Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Інтелектуальна система для оптимізації вибору мембранно-активних пептидів
(тема)

Виконав:
студент 2 курсу, групи СШІм-21-2
Гузенко М.Р.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Музика К.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____20____р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Гузенко Максиму Руслановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна система для оптимізації вибору мембранно-активних пептидів

затверджена наказом університету від 31 березня 20 23 р. № 306Ст
2. Термін подання студентом роботи до екзаменаційної комісії 23 травня 20 23 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження систем з використанням штучного інтелекту, середовище розробки Visual Studio Code 2019, мова програмування Python.

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної області
2) Розробка вимог до розроблювальної системи
3) Опис прийнятних проектних рішень при розробці системи
4) Аналіз та проектування архітектури алгоритму
5) Тестування розробленої інформаційної системи
6) Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	08.05.2023	Виконано
2	Опис системи	10.05.2023	Виконано
3	Моделювання та навчання системи	12.05.2023	Виконано
4	Визначення проблем при розробці системи	14.05.2023	Виконано
5	Знаходження та вирішення проблем	15.05.2023	Виконано
6	Розробка інтелектуальної системи	17.05.2023	Виконано
7	Написання пояснювальної записки	18.05.2023	Виконано
8	Попередній захист	19.05.2023	Виконано
9	Захист перед ЕК	23.05.2023	Виконано

Дата видачі завдання 3 квітня 20 23 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Музика К.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 с., 9 рис., 3 табл., 5 дод., 10 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, НЕЙРОННА МЕРЕЖА, ОПТИМІЗАЦІЯ ВИБОРУ ПЕПТИДІВ, DATA ANALYSIS, PYTHON.

Об’єкт дослідження – інтелектуальна система для оптимізації вибору мембранно-активних пептидів.

Мета роботи – полягає в розробці і реалізації інтелектуальної системи, яка буде використовуватися для оптимізації процесу вибору мембранно-активних пептидів. Основною метою є покращення ефективності та швидкості вибору оптимальних пептидних структур для мембранної активності.

Методи дослідження – аналіз літературних джерел, виконання систематичного огляду літератури щодо наявних методів та підходів до вибору мембранно-активних пептидів. Цей метод дозволяє ознайомитися зі станом справ у відповідній області досліджень, виявити ключові методології та інструменти, що вже використовуються, а також виявити потенційні проблеми чи недоліки існуючих підходів. Далі розроблення програмної системи, що базується на комп'ютерних алгоритмах та методах машинного навчання для оптимізації вибору мембранно-активних пептидів. Цей метод включає розробку алгоритмів, створення моделей машинного навчання, налаштування параметрів системи, реалізацію інтерфейсу та інших необхідних компонентів системи.

Система буде базуватися на аналізі великого обсягу даних, включаючи інформацію про структури пептидів, їх взаємодію з мембранами та біологічними структурами.

ABSTRACT

Explanatory note: 74 p., 9 fig., 3 tabl., 5 ann., 10 sources.

DATA ANALYSIS, INFORMATION SYSTEM, NEURAL NETWORK, OPTIMIZATION OF PEPTIDE SELECTION, PYTHON.

The object of research is an intelligent system for optimizing the selection of membrane-active peptides

The purpose of the work is to develop and implement an intelligent system that will be used to optimize the process of selecting membrane-active peptides. The main goal is to improve the efficiency and speed of selection of optimal peptide structures for membrane activity.

Research methods – analysis of literary sources, systematic review of the literature on available methods and approaches to the selection of membrane-active peptides. This method allows you to familiarize yourself with the state of affairs in the relevant research area, identify key methodologies and tools already in use, and identify potential problems or shortcomings of existing approaches. Next, the development of a software system based on computer algorithms and machine learning methods to optimize the selection of membrane-active peptides. This method includes the development of algorithms, the creation of machine learning models, the setting of system parameters, the implementation of the interface and other necessary system components.

The application of an intelligent system to optimize the selection of membrane-active peptides aims to improve research efficiency, reduce the time and effort required for peptide selection, and improve the accuracy of results. The developed system can be useful for scientific research, pharmaceutical and biotechnological industries and analysis, a useful tool was written based on machine learning for automatedselection of effective peptides, which accelerates the research process.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

Нейронна мережа – клас глибинних мереж, обчислювальні моделі, що натхненний біологічними нейронними мережами;

Ядерно магнітний резонанс – фізичне явище, яке використовується для отримання інформації про структуру та властивості атомів, молекул і матеріалів;

AMP – Antimicrobial Peptide – невелике біологічний з'єднання, яке виявляє активність проти мікроорганізмів, таких як бактерії, гриби, віруси і паразити;

CPP – Cell-Penetrating Peptide – клітинно проникаючий пептид, короткий та послідовний фрагмент амінокислот, який має здатність проникати через клітинну мембрану;

PTD – Protein Transduction Domain – коротка послідовність амінокислот, яка здатна проникати через клітинну мембрану і доставляти різноманітні молекули, включаючи білки, пептиди та нуклеїнові кислоти, до внутрішньоклітинного простору;

SMILES – система для кодування хімічних структур в текстовому форматі. Вона використовується в хімії та біології для представлення молекулярних структур, зокрема органічних сполук.

ВСТУП

В останні роки вивчення та дослідження мембранно-активних пептидів здобуло велике значення в багатьох галузях, зокрема в медицині, фармацевтиці та біотехнологіях. Мембранно-активні пептиди виявляють унікальні властивості, такі як взаємодія з біологічними мембранами, молекулярне розпізнавання та модуляція біологічних процесів. Це відкриває безліч можливостей для застосування цих пептидів у дослідженнях, діагностиці та терапії.

Однак, вибір оптимальних мембранно-активних пептидів для конкретних завдань залишається складним завданням через їхню велику кількість та різноманітність. В цьому контексті, розробка інтелектуальної системи, спроможної оптимізувати вибір мембранно-активних пептидів, набуває великого практичного значення. Інтеграція інтелектуальних алгоритмів та методів машинного навчання в цей процес може значно прискорити та поліпшити результати вибору пептидів для подальшого дослідження та застосування.

Метою даної дипломної роботи є розробка інтелектуальної системи для оптимізації вибору мембранно-активних пептидів. Робота буде фокусуватися на вивченні та аналізі різних методів машинного навчання та оптимізації, а також розробці алгоритмів, що дозволять підібрати найбільш ефективні та промислово значущі пептиди для використання в різних галузях. В результаті роботи очікується створення інтелектуальної системи, яка забезпечить автоматизований та ефективний процес вибору мембранно-активних пептидів. Система буде базуватись на аналізі великої кількості даних, таких як структурні характеристики пептидів, фізико-хімічні властивості та експериментальні дані про їхню активність.

Одним з ключових завдань дипломної роботи є розробка алгоритмів та моделей машинного навчання, які допоможуть визначити взаємозв'язки між структурою та властивостями пептидів, а також знайти оптимальні

комбінації параметрів для досягнення певних цілей. Враховуючи складність та об'єм інформації, необхідною буде розробка ефективних алгоритмів обробки та аналізу даних, а також створення інтерфейсу для користувачів системи.

Основними перевагами запропонованої інтелектуальної системи є зменшення часу та затрат на вибір мембранно-активних пептидів, покращення ефективності та точності вибору, а також розширення областей застосування цих пептидів. Дана робота має великий потенціал у медичних дослідженнях, розробці нових препаратів, діагностиці хвороб та багатьох інших областях, де мембранно-активні пептиди використовуються як ключові компоненти.

Загальним завданням є підвищення рівня автоматизації та точності процесу вибору мембранно-активних пептидів, що сприятиме подальшому розвитку наукових досліджень та інноваційних знахідок у галузі дизайну нових пептидних структур. Окрім того, розроблена інтелектуальна система може стати потужним інструментом для підтримки наукових досліджень та сприяти появі нових знань про взаємодію мембранно-активних пептидів з біологічними структурами. Аналіз інформації, отриманої за допомогою цієї системи, може розкрити нові механізми дії пептидів та сприяти виявленню потенційних мішеней для їхньої взаємодії.

У практичному аспекті, інтелектуальна система для оптимізації вибору мембранно-активних пептидів може стати невід'ємною складовою фармацевтичної та біотехнологічної промисловості. Сприяти прискоренню процесу пошуку ефективних препаратів, які базуються на пептидних структурах, та покращенню їхньої якості та безпеки. Використання інтелектуальної системи дозволить знизити витрати часу та ресурсів, які зазвичай затрачаються на дослідження та тестування.

Отже, розробка інтелектуальної системи для оптимізації вибору мембранно-активних пептидів має великий потенціал для наукових досліджень, фармацевтичної та біотехнологічної промисловості.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Роль машинного навчання у дослідження мембранно- активних пептидів

Машинне навчання відіграє важливу роль у дослідженнях мембранно-активних пептидів, що є об'єктом інтенсивного вивчення у біохімії та фармацевтиці. Машинне навчання дозволяє аналізувати та моделювати складні взаємодії пептидів з мембранами та визначати їх активність і властивості. Деякі з основних ролей машинного навчання в цій області.

1.1.1 Прогнозування активності

Машинне навчання дозволяє розробляти моделі для прогнозування мембранної активності пептидів. Використовуючи вхідні дані, такі як амінокислотні послідовності, фізико-хімічні властивості та структурні особливості пептидів, моделі можуть передбачати їх потенційну активність, що допомагає відібрати потенційно цінні пептиди для подальших досліджень.

1.1.2 Аналіз структури

Машинне навчання може бути використане для аналізу структури мембранно-активних пептидів. Застосовуючи методи, такі як згорткові нейронні мережі, можна виявляти структурні паттерни та взаємодії, що впливають на функцію пептидів. Це допомагає в розумінні механізмів дії пептидів та їх взаємодії з мембранами та іншими біологічними системами.

1.1.3 Вивчення взаємодій

Машинне навчання дозволяє досліджувати взаємодію мембранно-активних пептидів з ліпідними білками та іншими компонентами мембрани. Застосовуючи методи машинного навчання, можна прогнозувати взаємодії та встановлювати ключові взаємодіючі домені, що сприяє розумінню молекулярних взаємодій у системах з мембранно-активними пептидами.

1.1.4 Пошук нових пептидів

Машинне навчання допомагає у прискоренні процесу пошуку нових мембранно-активних пептидів. Шляхом аналізу великих обсягів даних та використання алгоритмів машинного навчання, можна ідентифікувати нові пептиди з потенційною мембранною активністю, що може мати важливі застосування у біології, медицині та інших галузях.

1.2 Розуміння основних властивостей мембранно-активних пептидів

Сучасна наука про пептиди використовує методи та засоби біо-інформатики та хіміко-інформатики. Ці підходи використовують різні мови для опису пептидних структур – амінокислотні послідовності та хімічні коди такі як SMILES. Останній може бути застосований, наприклад, у порівняльних дослідженнях структур і властивостей пептидів і пептидо-міметиків. Прогрес у пептидній науці «in silico» може бути досягнутий завдяки кращій комунікації між біологами та хіміками, включаючи трансляцію представлення пептиду з послідовності амінокислот у код SMILES. Останні рекомендації щодо належної практики

хімічної інформації включають ретельну перевірку даних та їх анотації.

Сучасна наука про пептиди охоплює біологічний і хімічний підходи. Медичні науки, фармакологія, біотехнологія та, нарешті, але не менш важливе, науки про харчові продукти та харчування потребують як біології, так і хімії. Пептиди перебувають у центрі інтересу всіх перерахованих вище областей.

Мембранно-активні пептиди проявляють свою біологічну активність взаємодіючи з клітинною мембраною, щоб або порушити її та призвести до лізису клітини, або транслокувати через неї, щоб доставити вантажі в клітину та досягти своєї мети. Мембранно-активні пептиди є привабливою альтернативою фармацевтичним препаратам, що використовуються в даний час, і кількість антимікробних пептидів (AMP) і пептидів, призначених для доставки ліків і генів у конвеєрі ліків, зростає. Існують два основних класи мембраноактивних пептидів: АМФ і пептиди, що проникають у клітину (CPP).

Антимікробні пептиди – це група мембраноактивних пептидів, які порушують цілісність мембрани або пригнічують клітинні функції бактерій, вірусів і грибків. Пептиди, що проникають у клітину, є ще однією групою мембрано-активних пептидів, які в основному функціонують як носії вантажу, хоча вони також можуть виявляти антимікробну активність. Біофізичні методи проливають світло на взаємодію пептид–мембрана з вищою роздільною здатністю завдяки прогресу в оптиці, обробці зображень і обчислювальних ресурсах. Дослідження структури мембранних активних пептидів у присутності мембрани дає важливі підказки щодо впливу мембранного середовища на конформації пептидів. Методи візуалізації в реальному часі дозволяють досліджувати дію пептидів на рівні однієї клітини або однієї молекули. На додаток до цих експериментальних біофізичних методів моделювання молекулярної динаміки дає підказки про взаємодію між пептидами та ліпідами та динаміку процесу проникнення в клітину в деталях атома.

