ФАКУЛЬТЕТ
БИОИНЖЕНЕРИИ И
БИОИНФОРМАТИКИ
МГУ ИМЕНИ
М.В. ЛОМОНОСОВА

teach-in
ЛЕКЦИИ УЧЕНЫХ МГУ

МАШИННОЕ ОБУЧЕНИЕ В СТРУКТУРНОЙ БИОЛОГИИ

ГОЛОВИН АНДРЕЙ ВИКТОРОВИЧ
ПЕНЗАР ДМИТРИЙ ДМИТРИЕВИЧ

ФББ МГУ

КОНСПЕКТ ПОДГОТОВЛЕН СТУДЕНТАМИ, НЕ ПРОХОДИЛ ПРОФ. РЕДАКТУРУ И МОЖЕТ СОДЕРЖАТЬ ОШИБКИ. СЛЕДИТЕ ЗА ОБНОВЛЕНИЯМИ НА VK.COM/TEACHINMSU.

ЕСЛИ ВЫ ОБНАРУЖИЛИ ОШИБКИ ИЛИ ОПЕЧАТКИ, ТО СООБЩИТЕ ОБ ЭТОМ, НАПИСАВ СООБЩЕСТВУ VK.COM/TEACHINMSU.

БЛАГОДАРИМ ЗА ПОДГОТОВКУ КОНСПЕКТА
СТУДЕНТКУ ФИЗИЧЕСКОГО ФАКУЛЬТЕТА МГУ
СМОЛЬСКУЮ ДИАНУ ВЛАДИМИРОВНУ

Оглавление

Лекция 1. Введение в структуру белка, молекулярная механика и квантовая химия 5
Введение 5
Уровни организации структуры белка 9
Типы взаимодействий в белках 12
Базисы 17
Силовые поля 20
Ковалентные взаимодействия 21
Нековалентные взаимодействия 24
Варианты ММ 28
Лекция 2. Хемоинформатика 33
Активные молекулы 33
Фарминдустрия 34
HTS 35
Хемоинформатика 37
QSAR 43
ML в хемоинформатике 46
Генеративные подходы 54
Frameworks 56
Лекция 3. Межмолекулярные взаимодействия белок-лиганд 58
Докинг 58
Фрагментарное построение лиганда 64
Методы ML для проблемы докинга 66
Лекция 4. Сравнительное моделирование 78
Введение 78
Сравнительное моделирование 80
Моделирование Ab initio 86
Threading – протягивание нити 87
Распознавание укладки 88
Мета серверы 88
ML методы для предсказания структуры 89
Варианты NN 92

Предсказание структуры белков 94
Заключение 100
Лекция 5. Белок-белковые взаимодействия 101
Макромолекулярный докинг 103
Rosetta 110
ML походы к PPI 112
Лекция 6. Машинные модели для расчёта свойств электронной структуры молекул 123
Введение 123
Наборы данных 124
Представление молекул 128
Типы ML методов 134
Силовые поля на основе ML потенциалов 135
Исследование химического разнообразия 136
Перспективы 138

Лекция 1. Введение в структуру белка, молекулярная механика и квантовая химия

Самым большим достижением в структурной информатике за последние года является AlphaFold 2, который сильно улучшил качество предсказания трёхмерной структуры белков. До этого был AlphaFold, и была проделана большая и долгая работа по его усовершенствованию. Самый важный момент – научиться представлять вещи, которые хотим исследовать и которые должны быть объектом работы алгоритмов машинного обучения в виде тех векторов, той презентации, которая может быть использована для этого.

Основная идеология развития данной области в том, что есть драйверы роста. Они идут за коммерческими компаниями, которые активно разрабатывают саму методологию. Это машинное зрение и многие другие. Учёным часто приходится адаптировать объекты, чтобы они работали в рамках тех фрейм блоков, которые есть в машинном обучении. Поэтому идеология данного курса сводится к тому, как правильно сформулировать задачу для фрейм блоков так, чтобы получались ответы, имеющие смысл.

Статей на эту тему очень много. Очевидно, что люди активно пытаются найти то, что не находилось раньше методами простого анализа, с помощью методов машинного обучения внутри биологических data sets. Делают работу, потом внутреннюю проверку, получается хорошая модель для чего-то. Но опубликованных примеров того, чтобы эти модели могли быть использованы прямо сейчас для создания чего-то нового и были валидированы не только как интерполяция, но и как экстраполяция, почти нет. Отсюда возникает парадокс, состоящий в том, что методы работают хорошо на тех объектах, которые известны, и это можно проверить, но можно ли их применить для создания чего-то нового и более интересного, всё ещё остаётся вопросом. Тем не менее, в этом курсе будут рассматриваться работы, которые привели к созданию чего-то нового, и мы попытаемся понять, как часто это имеет место быть, то есть насколько методы применимы для экстраполяции и действительно полного понимания процессов, которые за этим стоят.

Введение

Структура белка – вещь нетривиальная. Рис. 1.1 скачан из PDB. Когда мы скачиваем из PDB, это просто положение атомов в пространстве. Их можно соединить палочками, будут ковалентные связи, но суть не изменится. В том, что за этим стоит, приходится разбираться специалистам в области структурной биологии. Объектов много, координат тоже, и простой анализ, который мы могли бы себе представить, если бы рассматривали, например, формулу аспирина, здесь не подойдёт.

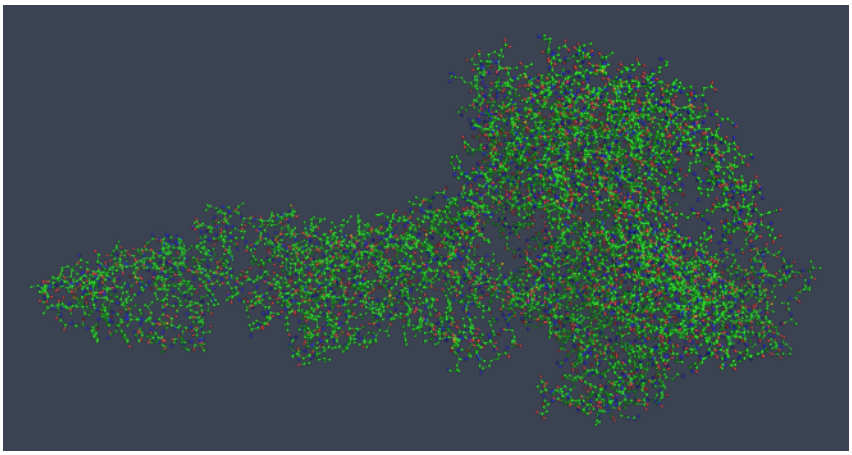

Рис. 1.1. Структура белка из PDB

Белки – это линейные полимеры из аминокислот. Аминокислот 20.

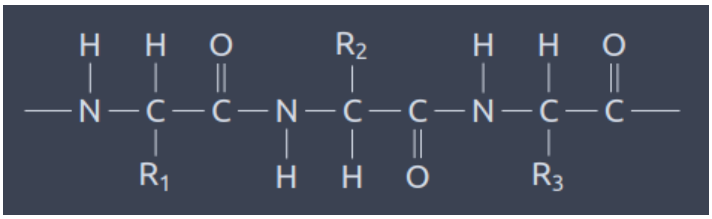

Рис. 1.2. Химическая формула белка

Все **аминокислоты** – L альфа-аминокислоты (рис. 1.3). Это означает, что их раствор может вращать плоскость поляризованного света влево. А есть их стереоизомеры, которые называются D-аминокислоты.

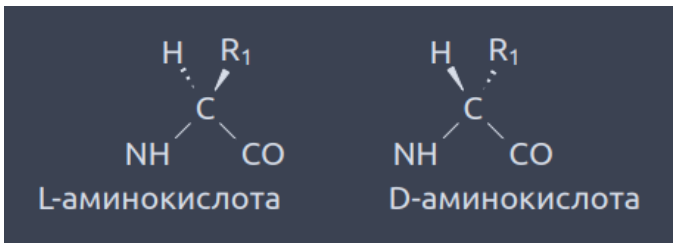

Рис. 1.3. L-аминокислота и D-аминокислота

Они одинаковые по химическому составу и разные по химическому строению. Это вызвано тем, что углерод имеет тетраэдрическое окружение (рис. 1.4).

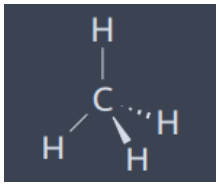

Рис. 1.4. Тетраэдрическое окружение атома углерода

Сами аминокислоты можно разделить на **4 класса** (рис. 1.5). Заряженные, незаряженные (алифатические, ароматические, ещё они могут быть гидрофобными). Есть аминокислоты, которые относятся к полярным, т.е. они незаряжены, но могут

образовывать водородные связи. И ещё есть специальный вид аминокислот типа пролина, глицина и цистеина, потому что они имеют специфическое строение. Допустим, глицин не имеет бокового радикала, поэтому у него очень упрощённое вращение вокруг углов φ и ψ. У пролина наоборот, он циклическая аминокислота, поэтому вращение затруднено. Цистеин же участвует в образовании S-S мостиков.

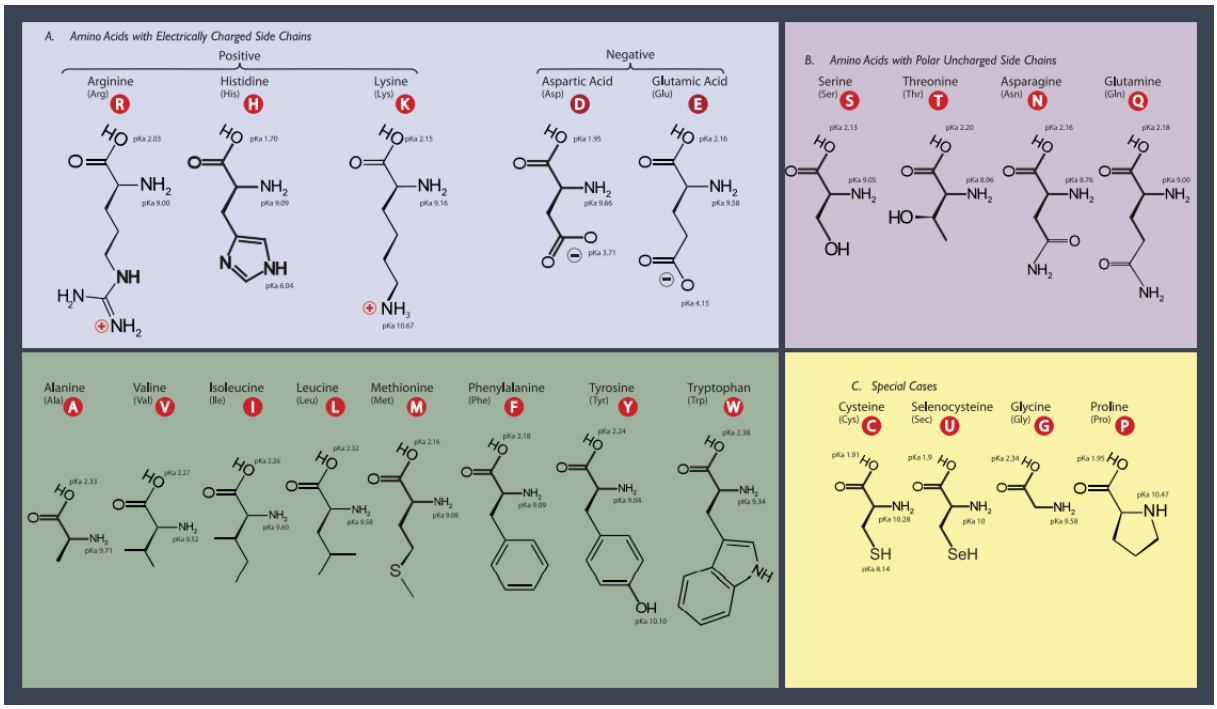

Рис. 1.5. 4 группы аминокислот

Пептидная связь – достаточно специфическая связь, которая образуется между аминокислотами (между карбоксильной и аминогруппой последующих аминокислот) (рис. 1.6).

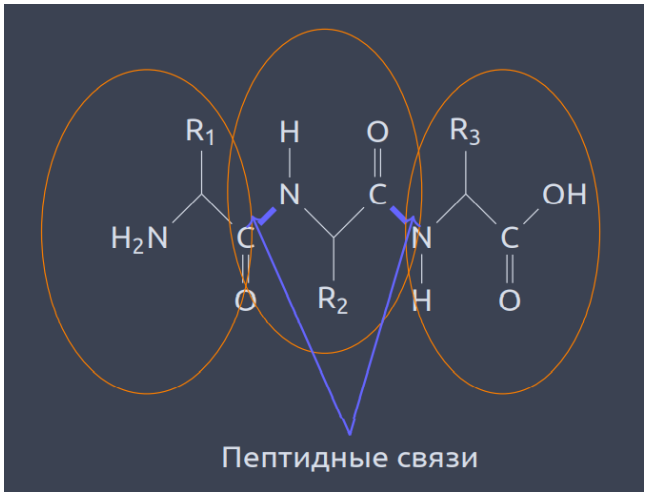

Рис. 1.6. Пептидные связи в белке

Самое важное свойство пептидной связи в том, что там может быть таутомерия (рис. 1.7). Она подразумевает, что возможно два состояния: одно, где есть двойная связь C=N, и второе, где нет. С уверенностью можно сказать, что в белках существует в основном та форма, которая не является аналогом енольной, но всё равно распределённая электронная плотность накладывает ограничения на вращение вокруг этой связи. Пептидная связь плоская, и вокруг неё вращение затруднено.

Валентные углы из-за sp2-гидбридизации 120 градусов. Теоретически возможны цис- и транс- состояния этой связи, но в белках в основном бывает транс.

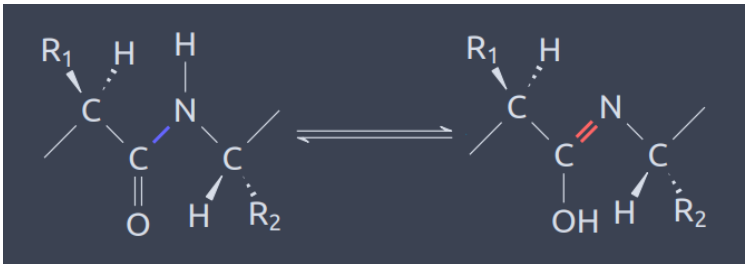

Рис. 1.7. Таутомерия

Рассмотрим **свойства**, которые влияют на структуру. Очевидно, что карбонильный кислород является активным акцептором протона, а азот – активным донором протона при образовании водородной связи. Это важно, т. к. из-за этого образуется вторичная структура белка.

Итак, у нас есть 3 основных **торсионных угла**, вокруг которых происходит вращение (рис. 1.8). Можно предположить, что φ и ψ могут быть практически любыми. А Ω ожидается всегда 180 градусов для того, чтобы были цис- и транс- состояния радикалов относительно пептидной связи.

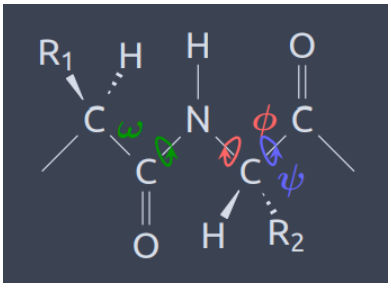

Рис. 1.8. Торсионные углы

Но при массовом анализе данных из рентгеноструктурного анализа оказалось, что углы φ и ψ не могут принимать любые значения. Даже для глицина есть запрещённые зоны, которые берутся из-за того, что радикалы должны отталкиваться друг от друга. Попытка отобразить распределения углов на 2D plot называется картами Рамачандрана (рис. 1.9). Карты Рамачандрана имеют немного разный вид для альфа-спиралей и бета-тяжей.

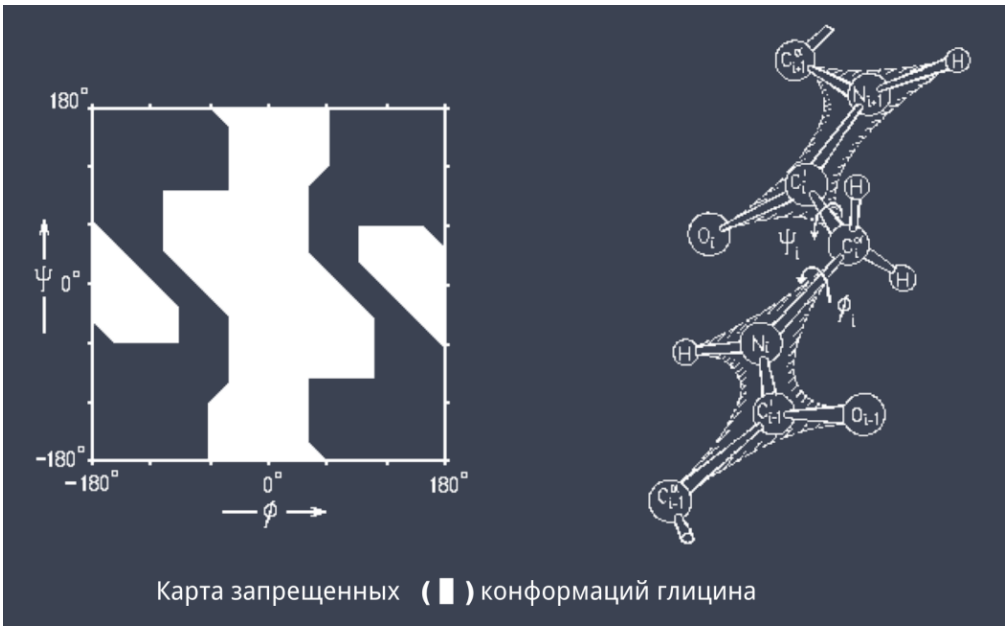

Рис. 1.9. Карты Рамачандрана

Уровни организации структуры белка

Обычно считается, что **уровней организации белка** четыре, с промежуточным утверждением, что такое фолд: первичная, вторичная структура, укладка (fold), третичная, четвертичная структура.

Первичная структура – это аминокислотная последовательность, по сути химическая формула белка (Met-Ala-Gly-Trp-Ala-Val-Asp …). Там перечисляются все ковалентные связи. Но это не отражает его строение в пространстве.

Когда стали анализировать результаты анализа структуры белка, увидели, что появляются некоторые периодические элементы в остове белка. Эти регулярные **вторичные структуры** получили название альфа-спирали и бета-тяжи (рис. 1.10).

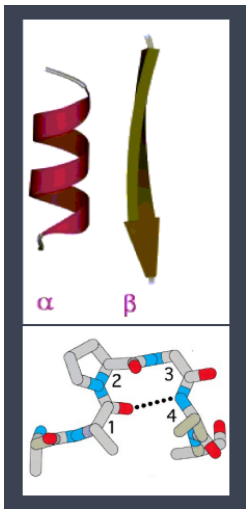

Рис. 1.10. Вторичная структура белка

Для них значения углов φ и ψ попадали в очень узкие диапазоны значений. И это позволяло говорить, что есть система упорядочивания остовов белка, которая приводила к формированию таких структур.

В **альфа-спиралях** образуются водородные связи между азотом и кислородом из остова белка, и эта периодическая структура образует спираль, которая насыщена водородными связями (рис. 1.11). В альфа-спирали все боковые радикалы смотрят в разные стороны от спирали. Это приводит к тому, что половина альфа-спирали может быть гидрофобная, половина гидрофильная, и это определяет свойства. Паттерн спиральности можно увидеть на последовательности и сделать на основе этого предсказания.

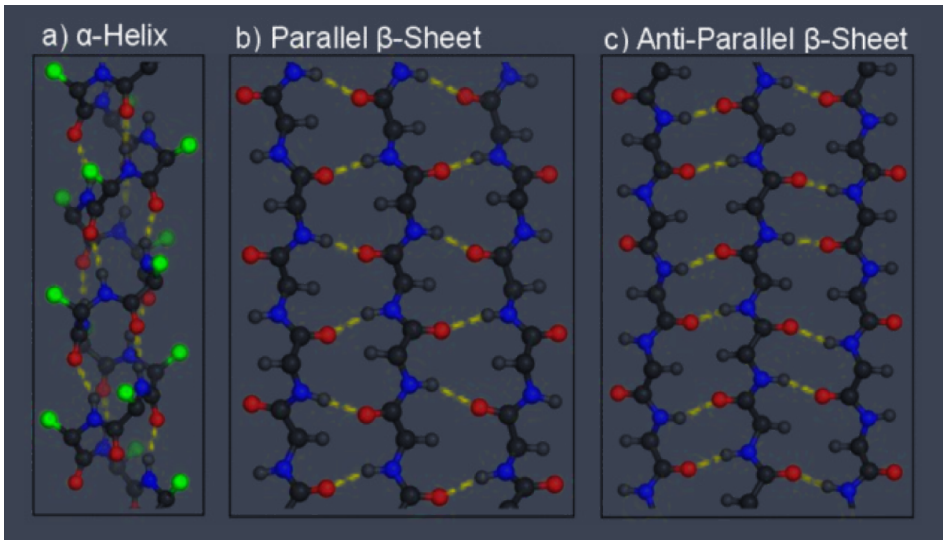

Рис. 1.11. Альфа-спирали и бета-слои

В **бета-тяжах** тоже водородные связи между кислородом и азотом остова (рис. 1.11). Но здесь есть особенность: бета-тяжи могут быть параллельными и антипараллельными. С антипараллельными всё понятно: ход цепи идёт, разворачивается, дальше идёт в обратном направлении. В случае с параллельным существуют некие элементы структуры, которые перещёлкиваются на другую сторону, допустим, альфа-спирали. То есть можно предположить, что в белках, в которых есть параллельные бета-слои, чередование альфа-спиралей и бета-тяжей.

Можно сказать, что какой-то уровень укладки элементов вторичной структуры в пространстве удовлетворяет такому понятию, как **fold**. fold – это расположение элементов вторичной структуры в пространстве друг относительно друга (рис. 1.12). Теоретически его можно описать как вектора в пространстве и угол между ними, а можно на бумаге рисовать их попарные контакты в 2D и пытаться объяснять, что это какой-то вариант fold.

Рис. 1.12. Fold

Можно найти fold, состоящий только из альфа-спиралей или только из бета-тяжей. А в природе распределения примерно равны: альфа-спиральные белки, бета-тяжевые, белки, в которых чередуются альфа-спирали и бета-тяжи, и белки, в которых есть и то, и другое без какой-то очерёдности (рис. 1.13). То есть хорошего правила устройства вторичной структуры белков не существует.

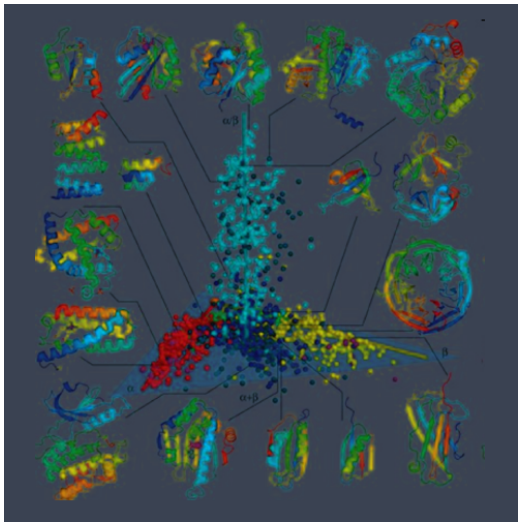

Рис. 1.13. Виды fold

Третичная структура – это расположение в пространстве всех атомов одной полипептидной цепи. Здесь мы знаем координаты и взаимное расположение всех атомов, в том числе и боковых цепей, радикалов. Это именно та структура, которую мы получаем из эксперимента, в отличие от вторичной. Она даёт основную информацию о том, как устроен белок. Т. е. описание третичной структуры включает в себя описание элементов вторичной структуры, типа укладки, структуры петель, конформаций боковых групп всех аминокислотных остатков.

Типы взаимодействий в белках

К тому, что существует третичная структура белков, приводят **водородные связи**. Для водородной связи между атомами, которые дают и принимают водород, должно быть определённое расстояние, а также должен быть определённый угол между связью O-H и прямой, соединяющей два атома. Угол важен, так как водородная связь – некая смесь донорно-акцепторного и кулоновского взаимодействия. Донорно-акцепторное взаимодействие – это когда электронная пара электроотрицательного атома может располагаться на орбитали, в данном случае, водорода. Это некий прообраз корреляционного взаимодействия, которое бывает в металлах. Отсюда вытекает требование по углу.

Водородные связи важны для образования структуры белка, потому что стоимость одной водородной связи – 5 ккал/моль (рис. 1.14). Белки зачастую находятся в воде. Если разорвалась водородная связь внутри белка, она может тут же образоваться с молекулами воды. Водородные связи часто обмениваются с растворителем, и этот обмен не приводит к изменению энергии системы.

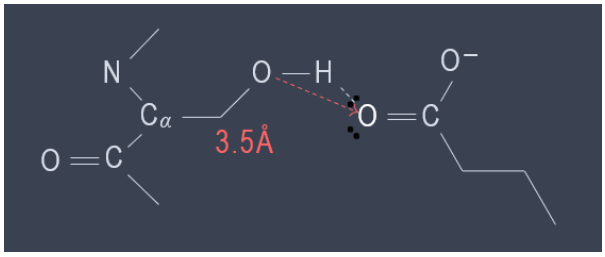

Рис. 1.14. Водородные связи

В белках есть ионные пары, они взаимодействию (рис. 1.15).

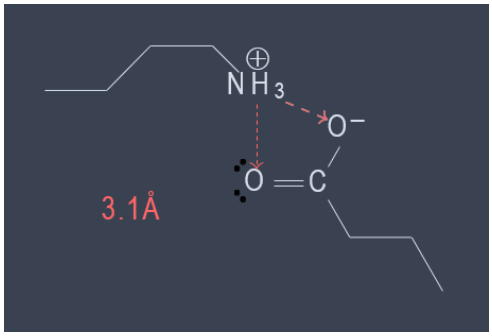

Рис. 1.15. Ионные пары

Очевидно, что если есть заряженные остатки, то плюс взаимодействует с минусом. Положительно заряженные остатки – лизин и аргинин. Оба обладают в явном виде водородами, которые присоединены к атому азота.

Один из распространённых способов стабилизации трёхмерной структуры – образование **дисульфидных мостиков** (рис. 1.16)**.** Важно понимать, что первично формирование структуры белка, после чего в окислительно-восстановительной среде в неком балансе происходит формирование S-S мостика. Этот мостик замыкается,

структура ещё больше стабилизируется. Одна из проблем массового применения S-S мостиков состоит в том, что они могут образовываться между разными молекулами, и у нас получаются нерастворимые осадки и другие нефункциональные формы белков. Поэтому S-S мостики полезны, но их избыточное количество чревато проблемами для формирования структуры белков в нативных условиях.

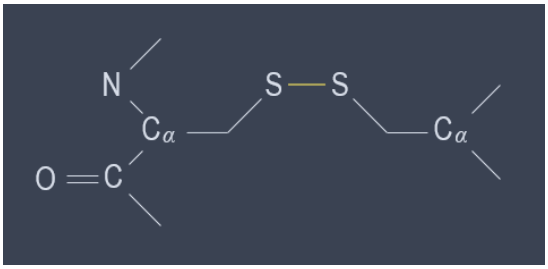

Рис. 1.16. Дисульфидные мостики

Теперь рассмотрим **гидрофобный эффект** (рис. 1.17). Мы не говорим «взаимодействия», т. к. под взаимодействием подразумевается парный эффект – есть два объекта, которые взаимодействуют, есть энергия взаимодействия. Гидрофобный эффект возникает из-за воды. Вода не может образовывать водородные связи с аминокислотами, поверхность этих аминокислот должна быть минимизирована, чтобы вода образовывала максимальное количество водородных связей сама с собой. Те остатки воды, которые остаются на поверхности, на контакте с гидрофобными аминокислотами, теряют энтропию.

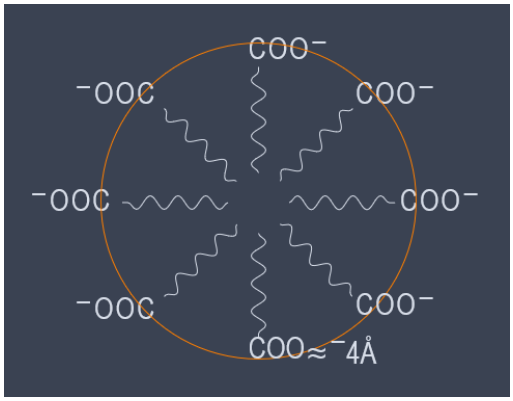

Рис. 1.17. Гидрофобный эффект

Чем меньше молекула воды потеряет энтропию, тем лучше, поэтому гидрофобный эффект будет сжимать любую гидрофобную поверхность до плотно упакованного вещества. Если берём воду, гидрофобный эффект исчезнет, и структура белка перестанет существовать. Гидрофобный эффект является одной из основных причин фолдинга белков. Гидрофобный коллапс – один из первых процессов при образовании структуры белка.

Следующий этап организации белков – образование **высокомолекулярных комплексов**, когда несколько разных белков могут собираться в большие комплексы, которые могут иметь достаточно сложную функцию (рис. 1.18). Например, АТФ-

синтаза, гемоглобин и т. д. Причины взаимодействия между субъединицами в таких комплексах разные. То, что некоторые белки образуют олигомерные или мономерные комплексы, которые выполняют ту или иную функцию в клетке, пока скорее эмпирическое знание.

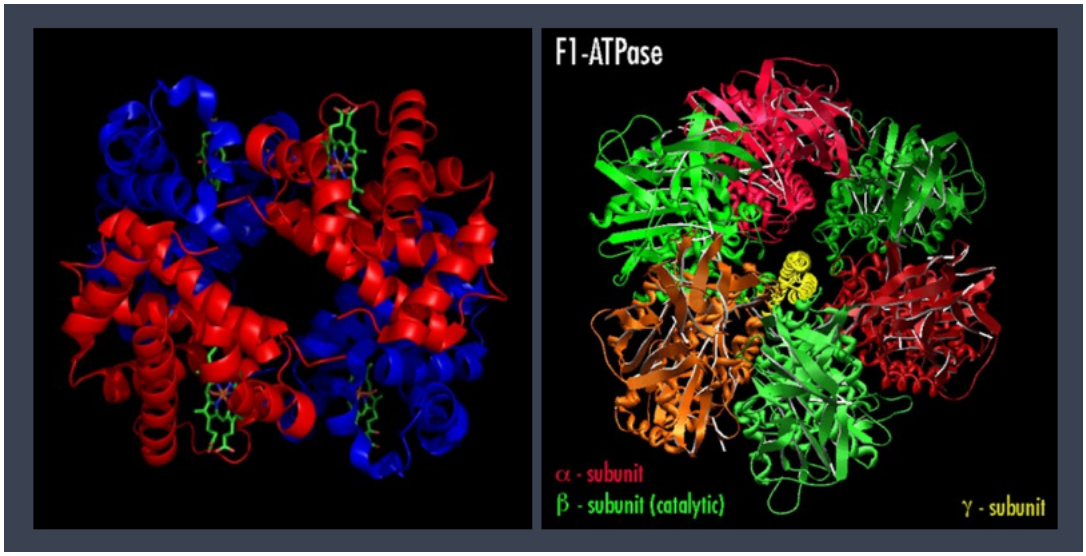

Рис. 1.18. Четвертичная структура белка

Чтобы понять, как это работает, надо понимать, что для структуры хорошо, а что плохо. Это нужно, чтобы проверить результаты экспериментального анализа и для того, чтобы попытаться проводить дальнейшие исследования белков.

Как нам сказать, что хорошо и плохо, если мы знаем положение атомов в пространстве? Основное решение – попытка применить для этого методы всевозможных расчётов, с помощью которых можно узнать, какое взаимодействие даёт какой эффект.

Самое базовое понимание о взаимодействиях нам даёт **квантовая химия**. Этот раздел опирается на то, что мы пытаемся исследовать структуру атома, понимая, что там есть ядро, электроны, электроны на внешних слоях могут образовывать ковалентные связи с соседствующими атомами, и в итоге получается молекула, у которой есть общая электронная плотность, несколько ядер. Из этого можно находить энергии, которые описывают эффективное взаимодействие тех или иных объектов. Объектом здесь являются в том числе ковалентные и нековалентные взаимодействия.

Но есть нюанс, который делает систему сложной. Электрон – это не совсем частица. Электроны обладают явно выраженными волновыми свойствами. А если есть поведение, как у волны, то частицу надо описывать **волновой функцией**. Определение волновой функции – довольно сложная математическая концепция. Волновая функция – комплекснозначная функция, используемая для описания чистого квантового состояния системы. Если говорить более физично, это некоторая функция, модуль квадрата которой отображает плотность электронов в том или ином месте пространства.

Как перейти к расчётам? Есть **уравнение Шрёдингера**. Если мы имеем некий оператор, который называется гамильтониан, к волновой функции, то он будет равен энергии этой системы, умноженной на волновую функцию:

$$H = -\frac{\hbar^2}{2m}\nabla^2 + V \tag{1.1}$$

$$H\Psi = E\Psi \tag{1.2}$$

Дифференциальное уравнение решается. Нас интересует знать одновременно волновую функцию и энергию, опираясь только на координаты ядер в веществе.

$$E = \frac{\int \dot{\Psi} H \Psi \partial r}{\int \dot{\Psi}\Psi \partial r} \tag{1.3}$$

Суть решения уравнения Шредингера для атома, у которого есть только один электрон, т. е. для водорода, сводится к тому, что мы представляем всё это в сферической системе координат.

$$H = -\frac{\hbar^2}{2m}\nabla^2 - \frac{Ze^2}{4\pi\epsilon_0 r}, \tag{1.4}$$

$$H = \frac{1}{2}\nabla^2 - \frac{Z}{r} \tag{1.5}$$

$$\left(-\frac{\hbar^2}{2}\nabla^2 - \frac{Ze^2}{4\pi\epsilon_0 r}\right)\psi(r,\theta,\psi) = E\psi(r,\theta,\psi) \tag{1.6}$$

$$\frac{\hbar^2}{2}\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\psi}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\psi}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2\psi}{\partial\phi^2}\right) - \frac{Ze^2}{4\pi\epsilon_0 r}\psi = E\psi \tag{1.7}$$

Дальше разбиваем гамильтониан, разделяем переменные:

$$\psi(r,\theta,\varphi) = R(r)Y(\theta,\varphi) \tag{1.8}$$

$$\left(\frac{\hbar^2}{2}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\psi}{\partial r}\right) - \frac{Ze^2}{4\pi\epsilon_0}r\right)R(r) = \lambda R(r) \tag{1.9}$$

$$\frac{\hbar^2}{2}\left(\frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\psi}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2\psi}{\partial\phi^2}\right)Y(\theta,\varphi) = -\lambda Y(\theta,\varphi) \tag{1.10}$$

В итоге приходим к тому, что наша волновая функция состоит из трёх основных переменных: радиальной составляющей, сферической гармоники и азимутальной части.

$$R_{n,l}(r) = R_\infty(r)b_0\exp\left(\frac{\mu Ze^2 r}{2\pi\epsilon_0\hbar^2 n}\right) \tag{1.11}$$

$$P_l^m = (1-x^2)^{\frac{m}{2}}\left(a_0\sum_{n=0}^{\infty}\frac{a_{2n}}{a_0}x^{2n} + a_1\sum_{n=0}^{\infty}\frac{a_{2n+1}}{a_1}x^{2n+1}\right), \text{где} \quad (1.12)$$

$$a_{n+2} = \frac{(n+m)(n+m+1)-A}{(n+1)(n+2)}a_n \quad (1.13)$$

$$\Phi_m(\varphi) = c_1 e^{im\varphi} \quad (1.14)$$

Все тригонометрические функции можно выразить с помощью экспонент в виде волн де Бройля.

