КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ВИСОКИХ ТЕХНОЛОГІЙ

Завідувач кафедри молекулярної біотехнології та біоінформатики
доцент Нипорко Олексій Юрійович
Протокол №____ засідання кафедри
від “____” _____20___ р.

ВИКОРИСТАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ТОКСИЧНИХ ПЕПТИДІВ НА ОСНОВІ ЇХ АМІНОКИСЛОТНОЇ ПОСЛІДОВНОСТІ

Випускна кваліфікаційна робота магістра
студента спеціальності 091
Біологія
ОП «Біоінформатика та структурна біологія»
Шматкова Максима Валерійовича

Науковий керівник від кафедри
доцент кафедри молекулярної
біотехнології та біоінформатики
к.ф.-м.н. **Войтешенко Іван Сергійович**

Робота виконана в лабораторії біоінформатики
FFTB Загребського університету
під керівництвом **Antonio Starčević**, PhD

Оцінка захисту роботи

_____

Київ-2024

Анотація

Шматков М. В. Використання методів машинного навчання для класифікації токсичних пептидів на основі їх амінокислотної послідовності. — Випускна кваліфікаційна робота магістра за спеціальністю 091 Біологія ОП «Біоінформатика та структурна біологія».

У роботі розроблено модель для класифікації амінокислотних послідовностей на токсичні та нетоксичні за допомогою імплементації інструменту Doc2Vec та штучної нейронної мережі. Встановлено, що отримана модель має здатність доволі точно класифікувати амінокислотні послідовності, має гарну специфічність та чутливість. Отримані результати можуть бути використані для покращення розробленої моделі та визначення нових токсичних білкових послідовностей, що можуть стати основою для нових лікарських препаратів білкової природи.

Ключові слова: Doc2Vec, нейронні мережі, токсини, пептиди.

Abstract

Shmatkov M. V. Using machine learning methods to classify toxic peptides based on their amino acid sequence - Master’s thesis in the specialty 091 Biology, educational program ”Bioinformatics and Structural Biology”.

In this work, a model for classifying amino acid sequences into toxic and non-toxic was developed by implementing the Doc2Vec tool and an artificial neural network. It was found that the resulting model has the ability to classify amino acid sequences quite accurately, has good specificity and sensitivity. The obtained results can be used to improve the developed model and identify new toxic protein sequences that can become the basis for new protein-based drugs.

Keywords: Doc2Vec, neural networks, toxins, peptides.

Зміст

Вступ

Серед усіх речовин, отрути є одними з найбільш біологічно активних. Зазвичай, основну їх частину складають пептиди чи білки, що записані в геномі організмів, які її продукують [1]. Тиск природного добору сприяв еволюції відповідних генів і призвів до того, що в більшості випадків застосування навіть невеликої кількості отруйної речовини достатньо для досягнення сильного токсичного ефекту на чутливий до неї організм. Та окрім небезпеки для життя та здоров'я людини, отрути є також одними з історично перших засобів лікування різноманітних хвороб та досі залишаються важливим джерелом для розробки нових препаратів [2].

Різноманітність механізмів дії, будови та мішеней отруйних пептидів у комбінації з високою специфічністю дозволяє використовувати їх як потенційних прототипів лікарських препаратів проти широкого спектру хвороб. Проте серйозним викликом залишається контроль над їхньою активністю, а отже й над силою ефекту майбутніх ліків. З одного боку, висока активність може бути плюсом, оскільки це передбачає зменшення дози, потрібної для досягнення терапевтичного ефекту. Та з іншого, це несе за собою загрозу вищого шансу передозування препаратом з досягненням токсичної концентрації і непередбачуваних побічних ефектів. Тому під час розробки ліків важливо, щоб різниця концентрацій для досягнення токсичного та терапевтичного ефектів була якомога більшою.

Внаслідок різкого збільшення популярності моделей керованого машинного навчання та штучного інтелекту, багато дослідників почали звертатися до цих методів для аналізу білкових послідовностей. Такі потужні інструменти, як AlphaFold2, що передбачає просторову структуру білка на основі амінокислотної послідовності, чи DeepFRI для передбачення його функцій, роблять справжню революцію в галузі протеоміки [3, 4]. Проте це лише невелика частина з того, на що можуть бути здатні методи машинного навчання, оскільки некеровані моделі не так часто використовуються для подібних зав-

дань, і не так досліджені в цій сфері.

Використання комп'ютерних моделей штучного інтелекту для генерації нових біологічно активних поліпептидних послідовностей, або модифікації існуючих для зміни їх властивостей в бажаному напрямку є перспективним методом отримання нових лікарських препаратів. Актуальність такого підходу полягає в тому, що це дасть можливість використати повний потенціал білків без потреби в довготривалих експериментах, економлячи час і ресурси та не експлуатуючи велику кількість піддослідних тварин.

Метою нашого дослідження було створити прототип нейронної мережі, що зможе з високою точністю класифікувати відомі послідовності амінокислот на отруйні та неотруйні, й перевірити її спроможність до класифікації білків за межами тренувальних даних.

Відповідно до мети, ми поставили перед собою наступні завдання:

1. Зібрати, підготувати та візуалізувати дані про амінокислотні послідовності токсичних та нетоксичних білків з публічно доступних джерел;

2. Розробити дизайн штучної нейронної мережі та використати отримані дані для її тренування, отпимізації параметрів і перевірки спроможності правильно класифікувати білки на токсичні та нетоксичні;

3. Згенерувати штучні амінокислотні послідовності з бажаними функціями за допомогою стороннього програмного забезпечення, що будуть імітувати білки з невідомими властивостями;

4. Перевірити здатність нейронної мережі до класифікації послідовностей за межами її тренувального набору та порівняти її з існуючими моделями.

1 Огляд літератури

1.1 Загальна характеристика токсинів білкової природи

Токсини широко розповсюджені в тваринному світі. Змії, павуки та скорпіони, деякі комахи, жаби, молюски, кишковопорожнинні й навіть ссавці використовують комплексні суміші з отруйних речовин для ураження здобичі чи захисту від інших тварин. Більшу частину таких коктейлів складають білки та пептиди, що мають активну дію на нервову систему або клітинні мембрани жертви [5].

Еволюційно, отруйні білки виникають як результат впливу мутацій на дублікати генів неотруйних білків, що в результаті отримали токсичні властивості та дали носіям цих генів перевагу над іншими. Такі процеси не є рідкістю, оскільки представники з різних неспоріднених таксономічних груп розвинули здатність до вироблення токсинів незалежно один від одного, та мають свої унікальні набори речовин [6]. Припускається, що кількість унікальних пептидів, які можуть входити до складу таких коктейлів, може сягати більше 10 мільйонів лише в павуків [7]. Проте наразі в базі даних UniProtKB із них задокументовано трохи більше 7 тисяч, що натякає на існування великої кількості досі невідомих білків з подібною активністю [8].

За механізмом дії, отрути можна класифікувати на декілька груп. Нейротоксини є одними з найпоширеніших. Вони вражають нервову систему жертви, впливаючи на синапси чи канали клітин через зв'язування з рецепторами або іншими білками їх на поверхні. В результаті провокується неконтрольоване вивільнення чи блокування нейромедіаторів, або інгібується активність їх розщеплення після виділення [9]. Також досить поширені токсини, що запобігають згортанню крові за рахунок руйнування відповідальних за це білків, чи навпаки провокують його (рис. 1.1) [10]. Ще однією важливою групою є цитотоксичні білки, що викликають некроз клітин шкіри, печінки, крові, м'язів тощо.

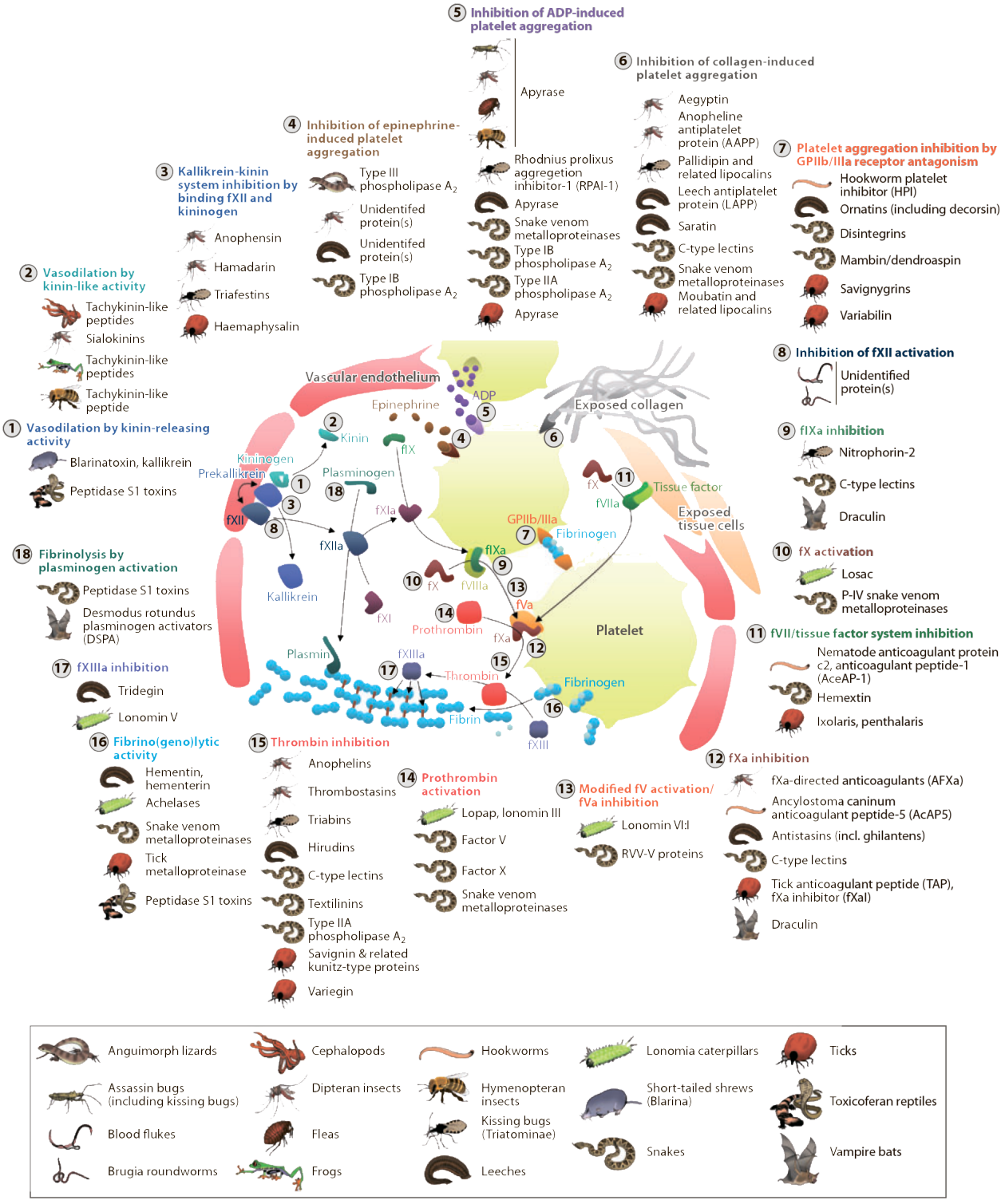


Рис. 1.1 - Класифікація токсинів за механізмом дії на згортання крові. Адаптовано з [1]

.

Більшість токсинів не утворюється одразу в активній формі. Зазвичай вони синтезуються разом із пропептидами, та активуються лише коли досягають місця постійного зберігання (наприклад, в отруйних залозах) або після

введення отрути в організм жертви. Такий механізм забезпечує довготривале збереження активності токсинів, захист носія отрути від ураження нею, правильне згортання у тривимірну структуру тощо [11]. Довжина амінокислотної послідовності активної форми токсинів зазвичай має невеликі розміри, що полегшує розповсюдження токсинів по тілу жертви, пришвидшуючи їх дію.

