МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра дискретной математики и алгоритмики

ИЛЬИН
Андрей Викторович

ПОИСК БЕЛКОВЫХ ИНТЕРФЕЙСОВ

Дипломная работа

Научный руководитель:
доктор физико-математических наук,
профессор А. В. Тузиков

Консультант:
магистр физико-математических наук,
А. Ю. Хадарович

Допущена к защите
“___” _____ 2018 г

Зав. кафедрой дискретной математики и алгоритмики
доктор физико-математических наук,
профессор В.М. Котов

Минск, 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 7
ГЛАВА 1 ОСНОВНЫЕ СВЕДЕНИЯ О БЕЛКАХ 8
1.1 Основные определения. 8
1.2 Структура и классификация белков. Модель белка 10
1.3 Системы хранения и обработки информации о белках 12
ГЛАВА 2 СУЩЕСТВУЮЩИЕ АЛГОРИТМЫ ПОИСКА БЕЛКОВЫХ ИНТЕРФЕЙСОВ 14
2.1 Алгоритм Йена-Доббса-Хонавара 14
2.2 Алгоритм PredUs 14
2.3 Алгоритм PrISE 16
2.4 Алгоритм Шикича-Томича-Влахович 17
2.5 Алгоритм ProMate 17
2.6 Алгоритм PAIRpred 18
ГЛАВА 3 ОСНОВНЫЕ СВЕДЕНИЯ О СТРОЕНИИ НЕЙРОННЫХ СЕТЕЙ 21
3.1 Общие сведения об архитектуре нейронных сетей 21
3.2 Полносвязные нейронные слои 22
3.3 Функции активации 23
3.4 Стандартные свёрточные нейронные сети 24
3.5 Слой графовой свёртки 26
3.4 Эвристические методы улучшения качества работы сети 27
ГЛАВА 4 ОПИСАНИЕ АЛГОРИТМА ПОИСКА БЕЛКОВЫХ ИНТЕРФЕЙСОВ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ 29
4.1 Постановка решаемой алгоритмом задачи 29
4.2 Графовое представление белковой цепи 29
4.3 Выделение признаков для вершин 29
4.4 Выделение признаков для рёбер 33
4.5 Архитектура нейронной сети 34
4.6 Обработка результата нейронной сети 36

ГЛАВА 5 ПРОВЕДЕНИЕ ЭСПЕРИМЕНТОВ И АНАЛИЗ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ АЛГОРИТМА НА ПРАКТИКЕ 37
5.1 Использованный набор данных 37
5.2 Обучение нейронной сети 38
5.3 Экспериментальная среда и использованные средства 39
5.4 Анализ результатов работы алгоритма 40
ЗАКЛЮЧЕНИЕ 42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 43

РЕФЕРАТ

Дипломная работа, 44 страницы, 26 источников.

Ключевые слова: БЕЛОК-БЕЛКОВЫЕ КОМПЛЕКСЫ, БЕЛКОВЫЕ ИНТЕРФЕЙСЫ, АЛГОРИТМЫ ПРЕДСКАЗАНИЯ, НЕЙРОННЫЕ СЕТИ, МАШИННОЕ ОБУЧЕНИЕ.

Объект исследования: белок-белковые комплексы, принципы взаимодействия белковых молекул.

Цель работы: разработка алгоритма поиска белок-белковых интерфейсов на основе нейронной сети.

Результат: изучены основные сведения о белках и существующие алгоритмы поиска белок-белковых интерфейсов, проанализированы алгоритмы работы нейронных сетей различных типов. На основе полученных знаний разработан, реализован и протестирован алгоритм нахождения белок-белковых интерфейсов на базе нейронной сети. Была показана эффективность созданного алгоритма и потенциал для дальнейших исследований в направлении применения нейронных сетей для решения задачи определения белкового взаимодействия.

Область применения: моделирование структуры белок-белковых комплексов и взаимодействия белковых молекул.

РЭФЕРАТ

Дыпломная праца, 44 старонкі, 26 крыніц.

Ключавыя словы: БЯЛОК-БЯЛКОВЫЯ КОМПЛЕКСЫ, БЯЛКОВЫЯ ІНТЭРФЭЙСЫ, АЛГАРЫТМЫ ПРАДКАЗАННЯ, НЕЙРОНАВЫЯ СЕТКІ, МАШЫННАЕ НАВУЧАННЕ.

Аб'ект даследавання: бялок-бялковыя комплексы, прынцыпы ўзаемадзеяння бялковых малекул.

Мэта работы: распрацоўка алгарытму пошуку бялок-бялковых інтэрфейсаў на аснове нейронавай сеткі.

Вынік: вывучаны асноўныя звесткі пра бялкі і існуючыя алгарытмы пошуку бялок-бялковых інтэрфейсаў, прааналізаваны алгарытмы працы нейронавых сетак розных тыпаў. На аснове атрыманых ведаў распрацаваны, рэалізаваны і пратэставаны алгарытм знаходжання бялок-бялковых інтэрфейсаў на базе нейронавай сеткі. Была паказана эфектыўнасць створанага алгарытму і патэнцыял для далейшых даследаванняў у напрамку прымянення нейронавых сетак для вырашэння задачы вызначэння бялковага ўзаемадзеяння.

Вобласць прымянення: мадэляванне структуры бялок-бялковых комплексаў і ўзаемадзеяння бялковых малекул.

ABSTRACT

Diploma thesis, 44 pages, 26 sources.

Keywords: PROTEIN-PROTEIN COMPLEXES, PROTEIN INTERFACES, PREDICTION ALGORITHMS, NEURAL NETWORKS, MACHINE LEARNING.

Object of research: protein-protein complexes, principles of interaction of protein molecules.

Objective: development of a protein-protein interfaces search algorithm based on a neural network.

The result: basic information about proteins, existing algorithms of searching of protein-protein interfaces and the principle of neural networks operation were studied. Based on the obtained knowledge, an algorithm for finding protein-protein interfaces, which is based on a neural network, was developed, implemented and tested. The effectiveness of the created algorithm and the potential for further research in the direction of using neural networks to solve the problem of determining protein interaction was demonstrated.

The scope: modeling of the structure of protein-protein complexes and interaction of protein molecules.

ВВЕДЕНИЕ

Одним из приоритетных направлений развития экономики Республики Беларусь является внедрение инновационных технологий. Многие из них рождаются на стыке различных наук, что делает весьма актуальным использование междисциплинарного подхода в научных исследованиях. В ходе выполнения данной дипломной работы осуществлялся синтез знаний из областей биологии и информатики.

Изучение строения белков и их свойств является в настоящее время предметом интенсивных исследований многих научных коллективов. В частности, особое место в современной биологии занимает анализ принципов белок-белкового взаимодействия. Важную роль в их изучении играет возможность определять те участки белка, в которых будет происходить данное взаимодействие, например, с целью улучшения качества моделирования структуры белковых комплексов. Такие участки называют интерфейсами.

Поиск интерфейсов может быть существенно облегчен с помощью применения методов информатики, в частности, машинного обучения – быстро развивающейся в наши дни области информатики. В настоящей дипломной работе для решения поставленной задачи требовалось применить нейросетевой подход.

Таким образом, объектом данного исследования являются белковые комплексы, а предметом исследования – белок-белковые взаимодействия.

Цель исследования: разработка алгоритма поиска белковых интерфейсов на основе нейронной сети.

Задачи исследования:

- ознакомиться с различными уровнями строения белка, химическими и физическими свойствами белка, классификацией белков;
- изучить принципы работы нейронных сетей и существующие алгоритмы поиска белковых интерфейсов;
- разработать собственный алгоритм поиска белковых интерфейсов с использованием полученных знаний и проанализировать его точность.

ГЛАВА 1 ОСНОВНЫЕ СВЕДЕНИЯ О БЕЛКАХ

1.1 Основные определения.

Белки – органические соединения, состоящие из аминокислотных остатков, последовательно соединенных пептидной связью (рисунок 1.1).

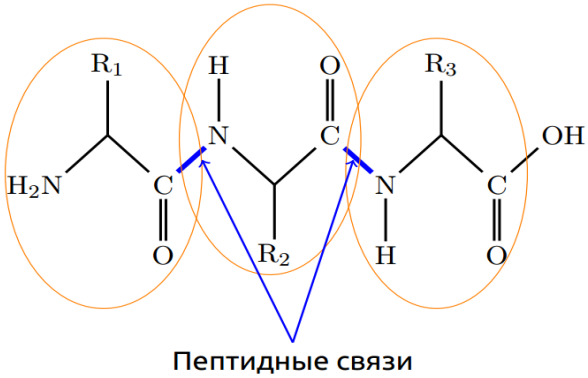

Рисунок 1.1 – Пример строения белка

Пептидная связь – вид химической связи, возникающей в результате взаимодействия карбоксильной группы (–COOH) одной аминокислоты с аминогруппой ($–NH_2$) другой аминокислоты.

Отличительными особенностями данного типа связи является затруднительное вращение вокруг данной связи, валентные углы у атомов C и N примерно равны 120º, расположение атомов каждого пептидного звена в одной плоскости, повышенная прочность относительно других разновидностей амидной связи.

Для каждого из звеньев цепи в связи, таким образом, можно выделить лишь три связи, вокруг которых возможно вращение (рисунок 1.2). Однако, для данных углов также имеются ограничения, которые представляются в виде карты Рамачандрана (рисунок 1.3).

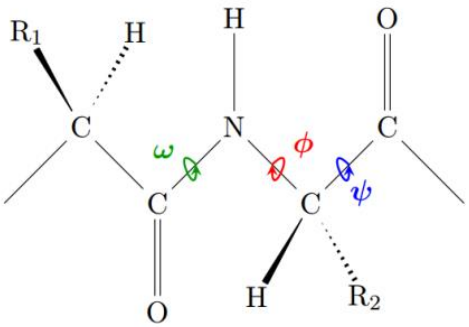

Рисунок 1.2 – Углы вращения вокруг связей в остове белка

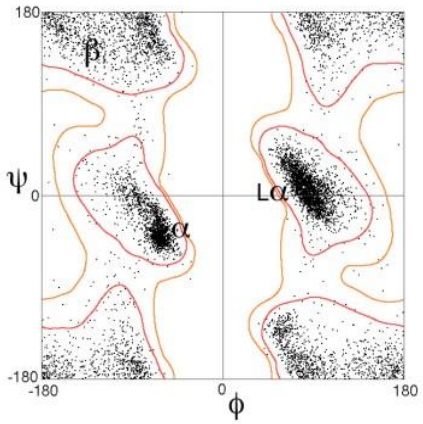

Рисунок 1.3 – Пример карты Рамачандрана для глицина

Всего в природе существует 20 стандартных остатков L-альфа-аминокислот, которые могут входить в состав белков. Аминокислотные остатки, не входящие в стандартный набор, либо не встречаются в составе белков вовсе, либо количество их вхождений в белковые цепи пренебрежимо мало, в связи с чем в большинстве исследований они не учитываются. Каждому из стандартных аминокислотных остатков ставятся в соответствие собственные одно- и трёхбуквенные обозначения (таблица 1.1). При этом, различные системы используют различные обозначения, из-за чего часто возникает необходимость перевода из одной системы в другую.

Название аминокислоты	Трёхбуквенная аббревиатура	Однобуквенная аббревиатура
Аланин	Ala	A
Аргинин	Arg	R
Аспарагин	Asn	N
Аспарагиновая кислота	Asp	D
Валин	Val	V
Гистидин	His	H
Глицин	Gly	G
Глутамин	Gln	Q
Глутаминовая кислота	Glu	E
Изолейцин	Ile	I
Лейцин	Leu	L
Лизин	Lys	K
Метионин	Met	M
Пролин	Pro	P
Серин	Ser	S
Тирозин	Tyr	Y
Треонин	Thr	T
Триптофан	Trp	W
Фенилаланин	Phe	F
Цистеин	Cys	C

Таблица 1.1 – Стандартные аминокислотные остатки и их аббревиатуры

В белках выделяют главную и побочные цепи. Главная цепь – наиболее длинная последовательность соединённых друг с другом атомов, содержащая карбоксильную и аминовую части всех входящих в состав белка аминокислот. Побочные цепи – все другие цепи атомов, образованные радикалами аминокислот (рисунок 1.4).

