівок є формування у студентів базових компетентностей у сфері біоінформатики, зокрема оволодіння методами аналізу біологічних послідовностей, структурного моделювання білків, молекулярного докінгу, а також візуалізації та інтерпретації біологічних даних. Виконання практичних робіт сприятиме розвитку аналітичного мислення, навичок роботи з біоінформатичним програмним забезпеченням і застосуванню отриманих знань у наукових дослідженнях та інженерних розробках.

Методичні вказівки також спрямовані на підвищення рівня самостійної роботи студентів, що є необхідною умовою для успішного оволодіння сучасними технологіями біоінформатики та їх подальшого застосування у професійній діяльності.

Практичні завдання побудовані таким чином, щоб сприяти поступовому засвоєнню матеріалу, починаючи з основ роботи з біоінформатичними базами даних та онлайн-інструментами, і завершуючи застосуванням спеціалізованого програмного забезпечення для моделювання біологічних систем. Значна увага приділяється навчанню студентів інтерпретації отриманих результатів та їх використанню для розв'язання прикладних задач у сфері біотехнологій, біоробототехніки та біоенергетики.

Запропоновані методичні матеріали сприятимуть формуванню у студентів системного підходу до аналізу біологічної інформації, розвитку навичок критичного мислення та використання сучасних біоінформатичних технологій у науковій та практичній діяльності. Виконання практичних робіт дозволить здобувачам вищої освіти підготуватися до роботи у сферах молекулярної біології, генетичної інженерії, фармацевтики, біомедицини та суміжних галузях, що активно використовують методи біоінформатики.

Таким чином, оволодіння навичками біоінформатичного аналізу є необхідною складовою професійної підготовки сучасного біотехнолога та біоінженера, що сприятиме їх конкурентоспроможності на ринку праці та успішному вирішенню наукових і прикладних завдань у майбутній професійній діяльності.

## **ПРАКТИЧНА РОБОТА № 1**

### **ВИРІВНЮВАННЯ БІЛКОВИХ ПОСЛІДОВНОСТЕЙ**

Мета роботи: ознайомлення зі способами вирівнювання амінокислотних послідовностей білків, основними алгоритмами та програмними засобами для їх аналізу. Формування навичок використання біоінформатичних інструментів для порівняння білкових послідовностей, оцінки рівня гомології та побудови філогенетичних зв'язків між білками.

#### **Теоретичні відомості**

##### **1. Вирівнювання послідовностей**

Вирівнювання білкових послідовностей – це процес знаходження подібностей між амінокислотними послідовностями шляхом встановлення оптимального розташування їхніх залишків. Це дозволяє виявляти гомологію між білками, прогнозувати їх функції та аналізувати еволюційні зв'язки.

Основними видами вирівнювання є:

- **Парне вирівнювання (Pairwise alignment)** – порівняння двох білкових послідовностей.
- **Множинне вирівнювання (Multiple sequence alignment, MSA)** – вирівнювання більше ніж двох послідовностей для виявлення консервативних регіонів.

##### **2. Основні алгоритми вирівнювання**

- **Глобальне вирівнювання (Needleman-Wunsch Algorithm)** – використовується для вирівнювання повної довжини двох послідовностей.
- **Локальне вирівнювання (Smith-Waterman Algorithm)** – застосовується для знаходження найбільш подібних фрагментів у послідовностях.
- **E-value (очікуване значення)** – статистична міра достовірності вирівнювання.

##### **3. Інструменти для вирівнювання білкових послідовностей**

- **BLASTp (Basic Local Alignment Search Tool for Proteins)** – інструмент для локального вирівнювання білкових послідовностей, що дозволяє знаходити схожі білки в базах даних.
- **Clustal Omega** – метод для множинного вирівнювання білкових послідовностей.
- **MUSCLE (Multiple Sequence Comparison by Log-Expectation)** – високоточний алгоритм для MSA.

- **T-Coffee** – точний інструмент для множинного вирівнювання.

### 3. Матеріали та програмне забезпечення

- Комп'ютер з доступом до Інтернету
- Веб-інструменти: **NCBI BLAST**, Clustal Omega
- Програмні пакети: MEGA, Jalview, BioEdit (за необхідності)

З розвитком методів біохімії та молекулярної біології, а також обчислювальної техніки, в біологічній науці сформувався окремий підхід до вивчення живих систем – *in silico*. Даний підхід передбачає дослідження біологічних систем із застосуванням обчислювальних машин, тобто розглядає живі системи як набір інформації про них. Однією з біологічних наук, що реалізують підхід *in silico*, є біоінформатика.

Предметом біоінформатики є алгоритми обробки даних про структуру біологічних макромолекул. Об'єкт біоінформатики – нуклеїнові кислоти і білки, біологічні макромолекули, структура яких принципово визначає властивості живих організмів і їх частин. З розвитком технологій секвенування, що дозволяють отримувати інформацію про первинну структуру біологічних макромолекул, і методів структурного аналізу, які досліджують їх просторову будову, виникла необхідність в обробці великої кількості відповідних даних. У сучасній біоінформатиці є значна кількість спеціальних методів роботи з біологічної інформацією; в той же час безперервно йде пошук нових методів, підходів і алгоритмів математичного (модельного) аналізу біомолекул.