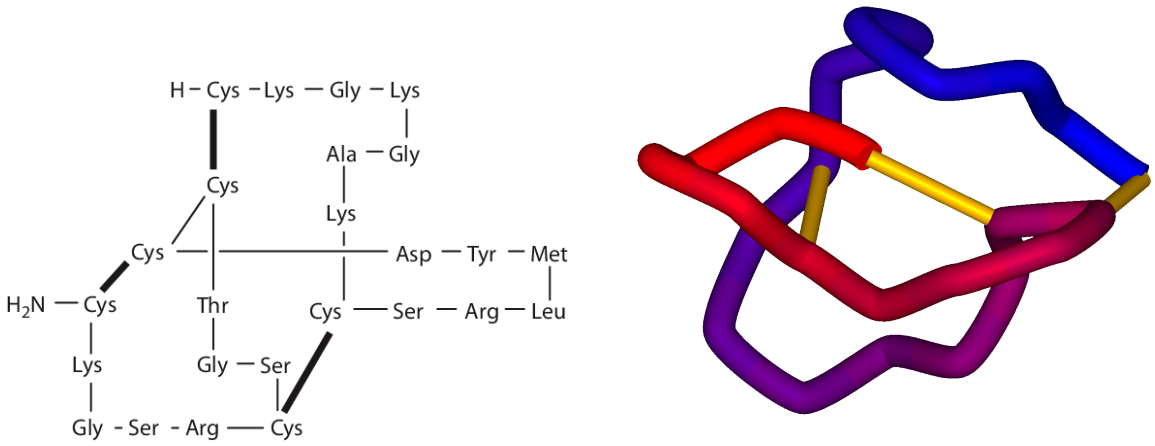
1.2 Використання токсинів у медицині та розробці ліків

Отруйні речовини здавна використовуються людьми в медичних практиках як ліки проти різноманітних хвороб чи полегшення симптомів. Отрута бджіл, як вважається, може полегшувати перебіг автоімунних хвороб, запальних процесів, захворювань сполучних тканин тощо [12]. Зміїну отруту використовували для лікування серцево-судинних захворювань через її комплексний вплив на систему кровообігу, а павучу — як протимікробний та протигрибковий засіб, знеболююче тощо [13, 14]. Хоч частина лікувальних ефектів отрут не підтверджені науковими дослідженнями, а деякі з них в є відверто сумнівними в контексті альтернативної медицини (наприклад, лікування ними ракових захворювань), проте ізоляція окремих токсинів із суміші справді є важливою частиною досліджень та розробки нових лікарських препаратів.

Будь-який процес розробки нового препарату має етап скринінгу бібліотек сполук. Для цього можуть використовувати як природні молекули з відомими властивостями, так і синтетичні молекули, яких ніколи не існувало в природі. Раціональний драг-дизайн передбачає використання різноманітних методів для передбачення структури, властивостей та активності досліджуваної сполуки в певних умовах. В такому випадку молекули з відомою активністю можуть слугувати основою для пошуку інших, оскільки схожість структури часто передбачає схожість властивостей. Токсини є одними з найкращих кандидатів на таку роль, оскільки мають високу специфічність та активність, а велике різномаїття їхніх структур передбачає велику кількість своє-

рідних молекул-лідів. Та іноді й самі отруйні пептиди часто мають потенціал стати ліками або їх частиною.

Одним з перших прикладів такого використання токсинів є розробка каптоприлу, антигіпертензивного препарату, що інгібує ангіотензин-перетворюючий фермент. Його синтезу передувало дослідження отрути жарараки — змії, що поширена на центральному сході Південної Америки. Структура пептидів, виділених із отрути, допомогла визначити основні властивості, які повинні були мати малі молекули для зв'язування із ангіотензин-перетворюючим ферментом та його інгібування. Іншим прикладом є зіконотид, пептидний засіб для полегшення сильного та хронічного болю (рис. 1.2). За назвою препарату ховається ω - конотоксин MVIIA, що блокує відповідальні за больові відчуття потенціал-залежні кальцієві канали N типу, і був виділений з черевоногого молюска *Conus magus* [15].



(а) (б)

Рис. 1.2 - Амінокислотна (а) та отримана за допомогою ЯМР (б) структури зіконотиду. Адаптовано з [16, 17]

.

Звичайно, не всі токсини перетворюються на ліки. Під час досліджень багато з них демонструють сильні побічні ефекти, що нівелюють користь від застосування препарату або становлять загрозу для життя та здоров'я людини. Тому розробка ліків на основі токсинів є складним завданням, що потребує великої кількості часу та ресурсів, і не завжди призводить до відкриття

нового препарату. Проте, зважаючи на велике різноманіття структур та властивостей пептидів, виділених з отрут, цей напрям є доволі перспективним, а його потенціал залишається слабко розкритим.

1.3 Перспективи використання методів машинного навчання в розробці нових ліків

Науково-технічний прогрес, розвиток експериментальних та розрахункових методів сильно полегшив шлях сполуки від її першої ізоляції чи синтезу до комерційного виробництва. Проте нещодавні досягнення в галузі машинного навчання та штучного інтелекту роблять справжню революцію в цьому питанні. Розроблені та навчені моделі уже сьогодні можуть досить точно передбачати властивості та біологічну активність молекул, просторову структуру білків, їх функції та взаємодії тощо [3, 4, 18]. В результаті відбувається процес переосмислення підходів для відкриття нових ліків, що будуть включати використання подібних передбачень для здешевлення та пришвидшення роботи, покращення результатів та збільшення кількості доступних ліків на ринку (рис. 1.3).

Процес розробки нових препаратів є багатоетапним процесом, і майже на кожному з цих етапів можливе застосування моделей машинного навчання. Для пошуку потенційних мішеней було розроблено систему BeFree та подібне програмне забезпечення, що на основі літературних джерел та баз даних передбачають взаємозв'язки між хворобою та її причиною, надаючи список варіантів генів та їх білкових продуктів, відповідальних за цей фенотип, і оцінюючи можливість впливу на них лікарських засобів [20, 21]. Для передбачення просторової структури білків за відсутності експериментальних даних широко використовується уже згадана вище нейронна мережа AlphaFold2, що має високу точність та надійність. Такі моделі як ScanNet чи ProSP можуть використовуватися для передбачення сайтів зв'язування на поверхні просторової структури мішені або на основі її первинної послідов-

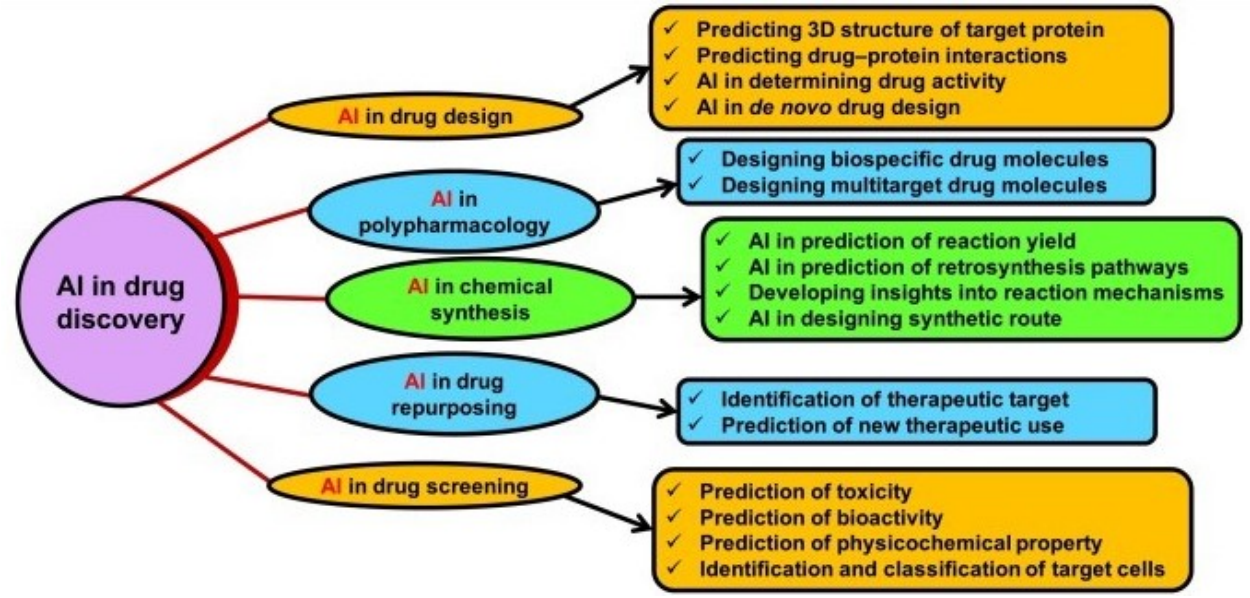


Рис. 1.3 - Застосування методів машинного навчання у фармакології. Адаптовано з [19]

.

ності [22, 23]. З іншого боку, за допомогою MolAICal та RFdiffusion на основі цих даних було показано можливість генерації великої кількості малих молекул або пептидів, що потенційно можуть міцно зв'язуватися з цими сайтами [24, 25]. Також існують моделі, що можуть перевірити наявність в отриманих молекул потрібних властивостей та активності, а також передбачити небезпечні побічні ефекти та неспецифічні взаємодії з іншими мішенями [26]. Для синтезу малих молекул було розроблено програмне забезпечення Chematica, що має змогу оптимально підібрати послідовність хімічних реакцій для їх отримання, а моделі аналізу зображень допомагають оцінити результати високопродуктивного скринінгу на основі фотографій [27, 28]. Також існують моделі для оптимізації структур лідів на основі їх властивостей та визначення оптимальних дозу препарату для переходу до експериментів над тваринами та людьми [29].

Перелічені завдання складають лише частину всієї розробки нових препаратів, проте наявність такого різноманіття наявних інструментів їх вирішення показує інтерес наукової спільноти до методів машинного навчання. Хоч такі інструменти й справді можуть сильно допомогти в дослідженнях,

проте наразі вони ще не досягли того рівня достовірності та універсальності, щоб замінити такими ними весь процес розробки. Моделі машинного навчання мають пройти ще довгий шлях удосконалення, перевірки та оптимізації, й мають ряд недоліків, найголовнішим із яких є кількість навчальних даних, потрібних для тренування моделей. Але в будь-якому випадку, частка інструментів на основі цього підходу наразі тільки росте, і вони поступово витісняють існуючі методи.

1.4 Великі мовні моделі в біології. Білкові мовні моделі

Останні прориви в галузі створення великих мовних моделей, що дозволили їх використання пересічними людьми для особистих потреб, уже залишили окремий слід в історії методів машинного навчання. Такі моделі як GPT-4 та Gemini використовуються у повсякденному житті для генерації текстів, що неможливо відрізнити від написаних людиною, перекладу з однієї мови на іншу, отримання узагальненої інформації на будь-які теми та навіть допомоги з написанням коду на різноманітних мовах. Це не могло не зацікавити дослідників з інших галузей, в тому числі й біологічних наук, що дало новий поштовх для розвитку подібних моделей для різноманітних завдань.

Одним із найочевидніших підходів є використання існуючих моделей, що використовують та аналізують людську мову, для вирішення біологічних завдань. Наприклад, було показано здатність чат-ботів ChatGPT 4.0, Bard та BingAI до діагностики та наданню рекомендацій щодо здоров'я та хвороб опорно-рухової системи [30]. Іншим підходом є тренування окремих моделей, що також аналізують людську мову, наприклад, для анотації наукових публікацій або текстів і розпізнавання в них іменованих сутностей, таких як назви хвороб, генів, білків тощо [31].