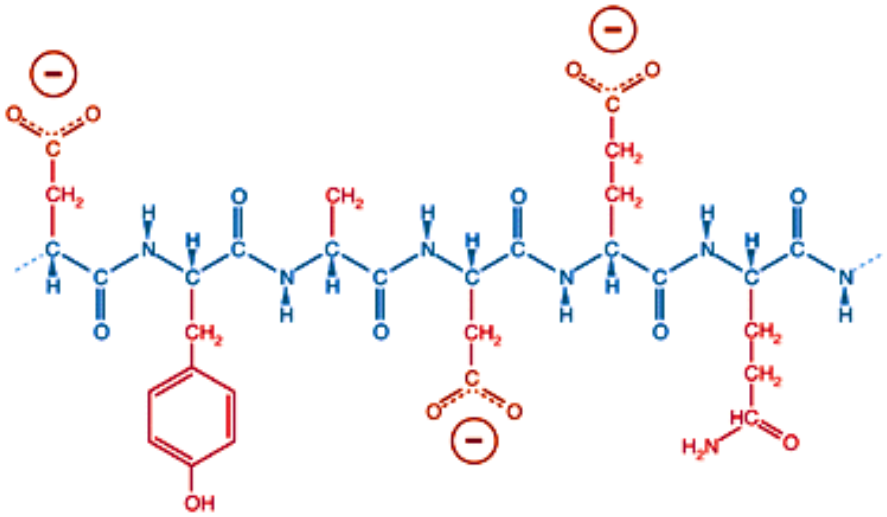

Рисунок 1.4 – Главная (выделенная синим) и побочные (выделены красным) белковые цепи.

Длина белковой цепи может быть различной – от 70 до нескольких тысяч мономеров. Основная белковая цепь, при этом, несимметрична, в связи с чем выделяют два её направления, которые различаются тем, какая группа является последней в цепи в данном направлении – аминная или карбоксильная.

1.2 Структура и классификация белков. Модель белка

Различают 4 уровня структурной организации белков:

- Первичная структура – последовательность аминокислотных остатков в белковой цепи.
- Вторичная структура – простое упорядочивание фрагментов цепи. Наиболее распространены такие типы вторичной структуры, как α-спирали (представляют собой плотные витки вокруг длинной оси) и β-листы (несколько зигзагообразных полипептидных цепей). Примеры вторичной структуры изображены на рисунке 1.5.
- Третичная (трёхмерная) структура – укладка вторичных структур одной полипептидной цепи в глобулу. Данный тип структурной организации стабилизируется различными дополнительными видами связей (дисульфидная, водородная, ионная), а также гидрофобным взаимодействием, которое играет наибольшую роль в построении структуры.
- Четвертичная структура представляет собой совокупность нескольких цепей третичной структуры, объединённых в составе белкового комплекса.

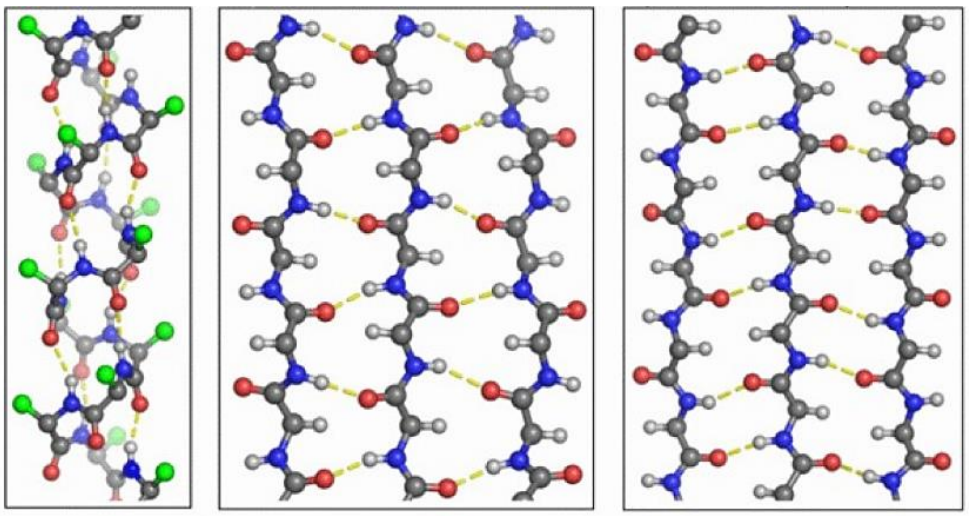

Рисунок 1.5 – Пример вторичной структуры в виде α-спирали (слева) и β-листов (в центре и справа).

В связи с вышенаписанным, наиболее простой моделью белка является последовательность символов, каждый из которых определяет аминокислотный остаток на соответствующей позиции в белке (например, MAGTAVAHTLLPF). Более сложные модели включают также описание типа укладки, структуры петель, расположения боковых групп всех аминокислотных остатков. В общем случае трёхмерная структура белка представляется в виде списка всех входящих в его состав атомов с их координатами. Чаще всего трёхмерная структура изображается в виде «палочковой» модели, на которой изображены атомы и связи между ними, либо в виде поверхности, образованной сферами радиусов Ван-дер-Ваальса вокруг атомов (рисунок 1.6).

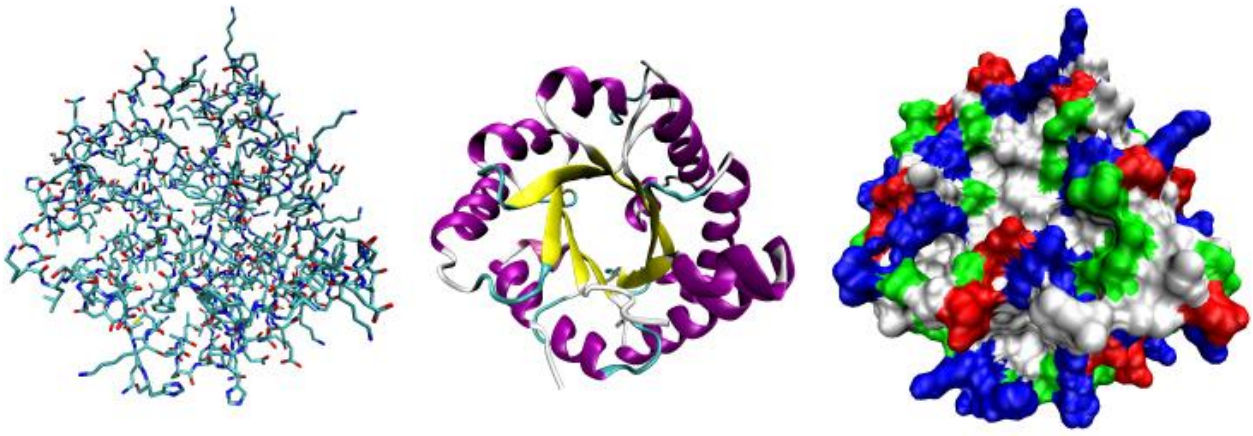

Рисунок 1.6 – Изображение трёхмерной структуры белка 1TIM в виде «палочковой» модели (слева), его вторичных структур (в центре) и контактной поверхности (справа)

Существует классификация белков по общему типу строения и расположению белков в клетке:

- Фибриллярные белки, образующие огромные агрегаты. По структуре такие белки высоко регулярны, их связи основаны чаще всего на взаимодействиях между разными цепями.
- Мембранные белки. Они находятся в мембране, где нет воды, но части их выступают из мембраны в водные растворы. Части таких белков, лежащие внутри клетки, как и фибриллярные белки, высоко регулярны и прошиты водородными связями, но размер этих регулярных частей ограничен толщиной мембраны.
- Водорастворимые, живущие в воде глобулярные белки наименее регулярны, их структура поддерживается взаимодействиями белковой цепи с самой собой, причем особенно важны взаимодействия далеких по цепи, но сблизившихся в пространстве углеводородных (гидрофобных) групп, а также взаимодействиями белковой цепи с кофакторами.
- Разупорядоченные белки – относительно недавно выделенный класс белков, не обладающих постоянной трехмерной структурой, либо приобретающих ее только на короткое время.

1.3 Системы хранения и обработки информации о белках

Наиболее известной базой данных является Брукхейвенский банк пространственных структур (Protein Data Bank, PDB), в котором содержится информация о пространственной структуре белков. Данная база поддерживается совместно университетом Rutgers (США, штат Нью-Джерси); EBI (Англия) и BIRD (Institute for Bioinformatics Research and Development, Япония). Вся информация доступна бесплатно через интернет [22].

Идентификатор соединения в PDB – четырехзначный код, состоящий из цифр и букв латинского алфавита. Структура белка записывается в файл с расширением *.pdb в строго определенном формате. Наибольший интерес в данных файлах представляет информация в секции «ATOM» (рисунок 1.7), в которой содержатся номера и имена аминокислотных остатков, названия и трехмерные координаты атомов, название цепи белка и некоторая другая информация.

Другой базой данных, содержащей классифицированную и точно аннотированную информацию о последовательностях белков, является Uniprot [24]. Во многих случаях имеется соответствие между данными в PDB и Uniprot, однако в некоторых случаях информация немного отличается. Довольно много информации содержится также в базе Национального центра Биотехнологической информации (NCBI) [14].

```
ATOM      1  N   HIS A  17     -12.690   8.753   5.446  1.00 29.32           N
ATOM      2  CA  HIS A  17     -11.570   8.953   6.350  1.00 21.61           C
ATOM      3  C   HIS A  17     -10.274   8.970   5.544  1.00 22.01           C
ATOM      4  O   HIS A  17     -10.193   8.315   4.491  1.00 29.95           O
ATOM      5  CB  HIS A  17     -11.462   7.820   7.380  1.00 23.64           C
ATOM      6  CG  HIS A  17     -12.551   7.811   8.421  1.00 21.18           C
ATOM      7  ND1 HIS A  17     -13.731   7.137   8.194  1.00 28.94           N
ATOM      8  CD2 HIS A  17     -12.634   8.384   9.644  1.00 21.69           C
ATOM      9  CE1 HIS A  17     -14.492   7.301   9.267  1.00 27.01           C
ATOM     10  NE2 HIS A  17     -13.869   8.058  10.168  1.00 22.66           N
ATOM     11  N   ILE A  18      -9.269   9.660   6.089  1.00 19.45           N
ATOM     12  CA  ILE A  18      -7.910   9.377   5.605  1.00 18.67           C
ATOM     13  C   ILE A  18      -7.122   8.759   6.749  1.00 16.24           C
ATOM     14  O   ILE A  18      -7.425   8.919   7.929  1.00 18.80           O
```

Рисунок 1.7 – Пример записей в разделе «ATOM» PDB-файла.

В ходе реализации разработанного алгоритма часто возникала необходимость проверки промежуточных данных. Для визуализации трёхмерных структур, при этом, использовалась программа PyMol. Данная программа распространяется бесплатно и имеет открытый исходный код [23].

ГЛАВА 2 СУЩЕСТВУЮЩИЕ АЛГОРИТМЫ ПОИСКА БЕЛКОВЫХ ИНТЕРФЕЙСОВ

Большинство систем, используемых в данный момент для поиска белковых интерфейсов, основываются на статистических методах машинного обучения, в частности, методе опорных векторов, случайном лесе, методе ближайших соседей и т. п. [2, 7]. При этом, большой упор при разработке таких систем делается на выбор признаков, что и было использовано далее при создании собственного алгоритма.

2.1 Алгоритм Йена-Доббса-Хонавара

Одним из наиболее простых алгоритмов предсказания белок-белковых интерфейсов является алгоритм Йена-Доббса-Хонавара [26], который состоит из двух этапов. Данный алгоритм использует лишь информацию о первичной структуре белка и основывается на том факте, что принадлежащие к белковому интерфейсу аминокислотные остатки, в подавляющем большинстве случаев, образуют кластеры в белковой последовательности.