Результат решения дифференциального уравнения:

$$\psi_{nlm}(r,\vartheta,\varphi) = \sqrt{\left(\frac{2}{na_0}\right)^3\frac{(n-l-1)!}{2n(n+l)!}}e^{-\frac{\rho}{2}}\rho^l L_{n-l-1}^{2l+1}(\rho)Y_l^m(\vartheta,\varphi) \quad (1.15)$$

$L_{n-l-1}^{2l+1}(\rho)$ – обобщённый полином Лагерра степени $n-l-1$, $\rho = \frac{2r}{na_0}$, $Y_l^m(\vartheta,\varphi)$ – сферическая гармоника.

Здесь возникают дискретные величины: n – основное квантовое число (1,2,3…), l – орбитальное число (0, 1, 2,.., $n-1$), m – магнитное число $(-l,..,l)$, которые характерны для каждого элемента.

n	l	m	функция
1	0	0	$\frac{1}{\sqrt{\pi}}\left(\frac{1}{a_0}\right)^{3/2}e^{-r/a_0}$
2	0	0	$\frac{1}{4\sqrt{2\pi}}\left(\frac{1}{a_0}\right)^{3/2}\left(2-\frac{r}{a_0}\right)e^{-r/2a_0}$
2	1	0	$\frac{1}{4\sqrt{2\pi}}\left(\frac{1}{a_0}\right)^{3/2}\frac{r}{a_0}e^{-r/2a_0}\cos\theta$
2	1	-1;1	$\frac{1}{8}\sqrt{\frac{1}{\pi}}\left(\frac{1}{a_0}\right)^{3/2}\frac{r}{a_0}e^{-r/2a_0}\sin\theta e^{\pm i\phi}$

Рис. 1.19. Примеры волновых функций одноэлектронного атома

То есть уравнение Шрёдингера в явном виде даёт представление о строении атомов химических элементов с описанием их электронов в численном виде (рис. 1.19). Когда у нас есть решение для атома водорода, l и m можно подставить любые, чтобы получить уравнение, которое описывало бы положение одного электрона на любом удалении от ядра водорода. То есть, можно в реальности восстановить форму любой орбитали вокруг атома водорода (s p d f…)

Решить уравнение Шрёдингера для системы, в которой более двух частиц, пока невозможно. Нужно применять упрощения. Основное упрощение: мы пытаемся представить систему таким образом, что у нас электрон, когда мы рассчитываем его волновую функцию, находится не только в поле, которое создаётся ядром, но и в усреднённом поле, созданном другими электронами. Это приближение позволяет повторить разделение переменных в сферических координатах.

$$H_i = -\frac{\hbar^2}{2m}\nabla^2 - \frac{Ze^2}{4\pi\epsilon_0 r_i} + \sum_{j\neq i}^{N} \left\langle \left(\frac{e^2}{4\hbar\epsilon_0 r_{ij}} \right) \right\rangle_j \tag{1.16}$$

Тогда для каждого электрона решаем уравнение Шрёдингера в одноэлектронном виде, но, делая итерации по разным электронам, можно построить в итоге волновую функцию, которая, к сожалению, не будет учитывать корреляционный эффект.

Корреляционный эффект – влияние одного электрона на другой при расчёте волновой функции. Это плохо, так как данный эффект явно выражен для ароматических систем. Для учёта этого есть свои решения.

Перейдём к подсчёту волновых функций в молекулах. Применяем **метод ЛКАО** (линейной комбинации атомных орбиталей). Любые орбитали, которые у нас есть у элемента, мы можем рассматривать как некое уравнение. Когда они находятся в молекуле – в элементе, где много орбиталей, – мы можем рассматривать их как линейную комбинацию с разными коэффициентами. Дальше решаем уравнение Шрёдингера для этих волновых функций таким образом, чтобы комбинация этих линейных коэффициентов приводила к тому, чтобы энергия была минимальна. Это итеративное решение приводит к тому, что делаем цикл и стараемся численно подобрать коэффициенты так, чтобы изменение энергии стремилось к нулю при изменении значений коэффициентов.

$$\psi_i = \sum_{\nu=1}^{K} c_{\nu i}\psi_\nu\,, \qquad \frac{\partial E}{\partial c_{\nu i}} = 0 \tag{1.17}$$

Базисы

Дальше поговорим об **упрощениях**, которые есть в этой области. Первое упрощение состоит в том, что работать с показанными функциями не очень удобно, потому что они вычислительно дороги. Можно сделать функции подешевле, если все функции, рассчитанные из уравнения Шрёдингера, подставить в виде комбинации гауссинов. **Гауссиан** – это очень простая функция, экспонента, зависящая от расстояния. Но в явном виде гауссиан не может хорошо описать вид волновых функций. Поэтому делается линейная комбинация из нескольких гауссианов. То есть волновая функция молекулы – это линейная комбинация линейных комбинаций гауссианов.

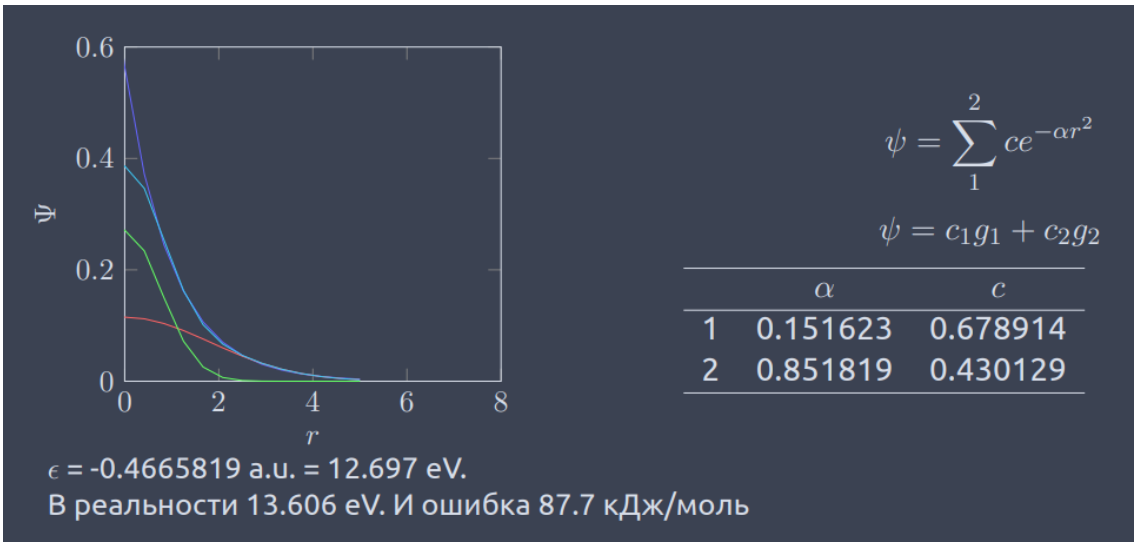

Рис. 1.20. Пример STO-2G для H

Уже из двух простых гауссианов можно так подобрать вид функции, что он будет очень похож на оригинальный. Здесь возможны вариации, особенно там, где нужно учитывать ортогональные функции, появляются дополнительные множители, но суть от этого не меняется.

$$1s = Ne^{\alpha r^2};\ 2p_x = Ne^{\alpha r^2}x;\ 2p_y = Ne^{\alpha r^2}y;\ 2p_z = Ne^{\alpha r^2}z; \quad (1.18)$$

$$3d_{xx} = Ne^{\alpha r^2}x^2;\ 3d_{xy} = Ne^{\alpha r^2}xy;\ 3d_{xz} = Ne^{\alpha r^2}xz; \quad (1.19)$$

$$3d_{yy} = Ne^{\alpha r^2}y^2;\ 3d_{yz} = Ne^{\alpha r^2}yz;\ 3d_{zz} = Ne^{\alpha r^2}z^2; \quad (1.20)$$

$$4f_{xxx} = Ne^{\alpha r^2}x^3;\ 4f_{xxy} = Ne^{\alpha r^2}x^2y;\ 4f_{xxz} = Ne^{\alpha r^2}x^2z \quad (1.21)$$

Существуют **базисные функции** нескольких типов. Самый распространённый тип – который используется в программе GAUSSIAN.

$$\psi = de^{-\alpha r^2} \quad (1.22)$$

Идея в том, что электроны делятся на два типа. Одни находятся на валентной (внешней) орбитали, другие не на валентных орбиталях. Описание валентных должно быть гораздо более точным, чем не валентных. Тогда можно описывать не валентные электроны обыкновенными гауссианами (например, брать их три или шесть) – ограниченными, а валентные – гауссианами, у которых можно менять и d, и α, – неограниченными (например, два гауссиана, у которых не меняются коэффициенты, и один, у которого меняется).

Ещё есть дополнительный гауссиан, который позволяет делать поправки, чтобы всё лучше и лучше описывать внешние орбитали, которые имеют сложную форму.

Существуют дополнительные значки, которые обозначают, что надо добавлять гауссианы, что это всё лучше и лучше описывалось. В литературе описывается много разных способов подсчётов: разные базисы, уровни теории и т. д.

Квантовая химия будет очень нужна. В биологических молекулах, особенно в белках, часто существуют перестановки ковалентных связей. В основном это в ферментах. Ферменты – функционально очень большая доля белков. Если мы хотим применять к ним методы машинного обучения, нужна квантовая химия. Но чтобы в целом понимать, о чём речь, достаточно поверхностного представления. А более глубокое понимание можно получить из литературных источников, где можно прочитать, для каких типов расчётов какой тип комбинаций, уровни теории, количество гауссианов более выгодно.

$$\left(-\frac{\hbar^2}{m}\left(\frac{\partial^2}{\partial x}+\frac{\partial^2}{\partial y}+\frac{\partial^2}{\partial z}\right)+V\right)\psi(r,t)=i\hbar\frac{\partial\psi(r,t)}{\partial t} \tag{1.23}$$

$$H\Psi=E\Psi, \qquad H=-\frac{\hbar^2}{2m}\nabla^2-\frac{Ze^2}{4\pi\epsilon_0 r} \tag{1.24}$$

Посчитать весь белок с помощью квантовой химии мы не можем, т. к. каждая аминокислота – это примерно 900 гауссианов. Аминокислот тоже много.

Для белков и других крупных систем существует приближение **молекулярной механики**. Будем считать, что белок сам по себе не меняет ковалентного состава, функционирование происходит без изменения химической формулы белка. Тогда нам всё равно, как ведут себя электроны на валентных орбиталях, мы это можем попытаться аппроксимировать методами классической физики, которые вместе с набором параметров, называемых силовыми полями, могут использоваться для подсчёта «энергии» - скорее score функции, чем энергии в реальности в эксперименте.

Если мы отбрасываем электроны, то работаем с атомами. Представлением атома являются координаты ядра, которые мы можем двигать. Это упрощение сильно ускоряет расчёты: есть раньше каждое ядро давало необходимость искать 60 коэффициентов, то теперь этого не надо. Если аккуратно параметризовать атомы в молекулярной механике, то их поведение может хорошо коррелировать с экспериментом и даже быть сравнимо с методами квантовой химии в целом.

Так выглядит **простое уравнение силового поля:**

$$U=\sum_{bonds}\frac{k_i}{2}(l_i-l_0)^2+\sum_{angels}\frac{k_i}{2}(\varphi_i-\varphi_0)^2+\sum_{torsions}\frac{V_n}{2}(1+\cos(n\omega-\gamma))+$$

$$+\sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12}-\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right)+\frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}\right) \quad (1.25)$$

Формула делится на две компоненты: первая – ковалентное взаимодействие, вторая – нековалентное. Принципиальная разница между этими компонентами в том, что ковалентные взаимодействия происходят по списку, который известен заранее – химической формуле белка, а нековалентные формально могут быть между любой парой атомов (водородная связь способна образоваться между любыми атомами). Мы должны перебрать все против всех, поэтому возникает двойная сумма.

Переменными являются положения атомов в пространстве (рис. 1.21). Если говорим о связях – это длина связи, угол между связями, торсионный угол, положения атомов друг относительно друга. Остальное – константы, и их надо уметь оптимизировать. Константы оптимизированы под конкретную формулу. Если формула силового поля меняется, то и константы должны меняться.

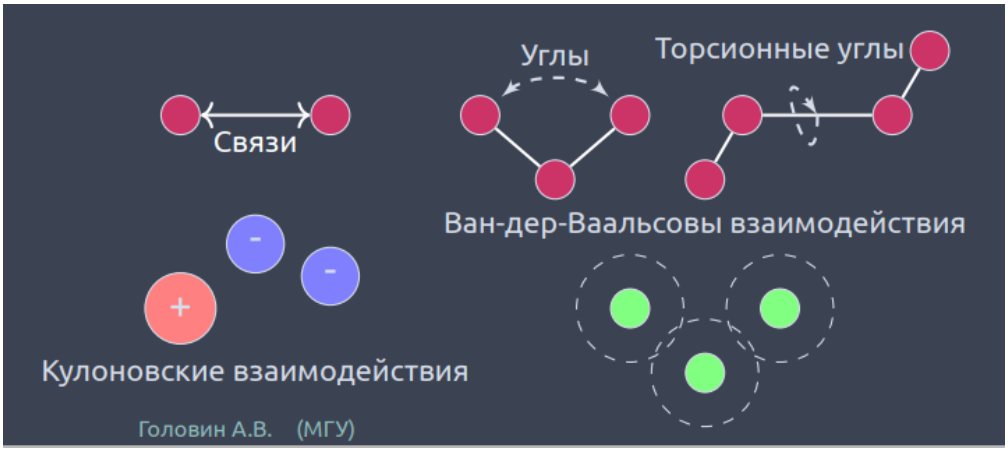

Рис. 1.21. Кулоновские и Ван-дер-Ваальсовы взаимодействия

Силовые поля

Параметризация часто зависит от целей исследования. Параметризация для белков опирается на то, чтобы константы воспроизводили их структуру, динамику в растворителе (в воде) или, с адаптациями, в мембране.

Параметры силовых полей в основном рассчитываются из квантовой химии за редким исключением (рис. 1.21). Особенно Ван-дер-Ваальсовы параметры, которые являются просто результатом постоянной оптимизации, нередко основывающейся на попытке добиться того, чтобы теплоты испарения простых органических веществ воспроизводились в данном силовом поле. Тогда можно будет сказать, что нековалентные взаимодействия оптимизированы хорошо для разных типов атомов.

Посмотрим на альфа-аминокислоту гистидин (рис. 1.22). Здесь есть три атома азота. Они химически разные. Самый нижний находится в пептидной связи. Остальные

два принадлежат ароматическому кольцу, и один из них ещё имеет протон. Один из атомов в кольце – акцептор водородной связи, другой донор, нижний тоже донор водородной связи.

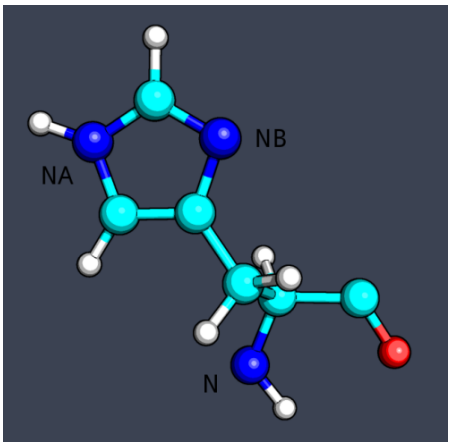

Рис. 1.22. Гистидин

Для каждого типа атомов нужно своё описание. Поэтому, несмотря на то, что в аминокислотах всего 6 элементов, типов атомов примерно 40. Для каждого типа мы можем прописать соответствующие параметры силового поля.

Ковалентные взаимодействия

Со **связью** здесь было бы всё просто, если бы мы понимали, как она рвётся. В принципе, описания разрыва ковалентной связи можно добиться с помощью потенциала Морзе:

$$U(l) = D_e\left(1 - l^{-a(l-l_0)}\right)^2 \tag{1.26}$$

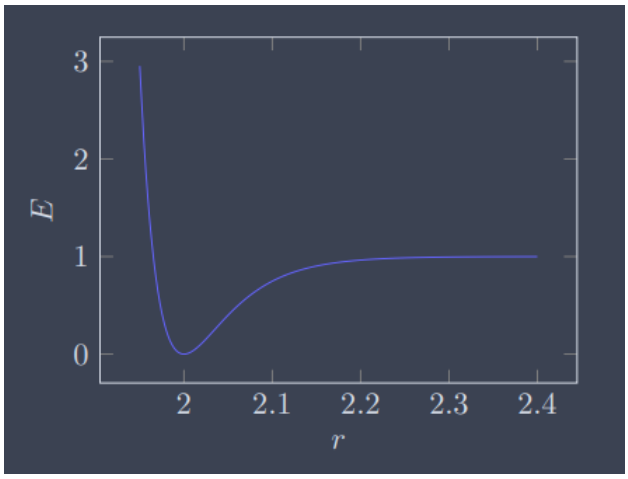

Рис. 1.23. Потенциал для описания связи

Но мы договорились, что в белках нет изменения ковалентной структуры, поэтому самое простое, чем можно описать ковалентную связь, это гармонический потенциал (рис. 1.23), который просто напоминает закон Гука:

$$U(l) = \frac{k_i}{2}(l_i - l_0)^2 \quad (1.27)$$

Параметры для описания ковалентной связи – её оптимальная длина и жёсткость. Чем больше кратность связи (двойная, тройная), тем выше жёсткость, чем чаще колебания.

$$E = \frac{k}{2}(r - r_0)^2 \quad (1.28)$$

связь	r_0,A	k, kcal mol^{-1}A^{-2}
Csp_3-Csp_3	1.523	317
Csp_2-Csp_2	1.337	690
Csp_2-Osp_2	1.208	777
Csp_3-Nsp_3	1.438	367

Рис. 1.24. Параметры для описания ковалентной связи

Можно попытаться **аппроксимировать** ковалентную связь более сложными рядами (рис. 1.25), но зачастую это не используется, так как колебания по ковалентной связи оказывают малое влияние на структуру белков. Здесь основными являются нековалентные взаимодействия.

$$U = \frac{k_i}{2}(l_i - l_0)^2(1 - k'(l_i - l_0) - k''(l_i - l_0)^2 - k'''(l_i - l_0)^3 - k''''(l_i - l_0)^4 \dots) \quad (1.29)$$

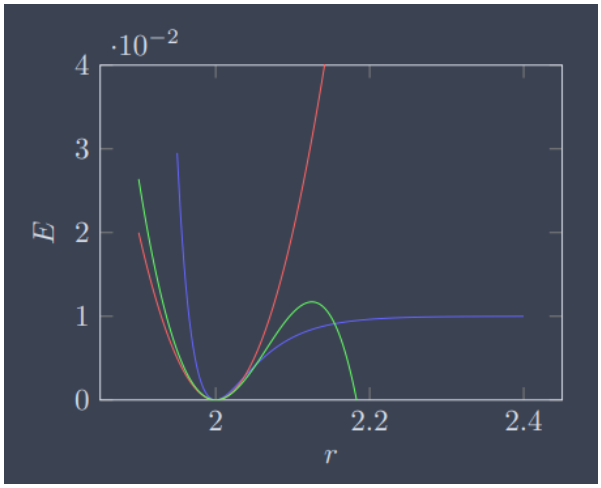

Рис. 1.25. Варианты аппроксимации

Потенциал валентного угла тоже описывается с помощью закона Гука, т. е. это гармонический потенциал, но зато он очень хорошо описывает энергию отклонения угла, в отличие от связи:

$$U(\varphi) = \frac{k_i}{2}(\varphi_i - \varphi_0)^2 \quad (1.30)$$

$$U(\varphi) = \frac{k_i}{2}(\varphi_i - \varphi_0)^2(1 - k'(\varphi_i - \varphi_0) - k''(\varphi_i - \varphi_0)^2 - k'''(\varphi_i - \varphi_0)^3 - k''''(\varphi_i - \varphi_0)^4 \dots) \quad (1.31)$$

Рассмотрим **торсионные углы**. Если есть разные заместители вокруг ковалентной связи, то и профиль энергии торсионного угла может иметь достаточно сложную форму. Для того, чтобы эту проблему решить, используются суммы соответствующих косинусов. Если складывать больше двух косинусов, вид функции может достаточно адекватно описывать потенциал торсионного угла, который можем рассчитать в квантовой химии (рис. 1.26).

$$U(\omega) = \sum_{torsions} \frac{V_n}{2}(1 + \cos(n\omega - \gamma)) \quad (1.32)$$

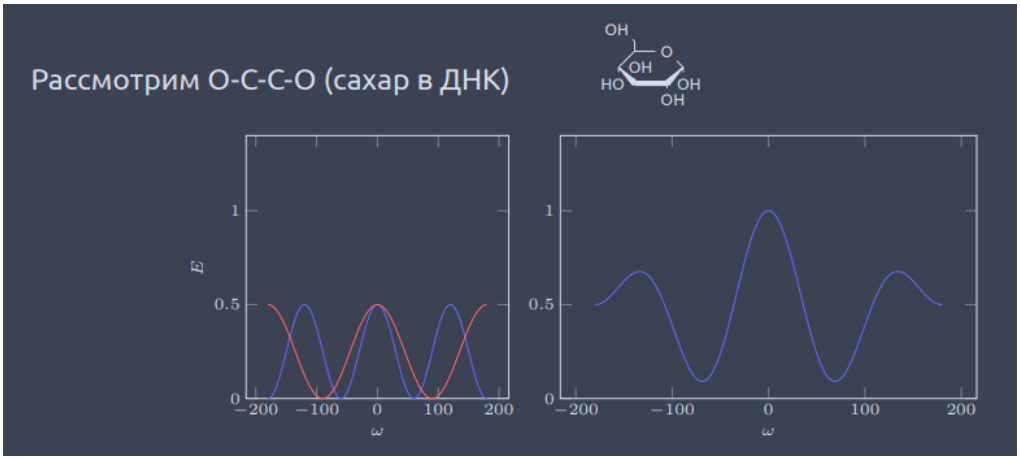

Рис. 1.26. Потенциал торсионного угла

В некоторых случаях может использоваться разложение в ряды Фурье в зависимости от типа силового поля:

$$U(\omega) = \frac{V_1}{2}(1 + \cos\omega) + \frac{V_2}{2}(1 + \cos 2\omega) + \frac{V_3}{2}(1 + \cos 3\omega) \dots \quad (1.33)$$

$$U(\omega) = \frac{1}{2}\left(F_1(1 + \cos\omega) + F_2(1 - \cos 2\omega) + F_3(1 + \cos 3\omega) + F_4(1 - \cos 4\omega)\right) \quad (1.34)$$

Интересной особенностью торсионных углов является необходимость того, что нам нужно делать так, чтобы некие группы атомов находились в одной плоскости. Любое окружение атомов, находящихся в sp2-гибридизации, является плоским.

Для этого используются некоторые приёмы. Один из самых простых – у нас есть 4 точки, мы хотим, чтобы они лежали в одной плоскости. Мы можем задать торсионный угол, который идёт не по связям. Между нижними атомами появляется виртуальная связь, которая определяет взаимное положение всех четырёх атомов. Тогда можно

запретить вращение вокруг этой связи путём описания этого угла как гармонического потенциала, например. Тогда все четыре атома не могут отклоняться из плоскости друг друга. Как бы запрещается описание неправильного торсионного угла (рис. 1.27). Такое работает для описания ароматических и других систем.

$$U(\omega) = V_1(1 - \cos\omega) \tag{1.35}$$

$$U(\omega) = V_1(\omega - \omega_0)^2 \tag{1.36}$$

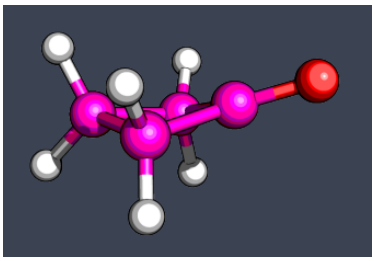

Рис. 1.27. Циклобутан

В силовых полях есть такое понятие, как **кросс-составляющие**, которые говорят о том, что у нас может изменяться жёсткость угла от колебания атома по связи (рис. 1.28). В белках это не используется.

$$U(l_1, l_2) = \frac{K_{l_1 l_2}}{2}\left(l1 - l_{1,0}\right)^2\left(l2 - l_{2,0}\right)^2 \tag{1.37}$$

$$U(l_1, l_2, \varphi) = \frac{K_{l_1 l_2 \varphi}}{2}\left(\left(l1 - l_{1,0}\right)^2 + \left(l2 - l_{2,0}\right)^2\right)(\varphi - \varphi_0) \tag{1.38}$$

Рис. 1.28. Кросс-составляющие в силовых полях

Нековалентные взаимодействия

Нековалентные взаимодействия – важные, определяют структуры белков, энергии их формирования, взаимодействия и т. д. Так как эти взаимодействия происходят через пространство, нам нужно использовать двойные суммы. Они считаются долго, поэтому поговорим о методах, ускоряющих этот процесс.

Начнём с **электростатического взаимодействия**, описывающегося кулоновскими формулами. Атомы в белке имеют обычно некратный заряд (0,2; 0,5; 1,5; – 0,7…). Эти значения зарядов можно примерно определить из квантовой химии путём фиттинга в электронную плотность неких частичных зарядов на атомах так, чтобы они воспроизводили поверхность единичного потенциального заряда.

$$U(q_1, q_2) = \frac{q_1 q_2}{4\pi\epsilon_0\epsilon_r r_{ij}} \tag{1.39}$$

$$U = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \frac{q_i q_j}{4\pi\epsilon_0 \epsilon_r r_{ij}} \tag{1.40}$$

Ещё здесь есть диэлектрическая потенциальность среды. Это параметр, который описывает падение эффективности взаимодействия двух зарядов в той или иной среде. Зачастую мы используем моделирование белков вместе с водой, и этот параметр там не сходится, потому что он в явном виде рассчитывается в ходе самих расчётов.

Есть несколько **подходов** для подсчёта электростатических взаимодействий.

Двойное обрезание – очень простой подход. В некой маленькой сфере мы оцениваем взаимодействие каждого атома с каждым, а за пределами этой сферы – взаимодействия атома с группами атомов (рис. 1.29). Атомы в группе формируют общий заряд, и мы оцениваем взаимодействие только с ним. Это сильно ускоряет расчёты. За пределами второго круга мы уже ничего не считаем. Это плохо, так как могут возникать краевые эффекты.

$$U_1 = \sum_{j=1}^{N_A} \frac{q_1 q_i}{4\pi\epsilon_0 \epsilon_r r_1} + \sum_{j=1}^{N_{group}} \frac{q_1 q_j}{4\pi\epsilon_0 \epsilon_r r_{1j}} \tag{1.41}$$

Рис. 1.29. Двойное обрезание

Чтобы избежать краевых эффектов на второй сфере, существует такое уравнение, как **потенциал реакционного поля**. Суть его сводится к тому, чтобы сгладить вид функции на грани второй сферы и свести это значение примерно к нулю. Но при этом недооцениваются удалённые классические взаимодействия.

$$U_{ij} = \frac{q_1 q_i}{4\pi\epsilon_0 \epsilon_r r_{1i}} \left(1 + \frac{\epsilon_{rf} - \epsilon_r}{2\epsilon_{rf} + \epsilon_r} \frac{r_{ij}^3}{r_c^3}\right) - \frac{q_1 q_i}{4\pi\epsilon_0 \epsilon_r r_c} \frac{3\epsilon_{pf}}{2\epsilon_{rf} + \epsilon_r} \tag{1.42}$$

Часто в моделировании мы работаем с системами, которые имеют собственное отражение. То есть, если мы рассматриваем ячейку, мы пытаемся её представить так, чтобы у неё были соседи справа, слева, сверху, снизу, и чтобы молекулы воды могли виртуально свободно мигрировать между соседями путём простого приближения: если вода достигает одной стенки, она переносится на другую стенку с противоположной стороны и продолжает движение в том же направлении. Это условие называется

периодическим граничным условием. Оно позволяет избежать проблем с граничными условиями. Но было бы полезно, используя его, уметь считать влияние удалённых виртуальных ячеек на нашу ячейку.

Эту проблему решил **Эвальд**, когда исследовал электростатические взаимодействия в кристаллах. Очевидно, что там есть повторяющиеся ячейки, и они друг на друга влияют. Проблема в том, что ячеек много, сумма пятимерная:

$$U_{ij} = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \sum_{z=1}^{N_z} \sum_{i=1}^{N} \sum_{j=1} N \frac{q_i q_i}{4\pi\epsilon_0 \epsilon_r r_{ij}} \tag{1.43}$$

Эвальд предложил разбить это на три компоненты, которые сходились достаточно быстро:

$$U = U_{dir} + U_{rec} + U_0 \tag{1.44}$$

$$U_{dir} = \frac{f}{2} \sum_{i,j}^{N} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \sum_{z=1}^{N_z} q_i q_i \frac{erfc\left(\beta_{r_{ij,n}}\right)}{r_{ij,n}} \tag{1.45}$$

$$U_{rec} = \frac{f}{2}\pi V \sum_{i,j}^{N} q_i q_i \sum_{m_x} \sum_{m_y} \sum_{n_z} \frac{\exp\left(\frac{-\pi m}{\beta}\right)^2 + 2\pi i m\left(r_i r_j\right)^2}{m} \tag{1.46}$$

$$U_0 = \frac{f\beta}{\sqrt{\pi}} \sum_{i}^{N} q_i^2 \tag{1.47}$$

Третий компонент с Фурье-преобразованиями и приводил к тому, что мы можем достаточно эффективно рассчитывать влияние удалённых ячеек.

Для того, чтобы понять, насколько важно такое усложнение для расчёта физических взаимодействий, можно рассмотреть простую систему – бислой, мембрану (рис. 1.30). Мембрану надо моделировать как достаточно большой протяжённый объект. К нему надо применять периодические граничные условия. Большинство фосфолипидов, из которых состоят мембраны, имеют как положительные, так и отрицательные заряды. Получается, что на границе липидного бислоя расположено очень много зарядов. И если мы это моделируем с помощью двойного обрезания, у нас получается шарик по причине того, что в какой-то момент появляется краевой эффект, который не учитывает некоторое взаимодействие. А когда применяем для этой системы метод Эвальда, получается хороший бислой с правильными параметрами плотности и др. Поэтому

электростатику хорошо считать с применением суммирования Эвальда, и эти расчёты справедливы для любых систем, где есть более-менее значимое количество зарядов (нуклеиновые кислоты, бислои…)

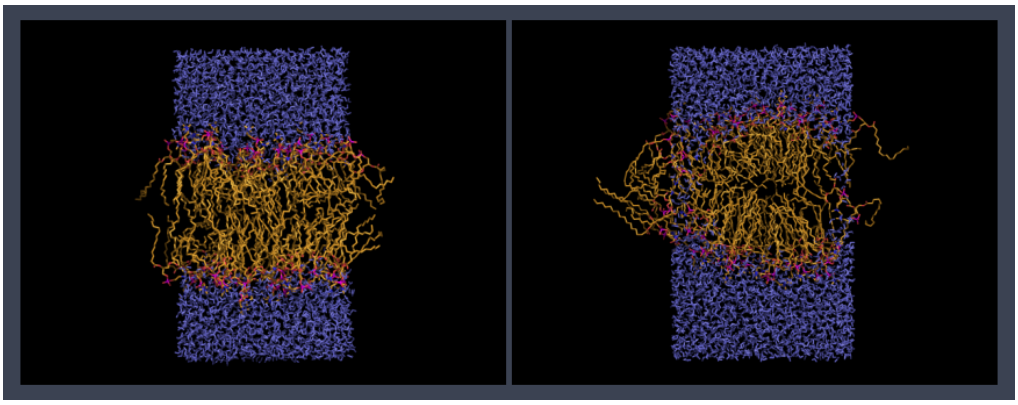

Рис. 1.30. Липидный бислой

Ван-дер-Ваальсовы взаимодействия – достаточно большой класс взаимодействий, в которые попадает ряд электронных эффектов: как дисперсионные, так и обменные. Эти эффекты достаточно тяжело считать в квантовой механике, но достаточно легко параметризовать в молекулярной механике. Самый простой метод параметризации Ван-дер-Ваальсовых взаимодействий – это формализм Леннарда-Джонса, он же потенциал 6-12, суть которого сводится к тому, что у нас две компоненты: на отталкивание и на притягивание. Чем меньше расстояние, тем сильнее работает отталкивание, чем больше – притягивание. Общий вид функции напоминает потенциал Морзе, но, в отличие от ковалентных потенциалов, здесь значения могут быть меньше нуля:

$$U_{V\,dW} = \sum_{i=1}^{N} \sum_{j=i+1}^{N} 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right), \tag{1.48}$$

где ϵ_{ij} – глубина ямы, σ_{ij} – параметр, описывающий то расстояние, на котором отталкивание и притягивание равны.

Это может быть расширено до потенциала Букингема. Параметры для разных типов атомов находятся очень просто: берут среднее арифметическое для σ, среднее геометрическое для ϵ, и таким образом для каждой пары атомов можно подобрать соответствующее значение параметров. То есть, уже не нужно иметь таблицу для 1600 значений, достаточно для 40, если в силовом поле всего 40 типов атомов.

$$V_{bh}(r_{ij}) = A_{ij} \exp(-B_{ij} r_{ij}) - \frac{C_{ij}}{r_{ij}^6} \tag{1.49}$$

$$\sigma_{AB} = \frac{1}{2}(\sigma_{AA} + \sigma_{BB}) \tag{1.50}$$

$$\epsilon_{AB} = \sqrt{\epsilon_{AA}\epsilon_{BB}} \tag{1.51}$$

В ряде силовых полей Ван-дер-Ваальсовы взаимодействия используются также при расчёте торсионных углов.

В реальности **водородную связь** очень удобно описывать специфической комбинацией кулоновских и Ван-дер-Ваальсовых параметров. Ван-дер-Ваальсова связь – это треугольник, в котором, варьируя параметры у водорода, у отрицательных атомов, не имеющие отношения к водородной связи, можно добиться того, чтобы это хорошо описывалось манипуляциями параметрами, напрямую не имеющими отношения к водородной связи. Также можно использовать функции с явным описанием угла между электронной парой и водородом, связью водорода с электроотрицательным атомом, но это делается редко, так как приводит к сложным расчётам и не даёт плюса в точности (рис. 1.31).

$$U_{HB} = \frac{A^{10}}{r} - \frac{C^{12}}{r} \tag{1.52}$$

$$U_{HB} = \left(\frac{C}{d^6} - \frac{D}{d^4}\right)\cos^m\theta \tag{1.53}$$

$$U_{HB} = \left(\frac{C}{r_{H\ldots Ac}^{10}} - \frac{D}{r_{H\ldots Ac}^{12}}\right)\cos^2\theta_{Don-H\ldots Acc}\cos^4\omega_{LP-Acc\ldots H} \tag{1.54}$$

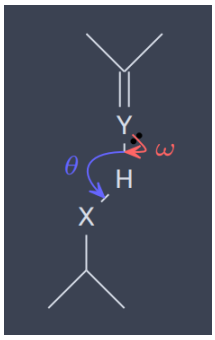

Рис. 1.31. Представление функции для описания водородной связи

Варианты ММ

Зачастую мы не можем, оценивая взаимодействие двух частиц, описать взаимодействие атомов так, чтобы они воспроизводили **свойства фазы**. Поясним, что это такое. Если у нас есть одна молекула воды, она ведёт себя каким-то образом (есть электронная плотность, образуются водородные связи, которые могут менять распределение зарядов в молекуле…). Теперь возьмём молекулу воды, которая находится среди других молекул воды. Во льду молекула воды образует 4 водородные связи. А в жидком состоянии в среднем 3 с чем-то. Когда такая высокая плотность

взаимодействий, статические свойства атомов отличаются от свойств атомов в молекуле в вакууме. Получается концепция фазы.