Пептиди, які взаємодіють з клітинною мембраною, руйнуючи її, проходячи крізь неї або залишаючись на межі мембрани і зливається з нею, відомі як мембранно-активні пептиди. Існує два основних класи мембрано-активних пептидів; антимікробні пептиди (AMP), які вбивають клітини, і пептиди, що проникають у клітини (CPP), які переносять вантажі через ліпідні подвійні шари.

1.2.1 Антимікробні пептиди

Антимікробні пептиди, також відомі як захисні пептиди господаря, вбивають бактерії, віруси та грибки або шляхом порушення цілісності їх мембран, або шляхом інгібування деяких клітинних функцій . Відкриття першого AMP, граміцидину, у 1939 році відкрило поле для АМП, які отримали постійний інтерес у результаті зосередженості на відкритті нових протимікробних препаратів через підвищення резистентності мікробів до звичайних антибіотиків. Їх унікальний спосіб дії разом із багатоцільовими властивостями зробили AMP перспективними кандидатами на розробку нових лікарських засобів. Антимікробні пептиди є частиною захисної системи хазяїна різних організмів [1], включаючи людину.

Антимікробні пептиди зазвичай складаються з менше ніж 100 амінокислотних залишків [2]. Зазвичай це L-амінокислоти, але AMP можуть також містити модифіковані залишки, такі як дисульфідні зв’язки або лантіоніни [3]. Як правило, вони мають позитивний сумарний заряд(від +4 до +6), як правило, завдяки розтягуванню залишків аргініну та/або лізину, які взаємодіють з негативно зарядженими фосфатними головними групами бактеріальної мембрани.

Вони також містять гідрофобну область, що робить їх амфіпатичними та мембрано-активними. Проте бактерії отримали стійкість проти катіонних AMP, придбавши позитивно заряджені групи на своїй

зовнішній клітинній стінці, і для боротьби з цими бактеріями також розвинулися деякі аніонні AMP, одним з яких є дермцидин, знайдений у людському поті. Існують також AMP із сумарним негативним зарядом (від -1 до -7), такі як хромацин [4], але вважається, що вони виконують інші основні біологічні ролі (рисунок 1.1).

Вторинна структура AMP різноманітна. Кластеризація 135 ядерно-магнітного резонансу (ЯМР) структур AMP на основі двогранних кутів основ показує, що вони можуть приймати широкий спектр вторинних структур, починаючи від повністю спіральних до всіх бета [5] . Антимікробні пептиди можна класифікувати на чотири підгрупи на основі їх вторинної структури; перша група включає лінійні, α-спіральні пептиди (меліттин, дермцидин, LL-37).

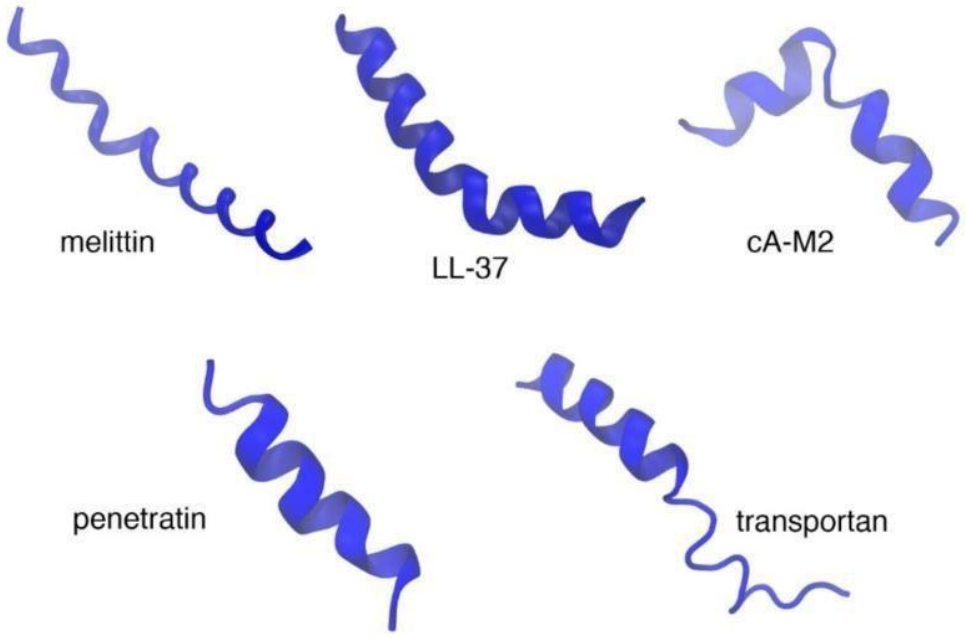


Рисунок 1.1 – α-спіральних антимікробних пептидів (AMP) (верхняпанель), пептиди що проникають у клітини (CPP) (нижня панель)

До другої групи належать пептиди з β-ланцюгами, з’єднаними двома або більше дисульфідними містками (дефензини та протегрини) (рисунок 1. 2), по-третє, AMP із міжмолекулярними

дисульфідними зв’язками, що демонструють петлеподібні/шпилькові структури (такі як бактенецин), і останньою групою є пептиди зі спеціальними або модифікованими амінокислотами (такі як багаті на пролін/аргінін пептид або лантібіотики, що містять лантіонін) [6]. Навіть у зв’язаному з мембраною стані пептид може приймати різні конформації залежно від концентрації. Інша група AMP, така як індоліцидин із сімейства кателіцидинів, не мають чітко вираженої вторинної структури.

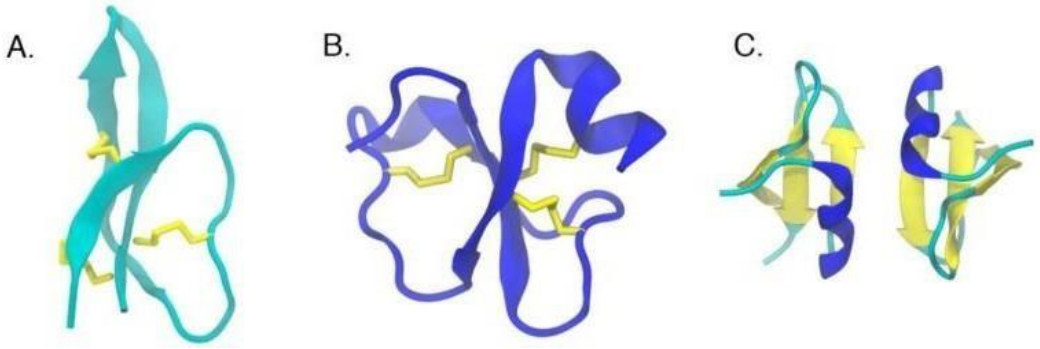


Рисунок 1.2 – Приклади циклічних пептидів з β-листовим ядром(A), Структура розчину людського дефензиву(B), кристалічні структури кротаміну(C)

Початковий контакт більшості AMP з мікробними мембранами зазвичай досягається неспецифічним шляхом за допомогою електростатичних і гідрофобних взаємодій. Після цього початкового контакту різні моделі намагаються пояснити дію AMP. Механізм дії AMP загалом можна розділити на пороутворюючі та непороутворюючі моделі.

Загальноприйняті моделі відомі як бочкоподібні та тороїдальні моделі утворення. Стовбурові пори використовують двошарове вуглеводневе ядро як матрицю для самоскладання пептидів, таким чином ліпідний шар майже не порушується. Пептиди орієнтовані перпендикулярно до площини мембрани, утворюючи досить жорсткий циліндричний ствол. У цій моделі спочатку AMP зв’язуються паралельно

поверхні мембрани як мономер, після чого відбувається олігомеризація та утворення пор. Коли вони вставляються в мембрану, пептиди зазвичай набувають амфіпатичної вторинної структури, в якій гідрофобні області взаємодіють з ліпідами мембрани, тоді як гідрофільні області утворюють просвіт каналу. Вони можуть бути α-спіральними або β-листовими структурами, але їх мінімальна довжина повинна охоплювати ліпідний шар. Аламетицин і пардаксин [7] є одними з цих пептидів, які створюють бочкоподібні пори.

У моделі тороїдальних пор відомо, що пептиди порушують нормальну сегрегацію полярних і неполярних частин мембрани. Коли пептиди вставляються в мембрану, вони утворюють пучок і спонукають ліпідні моно-шари згинатися. Ліпідна структура навколо тороїдальної пори зазнає сильного впливу, оскільки деякі з ліпідів беруть участь безпосередньо в порі, контактуючи з пептидами. У цій тороїдальній структурі пептиди можуть бути або перпендикулярними, або нахиленими відносно площини мембрани. Такі пори є тимчасовими, дозволяючи пептиду проникати в цитоплазму та націлюватися на внутрішньоклітинні компоненти. Іншими характеристиками тороїдальних пор є іонна селективність і дискретні розміри. Також були описані варіанти класичної тороїдальної пори, такі як величезна тороїдальна пора (для лактицину Q) і невпорядкована тороїдальна пора (для мелітину). Термін «невпорядкована тороїдальна» пора була введена для опису пор, які мають тороїдну форму, але лише один або два пептиди вистилають пори з частковою участю пептидів у порі. «Величезні тороїдальні» пори утворюються тільки вище критичного співвідношення пептидів і ліпідів і потребують локальної агрегації пептидів. Меліттин є прикладом тороїдальних пороутворюючих пептидів.

Антимікробні пептиди не обов'язково діють шляхом утворення пор можуть індукувати свої ефекти через неспецифічну мембранну пермеабілізацію. Одна з найпоширеніших моделей без утворення пор

відома як модель килима. У цій моделі АМР адсорбуються на поверхні мембрани подібно до миючих засобів і, таким чином, впливають на архітектуру мембрани. Взаємодії спочатку обумовлюються електростатикою, і коли досягається порогова концентрація АМР на поверхні мембрани, пептиди покривають поверхню мембрани у вигляді килима. На цьому етапі структура мембрани дестабілізується і більше не може підтримуватися, тому вона розпадається. Антимікробні пептиди, такі як ауреїн і цекропін, нав'язують свою діяльність за допомогою цієї моделі.

Різноманітна діяльність АМП може залежати від концентрації пептиду, типу клітини та властивостей мембрани, і з’ясування механізму дії цих АМП є ключовим у розумінні умов, у яких функціонують ці пептиди.

1.2.2 Пептиди що проникають в клітини

Пептиди, що проникають у клітину, також називають доменами білкової трансдукції (PTD), являють собою різноманітний набір мембранно-активних пептидів, що містять менше ніж 30 залишків, зазвичай із сумарним позитивним зарядом. Відомо, що вони полегшують доставку різних біо-молекул через клітинні мембрани еукаріотичних клітин з обмеженою токсичністю. Молекулярна маса біологічно активного вантажу, який може бути пов’язаний ковалентно або не ковалентно, може бути в декілька разів більшою за молекулярну масу CPP. Серед їхніх вантажів – плазмідна ДНК, олігонуклеотиди, siRNA (коротка інтерферуюча РНК), PNA (пептидна нуклеїнова кислота), білки та пептиди, засоби візуалізації, а також ліки.

Катіонний пептид був першим відкритим CPP [7], а потім пенетратин. Тат (пептид), злитий з β-галактозидазою, проникав через гематоенцефалічний бар’єр нетоксичним способом і розподілявся по

всьому мозку. Пенетратин, отриманий із третьої спіралі гомеодомену білка антенапедії дрозофіли, міг безпосередньо проникати у гігантські одношарові везикули (GUV). Амфіпатичні пептиди MPG і Pep-1 (Каріоти) складаються з трьох доменів: гідрофобного мотиву на їх N-кінці, гідрофільного домену, багатого на лізин, і лінкерного домену (WSQP), який підвищує гнучкість гідрофобного і гідрофільні домени. Інший амфіпатичний пептид, CADY, поєднує ароматичний триптофан і катіонні залишки аргініну в самозбірний пептид. Pep-1 може доставляти повнорозмірні антитіла та білки. Взаємодія амфіпатичного пептиду, зокрема залишків триптофану в гідрофобному домені, з пептидогліканами має вирішальне значення в кінцевому процесі інтерналізації(вторгнення в клітину). Серед найдавніших гідрофобних пептидів є транспортан, похідний від транспортану, який є химерним пептидом нейропептиду галаніну та мастопарану з отрути оси, з'єднаних ланцюгом лізину. Повідомлялося, що він глибоко занурюється в подвійний шар і перетинає його, несучи свій вантаж, такий як зелений флуоресцентний білок (GFP).

1.2.3 Анти-бактеріальні пептиди, та пептиди що проникають уклітини

Різниця між AMP і CPP не завжди чітка. Дійсно, багато CPP також служать AMP, і багато AMPs мають властивості проникати в клітини [8]. Наявність спільних характеристик, таких як амфіпатичність або наявність багатих на аргінін регіонів, між двома групами також свідчить про те, що вони можуть виконувати подвійну роль. Іноді одна мутація може змінити здатність пептиду проникати в клітину в бік антимікробної активності або навпаки. Наприклад, було виявлено, що вміст аргініну в пенетратині CPP впливає на його антимікробну активність.