### Хід роботи

#### Завдання 1. Парне вирівнювання за допомогою **BLASTp**

1. Перейдіть на сайт **NCBI BLAST**.
2. Оберіть **BLASTp** (вирівнювання білкових послідовностей).
3. Введіть або вставте амінокислотну послідовність білка (наприклад, з UniProt або власного дослідження).
4. Виберіть базу даних (наприклад, Swiss-Prot або nr).
5. Запустіть аналіз, дочекайтеся результатів.
6. Проаналізуйте отримані збіги, зверніть увагу на параметри **E-value**, ідентичність (% identity) та покриття послідовності (query coverage).
7. Збережіть результати вирівнювання.

## **Завдання 2. Множинне вирівнювання білкових послідовностей (Clustal Omega)**

1. Перейдіть на сайт Clustal Omega.
2. Вставте кілька білкових послідовностей (отриманих у BLAST або з бази UniProt).
3. Запустіть аналіз, дочекайтеся результатів.
4. Проаналізуйте консервативні регіони, зверніть увагу на місця мутацій.
5. Завантажте отримане вирівнювання у форматі FASTA або Clustal.

## **Завдання 3. Візуалізація та аналіз отриманих вирівнювань**

1. Відкрийте отримане множинне вирівнювання у Jalview або іншому редакторі.
2. Визначте найбільш консервативні ділянки білків.
3. Побудуйте філогенетичне дерево (у MEGA або іншій програмі).
4. Зробіть висновки щодо спорідненості білкових послідовностей.

### **Контрольні запитання**

1. Чим відрізняється глобальне та локальне вирівнювання?
2. Який алгоритм використовує BLASTp?
3. Для чого використовують множинне вирівнювання білкових послідовностей?
4. Як оцінюється якість вирівнювання білкових послідовностей?
5. Яке значення має E-value у результатах BLAST?

### **ПРАКТИЧНА РОБОТА № 2**

#### **ВИВЧЕННЯ ПРОСТОРОВОЇ СТРУКТУРИ МАКРОМОЛЕКУЛ**

*Мета роботи* Ознайомитися з просторовою структурою макромолекул (білків, нуклеїнових кислот, полісахаридів). Вивчити рівні структурної організації макромолекул. Навчитися використовувати спеціальні програми та бази даних для аналізу структури макромолекул.

### Теоретичні відомості:

Макромолекули – це великі органічні молекули, що складаються з повторюваних структурних одиниць (мономерів). До макромолекул належать:

- **Білки** (складаються з амінокислот, мають первинну, вторинну, третинну та четвертинну структури).
- **Нуклеїнові кислоти (ДНК, РНК)** (полімери з нуклеотидів, що утворюють спіралі).
- **Полісахариди** (вуглеводи, наприклад, крохмаль, целюлоза).

Для аналізу просторової структури використовують методи рентгеноструктурного аналізу, криоелектронної мікроскопії, спектроскопії.

У сучасній хімії існує надзвичайно велика кількість форматів опису просторової структури (конформації) молекул. Проте для зберігання інформації про структуру біологічних молекул в переважній більшості випадків використовується формат PDB (рис. 1). Він передбачає використання текстових файлів (з розширенням «pdb»), розмічених особливим чином. Як правило, більша частина PDB-файлу – а саме рядки, що починаються зі слів «ATOM» і «HETATM» – містить інформацію про позиції атомів в системі трьох координат. Рядки «ATOM» використовуються для опису атомів макромолекул – білків і нуклеїнових кислот; «HETATM» – це атоми інших з'єднань, зазвичай низькомолекулярних, що знаходяться в комплексі з макромолекулою або просто присутніх в складі структури (як молекули води).

ATOM	4893	O	ALA	B	360	49.951	41.917	16.965	1.00	9.36	O
ATOM	4894	C	SER	B	361	49.279	43.271	20.854	1.00	11.33	C
ATOM	4895	CA	SER	B	361	49.327	43.061	19.367	1.00	11.24	C
ATOM	4896	CB	SER	B	361	46.148	43.776	18.700	1.00	13.43	C
ATOM	4897	H	SER	B	361	49.313	41.440	19.097	1.00	10.14	H
ATOM	4898	O	SER	B	361	48.840	42.398	21.604	1.00	11.00	O
ATOM	4899	OG	SER	B	361	46.909	43.243	19.118	1.00	16.54	O
ATOM	4900	C	CYS	B	362	48.695	45.712	22.906	1.00	14.14	C
ATOM	4901	CA	CYS	B	362	49.797	44.791	22.660	1.00	12.69	C
ATOM	4902	CB	CYS	B	362	51.123	45.451	23.031	1.00	11.63	C
ATOM	4903	H	CYS	B	362	49.805	44.414	21.267	1.00	11.41	H
ATOM	4904	O	CYS	B	362	47.830	45.963	21.954	1.00	15.40	O
ATOM	4905	SHT	CYS	B	362	48.417	46.113	24.065	1.00	16.83	O
ATOM	4906	SG	CYS	B	362	52.450	44.205	23.117	1.00	9.98	S
TER											
HETATM	4907	C1	H2O	1	3	22.652	22.504	7.778	1.00	10.60	C
HETATM	4908	C2	H2O	1	3	22.211	21.391	8.711	1.00	11.16	C
HETATM	4909	C3	H2O	1	3	23.236	20.270	8.687	1.00	13.07	C
HETATM	4910	O4	H2O	1	3	23.586	19.850	7.247	1.00	12.90	O
HETATM	4911	O5	H2O	1	3	23.882	21.062	6.362	1.00	13.35	O

Рис. 1. Атом № 4899 є атомом кисню (див. символ у кінці рядка) в гамма-позиції серину - 361-го амінокислотного залишку В-ланцюга білка. Цей атом знаходиться в точці з координатами по осі X - 47 Å, Y - 43 Å, Z - 19 Å



Для роботи з просторовими структурами біомолекул використовуються спеціальні програми-переглядачі, або «в'юери» (viewer), що дозволяють представити молекулу у вигляді моделей різного вигляду (рис. 2). Також зазвичай присутній стереоскопічний режим, який полегшує сприйняття конформації молекули (але вимагає деякого тренування очей).