На первом этапе, используя в качестве классификатора метод опорных векторов с полиномиальным ядром, осуществляется первоначальное предсказание принадлежности аминокислотных остатков к белок-белковому интерфейсу. Выделяемые на данном этапе признаки – типы рассматриваемого аминокислотного остатка и восьми ближайших его соседей (по 4 с каждой стороны цепи). Типы остатков кодируются унитарным кодом, образуя, тем самым, 180-битный вектор признаков для каждого остатка в белковой цепи.

Второй этап служит для улучшения предсказания путём избавления от шумов. В качестве входных данных для каждого аминокислотного остатка выступают предсказанные на предыдущем этапе значения вероятностей принадлежности к белок-белковому интерфейсу рассматриваемого остатка и восьми ближайших его соседей. В качестве классифицирующего алгоритма на данном этапе используется байесовская сеть доверия.

2.2 Алгоритм PredUs

Другим алгоритмом поиска белковых интерфейсов, использующих методы машинного обучения, является PredUs [17]. На первом этапе работы для каждого находящегося на поверхности белка аминокислотного остатка осуществляется выделение 31 признака, которые можно разделить на две подгруппы.

Первая подгруппа – площадь доступной поверхности белка (рисунок 2.1). Данная характеристика вычисляется для рассматриваемого аминокислотного остатка и 14 его ближайших соседей, образуя, таким образом, 15 различных признаков. Для выбора ближайших соседей в данном алгоритме используется отношение пространственной близости, т. е. ближайшие соседи – другие аминокислотные остатки, а в качестве меры расстояния принимается минимальное из расстояний до входящих в их состав атомов.

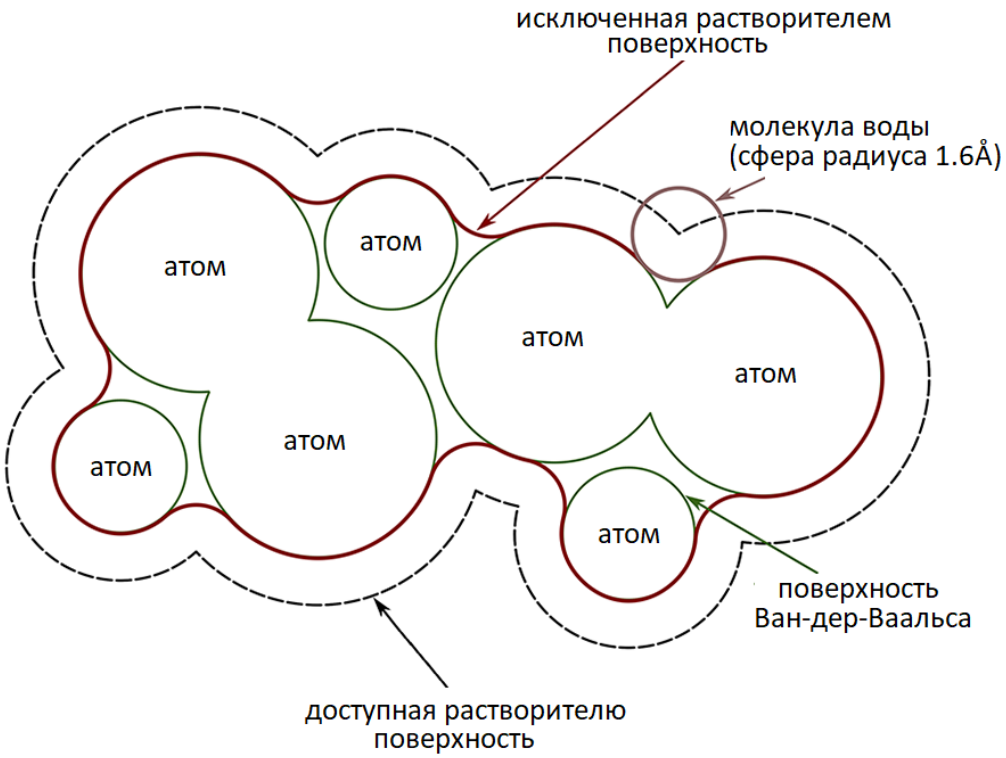

Рисунок 2.1 – Схема поверхности белковой молекулы в разрезе

Вторая группа признаков – частота встречаемости в интерфейсах аминокислотных остатков соответствующего типа. Данная характеристика также образует группу из 15 признаков: для рассматриваемого остатка и 14 наиболее пространственно близких к нему. Последним признаком является наибольшая из частот взаимодействия среди всех аминокислотных остатков обрабатываемого белка.

На втором этапе полученные наборы признаков обрабатываются методом опорных векторов: первым делом происходит переход к векторам высокой размерности с использованием радиальной базисной функции, а далее осуществляется попытка построения разделяющей гиперплоскости в полученном векторном пространстве.

В качестве меры вероятности принадлежности остатку используется расстояние до построенной гиперплоскости (расстояние берётся со знаком,

который определяет по какую из сторон от плоскости расположен каждый из векторов). По умолчанию, в PredUs все аминокислотные остатки, получившие положительную меру вероятности, считаются принадлежащими интерфейсу, однако порог принадлежности, при желании, может быть скорректирован пользователем.

2.3 Алгоритм PrISE

Следующим изученным алгоритмом поиска белковых интерфейсов был алгоритм PrISE [16]. По аналогии с предыдущим, данный алгоритм оперирует так называемыми «структурными элементами» – аминокислотными остатками на поверхности белка и их пространственными соседями. Основные отличия данного алгоритма заключены в используемых характеристиках и методе предсказания.

Для каждого из структурных элементов определяются следующие характеристики:

- название центрального аминокислотного остатка рассматриваемого структурного элемента;
- площадь доступной поверхности для центрального остатка рассматриваемого элемента;
- суммарная площадь доступной поверхности для всех аминокислотных остатков, входящих в структурный элемент;
- 36-группная количественная гистограмма атомных номенклатур, описывающих композицию типов атомов данного структурного элемента (α-углероды, β-углероды и т.п.).

В качестве биохимической основы для такого выбора характеристик лежит тот факт, что структурные элементы, имеющие схожие центральные аминокислотные остатки и площади доступной поверхности, имеют одинаковое строение, а элементы, похожие друг на друга по композиции входящих в состав поверхности атомов – схожие физико-химические свойства.

Для каждого запроса алгоритм PrISE первым делом преобразует белковую цепь в набор структурных элементов. Далее для каждого структурного элемента происходит поиск других похожих на него элементов в базе данных. В итоге, на основе информации о белках, содержащих найденные «похожие» структурные элементы, происходит предсказание принадлежности к интерфейсу для центрального аминокислотного остатка рассматриваемого структурного элемента.

Существует три разновидности данного алгоритма, различающихся механизмом поиска: использующая степени близости непосредственно

структурных элементов (PrISEL), основанная на близости поверхности всего белка (PrISEG) и использующий комбинированный подход (PrISEC).

2.4 Алгоритм Шикича-Томича-Влахович

Попытку использовать одновременно информацию о первичной, вторичной и третичной структурах предприняли в своей работе Миля Шикич, Санжа Томич, и Кристиан Влахович [20].

Информация о первичной структуре, извлекаемая в данном алгоритме, совпадает с первой из рассмотренных работ – типы аминокислотных остатков вокруг рассматриваемого. Ширина скользящего окна, при этом, устанавливалась равной девяти остаткам.

Что касается вторичной структуры, то тут всё очевидно – в качестве признака использовалось категориальное значение типа элемента вторичной структуры, к которому принадлежит рассматриваемый аминокислотный остаток.

Признаки, описывающие третичную структуру – все выходные данные, получаемые для белковой молекулы с помощью алгоритма PSAIA [19]. В число данных признаков входят доступная растворителю площадь поверхности, значение гидропатии Кайта-Дулитла, а также индексы глубины и выступа, о которых будет более подробно рассказано позже.

Все перечисленные признаки используются в качестве входных данных для случайного леса – метода машинного обучения, который используется для предсказания белок-белкового интерфейса в данном алгоритме. При этом, авторы в качестве одного из параметров случайного леса используют ограничение максимальной его глубины, тем самым предоставляя компьютеру право самостоятельно отбрасывать некоторые из признаков в каждом решающем дереве леса.

2.5 Алгоритм ProMate

Довольно большую работу в области поиска признаков, имеющих большое значение при поиске белок-белковых интерфейсов, выполнили создатели алгоритма ProMate [15].

В ходе экспериментов был предложен ряд признаков, для каждого из которых были построены сравнительные диаграммы. Данные диаграммы позволяют сделать вывод о наличии или отсутствии связи между вероятностью принадлежности аминокислотных остатков к белковому интерфейсу и значением рассматриваемого показателя. Например, было установлено, что большая часть остатков, являющихся элементами α-спиралей, не принадлежат к

белковому интерфейсу, а, скажем, для β-листов и изгибов ситуация противоположная – они преобладают в интерфейсах (рисунок 2.2).

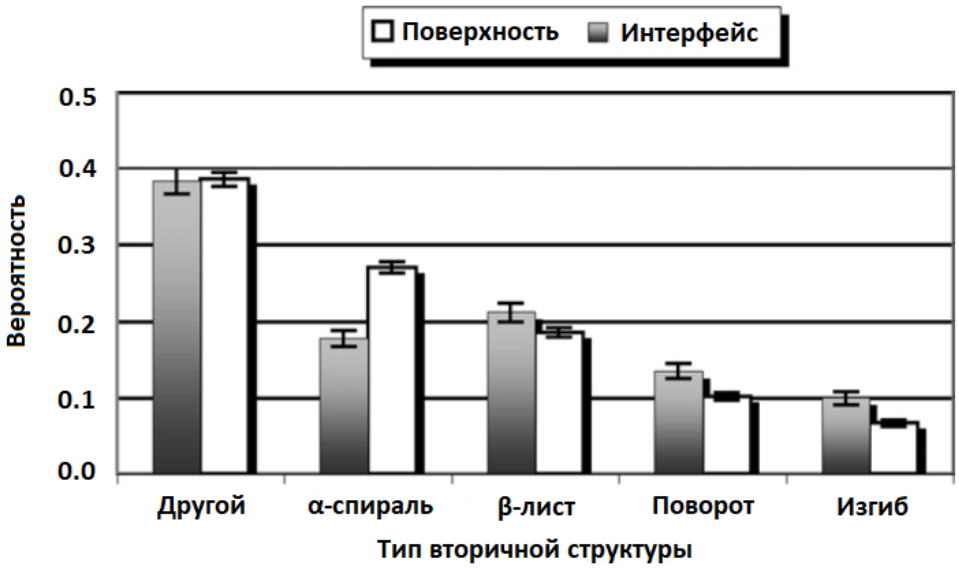

Рисунок 2.2 – Вероятность принадлежности различных вторичных структур к интерфейсу

Составленный в итоге набор важных признаков, по значениям которых осуществляется предсказание методом максимального правдоподобия, включает следующие характеристики:

- тип вторичной структуры;
- тип побочной цепи (наибольший интерес представляют полярные и ароматические радикалы);
- гидрофобность аминокислотных остатков;
- температурный фактор (B-фактор);
- количество молекул воды вблизи рассматриваемого остатка (при наличии соответствующих данных кристаллографии);

2.6 Алгоритм PAIRPred

Одним из наиболее новых алгоритмов поиска белок-белковых интерфейсов является алгоритм PAIRPred [13]. Данный алгоритм использует как пространственные характеристики, так и информацию о первичной белковой структуре для осуществления предсказания взаимодействия аминокислотных остатков.

Первая группа используемых признаков, по аналогии с рассмотренными в начале главы алгоритмами, базируется на информации о площади доступной растворителю поверхности белковой молекулы. Отличием используемых в PAIRPred показателей от применяемых в ранее рассмотренных версиях

является то, что вместо точного значения площади поверхности в качестве признака используется относительный вариант – какая часть от всей поверхности, образованной атомами аминокислотного остатка, является доступной для растворителя.