И для того, чтобы с этим бороться, мы зачастую при параметризации растворителя стараемся не просто оптимизировать параметры для описания взаимодействия атомов, а подгонять их так, чтобы они описывали свойства фазы: вязкость, дипольный момент и т. д. Это так называемый эффективный парный потенциал, когда мы пытается избежать тройных и прочих взаимодействий путём скалирования парных взаимодействий. Эти скалирования приводят к тому, что мы используем описание воды, которое характеризует состояние воды в фазе, в окружении других молекул воды. Поэтому есть появится задача, в которой вода ведёт себя как отдельная индивидуальная молекула, для неё надо будет придумывать своё локальное описание.

Существуют **три основных класса моделей воды**: простые модели, поляризуемые модели, ab initio модели. Рассмотрим первые два класса.

Рассмотрим **простые модели** (рис. 1.32). Вода – на вид очень простой объект. У него есть три атома – три точки в пространстве. Есть такие модели, которые описывают воду просто как три точки в пространстве. Для них мы не рассматриваем колебания связей. Здесь достаточно понимать, чтобы молекула двигалась целиком как жёсткое тело.

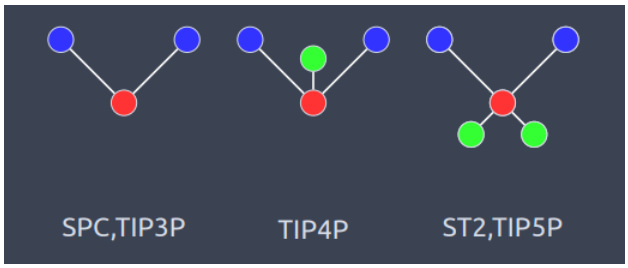

Рис. 1.32. Простые модели воды

Но это описание не всегда хорошо работает, и развитием данных моделей являются модели TIP4P, TIP5P (рис. 1.32). В TIP5P зелёные шарики – электронные пары, которые отображаются в виде виртуальных частиц с нулевой массой, которые находятся на каком-то расстоянии от ядра кислорода. А в TIP4P суммарное отображение этих электронных пар, смещённое в противоположном направлении. То есть, мы даём на кислород больше минуса, а на зелёный шарик больше плюса, и пытаемся таким образом уравновесить и иметь обобществлённое электронное облако, которое находится по центру. Эта модель ведёт себя очень хорошо, и с помощью неё можно моделировать образование льда: просто виртуально охладить воду, и она станет льдом.

Можно использовать **поляризуемые модели воды**. В них положение виртуального заряда смещается в зависимости от окружения. Такое часто случается в природе. Существуют поляризуемые модели не только воды, но и аминокислот. Но здесь

есть нюанс: каждый раз, добавляя дополнительную частицу для расчёта, мы этот расчёт сильно тормозим. Поэтому поляризуемые модели работают качественно, но долго.

Есть два подхода (рис. 1.33). В первом виртуальный заряд может колебаться вокруг атома на какой-то пружинке. Во втором есть несколько заранее заданных точек, и мы в этих точках варьируем значение заряда в зависимости от окружения. Первый алгоритм быстрее, но второй тоже когда-то активно использовался.

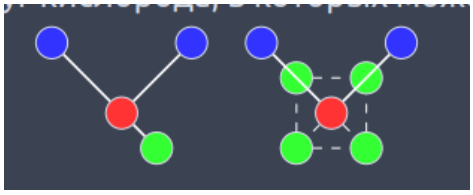

Рис. 1.33. Два подхода

Подобные поляризуемые модели справедливы и для белков, чаще всего когда белки взаимодействуют, например, с катионами металлов, потому что координация металлов в белках – это донорно-акцепторный процесс, и она не может быть рассчитана без явного учёта электронов, но может быть сделана с помощью таких частиц. Однако уже есть статьи о том, что хорошее взаимодействие катионов в металлах с белками достижимо и без поляризационых моделей с помощью простых.

Для биологических систем всегда существует проблема ускорения счёта. Если мы будем моделировать их в адекватном окружении – с мембраной, с водой и т. д., размеры систем могут иметь до полумиллиона атомов, а если мы хотим ещё и моделировать некую часть клетки, система станет размером в десятки миллионов атомов, и все взаимодействия надо будет рассчитывать.

Подобные задачи нуждаются в оптимизациях. Самая простая оптимизация – убрать все протоны, не участвующие в водородной связи, из системы, сделав их частью атомов, с которыми они связаны. Это называется **силовые поля с объединёнными атомами.** Однако здесь есть проблема. Если мы раньше, например, рассматривали $C\alpha$ атом с тем протоном, который делал его L-аминокислотой, и это было понятно, потому что там было явное описание тетраэдрического окружения атома углерода, то в случае, если мы этот протон объединим с $C\alpha$ углеродом, описание атома резко станет хуже, так как у нас оно будет треугольным (рис. 1.34). И чтобы L-аминокислоты не превращались в D-аминокислоты, надо использовать неправильные торсионные углы, чтобы держать аминокислоту в исходном виде.

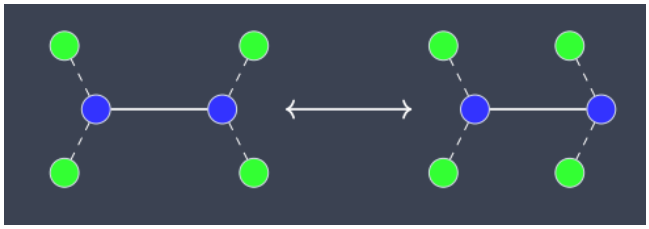

Рис. 1.34. Проблема

И всё равно это не является ключевым решением для больших систем, потому что у нас остаются заданные в явном виде протоны, которые участвуют в водородной связи, и их достаточно много. Количество атомов уменьшается, система ускоряется, но не очень существенно.

Более значительное укрупнение описания взаимодействий существует в **силовом поле Martini**. Идея в том, что мы несколько атомов, например, 4, объединяем в одну частицу. Тогда водороды как атомы исчезают как класс. Так как водород – самый лёгкий атом, у него самые быстрые колебания. И если мы от них избавимся, то можем рассматривать систему с шагом, гораздо большим, чем шаг при наличии водородов: от 1-2 фемтосекунд перейти сразу к 20-40. К тому же, размер системы тоже падает в 4-5 раз. Модели молекулярной динамики сложных систем начинают работать очень быстро.

Рассмотрим это на примере пептида. Возьмём просто случайную смесь липидов и воды. Меньше чем за 1,5 часа на обычном ноутбуке из неё можно будет получить хорошо упакованный липидный бислой и положение пептидов в нём (рис. 1.35).

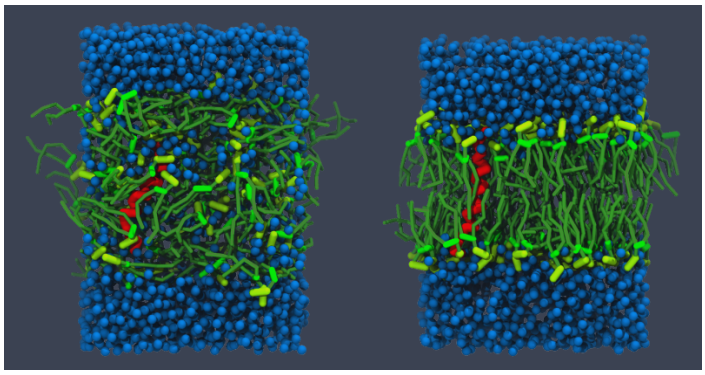

Рис. 1.35. Моделирование бислоя

Это нужно, чтобы считать системы действительно большого размера. Например, можно смоделировать динамику липосомы, в которой миллионы липидов (рис. 1.36). Сейчас липосомы – популярный способ доставки мРНК в некоторых вакцинах.

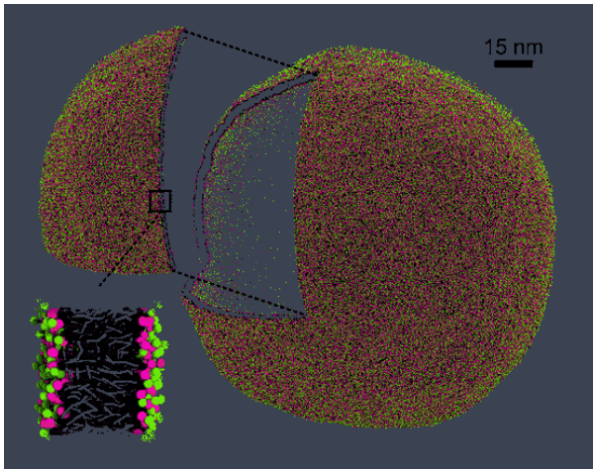

Рис. 1.36. Липосома

Силовые поля могут быть использованы не только для биологических объектов. Один из примеров использования силовых полей – в **молекулярной механике для физики твёрдого тела**, а именно для моделирования стекла. Кремний – важный объект в полупроводниковой промышленности, и многое держится на понимании механизмов формирования поверхности стекла.

$$U = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right) + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right) \tag{1.55}$$

$$U = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(D_0 \left(e^{r\left(1-\frac{r_{ij}}{r_0}\right)} - 2e^{\frac{r}{2}\left(1-\frac{r_{ij}}{r_0}\right)} \right) + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right) \tag{1.56}$$

Если в стекле постоянно образуются и разрушаются ковалентные связи, почему бы не описать их с помощью нековалентных потенциалов? Есть специальные силовые поля, где коэффициенты так адаптированы, что реально воспроизводят образование и разрыв ковалентных связей без квантовой химии. Это уже достаточно точно, чтобы работать с системами размера порядка миллиона атомов для моделирования поверхностей стекла.

Лекция 2. Хемоинформатика

Активные молекулы

Почему нас в структурной биоинформатике вообще могут интересовать органические молекулы? Потому что основная часть метаболитов, которые есть у живого – это небольшие органические молекулы, которые используются живым для накопления энергии, передачи сигналов и др. Также эти молекулы при определенных условиях могут быть использованы как некие инструменты для манипулирования активностью белков.

Здесь возникает такой термин как лекарства или биологически активные соединения. Потому что, если какое-то соединение не активно, то есть не влияет ни на какой белок или биополимер, трудно ожидать, что оно имеет какую-то биологическую активность. Биологическая активность произрастает из того факта, что эти молекулы способны образовывать нековалентные комплексы с белками и вообще с биополимерами.

В этой области есть интересная терминология, которая относится к тому времени, когда основные лекарства разрабатывались как объекты по отношению к рецепторам. Тогда появились такие термины как **агонист, антагонист** и **обратный агонист**. Все три типа связываются с рецептором. Но рецепторы не просто выполняют свою функцию, они ещё и передают сигнал. Сигнал может быть передан по-разному. Если в том виде, в котором связываются нативные сигналы, то это агонист. Антагонист препятствует связыванию нативной молекулы и прохождению сигнала. Обратный агонист даёт эффект, обратный связыванию сигнала.

Для того, чтобы молекула хорошо связывалась с белком, она должна быть комплементарна поверхности белка. То есть, у нас есть белок, у него на поверхности разные атомы – гидрофобные, гидрофильные, способные образовывать водородные связи и не способные, и эти атомы образуют нековалентные взаимодействия с органической молекулой. Чем больше поверхность этого контакта, тем более эффективно взаимодействие. Поэтому, когда мы говорим о комплементарности поверхности, мы считаем: чем большая доля органической молекулы взаимодействует с белком, тем больше поверхность взаимодействия, значит, тем оно лучше. Сам тип взаимодействия при этом не так важен.

Само по себе взаимодействие с биополимером – это не обязательное условие для того, чтобы вещество было **лекарством**, потому что для лекарства есть ещё дополнительное свойство: оно должно не только действовать на белок, но и как-то работать в рамках всего организма. И, самое важное, лекарство должно иметь приемлемую растворимость, то есть концентрация лекарства в воде должна быть таковой, чтобы как минимум сравняться с концентрацией объекта, на которой оно действует, а лучше быть ещё больше, потому что если концентрация меньше константы диссоциации, эффекта может и не быть.

Немаловажным бывает, когда часть объектов, на которые действует лекарство, находится внутри клетки. Тогда лекарство ещё должно уметь проникать сквозь мембрану. Тут оно должно быть уже не только хорошо растворимым, но и умеренно гидрофобным.

И самый хороший вариант – когда лекарство эффективно метаболизируется, то есть в организме есть ферменты, которые способны его расщепить, нейтрализовать, переварить так, чтобы в обработанном состоянии оно уже не имело активности. Если вещество будет накапливаться в тканях, в определённый момент оно может высвободиться и привести к коллапсу жизнедеятельности.

Как ищут активные молекулы? Можно искать в биоматериалах, выделять из растений и т. д. Но такая деятельность уже закончена, большинство уже изучено. Можно проводить роботизированное сканирование больших библиотек и проверять их в разных тестах. Даже сами реакции по набору библиотек могут выполняться роботами. Однако любой роботизированный анализ порождает большое число данных, которые надо обработать так, чтобы получались правильные ответы, а не набор сигналов, который потом трудно интерпретировать.

Разумеется, возможен высокий уровень шума. Это зачастую связано с тем, что сама тест-система может быть не очень чистой. Также могут быть побочные эффекты от действия органического вещества или смеси веществ.

Для всех этих задач активно используются методы информатики во всех её применениях: биоинформатики, хемоинформатики и т. д.

Фарминдустрия

Важно понимать особенности **работы рынка фарм-производителей**, чтобы оценить важность каждого этапа. Дженерики – лекарства, которые уже лишились патентной защиты (вышел срок) и активно синтезируются всеми, кем можно. Эти лекарства почти ничем не отличаются от оригинальных, но так как идёт активная борьба за удешевление производства, дженерики уже могут быть менее чистые и, соответственно, менее эффективные.

Разработка новых веществ идёт долго. Основное время занимает доказательство того, что вещество является клинически применимым. Тут возникают 4 основные фазы: открытие ингибитора, доведение его свойств до лекарственного статуса, испытания и продажа.

Исследование – это первые две строчки на рис. 2.1, где идентифицируется болезнь, для неё определяется молекулярный механизм, и если его можно регулировать с помощью ингибиторов ферментативной или другой белковой активности, это быстро переходит к испытанию на животных. На них показывается общая токсичность и эффективность.

Рис. 2.1. R&D

Какие **новые технологии** могут привести к тому, что этот процесс будет ускоряться? Все технологии направлены на то, чтобы получать как можно больше общих данных, а из них как можно более достоверную информацию.

Это экспрессия на чипах – когда мы следим эффективностью работы трансляционно-экспрессионного аппарата. Также есть структуры – роботизированный поиск комплексов с кристаллами белка. Например, когда мы можем капать на кристаллы белка маленькие кусочки ингибиторов и из этих фрагментов находить те, которые связываются в нужном месте. Ещё используется высокопроизводительный поиск роботов. А также вычислительные подходы, связанные с виртуальным поиском (докинг, скрининг), а также использование методов комбинаторной химии для генерации библиотек определённого профиля, чтобы узнать, можно ли в них искать ингибиторы. Все эти способы используются для расширения выборки.

HTS

Хемоинформатика помогает во многих частях. Это разработка методов и управление информацией о лигандах, оценка данных in silico для минимизации рисков из-за большой стоимости исследования: разработка библиотеки, виртуальный поиск, оценка стоимости и выгоды. А также организация доступа к информации и интеграция процессов.

Рис. 2.2 – пример того, как может выглядеть робот, который может обрабатывать до 100 тыс. соединений в сутки на сканирование ингибирующей активности. Его запуск – вещь дорогая, и наша задача в том, чтобы научиться хорошо контролировать как входные, так и выходные данные.

Рис. 2.2 Робот для высокопроизводительного поиска ингибиторов

Когда этот прибор отработал, у нас есть некие **данные**, что есть некий уровень сигнала и какой-то уровень шума. Для того, чтобы понять, есть ли в этой библиотеке что-то кроме шума, используются методы выявления сигнала. После того, как отобраны соединения, которые дают относительно хороший сигнал, необходимо показать, что эти соединения имеют что-то общее: их надо кластеризовать, дальше визуализировать для проверки человеком и провести идентификацию «основы» для каждого класса. Это некий общий скелет, который характерен для каждого кластера – для того, чтобы мы могли оценить, есть ли какая-то общая идея у соединений, которые ингибируют тот или иной белок. Дальше очень важен поиск причин и элементов структуры, причём не только тех, которые приводят к активности, но и тех, которые приводят к потере активности, потому что следующим этапом будет попытка изменить лидерное вещество, чтобы оно уже могло быть лекарством, то есть, чтобы у него были определённые свойства по растворимости, проницаемости через мембрану, метаболизации. После этого мы можем попытаться объяснить структурные причины, почему это происходит.

Один из примеров, как можно работать с результатами подобного анализа, это комбинаторная химия. Это попытка делать в одной пробирке сразу много веществ исходя из того, что они могут взаимодействовать с разными точками в остове или в скаффолде одновременно, и таким образом получать сразу комбинацию соединений.

Это проще смотреть на примере (рис. 2.3). Пусть у нас есть некий бицикл, в котором есть три позиции, на которые можно присоединить радикалы. Если в каждой группе радикалов примерно по пять участников, можно очень легко понять, что получится 125 комбинаций.

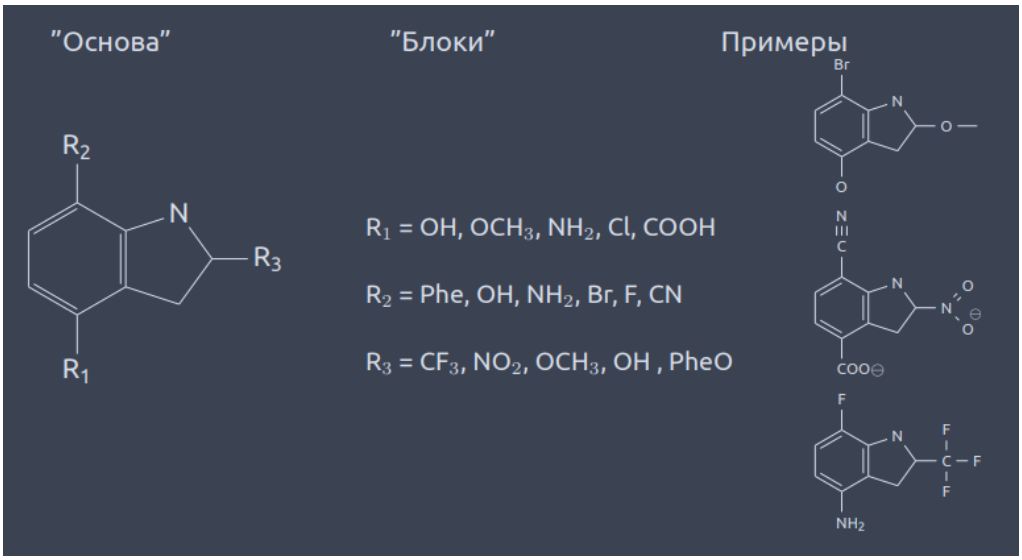

Рис. 2.3. Комбинаторная химия

То есть, закинув смесь в пробирку, в ней можно получить 125 соединений после простого термофазного синтеза. Это полезно в зависимости от того, насколько хорошо мы потом умеет различать сигналы от ингибирования. Если все эти соединения одновременно вылить на какую-то чашку и понять, есть там ингибирование или нет, потом можно быстро отбросить или выявить некий набор соединений как потенциальных ингибиторов белков.

Также полезно отвечать с помощью хемоинформатики на следующие **вопросы**. Какие белки выбирать, чтобы искать лекарства? Какие библиотеки строить? И ещё было бы правильно, имея о белке достаточно примитивную информацию, научиться сразу профилировать библиотеки под конкретный белок. Допустим, мы знает, что у белка есть активный сайт, и знаем, как он примерно выглядит. Было бы неплохо придумать способ, который позволил бы на основе этого знания сразу отсечь большое количество соединений, которые нам не нужно проверять экспериментально.

Хемоинформатика

Перейдём к тому, **что надо делать**. Очевидно, что любое органическое соединение, которое мы можем нарисовать, можно хранить в компьютере в нарисованном виде, но это не очень удобно, потому что не очень хорошо работает. Тогда любое органическое соединение можно представить в виде графа, у которого есть узлы и рёбра. Также граф можно представить в виде таблицы связей (рис. 2.4). В представленной на рисунке молекуле 4 атома: COCO. В такой записи о молекуле не упоминаются атомы водорода, так как они добавляются потом из соображений валентности. Это уксусная кислота, так как с 1-м атомом связываются и 4-й, и 2-й, и 3-й атомы, то есть есть атом, от которого отходят остальные три, и два их них атомы кислорода, значит, карбоксигруппа.

```
Marvin  04200617372D

  4  3  0  0  0  0            999 V2000
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7145   -0.4125    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -0.7145   -0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.8250    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
  1  4  2  0  0  0  0
  2  1  1  0  0  0  0
  3  1  1  0  0  0  0
M  END
```

Рис. 2.4. Компьютерное представление молекул

Здесь молекула также представлена в виде неких координат. Они могут быть как двумерными, чтобы просто уметь рисовать молекулу в 2D-пространстве и показывать пользователю, либо трёхмерными. В случае плоской молекулы, окружения атома в sp2-гибридизации, как здесь, будут нули. Есть два типа подобных описаний: для того, чтобы показывать молекулы в плоском состоянии, и в 3D. Если мы хотим представить эту молекулу в виде конформера, нужно изменить координаты так, чтобы они соответствовали пространственной организации этой молекулы.

Есть гораздо более удобный способ, который называется SMILES. Основная идея опять в том, что молекула является графом, и мы один раз проходим по молекуле каждый атом и из этого рисуем линейное представление молекулы (рис. 2.5). С помощью нотации SMILES можно описывать не только молекулы, но и их реакции. Эти реакции нужны для того, чтобы в будущем попытаться отсеивать, например, те молекулы, которые мы не можем синтезировать по химическим правилам. Есть специальная база, куда все подобные реакции уже записаны. Её можно использовать для того, чтобы генерировать для любой молекулы, которую можно сгенерировать с помощью машинных методов, ещё способ её синтеза.

CC	ethane	[OH3+]	hydronium ion
O=C=O	carbon dioxide	[[2H]]O[[2H]]	deuterium oxide
C#N	hydrogen cyanide	[[235U]]	uranium-235
CCN(CC)CC	triethylamine	F/C=C/F	E-difluoroethene
CC(=O)O	acetic acid	F/C=C/F	Z-difluoroethene
C1CCCCC1	cyclohexane	N[[C@@H]](C)C(=O)O	L-alanine
c1ccccc1	benzene	N[[C@H]](C)C(=O)O	D-alanine

Реакции в виде SMILES

[I-].[Na+].C=CCBr » [Na+].[Br-].C=CCI	реакция замещения
(C(=O)O).(OCC)»(C(=O)OCC).(O)	образование сложного эфира

Рис. 2.5. Молекулы и реакции в виде SMILES

Одну и ту же молекулу можно записать с помощью SMILES несколькими разными способами. Когда мы захотим сравнивать молекулы между собой, они должны

будут иметь некий стандартизированный вид (рис. 2.6). Стандартизация SMILES называется Unique SMILES. Это некая процедура, основанная на правилах IUPAC, для того, чтобы переводить последовательность атомов в такую, в какой они были бы при наименовании молекулы по правилам IUPAC. Тогда дальше из разного набора вариантов представления одной молекулы мы можем реализовать её описание на уровне Unique SMILES. Это важно для того, чтобы не было шума из-за разницы в написании.

Input SMILES	Unique SMILES
OCC	CCO
[CH3][CH2][OH]	CCO
C-C-O	CCO
C(O)C	CCO
OC(=O)C(Br)(Cl)N	NC(Cl)(Br)C(=O)O
ClC(Br)(N)C(=O)O	NC(Cl)(Br)C(=O)O
O=C(O)C(N)(Br)Cl	NC(Cl)(Br)C(=O)O

Рис. 2.6. Стандартизация SMILES

Однобуквенные **атомы в SMILES,** а именно B, C, N, O, P, S, F, I, записываются в виде в виде однобуквенных кодов, однако есть исключения: например, Cl и Br. Все остальные атомы записываются в квадратных скобках (так делается любая группировка SMILES). Атомы водорода обычно не указываются в явном виде, но мы хорошо понимаем, что один и тот же атом может находиться в разных состояниях, например, нейтральный азот и протонированная аминогруппа. Это указывается с помощью скобок, в которых мы дополнительно указываем свойства текущей группы и в явном виде указываем, допустим, валентность или заряд. Без этого алгоритмы пытаются сделать вывод на основе наблюдаемой валентности. Допустим, если мы наблюдаем рядом с азотом трёх соседей, если это углероды, у него валентность три, а если кислороды, то пять.

Если **связь** одинарная, то ничего не пишем (рис. 2.7). Если двойная – пишем знак «=», если тройная – «#», а в экзотических случаях пишем квадратные скобки, в явном виде указываем элемент, и становится понятно, какая связь в данном случае между этими атомами.

CC	этан
C=C	этилен
O=C=O	CO2
C#N	HCN
CCO	этанол
[H][H]	водород

Рис. 2.7. Обозначения связей в атомах

Ветвление цепи отображается в круглых скобках. Ответвление мы всегда можем указать тем способом, каким удобно. Но после этого полезно применить процедуру Unique SMILES для того, чтобы это соответствовало наименованию IUPAC.

Теперь рассмотрим **циклы**. Они могут быть сложными и их можно делать по-разному. Допустим, у нас есть простой цикл – циклогексан (рис. 2.8). Мы записываем в линейку 6 атомов углерода, а дальше сообщаем алгоритму, что хотим закоротить первый и шестой атомы. Для этого добавляем к этим атомам флажок (цифру) 1, и алгоритм при построении структуры их соединяет. Мы может делать прописывание индекса как в прямой цепи, так и при планировании квазиветвления, это не мешает замыканию цикла.

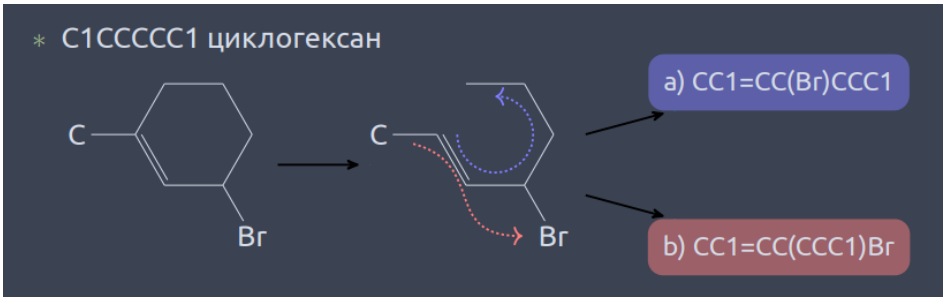

Рис. 2.8. Циклогексан

Есть более сложные варианты, когда у нас много циклов. Самый сложный вариант в углеродной химии, наверное, это кубан (рис. 2.9). У него каждый атом находится в трёх циклах.

Рис. 2.9. Кубан

Тогда мы должны у каждого атома, участвующего в замыкании цикла, поставить не менее чем по два индекса. Поэтому надо аккуратно пройти красным путём по всем атомам в молекуле, чтобы все атомы попали в линейное описание, а потом расставить индексы замыкания циклов.

Все индексы могут быть от 1 до 9, двузначных не существует, так как если первый цикл замкнулся, следующий по последовательности букв можно назвать первым. Только взаимно пересекающиеся циклы должны меняться в номерах.

У каждого цикла можно указать **свойства ароматичности**. Ароматичность вычисляется по расширенному алгоритму Хюккеля. Можно указать, что цикл является ароматичным, ставя все буквы маленькими. Но есть нюанс: если написать все буквы маленькими, а цикл не будет удовлетворять правилу Хюккеля, то он будет не ароматичным, система прочтёт буквы как большие.

Если будем писать указания двойных связей в явном виде, надо учесть, что замыкание циклов происходит через одинарную связь (рис. 2.10), поэтому чередование двойных связей должно начинаться с первого атома с двойной связью.

Ароматичными могут быть атомы: C, N, O, P, S, As, Se и *.

Нередко бывает так, что нужно для соединения указать в явном виде, где находится протон. Допустим, азот в ароматических циклах, с одной стороны, может давать двойную связь, с другой – неподелённую электронную пару для образования ароматической системы (рис. 2.10). И в последнем случае у него должен быть в явном виде указан протон. Иначе алгоритм представления молекул может не сработать.

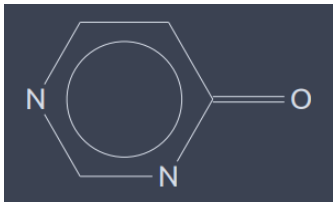

Рис. 2.10. Бензол

В принципе, можно указать и нековалентные взаимодействия. Например, ионные пары (рис. 2.10). Любая точка означает прекращение описания набора ковалентных связей, просто рядом что-то находится. Аналогично, через точку можно указывать реагенты в реакциях.

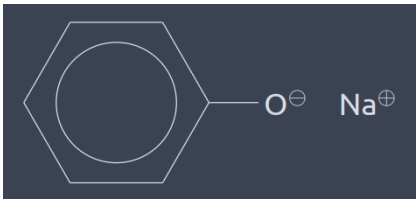

Рис. 2.11. Ионные пары

Изомеры могут быть изотопные (рис. 2.12). Здесь происходит группировка с помощью квадратных скобок. Могут быть цис- и транс- изомеры: вокруг двойной связи может быть расположение некоторых заместителей – как цис, так и транс. Если «/», то это цис изомер, если «\», то транс.

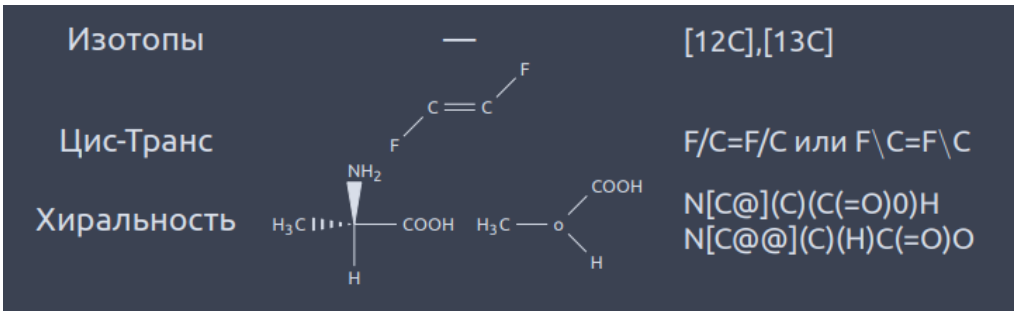

Рис. 2.12. Изотопы и хиральность

Когда мы рассматриваем хиральный центр (рис. 2.12), у нас должен быть атом, который является им и находится между двумя другими атомами. Три атома уже есть, осталось указать, куда идёт четвёртый. Это делается с помощью нотации «@». Если она одна, то описание идёт по часовой стрелке, а если две, то против часовой, заместители явно указаны. Для большинства лекарств хиральность имеет очень важное значение.

Есть достаточной простой язык **SMARTS** на основе SMILES, который позволяет искать паттерны, а с помощью паттернов находить молекулы, которые удовлетворяют некоторым свойствам (рис. 2.13). Если мы говорим о более сложных методах, например, о машинном обучении или векторном представлении, этот метод кажется примитивным, однако это всё равно хороший инструмент для того, чтобы делать предварительные фильтрации.

C	алифатический углерод
c	ароматический углерод
a	любой ароматический атом
[#6]	любой атом углерода
[++]	атом с зарядом +2
[R]	атом в кольце
[D3]	атом с тремя связями (не с атомами водорода)
[X3]	атом с тремя связями, включая атомы водорода
[v3]	атом с валентностью 3.

Рис. 2.13. SMARTS

Логика здесь очень простая. Если мы хотим объединить несколько свойств атома в одном запросе, группируем с помощью квадратных скобок, а в них перечисляем то, что должно быть (рис. 2.14).

Логика:

!e1	not e1
e1& e2	a1 and e2
e1,e2	e1 or e2
e1;e2	a1 and e2

Пример:

[!C;R]	не алифатический C в кольце
[n;H1], [n&H1], [nH1]	H в пирроле
[c,n&H1]	C или H в пирроле
[X3&H0]	Атом с тремя связями не с H
[c,n;H1]	N или C в связи с одним H1

Рис. 2.14. Логика SMARTS

Есть альтернативный способ представления молекул – это **InChI** = IUPAC International Chemical Identifier, который базируется тоже на IUPAC инверсии, но так как он гораздо хуже читается человеком, имеет гораздо меньшее распространение (рис. 2.15). Суть в том, что сначала есть брутто формула, потом через «/» перечисляем связанность с важными атомами водорода, которые могут быть специфическими, дальше перечисляем слой с кратностью связей и зарядами атомов, и последний слой через «/» с указанием хиральности некоторых атомов.

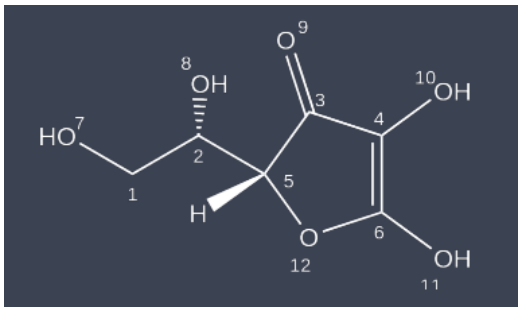

Рис. 2.15. InChI

QSAR

Дальше у нас впервые появляется такое понятие, как **дискриптор**. Самые примитивные дискрипторы опираются на **правило Лепински**, которое гласит о том, что молекула может быть лекарством, если она образует водородные связи, умеренно гибкая, умеренно гидрофобная. Есть несколько вариантов правил Лепински, но основное звучит так: должно быть не больше 5 доноров водородной связи, не больше 10 акцепторов водородной связи, молекула должна быть размером меньше 500 дальтон, разделение вода-этанол (коэффициент, отражающий липофильность) должно быть не больше 5. Очень жирные или очень гидрофильные молекулы нам не нужны.

Можем ли мы **искать по 3D-базам данных**, которых не так уж много? Если мы будем просто искать наши молекулы, основываясь на атомах типа SMARTS, будут слишком подобные соединения. Но, если мы хотим найти то, что имеет смысл при взаимодействии с белком, нам надо ввести такое понятие, как 3D-фармакофор (рис. 2.16). Его суть в том, что у нас есть какие-то группы в пространстве на каком-то расстоянии друг от друга.

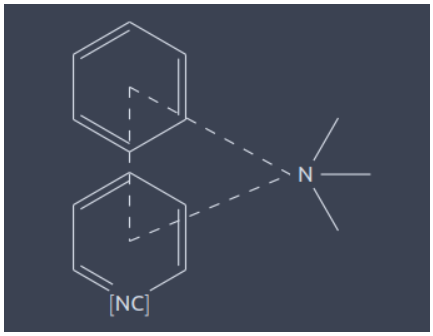

Рис. 2.16. Антигистаминный 3D-фармакофор

Дискриптор – это численное описание молекулы. Оно может быть нескольких типов. Самый примитивный дискриптор – 5 доноров водородных связей, 10 акцепторов водородных связей. Просто считаем количество атомов, которые могут быть донором или акцептором, и каждая запись будет отдельным дискриптором. Мы нарисовали SMILES, знаем формулу вещества и отсюда можем посчитать какой-то дискриптор.

Можно вещество разбить на группы (фрагментное разделение), у каждой группы есть какое-то значение, потом эти значения суммируют.

К молекуле можно применить исследование, связанное с её **3D-конформацией**, но эту конформацию надо откуда-то взять. Можем ли мы применить сканирование по 3D-фармакофору к нарисованным 2D-молекулам? Нет, так как в 2D не всегда очевидно расстояние, вращения могут сильно его увеличивать или уменьшать. Значит, мы должны построить набор конформаций для каждого органического вещества, после чего делать по этому 3D-поиск. Но тут уже надо посмотреть, какая база используется, что в ней есть и можно ли там реализовывать расчёты, связанные с 3D-поиском. Здесь есть сильные ограничения, потому что однозначного способа предсказания 3D-структуры органических молекул нет.