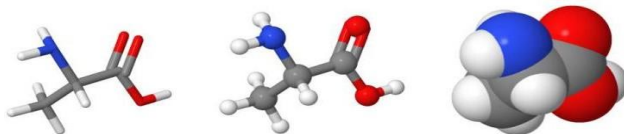


Рис. 2. Різні моделі молекули L-аланіну: а – каркасна (скелетна або дротяна – stick), б – кулько-стрижнева (ball and stick), в – модель Ван-дер-Ваальсових радіусів атомів (СРК)

Центральним сховищем даних про структуру біологічних макромолекул є Protein Data Bank, доступний за адресою: <https://www.rcsb.org>

Крім того, отримати доступ до цих даних і здійснювати пошук за ними можна на знайомому вам сайті NCBI (база даних Structure).

Унікальним ідентифікатором кожної структури в Protein Data Bank є PDB ID. Даний ідентифікатор складається з чотирьох символів, перший з яких, як правило, цифра; літери в PDB ID прийнято писати великими, хоча більшість програм, які з ними працюють, не чутливі до регістру символів. Приклади PDB ID: 4L9K, 1BTI, 2R33.

### ***Обладнання та матеріали:***

1. Комп'ютер з доступом до Інтернету.
2. Програмне забезпечення (PyMOL, RasMol, Chimera або інші).
3. База даних PDB (Protein Data Bank) – <https://www.rcsb.org>.

### ***Хід роботи:***

#### **1. Вибір макромолекули для дослідження**

1. Перейдіть на сайт **Protein Data Bank (PDB)**.
2. Введіть у пошук назву білка або іншої макромолекули (наприклад, гемоглобін – "Hemoglobin").
3. Виберіть один із запропонованих варіантів та завантажте файл **.pdb**.

## **2. Візуалізація структури макромолекули**

1. Відкрийте програму **PyMOL/RasMol/Chimera**.
2. Завантажте файл **.pdb**.
3. Ознайомтеся з первинною, вторинною, третинною структурою.
4. Використайте різні режими відображення (каркасна модель, стрічкова модель, поверхнева модель).

## **3. Аналіз структурних особливостей**

1. Визначте вторинні структури: альфа-спіралі, бета-листи.
2. Знайдіть активні центри (для ферментів).
3. Дослідіть взаємодію макромолекули з лігандами (якщо є).

## **4. Опис отриманих результатів**

1. Визначте основні особливості будови макромолекули.
2. Оцініть її функціональне значення у біологічних процесах.
3. Зробіть висновки про роль просторової організації в роботі макромолекули.

### ***Контрольні запитання:***

1. Які рівні структурної організації білка ви знаєте?
2. Як утворюється альфа-спіраль та бета-структура?
3. Чому важлива просторова структура макромолекул?
4. Які методи використовуються для вивчення структури макромолекул?
5. Як змінюється функція білка при порушенні його структури?

## **ПРАКТИЧНА РОБОТА № 3 АНАЛІЗ БІЛКОВИХ ТА НУКЛЕОТИДНИХ ПОСЛІДОВНОСТЕЙ**

*Мета роботи* Ознайомитися з методами аналізу білкових та нуклеотидних послідовностей. Навчитися використовувати біоінформатичні онлайн-інструменти та бази даних. Визначити гомологію послідовностей, знайти білкові домени та виконати вирівнювання послідовностей.

### ***Теоретичні відомості:***

Аналіз послідовностей є важливим етапом у вивченні геноміки та протеоміки. Основні методи включають:

- Вирівнювання послідовностей (**BLAST, Clustal Omega**).

- Визначення відкритих рамок зчитування (ORF).
- Аналіз доменної структури білків (**Pfam, InterPro**).
- Філогенетичний аналіз.

#### Бази даних для аналізу:

- NCBI (National Center for Biotechnology Information) – <https://www.ncbi.nlm.nih.gov>
  - UniProt (Universal Protein Resource) – <https://www.uniprot.org>
  - Pfam (Protein Families Database) – <https://pfam.xfam.org>
- Знайдіть за допомогою ресурсу NCBI структуру 1V2W (рис. 1).



Рис. 1. Пошук структури на сервісі NCBI

Відкрийте перший результат пошуку. Одразу завантажить PDB-файл цієї структури – кнопка Download (рис. 2) знаходиться у правій верхній частині сторінки (цей файл знадобиться нам пізніше).

#### in Variant X(Ssai)bt.B4

**PDB ID:** 1V2W [Download](#)

**MMDB ID:** 28039

**PDB Deposition Date:** 2003/10/17

Рис. 2. Посилання на PDB-файл структури

Розгляньте відкриту сторінку з об'ємною будовою цього молекулярного комплексу. **Що він собою представляє? Опишіть, що входить до його складу.**

Відкрийте білкову послідовність. Для цього натисніть «1\_protein» трохи нижче на сторінці (рис. 3).

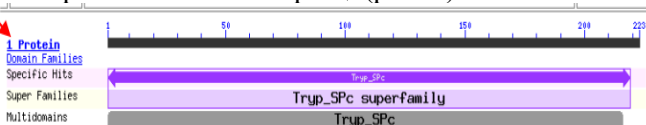


Рис. 3. Перехід до перегляду білкової послідовності

Прогляньте сторінку і перейдіть у розділ Graphics (рис. 4).