Хоча перераховані моделі є перспективними та мають свою нішу застосування, все ж найбільш популярним наразі є третій підхід, що передбачає

розробку моделей для аналізу не людської мови, а послідовностей нуклеїнових кислот чи білків. Він ґрунтується на припущенні, що амінокислотні та нуклеотидні послідовності мають схожу структуру що й послідовності слів у реченні, і за допомогою тренування мовних моделей нуклеїнових кислот або білків можна передбачати функції та властивості конкретних послідовностей, і генерувати їх на основі заданих даних. Через це моделі, що аналізують амінокислотні послідовності, називаються білковими мовними моделями, відповідно також виділяють мовні моделі ДНК та РНК. Часто вони імплементують архітектуру трансформерів та увагу або рекурентних нейронних мереж для аналізу не просто послідовності, а й вплив контексту окремих її елементів на інші, розташованих не безпосередньо один біля одного.

Сучасні білкові мовні моделі часто виконують завдання генерації штучних амінокислотних послідовностей на основі вхідних даних, такі як інформація про функцію, вторинну та третинну структуру, референсну послідовність тощо [32]. Однією з найновіших моделей є Chroma, що, як заявляється, дає можливість користувачу комбінувати разом умови генерації, такі як довжину, просторову структуру, симетрію, наявність доменів чи навіть написати ці умови природньою мовою (рис. 1.4), а також регулювати вплив кожної з цих умов на отриману структуру і відповідну послідовністю [33]. Схожі функції мають також такі моделі як ProtGPT2, ProGen тощо, і хоч вони не мають багатьох особливостей попередньої, проте на відміну від неї дозволяють зробити тонке налаштування параметрів за рахунок короткого тренування на певному наборі білків, щоб згенеровані послідовності відповідали розподілу тренувального набору [34, 35].

Іншим застосуванням моделей є передбачення функцій певних білків, їх фізико-хімічних властивостей, розташування в клітині тощо. Наприклад, модель PLUS дозволяє на основі послідовності передбачити гомологію, вторинну структуру, розчинність, локалізацію, стабільність, флуоресценцію тощо, а CNN-BiLSTM — визначити, чи білок має можливість зв'язуватися з ДНК [36, 37].

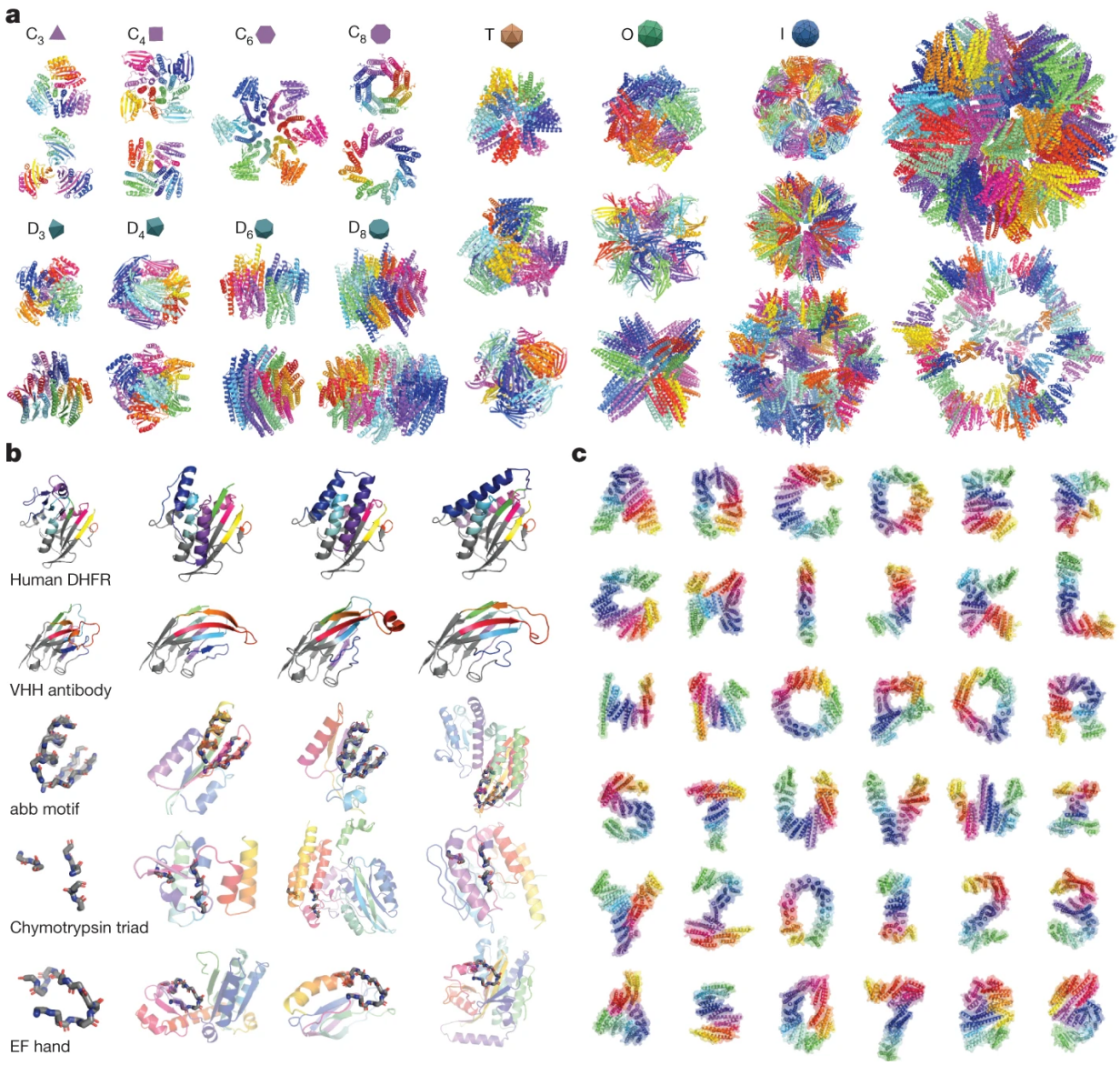


Рис. 1.4 - Можливості задання симетрії, підструктури та форми для генерації пептидів за допомогою білкової мовної моделі Chroma. Адаптовано з [33]

.

Ще одним застосуванням можна окремо зазначити передбачення впливу мутацій чи варіантів послідовності на функціонування білка чи інші його властивості. Наприклад, Tranception дозволяє передбачити вплив поліморфізмів та мутацій на виникнення генетичних хвороб людини, уникнення вірусами антитіл до їх капсидів чи дизайну нових пептидів для лікування хвороб [38].

Нейронні мережі швидко розвиваються та удосконалюються. Ті архітектури, що ми сьогодні вважаємо передовими зараз, завтра можуть поступитися іншим. Наприклад, зовсім нещодавно було розроблено технологію

MAMBA, що є ідейним продовженням рекурентних нейронних мереж, і майже в кожному тесті перевершувала існуючі нейронні мережі на основі трансформерів [39]. Скоріш за все, з кожним роком інструменти на основі нейронних мереж будуть все швидше витісняти або доповнювати існуючі моделі, і щоразу показувати все кращі результати. Можливо, через деякий час навіть експериментальні методи відійдуть на другий план, або навіть будуть лише доповнювати результати нових моделей, а не виступати самостійними методами.

2 Матеріали та методи

Програмний код для повторення отриманих нами результатів знаходиться за посиланням `https://github.com/Shmotyamax/Model_toxins`

2.1 Збір, підготовка та аналіз даних

Для нашого дослідження ми використовували повну базу даних перевірених біологічних послідовностей протеїнів Swiss-prot від 2024 року. Додатково, ми окремо завантажили дані про токсичні й отруйні білки організмів з *Metazoa* з бази Swiss-prot за допомогою анотацій від програми Tox-Prot [40, 41].

Далі за допомогою програмного забезпечення USearch ми виконали кластеризацію послідовностей Swiss-prot та залишили тільки унікальні [42].

Наступні кроки ми виконували за допомогою мови програмування Python версії 3.10.4 та встановлених бібліотек.

За допомогою бібліотеки MATPLOTLIB ми візуалізували розподіл довжин амінокислотних послідовностей токсичних білків, а також наклали на отриману гістограму їх таксономічну та функціональну приналежність для кращого розуміння розподілу та структури даних [43].

2.2 Перетворення послідовностей на числові вектори

Наступним кроком було перетворити дані первинної структури протеїнів на числові вектори, що зберігатимуть більшу частину інформації про їх амінокислотний склад та послідовність. Для цього, спочатку послідовності було перетворено в списки із триплетів за допомогою ковзаючого вікна з кроком в одну амінокислоту. Далі, призначивши кожному списку відповідний ідентифікатор, ми натренували декілька моделей Doc2Vec, що відрізнялися за методом імплементації, розміром вихідного вектора, кількістю тренувальних епох

тощо [44, 45]. Тренування було проведено на сталій підмножині з 75% всього набору даних, після чого залишкові 25% були використані для тестування моделей і коригування гіперпараметрів з метою отримання кращих моделей.

Моделі Doc2Vec є мілкими нейронними мережами, що зазвичай використовуються в обробці природної мови та тренуються на основі текстових документів та слів, що входять до його складу. Вона бере до уваги порядок слів у документі, частоту вживання слів окремо та в комбінації з іншими тощо. В залежності від імплементації, вони можуть передбачати наступне слово в реченні на основі векторів слів та/або узагальненого вектора контексту документа. В нашому випадку, слова є триплетами амінокислот, тоді як документи є списками цих триплетів. За допомогою навченої моделі можливо отримати вектори як для окремих слів, так і для документів в цілому. До того ж, якщо документи є схожими за змістом, то й їх вихідні вектори будуть направлені в багатовимірному просторі в подібному напрямку, й навпаки, що ідеально підходить для нашого завдання. Також важливо зауважити, що Doc2Vec є повністю некерованим методом, тобто модель сама визначає більш та менш важливі елементи послідовності.

Для імплементації моделі Doc2Vec існує два різних методи: PV-DM (Paragraph Vector - Distributed Memory) та PV-DBOW (Paragraph Vector - Distributed Bag Of Words), архітекутра кожного з яких зображена на рис. 2.1. Для того, щоб обрати кращий, було відібрано 5% від усіх даних та проведено тренування на цьому невеликому наборі протягом 100 епох за розмірності векторів 50. Перевірку здатності моделей до узагальнення було проведено шляхом використання тренувального набору як тестового, і порівняння передбачених векторів з тими, що були отримані в результаті навчання.

Після тренування моделі на тренувальному наборі даних, щоб перевірити якість моделі та отриманих векторів, було проведено декілька тестів. По-перше, було проведено тестування на частці тренувальних даних аналогічно до тесту в попередньому параграфі.

По-друге, ми перевіряли, наскільки отримана модель вміє передбачати

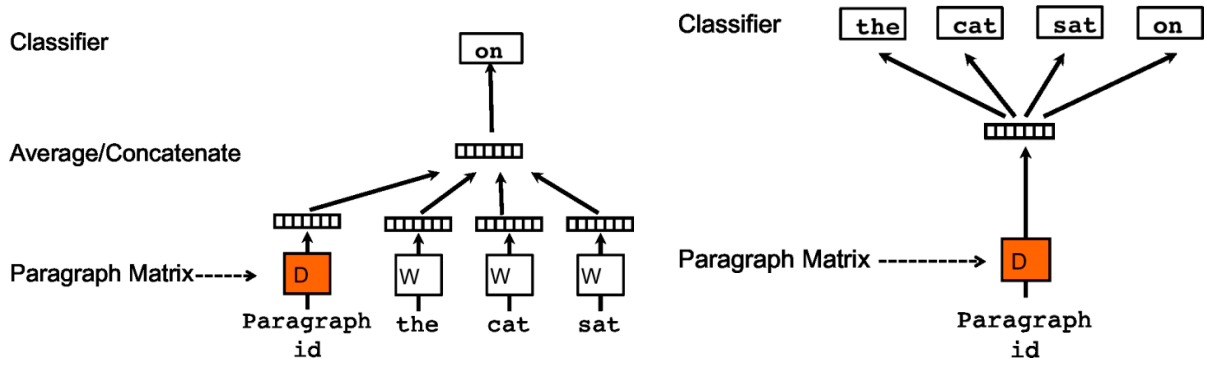


Рис. 2.1 - Імплементація моделі Doc2Vec у вигляді PV-DM (а) та PV-DBOW (б). PV-DM передбачає наступне слово у реченні на основі вектора документу та векторів слів, тоді як PV-DBOW генерує текст лише на основі вектора документа. Адаптовано з [44]

.