Если есть хоть какая-то подвижность структуры вокруг радикалов, никогда нельзя сказать точно, в каком состоянии лиганд будет связываться с белком. Это состояние может быть даже невыгодным, но связывание с белком делает весь процесс выгодным. Например, есть транс-изомер. Он живёт в растворе, но с белком связывается в цис-форме. В свободном поиске мы всегда будем находить самую оптимальную, и она не будет цис-. Это приводит к тому, что предсказывать структуры лигандов без понимания того, как они могли бы связаться с белком, не очень правильно.

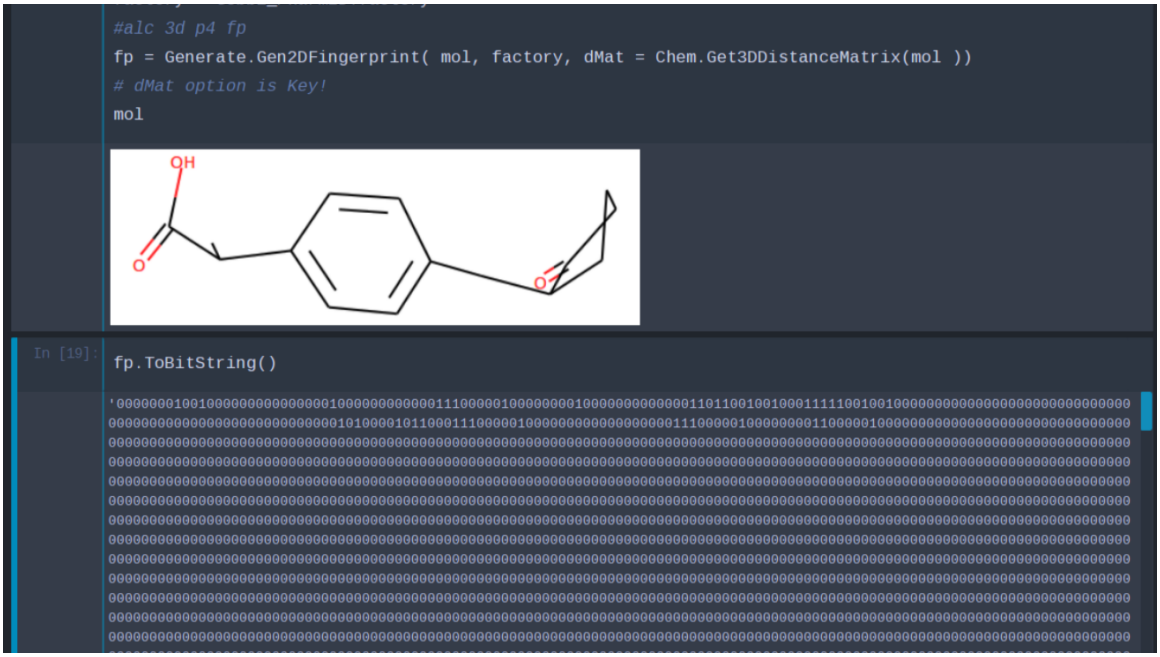

Рис. 2.17. Превращение 3D-конформации в вектор

На практикуме строили 3D-конформацию и превращали её описание в виде вектора (рис. 2.17). Здесь мы уже попадаем в область применения данных методов в машинном обучении.

Мы их уже называли некоторые **проблемы с фармакофорами.** Надо отметить, что не всегда можно понять, какой набор групп будет располагаться в биополимере идентично. Группы могут быть очень похожи, но их взаимное расположение неинвариантно.

Когда мы делаем обучающую выборку для построения регрессии, мы должны точно понимать, что все эти молекулы связываются одинаковым способом в одном и том же месте белка. В обучающей выборке мы всегда знаем, что такой структуре соответствует такая ингибирующая активность. Их у нас умеренно много. Дальше мы пытаемся построить регрессию, машинную модель, обучить нейронную сеть, и дальше на основании полученного результата предсказать активность тех соединений, для которых она неизвестна, основываясь только на их описании (SMILES, 2D и 3D структуры и т. д.).

Можно ещё пользоваться **систематическим поиском**, чтобы подбирать молекулы под конкретный белок. Идея очень простая: у нас есть точки в пространстве (положения определённых групп), и под них мы пытаемся подогнать конформации молекул так, чтобы туда попадало или не попадало. Элементов машинного обучения здесь ещё нет.

Баз данных очень много. Основные – PubChem (рис. 2.18), Cambridge database, Inorganic structural database.

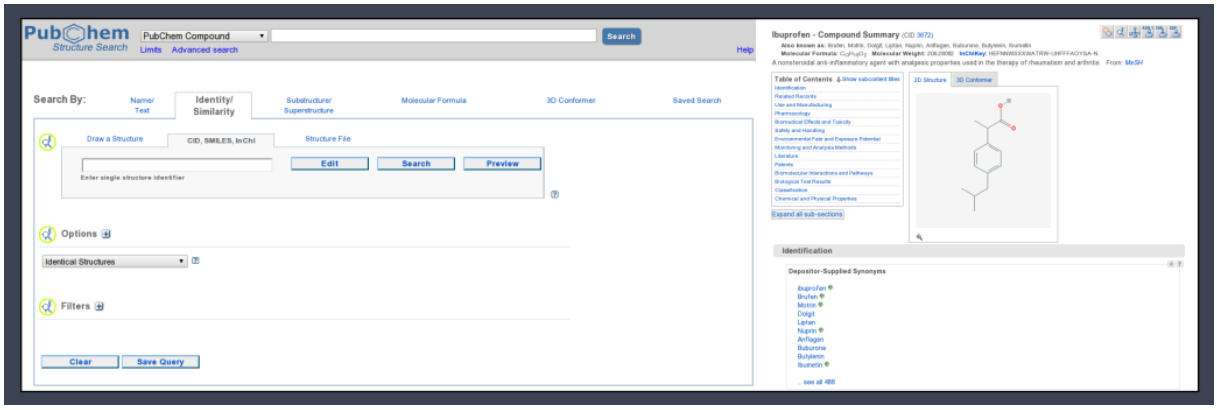

Рис. 2.18. PubChem

Как обычно мы работаем с веб-сайтами? Делаем поиск, получаем страницы выдачи, и чтобы скачать, например, 200 страниц, нужно нажать на скачивание 200 раз. Все эти действия можно описать с помощью XML запроса, в котором ищем что-то похожее на данное соединение. Мы посылаем запрос на сервер, под запрос генерируется место в очереди, и через пару часов сервер под него выдаст результаты. Так работает **pubcheapy** (рис. 2.19)**.**

```
In [1]: import pubchempy as pcp

In [21]: name='Aspirin'
         list=pcp.get_compounds(name, 'name')
         asp=list[0]
         hb=asp.h_bond_acceptor_count
         form=asp.molecular_formula
         xlogp=asp.xlogp
         print name, ": ", form," hb acceptors: " , hb , " Hydrophocibity: ", asp.xlogp

         Aspirin:  C9H8O4  hb acceptors:  4  Hydrophocibity:  1.2

In [30]: list=[]
         for i in range(1,10):
             try:
                 a=pcp.get_compounds('C1=N-C=C-N1', 'smiles',searchtype='similarity',listkey_count=50,listkey_start=i)
             except:
                 print "ended on: ",i*50
                 break
             list.extend(a)

         print "Downloaded: ", 50*len(list), "compounts"

         Downloaded:  22500 compounts

In [31]: list

Out[31]: [Compound(12749),
          Compound(82140),
          Compound(484),
          Compound(96125),
          Compound(283401),
          Compound(559542),
          Compound(2773261),
          Compound(2773328),
```

Рис. 2.19. Pubcheapy

В pubcheapy одному соединению соответствует запись SMILES и много дополнительной информации, в том числе и о действии данного вещества на биологические объекты. Если мы будет скачивать не только описания SMILES, но и целые записи, мы будем получать гораздо больше данных, но в output можно будет найти очень интересные вещи вроде активности по отношению к белкам и др.

Есть базы соединений, которые существуют, и базы соединений, которые пока не существуют. Разнообразие молекул соединений углерода во Вселенной примерно 10^{40}, и никакая база на данный момент такого не содержит. Даже базы типа GDN-17, где сгенерированные наборы из 166 млн. соединений, содержат не больше 10 тяжёлых атомов, из комбинаций которых построено столько соединений. И добавление каждого тяжёлого атома увеличивает данное количество в геометрической прогрессии.

ML в хемоинформатике

Перейдём к **применению методов машинного обучения**. Они могут быть использованы во всём, что называлось ранее, но с гораздо более высоким качеством. Каждый раз, когда мы делаем регрессию, поиск, классификацию, они эффективнее. А также есть генеративные возможности машинного обучения: если мы имеем выборку соединений, которая хорошо работает, можем под эту выборку сгенерировать новые соединения.

Дескрипторы с точки зрения машинного обучения – некие численные значения, вектора и др., которые мы можем использовать в методах, относящихся к области машинного обучения.

0D – просто числа: молекулярный вес, количество атомов и связей. 1D – графы: те же SMILES, фрагменты, функциональные группы. 2D – топология: индексы Weiner, Balaban, Randic, BCUTS. 3D – геометрия: WHIM, autocorrelation, 3D-MORSE, GETAWAY. 4D – добавление информации, например, о взаимодействиях с белком: Volsurf, GRID, Raptor.

Нульмерные дескрипторы – просто числа, которые можно получить из SMILES.

Одномерные дескрипторы – это скаляры: количество атомов, количество связей, молекулярный вес, суммы атомных свойств или количество фрагментов. Они страдают от вырожденности, когда под один и тот же набор дескрипторов попадают разные соединения. Одномерные дескрипторы обычно используются вместе с многомерными дескрипторами как дополнительная поддерживающая информация, которая позволяет проводить дополнительную дискриминацию молекул, чтобы можно было вычислительно понять, кто есть кто.

Двумерные дескрипторы, которые ещё можно отнести к топологическим, являются самыми распространёнными, так как получение 3D-структуры лиганда – вещь вычислительно затратная и зачастую не очень релевантная к тому, что мы хотим увидеть, то есть к той конформации, которая находится в активном состоянии с белком. Поэтому, возможно, есть смысл рассматривать просто формулы веществ и подразумевать в регрессии, что по 2D структуре вычисляется вероятность 3D структуры. Отсюда можно уже пытаться найти связь, активность и т. д.

Здесь появляется такое понятие, как топологические индексы: кто с кем связан и какие варианты взаимного расположения, молекулярные профили и двухмерные дескрипторы автокорреляции. Важной особенностью 2D-дескрипторов является инвариантность графа (любой SMILES можем сделать уникальным, то есть инвариантным).

Есть разные системы. Система Mold2 быстро генерирует около 200 типов 2D-дескрипторов для составления регрессии, которую хотим построить для больших наборов данных. Есть коммерческие варианты типа DRAGON, которые позволяют создавать до 5000 дескрипторов на нашей библиотеке, и это позволяет ещё более точно проводить поиск по базе данных.

3D дескрипторы чувствительны к структурным изменениям. В 3D структуре необходимо явно указывать, где находится какой заместитель. Можно включить в дескрипторы поверхность, попытаться описать её форму, объем, произвести дополнительные квантово-химические дескрипторы. Трехмерные химические дескрипторы позволяют легко идентифицировать «каркасы» или «скаффолды», которые формируют нужную связывающую активность (аффинность). Ключевым ограничением использования 3D дескрипторов является вычислительная сложность генерации конформеров и выравнивания структур.

Квантовая механика будет давать более точное описание структур органических соединений, чем молекулярная механика, но в большинстве случаев для 3D дескрипторов будет достаточно молекулярно-механического описания за редким исключением, когда есть нестандартные эффекты. Но даже если мы молекулярно-механически описываем 3D структуру какого-то вещества, здесь у нас будет много тонких моментов, которые могут привести к тому, что это не сработает. Самое неприятное в том, что при предсказании конформаций в некоторых случаях они будут соответствовать конформациям лиганда, которые связываются с белком, а в некоторых – нет. Там нужно будет затратить энергию, это будет конформация с не минимальной энергией, но она будет иметь возможность связаться с белком. Однако если мы делали дескриптор на основе оптимальной конформации, вещество будет потеряно в регрессии.

4D дескрипторы могут рассматривать в себе несколько структурных конформаций. А последнее поколение 4D дескрипторов – ещё и попытка описать окружение для каждой группы в органическом соединении. То есть мы как бы расширяем соединения так, что у них появляются нековалентные взаимодействия с группами из белка.

Можно использовать молекулярную динамику. Это способ наблюдения за тепловыми движениями систем. Когда мы делаем молекулярную динамику комплекса лиганда и белка, получаем множество конформаций и тем самым сильно обогащаем 3D дескриптор: и конформации лиганда, которые могут связываться с белком, и дополнительные конформации окружения белка вокруг этого лиганда, которые тоже можно использовать в дескрипторе.

Fingerprints (FP) – это многомерные векторы, элементами которых являются значения химических дескрипторов. То есть когда делаем комбинацию из нескольких дескрипторов, можем называть это FP. Комбинация может быть линейной или многомерной (стек из векторов). Например, MACCS – это двумерные двоичные FP (0 и 1), каждый из которых 166 бит указывает на наличие или отсутствие определенных ключей подструктуры.

Что такое ключи подструктуры? Например, в силовых полях элементов может быть немного, а типов взаимодействия много, но не бесконечно. Это верно для аминокислот и других биологических соединений. Для каждого типа строим колонку, и для каждого соединения ставим в колонку 0 или 1, когда описываем каждый атом. Получается вектор, который уже содержит в себе информацию о наличии каких-то групп внутри этого соединения. Группы могут быть классифицированы по-разному, их может быть много, длина этих векторов будет небольшой (вектор на 2048 бит – уже очень большое описание этого соединения).

Daylight FP и ECFP сканируют окружение каждого атома, то есть описывают молекулу как окружение центрального атома по разным срезам. Дескриптор позволяет выявлять окружения атомов разного размера. Это даёт возможность определять, есть ли в молекуле определённое окружение.

Рассмотрим пример – **FPCP** (рис. 2.20). Пусть у нас есть центральный атом. Мы решили по IUPAC, что его можно выбрать в качестве первого. Дальше мы можем создать вокруг него несколько сфер. Нулевая сфера – он сам (азот). На втором шаге оказывается, что у него есть заместитель водород, ставим флажок в соответствующей строчке описания вещества. Третий шаг – дополнительное описание по типу атома, что там есть карбонильный углерод и алифатический углерод, тоже ставим 0/1 в соответствующих местах. То есть, делая срез на трёх, получим дескриптор, который хорошо описывает достаточно окружение атомов. Таких дескрипторов можно генерировать много и по ним отбирать соединения, строить регрессии, активности, структуры.

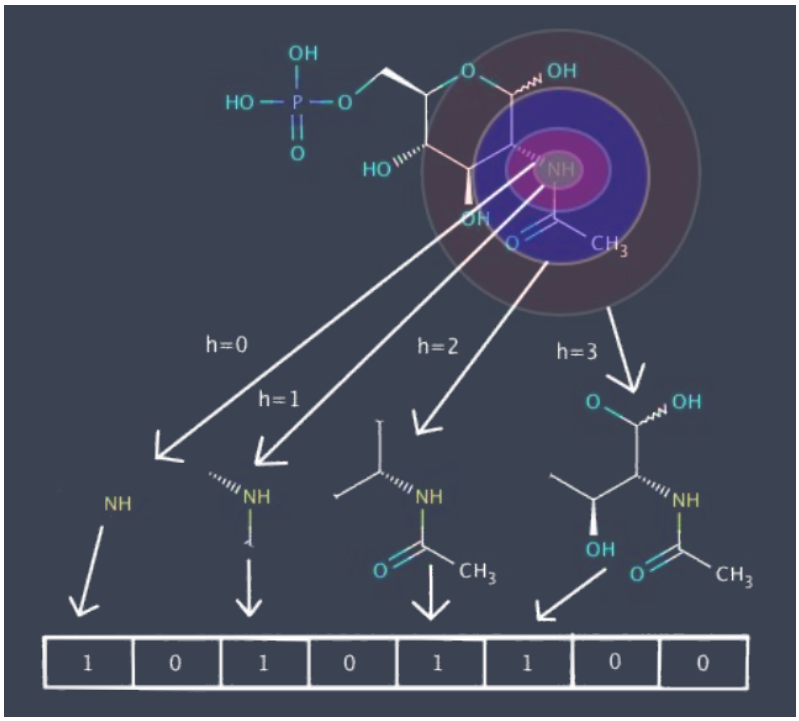

Рис. 2.20. FPCP

Строим бинарные вектора – таблицы для каждого дескриптора. Допустим, делаем 200 дескрипторов на каждое вещество. И когда строим регрессию, можно выяснить, сколько из этих дескрипторов влияет на активность. Потом сохраняем эти дескрипторы как хорошие и с помощью таких наборов можем быстро просканировать всю большую базу данных.

На старте должна быть равновзвешенная выборка, в которой есть и активные, и неактивные соединения.

Описанный выше вариант дескриптора – это просто численное соответствие. Было бы гораздо лучше иметь гладкую функцию, которая может быть проинтегрирована вперёд и назад. Это всё уже хорошо сделано.

Ещё можно использовать концепцию свертки на молекулы, представленные в виде двумерных молекулярных графов. Здесь мы пытается постоянно сужать графы до

одного вектора, чтобы получить это описание просто на основе дополнительной встроенной обработки.

3D FP включают химические характеристики, основанные на фармакофорных паттернах, свойствах поверхности, молекулярных объемах или взаимодействия молекул. Их самая удачная реализация – когда учитывается взаимодействие с белком.

Как в принципе реализованы FP, которые описывают взаимодействия? Тут всё идёт через MIF, реализованное в GRID. Мы разбиваем пространственные решётки, в узлах решётки можем давать свойства и говорить, что на определённом расстоянии шагов по решётке должно быть что-то определённое. Здесь в каждом узле решётке можно рассматривать и гидрофобные взаимодействия, и водородные связи.

Здесь можно перейти к более сложным вещам – попытаться построить молекулярные поля в попытке описания того, как должна выглядеть молекула во взаимодействии с белком, и из этого построить корреляцию с активностями соединений. В точке пространства мы можем разместить разные группы. Например, акцептором водородной связи может быть как карбонильная, так и OH группа, только одна лучше, а другая хуже. И если у нас будет построена регрессия, которая опирается на это, это в сумме будет называться упрощённым молекулярным полем.

Относительная ориентация молекул внутри сетки является основным ограничением. Может случиться так, что мы неудачно расположим молекулу в пространстве, когда будем пытаться её сравнить с моделью. То есть у нас есть сетка, и модель знает, что по данным этой сетки в некоторых местах должны находиться некоторые атомы. Значит, все новые соединения надо так расположить в этой сетке, чтобы было максимальное попадание в эти условия. Но сама эта процедура нетривиальная.

Можно попытаться делать непрерывное молекулярное поле (CMF), которое заменяет сетку непрерывными функциями. Тогда с помощью дифференцирования можно попытаться находить оптимальное положение, чтобы был максимум по какой-то функции расположения внутри модели.

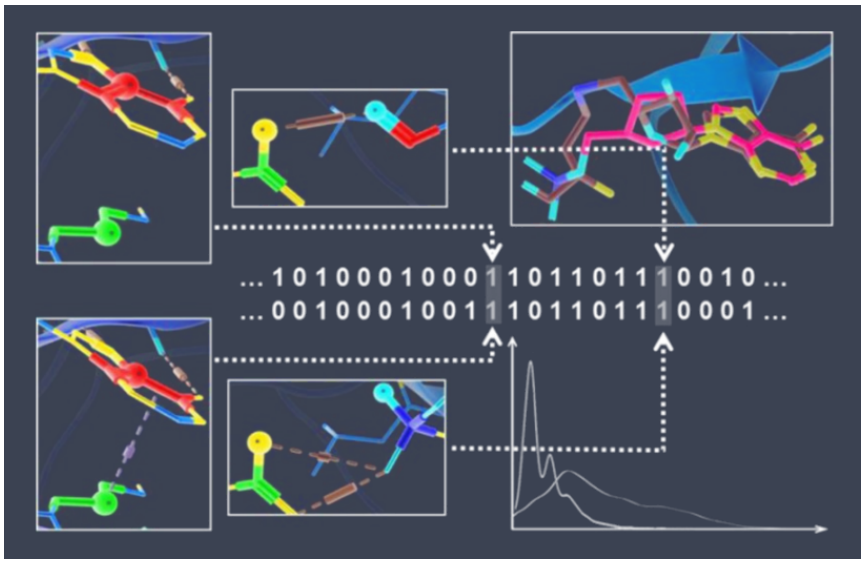

Рис. 2.21. SPLIF

Пример того, как это выглядит – дескриптор **SPLIF**. Он достаточно гладкий, описывает расстояния до белка.

Более понятное описание – на рис 2.22. У нас есть 3D координаты комплекса лиганда с белком. Для каждого атома лиганда мы определяем, какие атомы белка находятся рядом, и когда мы описываем взаимодействия, мы их экстраполируем как на 2D фрагменты. После этого записываем всё это в соответствующие координаты в бит (да/нет), и уже потом можно делать из этого обучение модели и добиваться, чтобы расположение лигандов хорошо соответствовало их активности.

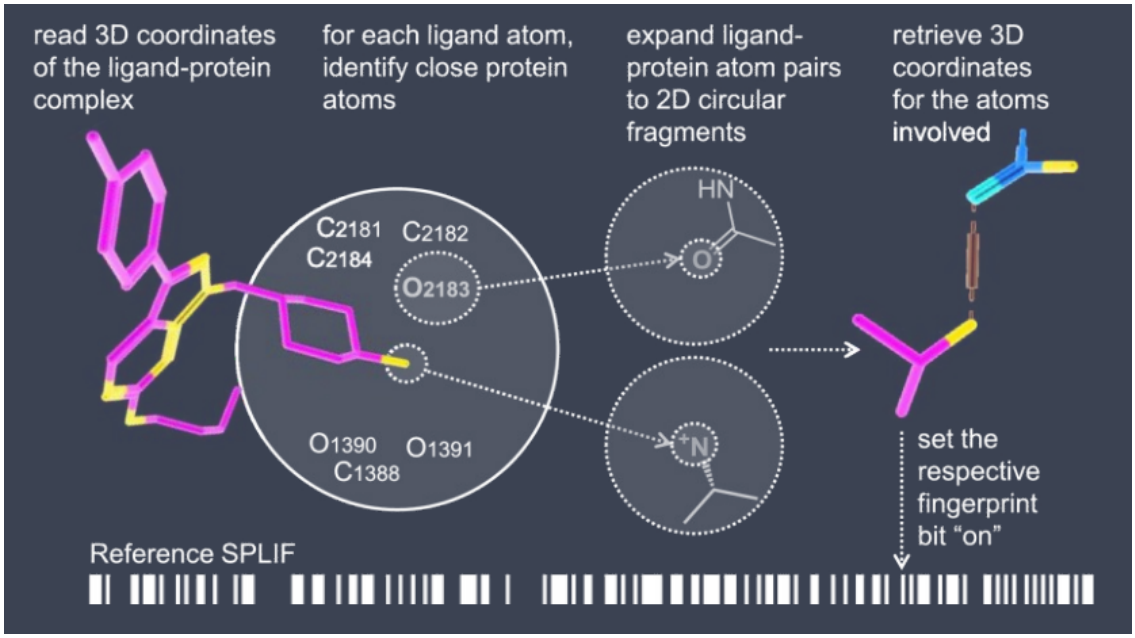

Рис. 2.22. Описание SPLIF

Это 4D дескриптор. 4D дескриптором иногда называют, когда есть экстраполяция на белок, а в других случаях – когда уже есть мультиконформация лиганда или даже молекулярная динамика лиганда с белком. Это будет сильно удлинять вектор, но не бесконечно.

Хороший пример применения данного метода – **ингибиторы киназ**. Киназы – ключевой класс ферментов, которые участвуют в передаче сигналов в клетках. Ингибирование сигналов – важная вещь, в том числе и для борьбы с раком. Известно много ингибиторов киназ. Но есть проблема: сайт связывания киназы очень похож на сайт связывания ATP по сути, потому что кинирование – это перенос фосфата из ATP на какой-то белок. А ингибировать все ферменты, которые связывают АТФ – плохая идея. Поэтому поиск хороших ингибиторов определённого вида киназ сложный.

Поэтому люди, когда строили модель, обучали по связыванию на конкретной киназе, брали окружение конкретной киназы для конкретного лиганда (рис. 2.23). То есть, мы можем брать для обучения не все киназы, а нужную часть. Это обучение привело к тому, что есть определённые дополнительные контакты с белком определённого типа в определённом бите. Включение мишень-специфичной информации через **IPF** улучшило предсказание эффективного связывания примерно на 10% по сравнению с испольованием традиционных FP. Это хороший результат, так как задача достаточно сложная.

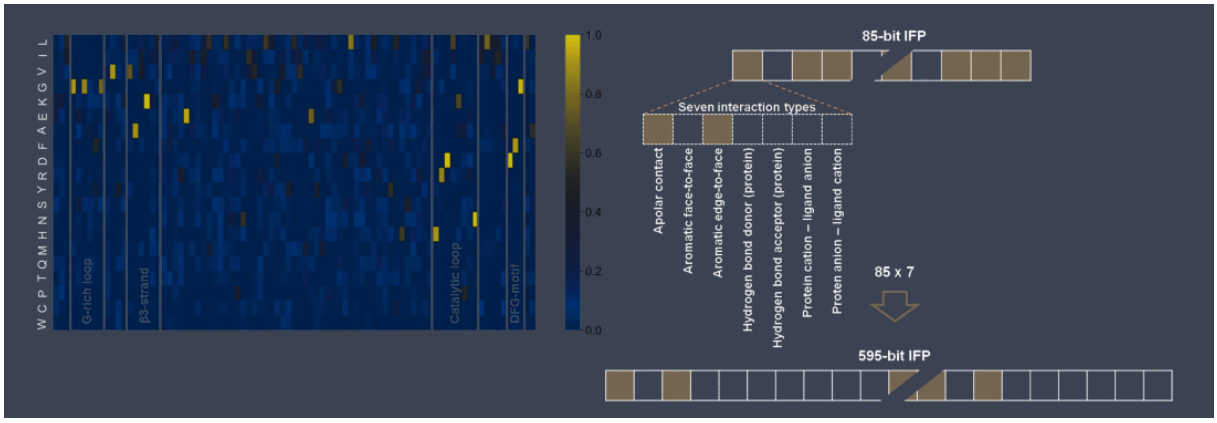

Рис. 2.23. IPF

Отсюда мы можем узнать Feature importances. Каждый бит – это некая Feature, и мы можем вычислить, какие биты вносят наибольший вклад, допустим, в активность. Азот в исследовании связывания с киназами всегда даёт хороший результат, так так атомы азота протонированы, а АТФ имеет заряд минус 4 (положительный заряд) (рис. 2.24). Значит, эти соединения могут ингибировать киназу. Взаимодействие с белком уже на малом количестве шагов обучения даёт хороший результат, значит, в статье описана высокочувствительная метрика, которая позволяет быстро вылавливать идею процесса.

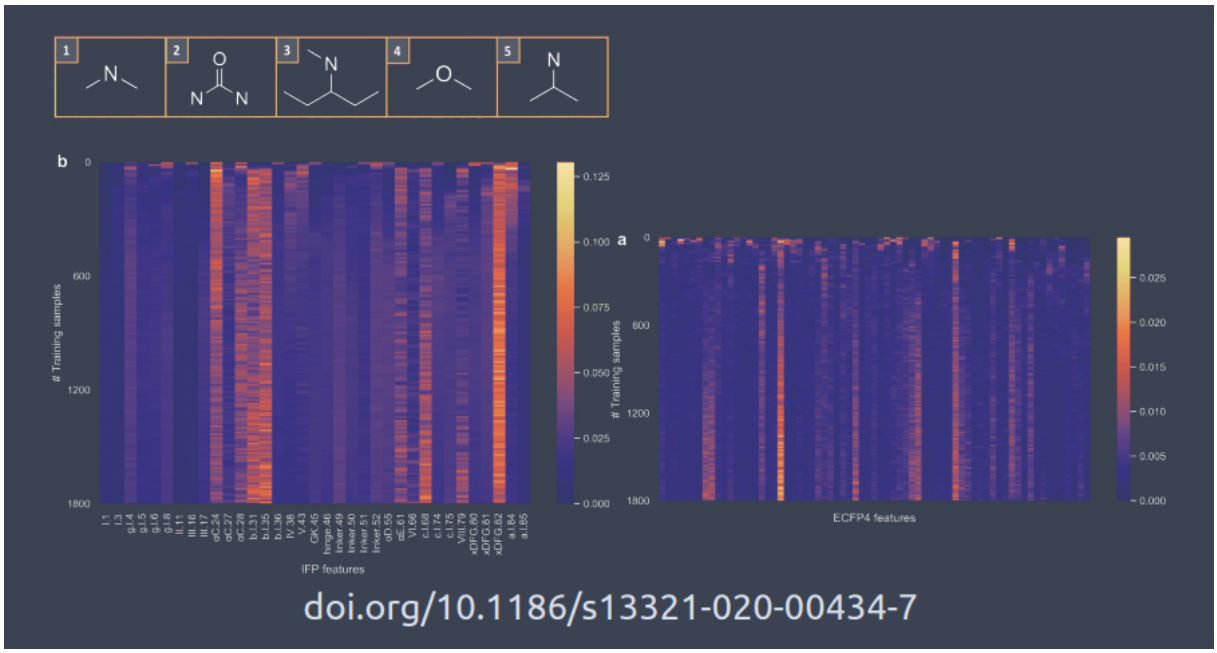

Рис. 2.24. Features

Можно включать ещё и **bioassay** – дополнительно добавлять данные об активности, которые позволяют более эффективно вычленять биты, и в итоге рассматривать уже несколько белков и несколько сайтов за раз (рис. 2.25). Это важно, так как специфичность действия лекарств зачастую тоже в приоритете, потому что если лекарство низкоспецифичное, у него будет много побочных эффектов. Это имеет отношение к высокопроизводительным схемам, когда мы можем прокапать одно вещество на несколько точек, чтобы получить сразу много данных об его активности в разных условиях. И дальше с помощью FP можно попытаться натренировать модель.

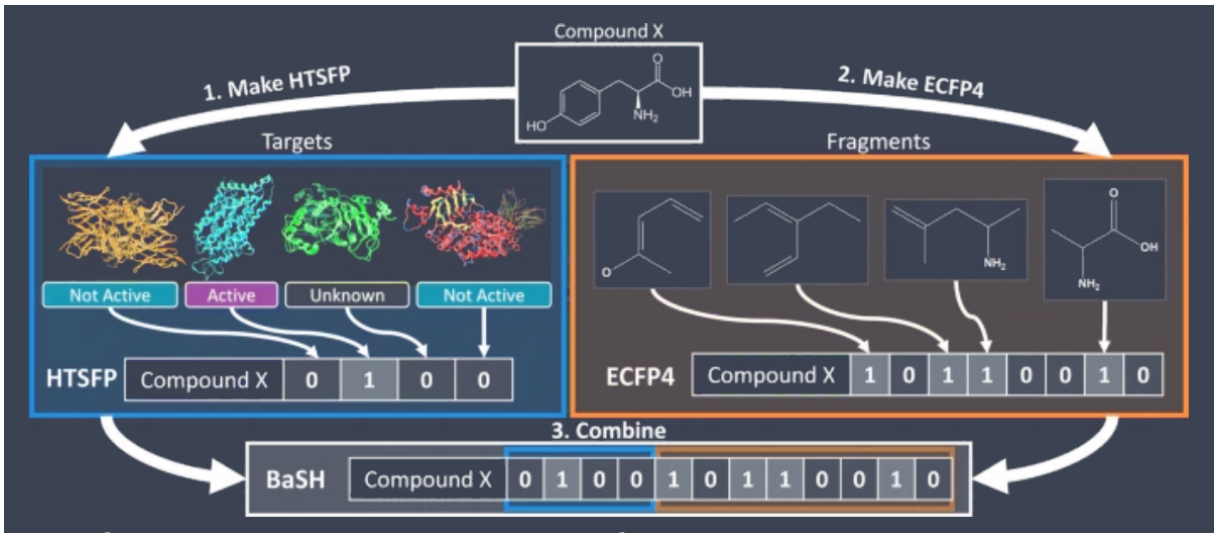

Рис. 2.25. Bioassay

Генеративные подходы

Переходим к **генеративным подходам.** Очевидно, что если у нас есть дескрипторы, мы можем пробежать все базы данных. Вообще, почти все базы данных уже исследованы при хороших дескрипторах. Остаётся исследовать то, чего нет в базах данных, а это всё химическое многообразие.

Есть разные способы, чтобы это делать. Один из них – делать с помощью SMILES какие-то построения, SMILES описывать в виде матриц и дальше использовать агентскую сеть. Есть некий **агент**, который вносит мутацию в дескриптор (рис. 2.26). Внесение этой мутации считается хорошим или плохим неким внешним наблюдателем. Это крутится на обучающей выборке до тех пор, пока у нас это не станет действовать хорошо. Агент учится отличать хорошее от плохого и генерирует всё новые и новые вещества.

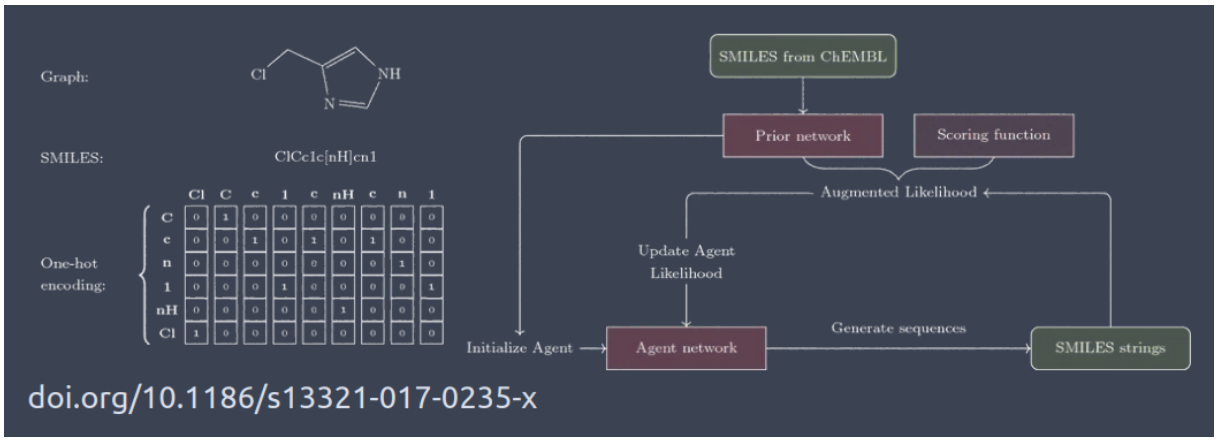

Рис. 2.26. Агент

То есть генерируются такие матрицы, в них меняются вероятности и, как результат, на выходе получаются разные соединения (рис. 2.27). Это называется **генерацией по подобию**, потому что мы хотим сгенерировать соединение, не очень сильно отличающееся от того, что было в обучающей выборки, внесением в него мутаций. Можно создавать подобное с точки зрения 2D, 3D и 4D FP, то есть в качестве подобия добавляем дополнительное описание взаимодействия с белком. Тогда на выходе получаются вещества, которые тоже взаимодействуют с этим белком. Если мы возьмём много веществ, которые взаимодействуют с данным рецептором по всем возможным атомам в белке, например, из докинга, можно сгенерировать вещества, которые одновременно реализуют все взаимодействия, которые встречались в докинге.