## Chain T, Trypsin Inhibitor

PDB: 1V2W\_T

[GenPept](#) [Identical Proteins](#) [Graphics](#)

>gi|49259469|pdb|1V2W|T Chain T,

Рис. 4. Розділ розширених даних щодо просторової структури білка

Дослідіть представлену на сторінці схему і способи роботи з нею, знайдіть елементи керування масштабуванням і прокручуванням (рис. 5). У верхній частині вказані номери амінокислотних залишків. Збільшуючи масштаб, можна досягти буквених позначень амінокислот.

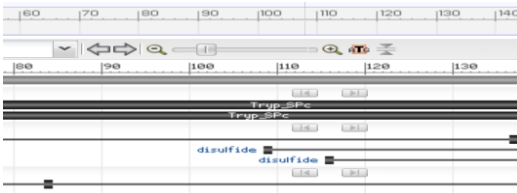


Рис. 5. Фрагмент розділу Graphics з елементами керування, номерами амінокислотних залишків та дисульфідними зв'язками

**Обладнання та матеріали:** Комп'ютер із доступом до Інтернету. Веб-браузер для роботи з онлайн-базами даних. Інструменти BLAST, Clustal Omega, ORF Finder, Pfam.

### *Хід роботи:*

#### **1. Отримання нуклеотидної або білкової послідовності**

1. Перейдіть на сайт **NCBI GenBank** або **UniProt**.
2. Виберіть цікавий вам ген або білок (наприклад, гемоглобін або GFP).
3. Скопіюйте його послідовність у форматі FASTA.

#### **2. Аналіз гомології за допомогою BLAST**

4. Відкрийте NCBI BLAST.
5. Вставте отриману послідовність у поле запиту.
6. Виберіть відповідний алгоритм:
7. **BLASTn** – для порівняння нуклеотидних послідовностей.
8. **BLASTp** – для порівняння білкових послідовностей.
9. Запустіть аналіз і проаналізуйте результати (ідентичність, e-значення, подібність).

### **3. Вирівнювання послідовностей за допомогою Clustal Omega**

1. Перейдіть на Clustal Omega.
2. Вставте кілька схожих білкових або нуклеотидних послідовностей.
3. Запустіть вирівнювання та проаналізуйте консервативні області.

#### **4. Визначення білкових доменів (Pfam, InterPro)**

1. Перейдіть на Pfam або InterPro.
2. Вставте білкову послідовність у поле пошуку.
3. Визначте, до яких родин або функціональних доменів належить білок.

#### **5. Визначення відкритих рамок зчитування (ORF Finder)**

1. Перейдіть на ORF Finder.
2. Вставте нуклеотидну послідовність.
3. Визначте можливі відкриті рамки зчитування та передбачувані білки.

#### ***Контрольні запитання:***

1. Яка роль вирівнювання послідовностей у біоінформатиці?
2. Що таке гомологія послідовностей?
3. Як працює BLAST і що означає e-value?
4. Які бази даних використовуються для аналізу білків?
5. Як можна визначити відкриті рамки зчитування?

### ***ПРАКТИЧНА РОБОТА №4***

#### ***ВИРІВНЮВАННЯ АМІНОКИСЛОТНИХ ПОСЛІДОВНОСТЕЙ. АЛГОРИТМ ГЛОБАЛЬНОГО ВИРІВНЮВАННЯ НІДЛМАНА – ВУНША***

***Мета роботи:*** Ознайомитися з принципами вирівнювання амінокислотних послідовностей. Вивчити алгоритм глобального вирівнювання Нідлмана – Вунша. Навчитися використовувати біоінформатичні інструменти для вирівнювання послідовностей.

#### ***Теоретичні відомості:***

##### **1. Вирівнювання послідовностей**

Вирівнювання амінокислотних або нуклеотидних послідовностей використовується для порівняння подібності між біомолекулами. Існує два основні типи вирівнювання:

• **Глобальне вирівнювання** – використовується для послідовностей приблизно однакової довжини (алгоритм Нідлмана – Вунша).

• **Локальне вирівнювання** – використовується для знаходження схожих фрагментів у довгих послідовностях (алгоритм Сміта – Ватермана).

## 2. Алгоритм Нідлмана – Вунша

Цей алгоритм використовує метод динамічного програмування для вирівнювання двох послідовностей по всій їх довжині. Основні етапи:

1. Створення матриці розміром  $(n+1) \times (m+1)$ , де  $n$  та  $m$  – довжини послідовностей.

2. Заповнення першого рядка та першого стовпця штрафами за пропуски.

3. Заповнення матриці відповідно до функції оцінки:

$$S(i, j) = \max \begin{cases} S(i-1, j-1) + \text{match/mismatch}, \\ S(i-1, j) + \text{gap}, \\ S(i, j-1) + \text{gap} \end{cases}$$

4. Відстеження шляху для побудови вирівняної послідовності.

## 3. Використання матриць заміни

• **PAM, BLOSUM** – використовуються для визначення подібності між амінокислотами.

• Наприклад, **BLOSUM62** широко застосовується для білкових послідовностей.

Алгоритм глобального вирівнювання двох біологічних послідовностей методом динамічного програмування був вперше запропонований Солом Нідлманом та Крістіаном Вуншем (Saul B. Needleman Christian D. Wunsch). Для ідентифікації локального збігу подібний алгоритм був вперше використаний Темплом Смітом та Міхаелем Уотерманом (Temple F. Smith i Michael S. Waterman).