токсичність білків лише на основі схожості вектора тестового білка до векторів тренувальних послідовностей, та порівняли її ефективність із передбаченнями, зробленими на основі результатів алгоритму BLASTp. BLAST (Basic Local Alignment Search Tool) є фундаментальним інструментом у біоінформатиці, що дозволяє швидко порівнювати первинні біологічні послідовності, як-от нуклеотидні або амінокислотні, із великими базами даних [46]. Цей метод забезпечує можливість виявлення регіонів локальної схожості, що можуть вказувати на біологічні взаємозв'язки між послідовностями, а також на їх функціональну або еволюційну спорідненість, що дозволяє передбачити токсичність білків. BLAST використовує алгоритм, який шукає максимальні локальні вирівнювання, що є дуже ефективним для аналізу великих баз даних. Інструмент складається з декількох версій, таких як BLASTn для нуклеотидних послідовностей, BLASTp для амінокислотних послідовностей тощо. Кожна версія оптимізована для різних типів даних та завдань, що дозволяє виконувати більш специфічний і точний аналіз. Для кожного тестового білка ми отримали вектор та передбачили п'ять найбільш подібних тренувальних векторів до нього. Якщо кількість векторів, що відповідають токсичним білкам, переважала, то й тестовий білок передбачався токсичним, і навпа-

ки, тобто використовувався метод простої більшості голосів (надалі — метод голосування). Також ми зробили передбачення на основі лише одного найбільш подібного вектора. Аналогічно, ми використали BLASTp, щоб серед бази даних SwissProt знайти 5 найбільш подібних послідовностей, за винятком її самої, і використовували ті ж самі правила передбачення.

Щоб перевірити роздільну здатність моделі, ми застосували схожий підхід для передбачення ключових слів для токсичних послідовностей, що позначають шлях їхньої дії на жертву, і наявні для більшості білків у базі даних. Для визначення правильності передбачень ми рахували кількість унікальних ключових слів, що зустрілися у п'яти білків з найподібнішими до певного білка передбаченими векторами. Ключове слово, що зустрічалося найчастіше, й було передбаченням моделі.

Після визначення фінальних гіперпараметрів, дві моделі, що відрізнялися лише своєю розмірністю, було натреновано на повному наборі даних. Після цього, кожен список триплетів було перетворено на два вектори, по одному з кожної моделі, зберігаючи свій ідентифікатор, а також отримавши позначку 1, якщо білок є токсичним, або 0 якщо ні.

2.3 Навчання нейронної мережі

Штучні нейронні мережі використовуються для завдань класифікації чи регресії, та можуть з високою точністю апроксимувати будь-яку функцію на деякому проміжку, яка залежить від великої кількості параметрів. Вони складаються з окремих елементів — нейронів, що мають параметр зміщення, розташовані шарами і пов'язані між собою ваговими коефіцієнтами. Нейрони отримують числові сигнали від попереднього шару, множать їх на вагові коефіцієнти, підсумовують разом зі зміщенням і цей результат використовується як сигнал для наступного шару. Перший шар нейронної мережі називається вхідним, і не має параметра зміщення. Він лише надає сигнали для наступного шару. Далі йдуть від одного (мілка мережа) до декількох (глибока мережа)

прихованих шарів, а в кінці розташований вихідний шар, що видає результат обчислення.

Нейронні мережі навчаються за рахунок тренування на великій кількості даних та змін значень зміщення та вагових коефіцієнтів в залежності від правильності передбачень. Зміни обчислюються на основі функції втрат та алгоритму градієнтного спуску, й поширюються від вихідного шару до вхідного. В результаті можна отримати модель, яка буде з високою точністю класифікувати об'єкти або передбачати їх певні параметри на основі вхідних числових значень. В нашому випадку відбувалося тренування нейромережі для класифікації білків на токсичні та нетоксичні.

Для тренування та тестування ми використовували імплементацію нейронних мереж на основі моделі багатошарового перцептрона з бібліотеки SCIKIT-LEARN [47]. Вхідними параметрами для мережі слугували вихідні вектори, отримані за допомогою моделі Doc2Vec, та відповідні позначки, що слугували індикатором токсичності чи нетоксичності білка. Як і в попередньому випадку, ми розділили дані на тренувальний набір, що містив 75% векторів, та тестовий з 25%. Проте додатково, під час навчання 10% даних від тренувального набору обиралися як набір для валідації. Як і з моделями Doc2Vec, було натреновано декілька моделей з різними гіперпараметрами для визначення їх оптимальних значень, після чого фінальна нейромережа була натренована на всьому наборі даних. Для тестування нейронних мереж ми використовували вбудовані в бібліотеку методи, які дозволяли автоматично оцінити якість моделі на основі тестового набору даних, а також побудувати й проаналізувати матриці невідповідностей, криві ROC та PR і площі під цими кривими.

2.4 Генерація штучних білкових послідовностей

Після тренування нейронної мережі на всьому наборі даних, використовувати його для тестування недоцільно, оскільки модель є упередженою до цих

даних. Щоб зробити тестування неупередженим було вирішено застосувати інструменти генерації штучних білкових послідовностей, де отриманий набір буде мати схожий розподіл до тренувальних даних, але не міститиме ідентичних до них послідовностей. Для цього було обрано інструмент Chroma, що представляє собою білкову мовну модель з архітектурою трансформера. Він має здатність генерувати білки з заданими характеристиками або використовувати існуючі білки як шаблон. Для отримання штучних послідовностей було обрано токсичні білки з відомою просторовою структурою. Вони були використані як шаблон, на основі якого шляхом часткової дифузії та перетворення третинної структури на амінокислотну послідовність було отримано набір штучних білків. Обробивши PDB-файли білків з відомою просторовою структурою, для кожного з них було створено по одному штучному білку. Після видалення низькоякісних послідовностей (наприклад, тих що складались цілком із 1-3 типів амінокислот), кожна з тих, що залишилися, була перетворена на вектор, і на його основі було передбачено її токсичність.

Іншим застосуванням штучних білкових послідовностей є розширення тренувального набору даних. Оскільки кількість токсичних білків є в десятки разів меншою за кількість нетоксичних, така незбалансованість може негативно впливати на здатність моделі до начання та класифікації. Тому як додатковий етап було вирішено використати комбінацію інструментів RFdiffusion та ProteinMPNN для схожої генерації штучних послідовностей [25, 48]. Проте цього разу було використано токсичні білки, що були хибно передбачені в попередніх тестах, мали структури в базі даних передбачень AlphaFold2 та довжину від 20 до 150 амінокислотних залишків. За допомогою RFdiffusion було проведено подібну до Chroma часткову дифузію структур, а ProteinMPNN перетворив штучні структури на амінокислотні послідовності. Отримані дані були додані до навчального набору, щоб перевірити вплив розширення тренувального набору на ефективність навчання нейронної мережі. Для кожної обраної структури було створено по 3 моделі, і для кожної моделі було передбачено по 2 штучні послідовності. Після цього було проведено звичайну

перевірку моделі на тестовому наборі даних.

Ще однією можливістю штучного розширення тренувального набору даних є методи оверсемплінгу. Вони передбачають використання існуючих векторів мінорного класу як основи для створення нових. Наприклад, метод випадкового оверсемплінгу передбачає утворення нового вектора в результаті випадкової невеликої зміни параметрів існуючого, тоді як метод SMOTE використовує два випадкових вектори, щоб утворити штучний на прямій, що їх з'єднує. З іншого боку, є методи андерсемплінгу, що передбачають зменшення вибірки класу, що переважає, або гібридні методи, що комбінують оверта андерсемплінг [49]. Ми спробували використати методи оверсемплінгу та андерсемплінгу в імплементації бібліотеки IMBALANCED-LEARN для того, щоб збалансувати тренувальний набір даних з метою покращення передбачень моделі[50].

2.5 Порівняння фінальної моделі із ToxinPred3.0

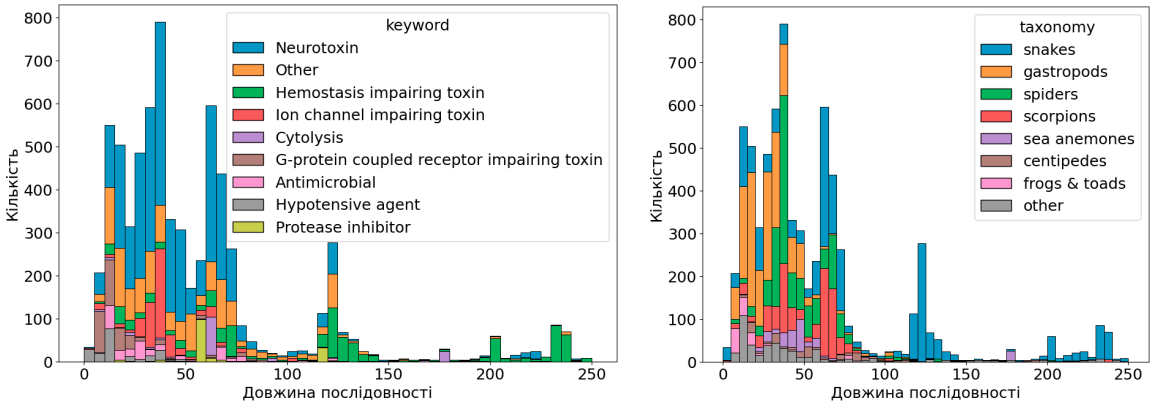
Вільнодоступний інстумент передбачення токсичності пептидів ToxinPred3.0 є гібридним методом, що комбінує результати моделі машинного навчання та алгоритму MERCI (Motif-EmeRging and with Classes-Identification) для отримання найбільш точних результатів [51, 52]. Серед існуючих інструментів, що знаходяться у вільному доступі, наразі він вважається одним із найпрогресивніших. Ми вирішили порівняти здатність нашої моделі та ToxinPred3.0 виконувати завдання класифікації.

Ми використали тестувальний набір даних ToxinPred3.0, що складався із 1104 токсичних та 1104 нетоксичних пептидів довжиною до 35 амінокислотних залишків включно, для перевірки нашої кінцевої моделі. З іншого боку, ми використали всі токсичні та співрозмірну частину нетоксичних білків з нашого набору даних для класифікації за допомогою ToxinPred3.0, а також окремо класифікували утворені за допомогою Chroma штучні амінокислотні послідовності.

3 Результати

3.1 Збір, підготовка та аналіз даних

Завантажена база даних Swiss-Prot налічувала 570 830 послідовностей. Серед них, 6 951 білок було позначено як токсичний. Після кластеризації послідовностей за допомогою алгоритму USearch було отримано 482285 унікальних послідовностей, серед них 6 875 токсичних. Розподіл токсичних послідовностей за довжиною, функціональною та таксономічною приналежністю можна розглянути на рисунку 3.1. В цей час, більшість з усіх білків у базі Swiss-Prot мають довжини від 100 до 400 амінокислот.