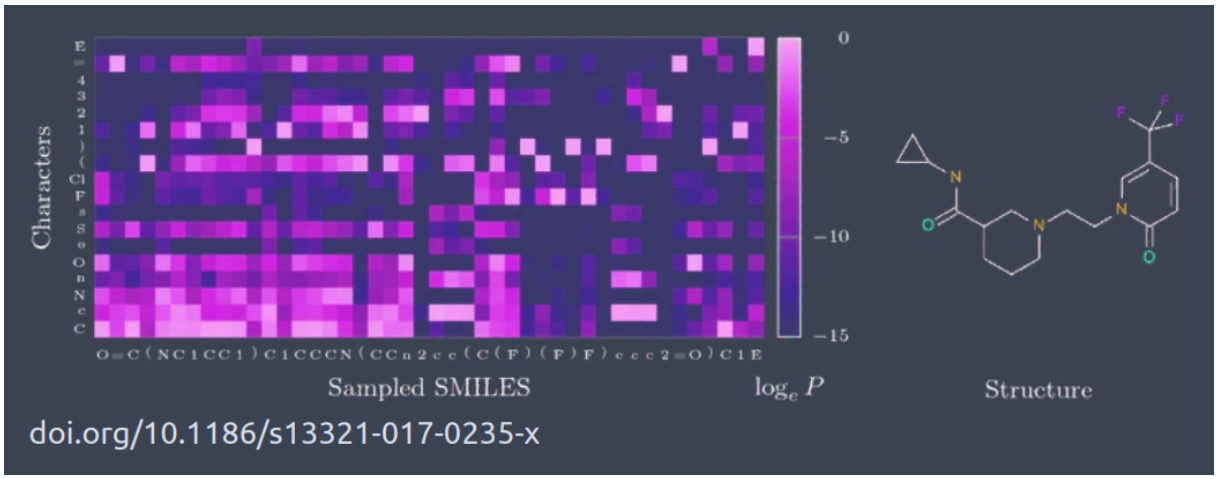

Рис. 2.27. Генерация по подобию

То же самое через **латентное пространство** (рис. 2.28). Можно обучить энкодер, чтобы получить такое пространство, из него обратно делать декод. Тут можно вводить дополнительные манипуляции, которые приводят к тому, что у нас получаются мутированные молекулы. Таким образом можно получать новые молекулы.

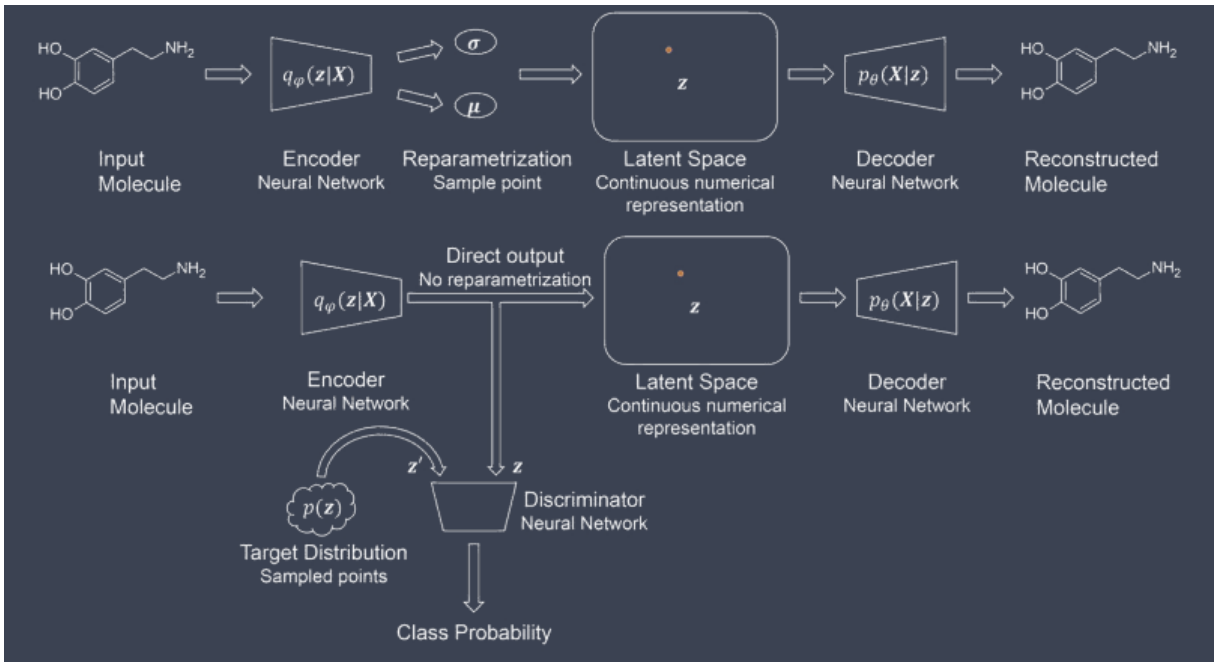

Рис. 2.28. Generative Autoencoder

Предварительно обученный автоэнкодер используется для сопоставления молекулярной структуры с латентным вектором. GAN обучается с использованием латентных векторов в качестве входных и выходных данных. После этого в данном пространстве можно смещаться и из этого получать молекулы, которые не являются напрямую входными для построения данной сети.

Это может выглядеть как на рис. 2.29. У нас есть дискриминатор, генератор и автоэнкодер.

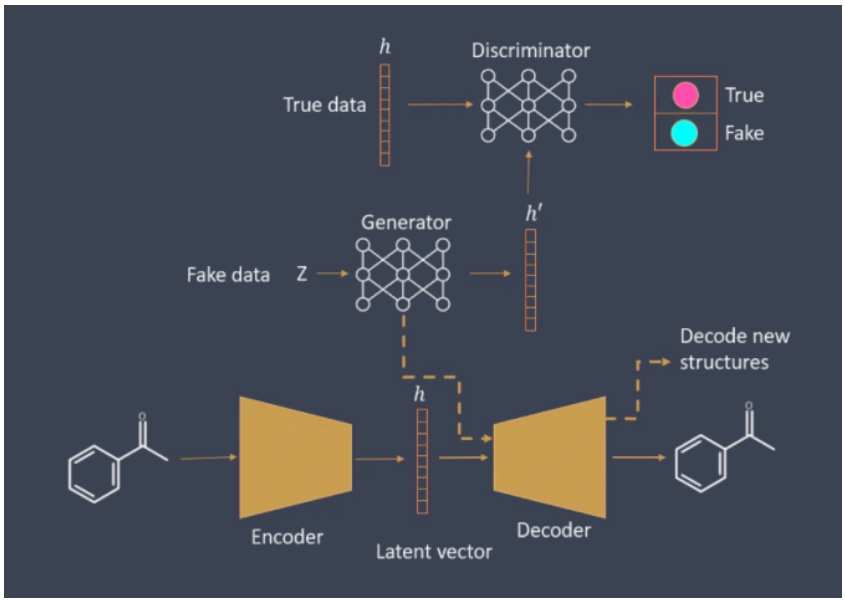

Рис. 2.29. AE и GAN

Автоэнкодер автоматически кодирует данные в какой-то вектор (пространство) и автоматически может из него декодировать данные без потери качества. Дальше можно сделать генератор, который генерирует некие фейковые значения в этом пространстве. Этот генератор надо предварительно подвергнуть дискриминатору, чтобы понимать, хорошее или плохое он генерирует. После этого мы декодируем хорошее из этого пространства и получаем новые молекулы.

Генератор может работать просто из рандомайзера, который генерирует вектор в латентном пространстве. Но предварительно мы дискримиринируем этот вектор, чтобы проверить, попадает ли он в это пространство. Если попадает, то декодируем его.

Классические методы делали регрессию и с помощью неё исследовали современные или сгенерированные базы данных. А с помощью машинного обучения мы просто генерируем новые молекулы и дальше уже определяем, хорошие они или плохие, фильтруем и проверяем.

Автоэнкодер – некий алгоритм на основе нейронных сетей, который позволяет из структуры данных получить описание в гладком пространстве функций. Функции – это латентное пространство. Если всё правильно сделано, можно, тыкнув в это пространство, получить ту же молекулу назад, то есть при данном x получаем эту молекулу: одна точка соответствует одной молекуле. Дальше задача двигать x и получать новые молекулы.

Frameworks

Какие есть фреймворки, с помощью которых это можно делать? Например, **DeepChem**: туда сейчас что-то добавляется, но развивается не активно.

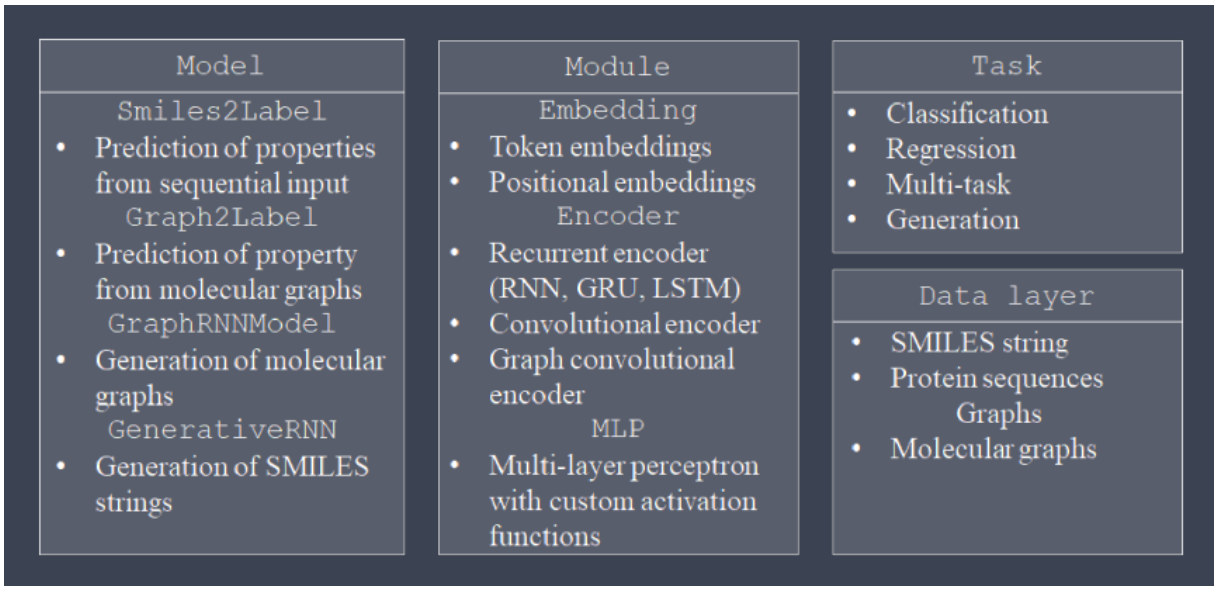

Рис. 2.30. OpenChem

Развитие методов нейронных сетей и др. идёт очень быстро, и быстро адаптировать методы хемоинформатики под новые методы не всегда получается, поэтому чаще люди не чинят старые фреймворки, а делают новые. Один из таких примеров – **OpenChem** из лаборатории Исаева (рис. 2.30).

Любой фреймворк имеет ограничения, которые заложил в него автор.

Основные **направления для работы** – пытаться делать правильные выборки. Обычно features у нас количественные, но, если нет, надо пытаться превращать их в гладкие функции, чтобы мы могли недискретно перемещаться от одного значения features к другому.

Если научиться ещё лучше включать структурно-функциональные данные с использованием конформационной информации или конформации белка с чем-то ещё, это может сильно помочь в разработке поиска лигандов.

Лекция 3. Межмолекулярные взаимодействия белок-лиганд

Сегодня мы поговорим о том, как в явном виде учитывать тот факт, что белок может взаимодействовать с соединения ми, и как на основе этого улучшить предсказания о том, как те или иные вещества могли бы взаимодействовать с белками.

Докинг

Межмолекулярные взаимодействия формируют материю. Всё, что мы видим – атомы, молекулы, фазы, которые имеют свойства. Если бы всё состояло из ковалентных связей, это было бы сложно. А так большинство того, что нас окружает, состоит из нековалентных связей, их описание и взаимодействие важно для понимания устройства мира. Живая материя тоже опирается на межмолекулярные взаимодействия.

Очень редко бывает, что молекулы из живого претерпевают химические изменения, то есть образуют ковалентные связи друг с другом. Обычно просто белки катализируют превращение низкомолекулярных агентов в более высокомолекулярные: транскрипция, трансляция, репликация. Эти три процесса являются ключевыми для формирования структуры биополимеров, и при этом они почти единственные, которые это делают.

Нековалентные взаимодействия основаны на слабых силах. Наше понимание об энергетике этих сил, о том, из чего они состоят, до сих пор далеко не полное.

Многие белки используют малые молекулы для выполнения своих функций. Например, они эти молекулы связывают и что-то используют в их гидролизе или в синтезе, либо это активаторы. Часто это является лекарственными средствами.

В чём разница между взаимодействиями белок-белок и **белок-лиганд**? Под лигандом надо понимать небольшую органическую молекулу. Белки большие, у них поверхности тоже большие, соответственно, поверхность взаимодействия большая, то есть энергетический выигрыш гидрофобного эффекта будет значительный. С водой лучше всего взаимодействует вода. Всё, что не является водой, находится под действием гидрофобного эффекта и может между собой взаимодействовать. Поэтому низкомолекулярные агенты, в отличие от белков, должны взаимодействовать очень эффективно.

Есть один нюанс. Конформационная динамика малых соединений гораздо менее выражена, чем конформационная динамика белков. Она менее сложна и зачастую более дискретна. То есть есть конформеры, на которых находятся малые соединения, и эти конформеры могут участвовать во взаимодействиях с белком. Любое подобное взаимодействие становится наиболее оптимальным, когда поверхность лиганда комплементарна поверхности белка. Это важно, потому что, если даже на поверхности нет явно хороших энергетических взаимодействий, комплементарность поверхности уже даёт факт того, что лиганд может хорошо взаимодействовать с белком просто за счёт гидрофобики.

Самый распространённый метод поиска способа взаимодействия небольшой молекулы с белком – это докинг. Вообще докингом называется целый класс таких методов. У нас есть структура белка, дальше мы выбираем место в белке, про которое мы хотим узнать, может или нет там связаться эта молекула, и пытаемся разместить её в этом месте.

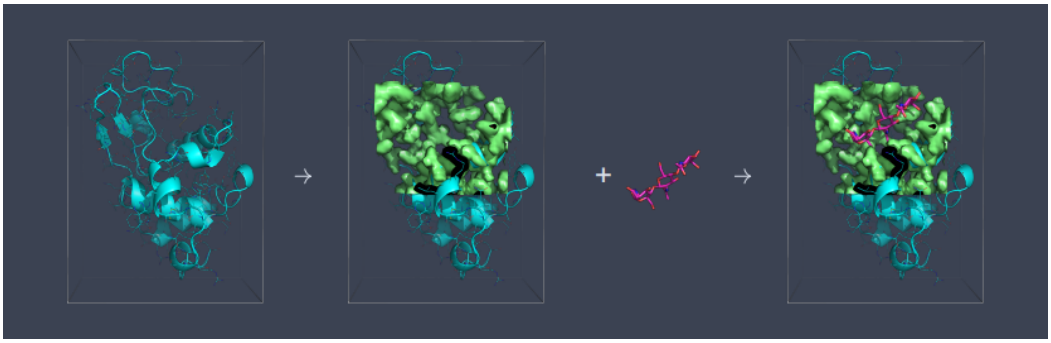

Рис. 3.1. Докинг белок-лиганд

Ключевой момент – именно в этом месте, потому что теоретически небольшая молекула всегда может найти хорошее место на любой поверхности любого белка. Поэтому молекулы могут связываться на этих поверхностях, давать неспецифические эффекты, оказывать неспецифическое действие, но, если константа в специфическом сайте сильно выше, то представленность такого связывания тоже будет гораздо выше.

Представленность состояния с константой связана с по закону $e^{-\frac{\Delta E}{kT}}$.

Чем больше ΔE, тем больше разница в представленности состояний. Разница в 3-4 килокалории означает, что одного состояния примерно в 1000 больше, чем другого. Поэтому хорошая энергия связывания важна. Если окажется, что на поверхности лиганда есть водородная связь, которая хорошо экранирована от воды, она будет давать хороший вклад в энергию, и это, возможно, будет фактом специфического связывания.

Сайт связывания – это конкретное место связывания лиганда, потому что их может быть несколько. Геометрия связывания – совокупность места связывания, ориентации и конформации лиганда.

Отсюда можно понять, что докинг как метод должен решать несколько задач. Во-первых, передвигать лиганд в пространстве. Во-вторых, менять на лету свою геометрию, потому что у него может быть несколько конформеров, и мало ли какой из этих конформеров удобен в данном месте пространства.

Как представить для расчётов место пространства, место связывания лиганда в белке? В основном структуры белков мы получаем только одним методом – методом рентгено-структурного анализа (рис. 3.2). Он нам даёт факт расположения атомов в определённом месте, но не даёт учёта динамики подвижности белков.

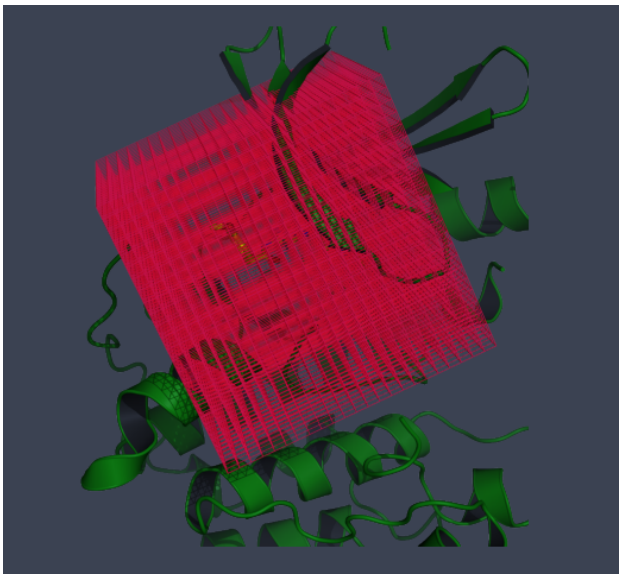

Рис. 3.2. Представление места связывания

Таким образом, мы можем представить белок как некую решётку, и в каждом узле решётке если окажется атом лиганда со специфическими свойствами определённого типа, тогда мы можем ему давать плюс или минус score функцию. И эти значения (плюс или минус) зависят от того, какие атомы белка находятся рядом с этим узлом решётки. То есть, один раз сделав решётку на основе неподвижной структуры белка, мы можем по этой решётке прогонять много соединений и проверять, как они будут связываться, находить оптимальное расположение и оптимальную конформацию в рамках этой решётки.

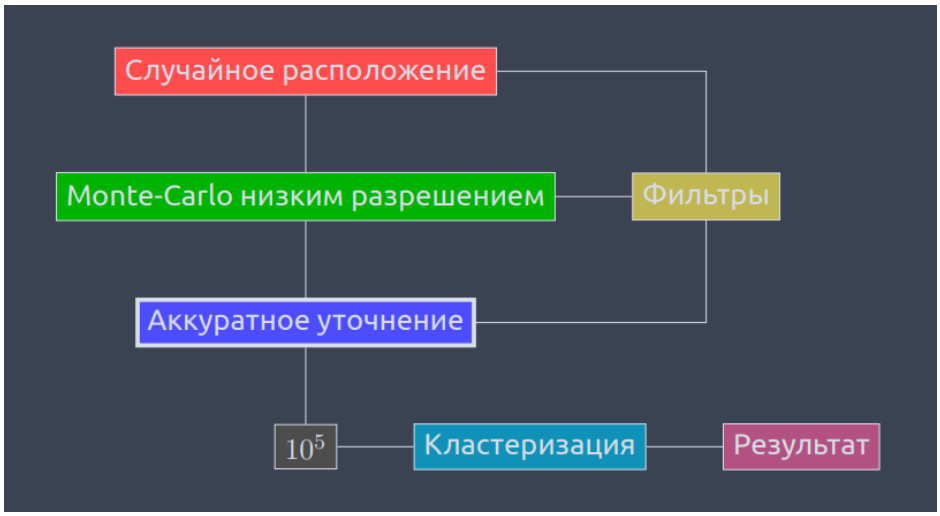

Рис. 3.3. Алгоритм поиска позиции и конформации лиганда

Дальше простой **алгоритм** (рис. 3.3). Мы генерируем случайное положение, дальше методом Монте-Карло начинаем искать в этой решётке оптимальные **конформацию и положение лиганда**. Дальше с помощью метода оптимизации геометрии улучшаем лучшие позы. У нас есть фильтры: допустим, мы запустили Монте-

Карло много раз, и 90% результатов дали одно и то же. Тогда можно взять из этих 90% результат один раз. После этого у нас достаточно существенно для лиганда изменяются координаты, после чего, накопив уже большое количество данных, можно их кластеризовать и выдать результат.

Здесь есть один практический момент. Если мы делаем кластеризацию, то весьма вероятно, что ответов у нас может быть очень мало. Например, мы получили 10^5 состояний лиганда в белка, и все они попадают в 2 или 3 кластера. Поэтому нередко ответ из молекулярного докинга состоит в двух или трёх конформациях лиганда, которые получаются в результате моделирования его положения в белке. И когда мы пытаемся заставить программу выдать ещё больше результатов, это может быть просто невозможно, так как остальные результаты попадают в те же самые кластеры.

Ещё есть **генетические алгоритмы**. Они более часто используются для докинга. У нас есть фитнес-функция, некие гены, манипуляцией которыми мы добиваемся того, чтобы фитнес-функция была максимальной. Это быстрее, чем вычисление просто в сканировании по стандартному Монте-Карло. В докинге генами будут положения, значения торсионных углов в лиганде. Можно вносить смертность особей, чтобы мы могли переваливать через локальные минимумы параметров.

Посмотрим на типичный результат подобного расчёта (рис. 3.4). Слева три кластера положения нуклеотида на поверхности белка, а дальше показано, что будет, если получить как можно больше кластеров. Видно, что в этом случае нуклеотид может быть расположен любым способом на поверхности белка. Но и из этого можно получить информацию, плотность каких атомов на поверхности предпочтительна со стороны подобных лигандов. Это можно использовать как нечёткий способ описания молекулы, которая могла бы в идеале с этим связываться. Это называется пробинг поверхности с помощью лиганда.

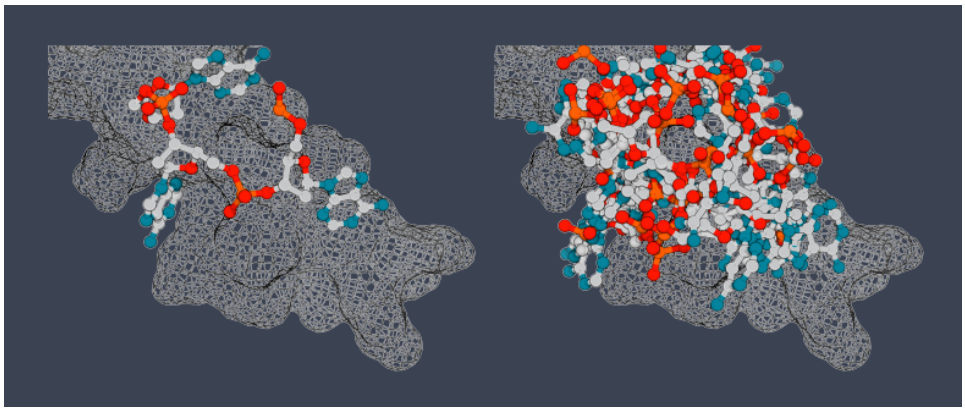

Рис. 3.4. Представление места связывания

То есть докинг – не обязательно способ нахождения одного конкретного лиганда на поверхности белка. С помощью него можно провести пробинг поверхности белка на то, какие типы атомов предпочтительно в каких областях этой поверхности связываются.

Исходя из этих плотностей, можно построить совершенно новую молекулу, которая наилучшим способом вписывается в этим плотности.

Основные **цели докинга** – виртуальный поиск лигандов и определение геометрии связывания лиганда. Каким должен быть докинг для виртуального поиска? Он должен быть очень быстрым, потому что мы берём базу на миллионы молекул и прогоняем всё через него. Если база содержит больше двух-трёх миллионов молекул, начинаются проблемы с файловыми системами, и нужно оптимизировать расчёты. Здесь мы просто пытается отобрать, что связывается лучше, что хуже, зачастую 99,9% вообще не связывается, и быстро получить положения.

Вторая цель докинга – определение положения конкретного лиганда. Если мы знаем, что этот лиганд с белком взаимодействует, нужно знать, где он это делает и как. Здесь докинг не должен быть быстрым, он должен быть точным. Нужно добиваться того, чтобы конкретный лиганд связывался максимально правильно с белком. То есть надо исследовать все положения и выбрать самое лучшее положение.

Здесь могут быть подводные камни. Например, можно неправильно задать сайт, куда хотим его связать, или он недостаточно большой, чтобы правильно расположить лиганд и т. д.

Если мы знаем, как связывается лиганд, то при наличии больших баз данных мы можем узнать, какие части важны для связывания. Но, если основа, к которой присоединены взаимодействующие группы, меняется, может сильно меняться геометрия взаимодействующих групп.

Также можно предложить изменения для улучшения константы связывания. Например, если у нас много статистики по соединениям, можно из набора соединений, которые хорошо связываются с этим белком, сделать химерное соединение, которое связывается ещё лучше.

В докинге есть чёткое разделение на **два основных этапа**. Первый этап – алгоритм поиска. Это установление места связывания, то есть просто перемещение по сетке, и установление геометрии связывания. А второй этап – расчёт энергий. Энергия может считаться очень дорого. Мы её считаем для ограниченного количества молекул и получаем ограниченное количество результатов. Тут можно использовать очень сложные способы подсчёта этой энергии для того, чтобы они наибольшим образом соответствовали экспериментальным результатам.

Большинство докинг-программ современного поколения дают правильные ответы о положении лиганда в белке, но очень большие ошибки в оценке их энергии связывания, потому что энергии связывания основываются на функциях типа силовых полей. Это парные взаимодействия, а чем больше молекула, тем больше в ней парных взаимодействий. Соответственно, энергия взаимодействия большой молекулы будет лучше, чем энергия взаимодействия малой. Но в жизни это может быть не так, потому что чтобы большая молекула начала взаимодействовать с белком, её нужно

десольватировать, и на это тоже идёт большое количество энергии. Некоторые score функции пытаются учитывать энергию десольватации молекул, но это делается очень неточно.

Существует много **программ для докинга:** AutoDock, DOCK, e-Hits, Flex, FRED, Glade, GOLD, LigandFit, QXP, Surflex-Dock и т. д. В последнее время все новые разработки в области программ по докингу связаны с улучшением score функций и скорости счёта. Существуют разные алгоритмы оценки аффинности и разные алгоритмы поиска.

Важно не путать лиганд-белковый докинг и белок-белоковый докинг. Последний так работать не будет, потому что мы вряд ли сможем эффективно подсчитать подвижность белка при взаимодействии с другим белком. Монте-Карло не учитывает гидрофобный эффект, потому что там нет температуры как таковой, нет в явном виде растворителя, и цепочка начинает сыпаться. Однако вообще программы по белок-белковому докингу есть.

Если будем готовить структуры и молекулы для докинга, надо учесть **практические аспекты.** Зачастую мы делаем докинг в структуру белка, у которой нет лигандов. Поэтому, если они там есть, их надо оттуда убирать. Это можно делать как автоматизированным способом, так и вручную.

Немаловажная особенность состоит в том, что некоторые программы докинга учитывают в явном виде положение протонов в белке. Протоны – это источник водородных связей. Например, гистидине положение протона может формально зависеть от того, какой лиганд туда добавили. Если у лиганда акцептор водородной связи, протон может встать близко к лиганду, а если донор, то водород встанет в другое положение гистидина.

Мы рассматриваем белок как жёсткую матрицу. Поэтому важно положение групп таких аминокислот как, допустим, аспарагин и глутамин. Там у некоторых групп одинаковая электронная плотность, и рентгеноструктурный анализ может дать неточное определение их ориентации, что сказывается на паттерне водородных связей внутри белка. Хорошего автоматического решения для этого нет.

Если исследуем связывание одного конкретного лиганда в белке, то можно вручную перебрать все комбинации положения аспарагинов, глутаминов, гистидинов и т. д. и все таутомерные формы лигандов, и из этого сделать заключение, какая из этих вариаций лучше связывается. Однако если мы смотрим за миллионами лигандов, на все эти вопросы мы не сможем ответить, но есть шанс исправить это на основе машинного обучения.

Докинг может быть двух типов: докинг жёсткого тела и подвижного лиганда. Rigid: лиганд не имеет внутренних степеней свободы, т.е. вращение вокруг связей запрещено. Flexible: предполагает учёт вращения вокруг связей лиганда.

Зачем нам делать **докинг лиганда как жёсткого тела**, если мы можем учесть его подвижность? Нередко программа расчёта конформации лиганда может быть неточной. Мы можем определить очень точно все конформации лиганда методом квантовохимического расчёта, а потом эти жёсткие конформации сделать докинг белок. То есть из-за того, что лиганд не движется, мы просто делаем его движение как жёсткого тела внутри сетки потенциалов, которую нам даёт белок. Скорость от этого сильно не падает, а может даже быть быстрее, чем менять на лету конформацию лиганда. И это может быть гораздо точнее.

Однако лиганд может иметь такую конформацию в комплексе с белком, которая ему не выгодна. Это бывает, если энергия связывания лиганда с белком такова, что ΔG перебивает потерю энергии от лиганда, находящегося в невыгодном состоянии.

Часто белок рассматривается как не очень жёсткое тело. То есть можно сказать, что внутри белка есть аминокислоты, которые имеют конформеры, способные менять конформацию от взаимодействия с лигандом. Они как бы являются дополнительными лигандами в процедуре докинга, лиганд становится длиннее на подвижные аминокислоты. Это будет удлинять расчёт в n раз в зависимости от того, сколько конформеров, т. к. под каждую их комбинацию мы будем проводить докинг. Если вещество одно, это не страшно, но, если их много, всё перестаёт работать.

Фрагментарное построение лиганда

Если есть сайт, почему бы не построить лиганд заново: придумать такое положение групп в пространстве, которое бы хорошо связывалось с этим сайтом. Сканировать просто, так как мы точно знаем, что такие молекулы существуют. С другой стороны, de novo тоже можно сделать много разных молекул, которые потенциально могут иметь смысл в синтезе. Логически кажется, что проще делать новые молекулы, а потом отсеивать те, которые не можем синтезировать.

Задача состоит в том, чтобы узнать, как фрагменты хорошо связываются в белке. Докинг небольших фрагментов можно сделать экспериментально или теоретически. А дальше их надо связать ковалентными связями.

Допустим, каждый из трёх лигандов связывается с константой 10^{-3}. Если соединить все три лиганда, с какой константой будет связываться финальное вещество? Когда объединяем плохо связывающиеся объекты в одну ковалентную структуру, константы должны перемножаться и становиться наномолярными, а то и фемтомолярными.

Возможно несколько подходов для **фрагментарного построения лиганда** (рис. 3.5). Первый – распихать по карманам заместители и потом пытаться их соединить. Второй – поставить в центр кармана некий скелет и на него начать наращивать заместители до тех пор, пока не будет покрыто всё пространство взаимодействий с ожидаемыми группами, которые там есть.

Рис. 3.5. Подходы для фрагментарного построения лиганда

Есть аналог докинга, который позволяет всё это делать, – GRID. Есть питоновская надстройка над автодоком – Autogrow, которая позволяет делать в автоматизированном виде докинг растущих лигандов, чтобы двигаться в сторону улучшения энергии связывания.

Есть подход MCSS: сайт наполняется фрагментами и с помощью ЕМ вычленяется место, где фрагмент наиболее предпочтителен. После сортировки из лучших делаются соединения. Взаимодействия между фрагментами не учитываются. LUDI использует информацию из банка PDB для задания фрагментов, образующих водородные связи и т. д., оптимизирует их положение.

Не обязательно всё моделировать для определения связывания фрагмента, можно использовать РСА и ЯМР. Берём кристалл белка, наливаем в него много небольших соединений, делаем рентгеноструктурный анализ и узнаём, что и где связалось.

А дальше идёт **реализация ковалентного связывания фрагментов**. Если есть два и более фрагмента, то можно искать способ их соединения по базам данных. Это работает, если мы знаем, как в пространстве должны быть расположены фрагменты,

которые связываем. В базе соединений должны быть конформеры – 3D структуры, чтобы можно было чётко сказать, что положение данных фрагментов отвечает такому же положению данных фрагментов в определённом соединении. Это реализовано в CAVEAT.

Можно автоматически строить скелеты, но здесь есть много нюансов, потому что надо сделать, например, чтобы конформация скелета была выгодна, учесть гидрофобные эффекты и т. д. Главный критерий – это сохранение взаимного положения фрагментов. Переход от скелета к молекуле сложен, так как надо реализовать возможность синтеза молекулы.

Одно из реально существующих лекарств было получено с помощью такого подхода (рис. 3.6).

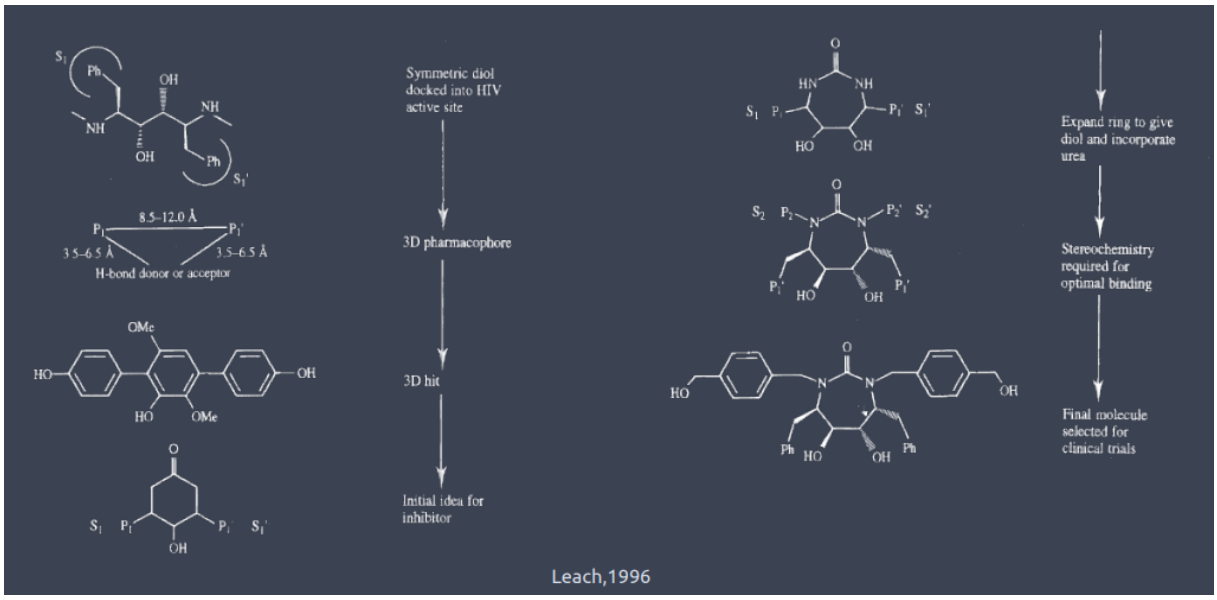

Рис. 3.6. Поиск лекарства с помощью построения скелета

Методы ML для проблемы докинга

Рассмотрим основные **направления применения ML** в этой области. Score функции в последнее время становятся всё более и более эмпирическими и менее понятными, потому что мы зачастую не можем предугадать все возможные взаимодействия между лигандом и белком.

Будем фиттить данные в модель. Это хорошо умеет делать машинное обучение. Для этого подходят, например, нейронные сети. Докинг здесь – вычислительно затратная процедура. Если мы сможем из всего химического многообразия сделать несколько миллионов соединений для докинга, это будет важным этапом, потому что так мы сильно сокращаем вычислительный ресурс, необходимый для поиска ингибиторов.