**Обладнання та матеріали:** Комп'ютер із доступом до Інтернету. Онлайн-інструменти для вирівнювання послідовностей: EMBOSS Needle ([https://www.ebi.ac.uk/Tools/psa/emboss\\_needle/](https://www.ebi.ac.uk/Tools/psa/emboss_needle/)), Clustal Omega (<https://www.ebi.ac.uk/Tools/msa/clustalo/>), Ручний розрахунок вирівнювання для простої послідовності.

### *Хід роботи:*

#### **1. Вибір амінокислотних послідовностей**

1. Перейдіть на UniProt.
2. Виберіть дві білкові послідовності для порівняння (наприклад, гемоглобін людини та миші).
3. Скопіюйте послідовності у форматі **FASTA**.

#### **2. Автоматичне вирівнювання за допомогою EMBOSS Needle**

1. Відкрийте EMBOSS Needle.
2. Вставте дві амінокислотні послідовності.
3. Оберіть **BLOSUM62** як матрицю заміни та встановіть штрафи за пропуски (наприклад, -10 для відкриття, -0.5 для розширення).
4. Запустіть аналіз і перегляньте вирівняні послідовності.

#### **3. Ручне вирівнювання за алгоритмом Нідлмана – Вунша**

Виконайте вирівнювання двох коротких амінокислотних послідовностей вручну.

#### **Приклад:**

Дано дві послідовності:

• **Seq1:** A C G T

• **Seq2:** A G G T

Кроки:

1. **Заповнення матриці (початкове значення - штраф за пропуски -2):**

	-	A	C	G	T
-	0	-2	-4	-6	-8
A	-2	<b>2</b>	0	-2	-4
G	-4	0	1	<b>2</b>	0
G	-6	-2	-1	<b>3</b>	<b>1</b>
T	-8	-4	-3	1	<b>4</b>

2. **Відстеження шляху для побудови вирівняної послідовності:**

**Seq1:** A – C – G - T

**Seq2:** A G - G T

#### **4. Аналіз результатів**

1. Оцініть схожість послідовностей за отриманими результатами.

2. Порівняйте автоматичне вирівнювання (EMBOSS Needle) та ручне обчислення.

#### ***Контрольні запитання:***

1. У чому різниця між глобальним і локальним вирівнюванням?

2. Як працює алгоритм Нідлмана – Вунша?

3. Яку роль відіграють матриці заміни (наприклад, BLOSUM62)?

4. Як змінюється вирівнювання при збільшенні штрафу за пропуски?

5. Де застосовується вирівнювання послідовностей у біоінформатиці?

### ***ПРАКТИЧНА РОБОТА №5*** ***МОДИФІКОВАНІ АЛГОРИТМИ ПАРНОГО*** ***ВИРІВНЮВАННЯ***

***Мета роботи:*** *Ознайомитися з модифікованими алгоритмами парного вирівнювання. Вивчити вдосконалені методи глобального та локального вирівнювання. Навчитися використовувати біоінформатичні інструменти для аналізу послідовностей.*

#### ***Теоретичні відомості:***

##### **1. Парне вирівнювання послідовностей**

Парне вирівнювання послідовностей використовується для порівняння двох біологічних послідовностей (нуклеотидних або білкових). Його основна мета — визначити схожість та відмінності між ними.

Існують два основних підходи до вирівнювання:

• **Глобальне вирівнювання** – використовується для порівняння повних послідовностей. (Алгоритм Нідлмана – Вунша)



- **Локальне вирівнювання** – використовується для знаходження схожих фрагментів у довгих послідовностях. (Алгоритм Сміта – Ватермана)

Модифіковані алгоритми парного вирівнювання покращують точність та швидкість аналізу.

## 2. Модифіковані алгоритми парного вирівнювання

2.1. Глобальне вирівнювання з покращеними штрафами за пропуски (Gotoh, Needleman-Wunsch з афінними штрафами)

- У стандартному алгоритмі Нідлмана – Вунша використовується фіксований штраф за пропуски.

- Gotoh-алгоритм додає афінні штрафи:

$$S(i, j) = \max \begin{cases} S(i-1, j-1) + \text{match/mismatch} \\ E(i, j) \\ F(i, j) \end{cases}$$

Де:

- $E(i, j) = S(i, j-1) - d$  (штраф за відкриття пропуску  $d$ )
- $F(i, j) = S(i-1, j) - e$  (штраф за продовження пропуску  $e$ )

**Перевага:** забезпечує більш реалістичне вирівнювання, оскільки розрізняє відкриття та продовження пропуску.

## 2.2. Локальне вирівнювання з оптимізацією (Smith-Waterman з SIMD та BLAST)

- **Стандартний алгоритм Сміта – Ватермана** використовується для локального вирівнювання.

- **Оптимізація через SIMD (Single Instruction, Multiple Data)** дозволяє виконувати багато обчислень паралельно, пришвидшуючи вирівнювання.

- **BLAST (Basic Local Alignment Search Tool)** використовує евристичний підхід:

- Спочатку знаходяться короткі схожі фрагменти.
- Потім розширюються лише найперспективніші вирівнювання.
- Це значно зменшує час обчислень у порівнянні з класичним алгоритмом.

**Перевага:** швидке та ефективне локальне вирівнювання для великих баз даних.

### 2.3. Прогресивне вирівнювання (Clustal Omega, T-Coffee)

- **Clustal Omega** використовує покращені методи для обліку еволюційної близькості послідовностей.