Подібним чином підвищення катіонного характеру CPP Pep-1 посилило його антимікробну активність. Катіонні антибактеріальні

пептиди (CAP) мають дуже схожі фізико-хімічні характеристики з CPP, але вони виконують дві різні функції. Відомо, що проникаючі в клітини пептиди проникають в еукаріотичні клітини без будь-якої видимої токсичності або пошкодження, тоді як основною функцією CAP є знищення бактерій. Цікаво, що кілька CPP мають антибактеріальну дію. Було виявлено, що багато CPP, не пошкоджуючи еукаріотичні клітини, є мембранолітичними в бактеріях або в модельних мембранах, що імітують композицію бактеріального подвійного шару.

1.2.4 Характеристика антимікробних та проникаючих у клітини пептидів та їх взаємодії з мембранами

Численні біофізичні методи були застосовані для вивчення структурних особливостей мембранних активних пептидів, які знаходяться у вільному стані в розчині або коли вони вставляються в мембрани. Вторинна та третинна структури, конформація, орієнтація, стани олігомеризації пептидів у модельних мембранах представляють особливий інтерес, оскільки ці пептиди є мембранно-активними пептидами, деякі з них приймають структуру лише після зв’язування з мембраною. Таким чином, розуміння структурних особливостей пептидів, коли вони взаємодіють з мембраною, має першочергове значення для розуміння механічних деталей їх введення та мембранної дії.

Для характеристики мембранних активних пептидів можна використовувати різні методи, причому кожен метод дає різний рівень деталізації структури та механізму пептиду. Нижче наведено стислий перелік широко використовуваних біофізичних методів, які використовуються окремо або в комбінаціях для вивчення AMP і CPP та їх взаємодії з біологічними мембранами (рисунок 1.3) є схематичним зображенням основних підходів до визначення біофізичних характеристик, розглянутих у цьому огляді.

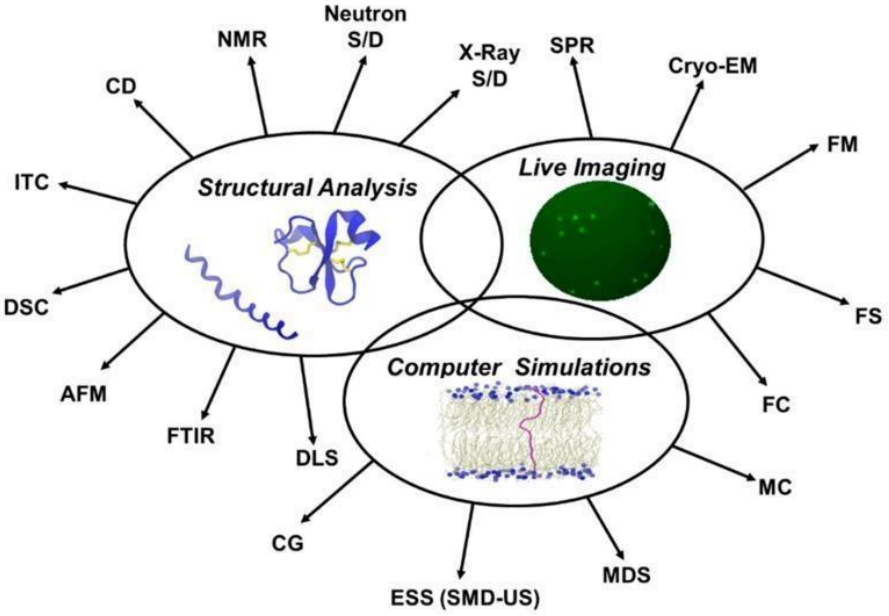


Рисунок 1.3 – Основні підходи, що використовуються для біофізичної характеристики мембранних активних пептидів

1.3 Методи дослідження пептидів

Інструменти для дослідження пептидів, поділяються на два види в залежності від якої сфери науки відштовхуватись. Бази даних і програми, використовують обидва підходи, які класифікуються як біо-інформатика та хіміко-інформатика, відповідно, хоча більшість спеціалізованих баз даних і програм, які призначені для пептидів, можна класифікувати як інструменти біоінформатики. Обидва підходи використовують різні мови для опису структур біо-молекул (пептидів). Біо-інформатика працює на основі амінокислотних послідовностей, а хіміо-інформатика – на універсальних хімічних кодах. Комунікація між цими двома сферами потребує перекладу з біологічної на хімічну мову.

Набір даних щодо біологічної активності пептидів, що впливають на смак, опублікований у нашому огляді, може служити прикладом переваг від об’єднання біологічного та хімічного підходів. Це не могло бути завершено без перевірки баз даних з використанням пептидних структур,

анотованих у коді SMILES, як запит. Інший приклад дослідження з використанням хімічного підходу нещодавно опублікували Оптік-Мартінес. Використовуючи програму SwissTargetPrediction, яка може прогнозувати взаємодії між малими пептидами кукурудзи та білками людського організму. Хімічні модифікації пептидів, спрямовані на зміну їх біологічної активності, останнім часом можна вважати «гарячою темою». Обробка пептидних послідовностей, включаючи небілкові або модифіковані амінокислоти, можлива, наприклад, за допомогою програми PepstrMod. В цій програмі використовуються сотні небілкових або модифікованих амінокислотних залишків.

Можливий простір неприродних або модифікованих амінокислот та інших можливих складових пептидів містить мільярди можливих молекул або фрагментів молекул SMILES та інші хімічні коди та формати дозволяють описувати будь-які штучно введені замінники для вивчення властивостей модифікованих пептидів. Іншим підходом є пошук пептидоміметиків, які можна використовувати як лікарський засіб, на основі відомих пептидних структур.

Багато програм, що використовують код SMILES, такі як, китайська BioTriangle, яка служить для розрахунку, наприклад, фізико-хімічних і топологічних параметрів малих молекул. Деякі програми, які використовують SMILES, на веб-сайтах біоінформатики, які дозволяє прогнозувати властивості, що впливають на застосовність речовини як ліків. Іншим прикладом програми, яка використовує хімічний код, є WebMolCS та інші програми. Крім зазначених вище безкоштовних програм, існують також комерційні інструменти, такі як JChem або MadFast, обидва надані ChemAxon, які використовують хімічні коди для перевірки бази даних або обчислень.

Зазначені вище ресурси інтегровані через метабазу SATPdb. Іншим прикладом є база даних сенсорних пептидів і амінокислот BIOPEP, надана Вармінсько-Мазурським університетом в Ольштині, Польща. Таким чином

ця сфера науки вже має багато досліджень завдяки насамперед нейроним мережам, що використовують хімічні коди.

Біо-інформаційний підхід до пептидів передбачає, наприклад, моделювання структур і прогнозування взаємодії з біо-макромолекулами на основі амінокислотних послідовностей. Структурне моделювання, що включає амінокислотні послідовності, можна виконувати за допомогою таких програм, як PepstrMod, Pep-Fold. Наприклад, підхід кількісного співвідношення структура-активність (QSAR) передбачає набір параметрів, які описують структуру та фізико-хімічні властивості окремих амінокислотних залишків. Підхід, заснований на послідовності, розширений за допомогою складу псевдо-амінокислот. Застосування хімічної інформації для анотації пептидів і обробки їх структури може розширити спектр інструментів, доступних для дослідження мембранно-активних пептидів.

Однак інструменти хіміо-інформатики не можна розглядати та використовувати некритично як «чорні ящики». Багато опублікованих наборів даних містять помилки. Користувачі або куратори баз даних і програм, що використовують хімічну інформацію, повинні бути готові розпізнавати та виправляти можливі помилки. Перевірка репрезентацій, ідентифікація та виправлення неправильно позначених сполук рекомендовані як один із важливих етапів підготовки та контролю набору даних сполук. Підготовка наборів пептидних даних, що включає трансляцію амінокислотних послідовностей у хімічні коди, не є винятком. Дані про пептиди, анотовані за допомогою кодів хімічної інформації, потребують ретельної перевірки перед використанням.

1.4 Використання Інтелектуальних систем для оптимізації вибору мембранно-активних пептидів: Результати, технології тапотенціал

Біологічні дослідження та розробки мають значний вплив на різні

галузі, включаючи фармацевтику, медицину та біоінженерію. Одним з важливих напрямків в цих галузях є дослідження мембранно-активних пептидів, які відіграють важливу роль в процесах транспорту речовин через клітинні мембрани. Вибір ефективних та безпечних мембранно-активних пептидів є великим викликом, оскільки потрібно оцінювати їхню активність, стабільність та токсичність.

1.4.1 Роль Інтелектуальних систем

Інтелектуальні системи, зокрема системи машинного навчання та штучні нейронні мережі, виявляють великий потенціал у покращенні вибору мембранно-активних пептидів. Ці системи можуть аналізувати великі обсяги даних та виявляти складні залежності між характеристиками пептидів та їхньою активністю.

Штучні нейронні мережі, зокрема згорткові та рекурентні мережі, можуть виявляти патерни та залежності в послідовностях амінокислот, що сприяє точнішому прогнозуванню мембранної активності (рисунок 1.4).

Рисунок 1.4 – Види основних нейронних мереж

1.4.2 Алгоритми та методи

Для оптимізації вибору мембранно-активних пептидів застосовуються різні алгоритми та методи. Починаючи зі збору даних про пептиди та їхні властивості, проводиться попередня обробка даних, включаючи фільтрацію та нормалізацію. Далі використовуються методи машинного навчання, такі як навчання з учителем або навчання без учителя, для створення моделей, які можуть прогнозувати мембранну активність на основі вхідних даних. Важливо налаштувати параметри мережі, включаючи розмір фільтрів, крок зсуву та функції активації, щоб досягти оптимальної точності та швидкості обчислень.

1.4.3 Експерименти та результати

В ході досліджень проводяться експерименти та тестування розроблених систем на різних наборах даних. Це дозволяє оцінити ефективність та точність системи в прогнозуванні мембранної активності пептидів. Результати експериментів показують, що Інтелектуальні системи здатні досягати високої точності в класифікації пептидів на мембранно-активні танеактивні.

1.4.4 Потенціал та виклики

Біологічні дослідження та розробки мають значний вплив на різні галузі, включаючи фармацевтику, медицину та біоінженерію. Одним з важливих напрямків в цих галузях є дослідження мембранно-активних пептидів, які відіграють важливу роль в процесах транспорту речовин через клітинні мембрани. Вибір ефективних та безпечних мембранно-активних пептидів є великим викликом, оскільки потрібно оцінювати їхню активність, стабільність та токсичність.

1.4.5 Роль Інтелектуальних систем

Інтелектуальні системи, зокрема системи машинного навчання та штучні нейронні мережі, виявляють великий потенціал у покращенні вибору мембранно-активних пептидів. Ці системи можуть аналізувати великі обсяги даних та виявляти складні залежності між характеристиками пептидів та їхньою активністю. Штучні нейронні мережі, зокрема згорткові та рекурентні мережі, можуть виявляти патерни та залежності в послідовностях амінокислот, що сприяє точнішому прогнозуванню мембранної активності.

1.4.6 Алгоритми та методи

Для оптимізації вибору мембранно-активних пептидів застосовуються різні алгоритми та методи. Починаючи зі збору даних про пептиди та їхні властивості, проводиться попередня обробка даних, включаючи фільтрацію та нормалізацію. Далі використовуються методи машинного навчання, такі як навчання з учителем або навчання без учителя, для створення моделей, які можуть прогнозувати мембранну активність на основі вхідних даних. Важливо налаштувати параметри мережі, включаючи розмір фільтрів, крок зсуву та функції активації, щоб досягти оптимальної точності та швидкості обчислень.

2 ПОСТАНОВКА ЗАДАЧІ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

2.1 Інтелектуальна система

Інтелектуальна система – це програмне або апаратне забезпечення, що має здатність здійснювати аналіз, розуміння, вирішення проблем, прийняття рішень або виконання завдань, які зазвичай потребують інтелекту людини. Вона базується на принципах штучного інтелекту (AI) і використовує різноманітні методи та алгоритми для виконання своїх функцій.

Інтелектуальні системи (ІС) можуть використовуватись у багатьох галузях, включаючи науку, медицину, фінанси, транспорт, енергетику та інші. Вони можуть виконувати такі завдання, як автоматичний аналіз даних, розпізнавання образів, голосове управління, рекомендації, передбачення, оптимізація процесів, управління ресурсами та інші.

ІС можуть включати в себе різні компоненти, такі як машинне навчання, глибоке навчання, нейронні мережі, експертні системи, обробку природної мови, комп'ютерне зору, розпізнавання шаблонів та інші технології. Вони надають можливість автоматизувати складні процеси, збільшити ефективність роботи та приймати рішення на основі аналізу великих обсягів даних.

Однією з ключових характеристик ІС є їх здатність до самоадаптації та навчання на основі даних. Вони можуть покращувати свою продуктивністьта результати з часом шляхом аналізу інформації, взаємодії з оточенням та зворотного зв'язку.

Цей процес навчання та адаптації інтелектуальних систем зазвичай здійснюється за допомогою методів машинного навчання, де система «вчиться» на основі набору даних і здатна виявляти закономірності, зробити узагальнення та здійснювати прогнози на нових даних. Інтелектуальні системи можуть мати різні рівні складності і

функціональності, від простих систем, що виконують обмежені завдання, до складних систем зі здатністю до самостійного прийняття рішень та розуміння складних контекстів. Вони можуть використовувати одну або кілька технологій штучного інтелекту, таких як нейронні мережі, генетичні алгоритми, логічне програмування, експертні системи та інші.