Ещё важен поиск сайтов связывания. Мы выбираем сайт связывания и делаем из него докинг. Правильно ли мы его выбрали, никому не известно. Место связывания внутри GRID, а где он находится, мы определяем сами. То есть мы указываем область

связывания, и с этим можно ошибиться. Значит, хорошего способа определить, куда могут связываться лиганды в белке, у нас нет, и машинное обучение в этом помогает. Очевидно, что у природы был какой-то эволюционный отбор по белкам, и количество органических молекул, которые принимают участие в функционировании живого, ограничено, поэтому все сайты связывания в белке тоже имеют ограниченную вариативность.

Ещё один сильный момент со стороны машинного обучения – генеративные возможности. Мы можем на основе неких требований генерировать новые и новые молекулы с высокой аффинностью. Скорость этого метода гораздо больше, чем любого другого.

Теперь рассмотрим **перевзвешивание**. Нахождение геометрии лиганда считается быстро, а сортировка лигандов по их активности внутри белка – непредсказуемая и не точная. Классические функции, которые мы используем для скоринга в докинге, – парные взаимодействия в силовом поле. Но сумма парных взаимодействий может и не давать общую энергию взаимодействия по причине того, что та же гидрофобика не строится на парных взаимодействиях.

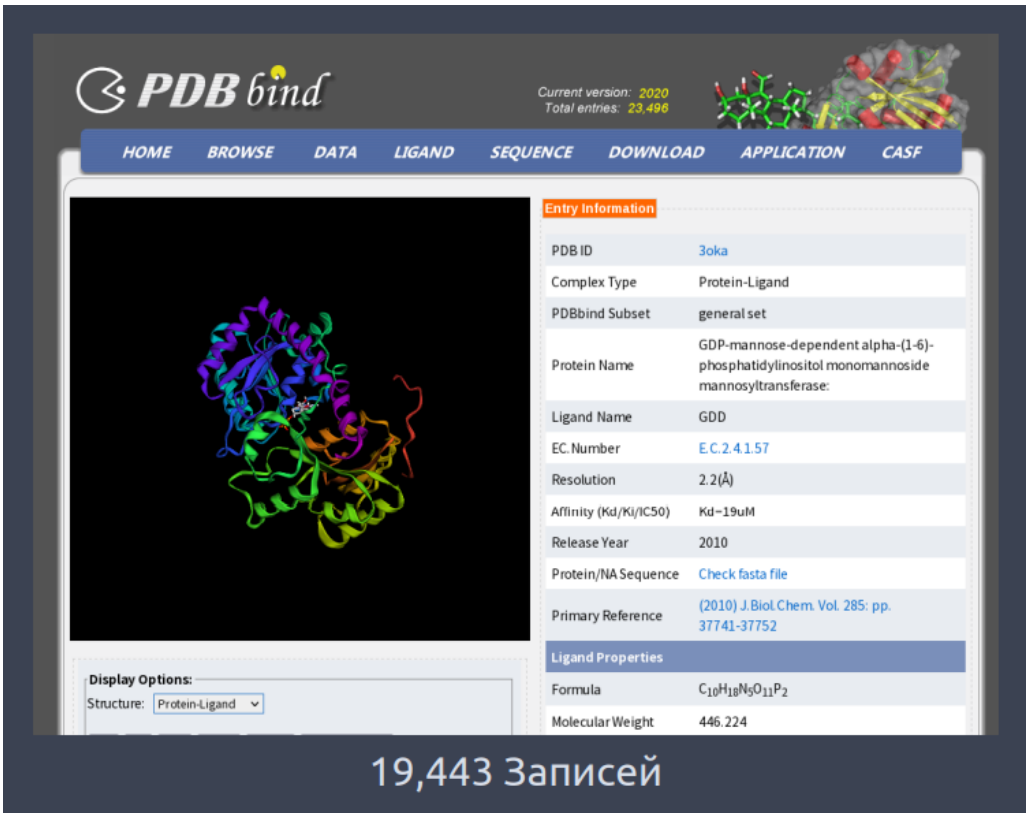

Рис. 3.7. PDBbind

Откуда нам брать **данные** для этого? Самый распространённый объект – база **PDBbind**, в которой есть тип комплекса, к какому data set он относится, идентификатор белка и описание лиганда (рис. 3.7). Самое важное в этой базе – данные о структуре, о

химическом строении и экспериментальные данные о константе связывания. Но здесь всего 19443 записи, что на фоне миллионов соединений очень мало. Но структурных данных в принципе мало.

Представления каждый придумывает сам, и они бывают достаточно сложными. В статье, о которой ниже, CNN исходно оптимизированы на изображения, сайт связывания представляется как 3D сетка, эту сетку пытаемся перевести как искривление плотности, желательно в функции, потому что всегда важнее иметь функции, а не дискретные значения, чтобы иметь производные и т. д. Атомам присваиваются типы: 30 для лигандов, 16 для белков. Для каждого типа атомов можем строить подобные функции. Атомы представляются как распределение плотности с гауссианами, чтобы получилось гладкое описание плотности расположения атомов в активном сайте:

$$A(d,r) = \begin{cases} e^{-\frac{2d^2}{r^2}}, & 0 \le d < r \\ \frac{4}{e^2 r^2} d^2 - \frac{12}{e^2 r} d + \frac{9}{e^2}, & r \le d < 1{,}5r \\ 0, & d \ge 1{,}5r \end{cases} \quad (3.1)$$

Дальше начинаем учиться с помощью ко-эволюционных нейронных сетей, чтобы построить моделирование. Сама по себе программа **GNINA** работает очень просто (рис. 3.8). Это модификация алгоритма докинга плюс включение дополнительных нейросетевых потенциалов для оценки того, какое положение лиганда лучше, какое хуже. Есть некоторое количество моделирований методом Монте-Карло, из них получаем количество конформаций. В итоге делается кластеризация, которая даёт финальный ответ.

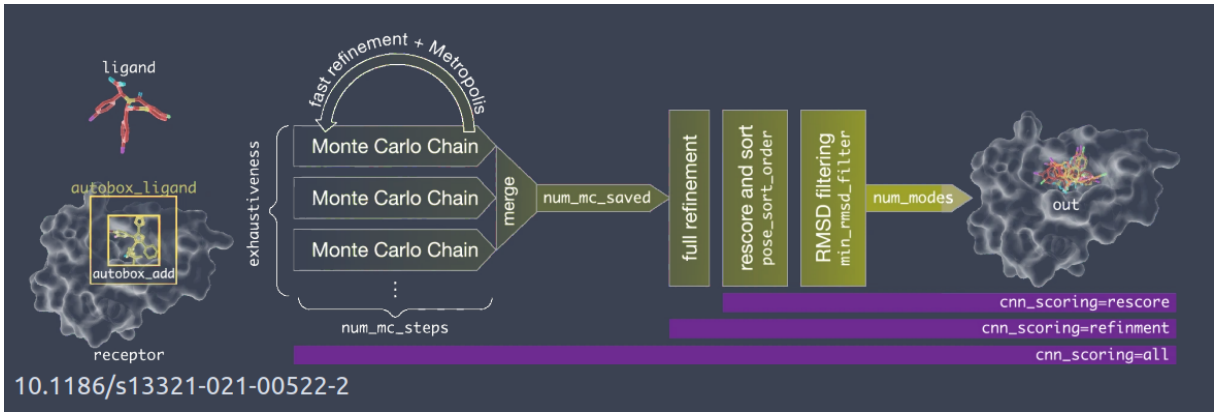

Рис. 3.8. GNINA, метод

Отличие от докинга только в скоринге с помощью потенциала, который основан на основе нейронных сетей. Этот потенциал можно постепенно обновлять (рис. 3.9). В большинстве случаев GNINA даёт гораздо более высокую точность.

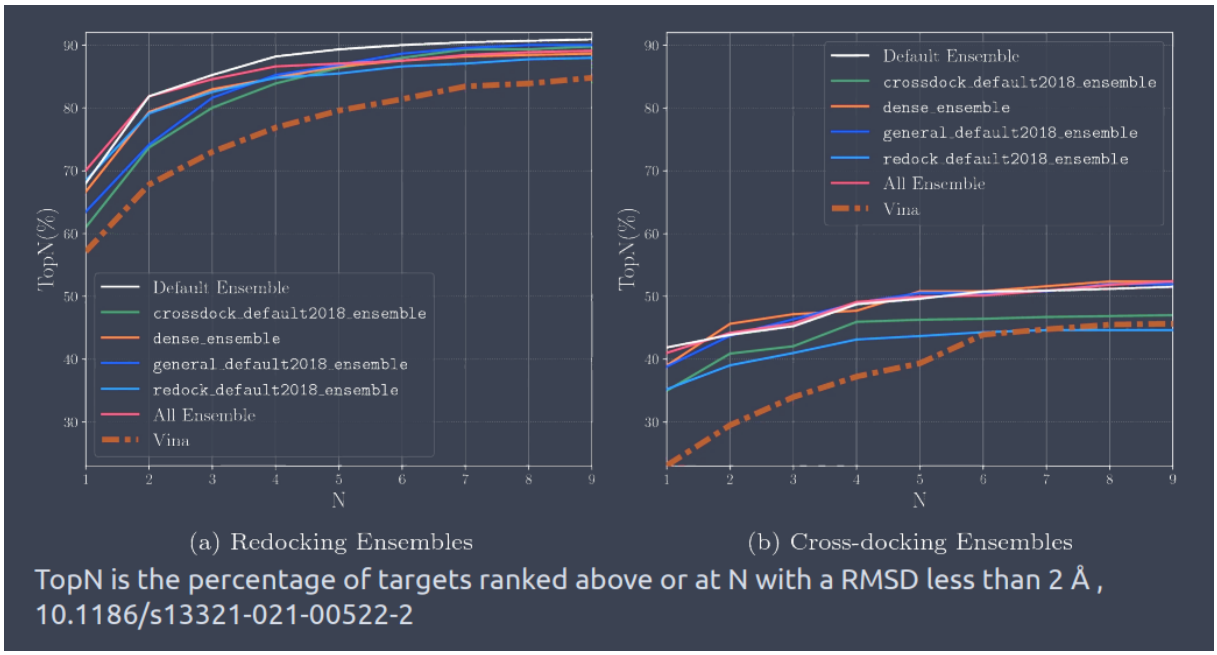

Рис. 3.9. GNINA, результаты

Мы можем использовать несколько разных нейросетевых скоринг-функций в разных пропорциях – ансамбли. Это способ расчёта score на основе нейросетевых оценок по плотности и т. д. Плотности можно обучать на разных data set, и комбинация data set может быть объединена в ансамбли. Эти ансамбли могут быть использованы для выявления хороших поз, которые есть у лиганда.

Кросс-докинг тут тоже получается не хуже. Мы пытается сделать тот же самый рескоринг, но берём заведомо неправильный объект, то есть находим положение лиганда в объекте, в котором он плохо связывается.

Рескоринговых функций существует много. Их подавляющее большинство на сегодняшний день получено машинным обучением, и они не опираются на нейронные сети, только на методы выделения и регрессии. Но это не очень интересно, потому что нам хотелось бы иметь представление связывания в 3D, а не просто пытаться находить регрессию между двумя data set.

Рассмотрим **профилирование библиотек**. Ожидаемое химическое разнообразие сравнимо с 10^{23}. Это то количество соединений, которые мы можем сгенерировать, опираясь на правила Лепински, из известных на сегодняшний день фрагментов органических соединений. Даже первичная генерация библиотеки требует гигантских ресурсов, поэтому сделать базу на 10^{23} молекул мы в ближайшее время вряд ли сможем. Поэтому есть смысл профилировать разнообразие на лету под конкретную задачу. То есть мы должны научиться генерировать молекулу, быстро проверять, насколько хорошо она подходит для наших целей, и, если подходит, использовать её дальше.

Алгоритм докинга из данной статьи приведён на рис. 3.10.

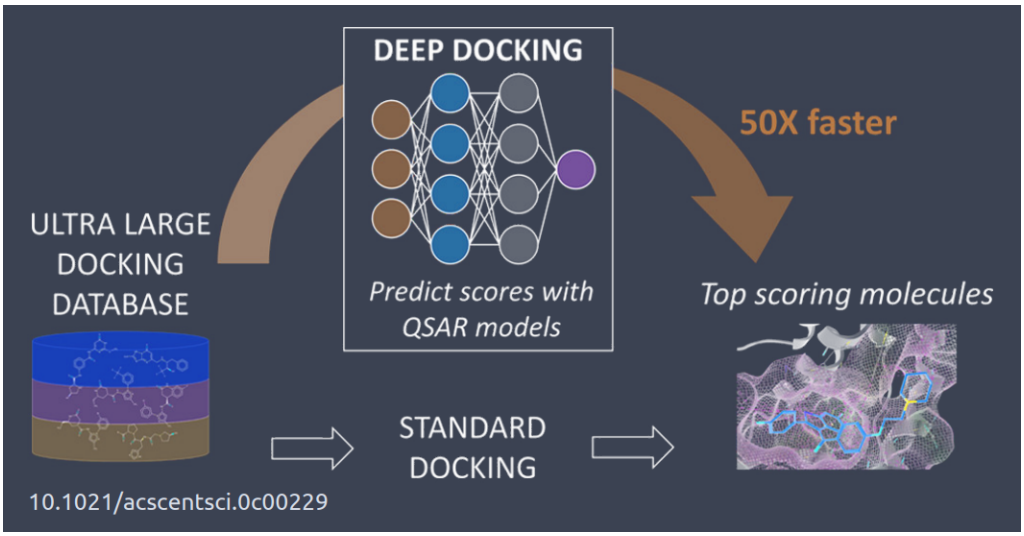

Рис. 3.10. Deep Docking, шаг 1

Подобных подходов много. У нас есть большая база данных, которая генерируется в процессе. Если мы будем предварительно отбирать из генерируемых молекул те, которые не подходят, то мы можем сильно ускорить докинг, который можем сделать на основании данного подхода.

Есть большая база данных – уже существующая или результат итеративной генерации, мы делаем из неё случайную выборку и пытаемся сделать докинг (рис. 3.11). 1. Если докинг для некоторых фрагментов этой выборки оказался успешным, то мы создаём 3 сета, создаём дескрипторы из этой базы и таким образом постоянно их обновляем на лету, и они позволяют нам уже получать соединения, которые имеют смысл.

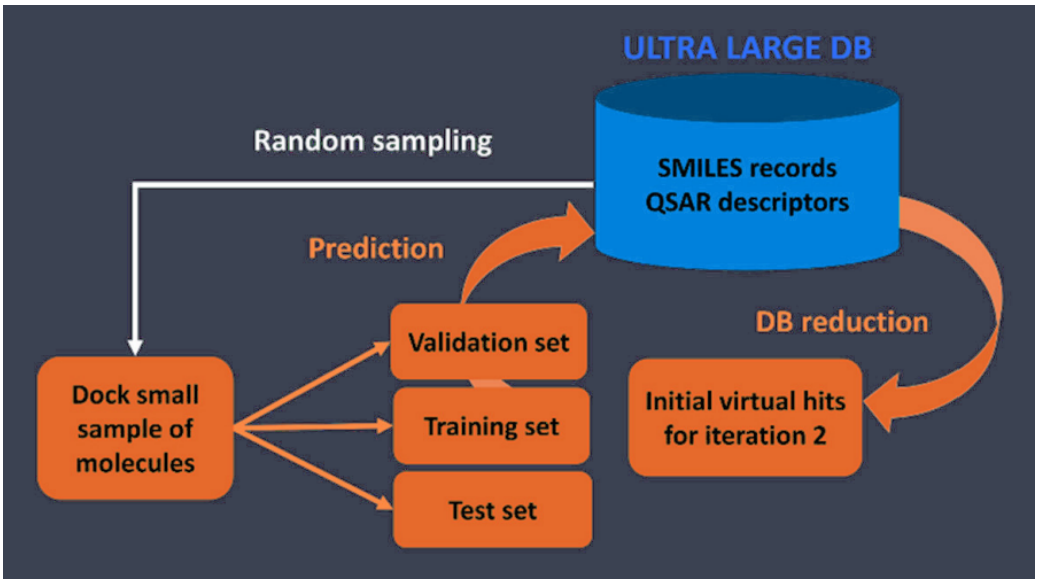

Рис. 3.11. Deep Docking, шаг 1

На прошлой лекции мы обсуждали, как из 2D описания молекул получать из описания в виде 3D фингерпринтов и др. Смотрели, чем, допустим, отличаются этанол и метанол с точки зрения химического окружения, делали описание побайтово. Корреляция между вектором, который мы получали из такого описания, и его потенциальной активность, называется курсаром (количественное взаимоотношение активность-структура). То есть мы составляем для тех молекул, которые у нас хорошо отдочились, этот байтовый вектор, и пытаемся на основании его сказать, что он работает хорошо, потому что у него докинг получился лучше, а этот плохо, потому что у него докинг получился хуже. У нас появляется курсарная модель, по сути, регрессия по пространству векторов, которое мы исследовали. Это пространство постоянно обновляется, и в результате постоянно улучшается курсар-модель.

Дальше мы можем постоянно генерировать или выбирать из этой базы вещества, перед докингом делать их оценку предполагаемого связывания, и постепенно с определённым шумом мы может получить такой курсар, который на основании этого докинга будет постоянно обновляться. Таким образом мы можем сканировать очень большие библиотеки.

Мы создали set, натренировали, дальше идём по циклу шагом два, всё больше уменьшая исходную базу данных, которую хотим отсканировать (рис. 3.12). Это позволяет сильно ускорить виртуальный скрининг.

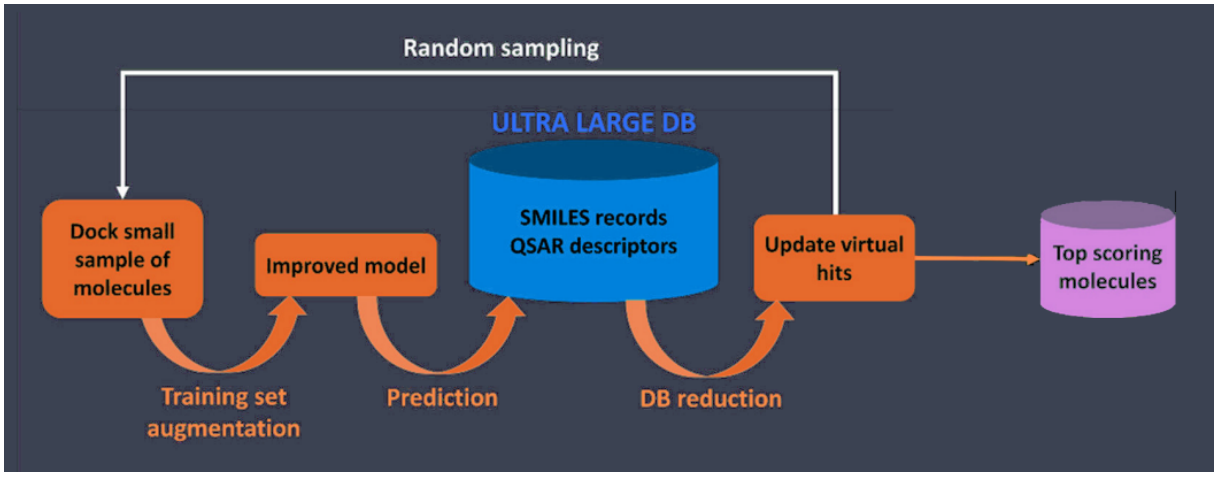

Рис. 3.12. Deep Docking, шаг 2

То есть мы улучшаем модель, из неё можем сделать новую выборку и так крутить до тех пор, пока селекция не выйдет на плато. Если стандартный курсар работает так, что есть вектор и есть ответ (два набора данных, структура и активность), и дальше просто делаем регрессию, то тут получаем активность на лету с помощью докинга и постоянно улучшаем качество регрессии. Докинг даёт улучшение качества курсара, который является фильтром из большой базы данных.

В данной статье был отбор из базы данных размером $10^3 - 10^6$. Использовались молекулярные QSAR дексрипторы, точнее Morgan fingerprints, а также "feed-forward" DNS сети в библиотеке Keras.

Перейдём к следующей задаче – **выявлению сайтов связывания**. Зачастую мы можем связать одну и ту же молекулу во многих местах поверхности, если есть значительная разница в энергии. Здесь надо научиться понимать, какой сайт является истинным, а какой неистинным. Тут появляется понятие «геодезия», – когда мы с помощью с помощью координат пытаемся описать в виде вектора поверхность белка. Ещё мы можем добавить туда типы атомов, химическую информацию, которая позволила бы сказать, что сайт действительно более адекватный. Например, если есть водородная связь, окруженная гидрофобными остатками, может ли это быть сайтом? Да, это должен быть хороший сайт, так как если есть водородная связь, изолированная от доступа воды, её энергия 5 ккал/моль, плюс ещё гидрофобика.

Для этого есть **MASIF**. Мы делаем геодезическое описание (рис. 3.13) и каждой точке навешиваем свойства: химические, гидрофобные и т. д. Покрываем весь белок такими патчами, и на основе набора этих патчей можно пытаться начать обучение.

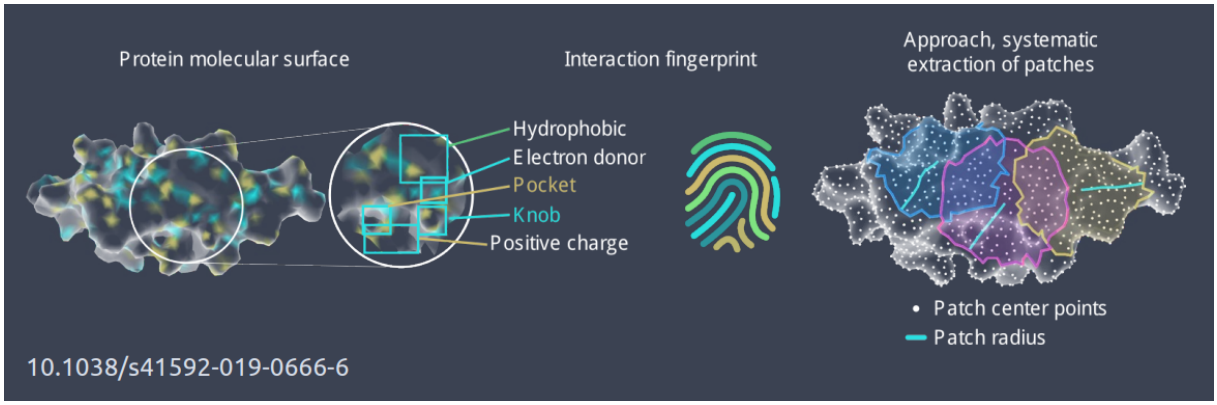

Рис. 3.13. MASIF, общая идея

Всё делается в радиальных координатах (рис. 3.14). С помощью конволюции добиваемся того, чтобы перешли к линейному дескриптору, и дальше начинается обучение.

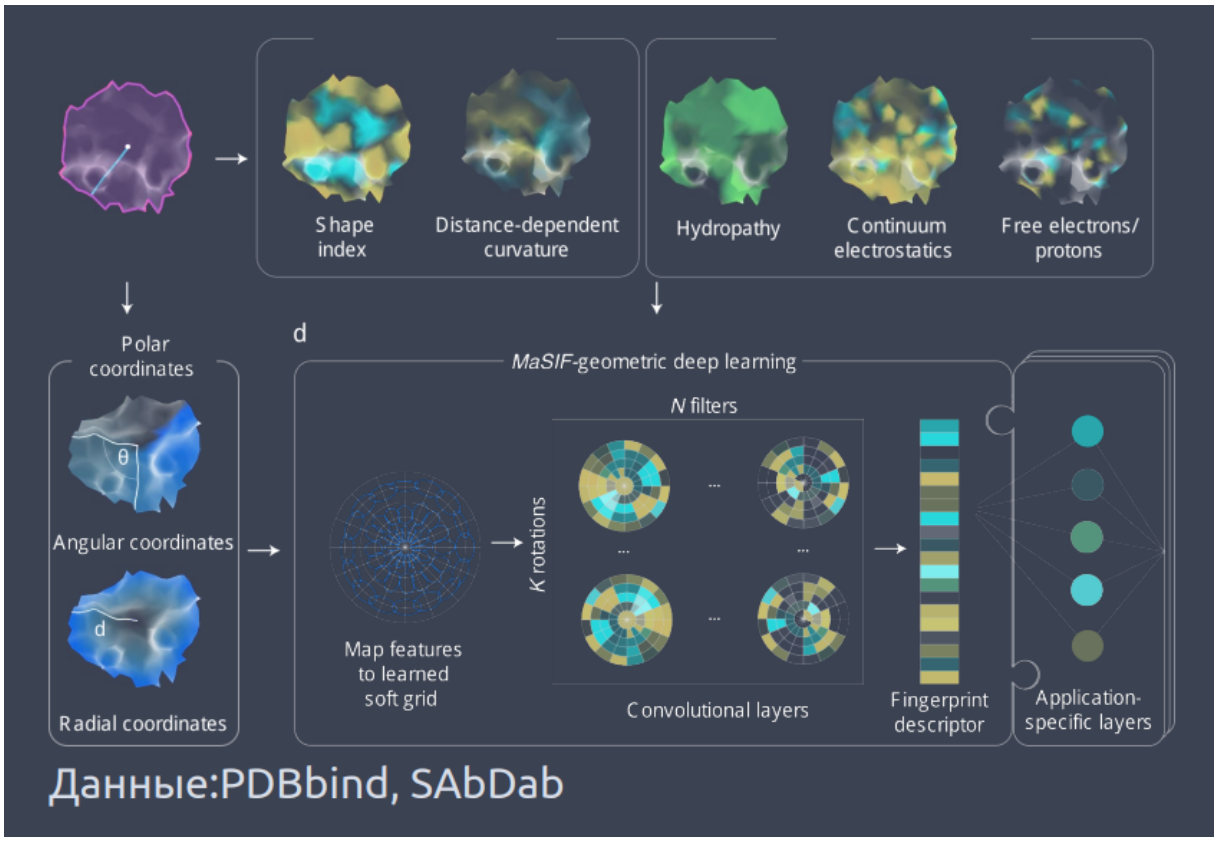

Рис. 3.14. MASIF, реализация

Важно отметить, что здесь мы используем информацию не только о том, как выглядит поверхность, но и какие атомы на ней присутствуют.

Это перспективный подход, потому что он визуализирует то, что лежит на поверхности. Если у нас есть геодезическое описание поверхности и типы атомов, мы сразу можем представить, что является комплементом к этой поверхности. Так как локальное представление поверхности в виде вектора, комплемент – обратный вектор: геодезически обратный, а с точки зрения химических взаимодействий – тоже обратный, кроме гидрофобики.

То есть, в результате мы имеем вектор, потерявших химическую информацию, поверхность, которая хороша к этому белку. А потом берём соединения, генерируем конформации для них, из этих конформаций вектора, и сравниваем с желаемым вектором.

Получилось, что среди подобных соединений (например, где много ароматики или заряда) этот метод очень хорошо определил специфичность к конкретному лиганду (рис. 3.15).

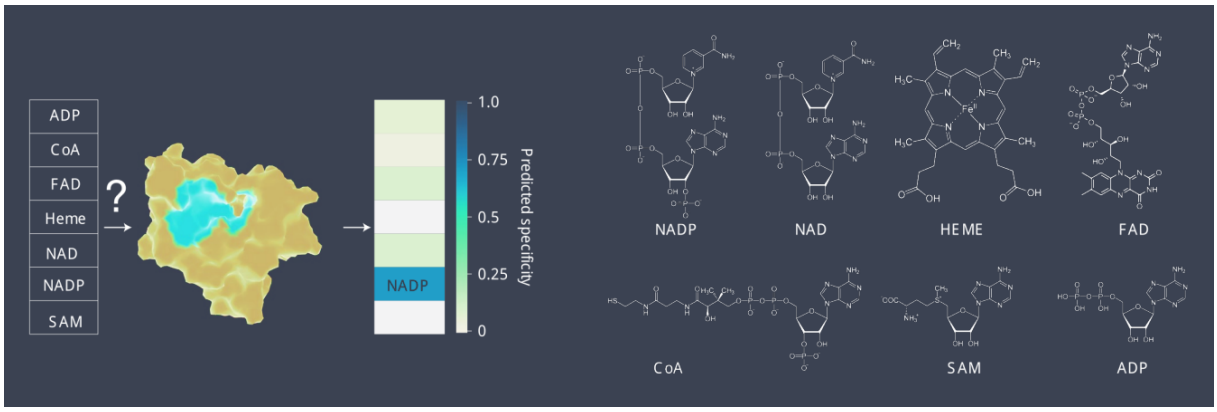

Рис. 3.15. Результат

Это хорошо, потому что нам не надо каждый раз пытаться удачно расположить лиганд в белке. Мы просто пытаемся сравнить лиганд с идеальным вектором. Для каждой конформации лиганда будет свой вектор. Это работает быстрее, чем докинг. Потенциально можно даже определить, сколько энергии нужно, чтобы загнать лиганд в этот вектор: смотреть баланс между молекулярно-механической энергией и близостью к желаемому вектору.

Здесь проверяется эффективность (рис. 3.16). Используется геодезия, геодезия плюс химические атомы. Видно, что химия зачастую даёт больший вклад в аккуратность.

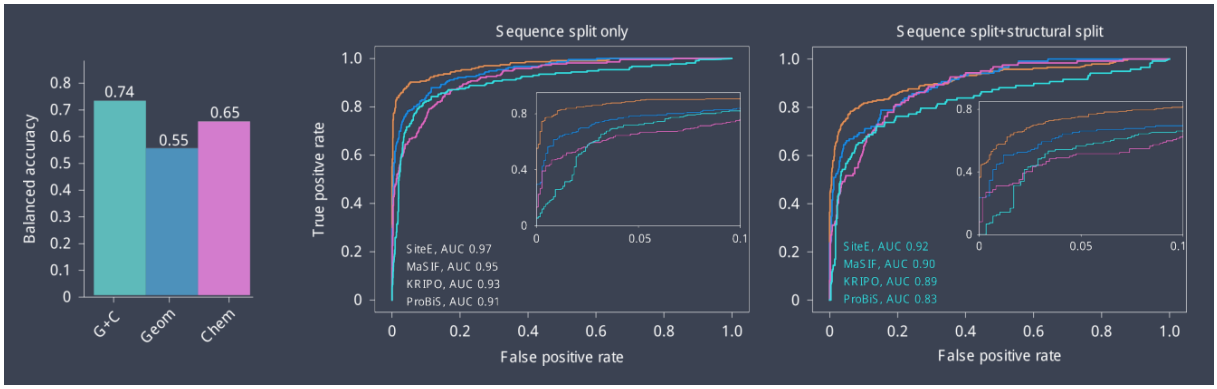

Рис. 3.16. Аккуратность

Ещё есть некий алгоритм – сайт e. Там тоже есть геодезическое представление поверхности, только они используют консенсус из поверхностей близкородственных белков, то есть вектор строится на основе нескольких структур из близкородственных белков с одинаковой активностью, например, из разных организмов. Также в этом случае можно подключить молекулярную динамику. Это гораздо более затратно, потому что проверяется один против многих векторов и ищется минимальное расстояние, но и точность гораздо больше.

Хороший пример – это кинированный НАДФ (рис. 3.17). Видим, что даже стереоспецифичность тоже хорошо реализуется. Специфичность по отношению к добавлению одной фосфатной группы приводит к тому, что это чувствительно и хорошо работает.

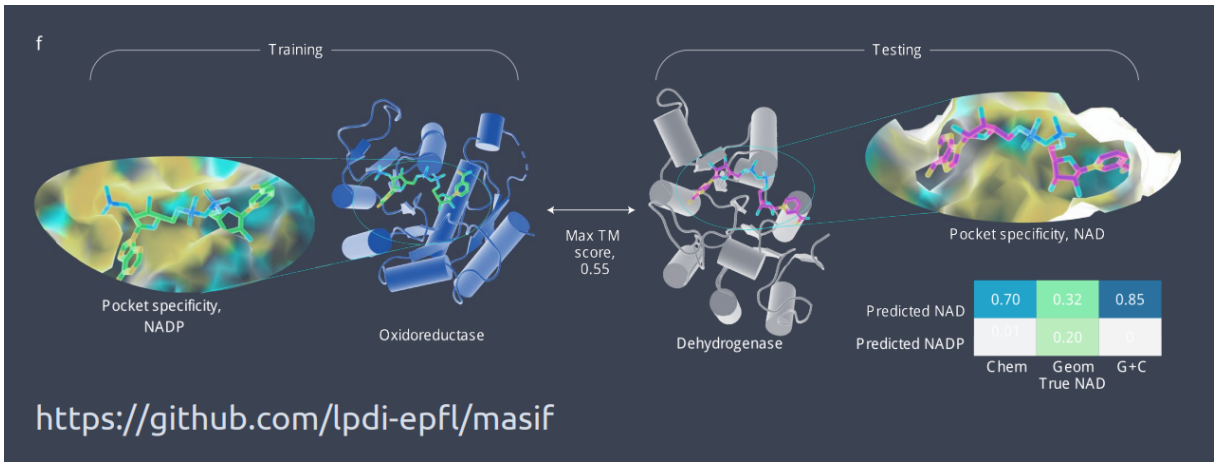

Рис. 3.17. Кинирование НАДФ

Следующая задача похожа на генеративные подходы. Докинг и межмолекулярное взаимодействие – это не совсем напрямую генеративный подход, скорее **подход по улучшению**. Если у нас есть результаты докинга нескольких сотен веществ, было бы полезно обучиться, что можно поменять в веществе, чтобы докинг стал ещё лучше. Это как бы фрагментарный подход на лету. То есть, у нас у нас есть соединение в сайте связывания, начинаем вставлять разные заместители, и в процессе работы идёт обучение, что куда вставлять, чтобы получились правильные ответы. Вознаграждения и штрафы приходят из докинга.

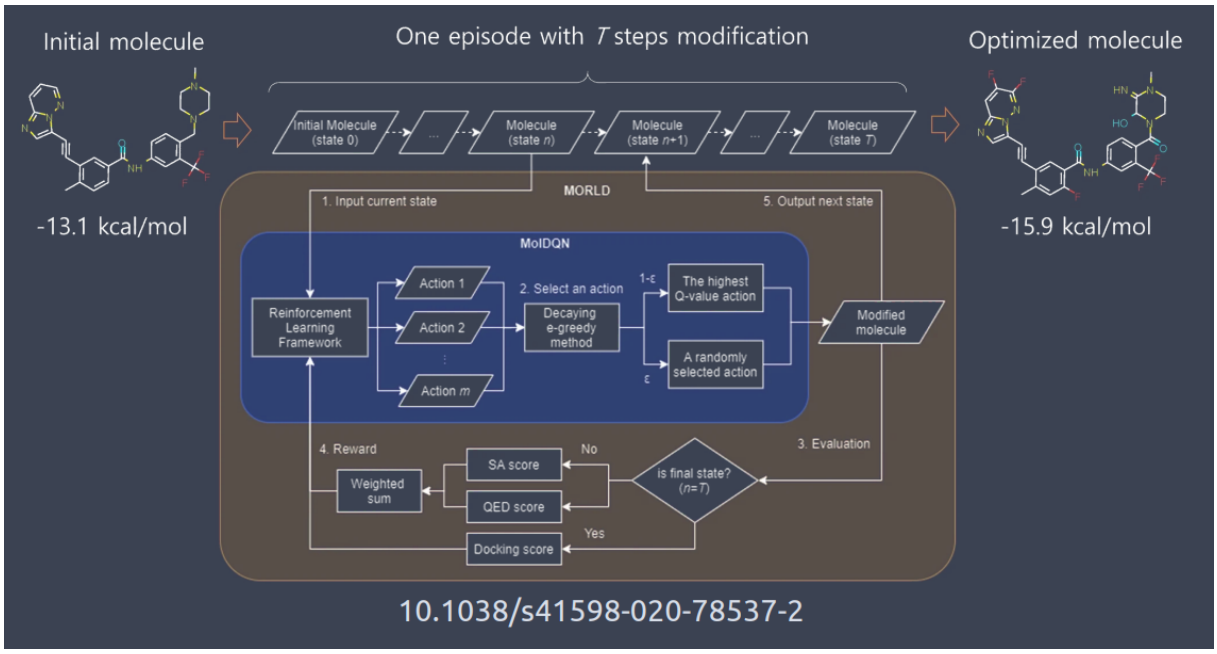

Рис. 3.18. MORLD

Идея такова, что у нас есть **MORLD** – сетка по генерации заместителей от данных веществ (рис. 3.18). Она работает, и ей нужны функции, которые говорят, что хорошо, что плохо. Эта информация приходит из докинга. Постоянно происходит итеративное

улучшение. С помощью этого подхода смогли улучшить энергию связывания на 2 ккал/моль, а это много.