- **T-Coffee** комбінує інформацію з різних алгоритмів для отримання точнішого вирівнювання.

**Перевага:** точніше вирівнювання для еволюційного аналізу.

Якщо біологічні послідовності, для яких необхідно отримати оптимальне локальне вирівнювання, мають велику довжину, можливо, існує велика кількість локальних вирівнювань, що мають значну вагу. В такій ситуації для об'єктивної оцінки результатів вирівнювання, необхідно враховувати такі локальні збіги. Прикладом може бути білок, що містить велику кількість копій одного домену або мотиву. Модифікований алгоритм локального вирівнювання дозволяє знаходити такі повтори. Цей метод знаходить одну або більше копій фрагментів однієї послідовності в іншій.

Нехай – це послідовність, що містить домен або мотив, а - послідовність, в якій ми шукаємо копії ділянок. У фінальному вирівнюванні послідовність буде розділена на ділянки, що вирівняні з фрагментами послідовності. Між вирівняними фрагментами послідовності можуть бути присутні розриви (не вирівняні ділянки).

Схематично такі ділянки позначають крапкою. Під вагою закінченої ділянки вирівнювання ми будемо розуміти її вагу, що може бути обчислена за допомогою базового алгоритму (локального вирівнювання) мінус граничне значення T.

**Обладнання та матеріали:** Комп'ютер із доступом до Інтернету. Онлайн-інструменти для вирівнювання послідовностей: EMBOSS Needle ([https://www.ebi.ac.uk/Tools/psa/emboss\\_needle/](https://www.ebi.ac.uk/Tools/psa/emboss_needle/)) – глобальне вирівнювання. BLAST (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>) – локальне вирівнювання. Clustal Omega (<https://www.ebi.ac.uk/Tools/msa/clustalo/>) – прогресивне вирівнювання.

### *Хід роботи:*

#### **1. Отримання послідовностей**

1. Перейдіть на сайт **UniProt** (<https://www.uniprot.org>).
2. Виберіть білкові послідовності двох організмів (наприклад, гемоглобін людини та миші).
3. Скопіюйте їх у форматі **FASTA**.

#### **2. Вирівнювання за алгоритмом Gotoh (глобальне вирівнювання)**

1. Відкрийте **EMBOSS Needle** ([https://www.ebi.ac.uk/Tools/psa/emboss\\_needle/](https://www.ebi.ac.uk/Tools/psa/emboss_needle/)).
2. Вставте дві білкові послідовності.
3. Оберіть матрицю заміни **BLOSUM62**.
4. Встановіть афінні штрафи (наприклад, -10 за відкриття пропуску, -0.5 за продовження).
5. Запустіть аналіз і перегляньте результати вирівнювання.

#### **3. Використання BLAST (локальне вирівнювання)**

1. Перейдіть на сайт **BLASTp** (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>).
2. Вставте білкову послідовність.
3. Оберіть організм для порівняння.
4. Запустіть аналіз і перегляньте схожі послідовності.

#### **4. Використання Clustal Omega (прогресивне вирівнювання)**

1. Перейдіть на сайт **Clustal Omega** (<https://www.ebi.ac.uk/Tools/msa/clustalo/>).
2. Вставте кілька білкових послідовностей.
3. Запустіть аналіз і перегляньте консервативні області.

### *Контрольні запитання:*

1. Чим модифікований алгоритм Gotoh відрізняється від стандартного алгоритму Нідлмана – Вунша?
2. Як BLAST прищвидшує локальне вирівнювання?
3. Чому використання афінних штрафів покращує глобальне вирівнювання?
4. Де застосовують прогресивне вирівнювання?
5. Як SIMD оптимізує алгоритм Сміта – Ватермана?

**ПРАКТИЧНА РОБОТА №6**  
**БАНК ДАНИХ БІЛКОВИХ ПОСЛІДОВНОСТЕЙ UNIPROT**

*Мета роботи: ознайомитися з базою даних UniProt та її функціональними можливостями. Навчитися шукати білкові послідовності та аналізувати їх характеристики. Використовувати UniProt для отримання інформації про білкові домени, функції та еволюційну спорідненість білків.*

**Теоретичні відомості:**

**1. Що таке UniProt?**

*UniProt (Universal Protein Resource)* – це найбільша у світі база даних білкових послідовностей та їх анотацій. Вона містить інформацію про білкову структуру, функцію, взаємодії та еволюцію.

UniProt складається з таких основних розділів:

- **UniProtKB (Knowledgebase)** – містить експериментально перевірені (Swiss-Prot) та автоматично анотовані (TrEMBL) білкові записи.

- **UniRef (Reference Clusters)** – групи схожих білкових послідовностей для швидкого пошуку.

- **UniParc (Protein Archive)** – архів усіх білкових послідовностей з різних джерел.

**2. Основні можливості UniProt:**

- Пошук білкових послідовностей за назвою, організмом або ідентифікатором.
- Отримання структурної та функціональної інформації.
- Визначення білкових доменів та модифікацій.
- Аналіз еволюційної спорідненості білків.
- Завантаження даних у форматах FASTA, XML, TSV.