Одним з основних завдань інтелектуальних систем є підвищення продуктивності та ефективності процесів у різних сферах. Наприклад, вони можуть бути використані для автоматизації бізнес-процесів, оптимізації виробничих ліній, прогнозування ринкових тенденцій, покращення медичної діагностики та лікування, підвищення безпеки та багато інших областей.

Інтелектуальні системи мають потенціал розширювати межі людського розуму і допомагати вирішувати складні завдання. Вони можуть здатися або навіть перевершити людські здібності в деяких аспектах, таких як обробка великих обсягів даних, швидкість прийняття рішень та точність прогнозів. Проте, важливо пам'ятати, що інтелектуальні системи все ще базуються на алгоритмах та моделях, розроблених людьми. Вони можуть використовувати складні математичні моделі, статистичні методи, природні мови, виокремлювання ознак та багато інших технік для аналізу даних та прийняття рішень.

ІС можуть бути розроблені для вирішення конкретних завдань, таких як розпізнавання образів, класифікація тексту, рекомендації, прогнозування та багато інших. Для цього вони можуть використовувати різні алгоритми та моделі, такі як нейронні мережі, дерева рішень, методи опорних векторів, байєсовські мережі та інші.

Крім того, розвиток інтелектуальних систем пов'язаний з постійним вдосконаленням та оптимізацією алгоритмів. Дослідники та розробники постійно працюють над вдосконаленням методів навчання, покращенням швидкості та точності моделей, зменшенням обчислювального затрат та розробкою нових підходів до розв'язання складних завдань. Важливим

етапом в розробці інтелектуальних систем є навчання моделей на великих обсягах даних. Цей процес вимагає наявності якісних та репрезентативних наборів даних, а також ефективних алгоритмів навчання, які дозволяють моделям виявляти закономірності та робити узагальнення на нових даних.

Загалом, інтелектуальні системи є потужним інструментом для аналізу даних, виявлення складних залежностей та прийняття рішень на основі цих аналізів. Вони здатні працювати з великими обсягами даних, виявляти тенденції, патерни та кореляції, які можуть залишатися непоміченими людським спостереженням. Інтелектуальні системи можуть виявляти приховані закономірності в даних, навіть коли вони не очевидні для людей. Вони здатні автоматично вирішувати завдання класифікації, кластеризації, прогнозування, розпізнавання образів та інші, що дозволяє виконувати складні аналітичні завдання з високою точністю та ефективністю.

Однак, важливо пам'ятати, що інтелектуальні системи є інструментом, який потребує належного налаштування та контролю. Це означає, що їх використання повинно бути обгрунтованим, а результати їх роботи слід перевіряти та аналізувати з урахуванням контексту та потенційних обмежень. Крім того, інтелектуальні системи можуть бути ефективні лише при використанні якісних та достовірних даних. Неправильні або неповні дані можуть призвести до некоректних аналітичних результатів та невірних висновків. Тому важливо мати процеси збору, очищення та підготовки даних перед їх використанням у системі.

2.2 Виявлення проблем та актуалізація рішень

Вивчення та аналіз пептидів, які складаються з послідовності амінокислот, має велике значення для багатьох галузей, включаючи фармацевтику, біоінформатику та медичну науку. За допомогою штучних

нейронних мереж можна ефективно сортувати та аналізувати ці послідовності з метою знаходження спільних характеристик, класифікації або ідентифікації певних структур.

Перед розробкою штучної нейронної мережі для сортування пептидів, необхідно виявити проблеми, що виникають у процесі сортування даних у текстовому форматі .fasta. Однією з таких проблем може бути недостатня точність класифікації пептидів або низька швидкість обробки даних. Для цього потрібно провести аналіз існуючих методів сортування та виявити їх обмеження.

Після виявлення проблем можна актуалізувати рішення шляхом створення штучної нейронної мережі, яка буде спроможна ефективно сортувати дані у форматі .fasta. Для досягнення цієї мети можна розглянути різні архітектури нейронних мереж, такі як рекурентні

нейронні мережі (RNN), згорткові нейронні мережі (CNN) або комбінації цих підходів. Також можна дослідити методи переднього навчання (pretraining) для поліпшення ефективності штучної нейронної мережі при сортуванні пептидів. Переднє навчання передбачає попередню підготовку моделі на великому обсязі несортованих даних перед фінальним навчанням на вузькій області інтересу.

Такий підхід може бути особливо корисним у випадках, коли доступні обмежені обсяги анотованих даних для сортування пептидів. Під час переднього навчання модель може навчитися загальним характеристикам пептидних послідовностей, що допомагає поліпшити її здатність до класифікації та сортування нових даних. Одним з підходів до переднього навчання є використання безлабельних (unsupervised) методів, наприклад, автокодування (autoencoders) або генеративних засобів, для виявлення внутрішньої структури та закономірностей в пептидних послідовностях. Це може допомогти моделі побудувати компактні та виразні подання даних, що сприяє покращенню її здатності до сортування. Крім того, під час переднього навчання можна використовувати

недоступні для сортування допоміжні дані, такі як фізико-хімічні властивості амінокислот або структури білків, щоб покращити здатність моделі до розпізнавання та сортування пептидів.

Однак, важливо зазначити, що переднє навчання може бути обчислювально витратним та вимагати великих обсягів даних. Тому вирішення проблеми вибору оптимального алгоритму переднього навчання та підходу є важливим кроком у дослідженні сортування пептидів. Для цього можна провести експерименти з різними методами переднього навчання та оцінити їх ефективність.

Одним з підходів може бути використання переднього навчання з використанням великої бази несортованих пептидних послідовностей, яка охоплює широкий спектр біологічних властивостей. Такий підхід може допомогти моделі виявити загальні закономірності та характеристики пептидів, що сприятимуть точному сортуванню.

Також, варто дослідити можливість використання попереднього навчання на підмножинах пептидних послідовностей з відомими класифікаціями або анотаціями. Наприклад, можна використовувати навчання з вчителем (supervised learning) на невеликих підмножинах даних, що допоможе моделі покращити свою здатність до розпізнавання та сортування пептидів.

Оцінка ефективності різних алгоритмів переднього навчання може проводитись за допомогою метрик, таких як точність класифікації, чутливість та специфічність. Для порівняння результатів можна також використовувати стандартні набори даних, що містять пептидні послідовності з відомими анотаціями. Вибір оптимального алгоритму переднього навчання та підходу має бути зроблений на основі ретельного аналізу результатів експериментів та порівняння їх з вимогами і задачами сортування пептидів. Також слід враховувати обмеження та вимоги щодо обчислювальних ресурсів та швидкодії моделі при виборі оптимального алгоритму переднього навчання. Деякі методи переднього навчання

можуть бути більш витратними з точки зору обчислювальних ресурсів та часу навчання, тому слід уважно розглядати ці аспекти. Крім того, важливо враховувати наявність належних даних для переднього навчання. Дані, які використовуються для попереднього навчання, повинні бути репрезентативними та відображати різноманітність пептидних послідовностей, з якими модель буде працювати.

2.3 Постановка задачі

Задачею розробки цієї роботи є створення інтелектуальної системи, яка зможе з 1000 різноманітних пептидів визначити саме мембранно-активні пептиди, шляхом навчання штучної нейронної мережі аналізувати характеристики амінокислот, та виділяти необхідні.

Після фільтрації та знаходження мембранно-активних пептидів, сортувати виділені амінокислоти за їх токсичністю, опираючись на їх структуру.

Критерії вибору необхідних для завдання пептидів мають такі характеристики:

– гідрофобність: мембранно-активні пептиди часто мають значну гідрофобну складову, що сприяє їх взаємодії з ліпідними білками мембрани;

– амфіпатичність: вони мають амфіпатичну структуру, тобто складаються з гідрофобної (нефільної до води) та гідрофільної (фільної до води) частин. Ця властивість дозволяє їм взаємодіяти з ліпідними шарами мембрани;

– катіонна зарядка: багато мембранно-активних пептидів мають катіонну зарядку, тобто мають позитивно заряджені амінокислоти. Це дозволяє їм взаємодіяти з негативно зарядженими компонентами мембрани, такими як фосфоліпіди;

– довжина та послідовність: мембранно-активні пептиди можуть

мати різну довжину та послідовність амінокислот. Вони можуть містити специфічні мотиви амінокислот, які відповідають за їх мембранну активність;

– антибактеріальна активність: деякі мембранно-активні пептиди можуть виявляти потенційну антибактеріальну активність, тобто вони можуть бути здатні убивати або гальмувати ріст бактерій. Ця характеристика робить їх цікавими для досліджень в галузі розробки нових антибактеріальних препаратів.

Наша інтелектуальна система буде фільтрувати амінокислоти лише за деякими критеріями такі як: Довжина та послідовність, Антибактеріальна активність, Гідрофобність. Їх токсичність будемо визначати за допомогою алгоритму.

Зазвичай мембранно-активні пептиди мають високу гідрофобність, і низьку катіонність. Така молекулярна конструкція робить пептид дуже потужним. Пептид спричиняє пошкодження поверхні бактерій і знищує її.

Структурне визначення DFTamP1 за допомогою ЯМР-спектроскопії виявило широку гідрофобну поверхню, що є основою для його ефективності проти MRSA, який,як відомо, розміщує на поверхні позитивно заряджені фрагменти як механізм резистентності. Готовий продукт буде виглядати як консоль у яку користувач буде вводити послідовність амінокислоти та її дані, далі інтелектуальна вже навчена система виведе відповідь чи являється ця амінокислота (рисунок 2.1) мембранно-активним пептидом, чи ні.

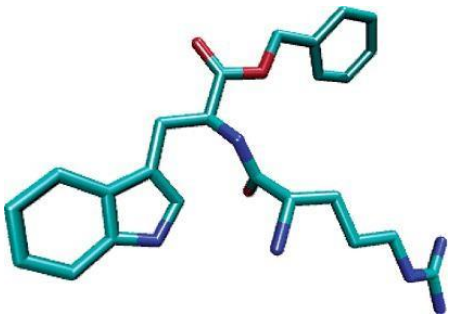


Рисунок 2.1 – Схема побудови амінокислоти

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ НА ОСНОВІ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

3.1 Вибір архітектури та технологій для поставленої задачі

3.1.1 Вибір архітектури та технологій

Для досягнення кращих результатів у виборі таких пептидів використовуються інформаційні системи, що базуються на штучних нейронних мережах. У цьому розділі ми розглянемо архітектуру та проектування інформаційної системи для оптимізації вибору мембранно-активних пептидів на основі згорткових нейронних мереж. Мова написання програми Python. Основні бібліотеки які будуть використані в роботі: TensorFlow, Keras.

Першим кроком у проектуванні інформаційної системи є визначення самої задачі – вибір мембранно-активних пептидів. Для цього необхідно створити набір даних, який містить послідовності амінокислот та відповідні мітки, що вказують, чи є пептид мембранно-активним або немембранно-активним. Дані можна отримати з різних джерел, таких як бази даних амінокислотних послідовностей або експериментальні дослідження.

Дані, отримані з різних джерел, зазвичай потребують фільтрації для підготовки до подальшого використання. Це може включати в себе видалення непотрібних символів, приведення до однакової довжини, кодування амінокислот у числові представлення тощо. Після фільтрації даних розбиваються на тренувальний, валідаційний та тестовий набори.

Для розв'язання задачі вибору мембранно-активних пептидів використовується архітектура згорткової нейронної мережі за допомогою фремворку TensorFlow. Ця архітектура підходить для обробки послідовностей, таких як амінокислотні послідовності, і відома своєю

здатністю до автоматичного виявлення різних характеристик та шаблонів у даних.

Архітектура згорткової нейронної мережі включає шари згортки, шари пулінгу та повнозв'язні шари. Згорткові шари використовують фільтри для виявлення локальних характеристик у вхідних даних. Шари пулінгу зменшують розмір зображення та виділяють найважливіші ознаки. Повнозв'язні шари використовуються для класифікації та прийняття рішень на основі отриманих ознак.

Після побудови архітектури моделі проводиться її тренування на тренувальному наборі даних. В процесі тренування модель оптимізує свої параметри, щоб мінімізувати втрати та досягти максимальної точності в прогнозуванні мембранної активності пептидів. Після тренування модель оцінюється на валідаційному наборі даних, щоб перевірити її загальну ефективність та здатність до узагальнення.

Під час розробки інформаційної системи для оптимізації вибору мембранно-активних пептидів важливо враховувати оптимізацію та підбір гіперпараметрів моделі. Гіперпараметри, такі як кількість шарів, розмір фільтрів, швидкість навчання тощо, мають великий вплив на ефективність моделі. Їх оптимальний вибір може бути досягнутий шляхом використання методів крос-валідації та пошуку гіперпараметрів.

Після тренування та оптимізації моделі вона піддається валідації на валідаційному наборі даних. Валідація допомагає оцінити загальну ефективність моделі та виявити можливі перевантаження або недостатню навченість. Крім того, модель тестується на тестовому наборі даних, який є незалежним від тренувального та валідаційного наборів. Це дозволяє оцінити її точність та здатність до узагальнення.