Второй характеристикой аминокислотных остатков является их глубина относительно поверхности белка – минимальное из расстояний от атомов рассматриваемого остатка до какого-нибудь из атомов, расположенных на поверхности белковой молекулы. Перед дальнейшим использованием данный признак нормируется так, чтобы все значения попали в отрезок от 0 до 1.

Следующая группа признаков основывается на работе Томаса Хамелрика [9], который показал, что геометрические и физико-химические свойства регионов белка в направлении побочной цепи и противоположном направлении могут в значительной степени отличаться. Основываясь на этом, авторы алгоритма PAIRPred решили ввести группу признаков, характеризующих композицию типов аминокислотных остатков, разделив их на две подгруппы – в зависимости от направления. Каждый отдельный признак из данной группы равен количеству аминокислотных остатков конкретного типа, расположенных в соответствующей подгруппе полусфере радиуса 8.0 Å. Для данной группы признаков, получившей название «Half Sphere Amino Acid Composition» («HSAAC»), перед дальнейшим использованием, как и ранее, выполняется процесс нормализации значений.

Четвёртая группа признаков называется «индекс выступа». Для их определения вокруг каждого из атомов белковой цепи строится сфера радиуса 10.0 Å, подсчитывается объём свободного от каких-либо атомов пространства, а затем для каждого аминокислотного остатка в цепи происходит агрегация значений, полученных для входящих в его состав атомов, и нормализация этих значений.

Последняя группа признаков основана на информации о первичной структуре белка, т. е. последовательности его аминокислотных остатков. Используя алгоритм PSI-BLAST [8], на основе информации из баз данных о белках было выполнено вычисление «позиционно-зависимой оценивающей матрицы» (Position Specific Scoring Matrix, PSSM) и «позиционно-зависимой частотной матрицы» (Position Specific Frequency Matrix, PSFM). Далее, на основе значений в данных предпросчитанных матрицах, методом скользящего окна шириной в 11 аминокислотных остатков для каждого из элементов белковой последовательности происходит вычисление 220-мерного вектора признаков, который получил название «вектор профильных признаков».

В отличие от рассмотренных ранее алгоритмов, PAIRPred осуществляет предсказание не напрямую для белковой цепи, а для пар аминокислотных остатков, где один из элементов пары принадлежит белку-рецептору, а второй –

белку-лиганду. Для осуществления данного предсказания используется метод опорных векторов, а далее, при необходимости, можно выполнить переход от маркеров взаимодействия пар аминокислотных остатков к маркерам принадлежности интерфейсу элементов каждой отдельно взятой белковой цепи.

ГЛАВА 3 ОСНОВНЫЕ СВЕДЕНИЯ О СТРОЕНИИ НЕЙРОННЫХ СЕТЕЙ

Нейронная сеть — это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир программирования из биологии и благодаря ей компьютерные системы обретают способность анализировать и запоминать различную информацию.

В ходе подготовительного этапа дипломного исследования были на углубленном уровне изучены принципы работы нейронных сетей и другая литература по данной теме.

3.1 Общие сведения об архитектуре нейронных сетей

В качестве вычислительной единицы в нейронных сетях выступают нейроны. Большая часть нейронов в сети, которые получают информацию от некоторых других нейронов, обрабатывают её и передают результат последующим нейронам, называют «скрытыми». Кроме того, существуют «входные» и «выходные» нейроны, отвечающие за считывание входных данных и выдачу результатов работы нейронной сети соответственно.

В общем случае нейронная сеть состоит из слоя входных нейронов, нескольких слоёв скрытых нейронов и слоя выходных нейронов. Каждый слой при этом является набором однотипных нейронов, использующих в качестве источника входных данных результат с одного или нескольких предшествующих слоёв. Конкретный алгоритм определения наличия связей между нейронами (синапсов), а также механизма обработки данных внутри нейронов зависит от типа слоя.

Под архитектурой нейронной сети понимают чёткое задание количества слоёв, их типа и взаимосвязи между ними (топологии сети). Архитектуру нейронной сети, таким образом, можно представить в виде схемы, пример которой приведён на рисунке 3.1.

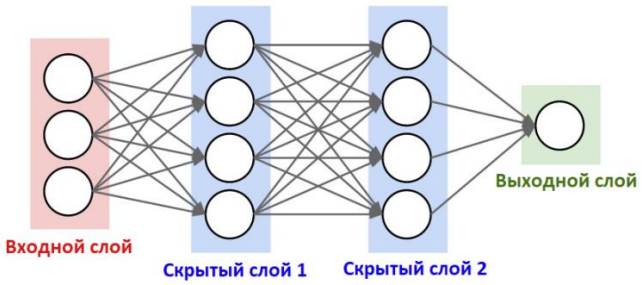

Рисунок 3.1 – Пример схемы нейронной сети
с двумя скрытыми слоями

3.2 Полносвязные нейронные слои

Наиболее простые архитектуры сетей используют в качестве обучаемой единицы нейроны линейных преобразований. Полносвязный слой является реализацией данного подхода, где каждый нейрон слоя имеет связь с каждым из выходных сигналов предыдущего слоя. При этом каждому из синапсов приписывается некоторый вес w_{ij}, а выходной сигнал нейрона определяется по формуле (1).

$$y_h = w_{1h}x_1 + \cdots + w_{jh}x_j + \cdots + w_{nh}x_n + b_h \qquad (1)$$

Нетрудно видеть, что в таком случае итоговое преобразование, выполняемое полносвязным слоем, можно задать в матричном виде (2).

$$\mathrm{y} = \mathrm{Wx} + \mathrm{b} \qquad (2)$$

При этом, веса в матрице W являются переменными коэффициентами, подбор которых как раз и происходит в ходе процесса «обучения» нейронной сети. Вектор b в данном случае необходим для того, чтобы получить полноценное линейное преобразование, а именно, реализовать, помимо поворотов, также операцию сдвига. Данный элемент можно представить через дополнительный источник постоянного входного сигнала. Общая схема полносвязной нейронной сети в таком случае имеет вид, представленный на рисунке 3.2.

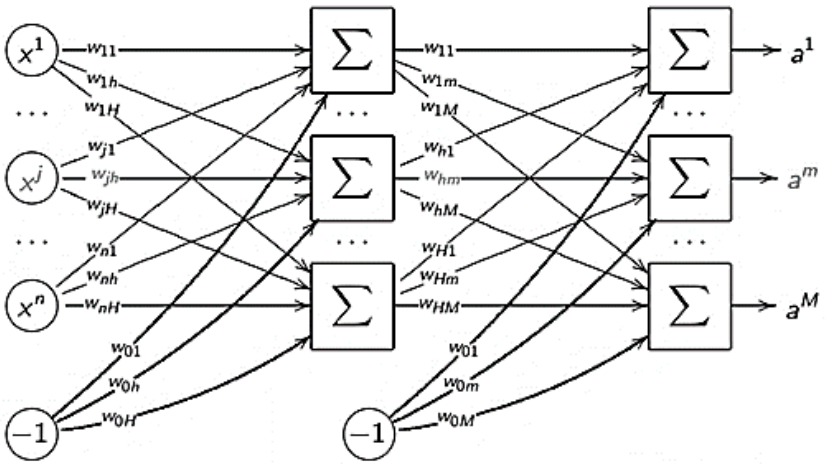

Рисунок 3.2 – Общая схема полносвязного слоя нейронной сети

У представленного типа сетей имеется два основных недостатка: ограниченность множества воссоздаваемых функций множеством линейных преобразований и огромное количество параметров (каждый полносвязный

слой имеет (N+1)*M обучаемых параметров, где N и M – размеры входного и выходного векторов сигналов соответственно).

3.3 Функции активации

Основной проблемой нейронных сетей, состоящих лишь из ранее описанных слоёв, является невозможность определения сложных взаимосвязей и, соответственно, неспособность выявления высокоуровневых признаков. Действительно, независимо от количества слоёв в такой сети, они будут равноценны однослойной сети, ибо композиция линейных преобразований по-прежнему является линейным преобразованием.

Для получения нелинейных преобразований используют различные функции активации. Такого рода функции применяются к выходным сигналам слоёв линейных преобразований, изменяя их по некоторой фиксированной нелинейной формуле, зависящей от используемого типа слоя активации.

Наиболее часто используемыми функциями активации являются сигмоида (рисунок 3.3) и ReLU (рисунок 3.4).

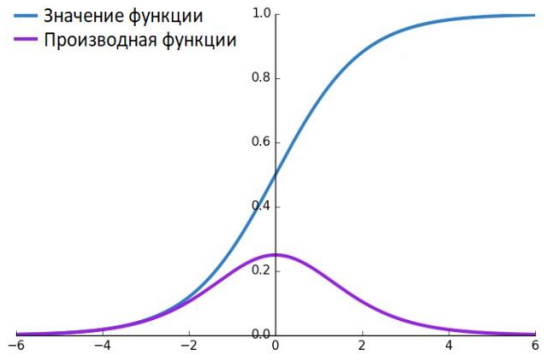

Рисунок 3.3 – Сигмоидная функция активации и её производная

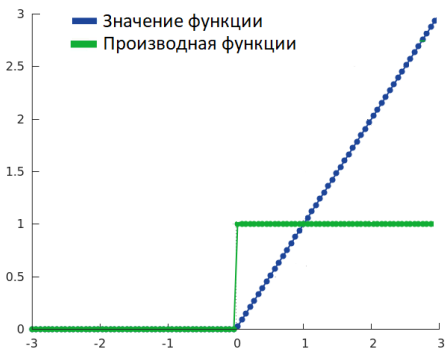

Рисунок 3.4 – Функция активации ReLU и её производная

Как нетрудно заметить исходя из графика, сигмоидная функция, активно использовавшаяся первоначально, почти неприменима в многослойных сетях, ведь в силу близости к нулю за пределами небольшого отрезка, в большинстве случаев будет происходить так называемое «затухание градиента», что будет препятствовать эффективному обучению глубоких слоёв.

При использовании функции ReLU данная проблема частично решается в силу ненулевого градиента на целом полупространстве. Тем не менее, на второй половине пространства градиент всё равно равен нулю. На практике использование данной функции приводит к хорошим результатам в сетях с большим числом слоёв, однако существуют некоторые её модификации (например, PReLU), которые могут ещё немного улучшить результаты.

3.4 Стандартные свёрточные нейронные сети

Для решения второй проблемы полносвязных нейронных сетей (количество параметров) вместо операции линейного преобразования будем применять к входным данным операцию «свёртка».

Выполнение свёртки происходит с помощью обхода входных данных методом скользящего окна. Для каждой позиции окна происходит поэлементное перемножение входных сигналов со значениями некоторой единой для всех позиций матрицы, называемой ядром свёрточного преобразования:

$$z_{ij} = \sum_{(a,b) \in N_{ij}} W_{ab}\, x_{i+a,j+b} \qquad (3)$$

где x – значение входного сигнала, z – выходного, а W – ядро свёртки.

Схематичное представление процесса применения операции свёртки к двумерной входной матрице представлено на рисунке 3.5.