(а) (б)

Рис. 3.1 - Розподіл токсичних білків за довжиною амінокислотної послідовності та: а) механізмом дії на жертву; б) таксономічною приналежністю власників токсинів. Показаний розподіл активних форм білків без сигнальних послідовностей та пропептидів. Найбільшу частину серед всіх білків складають короткі нейротоксини, що продукуються черевоногими молюсками з родини *Conidae*. Також значну частину складають білки змій, павуків та скорпіонів. Можна побачити, що токсини змій мають довші послідовності за більшість токсинів.

3.2 Перетворення послідовностей на числові вектори

Після перетворення кожної амінокислотної послідовності на список триплетів з відповідним ідентифікатором, було проведено тестування методів навчання PV-DM та PV-DBOW (рис. 3.2). Ми обрали метод PV-DBOW для навчання всіх подальших моделей Doc2Vec, оскільки він показав набагато більшу швидкість навчання моделі.

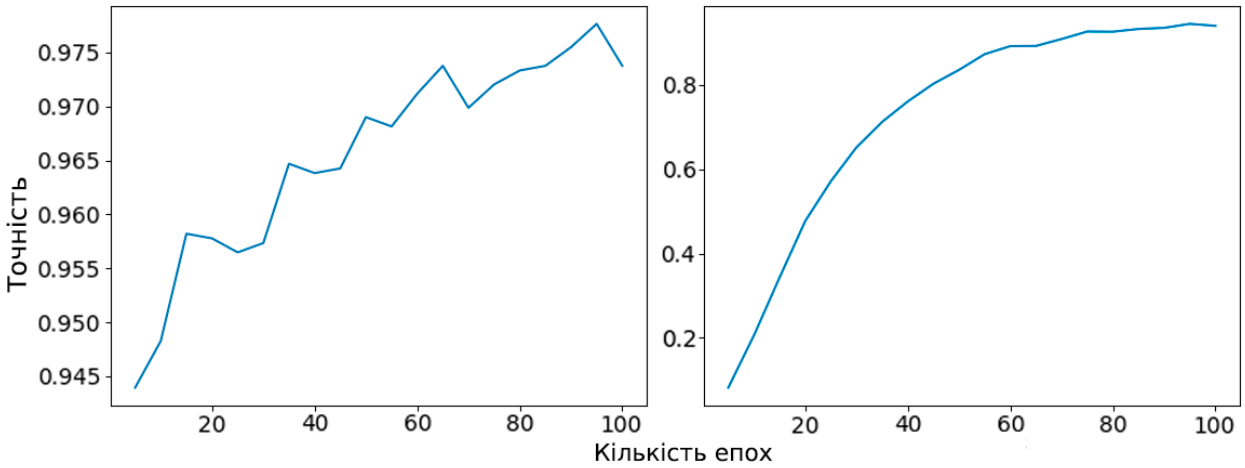


Рис. 3.2 - Криві навчання моделі Doc2Vec за методом: а) PV-DM; б) PV-DBOW. Точність передбачень оцінювалась кожні 5 епох. Під точністю мається на увазі відсоток послідовностей, для яких передбаченим найподібнішим до них вектором є вектор їх самих.

Далі ми відібрали 75% тренувальних та 25% тестувальних даних, використали тренувальний набір для навчання моделей з різними параметрами, та проводили їх тестування, щоб знайти оптимальні параметри. Зрештою, ми зупинилися на наступних: кількість тренувальних епох — 50; розмірність векторів — 100 та 200; мінімальна кількість зразків слова для включення у словник — 5; від'ємна вибірка — 5, інші параметри ми залишили за промовчанням. В першому тесті, що передбачав використання тренувальних даних як тестових, із 125 667 послідовностей, найбільш подібними самі до себе були 89,5% векторів для моделі з розмірністю 100 та 92,8% — із розмірністю 200, а серед 5 найподібніших були 98,1% та 99,1% відповідно. Це означає, що модель вдало перетворює послідовності на вектори, а збільшення розмірності

Таблиця 3.1 - Передбачення механізму дії токсичних білків за допомогою моделей Doc2Vec різних розмірностей та BLASTp

Модель	Правильні передбачення	Хибні передбачення
Розмірність 100	4752	2123
Розмірність 200	4880	1995
BLASTp	4527	2348

векторів у моделі Doc2Vec прямо пропорційне з покращенням результуючих векторів.

Результати передбачення токсичності білків на основі голосування та одного найподібнішого вектора вказані на рисунку 3.3. Матриці нормалізовані по рядкам. На головній діагоналі матриць розташовані істинно негативні та істинно позитивні передбачення, на побічній діагоналі — хибно-негативні та хибно-позитивні. Загалом, найкращі результати показала модель з розмірністю вектора 200 на основі передбачення найближчого вектора, тоді як BLAST хоч і мав подібні результати, але не зміг достовірно знайти жодних подібних білків для більш ніж 700 послідовностей. Це означає, що моделі Doc2Vec не тільки вдало перетворюють амінокислотні послідовності на унікальні вектори, а й зберігають інформацію, закладену в цих послідовностях, а також частково перекривають прогалини в тренувальних даних.

Результати перевірки роздільної здатності моделі вказані на таблиці 3.1. Загалом, можна побачити, що моделі передбачають тип токсинів не гірше за BLASTp, що доводить здатність моделі Doc2Vec утворювати вектори достатньо відмінні один від одного, щоб фіксувати навіть невеликі відмінності у послідовностях.

3.3 Навчання нейронної мережі

Після навчання фінальних моделей Doc2Vec та перетворення всіх послідовностей на відповідні вектори, було навчено декілька нейронних мереж з різними гіперпараметрами для визначення оптимальних. Зрештою, ми зупи-

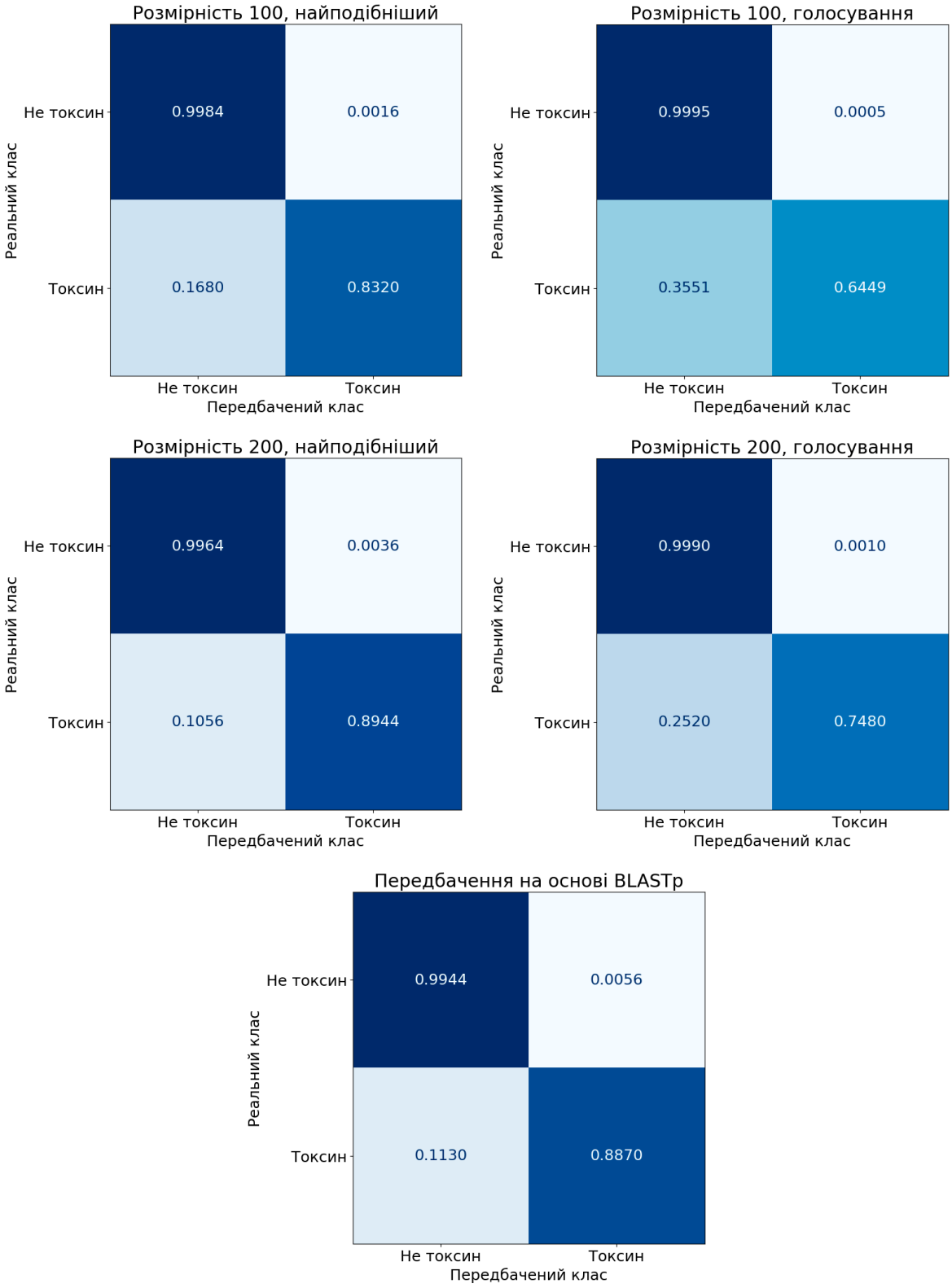


Рис. 3.3 - Матриці невідповідностей передбачень токсичності білків моделями Doc2Vec та на основі результатів BLASTp

нилися на наступних: кількість нейронів у прихованих шарах: 500 у першому та 250 у другому; параметр регуляризації: 10^{-8}, алгоритм оптимізації: Adam.

Також ми впровадили ранню зупинку навчання тоді, коли здатність моделі передбачати набір для валідації не покращувалась на 0,01% протягом 10 навчальних епох.

Результат тестування нейронної мережі, навченої на тренувальному наборі векторів з розмірністю 200, зображено на рисунку 3.4. Загалом, можна побачити що отримана модель успішно класифікує білки на токсичні та нетоксичні, проте потребує зміщення порогового значення для знаходження оптимального балансу між кількістю хибно-позитивних та хибно-негативних передбачень. Надалі ми використовували значення 10^{-5} як порогове.

Також, ідентично до рисинку 3.1, ми побудували гістограми розподілу білків, що були передбачені хибно-позитивно та хибно-негативно, щоб виявити закономірності, які можуть об'єднувати їх та пояснювати хибну класифікацію. З гістограми 3.5а можна побачити, що розподіл довжин білків, класифікованих як хибно-позитивні, майже повторює розподіл токсинів, більша частина таких білків секретуються, як і отрути, а деякі з них навіть мають антимікробну дію чи входять до складу жалких клітин кишковопорожнинних, що свідчить про їх близькість до токсинів і частково пояснює їх хибну класифікацію. На жаль, для хибно-негативно класифікованих білків подібних закономірностей ми не знайшли, хоча й не виключаємо її існування.

3.4 Генерація штучних білкових послідовностей

Після навчання фінальних нейронних мереж на всьому наборі даних ми приступили до створення штучних токсичних амінокислотних послідовностей. В результаті було отримано 447 штучних послідовностей, серед яких модель передбачила 364 (або 81%) як токсичні.