Як і більшість архівних БД біомедичного спрямування ( БД DDBJ, БД GenBank, БД EMBL ) є двовимірними ( тобто одному полю відповідає один запис), а їх функціональність розширюється за рахунок використання різних ідентифікаторів ( в межах одного запису). Однак на відміну від архівних БД кожен запис оцінюється експертом, який визначає ступінь достовірності як функцій самої органічної сполуки ( білка для БД UniProt ), так і функціональних властивостей окремих його частин. Банк даних UniProt об'єднує два банкі білкових послідовностей: 1. Swissprot – БД, що керується

та містить найбільш достовірну інформацію про білки. 2 TrEMBL – архівна БД, що містить автоматично проанотовані не перевірені записи.

**Обладнання та матеріали:** Комп'ютер із доступом до Інтернету. Веб-браузер для роботи з базою UniProt (<https://www.uniprot.org>). Текстовий редактор або спеціальні програми (наприклад, Notepad++, MEGA).

*Хід роботи:*

### **1. Відкриття бази UniProt**

1. Перейдіть на сайт **UniProt**.

2. У полі пошуку введіть назву білка, наприклад, "**Hemoglobin human**" (гемоглобін людини).

3. Натисніть **Search** (Пошук).

### **2. Аналіз запису про білок**

1. Виберіть запис білка **Hemoglobin subunit beta (P68871)**.

2. Ознайомтеся з основною інформацією:

○ **Protein name** – назва білка.

○ **Gene name** – назва гена.

○ **Organism** – організм, у якому знайдено білок.

○ **Sequence length** – довжина амінокислотної послідовності.

○ **Molecular weight** – молекулярна маса.

3. Перегляньте **FASTA-послідовність** білка.

### **3. Визначення білкових доменів і модифікацій**

1. У розділі "**Family & Domains**" знайдіть інформацію про білкові домени.

2. Перегляньте "**Post-Translational Modifications (PTMs)**" – які хімічні модифікації має білок.

### **4. Аналіз взаємодій білка**

1. Перейдіть у розділ "**Interaction**".

2. Ознайомтеся з білками-партнерами, з якими взаємодіє гемоглобін.

3. Визначте біологічне значення цих взаємодій.

### **5. Завантаження послідовності білка**

1. Натисніть "**Download**" (Завантажити).

2. Оберіть формат **FASTA** або **XML**.

3. Збережіть файл для подальшого аналізу.

### **Контрольні запитання:**

1. Що таке база даних UniProt і для чого вона використовується?
2. Які основні розділи має UniProt?
3. Як знайти білкову послідовність у UniProt?
4. Де можна знайти інформацію про функцію білка?
5. Як можна отримати взаємодії білка з іншими білками?

### **ПРАКТИЧНА РОБОТА №7 ВИЗНАЧЕННЯ РЕДАКЦІЙНОЇ ВІДСТАНИ МІЖ БІОЛОГІЧНИМИ ПОСЛІДОВНОСТЯМИ. АЛГОРИТМ ВАГНЕРА–ФІШЕРА. ВИРІВНЮВАННЯ БІОЛОГІЧНИХ ПОСЛІДОВНОСТЕЙ З ЛІНІЙНОЮ ПАМ'ЯТТЮ. АЛГОРИТМ МІЛЛЕРА-МАЙЄРСА**

**Мета роботи:** Ознайомитися з методами обчислення редакційної відстані між біологічними послідовностями. Вивчити алгоритми Вагнера–Фішера та Міллера–Майєрса. Навчитися виконувати вирівнювання біологічних послідовностей з використанням обмеженої пам'яті.

### **Теоретичні відомості:**

#### **1. Редакційна відстань (відстань Левенштейна)**

Редакційна відстань між двома послідовностями визначає мінімальну кількість операцій (вставка, видалення, заміна), необхідну для перетворення однієї послідовності в іншу.

Редакційну відстань часто використовують для порівняння біологічних послідовностей (ДНК, РНК, білків).

#### **2. Алгоритм Вагнера–Фішера**

Це динамічний алгоритм, який використовує матрицю розміром  $(m+1) \times (n+1)$ , де  $m$  і  $n$  — довжини порівнюваних послідовностей.

#### **Основні кроки:**

1. Створюється матриця  $DDD$  розміром  $(m+1) \times (n+1)$ .
2. Ініціалізуються перший рядок та перший стовпець значеннями  $0, 1, 2, \dots, m$  і  $0, 1, 2, \dots, n$ .
3. Заповнення матриці відбувається за формулою:

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 & \text{(видалення)} \\ D(i, j-1) + 1 & \text{(вставка)} \\ D(i-1, j-1) + (s_i \neq t_j) & \text{(заміна)} \end{cases}$$

4. Остаточне значення  $D(m, n)$  — це редакційна відстань між двома послідовностями.

**Перевага:** працює для будь-яких послідовностей.

**Недолік:** займає  $O(m \times n)$  пам'яті.

### 3. Алгоритм Міллера–Майєрса

Цей алгоритм є оптимізованою версією Вагнера–Фішера, що використовує **лінійну пам'ять**  $O(\min(m, n))$ .

#### Основна ідея:

1. Використовується **розділяй і володарюй**: обчислення виконується спочатку **зліва направо**, потім **справа наліво**, що дозволяє знайти **середню точку вирівнювання**.

2. Рекурсивно розбиває проблему на дві менші задачі.

3. Використовує лише два рядки матриці одночасно, що зменшує споживання пам'яті.

**Перевага:** займає **лінійну пам'ять**  $O(m+n)$ .

**Недолік:** складний у реалізації через рекурсію.

Редакційна відстань або відстань Левенштайна – метрика що дозволяє визначити на скільки подібні дві послідовності. Редакційна відстань – мінімальна кількість операцій редагування (вставка, видалення або заміна символу), що дозволяє перетворити одну послідовність в іншу. Алгоритм Вагнера – Фішера методом динамічного програмування дає можливість визначити редакційну відстань та дозволяє отримати редакційний припис (послідовність операцій редагування).