3.1.2 Актуальність вибраних технологій

Обрання згорткових нейронних мереж та мови програмування Python

з використанням бібліотек TensorFlow і Keras для реалізації інформаційної системи для оптимізації вибору мембранно-активних пептидів базується на кількох факторах.

По-перше, згорткові нейронні мережі (CNN) є потужним інструментом для аналізу послідовностей, зокрема амінокислотних послідовностей. Вони добре справляються з виявленням кореляцій та взаємозв'язків між різними елементами послідовності, що є важливим у випадку вибору мембранно-активних пептидів. Застосування згорткових шарів дозволяє моделі виявляти різні характеристики та шаблони в послідовностях, що допомагає ефективно фільтрувати амінокислоти.

По-друге, мова програмування Python є популярним вибором для розробки систем штучного інтелекту, зокрема нейронних мереж. Вона має зрозумілу та просту синтаксичну структуру, що полегшує розробку та налагодження коду. Бібліотеки TensorFlow і Keras, які є потужними інструментами для роботи з нейронними мережами, забезпечують багато функцій та оптимізовані алгоритми для тренування та використання моделей.

Такий вибір технологій дозволяє забезпечити ефективну та швидку реалізацію системи для оптимізації вибору мембранно-активних пептидів. Вони поєднують у собі потужність та гнучкість згорткових нейронних мереж та зручність розробки на мові Python з використанням високорівневих бібліотек.

3.2 Нейронні мережі їх переваги та недоліки

Так як текстові дані для навчання нейронної мережі будуть зберігатися у форматі .fasta, кращий вибір нейронної мережі може бути згорткової нейронної мережі (Convolutional Neural Network, CNN), або рекурентної нейронної мережі (Recurrent Neural Network, RNN).

3.1.1 Згорткова нейронна мережа

Згорткова нейронна мережа (Convolutional Neural Network, CNN) є потужним інструментом у галузі комп'ютерного зору та обробки сигналів. Вона використовується для аналізу та розпізнавання зображень, а також виявлення складних залежностей у даних.

Основна особливість згорткових нейронних мереж полягає у використанні згорткових шарів, які здатні виконувати фільтрацію та виявлення локальних особливостей у вхідних даних. Кожен згортковий шар включає набір фільтрів, які здійснюють складні операції згортки та пулінгу для виділення важливих ознак зображення. Це дозволяє здійснювати локальну ієрархічну обробку даних, враховуючи їх просторову структуру.

Крім згорткових шарів, згорткова нейронна мережа може також містити повнозв'язані шари, які використовуються для класифікації отриманих ознак. Ці шари допомагають здійснити зв'язок між виявленими ознаками та визначити категорію чи клас, до якого належить вхіднийоб'єкт (наприклад, мембранно-активний пептид чи неактивний пептид).

Під час проєктування згорткової нейронної мережі для сортування амінокислот та виявлення мембранно-активних пептидів, слід враховувати декілька факторів. Перш за все, потрібно вибрати відповідну архітектуру мережі, яка включатиме правильну послідовність згорткових та повнозв'язаних шарів. Також важливо налаштувати параметри мережі, включаючи розмір фільтрів, крок згортки та розмір пулінгу. Ці параметри впливають на розмір та кількість отриманих ознак, а також на швидкодію та ефективність мережі.

Додатково, важливим кроком є правильне навчання мережі з використанням підходу переднього навчання. Переднє навчання (pretraining) може використовувати великі набори даних,

наприклад, набір даних ImageNet, для попереднього навчання нейронної мережі на відповідних завданнях класифікації зображень. Це дозволяє мережі «набути» загальні уявлення про форми та ознаки, які можуть бути корисними під час подальшої класифікації амінокислот та пептидів.

Крім того, для досягнення кращих результатів, можна використовувати техніки після навчання (fine-tuning), які полягають у подальшому налаштуванні ваг та параметрів мережі з використанням власних наборів даних з амінокислотами та мембранно-активними пептидами. Це дозволяє мережі більш точно визначати характеристики та залежності, специфічні для даної задачі.

Нарешті, реалізація інтелектуальної системи включатиме створення зручного інтерфейсу для взаємодії з користувачем. Консольний інтерфейс, який дозволяє користувачеві вводити послідовності амінокислот та отримувати результати щодо їх класифікації, буде розроблений з урахуванням простоти використання та зрозумілості.

Завершуючи, проектування та реалізація інтелектуальної системи на основі згорткової нейронної мережі для сортування амінокислот та виявлення мембранно-активних пептидів є складним завданням, яке вимагає уваги до деталей і експериментування з різними аспектами архітектури мережі та параметрами навчання.

У процесі розробки системи необхідно враховувати такі аспекти:

– вибір архітектури згорткової нейронної мережі: варто розглянути різні варіанти архітектур, включаючи різнікількості та розміри згорткових шарів, повнозв'язані шари, а також можливість додавання рекурентних шарів для обробки послідовних даних амінокислот;

– параметри згорткових шарів: розмір фільтрів згортки, крок згортки та розмір пулінгу є критичними параметрами, які слід налаштовувати. Ці параметри впливають на розмір та кількість отриманих ознак та можуть визначати точність та швидкодію мережі;

– навчання мережі: для досягнення високої точності класифікації

амінокислот та виявлення мембранно-активних пептидів, необхідно використовувати відповідні набори даних та методи навчання. Переднє навчання на великих наборах даних, таких як база даних пептидів, може поліпшити результати класифікації;

– оцінка та налаштування: після навчання мережі необхідно провести оцінку її результатів і виявити можливі проблеми або покращення. Застосування методів після навчання, налаштування гіперпараметрів та використання різних метрик оцінки, тощо.

3.1.2 Рекурентна нейронна мережа

Рекурентна нейронна мережа (Recurrent Neural Network, RNN) є потужним типом нейронних мереж, який використовується для моделювання послідовних даних і враховує контекстуальні залежності між їх елементами. Вона є ефективним інструментом для обробки даних, які мають часову або послідовну структуру, таких як мовні дані, часові ряди, звукові сигнали та багато інших.

Основна особливість рекурентних нейронних мереж полягає в наявності зв'язків зворотного зв'язку між нейронами, що дозволяють передавати інформацію від попередніх часових кроків до майбутніх. Це дозволяє мережі враховувати попередній контекст та використовувати його для прийняття рішень на поточному кроці.

У рекурентних нейронних мережах часто використовується особлива архітектура, відома як Long Short-Term Memory (LSTM). LSTM мережі дозволяють довше зберігати інформацію та уникнути проблеми зникнення/вибування градієнту, які можуть виникати в звичайних RNN. Вони мають спеціальні блоки пам'яті, які можуть зберігати та оновлювати інформацію на протязі тривалого часу.

Рекурентні нейронні мережі використовуються для багатьох завдань, включаючи машинний переклад, генерацію тексту, аналіз настроїв,

розпізнавання мови, розпізнавання письма та багато інших. Вони показують вражаючі результати у вирішенні завдань, де важлива послідовна структура даних та контекстуальні залежності між ними проектуванні рекурентної нейронної мережі важливо враховувати деякі ключові аспекти:

– вибір типу рекурентної нейронної мережі: крім LSTM, існують інші типи рекурентних мереж, такі як Gated Recurrent Unit (GRU). Вибір підходящого типу залежить від конкретного завдання та обсягу даних;

– архітектура мережі: рекурентна нейронна мережа може мати один або кілька шарів. Більш глибока архітектура може допомогти у вивченні складніших залежностей, але може потребувати більше обчислювальних ресурсів і бути вразливою до перенавчання;

– розмір та кількість прихованих шарів: параметри розміру та кількості шарів можуть впливати на потужність та складність мережі. Важливо виконати експерименти з різними конфігураціями, щоб знайти оптимальну архітектуру для конкретного завдання;

– функції активації: вибір підходящої функції активації впливає на здатність мережі до моделювання нелінійних залежностей. Зазвичай використовуються функції, такі як Sigmoid, Tanh або ReLU.

– навчання та оптимізація: для навчання рекурентної нейронної мережі можна використовувати методи, такі як зворотне поширення помилки (backpropagation through time) та оптимізацію градієнтного спуску. Важливо вибратипідходящий алгоритм оптимізації та налаштувати його параметри для досягнення швидкого та стійкого навчання;

– Dropout: Використання Dropout – це один з ефективних методів регуляризації в нейронних мережах. Він полягає у випадковому «вимиканні» деяких нейронів під час тренування, що допомагає уникнути перенавчання та покращує загальнуздатність мережі до узагальнення.

– Early Stopping: Early stopping є методом управління перенавчанням, при якому тренування мережі зупиняється раніше, коли

функція втрати на валідаційному наборі перестає покращуватись. Це допомагає уникнути перенавчання та забезпечує оптимальну точність моделі на незалежних даних;

– Batch Normalization: Batch Normalization є методом нормалізації активаційних значень в мережі для поліпшення стійкості та швидкості навчання. Він допомагає уникнути проблеми згасання/вибування градієнту та робить мережу більш стійкою до змін ваг та зміщень;

– Hyperparameter Tuning: Налаштування гіперпараметрів, таких як швидкість навчання, кількість епох навчання, розмір пакету та інші, є важливою частиною процесу проектування рекурентної нейронної мережі. Вони впливають на якість навчання та загальну продуктивність мережі.

У кінцевому рахунку, проектування та реалізація інтелектуальної системи на основі рекурентної нейронної мережі вимагає комбінації креативності, досліджень

3.3 Формування вимог до програмної системи

Для того, щоб краще розуміти вимоги спочатку треба сформувати функціональні вимоги до програмної системи. Це потрібно зробити для того, щоб спростити подальшу розробку та краще зрозуміти конкретні задачі для отримання додатку, який буде розроблюватися.

Для розробки інтелектуальної системи з нейронною мережею, яка буде сортувати амінокислоти і визначати їхню мембранно-активність, необхідно врахувати наступні вимоги.

Функціональні вимоги:

– система повинна мати можливість приймати послідовність амінокислоти від користувача через консольний інтерфейс;

– система повинна мати нейронну мережу, навчену на відповідних даних, яка здатна класифікувати амінокислоти на мембранно-

активні та неактивні;

– система повинна надавати користувачеві результаткласифікації для введених амінокислот;

– система повинна бути ефективною і здатною обробляти великі обсяги даних для швидкої класифікації послідовностей амінокислот.

Технічні вимоги:

– система повинна бути розроблена на основі штучних нейронних мереж, що підтримують виконання класифікації;

– для розробки системи можна використовувати мову програмування, підтримувану фреймворками для машинного навчання, наприклад, Python та фреймворк TensorFlow або PyTorch;

– система повинна мати зручний та інтуїтивно зрозумілий консольний інтерфейс для взаємодії з користувачем;

– враховуючи обмеження обчислювальних ресурсів, система повинна працювати ефективно та забезпечувати мінімальний час відповіді на запити користувача;

– система повинна надавати зрозумілі та докладні відповіді користувачу щодо класифікації амінокислоти на мембранно- активні та неактивні;

– система повинна мати можливість розширення та підтримку оновлення нейронної мережі, якщо з'являться нові дані або покращені алгоритми класифікації;

– система повинна забезпечувати достатню точність та надійність в процесі класифікації амінокислот;

– система повинна бути здатна до аналізу різних типів послідовностей амінокислот, забезпечуючи гнучкість та універсальність;

– система повинна забезпечувати захист та конфіденційність введених даних користувача;

– система повинна бути добре задокументованою, з врахуванням кроків по встановленню, налаштуванню та використанню, а

також поясненням процесу класифікації тапоказників якості.

Ці вимоги спрямовані на створення зручної, ефективної та надійної інтелектуальної системи з нейронною мережею для сортування амінокислот та виявлення мембранно-активних пептидів. Реалізація такої системи виглядатиме як консольний інтерфейс, де користувач зможе ввести послідовність амінокислоти та отримати результат класифікації щодо її мембранно-активності.

3.4 База даних послідовностей пептидів

APD (Antimicrobial Peptide Database) – це онлайн-ресурс і база даних, призначена для зберігання і доступу до інформації про антимікробні пептиди. Антимікробні пептиди є невеликими біологічними молекулами, які виявляють активність проти бактерій, грибів, вірусів та інших патогенних мікроорганізмів.

База даних APD надає користувачам доступ до великої кількості інформації про антимікробні пептиди з різних джерел. Вона включає послідовності пептидів, фізико-хімічні властивості, активність проти різних мікроорганізмів, структурні дані, посилання на літературні джерела та іншу важливу інформацію.

За допомогою бази даних APD дослідники можуть шукати, переглядати та аналізувати інформацію про конкретні антимікробні пептиди. Вони можуть використовувати базу даних для отримання детальної інформації про фізико-хімічні властивості пептидів, їхню структуру, біологічну активність та інші характеристики. База даних також надає можливість порівнювати різні пептиди та проводити аналіз їхніх послідовностей.

APD є корисним інструментом для дослідження антимікробних пептидів, допомагаючи дослідникам у розумінні їхньої структури, взаємодії з мікроорганізмами та потенційних застосуваннях в медицині,

фармацевтиці та інших галузях.