Також були створені штучні послідовності токсичних білків для розширення тренувального набору, що збільшило його на 2702 послідовності, кожна з яких була позначена токсичною. Проте, на жаль, тестування нової моделі призвело лише до змін, співрозмірних зі зміщенням порогового значення, і не

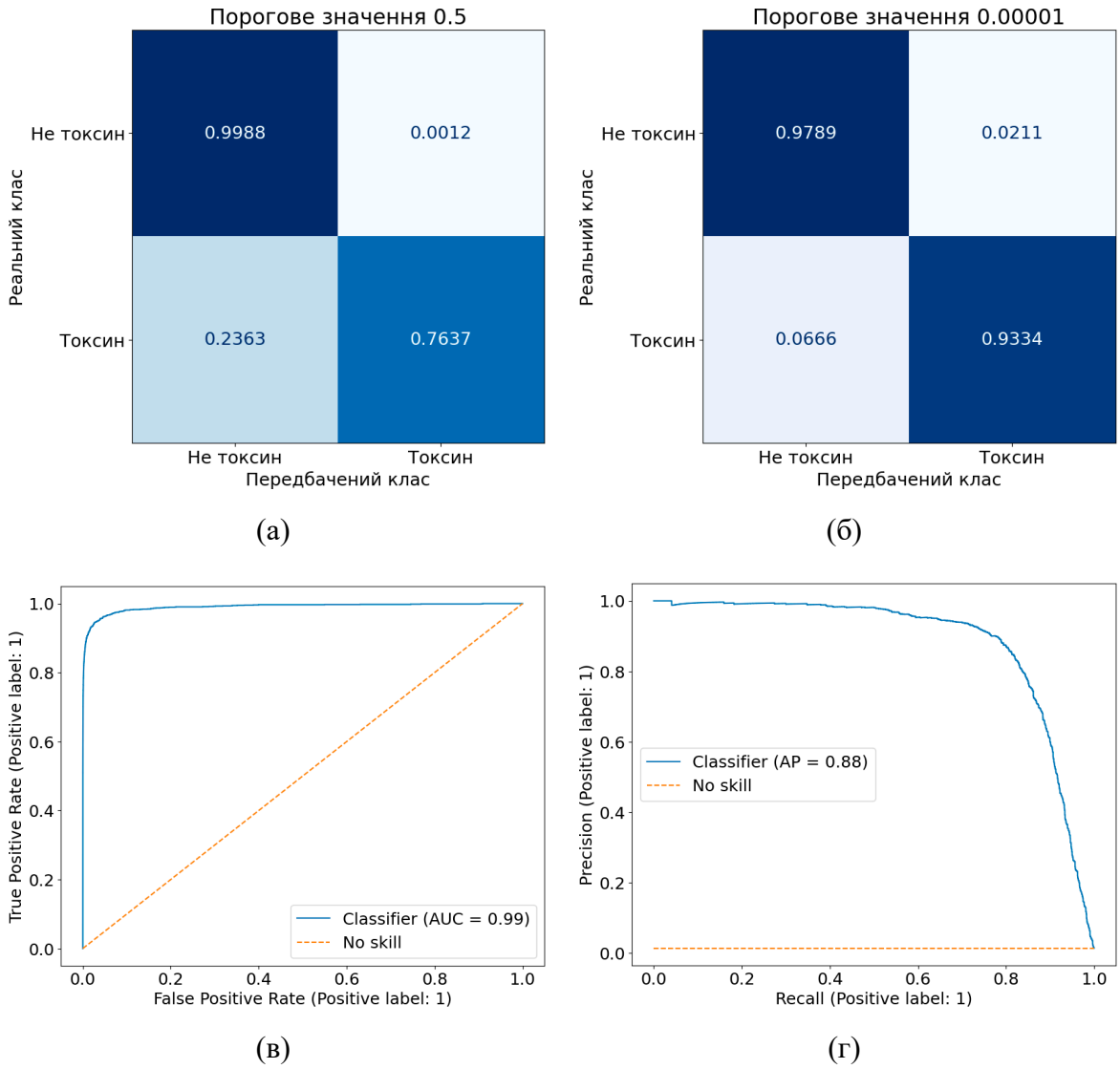
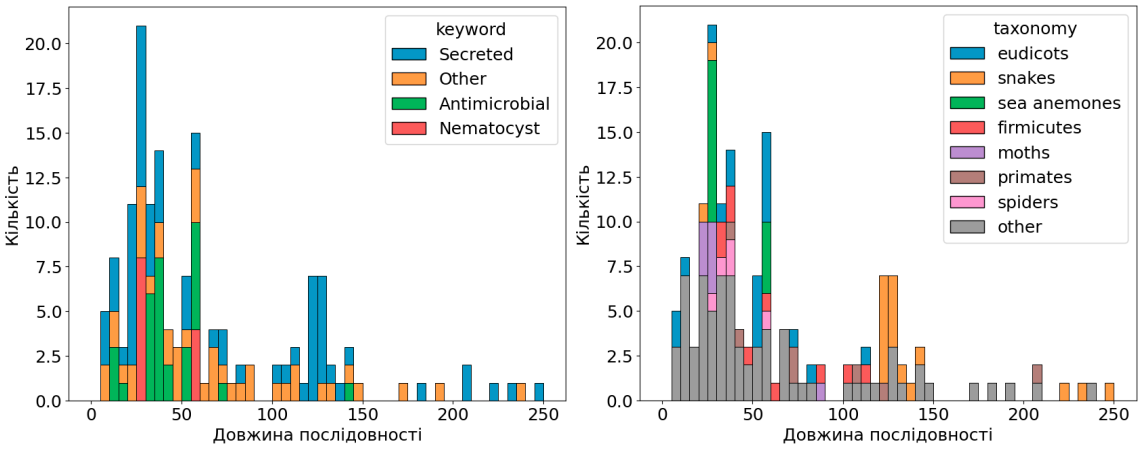


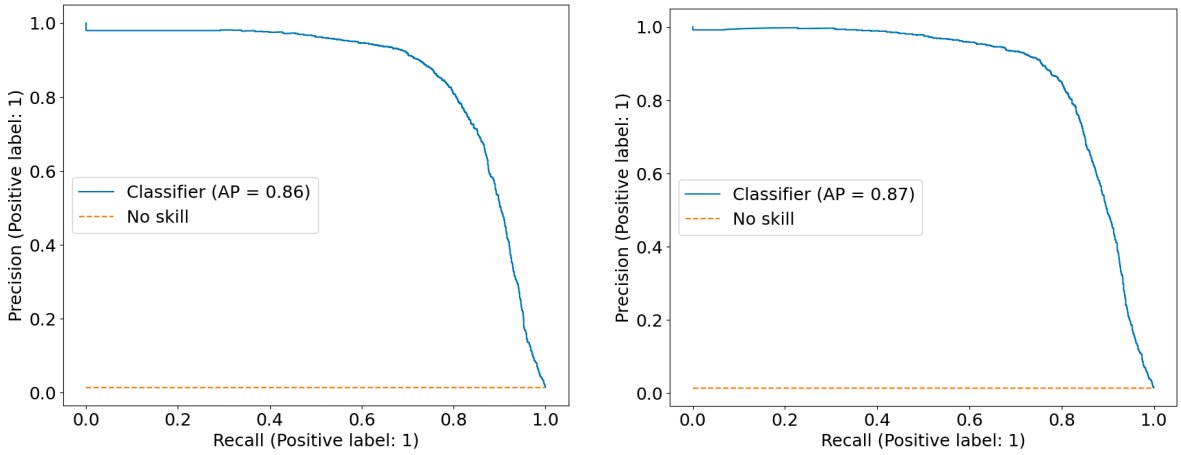
Рис. 3.4 - а-б) Матриці невідповідностей для передбачень токсичності білків із тестового набору за допомогою штучної нейронної мережі з розмірністю вхідного вектора 200 за порогового значення 0,5 (а) та 10^{-5} (б); в) Крива ROC та розрахована площа під кривою; г) Крива Precision-Recall та розрахована площа під кривою.

покращувало загальну точність моделі (рис. 3.6). Аналогічні результати були досягнені в результаті використання математичних методів оверсемплінгу та андерсемплінгу.



(а) (б)

Рис. 3.5 - Розподіл хибно-позитивно класифікованих білків за довжиною амінокислотної послідовності та: а) ключовими словами; б) таксономічною приналежністю власників білків. Показаний розподіл активних форм білків без сигнальних послідовностей та пропептидів.



(а) (б)

Рис. 3.6 - Криві Precision-Recall та площі під ними, отримані після тестування штучних нейронних мереж за вхідних векторів розмірності 100 (а) чи 200 (б) після збільшення кількості токсичних білків у тренувальному наборі за допомогою комбінації інструментів RFdiffusion та ProteinMPNN.

3.5 Порівняння фінальної моделі із ToxinPred3.0

Результати передбачень нашої моделі та моделі машинного навчання ToxinPred3.0 можна побачити на рисунку 3.7. Як виявилося під час тесту-

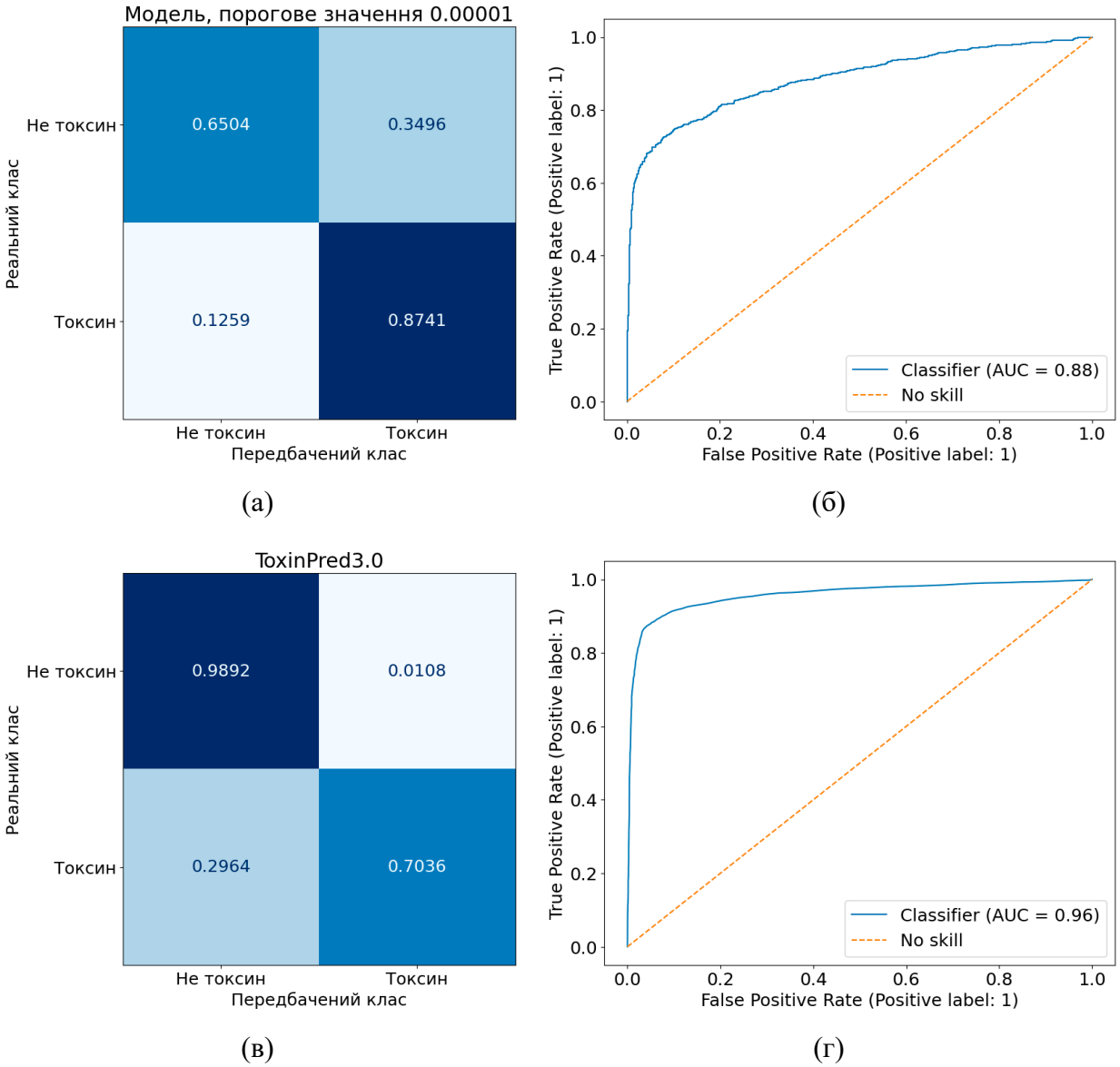


Рис. 3.7 - а-б): результати тестування штучної нейронної мережі з розмірністю вхідного вектора 200 на тестовому наборі даних ToxinPred3.0. а) матриця невідповідностей за порогового значення 10^{-5}; б) крива ROC та розрахована площа під нею. в-г) результати тестування інструмента ToxinPred3.0 на збалансованій частині тестового набору даних для нашої моделі. в) матриця невідповідностей за порогового значення 0,5; г) крива ROC та розрахована площа під нею.