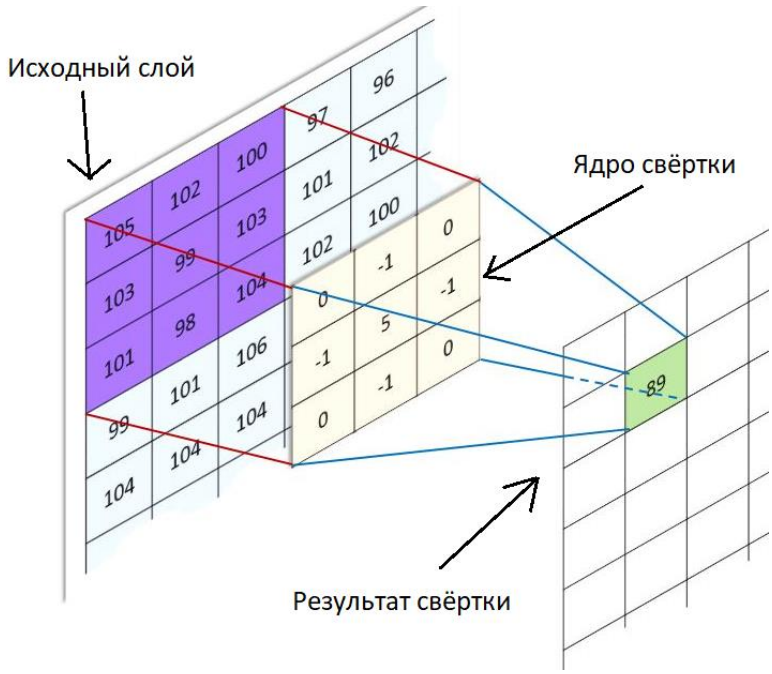

Рисунок 3.5 – Схематичное представление процесса применения операции свёртки

Под стандартным свёрточным слоем понимают изначальный вариант использования описанной идеи, применяемый, в основном, для анализа изображений и видео. В этом случае входной и выходной сигналы представляют собой трёхмерные матрицы с H*W*D элементами, где W и H – ширина и высота изображения соответственно, а D – количество каналов или

глубина изображения. При этом, как правило, на каждом слое свёрточной сети происходят вычисления для сразу нескольких различных ядер свёртки. Полученные в этом случае результаты рассматриваются как отельные каналы при формировании выходной матрицы, называемой также многоканальной картой признаков.

Переменными коэффициентами, подбираемыми в процессе обучения, являются элементы ядра свёртки. В силу довольно небольшого размера ядра (зачастую используются фильтры размеров 5x5 или 3x3), количество параметров сети значительно уменьшается даже при относительно большом количестве каналов выходной карты. В совокупности с идеей одновременного применения целого набора различных ядер, данный подход позволяет достигать более высокоточных результатов за значительно более короткое время.

Помимо непосредственно свёрточных слоёв, важную роль в свёрточных нейронных сетях играют слои, реализующие операцию субдискретизации (пулинга). Каждый из таких слоёв уменьшает размер карт признаков для каждого из каналов. При этом имеется возможность задавать различные алгоритмы субдискретизации. Помимо регулировки размера и величины сдвигов окна пулинга, определяющих степень уменьшения размеров карты признаков, могут также использоваться различные методы получения субдискретизированной карты (выбор максимума, среднее значение и т. п.).

Начальная часть свёрточных нейронных сетей, как правило, представляет собой блоки из последовательно применяемых слоёв свёртки, функций активации, пулинга и, возможно, эвристик. Со смысловой точки зрения, такая архитектура позволяет осуществить распознавание сложных иерархий признаков путём выделения в более глубоких слоях большого числа малоразмерных высокоуровневых признаков.

Конечная часть свёрточной сети, как правило, представляет собой полносвязную нейронную подсеть, состоящую из полносвязных слоёв, слоёв активации и дополнительных эвристик. Входным сигналом для данной подсети является совокупность большого числа абстрактных признаков, выявленных в свёрточной части сети. На практике выходные данные свёрточной подсети преобразуют в вектор для удобства использования полносвязными слоями.

Типичная архитектура свёрточной нейронной сети представлена на рисунке 3.6.

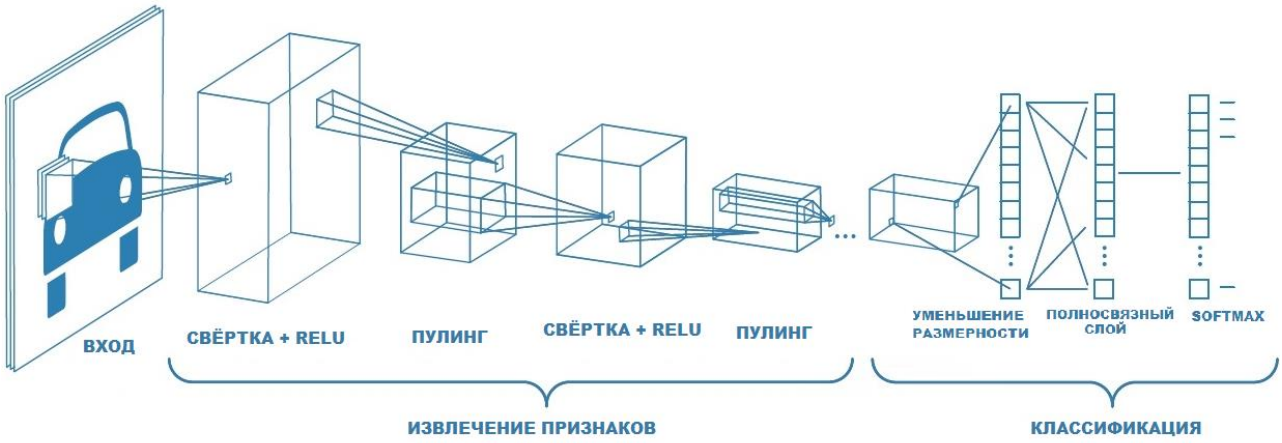

Рисунок 3.6 – Архитектурная схема свёрточной нейронной сети для классификации изображений

3.5 Слой графовой свёртки

В ходе развития направления нейронных сетей описанная в предыдущем пункте идея свёрточного подхода была адаптирована и на другие модели представления данных, в частности, на графовую модель. В рамках данного исследования использовались вершинный и вершинно-рёберный типы графовых свёрток.

Результат вершинной свёртки зависит лишь от значений признаков непосредственно в рассматриваемой и соседних к ней вершинах и может быть вычислен по формуле (4):

$$z_i = W^C x_i + \frac{1}{|N_i|} \sum_{j \in N_i} W^V x_j + b, \qquad (4)$$

где z_i – результат свёртки в вершине i, x_i – вектор признаков вершины i, N_i – множество вершин, соединённых с вершиной i ребром, b – постоянная добавка свёрточного фильтра, W^V – матрица ядра свёртки для признаков соседних вершин, а W^C – матрица ядра для признаков в рассматриваемой вершине.

Результат вершинно-рёберной свёртки зависит дополнительно от значений характеристик, приписанных к рёбрам и может быть вычислен по формуле (5):

$$z_i = W^C x_i + \frac{1}{|N_i|} \sum_{j \in N_i} W^V x_j + \frac{1}{|N_i|} \sum_{j \in N_i} W^E A_{ij} + b, \qquad (5)$$

где A_{ij} – вектор признаков, соответствующих ребру (i, j), а W^E – матрица ядра свёртки для рёберных признаков.

3.4 Эвристические методы улучшения качества работы сети

Для улучшения точности работы нейронной сети могут использоваться различные эвристические слои.

Примером такого слоя является слой типа Dropout. Идея его использования заключается в том, что на этапе обучения часть элементов вектора сигналов можно заменить на нули с некоторой фиксированной вероятностью. В процессе использования уже натренированной сети никакие из сигналов не заменяются нулём, однако осуществляется поправка значений сигналов на вероятность сохранения при обучении.

Со смысловой точки зрения, данный подход равносилен отключению случайных нейронов на этапе обучения для симулирования ситуации отсутствия части информации. Результат применения данного метода к полносвязной сети на этапе обучения представлен на рисунке 3.7.

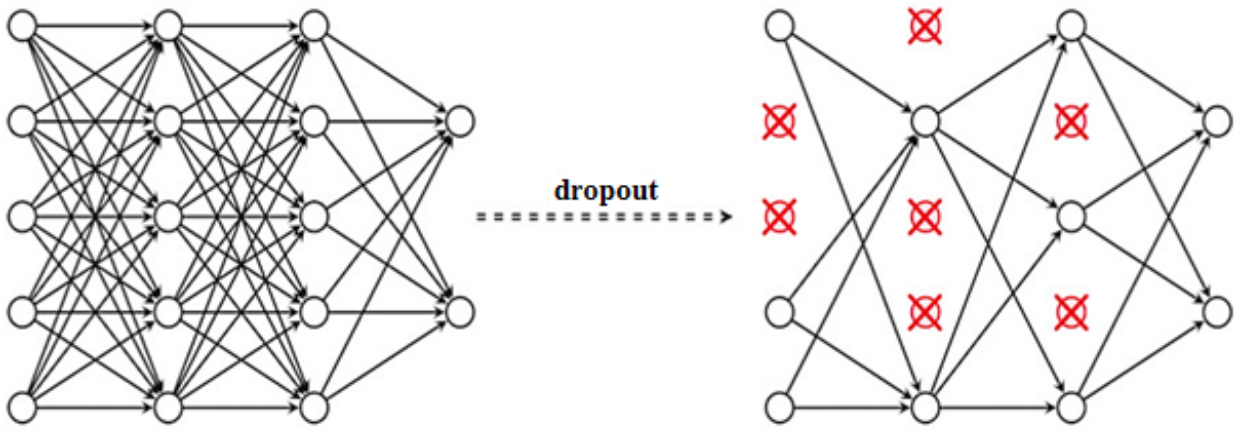

Рисунок 3.7 – Результат применения эвристики Dropout к полносвязной сети.

Использование описанного подхода позволяет повысить качество работы сети, ибо сеть становится более устойчивой к случайным шумам во входных данных.

Другой эвристикой, позволяющей увеличить устойчивость нейронной сети, является пакетная нормализация. Суть данного метода заключается в том, что полученные с предыдущего слоя выходные сигналы для всех элементов обрабатываемого пакета сдвигаются и масштабируются.

Исходно сдвиг значений каждого отдельного признака происходит на величину среднего значения данного признака среди всех входящих в состав

пакета объектов, а масштабирование происходит путём деления на величину стандартного отклонения значений данного признака.

Впоследствии, в процессе обучения нейронной сети, происходит корректировка этих двух параметров для достижения более хороших результатов.

ГЛАВА 4 ОПИСАНИЕ АЛГОРИТМА ПОИСКА БЕЛКОВЫХ ИНТЕРФЕЙСОВ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ

4.1 Постановка решаемой алгоритмом задачи

Входными данными является информация о двух независимых белковых цепях до начала их взаимодействия (в несвязанном состоянии): последовательность аминокислотных остатков, типы и координаты всех атомов. Данная информация предоставляется в виде двух файлов в формате PDB, соответствующих данным о белке-рецепторе и белке-лиганде.

На основе предоставленных сведений требуется определить список аминокислотных остатков в обеих цепях, которые будут участвовать во взаимодействии, или, другими словами, список аминокислотных остатков, образующих интерфейс связывания белков. В рамках данного исследования в качестве порога расстояния принадлежности интерфейсу было использовано значение 6Å, наиболее часто используемое на практике.

4.2 Графовое представление белковой цепи

В качестве модели белка, используемой в данной работе, было выбрано графовое представление белковой молекулы. Вершинами графа являются аминокислотные остатки цепи, которым приписываются наборы соответствующих им признаков. Рёбрами графа являются отношения пространственной близости соответствующих соединяемым вершинам аминокислотных остатков, которым также соответствуют некоторые признаки. В данной работе для каждой вершины в граф добавлялось 20 рёбер, соединяющих соответствующий остаток с наиболее близкими к нему элементами белковой последовательности.

4.3 Выделение признаков для вершин

Для каждого аминокислотного остатка выделяется набор из числовых и категориальных признаков, который впоследствии преобразовывается в вектор признаков, а затем нормируется и приписывается к вершине графа, соответствующей аминокислотному остатку. При этом для кодирования категориальных признаков применяется унитарный код – набор из фиксированного числа признаков, каждый из которых равен нулю, за исключением одного, соответствующего нужной категории, который равен единице (рисунок 4.1).