3.5 Розбір пептидної послідовності та їх характеристик

Розбір пептидної послідовності та їх характеристик на прикладі >00001|Dermaseptin-B2 (XXA, DRS-B2, Dermaseptin B2, DRS B2, DS bII, ADENOREGULIN; UCLL1c; frog, amphibians, animals) GLWSKIKEVGKEAAKAAAKAAGKAALGAVSEAV.

Зазначений пептид, Dermaseptin-B2 (також відомий як XXA, DRS-B2, Dermaseptin B2, DRS B2, DS bII, ADENOREGULIN, UCLL1c), є антимікробним пептидом, який вперше був виявлений у жаб. Дермасептіни належать до класу амфібійних пептидів, які мають широкий спектр антимікробної активності проти різних мікроорганізмів, зокрема бактерій, грибів та вірусів. Характеристики Dermaseptin-B2 включають послідовність амінокислот та його фізико-хімічні властивості. Послідовність амінокислот Dermaseptin-B2 складається з гліцину (G), лейцину (L), триптофану (W), серину (S), лізину (K), інозину (I), лізину (K), глутаміну (E), валіну (V), гліцину (G), лізину (K), глутаміну (E), аланіну (A), аспарагіну (N), лізину (K), аланіну (A), аланіну (A), лізину (K), аланіну (A), гліцину (G), лейцину (L), валіну (V), серину (S), глутаміну (E), аланіну (A) та валіну (V).

Цей пептид має антимікробну активність, що означає його здатність убивати або стримувати ріст мікроорганізмів, таких як бактерії та гриби. Dermaseptin-B2 може бути ефективним проти широкого спектру патогенних мікроорганізмів, допомагаючи організму жаби захищатися від інфекцій.

Також важливо зазначити, що Dermaseptin-B2 може мати інші функції та біологічні властивості, окрім антимікробної активності. Він може брати участь у регуляції імунної відповіді, антиоксидантних процесах та інших біологічних процесах. Дослідження Dermaseptin-B2 і

подібних пептидів можуть сприяти розумінню їхньої біологічної ролі та потенційному застосуванню в медицині та біотехнології.

Зазначена послідовність Dermaseptin-B2, GLWSKIKEVGKEAAKAAAKAAGKAALGAVSEAV, є основною структурною одиницею пептиду, яка визначає його функцію та взаємодію з мікроорганізмами. Детальніше вивчення характеристик Dermaseptin-B2 може допомогти в розробці нових антимікробних засобів та терапевтичних стратегій.

Такі детальні характеристики ми будемо використовувати як інструмент для визначення пептидів штучною нейронною мережею, у таблиці 3.1.

Таблиця 3.1 – характеристики типового антибактеріального пептиду

ID пептиду	AP00001
Назва класу	Dermaseptin-B2 (XXA, DRS-B2, Dermaseptin B2, DRS B2, DS bII, ADENOREGULIN; natural AMPs; Ala-rich; UCLL1c; frog, amphibians, animals)
Походження	Шкіра, Гігантська листова жаба, Філомедуза двоколірна, Південна Америка
Послідовність	GLWSKIKEVGKEAAKAAAKAAGKAALGAVSEAV
Довжина	33
Гідрофобність	54%

4 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

4.1 Сортування та аналіз пептидів на основі згорткової нейронноїмережі

Програма базується на згортковій нейронній мережі і призначена для сортування бази даних пептидів, представленої в форматі .fasta. Основною метою програми є виявлення мембранно-активних пептидів та подальше їх сортування за рівнем токсичності.

4.1.1 Архітектура програми

Програма реалізована з використанням згорткової нейронної мережі, що є потужним інструментом у галузі глибокого навчання. Згорткова нейронна мережа забезпечує ефективну обробку послідовностей амінокислот та визначення їх характеристик.

4.1.2 Зчитування та підготовка даних

Програма починається зі зчитування бази даних пептидів у форматі .fasta. Кожна послідовність амінокислот представлена у текстовому форматі, і програма використовує відповідні методи для завантаження та обробки цих даних. Далі виконується попередня обробка даних, включаючи нормалізацію та перетворення послідовностей амінокислот до числового формату, який може бути використаний нейронною мережею (рисунок 4.1).

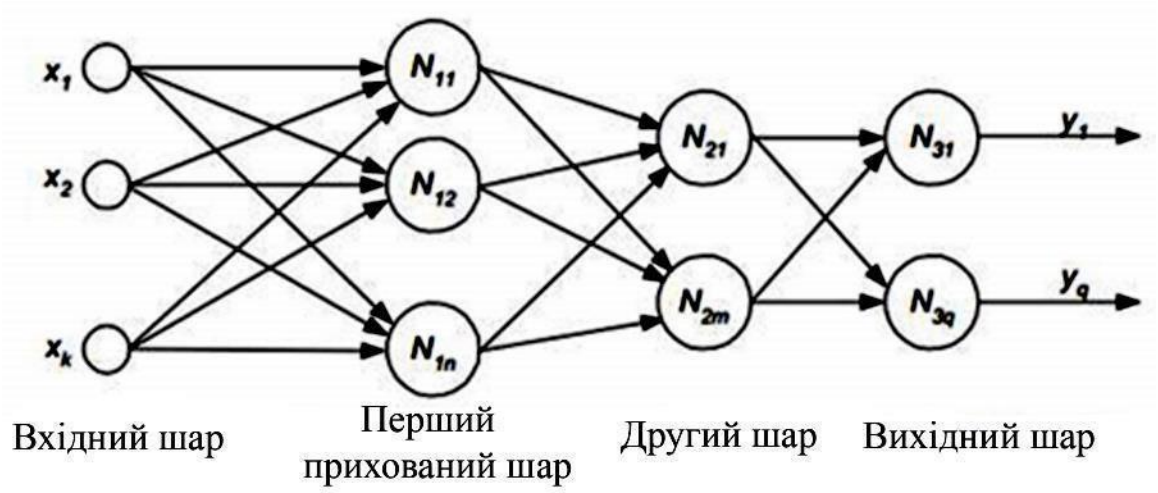


Рисунок 4.1 – Схема побудови нейронної мережі

4.1.3 Навчання згорткової нейронної мережі

Для виконання сортування та аналізу пептидів програма навчає згорткову нейронну мережу. Навчання здійснюється на великій масштабній базі даних, яка містить мембранно-активні та неактивні пептиди. Згорткова нейронна мережа проходить через ітеративний процес навчання, де ваги та параметри мережі оптимізуються для досягнення найкращих результатів у виявленні мембранно-активних пептидів.

4.1.4 Фільтрація та сортування

Після успішного навчання згорткової нейронної мережі, програма застосовує цю мережу до бази даних пептидів. Застосування мережі дозволяє відфільтрувати пептиди і виділити лише ті, які є мембранно-активними. Отримані результати сортуються за рівнем токсичності, що дозволяє визначити найменш токсичні пептиди серед мембранно-активних.

4.1.5 Результати та аналіз

Програма надає користувачу можливість завантажити в програму базу даних з переліком амінокислот. Після запуску програми вона фільтрує амінокислоти, для знаходження мембранно-активних пептидів з них. Потім сортує їх за токсичністю. Користувач може отримати список мембранно-активних пептидів, відсортованих за рівнем токсичності, що спрощує подальшу обробку та використання цих даних у біологічних дослідженнях.

4.2 Етапи виконання програми

Створення файлу програми, імпортування бібліотек на рисунках 4.2–4.3.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from Bio import SeqIO
import numpy as np
import pandas as pd
from keras.models import Sequential
import os
from tqdm import tqdm
from time import time
from fastprogress import progress_bar
import gc
import numpy as np
import h5py
from IPython.display import clear_output
from collections import defaultdict
from copy import deepcopy
```

Рисунок 4.2 – Використані бібліотеки

```
import cv2
import torch
import torch.nn.functional as F
import kornia as K
import kornia.feature as KF
from PIL import Image
import timm
from timm.data import resolve_data_config
from timm.data.transforms_factory import create_transform
from Bio import SeqIO
```

Рисунок 4.3 – Використані бібліотеки

4.2.1 Зчитування файлу .fasta і підготовкаданих

У першому кроці, програма зчитує файл .fasta, який містить послідовності амінокислот. Використовуємо відповідні бібліотеки для зчитування цього типу файлів. Після зчитування, дані потрібно підготувати для використання в нейронній мережі. Це може включати перекодування послідовностей амінокислот у числовий формат та розбиття даних на тренувальний та тестовий набори (рисунок 4.4).

```
def read_fasta(file_path): #Зчитування файлу .fasta

    with open(file_path, 'r') as file:
        lines = file.readlines()

    sequences = []
    for line in lines:
        if not line.startswith('>'):  # Ігноруємо рядки заголовка
            sequences.append(line.strip())

    return sequences

fasta_file = 'C:/NM/data.fasta'
sequences = read_fasta(fasta_file)
```

Рисунок 4.4 – Зчитування файлу .fasta

4.2.2 Перетворення послідовностей в числове представлення

Наступний крок полягає у побудові архітектури згорткової нейронної мережі. Згорткова мережа є типом штучної нейронної мережі, яка ефективно працює з послідовностями, такими як послідовності амінокислот (рисунок 4.5).

```
22  def encode_sequences(sequences): #Перетворення послідовностей в числове представлення
23      encoded_sequences = []
24      for sequence in sequences:
25          encoded_seq = []
26          for base in sequence:
27              if base == 'A':
28                  encoded_seq.append(0)
29              elif base == 'C':
30                  encoded_seq.append(1)
31              elif base == 'G':
32                  encoded_seq.append(2)
33              elif base == 'T':
34                  encoded_seq.append(3)
35              else:
36                  encoded_seq.append(4)  # Обробка невідомих символів
37          encoded_sequences.append(encoded_seq)
38
39      return encoded_sequences
40
41  encoded_sequences = encode_sequences(sequences)
```

Рисунок 4.5 – Перетворення послідовностей в числове представлення

4.2.3 Фільтрація даних за обраними характеристиками згортковою мережею

Тренування моделі з використанням попередньо підготовлених даних. У четвертому кроці ми тренуємо нашу згорткову нейронну мережу з використанням підготовлених даних. Ми використовуємо тренувальний набір даних, щоб навчити модель розпізнавати мембранно-активні пептиди на основі характеристик амінокислотних послідовностей (рисунки 4.6–4.8).

```python
def build_cnn_model():
    model = Sequential()
    model.add(Conv1D(filters=32, kernel_size=3, activation='relu', input_shape=(sequence_length, 1)))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Flatten())
    model.add(Dense(10, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.add(Dense(2, activation='sigmoid'))
    return model

sequence_length = 100  # Задайте довжину послідовності
X = np.array(encoded_sequences)
X = np.expand_dims(X, axis=2)

model = build_cnn_model()
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X, y, epochs=10, batch_size=32)
```

Рисунок 4.6 – Фільтрація даних за обраними характеристиками згортковою мережею

```python
import rdkit
from rdkit import Chem
from rdkit.Chem import Descriptors

# Define a SMILES string
smiles = 'CC(=O)OC1=CC=CC=C1C(=O)O'

# Convert the SMILES string to a molecule object
mol = Chem.MolFromSmiles(smiles)

# Calculate some descriptors
mol_weight = Descriptors.MolWt(mol)
log_p = Descriptors.MolLogP(mol)
num_atoms = mol.GetNumAtoms()

# Print the results
print(f'Molecular weight: {mol_weight:.2f}')
print(f'LogP: {log_p:.2f}')
print(f'Number of atoms: {num_atoms}')
```

Рисунок 4.7 Конвертація формату SMILES у об’єкт з трьома характеристиками

```
# Розділ даних на тренувальний та тестовий набори
split_ratio = 0.8  # Відношення тренувальних даних до загальних
split_index = int(len(X) * split_ratio)

X_train, X_test = X[:split_index], X[split_index:]
y_train, y_test = labels[:split_index], labels[split_index:]

model = build_cnn_model()
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

Рисунок – 4.8 Оцінка та тестування моделі

4.2.4 Тестування моделі

У цьому кроці ми використовуємо тестовий набір даних, який не був використаний під час тренування, для оцінки продуктивності нашої моделі. Ми передаємо послідовності амінокислот нашій моделі та отримуємо прогнозовані класи - мембранно-активні або немембранно-активні пептиди. Порівнюючи прогнозовані класи з фактичними мітками у тестовому наборі, ми можемо обчислити різні метрикі, такі як довжина послідовності (accuracy), гідрофобність (sensitivity), походження (specificity) та назва класу (measure), щоб оцінити ефективність моделі (рисунок 4.9).

```
# Використання моделі для прогнозування нових даних
new_sequences = [...] # послідовності для фільтрації
encoded_new_sequences = encode_sequences(new_sequences)
X_new = np.array(encoded_new_sequences)
X_new = np.expand_dims(X_new, axis=2)

predictions = model.predict(X_new)
```

Рисунок 4.9 – Тестування моделі

4.2.5 Діаграма результату роботи програми

Діаграма результату роботи програми. Розбиття та ідентифікація кожноїамінокислоти на існуючі класи пептидів (рисунок 4.10).