вання, гібридний метод ET+MERCI є сильно упередженим до класифікації коротких послідовностей як токсичних, і довгих як нетоксичних, і не виконував якісні передбачення. Тому для порівняння ми використовували лише

метод ET. З графіків можна побачити, що хоч наша модель і поступається в точності, проте загалом зберігає здатність до класифікації пептидів навіть для такого специфічного набору даних. Також ToxinPred3.0 передбачив, що із 447 штучно згенерованих послідовностей токсичними є 398, тобто 88%, а отже наша модель передбачила лише на 9% менше токсинів у порівнянні з найкращою з існуючих моделей.

Також для цього завдання ми спробували використати моделі, натреновані на наборах даних, розширених з використанням інструментів RFdiffusion та ProteinMPNN або методів оверсемплінгу. Серед них лише модель, що в навчанні використовувала метод випадкового оверсемплінгу, показала незначне покращення передбачень моделі: площа під кривою ROC збільшилася з 0,88 до 0,90, тоді як інші методи майже не мали впливу на точність.

4 Обговорення

Незважаючи на простоту, невеликі розміри та прямолінійність дизайну нашої моделі, вона показала доволі гарні результати. Звичайно, вона не є ідеальною, і хоч має досить гарну специфічність, проте її чутливість до токсинів є об'єктом подальшого покращення, що ми компенсували за допомогою зміщення порогового значення до екстремального значення. На нашу думку, головними факторами, що лімітували успішність нашої моделі, могли бути: недостатня точність перетворень послідовностей на вектори та часткова втрата інформації під час цього процесу; незбалансованість використаного набору даних; субоптимальний вибір гіперпараметрів під час тренування; наявністю гомології між деякими токсичними та нетоксичними білками.

Загалом, отримані результати дослідження є більш ніж задовільними, оскільки головну мету, а саме створення прототипу нейронної мережі, було успішно виконано. Більше того, наша нейронна мережа лише незначно поступається у точності сучасній моделі ToxinPred3.0. Проте, варто зауважити, дві особливості цього порівняння: по-перше, тренувальний набір нашої моделі був у десятки разів більшим, аніж у вищезгаданої, а по-друге, незалежний тестувальний (як і навчальний) набір ToxinPred3.0 складався з дуже специфічного набору пептидів довжиною до 35 амінокислотних залишків, тоді як наша модель тренувалася на всьому різномаїтті білків. Наразі ми не можемо передбачити, який вплив це могло мати на отримані нами результати, тому важливо бути обережними з їх інтерпретацією. Гіпотетично, тренування на специфічному наборі коротких пептидів може зробити модель нечутливою до токсинів, що кардинально відрізнятимуться від тренувальних, а отже лімітувати можливості ToxinPred3.0 для передбачення нових, ще недосліджених видів токсичних білків, тоді як наша модель зможе краще узагальнювати та виявляти невідомі токсини. З іншого боку, тестування нашої моделі на наборі таких коротких послідовностей могло іноді підривати здатність моделі Doc2Vec адекватно оцінювати та

перетворювати амінокислотну послідовність на вектор.

4.1 Використання моделі Doc2Vec для аналізу амінокислотних послідовностей

Застосування методу Doc2Vec для перетворення послідовностей амінокислот виявилося досить ефективним як в плані обчислювальних ресурсів та часу, оскільки вся робота була виконана на ноутбуці середньої цінової категорії, так і в плані якості результуючих векторів, у яких була закодована інформація про послідовність. На нашу думку, подальші дослідження у цьому напрямку за використання потужніших машин можуть бути важливим кроком у розвитку білкових мовних моделей. Наразі серед публікацій, розміщених на агрегаторі наукових статей PubMed, лише 44 згадують Doc2Vec, і ще менше використовують його для аналізу біологічних послідовностей. Це означає, що ця сфера застосування цього методу є погано дослідженою і перспективною.

Також важливо зауважити, що метод PV-DBOW, що був використаний у роботі, зазвичай не використовується самостійно, а працює в синергії з PV-DM, що сильно покращує результати перетворення тексту на вектори. Проте оскільки тренування моделі на основі PV-DM у нас зайняло б набагато більше часу та ресурсів, ми вирішили зупинитися на одному методі. Тому в майбутньому було б цікаво дослідити як ці два методи працюють разом над завданням перетворення амінокислотних послідовностей на вектори. Також цікавою є можливість перевірити результати розбиття послідовностей на фрагменти різних розмірів та вплив цього на якість передбачень. Іншим, непов'язаним із метою дослідження, але теж цікавим застосуванням цього методу може стати передбачення гомологічних послідовностей як альтернатива або доповнення до алгоритму BLAST, оскільки, як було показано, результати передбачень схожих білків навіть за допомогою такої невеликої моделі були дуже схожими до результатів BLASTp.

Проте не можна забувати й про деякі недоліки застосування моделі. По-

перше, як було показано в результатах, невелика частина послідовностей була визначалась як найбільш подібна не до самої себе, а до деякої іншої послідовності, що скоріш за все мало свій вплив на точність передбачень моделі, і нам незрозуміло чому так траплялося. Скоріше за все, це могло бути пов'язано з тим, що модель PV-DBOW не враховує контекст триплетів під час навчання та передбачення, а також має невеликий елемент випадковості. Можливо, цього можна було б запобігти використанням методу PV-DM та отримання як вектора цілої послідовності, так і кожного триплета цієї послідовності, проте це б завадило використанню звичайної нейронної мережі через нестабільну розмірність векторів, що б примусило нас використовувати іншу архітектуру для неї, а також набагато збільшило б час навчання моделі. По-друге, передбачення вектора для послідовності передбачає етап короткого навчання моделі на цій послідовності, і лише потім передбачення, що може призводити до повільної роботи цього методу за великих наборів даних. По-третє, цей метод потребує великої кількості вільної оперативної пам'яті для роботи, особливо за використання комбінації методів PV-DM та PV-DBOW, що може вносити деякі корективи та обмеження.

4.2 Нейронна мережа як елемент моделі

Впровадження штучної нейронної мережі наступною частиною моделі після Doc2Vec покращило її здібності до передбачення, спростило процес класифікації, оцінки її якості та надало необхідної гнучкості у використанні. Проте в існуючій імплементації результат передбачення залежить не стільки від нейронної мережі, скільки від векторів, отриманих за допомогою Doc2Vec. Навчання ШНМ тривало близько 10 епох, після чого ще 10 епох не призводили до значних покращень передбачень моделі та відбувалася рання зупинка навчання. Можливо, причиною такого швидкого тренування була значна залежність схожості функцій білків від схожості їх вхідних векторів, і нейронна мережа швидко виявляла цю закономірність, не виявляючи

достатньої кількості неочевидних взаємозв'язків. Зниження порогового значення на декілька порядків дозволяє виявлення таких слабких сигналів та покращення передбачень моделі. Проте це також спричиняє класифікацію деяких нетоксичних білків як токсичних, що може свідчити про те, що, з одного боку, ці білки не мають токсичної дії, незважаючи на подібність до токсинів, а з іншого — можуть бути попередниками інших токсинів, і є перспективними кандидатами для дослідження.

На нашу думку, використання комбінації методів PV-DM та PV-DBOW для перетворення кожного триплету на певний вектор або застосування ШНМ з архітектурою трансформера та імплементація механізму уваги дозволить враховувати контекст послідовності та вплив різних її ділянок одна на одну, і в свою чергу покращити виконання передбачень, які будуть спиратись не тільки на подібність послідовності, а й на різноманітні взаємодії амінокислот між собою. З іншого боку, така архітектура моделі вимагає набагато більшої кількості обчислювальних ресурсів, часу, а також тренувальних даних для навчання, тому була б надто складною в реалізації в цьому дослідженні.

4.3 Особливості набору даних

Важливо звернути увагу на незбалансованість набору даних Swiss-Prot. Оскільки на кожну токсичну послідовність припадає приблизно по 70 нетоксичних, це може значно впливати на здатність моделі класифікувати білки. В нашому випадку цей вплив можна спостерігати як зміщення оптимальних порогових значень для нейронної мережі.

Для усунення цієї незбалансованості в звичайних даних часто використовуються методи оверсемплінгу та андерсемплінгу, проте в нашому випадку вони не допомогли значно покращити набір даних. Хоч ми і не знаємо причини такого результату, проте гіпотетично це могло бути зумовлено особливостями нашого набору та представлення послідовностей у вигляді векторів.

Можливо, використання методів оверсемплінгу саме на етапі тренування моделі Doc2Vec могло б бути успішнішим, аніж на етапі тренування нейронної мережі. Одним з потенційних рішень могло бути використання токсинів з бази даних TrEMBL неверифікованих послідовностей, проте це могло внести інші проблеми, пов'язані із недостовірністю частини тренувальних даних, тому ми вирішили зупинитися на перевірених білках.

Використання комбінації інструментів RFdiffuson та ProteinMPNN також не призвело до значимого покращення передбачень. Можливо, це пов'язано з тим, що ми використовували однакові параметри часткової дифузії для різних протеїнів. І коли для однієї частини білків вони були оптимальними, то для інших ні, що могло призвести до занадто сильної дифузії та втрати стабільності структури. Також це могло бути пов'язано із тим, що як шаблон для дифузії структур ми використовували передбачення AlphaFold2, тоді як для більшості токсинів ці передбачення могли бути низькоякісними та некоректними через відсутність реальних даних про їх структуру та короткі розміри послідовності. Та з іншого боку, майже повна відсутність експериментально визначених структур та неповне покриття багатьох існуючих моделей токсинів не дозволяли зібрати достатню кількість таких шаблонів для часткової дифузії.

Ще важливо взяти до уваги різницю в довжинах амінокислотних послідовностей токсичних та нетоксичних білків. Більшість токсинів мають короткі послідовності, і ще коротші активні регіони за видалення пропептидів та сигнальних частин. В той час переважна кількість нетоксичних білків мають довжину від 100 до 400 амінокислот. З одного боку, такий розподіл може робити модель упередженою до токсичності коротких пептидів, що можна побачити на рисунку 3.7а, з іншого, ці ж короткі пептиди можуть менш точно перетворюватися на числові вектори моделлю Doc2Vec.

4.4 Подальший розвиток моделі

Отримана модель є лише прототипом, який мав довести, що розроблена нами архітектура дозволяє працювати з амінокислотними послідовностями як із мовою, що має свій синтаксис. Оскільки застосування цієї моделі наразі обмежене лише класифікацією білків на токсичні та нетоксичні, логічним наступним кроком після покращення її точності було б створити регресійну модель, яка передбачатиме силу токсичної дії, а також залежність цієї дії від організму, на який вона діє. Це дозволить не тільки отримати набагато більше інформації про токсини, а й зрозуміти як саме амінокислотна послідовність впливає на специфічність отрути до представників певних таксонів.