Мы даём на вход первичный докинг – молекулу известного лекарства. Мы знаем, как она расположена. Дальше на основе сетки MORLD наносим разные модификации. Модифицированные молекулы делаем докинг так, чтобы он был максимально близок к тому докингу, который был исходно. И докинг говорит, что хорошо, что плохо. Происходит итеративное обучение, и в результате получается новая молекула на основе известной.

В этой статье была попытка сделать структурные представления. С точки зрения нейронных сетей очень полезно иметь возможность интегрировать весь процесс от начала до конца и не иметь молекулярно-механической подложки типа докинга или оптимизации геометрия белка, потому что данный процесс нельзя оптимизировать, он просто говорит «да» или «нет».

Альфадок эту проблему решил. В первом AlphaFold у нас были попарные расстояния, они оптимизировались молекулярно-механически. Потом там эта процедура была заменена нейросеткой, отчего попарные расстояния превращаются в структуру, и всё интегрируется вперёд и назад. Это позволяет быстро улучшать предсказания структуры, чтобы было максимально похоже на PDB, потому что обучить этот последний шаг по молекулярно-механической на основе попарных расстояний нельзя.

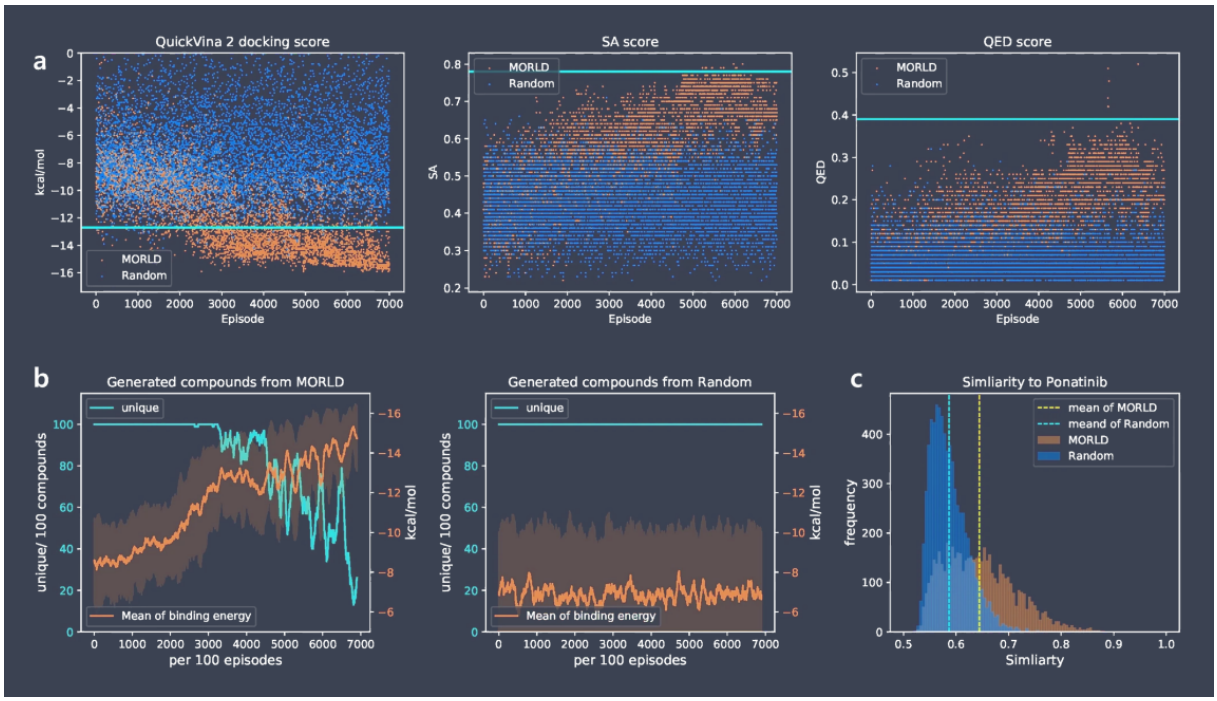

Рис. 3.19. Результаты

Но здесь эта проблема не решена, потому что при модификации лиганда происходит его смещение, и предсказать смещение уже нельзя. Поэтому важно не только научиться представлять будущую молекулу в 3D в том же положении и получать

изменение геометрии, но и смещать её относительно положения в белке. Пока это можно делать только докинг.

Посмотрим на результаты (рис. 3.19). Достаточно хорошо происходит генерация, постоянно увеличивается средняя энергия.

Приведём ещё один пример подобного подхода (рис. 3.20). Наши коллеги из России сделали новое соединение за два месяца просто исходя из того, что у них был набор стартовых значений, и дальше они методом генерации с вознаграждением добились того, что у них получилась структура, которая обладала хорошей биологической активностью.

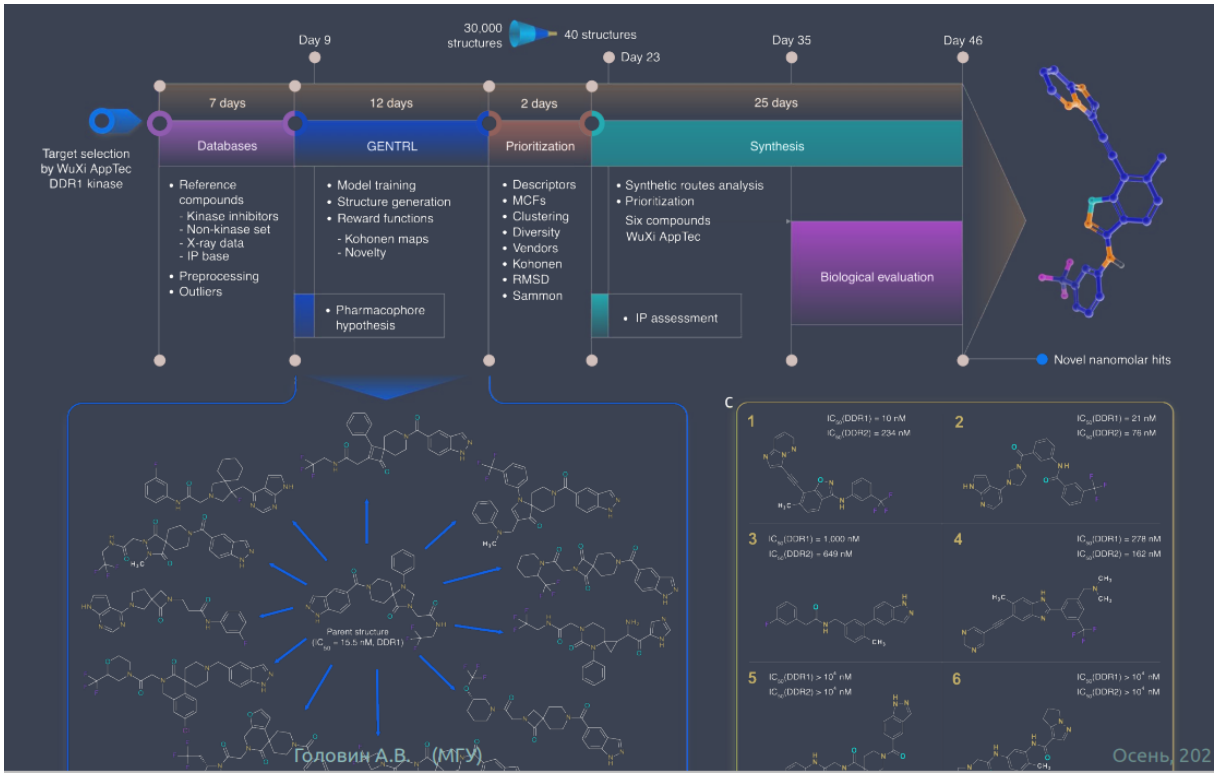

Рис. 3.20. GENTRL

Любое лекарство должно хорошо связываться с белком. Чем лучше константа связывания, тем меньше нужна концентрация лекарства в крови.

Концепция агонистов, антагонистов и т. д. возникает из истории про рецепторы. Рецепторы – это сигналящие молекулы, которые от взаимодействия на внешней стороне клетки передают сигнал внутрь клетки через мембрану. Нельзя ожидать, что мы узнаем переход от конформационной динамики к передаче сигнала, просто исследуя сайт связывания. Это можно узнать из механических соображений, но, когда мы выбираем два сета для конкретного рецептора, они зачастую ничтожно малы. В каждом сете максимум сотня соединений, поэтому ошибка будет большой. Поэтому сравнение сетов доверия не вызывает.

Лекция 4. Сравнительное моделирование

Введение

Рассмотрим **основные проблемы**. Если мы возьмём белок, состоящий из 100 аминокислот, у нас будет 200 углов фи и пси, которые надо просканировать. У каждого угла может быть много значений, степеней свободы 3N. В итоге всё разнообразие значений торсионных углов можно представить как 10^{48} разных конформаций.

Очевидно, что даже если бы мы просто сканировали все конформации белка, а сворачивалась бы самая удачная, всё равно для того, чтобы белок собирался всегда одним и тем же образом, уходило бы очень много времени. Естественно, из случайного набора разных структур это можно сделать гораздо быстрее. Но наша цель в том, чтобы белки собирались одинаково.

Природа подсказывает, что, если белок выделен, он всегда находится в одной конформации или в функционально близких конформациях. Здесь возникает вопрос, как это вообще может быть. Данный парадокс сформулирован как **парадокс Левинталя:** «Промежуток времени, за который собираются белки, на много порядков меньше, чем если бы полипептид просто перебирал все возможные конфигурации».

Обратимся к уже накопленным данным о структурах и попытается понять, **почему так происходит.**

Все наши теоретические модели, как правило, основываются на химии и физике равновесных систем. Зачастую, когда мы говорим о химии и пытаемся что-то моделировать, рассматриваем поведение одной молекулы или небольшого ансамбля. А в природе очень большое разнообразие молекул находится в очень компактной среде, и описать это численно достаточно сложно.

Ещё в природе часто важна не самая оптимальная по энергии молекула, а такая, которая обладает определённым функционалом, то есть чтобы молекула в определённом диапазоне температур могла, например, изменять конформацию, проводить узнавание и т. д. Поэтому можно предположить, что в природе суперстабильные биополимеры не нужны, потому что их невозможно регулировать. Это важно, так как у природных объектов постоянно меняется окружение, и эффективность процессов должна напрямую зависеть от изменения условий.

Если меняются условия, эволюционно выживают те организмы, которые к ним хорошо приспосабливаются. Когда мы говорим о структуре белков, это те, у кого белки быстро собираются, быстро работают, быстро производят продукт. Возможно, не так эффективно, как было бы идеально, не в таком широком температурном диапазоне, но достаточно для того, чтобы иметь эволюционное преимущество.

Белки могут сворачиваться разными способами, не обязательно следуя глобально оптимальному пути. Это мультинаправленный процесс, в котором есть разные кинетические барьеры, которые преодолеваются по-разному. Есть общие наблюдения,

но количественные законы, определяющие, как сворачивается белок, мы пока вывести не можем.

Считается, что структура определяется последовательностью, но бывает так, что развитые организмы используют дополнительные белки для того, чтобы структура гарантированно собиралась нужным образом, например, шапероны.

И ещё структура белка более консервативна с точки зрения общей геометрии, чем последовательность. Это неудивительно, так как если у нас есть гидрофобное ядро, мы может в нём перетасовать остатки практически любым способом, главное, чтобы у него сохранялся общий гидрофобный объём. Это пример того, когда сохранение структуры возможно при значительном изменении последовательности.

Здесь у нас появляется такое понятие, как **сравнительное моделирование**.

Зачем искать конформации, если можно представить, что при высокой идентичности последовательностей очень высока вероятность, что подобны и структуры. Единственное, здесь надо понять, где находится trash hold, степень идентичности между той структурой, которая нам известна, и последовательностью той структуры, которая неизвестна, чтобы мы могли судить, что структуры у них должны быть близкие.

Также надо понимать, где возможны экспериментальные допущения.

Сейчас известно порядка $1{,}6 \cdot 10^5$ **структур уникальных белков**. Большинство из них (562000 белков) представлены в UniProt. Также там есть небольшое количество других белков, например, те, которые не кристаллизованы. Ещё есть неструктурированные белки, трансмембранные, поэтому есть большое число последовательностей, для которых структура не определена, но может быть представлена. Для 50% известных последовательностей можно легко предсказать способ укладки. А если сказать честно, то мы сможем предсказать способ укладки примерно для трети вообще всех известных последовательностей.

Посмотрим, для чего может быть использовано сравнительное моделирование, когда мы на основе структуры одного белка строим структуру для той последовательности, которую мы хотим исследовать. В большинстве случаев, когда у нас степень идентичности попадает в диапазон от 30% до 100%, мы получаем очень качественные структуры, которые могут быть использованы для разработки лекарств (рис. 4.1). Но, если степень невысокая и мы не уверены в том, что у нас получается, мы можем попытаться использовать эту модель для работы с другими биологическими данными, чтобы оценить качественно сближенность некоторых остатков.

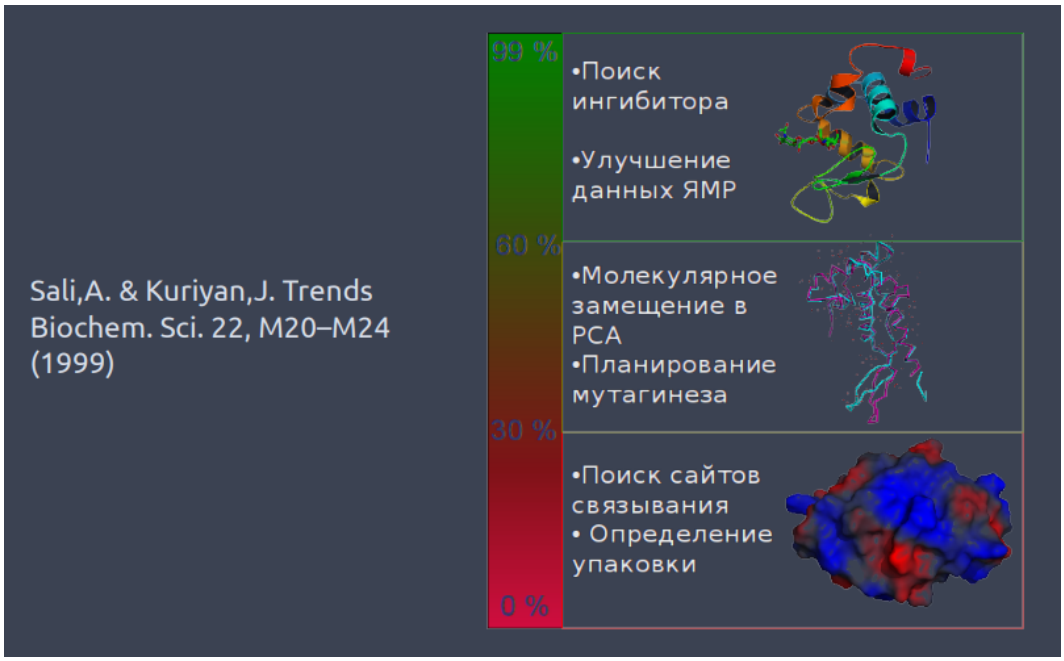

Рис. 4.1. Степень идентичности

Сейчас появилась достаточно популярная область исследования: интегративное моделирование. В нём нескольких экспериментальных данных о сближенности остатков или об определённой форме белка достаточно для того, чтобы очень эффективно строить качественные модели, потому что небольшие ограничения сильно сужают пространство поиска. Здесь сравнительное моделирование является одним из первых шагов, чтобы эффективно пользоваться инструментами области такого моделирования.

Сравнительное моделирование

Как сделать модель на основе известной структуры и предположении о совпадении исследуемой последовательности и известной? Сначала надо построить первичное выравнивание двух последовательностей: расположить буквы так, чтобы они давали максимальную идентичность. После этого выравнивание надо улучшить. Потом строится ход основной цепи, моделируются петли, достраивается или моделируется положение боковых радикалов.

Пройдёмся подробнее по этим шагам. Как в принципе **найти белкок-заготовку**, на основе которой мы можем построить модель? Берём Blast и делаем по поиск по PDB – по базе данных структур. Очень вероятно, что мы из этого ничего не найдём. Тогда надо Psi-Blast, который позволит вытянуть хоть какие-то родственные соединения. Ещё используются методы распознавания укладки и др.

Можно использовать биологическую информацию. Например, мы знаем, что этот белок-заготовка и наша последовательность точно имеют одинаковый fold или вид структуры, и тогда на основе биологических соображений что-то делаем. Часто используется функциональное аннотирование в базах данных, а также информация об активных сайтах, или мотивы.

После того, как мы приняли решение, что для этой последовательности заготовкой будет такая структура, мы делаем саму последовательность. Способов сделать выравнивание достаточно много. При выравнивании последовательности полезно учитывать, что мы имеем дело со структурой. Значит, когда мы делаем выравнивание, надо понимать, при возникновении гэпов можно ли сместить их так, чтобы расстояние между остатками, которые потом находятся на краях этого гэпа, было минимальным.

Например, на рис 4.2. зелёная последовательность основная, и есть два способа выравнивания – красный и синий. Красный даёт гэп с большими расстояниями между аминокислотами, а синий – с маленькими. Очевидно, это гораздо правильнее с точки зрения структуры выравнивания.

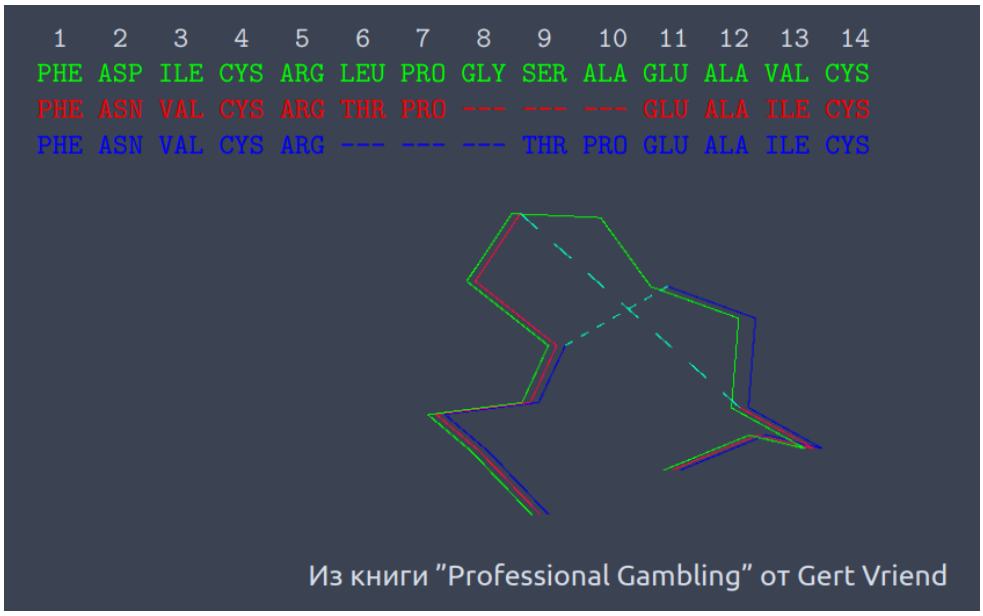

Рис. 4.2. Улучшение моделирования

Как влияет **качество белка заготовки** на моделирование? Качественные модели можно получить, только имея качественную структуру. Чем выше разрешение, тем лучше (рис. 4.3). Небольшая разница в идентичности не так важна, главное, чтобы было высокое качество исходной структуры, тогда и модель будет хорошей.

Важно понимать, что даже идентичность 100% может быть бессмысленной для коротких пептидов. Идентичность рассчитывается в количестве совпадений на всю длину последовательности. Поэтому совпадение этих последовательностей может быть просто случайным.

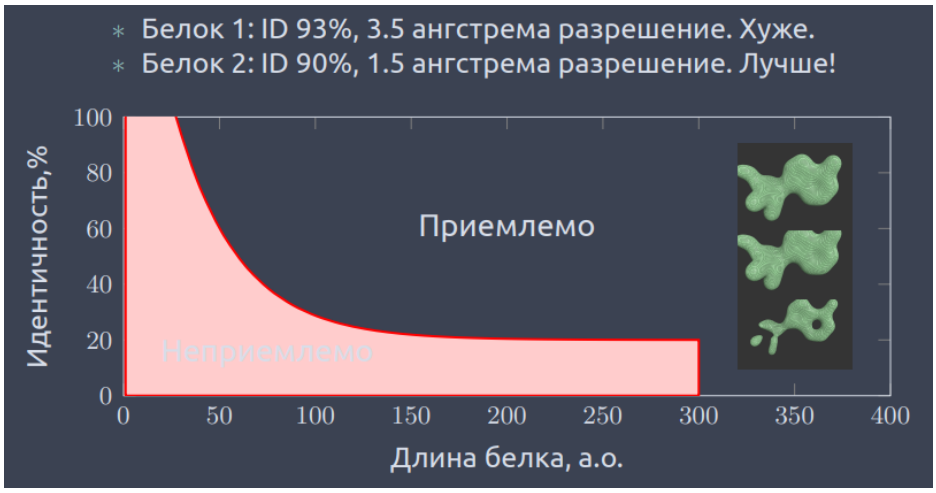

Рис. 4.3. Качество белка заготовки

Очевидно, что для моделирования есть смысл рассматривать только те белки, длина которых больше 100 аминокислот.

Можно ли использовать для моделирования данные **ядерно-магнитного резонанса**? Формально эти структуры и модели лежат в базе данных PDB, и мы можем их использовать для построения нашей модели.

Но не всё так просто. На рис. 4.4. видно, что есть некая часть белка, которая подвижна. Так получается, например, когда все модели, которые есть в ЯМР, накладываются друг на друга. С одной стороны, можно подумать, что в ЯМР мы смотрим белок в растворе и получаем много конформаций, потому что раствор – это не кристалл. А в реальности зачастую, когда мы видим такое разнообразие конформаций, это означает, что мы не получили в ЯМР информацию о попарных расстояниях на основе спектров. В таком случае все эти модели достраиваются с помощью молекулярного моделирования. Зачастую там используются примитивные алгоритмы, и мы не знаем, как выглядят эти петли в реальном белке. Поэтому мы имеем несколько разных моделей, из которых люди выбирают на своё усмотрение. Но говорить, что все эти варианты имеют право на жизнь нельзя, пока под это не подложены конкретные данные из спектрального анализа в ЯМР.

Рис. 4.4. Белок с подвижной частью в ЯМР

И тут можно сказать, что желательно, чтобы высокая степень идентичности не попадала на эти участки, потому что мы не можем построить модель на основе непонятно как сделанной ещё одной модели.

Можно посмотреть на **карты Рамачандрана** (рис. 4.5). Слева карта, сделанная для РСА, справа – для ЯМР. Первая выглядит гораздо лучше, но здесь есть недостаток: вся эта статистика опирается на базу данных PDB, а она в основном наполнена RCA. Поэтому неудивительно, что рентгеноструктурные данные хорошо совпадают со статистикой по рентгеноструктурным данным.

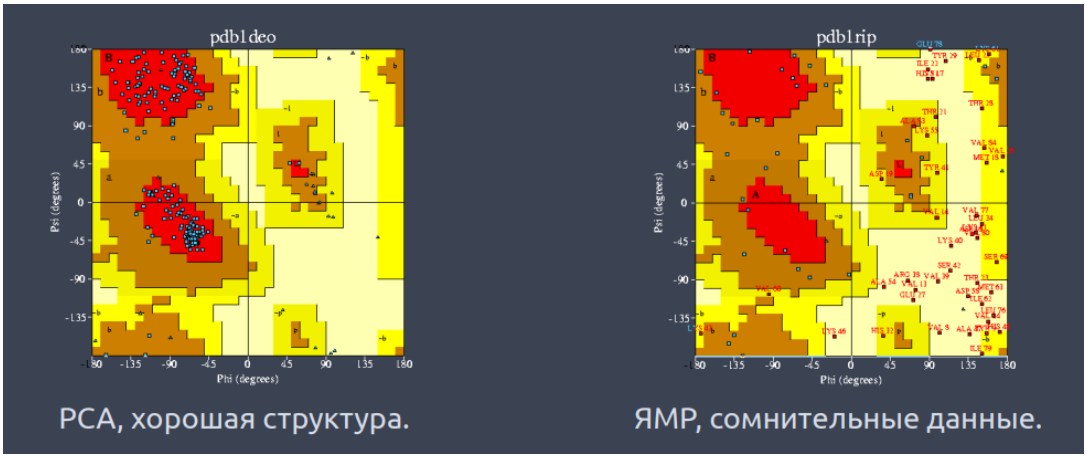

Рис. 4.5. Карты Рамачандрана для РСА и ЯМР

Итак, мы определились с белком-заготовкой и с выравниванием. Как **построить остов**? Для совпадающих аминокислот, да и практически для всех, которые не принадлежат выпетливаниям и гэпам, можно скопировать координаты и получить расположение, как минимум, их остова, потому что вне зависимости от того, совпадают или нет аминокислоты выравнивания, атомы остова у них одинаковые.

Надо понимать, что для построения не обязательно всегда использовать координаты. Можно использовать попарные расстояния между некоторыми атомами (дистанционные ограничения). Большинство исследователей предпочитают для этого пакет Modeller.

Что делать для **моделирования петель**? Петли там, где есть гэпы выравнивания, в модели или в исходной структуре. Последнее хуже, потому что появляются дополнительные аминокислоты, которые надо как-то расположить, а как, мы не знаем. Можно попытаться скопировать их геометрию из базы данных PDB или использовать другие базы данных с геометрией петель (например, LIP). В них можно сканировать так, чтобы начало хорошо совпадало и подбирать те, которые хорошо влезают в это окружение. Также можно воспользоваться любым методом молекулярной механики.

Можно перебрать состояния в петле с помощью метода Монте-Карло, но это довольно затруднительно. Разумнее использовать Rosetta: генерировать петли с помощью фрагментов, близких по последовательности, или сделать комбинаторный поиск с помощью Монте-Карло.

Все эти методы можно варьировать между собой. Например, сделать грубую прикидку из PDB, потом оптимизировать её Rosetta с Монте-Карло, потом провести ещё молекулярную динамику и посмотреть, что из этого получится. Первый метод даёт приемлемое расположение остатков внутри петли, второй двигает так, чтобы это не пересекалось с окружением в белке, а третий оптимизирует гидрофобный эффект так, чтобы при правильности предыдущих шагов получился верный ответ.

Что делать с **боковыми радикалами**? В большинстве случаев самое верное – копировать их состояние из белка-заготовки. Если боковые радикалы, которые образовывали контакты в исходном белке, консервативны, можно попытаться восстановить все третичные контакты, которые есть в белке заготовки. Исключение – когда значительно меняется природа аминокислот. Здесь самое правильное – взять как можно больше конформаций боковых радикалов из белка-заготовки, а всё остальное вокруг этого навесить.

Есть наблюдения, говорящие о том, что если мы проанализируем PDB, то конформация боковых радикалов зависит от конформации основной цепи. Существуют базы данных ротамеров, которые можно использовать, чтобы реализовать то или иное состояние боковых радикалов в данном случае. Методы могут быть разные, но один из них уже давно популярен – SCWRL. Это эмпирический метод на основе теории графов, который пытается расположить боковые радикалы наиболее оптимальным способом.

Точность моделирования для боковых радикалов зависит от того, куда этот радикал направлен. Если внутрь белка, где высокая плотность упаковки, и мы может ожидать, что эта плотность в большинстве случаев инвариантна, то если скопировать из модели и расположить основные боковые радикалы аккуратно, то остальные тоже расположатся хорошо, потому что других вариантов обычно стерически не предполагается.

С другой стороны, очевидно, что если мы будем рассматривать остатки на поверхности белка, здесь всё хуже по той причине, что они там могут быть подвижны, а также участвовать во взаимодействиях с другими молекулами, поэтому трудно оценить то, в каком состоянии они должны быть. Ещё там есть контакт с участием воды, а описанное нами моделирование не предполагает в явном виде учёт воды вокруг белка, потому что получается слишком сложно.

Как можно **улучшить модель**? В большинстве случаев можно попытаться оптимизировать энергию, но это не надо, потому что все современные движки, которые строят модели белков на основе попарных расстояний, используют стандартные силовые поля для оптимизации геометрии. Иногда с какими-то поправками, но в итоге это всё становится стандартным силовым полем, и новый запуск минимизации вряд ли покажет что-то значимое.

Запуск молекулярной динамики может сильно улучшить ситуацию по оптимизации гидрофобики. Также можно дооптимизировать модель с помощью методов Монте-Карло или использовать любой другой метод оптимизации структуры.

Ошибки, допущенные на ранних этапах моделирования, не могут быть исправлены на более поздних, потому что они будут принципиальными. Хорошее выравнивание не исправит плохой выбор белка заготовки. Молекулярная динамика не исправит ошибки при составлении выравнивания. Поэтому при обнаружении ошибки надо переделать некоторые этапы. Это делается быстро и не очень затратно. Возможно, даже было бы удобно делать много белков заготовок, потом применять методы оптимизации, а потом делать выбор в пользу той или иной модели.

Как можно **провести** проверку того, что мы делаем? Проверять значения длин углов и значения по связям бессмысленно, потому что все эти модели оптимизированы с помощью методов молекулярной механики. Пытаться выявить неправильные аминокислоты с помощью карт Рамачандрана можно, но, если появились новые петли, все они будут соответствовать неправильным регионам на карте. Это моделирование, а оно не всегда хорошо попадает в статистику по рентгеноструктурному анализу. Детальные анализы типа положения или ориентации заряженных аминокислот, например, нахождение заряженной аминокислоты в гидрофобном окружении, трудно автоматизировать, но за ними может следить человек.

Любые экспериментальные данные, которые могут сказать, правильная модель или нет – допустим, проверки по каталитическим сайтам, по активным сайтам, сайтам связывания. Если какие-то аминокислоты участвуют в сайте связывания определённого белка, они должны находиться близко на поверхности белка в модели. Это очень хороший критерий для отсеивания неправильных моделей. То же самое можно говорить о местах контактов, местах модификаций, и, если знать, что определённые аминокислоты модифицируются некими реагентами, можно ожидать, что они находятся на поверхности, и в модели это должно тоже быть.

Не все способы проверки модели можно сделать автоматическими. Есть ProQ сервер, но он оптимизирован на поиск модели, правильной с точки зрения канонов моделирования, а не нативной структуры. Мета-сервисы отличаются тем, что пытаются найти консенсус между разными подходами к моделированию. Этот поиск нередко приводит к тому, что с какой-то вероятностью правильная модель находится в топе, но какая из них правильная – неизвестно.

На сегодняшний день достаточно **много пакетов и серверов для гомологичного моделирования.** Modeler и Swiss Model занимаются классическим сравнительным моделированием. Первый – модуль питона, второй – веб-сервер. В Modeler можно вручную задавать ограничения и лепить белок так, как мы хотим. Swiss Model автоматический: мы копируем последовательность в поле ввода и получаем на выходе желаемую структуру. Ещё есть Eva-CM, Nest и т. д.

Моделирование Ab initio

Можем ли мы использовать какие-то методы для **предсказания белков** с точки зрения **Ab initio** моделирования? Ab initio – это когда нам не известна структура белка-заготовки или мы не смогли её найти. Теоретически можно использовать молекулярную динамику. Это нереалистично для больших белков, но для маленьких возможно. Можно использовать методы моделирования отжига. Чтобы это получилось, надо иметь много опыта, и он не будет экстраполироваться на другие объекты. Можно попытаться построить на основе фрагментов, как это делает Rosetta.

Rosetta пытается на основании предсказаний подобрать фрагменты, из которых можно было бы собрать данный белок (рис. 4.6). Фрагменты небольшие, от 3 до 9 остатков. Перебирая все фрагменты друг против друга, делая между ними разные торсионные углы, пытаемся добиться того, чтобы из них собралась относительно стабильная структура.

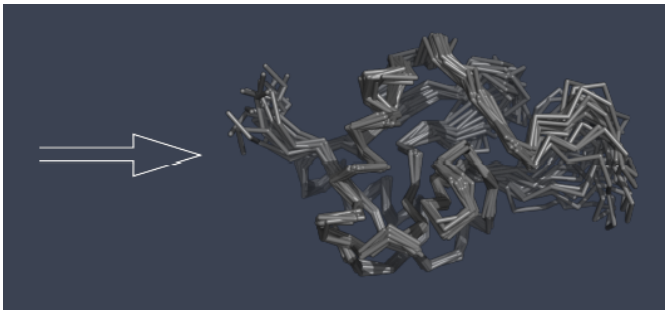

Рис. 4.6. Ab initio с помощью Rosetta

Подобный подход имеет много ограничений. Мы не можем предугадать, сколько нам нужно считать для данного белка.

Rosetta – большое моделирование методом Монте-Карло с разными уровнями точности описания, детализации атомарной структуры белка. Допустим, может быть описание фрагмента как жёсткого тела, а может добавляться движение. Эти описания направлены на то, чтобы как можно быстрее сделать большое конформационное пространство, которое имеет этот белок.

Монте-Карло – достаточно сложная процедура с точки зрения моделирования растворителя, растворители там задаются в неявном виде. Отсюда возникает необходимость в эмпирическом описании большинства попарных взаимодействий, которые у нас есть в белках. Гидрофобные потенциалы тоже являются попарными.

Здесь появляется дополнительное требование к потенциалу для третичных контактов. У нас есть белок-заготовка, а если нет, можно найти таблицу, где написано, какие аминокислоты могут образовывать между собой третичные контакты. Сюда же можно добавить информацию о дисульфидных мостиках, местах связывания катионов металлов и т. д.

Threading – протягивание нити

Альтернативный способ – метод **Threading.** На данный момент он практически не используется. Он действует только для тех белков, для которых если и находятся заготовки, то с очень низкой степенью идентичности. Тогда гораздо проще прогнать эту последовательность через все известные способы укладки и понять, какая из них лучше всего подходит к данной последовательности.

Мы стараемся во всех известных способах укладки определить некие тенденции о третичных контактах разных аминокислот, после чего каждую аминокислоту в нашей модели помещаем в соответствующую позицию в разных укладках, то есть делаем некие матрицы, которые соответствуют разным укладкам, и отсюда высчитываем их score. Та укладка, в которую данная последовательность укладывается лучше всего, будет иметь максимальный score и станет нашим белком заготовки.

На рисунке видно 4.7, что одну и ту же последовательность можно скрутить по-разному, и будет разный score. Получается, что какие-то способы скручивания полипептидной цепи гораздо лучше, чем другие.