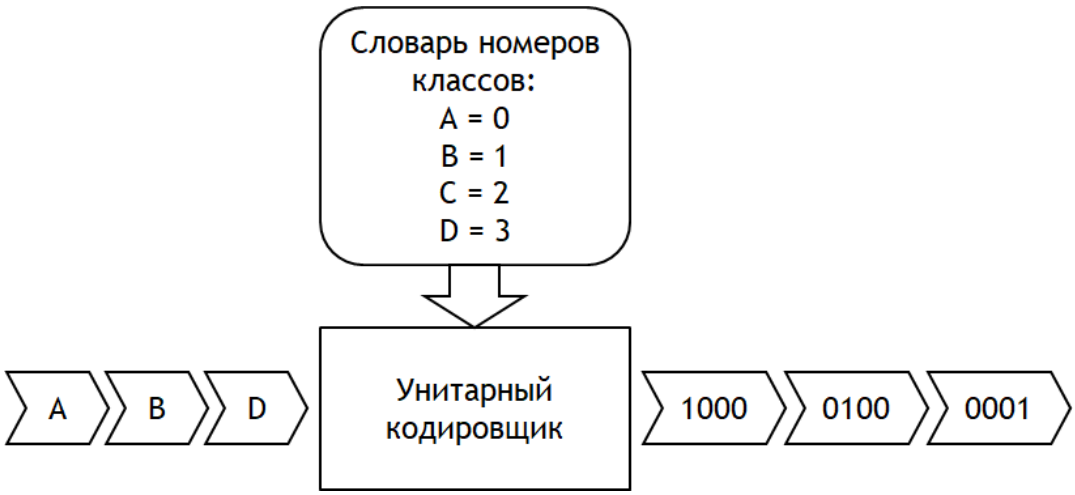

Рисунок 4.1 – Схема унитарного кодирования

Первой группой выделяемых признаков является унитарный код типа аминокислотного остатка. Так как в алгоритме учитываются только наиболее распространённые стандартные аминокислотные остатки, то данная группа состоит из 20 признаков.

Следующей группой признаков является количество и композиция (попризнаковая сумма унитарных кодов) типов аминокислотных остатков, находящихся в непосредственной пространственной близости от рассматриваемого. При этом признаки для остатков, расположенных в каждом из двух направлений в белковой цепи относительно данного остатка, подсчитываются раздельно. Кроме того, используются два различных пороговых значения для признака пространственной близости (8Å и 14Å), каждый из которых образует свою подгруппу признаков. Таким образом, получается 2 * 2 * (20+1) признаков пространственной близости для каждого аминокислотного остатка.

Далее происходит анализ трёхмерной геометрической структуры всей белковой цепи, определение поверхности белка и выделение ряда групп признаков, довольно хорошо зарекомендовавших себя при использовании в статистических методах машинного бучения.

Первая часть признаков трёхмерной структуры содержит информацию о доступной поверхности – части атомной поверхности белковой молекулы, которая является доступной для атомов растворителей. Первой группой признаков такого рода является доступная растворителю площадь поверхности для цепи в целом (Chain Accessible Surface Area, CASA). Данная группа состоит из пяти численных показателей – общей площади, площадей для атомов главной и побочных цепей, а также полярной (для атомов кислорода, азота и фосфора) и неполярной площади (для атомов углерода). Следующей группой является доступная растворителю площадь поверхности для аминокислотного

остатка в отдельности (Residue Accessible Surface Area, RASA), которая также состоит из пяти показателей для различных групп атомов. Третьей группой признаков, характеризующих трёхмерную поверхность белка, является относительная доступная растворителю площадь для остатка (Relative Residue Accessible Surface Area, RRASA). Отличие данной группы от предыдущей заключается в том, что вместо точного значения площади, выражаемого в квадратных ангстремах, используется отношение данного значения к стандартной для рассматриваемого аминокислотного остатка доступной площади.

Вторая часть структурных признаков состоит из индексов глубины и выступа, подсчитываемых для каждого аминокислотного остатка. Индекс глубины (Residue Depth Index, RDPX) для атома определяется как расстояние в ангстремах от него до ближайшего доступного растворителю атома. Если атом сам принадлежит к доступной растворителю поверхности, то индекс глубины для него считается равным нулю. Значения индекса глубины для атомов аминокислотного остатка агрегируют различными способами, образуя, таким образом, индекс глубины остатка, состоящий из 6 признаков: минимальное, максимальное и среднее значение, а также стандартное отклонение среди всех атомов остатка; среднее значение и стандартное отклонение среди атомов побочной цепи. Индекс выступа остатка (Residue Protrusion Index, RCX) атома равен отношению занятого атомами белка объёма к свободному объёму внутри сферы фиксированного радиуса (10 Å) с центром в рассматриваемом атоме. Полученные значения индекса для атомов впоследствии агрегируются по аналогии с индексом глубины.

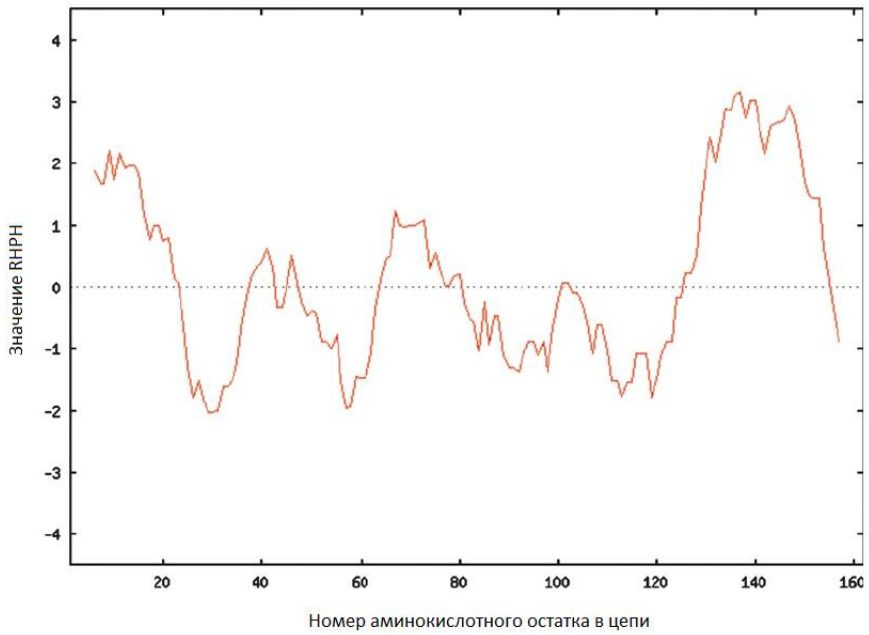

Рисунок 4.2 – Пример RHPH-профиля для некоторой белковой цепи

Для извлечения данных признаков из информации о координатах атомов белка используется приложение Protein Structure Analyzer (PSA), являющееся составной частью системы Protein Structure and Interaction Analyzer PSAIA [19]. При этом, помимо описанных выше признаков, данная программа также определяет значение гидропатии Кайта-Дулитла (RHPH) [12]. Сначала происходит построение профиля гидропатии для всей белковой цепи, а затем сглаживание значений усредняющим фильтром (рисунок 4.2).

Помимо строения поверхности белковой молекулы, полезную информацию можно извлечь также из вторичной структуры белковой цепи. Для её получения применяется алгоритм STRIDE (Structural Identification) [10], который является расширением популярного и по сей день алгоритма DSSP [4]. С использованием данного алгоритма, помимо типа вторичной структуры, также можно получить углы связей внутри аминокислотного остатка φ и ψ (рисунок 4.3). При этом распознаётся расширенный набор из семи типов вторичных структур: альфа-спираль (H), 3_{10}-спираль (G), π-спираль (I), расширенная конформация (E), изолированный мост (B), поворот (T), другое (C). Тип вторичной структуры, как и прочие использованные ранее категориальные признаки, кодируется с помощью унитарного кода.

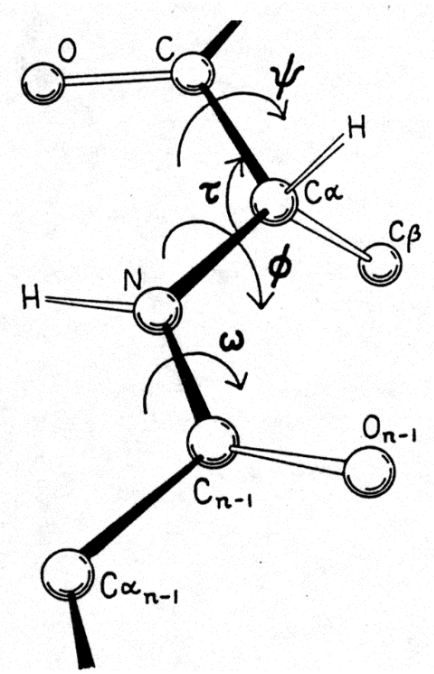

Рисунок 4.3 – Углы межатомной связи в полипептидной цепи

Последней группой вершинных признаков является значение профиля BLOSUM (Blocks Substitution Matrix) [11], известного также под названием «BLAST-профиль». Значения из данной матрицы содержат информацию об относительной вероятности замены аминокислотных остатков другими

остатками и используются практически во всех алгоритмах выравнивания ДНК, РНК и белков для определения схожести регионов. Среди всех разновидностей BLOSUM-матриц для использования в данном алгоритме выбрана симметричная матрица BLOSUM62 (рисунок 4.4). Несмотря на то, что, как выяснилось, значения в данной матрице несколько расходятся с описанным в статье алгоритмом, эти немного неверные значения на практике дают несколько лучшие результаты по сравнению с результатами при использовании точно посчитанных значений [6].

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Рисунок 4.4 – Значения матрицы BLOSUM62

4.4 Выделение признаков для рёбер

Для каждого из рёбер в данном алгоритме определяется два признака – длина и угол связи. Расстояние между аминокислотными остатками при этом рассчитывается как минимальное из расстояний между атомами, входящими в их состав. Угол связи определяется как угол между векторами нормали амидных плоскостей (рисунок 4.5) связываемых аминокислотных остатков.

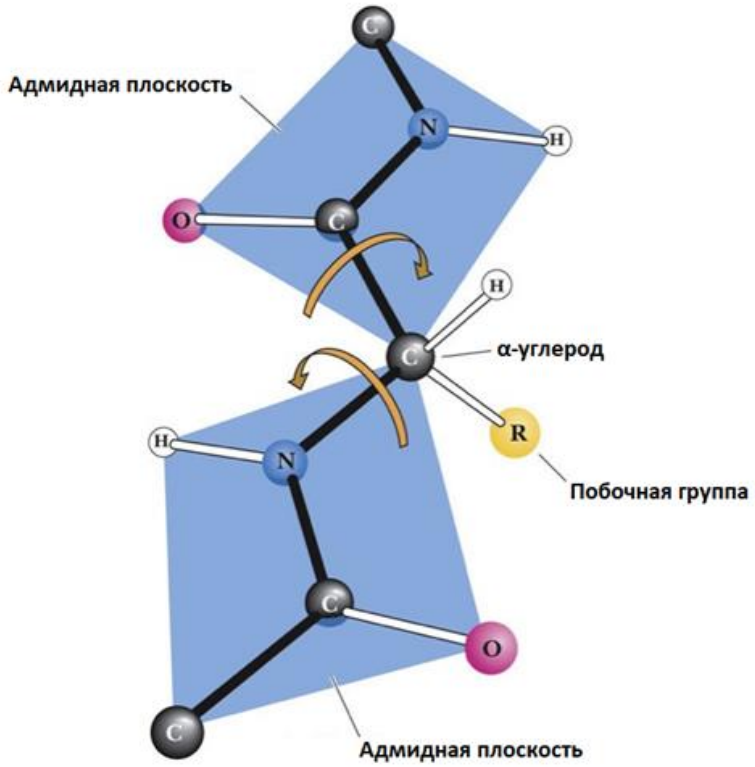

Рисунок 4.5 – Схема принадлежности атомов основной белковой цепи к амидным плоскостям

4.5 Архитектура нейронной сети

Основываясь на изученном материале о принципах работы нейронных сетей, описанных в предыдущей главе, была разработана следующая архитектура нейронной сети:

- изначально графовые представления с начальными значениями признаков проходят раздельную обработку несколькими блоками графовой свёртки;
- далее для каждой пары аминокислотных остатков (один из остатков в паре принадлежит белку-рецептору, а другой – белку-лиганду) ставится в соответствие вектор признаков, представляющий собой соединённые векторы признаков, полученных после последних блоков графовой свёртки;
- последней частью нейронной сети является полносвязная нейронная подсеть, которая для каждого полученного в результате соединения вектора признаков осуществляет итоговое предсказание о наличии или отсутствии взаимодействия между соответствующими аминокислотными остатками белковых цепей.