Другим кроком розвитку моделі може бути імплементація алгоритму, що дозволить генерувати нові, раніше невідомі амінокислотні послідовності, що потенційно матимуть сильну токсичну дію специфічну до деяких організмів (наприклад, для боротьби зі шкідниками в сільському господарстві, бактеіальними інфекціями людини тощо), не шкодячи іншим істотам. З іншого боку, це дозволить створювати токсини, що матимуть слабку токсичну дію і можуть стати попередниками лікарських препаратів.

Третім кроком може бути імплементація алгоритму еволюції білків, що зможе дозволити нам вивчати можливі шляхи виникнення та розвитку токсинів впродовж їх існування, модифікувати існуючі пептиди для посилення чи послаблення токсичних або терапевтичних характеристик, а також створювати отрути з бажаною активністю та специфічністю.

Висновки

1. Попри незбалансованість набору даних, модель Doc2Vec виявилася ефективним методом перетворення амінокислотних послідовностей на вектори сталої довжини, що зберігають достатню кількість інформації для подальшого навчання.

2. Ідея комбінації моделі Doc2Vec та штучної нейронної мережі для створення класифікатора, що буде відрізняти токсичні протеїни від нетоксичних, виявилася успішною. Навчена модель виявилася достатньо специфічною та чутливою для ефективного виконання завдання класифікації. Зниження порогового значення до 10^{-5} покращило ефективність нейронної мережі в цьому завданні.

3. Наша модель показала збереження здатності до класифікації білків за межами свого тренувального набору даних, а також поступалася у перехресному порівнянні найуспішнішій існуючій моделі ToxinPred3.0 лише на 8% площі під кривою ROC.

4. Інструмент Chroma показав здатність до штучної генерації амінокислотних послідовностей на основі структур токсичних білків, а наша модель була здатна до правильної класифікації 81% з них.

5. Головними лімітуючими факторами для успішності нашої моделі стали незбалансованість набору даних, недосконале перетворення амінокислотної послідовності білків на вектори, субоптимальні параметри навчання та гомологія між деякими токсичними та нетоксичними білками.

Перелік використаних джерел

[1] Bryan G Fry, Kim Roelants, Donald E Champagne, Holger Scheib, Joel D A Tyndall, Glenn F King, Timo J Nevalainen, Janette A Norman, Richard J Lewis, Raymond S Norton, Camila Renjifo, and Ricardo C Rodríguez de la Vega. The toxicogenomic multiverse: convergent recruitment of proteins into animal venoms. *Annu. Rev. Genomics Hum. Genet.*, 10(1):483–511, 2009.

[2] Michael W Pennington, Andrzej Czerwinski, and Raymond S Norton. Peptide therapeutics from venom: Current status and potential. *Bioorg. Med. Chem.*, 26(10):2738–2758, September 2017.

[3] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021.

[4] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, Ramnik J Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.*, 12(1):3168, May 2021.

[5] Dessislava Georgieva, Raghuvir K Arni, and Christian Betzel. Proteome analysis of snake venom toxins: pharmacological insights. *Expert Rev. Proteomics*, 5(6):787–797, December 2008.

[6] Camilla M Whittington and Katherine Belov. Tracing monotreme venom evolution in the genomics era. *Toxins (Basel)*, 6(4):1260–1273, April 2014.

[7] Quentin Kaas and David J Craik. Bioinformatics-aided venomics. *Toxins (Basel)*, 7(6):2159–2187, June 2015.

[8] Florence Jungo, Lydie Bougueleret, Ioannis Xenarios, and Sylvain Poux. The UniProtKB/Swiss-Prot Tox-Prot program: A central hub of integrated venom protein data. *Toxicon*, 60(4):551–557, September 2012.

[9] Kunming Zhou, Weifeng Luo, Tong Liu, Yong Ni, and Zhenghong Qin. Neurotoxins acting at synaptic sites: A brief review on mechanisms and clinical applications. *Toxins (Basel)*, 15(1):18, December 2022.

[10] R Manjunatha Kini. Serine proteases affecting blood coagulation and fibrinolysis from snake venoms. *Pathophysiol. Haemost. Thromb.*, 34(4-5):200–204, 2005.

[11] Nicolas Langenegger, Wolfgang Nentwig, and Lucia Kuhn-Nentwig. Spider venom: Components, modes of action, and novel strategies in transcriptomic and proteomic analyses. *Toxins (Basel)*, 11(10):611, October 2019.

[12] Abdelwahab Khalil, Basem H Elesawy, Tarek M Ali, and Osama M Ahmed. Bee venom: From venom to drug. *Molecules*, 26(16):4941, August 2021.

[13] Erij Messadi. Snake venom components as therapeutic drugs in ischemic heart disease. *Biomolecules*, 13(10), October 2023.

[14] Natalie J Saez and Volker Herzig. Versatile spider venom peptides and their medical and agricultural applications. *Toxicon*, 158:109–126, February 2019.

[15] Michael W Pennington, Andrzej Czerwinski, and Raymond S Norton. Peptide therapeutics from venom: Current status and potential. *Bioorg. Med. Chem.*, 26(10):2738–2758, September 2017.

[16] Eva Ternon and Olivier Thomas. *Toxins from Marine Invertebrates*, page 77. 03 2014.

[17] R Andrew Atkinson, Bruno Kieffer, Annick Dejaegere, Finton Sirockin, and Jean-François Lefèvre. Structural and dynamic characterization of ω-Conotoxin MVIIA: The binding loop exhibits slow conformational exchange,. *Biochemistry*, 39(14):3908–3919, April 2000.

[18] Do Hoang Tu, Tran Van Lang, Pham Cong Xuyen, and Le Mau Long. Predicting the activity of chemical compounds based on machine learning approaches, 2023.

[19] Debleena Paul, Gaurav Sanap, Snehal Shenoy, Dnyaneshwar Kalyane, Kiran Kalia, and Rakesh K Tekade. Artificial intelligence in drug discovery and development. *Drug Discov. Today*, 26(1):80–93, January 2021.

[20] Pedro R Costa, Marcio L Acencio, and Ney Lemke. A machine learning approach for genome-wide prediction of morbid and druggable human genes based on systems-level data. *BMC Genomics*, 11 Suppl 5(S5):S9, December 2010.

[21] Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC Bioinformatics*, 16(1):55, February 2015.

[22] Jérôme Tubiana, Dina Schneidman-Duhovny, and Haim J Wolfson. ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction. *Nat. Methods*, 19(6):730–739, June 2022.

[23] Michela Quadrini, Massimo Cavallin, Sebastian Daberdaku, and Carlo Ferrari. Prosps: Protein sites prediction based on sequence fragments. In Giuseppe Nicosia, Varun Ojha, Emanuele La Malfa, Gabriele La Malfa,

Giorgio Jansen, Panos M. Pardalos, Giovanni Giuffrida, and Renato Umeton, editors, *Machine Learning, Optimization, and Data Science*, pages 568–580, Cham, 2022. Springer International Publishing.

[24] Qifeng Bai, Shuoyan Tan, Tingyang Xu, Huanxiang Liu, Junzhou Huang, and Xiaojun Yao. MolAICal: a soft tool for 3D drug design of protein targets by artificial intelligence and classical algorithm. *Brief. Bioinform.*, 22(3), May 2021.

[25] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, Basile I M Wicky, Nikita Hanikel, Samuel J Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023.

[26] Liyi Yu, Zhaochun Xu, Wangren Qiu, and Xuan Xiao. MSDSE: Predicting drug-side effects based on multi-scale features and deep multi-structure neural network. *Comput. Biol. Med.*, 169(107812):107812, February 2024.

[27] Bartosz A Grzybowski, Sara Szymkuć, Ewa P Gajewska, Karol Molga, Piotr Dittwald, Agnieszka Wołos, and Tomasz Klucznik. Chematica: A story of computer code that started to think like a chemist. *Chem*, 4(3):390–398, March 2018.

[28] Rajesh Tripathy, Sailendra Mahanta, and Subhankar Paul. Artificial intelligence-based classification of breast cancer using cellular images. *RSC Advances*, 4, 01 2014.

[29] Yinzhong Shen, Tingyi Liu, Jun Chen, Xin Li, Li Liu, Jiayin Shen, Jiangrong Wang, Renfang Zhang, Meiyan Sun, Zhenyan Wang, Wei Song, Tangkai Qi,

Yang Tang, Xianmin Meng, Lijun Zhang, Dean Ho, Chih-Ming Ho, Xianting Ding, and Hong-Zhou Lu. Harnessing artificial intelligence to optimize long-term maintenance dosing for antiretroviral-naive adults with hiv-1 infection. *Advanced Therapeutics*, 3(4):1900114, 2020.

[30] Michelle Cung, Branden Sosa, He S Yang, Michelle M McDonald, Brya G Matthews, Annegreet G Vlug, Erik A Imel, Marc N Wein, Emily Margaret Stein, and Matthew B Greenblatt. The performance of artificial intelligence chatbot large language models to address skeletal biology and bone health queries. *J. Bone Miner. Res.*, 39(2):106–115, March 2024.

[31] Mujeen Sung, Minbyul Jeong, Yonghwa Choi, Donghyeon Kim, Jinhyuk Lee, and Jaewoo Kang. Bern2: an advanced neural biomedical named entity recognition and normalization tool. *Bioinformatics*, 38(20):4837–4839, September 2022.

[32] Mehrsa Mardikoraem, Zirui Wang, Nathaniel Pascual, and Daniel Woldring. Generative models for protein sequence modeling: recent advances and future directions. *Briefings in Bioinformatics*, 24(6):bbad358, 10 2023.

[33] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, Shan Tie, Vincent Xue, Sarah C Cowles, Alan Leung, João V Rodrigues, Claudio L Morales-Perez, Alex M Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V Panwar, Fritz Obermeyer, Adam R Root, Andrew L Beam, Frank J Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, November 2023.

[34] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.*, 13(1):4348, July 2022.

[35] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, James S Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.*, 41(8):1099–1106, August 2023.

[36] Seonwoo Min, Seunghyun Park, Siwon Kim, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Pre-training of deep bidirectional protein sequence representations with structural information. *IEEE Access*, 9:123912–123926, 2021.

[37] Siquan Hu, Ruixiong Ma, and Haiou Wang. An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences. *PLoS One*, 14(11):e0225317, November 2019.

[38] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan Gomez, Debora S. Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval, 2022.

[39] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[40] Florence Jungo and Amos Bairoch. Tox-Prot, the toxin protein annotation program of the Swiss-Prot protein knowledgebase. *Toxicon*, 45(3):293–301, March 2005.

[41] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O'Donovan, Isabelle Phan, Sandrine Pilbout, and Michel Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, 31(1):365–370, January 2003.

[42] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 08 2010.

[43] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science and Engineering*, 9(3):90–95, 2007.

[44] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.

[45] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[46] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[47] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[48] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[49] Vitor Werner de Vargas, Jorge Arthur Schneider Aranda, Ricardo dos Santos Costa, Paulo Ricardo da Silva Pereira, and Jorge Luis

Victória Barbosa. Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. *Knowledge and Information Systems*, 65(1):31–57, January 2023.

[50] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[51] Anand Singh Rathore, Akanksha Arora, Shubham Choudhury, Purva Tijare, and Gajendra P. S. Raghava. Toxinpred 3.0: An improved method for predicting the toxicity of peptides. *bioRxiv*, 2023.

[52] Celine Vens, Marie-Noëlle Rosso, and Etienne G. J. Danchin. Identifying discriminative classification-based motifs in biological sequences. *Bioinformatics*, 27(9):1231–1238, 03 2011.