**Обладнання та матеріали:** Комп'ютер із доступом до Інтернету. Мови програмування (Python, C++ або Java) для реалізації алгоритмів. Онлайн-інструменти для вирівнювання послідовностей:

EMBOSS  
([https://www.ebi.ac.uk/Tools/psa/emboss\\_needle/](https://www.ebi.ac.uk/Tools/psa/emboss_needle/))

Needle  
BLAST  
(<https://blast.ncbi.nlm.nih.gov/Blast.cgi>)

## Хід роботи:

### 1. Визначення редакційної відстані за алгоритмом Вагнера–Фішера

1. Візьмемо дві нуклеотидні послідовності, наприклад:

```
ini Копіювати Редагувати  
  
Seq1 = "GATTACA"  
Seq2 = "GCATGCU"
```

2. Побудуємо матрицю редакційної відстані вручну або за допомогою Python.

### Приклад коду на Python:

Python

```
python Копіювати Редагувати  
  
import numpy as np  
  
def wagner_fischer(seq1, seq2):  
    m, n = len(seq1), len(seq2)  
    D = np.zeros((m+1, n+1), dtype=int)  
  
    for i in range(m+1):  
        for j in range(n+1):  
            if i == 0:  
                D[i][j] = j  
            elif j == 0:  
                D[i][j] = i  
            else:  
                D[i][j] = i  
            else:  
                cost = 0 if seq1[i-1] == seq2[j-1] else 1  
                D[i][j] = min(D[i-1][j] + 1, # Видалення  
                             D[i][j-1] + 1, # Вставка  
                             D[i-1][j-1] + cost) # Заміна  
  
    return D, D[m, n]  
  
seq1 = "GATTACA"  
seq2 = "GCATGCU"  
matrix, distance = wagner_fischer(seq1, seq2)  
print("Редакційна відстань:", distance)  
print(matrix)
```

**Результат:** отримана матриця та редакційна відстань між послідовностями.



## 2. Визначення редакційної відстані за алгоритмом Міллера–Майєрса

### 1. Реалізуємо оптимізовану версію алгоритму:

```
python 📄 Копіювати 🗑️ Редагувати  
  
def miller_myers(seq1, seq2):  
    m, n = len(seq1), len(seq2)  
  
    def forward_pass(d):  
        current = list(range(n + 1))  
        for i in range(1, m + 1):  
            previous, current = current, [i] + [0] * n  
            for j in range(1, n + 1):  
                cost = 0 if seq1[i - 1] == seq2[j - 1] else 1  
                current[j] = min(previous[j] + 1, # Видалення  
                                current[j - 1] + 1, # Вставка  
                                previous[j - 1] + cost) # Заміна  
  
        return current  
  
    return forward_pass(n)[-1]  
  
seq1 = "GATTACA"  
seq2 = "GCATGCU"  
distance = miller_myers(seq1, seq2)  
print("Редакційна відстань (Miller-Myers):", distance)
```

**Результат:** алгоритм виконує ті ж операції, але використовує лише лінійну пам'ять.

### *Контрольні запитання:*

1. Що таке редакційна відстань?
2. Як працює алгоритм Вагнера–Фішера?
3. Чим алгоритм Міллера–Майєрса відрізняється від Вагнера–Фішера?
4. Чому алгоритм Міллера–Майєрса ефективніший?
5. Як можна використовувати ці алгоритми у біоінформатиці?

## ЛІТЕРАТУРА

1. Frances S. Turner. Assessment of insert sizes and adapter content in fastq data from NexteraXT libraries. *Frontiers in Genetics. Bioinformatics and Computational Biology*. January 2014. V. 5. Article 5
2. Eva C Berglund, Anna Kiialainen, Ann-Christine Syudnen. Next-generation sequencing technologies and applications for human genetic history and forensics. *Investig Genetic*. 2011; 2: 23.
3. Babraham bioinformatics URL:  
<https://www.bioinformatics.babraham.ac.uk/projects/fastqc>
4. Coursera. URL:  
<https://www.coursera.org/learn/bioinformatika/home/welcome>
5. David J. Edwardsand, Kathryn E. Holt. Beginner's guide to comparative bacterial genome analysis using next-generation sequence data. *Microb Inform Exp*. 2013. 3: 2.
6. David J. Edwards, Kathryn E. Holt. Bacterial Comparative Genomics Tutorial. URL:  
<https://holtlab.net/2015/02/25/tools-for-bacterial-comparative-genomics/>
7. Мороховець Г. Ю., Сілкова О. В. Біоінформатика. Вступний курс : навчальний посібник. Полтава : Видавець Шевченко Р. В., 2017. 118 с., іл.
8. Основи молекулярної біології та біоінформатики: комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки та інформаційні технології спеціалізації «Інформаційні технології в біології та медицині» / С. В.Кисляк, Є. А.Настенко; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2018. 95 с.
9. Біоінформатика. Практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 162 «Біотехнологія та біоінженерії» / КПІ ім. Ігоря Сікорського; автор.: С. В. Горобець, О. Ю.Горобець, І. В. Дем'яненко. Київ : КПІ ім. Ігоря Сікорського, 2020. 87 с.
10. Кеца О. В. Основи біоінформатики : навч.-метод. посібник. Чернівці : Чернівецький нац. ун-т ім. Ю. Федьковича, 2018. 192 с.