Более наглядно архитектуру нейронной сети можно понять из схемы, представленной на рисунке 4.6.

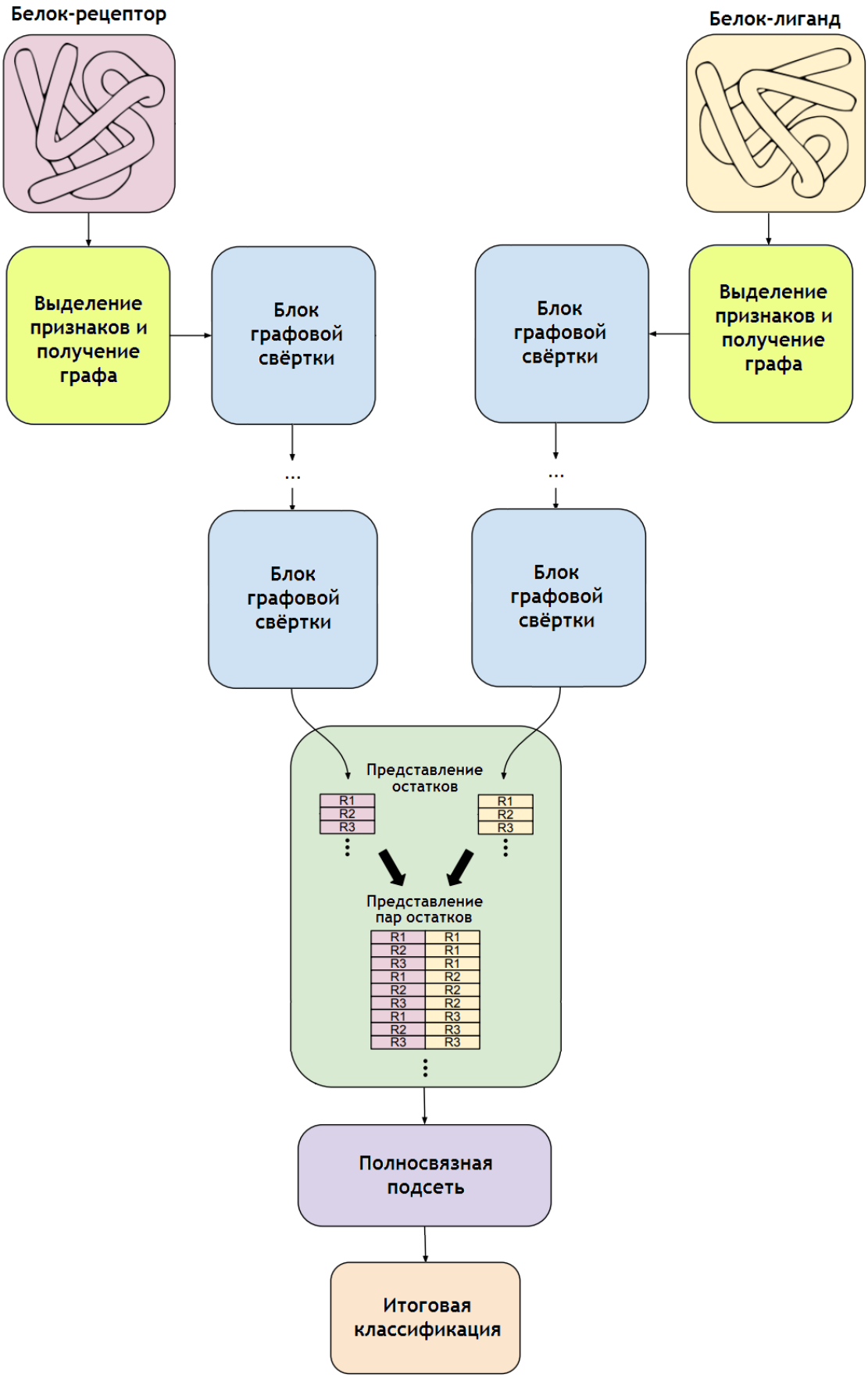

Рисунок 4.6 – Схема работы алгоритма

4.6 Обработка результата нейронной сети

Последним шагом алгоритма является переход от маркеров взаимодействия для пар к непосредственно формату выходных данных алгоритма, содержащих информацию для каждого из аминокислотных остатков.

Для осуществления данного перехода используется следующее правило: аминокислотный остаток является взаимодействующим тогда и только тогда, когда существует хотя бы одна взаимодействующая пара, в состав которой он входит. При этом, помимо маркера наличия взаимодействия, для остатка выписывается также список из аминокислотных остатков другой белковой цепи, с которыми такое взаимодействие будет происходить.

ГЛАВА 5 ПРОВЕДЕНИЕ ЭСПЕРИМЕНТОВ И АНАЛИЗ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ АЛГОРИТМА НА ПРАКТИКЕ

5.1 Использованный набор данных

Для обучения и тестирования алгоритма применялся открытый набор данных Protein-Protein Docking Benchmark [25].

Данный набор для каждого из входящих в его состав белкового комплекса содержит раздельно 4 файла: информацию о лиганде и рецепторе до и после связывания. В соответствующие несвязанному состоянию файлы, при этом, добавляется смещение, поворот и небольшой искусственный шум для обеспечения гарантии отсутствия информации о взаимодействии белковых цепей непосредственно во входных данных.

В качестве обучающей выборки использовались белковые комплексы, входящие в состав версии 4 данного набора данных, а в качестве тестовой – комплексы, входящие в состав версии 5, но отсутствующие в предыдущей версии (белковые комплексы, информация о строении которых не была опубликована на момент создания версии 4). Размеры обучающей и тестовой выборок составили 175 и 55 белковых комплексов соответственно, а длины белковых цепей в них варьировались в пределах от 29 до 1979 аминокислотных остатков (рисунки 5.1 и 5.2).

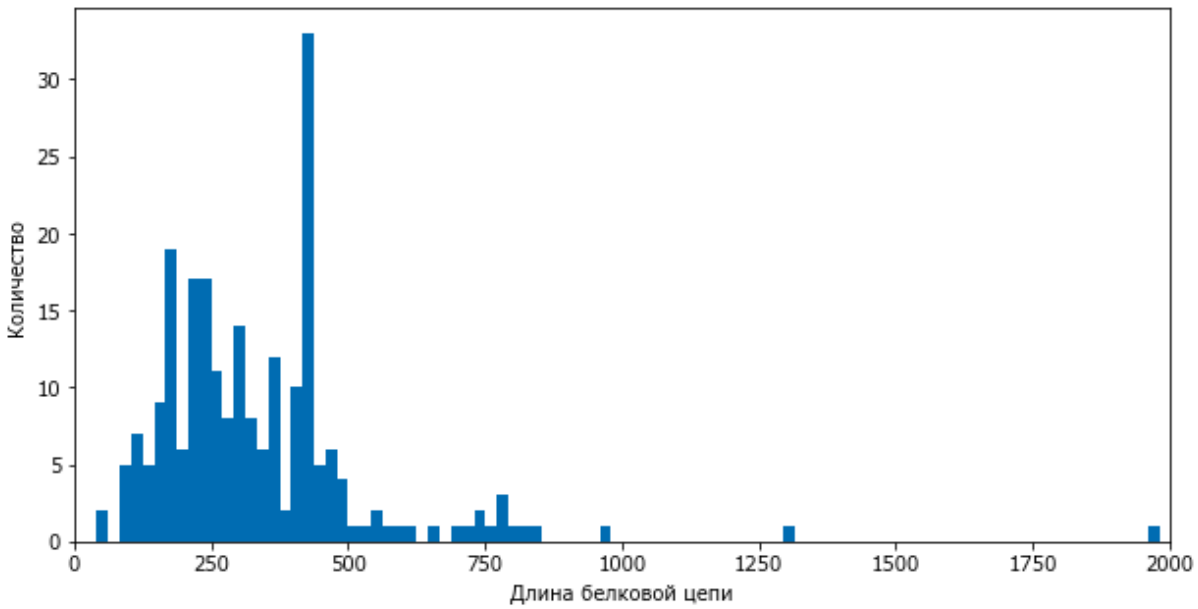

Рисунок 5.1 – Распределение длин белков-рецепторов

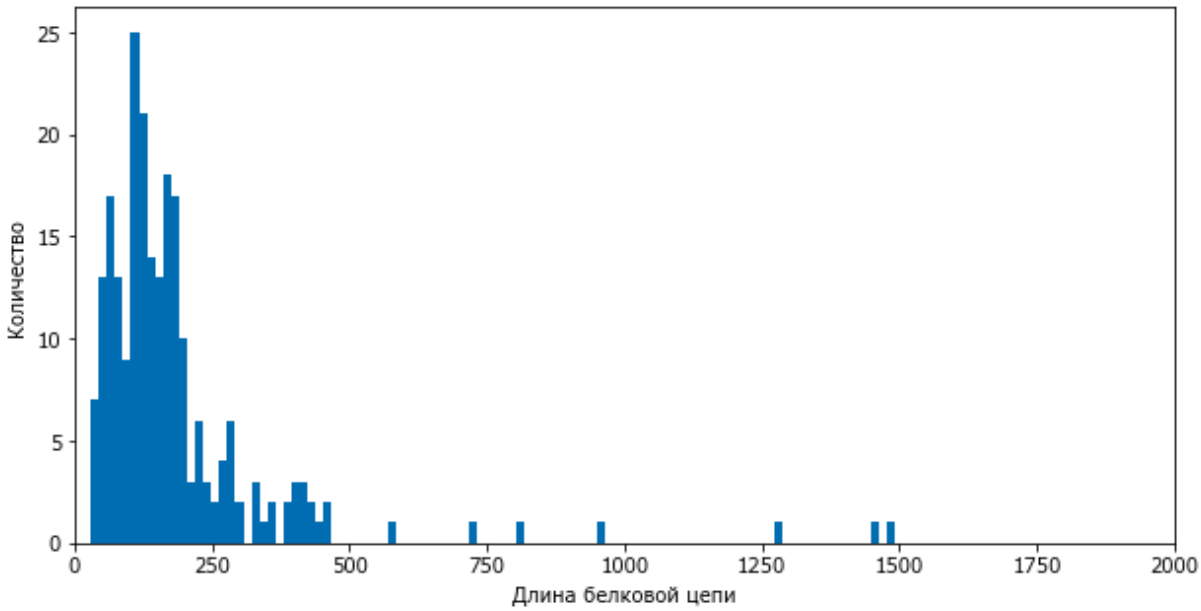

Рисунок 5.2 –Распределение длин белков-лигандов

5.2 Обучение нейронной сети

Для обучения нейронной сети использовался метод обратного распространения ошибки. В качестве функции потерь была использована часто применяемая при создании классификационных нейронных сетей функция кросс-энтропии:

$$H(y, \tilde{y}) = -w_0 y_0 \log \widetilde{y_0} - w_1 y_1 \log \widetilde{y_1}, \qquad (6)$$

где y_0и y_1 – целевые вектора классов, $\widetilde{y_0}$и $\widetilde{y_1}$ – предсказанные нейронной сетью вероятности принадлежности к ним, а w_0 и w_1 – весовые коэффициенты.

В качестве оптимизационного метода был выбран метод градиентного спуска. Обучающий коэффициент изначально устанавливался равным 0.01, однако для получения более точных результатов за 10 эпох до окончания процесса обучения происходило его уменьшение в 10 раз. Обучение проводилось в течение 90 эпох.

Так как количество пар невзаимодействующих аминокислотных остатков значительно превышает количество взаимодействующих, то в процессе обучения для каждого белкового комплекса использовались все его взаимодействующих пары и часть пар, не вступающих в контакт. Последние отбирались случайным образом в количестве, в 10 раз превышающем количество контактирующих пар остатков. Полученный дисбаланс в

количестве объектов разных классов был учтён путём задания соответствующих значений весовых коэффициентов функции потерь.