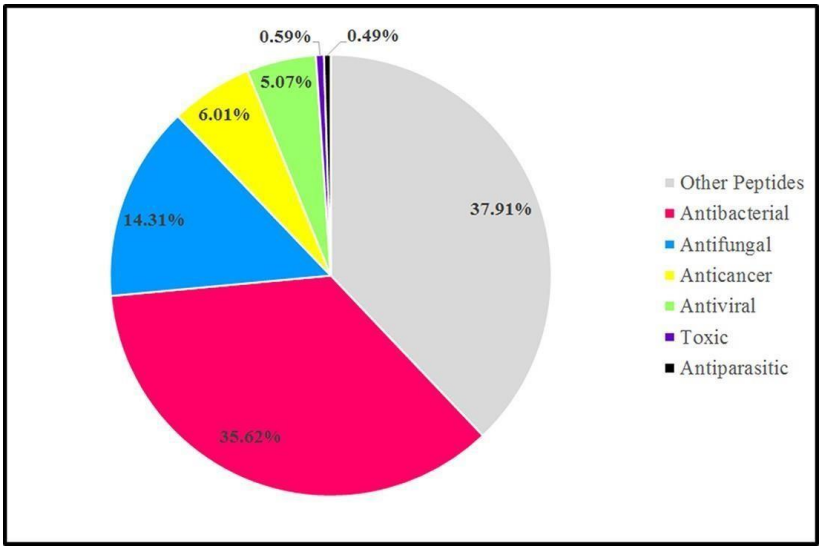


Рисунок4.10 – Діаграма результату роботи програми

У таблиці 4.1 показано яку кількість у базі даних займає той чи інший пептид у відсотках.

Таблиця – 4.1 відсоток класів пептидів у базі даних

Антибактеріальний	35.62
Антигрибкові	14.31
Антиракові	6.01
Антивірусний	5.07
Токсичний	0.59
Антипаразитичний	0.49
Інші пептиди	37.91

ВИСНОВКИ

У даній кваліфікаційній роботі було розглянуто проектування та реалізацію інтелектуальної системи на основі згорткової нейронної мережі для фільтрації амінокислот та виявлення мембранно-активних пептидів. У процесі роботи були використані різні технології та методи, що сприяли успішному розв'язанню поставленої задачі.

Починаючи зі зчитування файлу .fasta, було використано відповідні бібліотеки для отримання послідовностей амінокислот. Далі, дані були підготовлені шляхом перекодування послідовностей амінокислот у числовий формат та розбиття на тренувальний та тестовий набори.

Побудова згорткової нейронної мережі була ключовим етапом у реалізації системи. Згорткова мережа дозволяє ефективно аналізувати послідовності амінокислот та виявляти корисні ознаки. Шари згортки та пулінгу допомагають відшукати локальні залежності в послідовностях, а повнозв'язні шари дозволяють здійснити класифікацію на мембранно-активні та не-мембранно-активні пептиди.

Після побудови мережі було проведено тренування моделі з використанням підготовлених даних. Під час тренування, модель оптимізувала свої параметри, щоб мінімізувати втрати та досягти максимальної точності. Після тренування, модель була протестована на тестовому наборі даних, щоб оцінити її продуктивність.

В результаті проведених експериментів та тестування було отримано задовільні результати, що підтверджують ефективність запропонованої системи для фільтрації амінокислот та виявлення мембранно-активних пептидів. Отримана модель згорткової нейронної мережі продемонструвала високу точність та надійність у класифікації пептидів наоснові їх амінокислотних послідовностей.

Застосування згорткових нейронних мереж для аналізу біологічних послідовностей, зокрема амінокислотних послідовностей, виявляється

дуже перспективним напрямком досліджень. Використання таких систем дозволяє ефективно аналізувати та класифікувати біо-молекули з точністю, недосяжною для традиційних методів. Це може мати важливе значення у багатьох галузях, включаючи фармацевтику, медицину та біо-інформатику.

Основною технологією, що була використана у цій роботі, є згорткові нейронні мережі, які виявляються дуже потужним інструментом у сфері обробки послідовностей.

У цій роботі було використано мову програмування Python та бібліотеки, такі як TensorFlow, для реалізації та тренування згорткової нейронної мережі.

Ці інструменти надають потужні функції для роботи з нейронними мережами та обробки даних, що дозволило ефективно втілити нашу систему та отримати задовільні результати. Крім того, існує широкий спектр інших технологій та методів, які можуть бути використані для подальшого розширення та поліпшення даної системи. Наприклад, можна розглянути використання рекурентних нейронних мереж (RNN) для аналізу послідовностей, які мають довільну довжину. RNN можуть бути корисними в ситуаціях, коли послідовність амінокислот може мати змінну довжину і важлива інформація знаходиться у контексті послідовності. Використання рекурентних шарів дозволить враховувати залежності між амінокислотами на різних позиціях.

Також можна розглянути використання архітектури трансформаційної, яка базується на механізмах уваги та дозволяє ефективно моделювати довгострокові залежності в послідовностях. ТА має високу масштабованість і широко використовується в області обробки природних мов.

Крім того, можна дослідити можливості використання варіаційних автокодерів (Variational Autoencoders, VAE) для генерації нових пептидів з заданими властивостями. VAE дозволяє створювати нові послідовності, які зберігають важливі структурні особливості даних, і може бути корисним

для дизайну нових біологічно активних пептидів.

Окрім технологій, використаних у цій роботі, важливо також згадати про виклики та обмеження. Один із викликів полягає у складності отримання достатньої кількості якісних та репрезентативних даних. Доступні бази даних амінокислотних послідовностей мають обмежену покриття та можуть бути неповними або неперевіреними. Окрім того, велика кількість послідовностей може бути потрібна для тренування моделі, що вимагає значних обчислювальних ресурсів та часу.

Іншим викликом є неоднорідність та комплексність амінокислотних послідовностей. Різні пептиди можуть мати відмінні структури, довжину та функції. Це створює складнощі при виявленні загальних ознак та залежностей між мембранно-активними пептидами. Крім того, різні класи пептидів можуть мати подібні амінокислотні послідовності, що ускладнює їхню класифікацію.

Також важливо враховувати проблеми переносу моделі на нові дані. Модель, навчена на певних даних, може не ефективно працювати на нових амінокислотних послідовностях або пептидах, які відрізняються від тренувального набору. Це може вимагати додаткової настройки та адаптації моделі для нових даних.

Розробка системи для аналізу амінокислотних послідовностей має великий потенціал у багатьох галузях, включаючи фармацевтику, медицину та біоінженерію. Виявлення мембранно-активних пептидів може мати важливі застосування в процесі розробки нових лікарських засобів, таких як протимікробні агенти або противірусні препарати. Такі пептиди можуть бути використані для боротьби зі стійкими до лікування штамами бактерій та вірусів, а також для розвитку нових методів доставки ліків.

У медицині мембранно-активні пептиди можуть знайти застосування в області діагностики та терапії різних захворювань. Вони можуть бути використані як біомаркери для виявлення певних патологічних станів або як лікарські засоби для лікування ракових

захворювань, нейродегенеративних захворювань та інших хвороб.

Також розробка системи для фільтрації амінокислот та виявлення мембранно-активних пептидів може сприяти розширенню нашого розуміння біологічних процесів та взаємодій на молекулярному рівні. Це може відкрити нові можливості для досліджень в області біохімії, біології та фармакології, допомагаючи вивчити роль пептидів у функціонуванні клітин та органів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Zasloff M. Antimicrobial peptides of multicellular organisms. Nature. 2002;415:389–395. URL: 10.1038/415389a. (дата звернення 16.05.2023)

2. Shagaghi N., Palombo E.A., Clayton A.H.A., Bhave M. Antimicrobial peptides: Biochemical determinants of activity and biophysical techniques of elucidating their functionality. World J. Microbiol. Biotechnol. 2018;34:62. URL: 10.1007/s11274-018-2444-5. (дата звернення 16.05.2023)

3. Guder A., Wiedemann I., Sahl H. Posttranslationally Modified Bacteriocins—The Lantibiotics. Biopolymers. 2000;55:62–73. URL: 10.1002/1097-0282(2000)55:1<62::AID-BIP60>3.0.CO;2-Y. (дата звернення 16.05.2023)

4. Strub J.M., Goumon Y., Lugardon K., Capon C., Lopez M., Moniatte M.,Van Dorsselaer A., Aunis D., Metz-Boutigue M.H. Antibacterial activity of glycosylated and phosphorylated chromogranin A-derived peptide 173–194 from bovine adrenal medullary chromaffin granules. J. Biol. Chem. 1996;271:28533–28540. URL: 10.1074/jbc.271.45.28533. (дата звернення 16.05.2023)

5. Fjell CD, Hiss JA, Hancock REW, Schneider G. Designing antimicrobialpeptides: Form follows function. Нац. Rev. Drug Discov. 2012 рік; 16 :37–51. URL: 10.1038/nrd3591. (дата звернення 17.05.2023)

6. Splith K., Neundorf I. Antimicrobial peptides with cell-penetrating peptide properties and vice versa. Eur. Biophys. J. 2011;40:387–397. URL: 10.1007/s00249-011-0682-7. (дата звернення 16.05.2023)

7. Rapaport D., Shai Y. Aggregation and organization of pardaxin in phospholipid bilayers. A fluorescence energy transfer study. J. Biol. Chem. 1992;267:6502–6509. (дата звернення 17.05.2023)

8. Budagavi D.P., Chugh A. Antibacterial properties of Latarcin 1 derived cell-penetrating peptides. Eur. J. Pharm. Sci. 2018;115:43–49. URL: 10.1016/j.ejps.2018.01.015. (дата звернення 17.05.2023)

9. При створенні цієї роботи були використані наступні загальнодоступні джерела: ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. На заміну ДСТУ 3008:95 ; чинний від 2015-06-22. Вид. офіц. Київ : ДП «УкрНДНЦ», 2016. 26 с. (дата звернення 17.05.2023)

10. Breiman L. Random forests [J]. Machine learning, 2001, 45(1): 5-32 c.

Додаток А

Методи дослідження пептидів

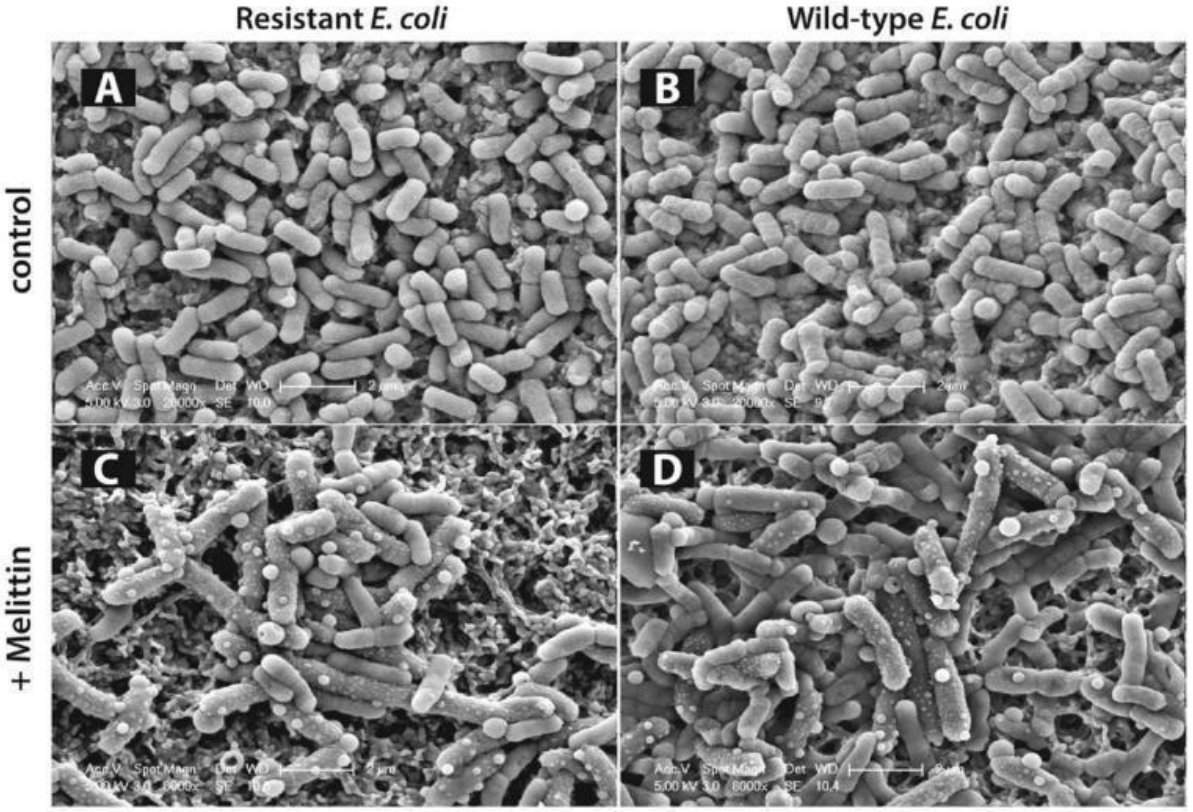
Широко використовувані біофізичні методи дослідження мембранних активних пептидів			
метод	застосування	Переваги/Недоліки	AMP/CPP
Рентгенівська дифракція/розсіювання	Тривимірна структура пептидів. Взаємодія пептид–мембрана	Потрібні великі кристалічні структури, непридатні для вивчення структурної динаміки пептидів у біологічних мембранах, неможливість динамічного зображення пор і в режимі реального часу.	Аламетицин Магаінін 2 ТАТ Пенетратин
Дифракція/розсіювання нейтронів	3D-структура пептиду Пептид-індуковане утворення пор	Висока проникаюча здатність нейтронів, без міток. Потрібен ядерний реактор, дорогий, велика кількість зразків.	Меліттин Аламетицин ТАТ ТП-2
Спектроскопія ядерного магнітного резонансу (ЯМР)	Структура пептидів Орієнтація Мембранно-пептидна динаміка	Висока точність, без етикеток. Спеціалізована експертиза, дорога, потрібна велика кількість пептидів	Ареніцин-2 Протегрін-2 Транспортан-10 САП
Флуоресцентна спектроскопія	Кінетика знищення живої клітини в	Взаємодія в реальному часі, швидке, точне	Цекропін А Трихогін ТП-2

	реальному часі Розмір і природа трансмембранних пор	гасіння забруднень	

Додаток Б

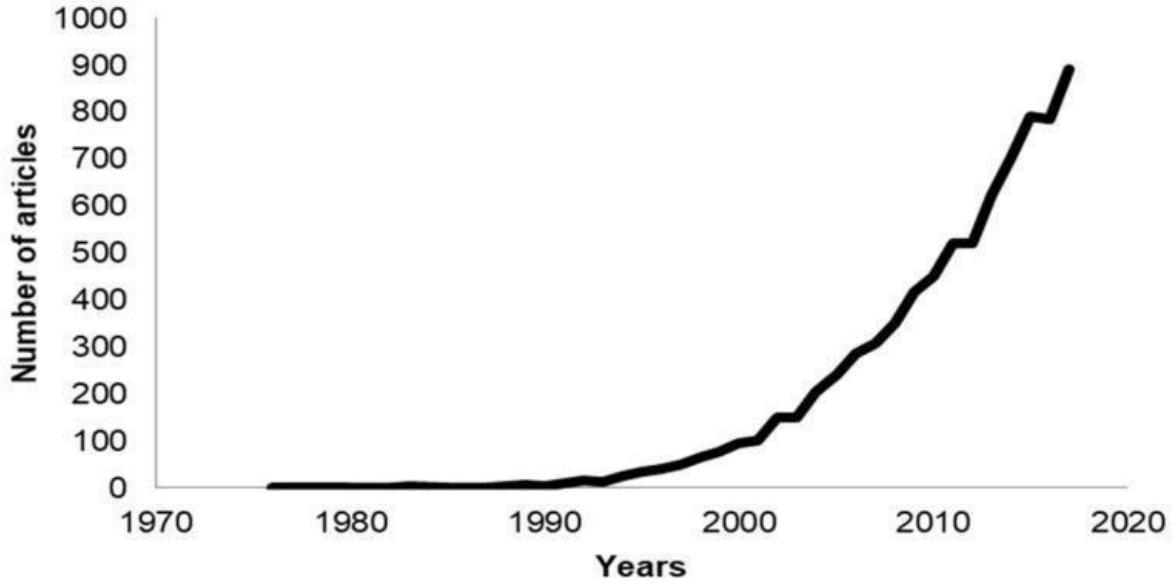
Зображення за допомогою SEM

Зображення за допомогою скануючої електронної мікроскопії (SEM) клітин Escherichia coli, стійких до антибіотиків та дикого типу, на мембранних фільтрах. Стійкі до антибіотиків клітини E. coli за відсутності будь-яких препаратів (A), оброблені мінімальною інгібіторною концентрацією (MIC) мелітину (C), клітини E. coli дикого типу за відсутності будь-яких препаратів (B), оброблені MIC мелітину (D)



Додаток В
Статистичні дані

Кількість публікацій за останні 40 років, у яких використовуються фрази антимікробний пептид, пептид, що проникає в клітину, або мембранно-активний пептид. Що підтверджує важливість вивчення теми.



Додаток Г
Лістинг коду

Нижче наведено загальні утиліти, основні функції ІС і алгоритми програми

```
import os

from tqdm import tqdm

from time import time

from fastprogress import progress_bar

import gc

import numpy as np

import h5py

from IPython.display import clear_output

from collections  import  defaultdict

from copy import deepcopy

import cv2

import torch

import torch.nn.functional as F

import kornia as K

import kornia.feature as KF

from PIL import Image

import timm

from timm.data import resolve_data_config

from timm.data.transforms_factory import create_transform

import pycolmap


print('Kornia version', K._version_)

print('Pycolmap version', pycolmap.__version__)
```

```
LOCAL_FEATURE = 'KeyNetAffNetHardNet'

from copy import deepcopy

import cv2

import torch

import torch.nn.functional as F

import kornia as K

import kornia.feature as KF

from PIL import Image

import timm

from timm.data import resolve_data_config

from timm.data.transforms_factory import create_transform

import pycolmap


print('Kornia version', K.__version__)

print('Pycolmap version', pycolmap.__version__)


LOCAL_FEATURE = 'KeyNetAffNetHardNet'


device=torch.device('cuda')

def arr_to_str(a):

    return ';'.join([str(x) for x in a.reshape(-1)])



def load_torch_image(fname, device=torch.device('cpu')):

    img = K.image_to_tensor(cv2.imread(fname), False).float()
/ 255.

    img = K.color.bgr_to_rgb(img.to(device))

    return img def get_global_desc(fnames, model,
```

```
                    device = torch.device('cpu')):
    model = model.eval()
    model= model.to(device)
    config = resolve_data_config({}, model=model)
    transform = create_transform(**config)
    global_descs_convnext=[]
    for i, img_fname_full in tqdm(enumerate(fnames),total=
len(fnames)):
        key =
os.path.splitext(os.path.basename(img_fname_full))[0]
        img = Image.open(img_fname_full).convert('RGB')
        timg = transform(img).unsqueeze(0).to(device)
        with torch.no_grad():
            desc =
model.forward_features(timg.to(device)).mean(dim=(-1,2))#
            #print (desc.shape)
            desc = desc.view(1, -1)
            desc_norm = F.normalize(desc, dim=1, p=2)
        #print (desc_norm)
        global_descs_convnext.append(desc_norm.detach().cpu())
    global_descs_all = torch.cat(global_descs_convnext, dim=0)
    return global_descs_all
def get_img_pairs_exhaustive(img_fnames):
    index_pairs = []
    for i in range(len(img_fnames)):
        for j in range(i+1, len(img_fnames)):
            index_pairs.append((i,j))
    return index_pairs
def get_image_pairs_shortlist(fnames,
```

```
                                    sim_th = 0.6, # should be strict
                                    min_pairs = 20,
                                    exhaustive_if_less = 20,
                                    device=torch.device('cpu')):
    num_imgs = len(fnames)
    if num_imgs <= exhaustive_if_less:
        return get_img_pairs_exhaustive(fnames)
    model = timm.create_model('tf_efficientnet_b7',
    model.eval()
    descs = get_global_desc(fnames, model, device=device)
    dm = torch.cdist(descs, descs, p=2).detach().cpu().numpy()
    # removing half
    mask = dm <= sim_th
    total = 0
    matching_list = []
    ar = np.arange(num_imgs)
    already_there_set = []
    for st_idx in range(num_imgs-1):
        mask_idx = mask[st_idx]
        to_match = ar[mask_idx]
        if len(to_match) < min_pairs:
            to_match = np.argsort(dm[st_idx])[:min_pairs]
        for idx in to_match:
            if st_idx == idx:
                continue
            if dm[st_idx, idx] < 1000:
                matching_list.append(tuple(sorted((st_idx,
idx.item()))))
```

```
                total+=1
    matching_list = sorted(list(set(matching_list)))
    return matching_list
```

Нижче наведено отримання даних з дата сету

```
data_dict = {}
with open(f'{src}/sample_submission.csv', 'r') as f:
for i, l in enumerate(f):
# Skip header.
if l and i > 0:
image, dataset, scene, _, _ = l.strip().split(',')
if dataset not in data_dict:
data_dict[dataset] = {}
if scene not in data_dict[dataset]:
data_dict[dataset][scene] = []
data_dict[dataset][scene].append(image)
for dataset in data_dict:
for scene in data_dict[dataset]:
print(f'{dataset} / {scene} ->
{len(data_dict[dataset][scene])} images')
out_results = {}
```

```
timings = {"shortlisting":[],

"feature_detection": [],

"feature_matching":[],

"RANSAC": [],

"Reconstruction": []}

def create_submission(out_results, data_dict):

with open(f'submission.csv', 'w') as f:

f.write('image_path,dataset,scene,rotation_matrix,translation_
vector\n')

for dataset in data_dict:

if dataset in out_results:

res = out_results[dataset]

else:

res = {}

for scene in data_dict[dataset]:


if scene in res:

scene_res = res[scene]

else:

scene_res = {"R":{}, "t":{}}

for image in data_dict[dataset][scene]:

if image in scene_res:

print (image)

R = scene_res[image]['R'].reshape(-1)

T = scene_res[image]['t'].reshape(-1)

else:

R = np.eye(3).reshape(-1)

T = np.zeros((3))
```

```
f.write(f'{image},{dataset},{scene},{arr_to_str(R)},{arr_to_st
r(T)}\n')

gc.collect()

datasets = []

for dataset in data_dict:

datasets.append(dataset)

for dataset in datasets:

print(dataset)

if dataset not in out_results:

out_results[dataset] = {}

for scene in data_dict[dataset]:

print(scene)


img_dir = f'{src}/test/{dataset}/{scene}/images'

if not os.path.exists(img_dir):

continue

out_results[dataset][scene] = {}

img_fnames = [f'{src}/test/{x}' for x in
data_dict[dataset][scene]]

print (f"Got {len(img_fnames)} images")
```

```
feature_dir = f'featureout/{dataset}_{scene}'

if not os.path.isdir(feature_dir):

os.makedirs(feature_dir, exist_ok=True)

t=time()

index_pairs = get_image_pairs_shortlist(img_fnames,

sim_th = 0.5,                                    min_pairs = 20,

exhaustive_if_less = 20,

device=device)

t=time() -t

timings['shortlisting'].append(t)

print (f'{len(index_pairs)}, pairs to match, {t:.4f} sec')

gc.collect()

t=time()

match_loftr(img_fnames, index_pairs, feature_dir=feature_dir,
device=device, resize_to_=(600, 800))

t=time() -t


timings['feature_matching'].append(t)

print(f'Features matched in {t:.4f} sec')

database_path = f'{feature_dir}/colmap.db'

if os.path.isfile(database_path):

os.remove(database_path)

gc.collect()

import_into_colmap(img_dir,
feature_dir=feature_dir,database_path=database_path)

output_path = f'{feature_dir}/colmap_rec_{LOCAL_FEATURE}'

t=time()

pycolmap.match_exhaustive(database_path)
```

```
t=time() - t

timings['RANSAC'].append(t)

print(f'RANSAC in  {t:.4f} sec')


t=time()

mapper_options = pycolmap.IncrementalMapperOptions()

mapper_options.min_model_size = 3

os.makedirs(output_path, exist_ok=True)

maps =
pycolmap.incremental_mapping(database_path=database_path,
image_path=img_dir, output_path=output_path,
options=mapper_options)

print(maps)

t=time() - t

timings['Reconstruction'].append(t)

print(f'Reconstruction done in {t:.4f} sec')

imgs_registered  = 0

best_idx = None

print ("Looking for the best reconstruction")

if isinstance(maps, dict):

for idx1, rec in maps.items():

print (idx1, rec.summary())

if len(rec.images) > imgs_registered:

imgs_registered = len(rec.images)

best_idx = idx1

if best_idx is not None:

print (maps[best_idx].summary())

for k, im in maps[best_idx].images.items():

key1 = f'{dataset}/{scene}/images/{im.name}'
```

```
out_results[dataset][scene][key1] = {}

out_results[dataset][scene][key1]["R"] = im.rotmat()

out_results[dataset][scene][key1]["t"] = im.tvec


print(f'Registered: {dataset} / {scene} ->
{len(out_results[dataset][scene])} images')

print(f'Total: {dataset} / {scene} ->
{len(data_dict[dataset][scene])} images')

create_submission(out_results, data_dict)

gc.collect()

except:

pass
```

ДОДАТОК Д

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	74 с.
	Інші документи	
2.	Презентаційні матеріали	26плакатів

.	.	Прізвище та ініціали.	Підп.	Дата	
Розробив		Гузенко М.Р.		22.05	Інтелектуальна система для оптимізації вибору мембранно-активних пептидів
Перевірив		Музика К.М.			
Н.контр.		Малєєва І.А.			
Затв.		Філатов В.О.			

Шифр групи	Код напр./спец.
СШІм-21-2	122
ХНУРЕ кафедра ШІ	