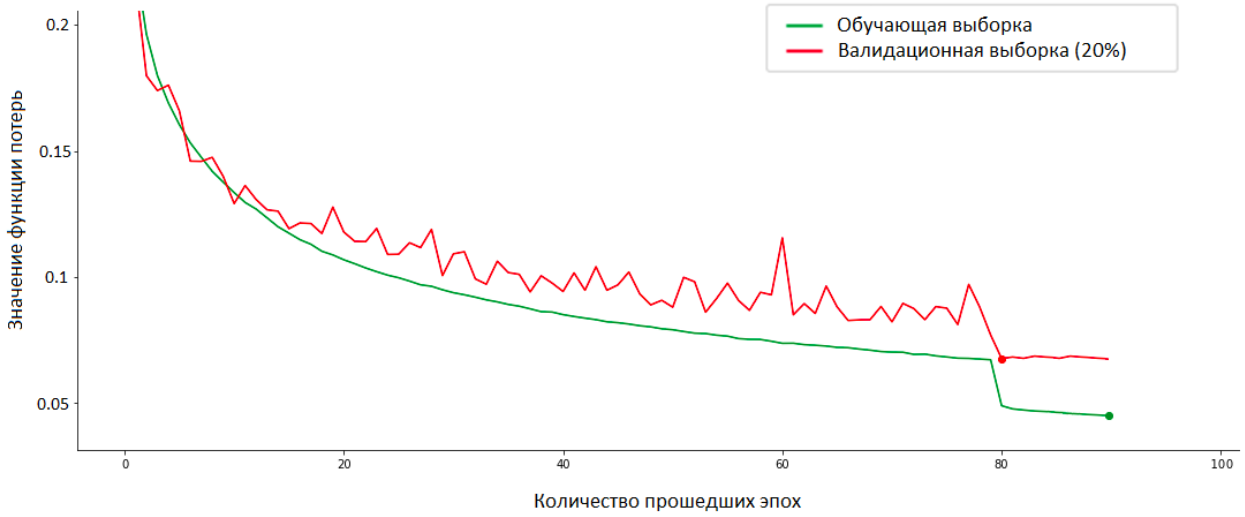

Рисунок 5.3 – График изменения значения функции потерь в ходе одного из экспериментов

5.3 Экспериментальная среда и использованные средства

Реализация алгоритма и вспомогательных средств выполнена на языке Python версии 3.5 в среде Anaconda с использованием инструментов Jupyter Notebook и PyCharm.

Для реализации нейронной сети использовалась открытая библиотека TensorFlow версии 1.4 [21], разрабатываемая компанией Google. Помимо реализации основных методов для организации работы нейронной сети, данная библиотека также позволяет производить расчёты на графических процессорах по технологии CUDA, что сильно уменьшает время, необходимое для обучения нейронной сети.

Чтение файлов в формате PDB, их разбор и удобное структурированное представления в памяти информации о белках выполнялось при помощи свободно распространяемой библиотеки BioPython [5]. Кроме того, в процессе работы использовался ряд иных стандартных библиотек (NumPy, MatPlotLib и другие).

Все экспериментальные запуски проходили на компьютере под управлением ОС Windows 10 со следующими характеристиками: Intel Xeon E3-1505M v6 (3–3.8 ГГц, 8 потоков), 64 ГБ DDR4, NVidia Quadro M2200 (4 ГБ).

5.4 Анализ результатов работы алгоритма

В ходе экспериментов были опробованы различные по количеству и размеру слоёв разновидности описанной ранее архитектуры нейронной сети. Результаты проверки их точности представлены в таблице 5.1. Наилучший результат был получен при использовании вершинно-рёберного типа графовой свёртки в сети, состоящей из трёх свёрточных блоков.

Тип свёрточных блоков	Количество свёрточных слоёв	Время обучения	ROC AUC
Вершинная	1	40 минут	0,75
	2	1,5 часа	0,78
	3	3 часа	0,80
	4	5 часов	0,80
Вершинно-рёберная	1	1 час	0,77
	2	2,5 часа	0,79
	3	5 часов	0,81
	4	8 часов	0,80

Таблица 5.1 – Результаты проведённых экспериментов

Кроме того, были проверены различные параметры и расположение эвристических слоёв, размеры и количество слоёв в полносвязной подсети. В итоге, наилучшие полученные результаты, представленные выше, были достигнуты при конфигурациях, описанных в таблицах 5.2 и 5.3.

Номер слоя	Тип слоя	Параметры	Dropout	Нормализация
1	Свёрточный (вершинный)	256 фильтров	Нет	Есть
2		256 фильтров	Есть, P=0.3	Есть
3		512 фильтров	Есть, P=0.3	Есть
4		512 фильтров	Есть, P=0.2	Есть
5	Полносвязный	512 нейронов	Нет	Нет
6		2 нейрона	Нет	Нет

Таблица 5.2 – Конфигурация нейронной сети, содержащей четыре блока вершинной свёртки

Номер слоя	Тип слоя	Параметры	Dropout	Нормализация
1	Свёрточный (вершинно-рёберный)	256 фильтров	Нет	Есть
2		256 фильтров	Есть, P=0.3	Есть
3		512 фильтров	Есть, P=0.2	Нет
4	Полносвязный	512 нейронов	Есть, P=0.1	Нет
5		256 нейронов	Нет	Нет
6		2 нейрона	Нет	Нет

Таблица 5.3 – Конфигурация нейронной сети, содержащей три блока вершинно-рёберной свёртки

При сравнении полученных результатов точности с существующими методами определения белок-белковых интерфейсов, данный алгоритм хоть и не является хуже, но и даёт не настолько существенный прирост, как хотелось бы. Тем не менее, полученные в ходе данного исследования результаты показывают наличие возможности применения нейронных сетей для решения поставленной задачи, а также открывают простор для дальнейших экспериментов в данном направлении.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы были изучены основные понятия, классификация и типы структуры белков, а также определены различные значимые биохимические характеристики. Кроме того, были проанализированы алгоритмы работы нейронных сетей различных типов (включая наиболее актуальные на сегодняшний день разработки) и выработаны практические навыки их применения для решения различных задач.

На основе полученных знаний впоследствии был разработан и реализован собственный алгоритм поиска белок-белковых интерфейсов. На основании результатов проведённых экспериментов была выбрана его конфигурация, показывающая наиболее хорошие результаты. При сравнении с существующими аналогами, основанными на статистических методах машинного обучения, данный алгоритм дал небольшой прирост в точности, открыв, тем самым, возможности для дальнейшего улучшения показателей путём исследований в рассмотренном направлении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Машинное обучение: курс лекций / К. В. Воронцов // Школа анализа данных [Электронный ресурс] – 2016-2017. – Режим доступа: https://wiki.school.yandex.ru/shad/MachineLearning – Дата доступа: 06.02.2018
2. Структурная Биоинформатика в ШАД: курс лекций / А. В. Головин // Школа анализа данных [Электронный ресурс] – 2012-2014. – Режим доступа: http://vsb.fbb.msu.ru/projects/edu/wiki/Shad – Дата доступа: 08.02.2018.
3. Финкельштейн, А. В. Введение в физику белка / А. В. Финкельштейн // Учебный центр Института белка РАН [Электронный ресурс] – 1999-2000. – Режим доступа: http://phys.protres.ru/lectures/protein_physics – Дата доступа: 11.02.2018
4. A series of PDB related databases for everyday needs. / W. G. Touw [et al.] // Nucleic Acids Research. – 2015. – Vol. 43, Database issue. – P. D364- D368.
5. Biopython: freely available Python tools for computational molecular biology and bioinformatics / P. Cock [et al.] // Bioinformatics – 2009. – Vol. 25, No 11. – P. 1422–1423.
6. BLOSUM62 miscalculations improve search performance / M. Styczynski [et al.] // Nature Biotechnology. – 2008. – Vol. 26, No 3. – P. 274-275.
7. Different protein-protein interface patterns predicted by different machine learning methods / W. Wang [et al.] // Scientific Reports. – 2017. – Vol. 7, Article 16023
8. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs / S. F. Altschul [et al.] // Nucleic Acids Research. – 1997. – Vol. 25, No 17. – P. 3389- 3402.
9. Hamelryck, T. An amino acid has two sides: A new 2D measure provides a different view of solvent exposure / T. Hamelryck // Proteins: Structure, Function, and Bioinformatics. – 2005. – Vol. 59, No 1. – P. 38-48.
10. Heinig, M. STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins / M. Heinig, D. Frishman // Nucleic Acids Research. – 2004. – Vol. 32 – P. W500- W502.
11. Henikoff, S. Amino acid substitution matrices from protein blocks / S. Henikoff, J. G. Henikoff // Proceedings of the National Academy of Sciences of the United States of America. – 1992. – Vol. 89, No. 22. – P. 10915-10919.
12. Kyte, J. A simple method for displaying the hydropathic character of a protein. / J. Kyte, R. F. Doolittle // Journal of Molecular Biology. – 1982. – Vol. 157, No 1. – P. 105-132.

13. Minhas F. PAIRpred: Partner-specific prediction of interacting residues from sequence and structure / F. Minhas, B. J. Geiss, A. Ben-Hur // Proteins: Structure, Function, and Bioinformatics. – 2014. – Vol. 82, No 7. – P. 1142-1155.
14. National Center for Biotechnology Information online resources // National Center for Biotechnology Information [Electronic resource] – 2001-2018. – Mode of access: https://www.ncbi.nlm.nih.gov – Date of access: 19.04.2018
15. Neuvirth, H. ProMate: A Structure Based Prediction Program to Identify the Location of Protein–Protein Binding Sites / H. Neuvirth, R. Raz, G. Schreiber // Journal of Molecular Biology. – 2004. – Vol. 338. – P. 181-199.
16. Predicting protein-protein interface residues using local surface structural similarity / R. A. Jordan [et al.] // BMC Bioinformatics. – 2012. – Vol. 13 – Article 41.
17. PredUs: a web server for predicting protein interfaces using structural neighbors / Q. C. Zhang [et al.] // Nucleic Acids Research. – 2011. – Vol. 39, Web Server issue. – P. W283-287.
18. Protein-Protein Interface Predictions by Data-Driven Methods: A Review / L. C. Xue1 [et al.] // FEBS Letters. – 2015. – Vol. 589, No 23. – P. 3516-3526.
19. PSAIA – Protein Structure and Interaction Analyzer / J. Michel [et al.] // BMC Structural Biology. – 2008. – Vol. 8, Article 21.
20. Sikic M., Prediction of Protein–Protein Interaction Sites in Sequences and 3D Structures by Random Forests. / M. Sikic, S. Tomic, K. Vlahovicek // PLoS Computational Biology – 2009 – Vol. 5, No 1. – Article e1000278
21. TensorFlow: An open source machine learning framework // Google Brain [Electronic resource] – 2015-2018. – Mode of access: https://tensorflow.org – Date of access: 03.04.2018.
22. The Protein Data Bank / H. M. Berman [et al.] // Nucleic Acids Research – 2000. – Vol. 28, No 1. – P. 235–242.
23. The PyMOL Molecular Graphics System // Schrödinger, Inc. [Electronic resource] – 2011-2018. – Mode of access: https://pymol.org – Date of access: 15.03.2018.
24. UniProt: the Universal Protein knowledgebase / The UniProt Consortium // Nucleic Acids Research – 2017. – Vol. 45, Database issue. – P. D158-D169.
25. Updates to the integrated protein-protein interaction benchmarks: Docking benchmark version 5 and affinity benchmark version 2. / T. Vreven [et al.] // Journal of Molecular Biology. – 2015. – Vol. 427, No 19. – P. 3031-3041.
26. Yan, C. A two-stage classifier for identification of protein–protein interface residues / C. Yan, D. Dobbs, V. Honavar // Bioinformatics. – 2004. – Vol. 20, No. suppl_1. – P. i371-i378.