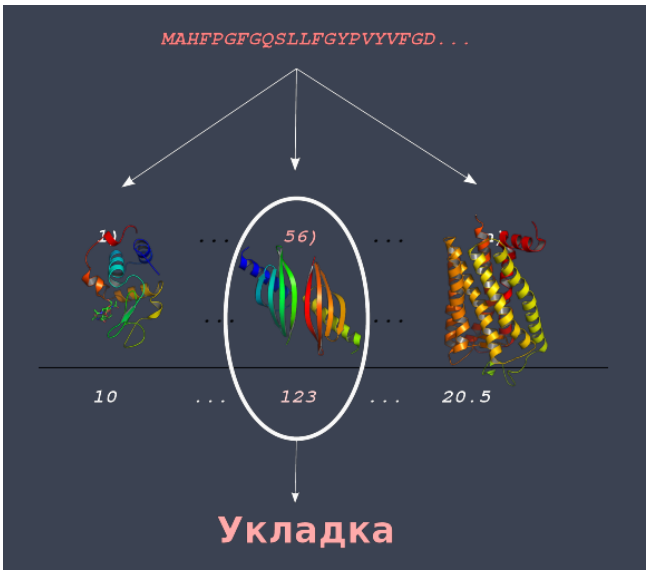

Рис. 4.7. Разные способы скрутить последовательность

Основные **недостатки** данного метода в том, точность моделирования в целом невысока из-за того, что взаимодействия в белке не всегда описываются парными контактами. Допустим, гидрофобное ядро так описать затруднительно. Все потенциалы, с помощью которых пытаются описать потенциальные взаимодействия между аминокислотами, опираются на профили аминокислотных последовательностей. У профилей разнообразие гораздо больше, чем при выравнивании двух последовательностей, но пока мы всё равно теряем некоторую структурную информацию, которая могла быть в нашем белке.

Один из лучших сервисов для предсказания структуры белков – гибридный метод, который использует и Rosseta, и Threading. Он называется I-Tasser.

Распознавание укладки

Один из примеров хорошего использования профилей – это распознавание укладки с помощью алгоритма Phyre2 (рис. 4.8). У нас есть последовательность, и мы для неё набираем её гомологи из всех известных данных. Если все подобранные вещества – действительно гомологи, они хорошо выравниваются и имеют большое разнообразие, то из этого можно построить HMM профиль. Это таблица, в которой написано, с какой вероятностью та или иная аминокислота находится в той или иной позиции последовательности, основываясь на выравнивании.

Рис. 4.8. Распознавание укладки с помощью Phyre2

С таким же успехом мы можем сравнить этот профиль с известными профилями для всех известных структур (рис. 4.9). Для каждой известной структуры белка можно построить её профиль с её гомологами. Такое выравнивание профиля против профиля гораздо эффективнее, чем выравнивание последовательности против последовательности или последовательности против профиля. В результате мы получаем эффективный выбор белка заготовки, опирающийся не просто факт схожести последовательностей, но и на то, что есть пересечение между семействами. Итог в том, что у нас получается структура, на основе которой можно построить 3D модель. И нередко там выравнивание бывает хорошим.

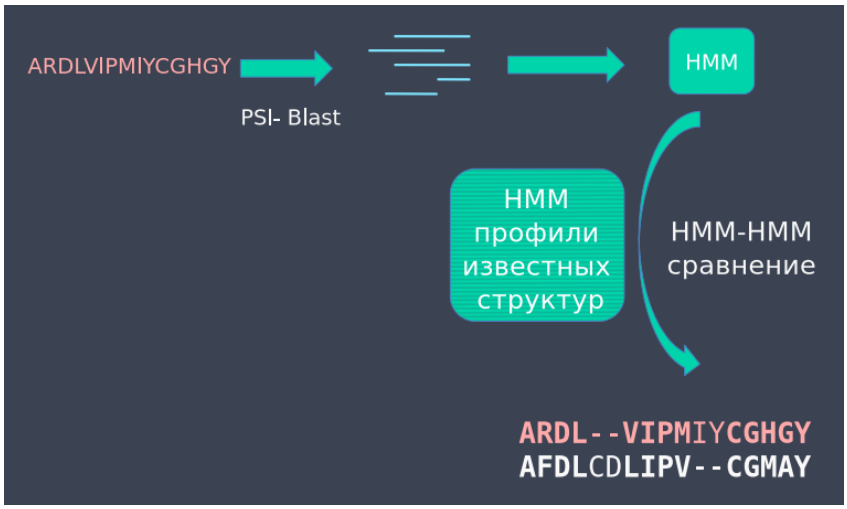

Рис. 4.9. Phyre2, сравнение с профилями для всех известных структур

Мета серверы

Мета серверы пытаются запускать один и тот же запрос разными способами и делают из этого консенсус. Если считать, что разные подходы дадут разные ответы, то в этом есть смысл. Но зачастую все подходы дают один и тот же белок или очень похожие

белок-заготовки. Так что, если мы берём на вход одно и то же, на выходе тоже получим одно и то же, потому что строиться всё будет просто по подобию.

ML методы для предсказания структуры

Рассмотрим, как растёт представленность новых последовательностей в современном мире (рис. 4.10). Голубое – количество последовательностей, которые появляются каждый год, их много. А вот новых кластеров по последовательностям практически не появляется. Скорее всего, мы скоро узнаем все последовательности и кластеры белков, которые нам доступны.

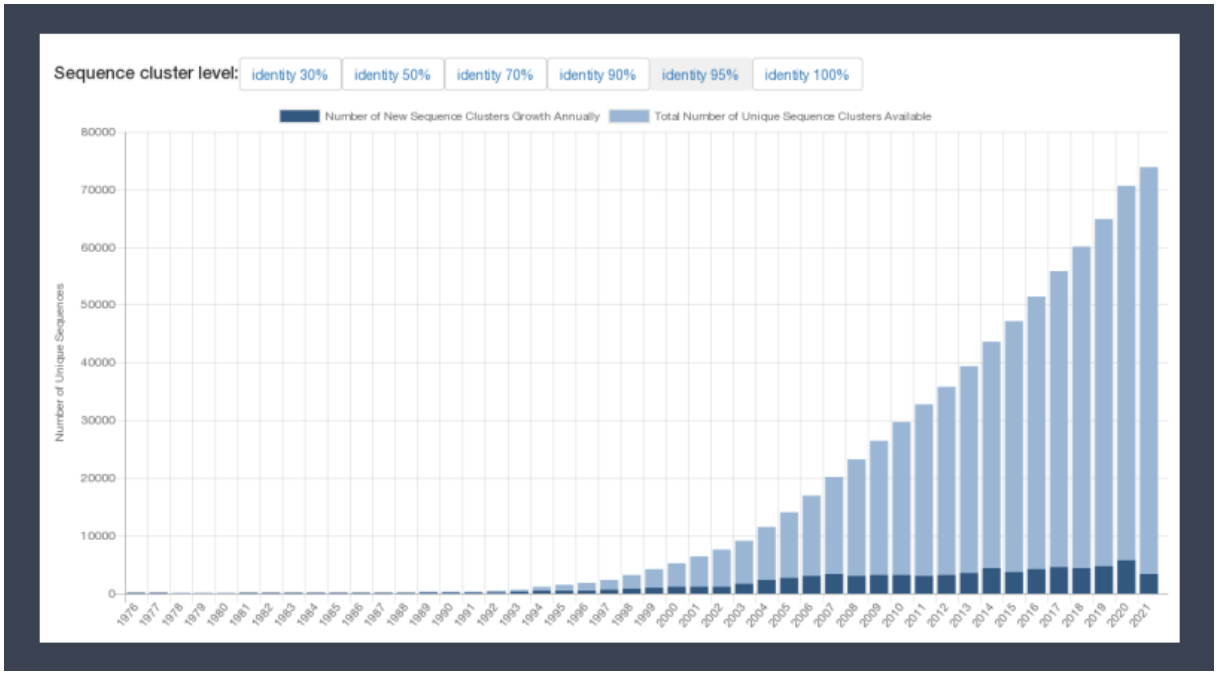

Рис. 4.10. Рост представленности новых последовательностей в современном мире

Количество записей в **UNIPROT** уже почти перестало расти, оно выходит на некое плато (рис. 4.11).

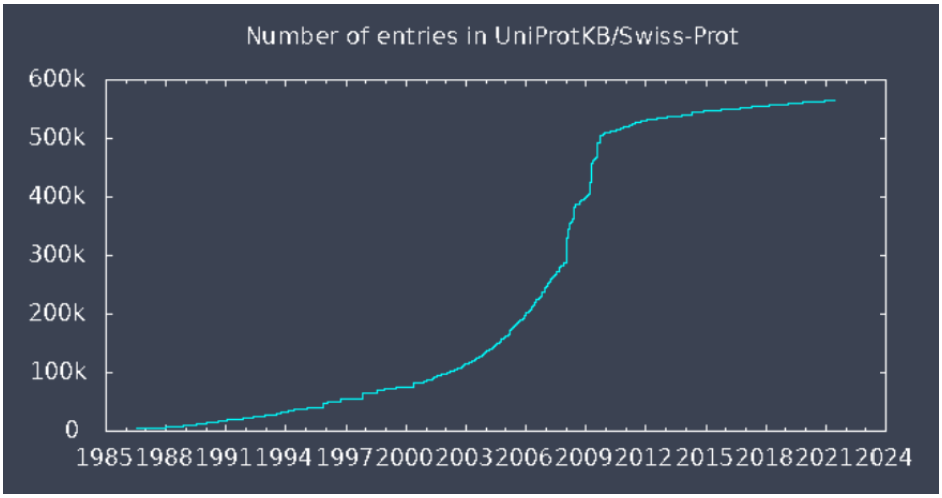

Рис. 4.11. Открытия в UNIPROT

Здесь в базе данных описывается именно белок, а мутации – уже внутри данных записей. Видно, что даже если новые последовательности появляются, новые белки оттуда появляются всё хуже и хуже.

На сегодняшний момент можно сказать, что мы обработали примерно 80% белков. **PFAM** – это база данных, профилей семейства белковых. В реальности уже почти все белки отнесены к семействам. И уже можно эффективно использовать накопленную информацию о структурах для эффективного предсказания структур белков.

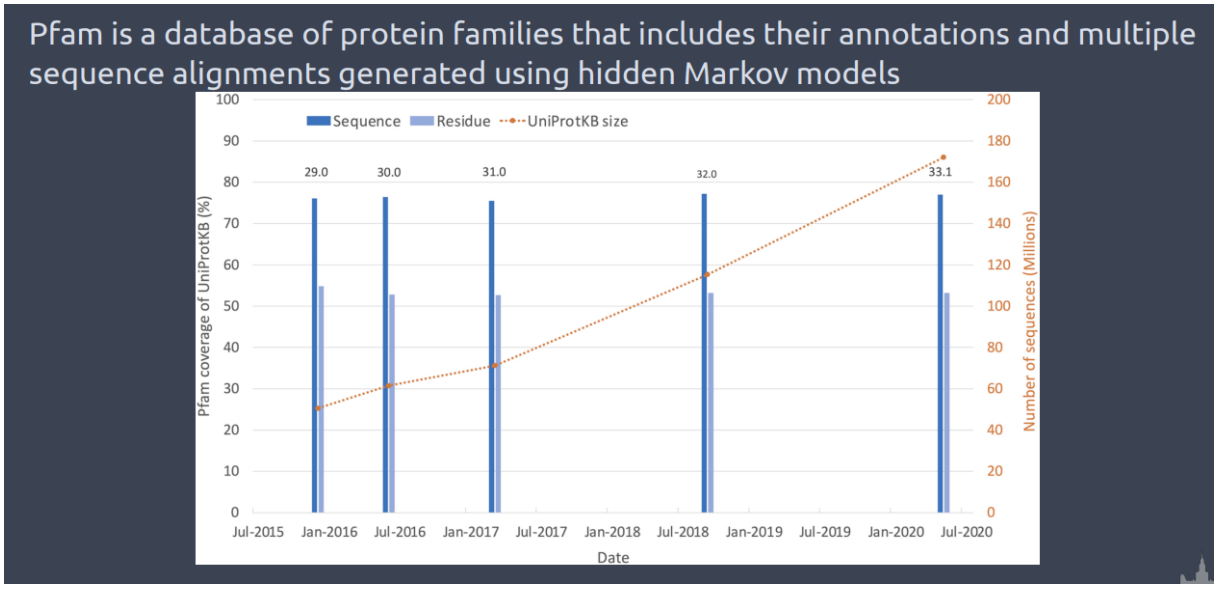

Рис. 4.12. *PFAM*

Переходим к машинному обучению. Очевидно, что структура белка содержит в себе очень много **Features** (рис. 4.13).

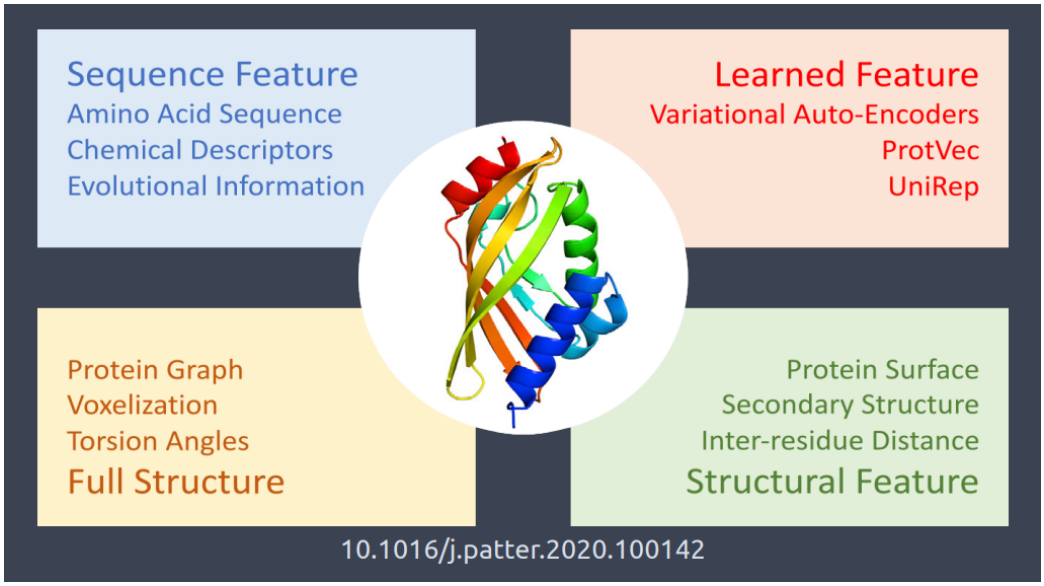

Рис. 4.13. Features в структуре белка

Мы можем опираться на последовательности и использовать эволюционную информацию, химические дескрипторы для описания структуры белка. Можем автоматически найти дополнительные Feature с помощью автоинкодеров и т. д. Можем описывать структуру белка через теорию графов. Ещё могут быстро структурные особенности: поверхность, вторичная структура, расстояние между остатками.

В чём принципиально отличие расстояния остаток-остаток от полной структуры? Одному набору расстояний остаток-остаток может соответствовать несколько разных структур, причём они могут быть зеркальным отражением друг друга. Поэтому попарные расстояния хороши, но не являются полным описанием структуры.

Пройдёмся по этапам. Какое **представление последовательности** мы можем использовать при описании белка? Естественное представление – это аминокислота как одно целое число. Здесь можем добавлять особенности, связанные с выравниванием аминокислот, с том числе со взвешенными матрицами, например, MSA, PSSM как реальное число. Дальше к вектору можем добавлять информацию о вторичной структуре – 3 или 8 букв в зависимости от того, насколько точно нужно описывать. А также данные о коэволюции: если белок эволюционирует, то некоторая часть аминокислот, которая взаимодействует сама с собой и нужна для формирования структуры этого белка, будет консервативна или будет коэволюционировать – чтобы сохранялся контакт при изменении одного будет меняться другое. Эта информация может быть очень полезна при предсказании попарных контактов. Так на основе просто выравнивания и весовых матриц можно научиться строить векторы, которые учитывают всё вышеперечисленное.

Мы можем **экстрагировать представления** с помощью методов альтернативного языка. Здесь некоторые подходы работают очень хорошо. Можно разбивать на неперекрывающиеся трипептиды, объявлять их словами и с помощью слов обучать модель, чтобы предсказывать текст всего белка, и уже это представление экстраполировать на структуру. Это хорошо работает для небольших пептидов.

А вот BERT и GPT3 хорошо работают в узнавании вторичной структуры. Если есть паттерн гидрофобных и гидрофильных аминокислот, он формально отображает, к какому элементу вторичной структуры может принадлежать часть последовательности.

AE и VAE удачно применяются для связи последовательности со стабильностью. Здесь есть последовательность и ΔG_1. Есть мутант данной последовательности и ΔG_2. Данным методом строятся разумные корреляции.

Если речь про **предсказание структуры**, прямое использование координат атомов затруднительно, потому что структуры формально инвариантны. Они могут вращаться, двигаться в пространстве, и координаты не являются постоянными при сравнении разных структур. Мы могли бы привести все структуры к неким общим координатам, но сделать это аккуратно почти нереально.

Voxels – достаточно удобная вещь, которая используется, и из неё строятся 3D сетки окружения для CNN, и из этого делаются модели.

Торсионные углы неудобны, потому что малые изменения значения торсионного угла в остове в центре белка приводят к тотальному изменению структуры. Соответственно, будет шум.

Если представить структуру как набор парных контактов, это может работать хорошо, потому что уже сделано на примере сравнительного моделирования, когда сравниваются последовательность и последовательность, профиль и профиль и т. д. Здесь надо уметь работать с картами контактов и использовать эти представления для обучения нейронных сетей.

С помощью некоторых графов для CNN можно отделить ферменты от белков, то есть узнать, может ли поверхность белка иметь каталитическую функцию или нет, а также предсказать интерфейсы.

MASIF имеет концепцию геодезии и химической информации для описания поверхностей как таковых.

Каким способом можно **проверять получаемые модели**? Всё зависит от того, какие взаимодействия ожидаются в той модели белка, которую мы хотим построить. Силовые поля сбалансированы для всех стандартных взаимодействий, которые есть в белке. Можно использовать машинное обучение, чтобы внедрить дополнительные попарные расстояния в тех случаях, когда силовые поля не умеют с ними справляться. Если это правильно сделать, все попарные расстояния, которые мы получим из ML, можно эффективно использовать при построении структуры белков.

Варианты NN

Какие типы алгоритмов могут быть использованы? Например, **конволюционные нейронные сети**. (рис. 4.14). Здесь можно обучиться на известных структурах и построить парные контакты, их карты. Это очень похоже на обработку изображений. Отсюда для любой последовательности мы можем попытаться предсказать матрицу попарных контактов, а дальше из этой матрицы построить структуру.

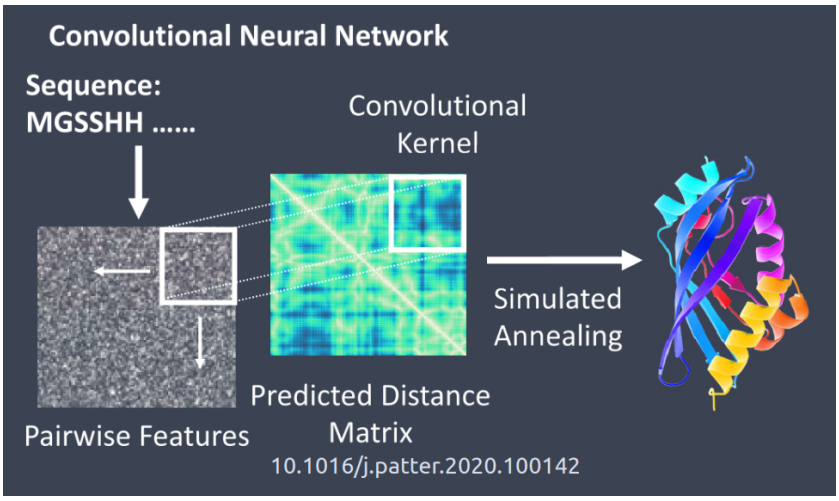

Рис. 4.14. Convolutional NN

Ещё есть **рекуррентные нейронные сети** (рис 4.15). Мы пытаемся понять, насколько хорошо наша последовательность может быть зафиттена в разные структуры. Здесь мы проверяем каждую аминокислоту и из этого делаем заключения о структуре.

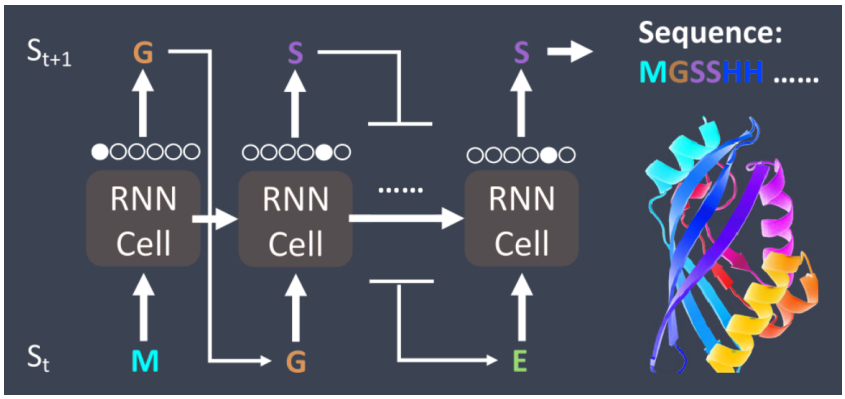

Рис. 4.15. Recurrent NN

Вариационные автоэнкодеры работают через латентное пространство (рис. 4.16). Если мы аккуратно научим кодировать и декодировать структуры белка и сохранять в этом пространстве, то открывается много генеративных возможностей по синтезу новых структур белков.

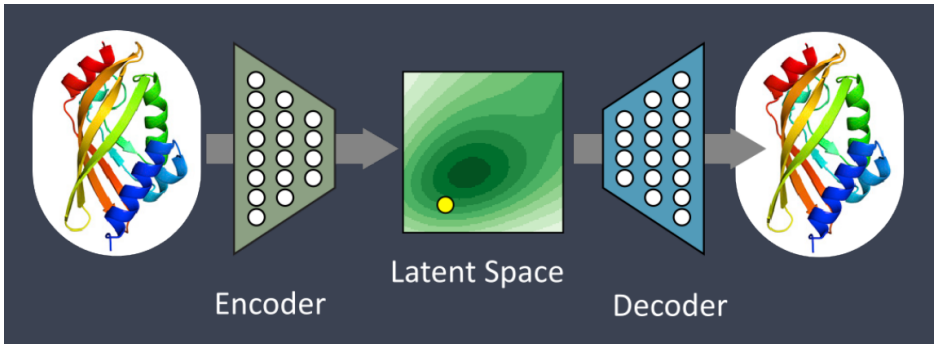

Рис. 4.16. Variational Auto Encoder

Например, на рис. 4.17 мы подаём на вход просто случайный шум, генератор синтезирует нужную структуру, мы получаем статистические данные, и дальше эти данные на основе некого дискриминатора определяются как правда или ложь.

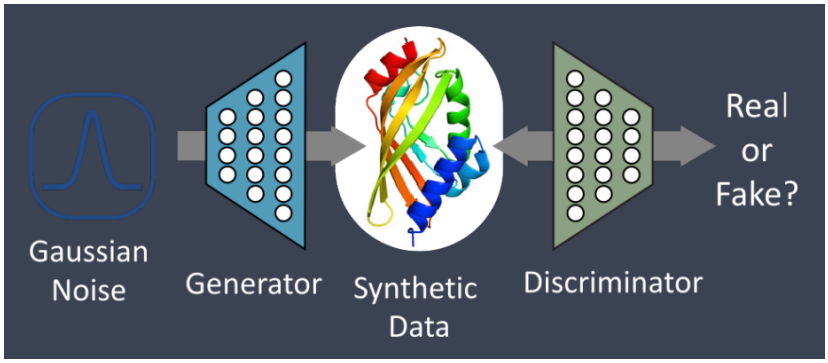

Рис. 4.17. Generative Adversarial Network

Если это работает достаточно быстро, мы можем создавать структуры белков нужной формы. Главное – правильно настроить дискриминатор.

Предсказание структуры белков

Большинство подходов машинного обучения, которые существовали до 2020 года, опирались то, что генерировались попарные контакты, а дальше с помощью молекулярной механики мы пытались восстановить структуру. Однако это не end2end решение, которое позволяет удачно и аккуратно оптимизировать нейтронную сеть так, чтобы получались более точные ответы.

Чем больше информации мы заложим в нейронную сеть, тем лучше для нас, потому что если у нас есть возможность обрабатывать большие массивы данных, то данных должно быть как можно больше. Общая проблема структурной биологии в том, что данных здесь мало. Поэтому, если есть возможность добавить любые данные, которые позволят расширить выборку, это хорошо.

Используется коэволюция, применяются множественные выравнивания. Здесь появляются два основных участника, которые пытаются преодолеть эти проблемы, а именно сделать end2end решение и использовать большие данные, чтобы сделать нейронные сети более богатыми: RaptorX и AlphaFold.

Есть разные **способы для генерации попарных расстояний**. На рис. 4.18 приведён достаточно сложный пример. Но самое важное, что у нас в начале есть последовательность, потом оттуда мы пытаемся выделить 2D features, после чего пытаемся объединить их в общий вектор, из которого можем попытаться построить попарную матрицу расстояний.

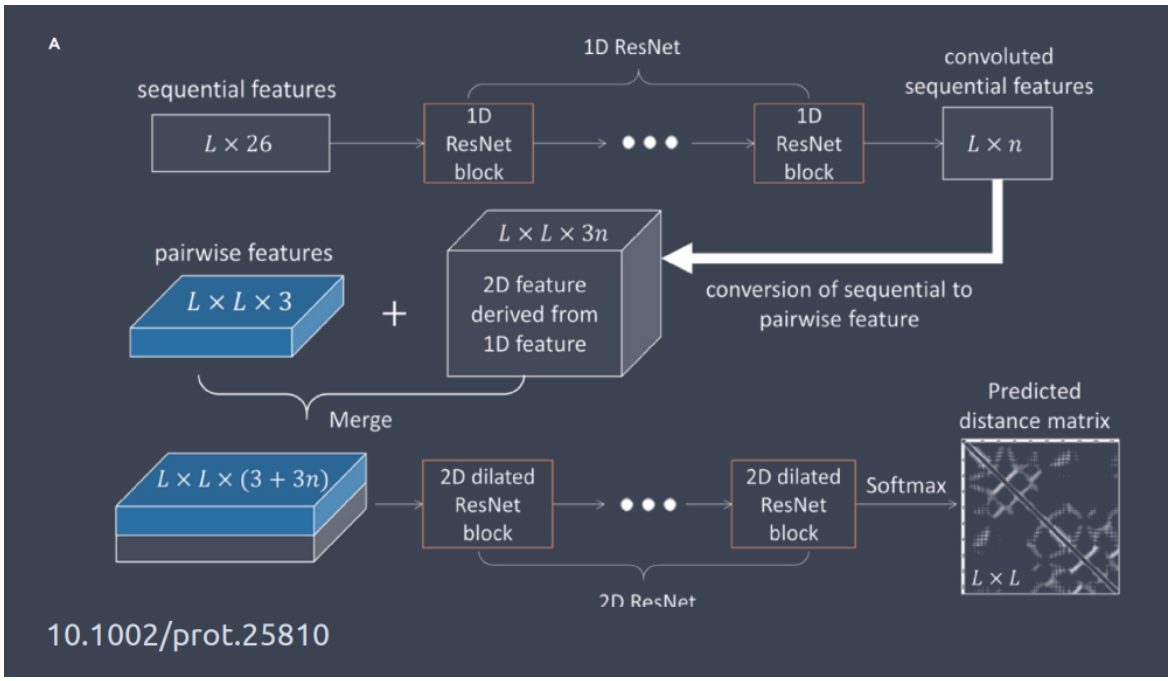

Рис. 4.18. Архитектуры, расстояния

Это не является end2end решением, поэтому обратимся к следующей статье того же автора. Там есть приложение, как реализовать end2end решение (рис. 4.19).

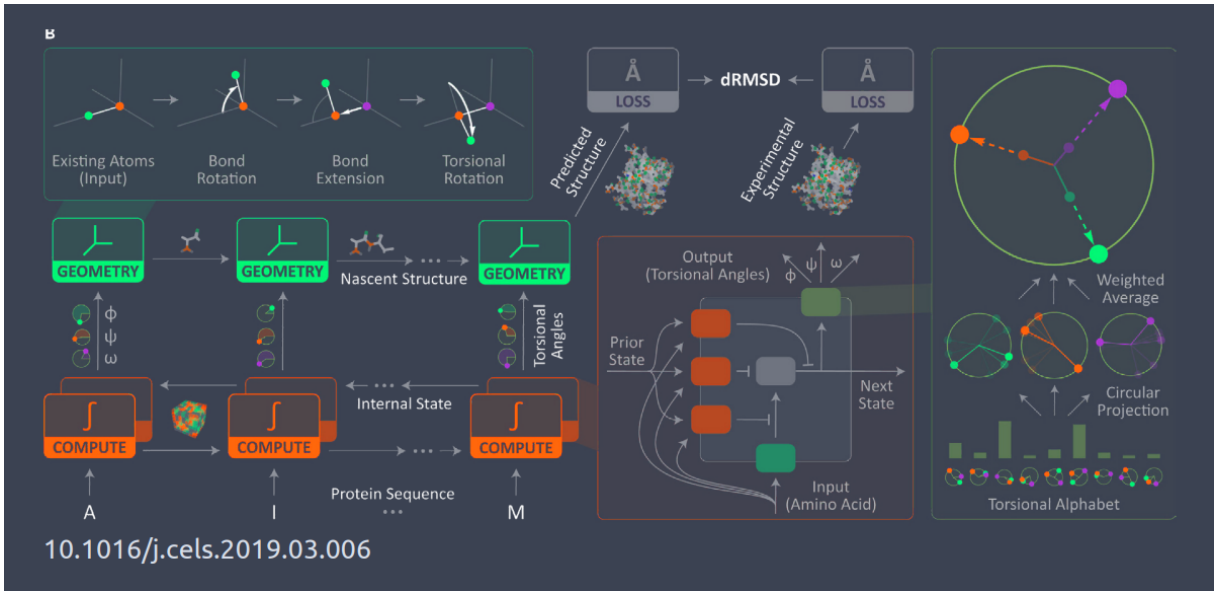

Рис. 4.19. Архитектуры, end2end решение

На сегодняшний день законченного продукта здесь нет, но идея хорошая. Есть наборы углов между атомами, углов между связями, есть углы по ходу цепи, куда всё это расширяется, есть торсионные углы. Имея их, можно пытаться по мере роста цепи генерировать всё больший и больший вектор, который описывает наш белок. Можно строить из структуры вектор и наоборот. Это приводит к тому, что здесь **можно реализовать end2end решение**. Такие архитектуры действительно существуют.

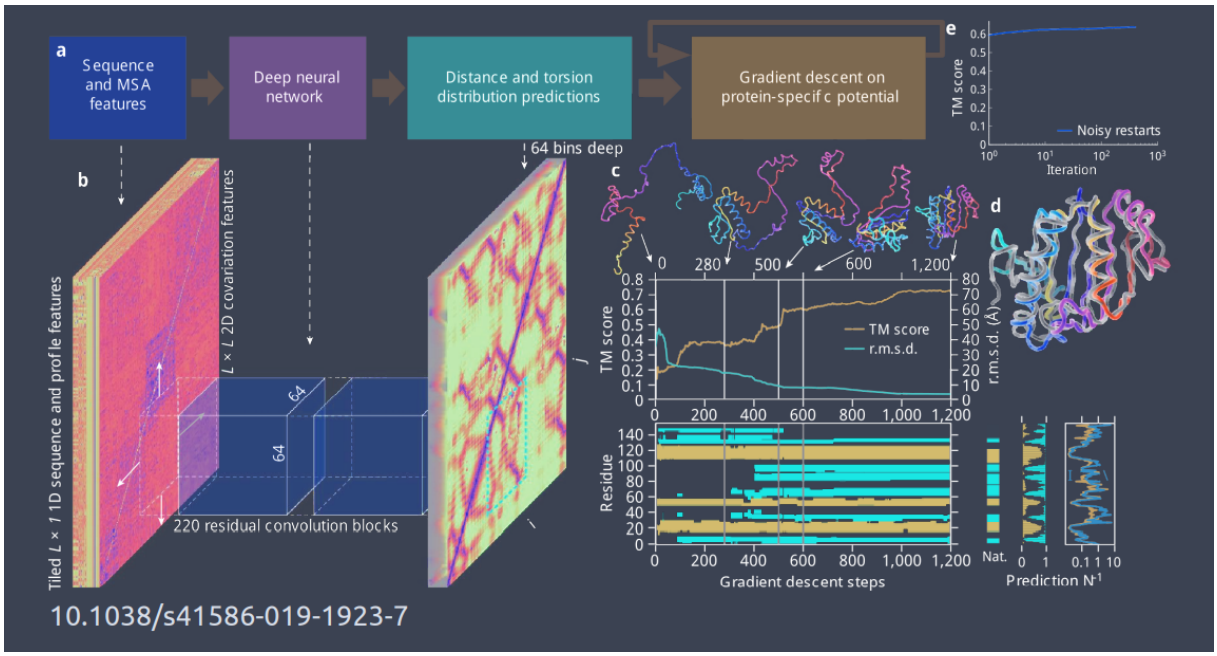

Рис. 4.20. Alphafold 1, идея

В такой же исторической стезе шёл и Alphafold. Сначала был Alphafold 1 (рис. 4.20), он генерировал попарные расстояния между остатками на основе обученной нейронной сети, после чего включался алгоритм градиентного спуска (по сути – молекулярной механики с пружинками), и получалась вторичная структура. Это всё работало хорошо.

Посмотрим, к чему это привело с точки зрения соревнования CASP (рис. 4.21). В тот год он победил, около 60% структур предсказывались правильно, при том, что не было заготовки. Это не являлось end2end решением.

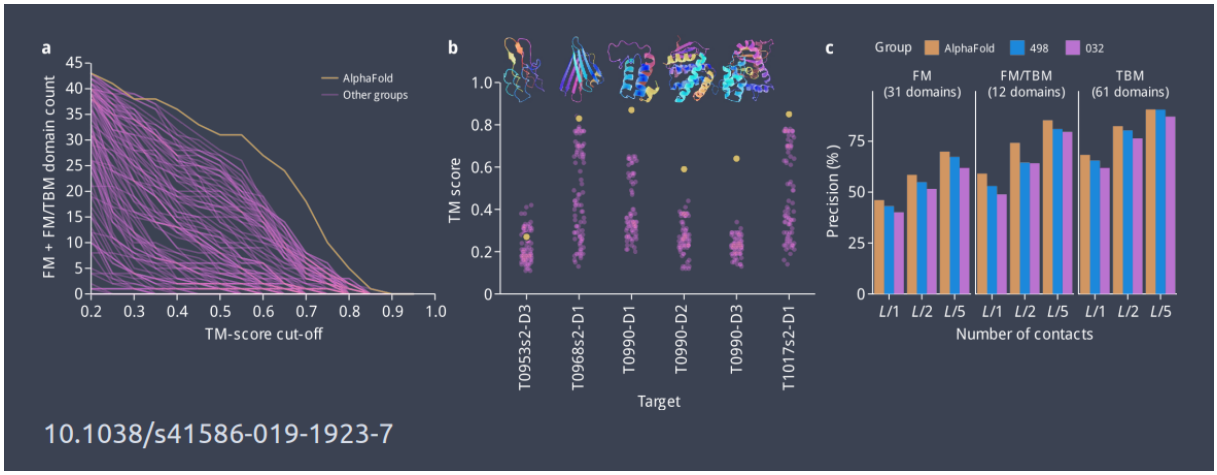

Рис. 4.21. Alphafold 1, CASP

На смену пришёл **Alphafold 2.** Алгоритмы нейронных сетей переоценивают локальные взаимодействия. Это неудивительно, когда есть сближенные атомы, им легко можно поставить высокий score из-за этого, а не из-за того, что они в реальности имеют влияние на всю структуру. Здесь это преодолели. Использовалась множественная редистилляция данных, реализовалось перевзвешивание удалённых взаимодействий. Это привело к тому, что потенциально можно использовать небольшие выравнивания для получения структуры. Дальше была попытка улучшить вес коэволюционных остатков. Это полезно, потому что говорит об однозначном контакте в трёхмерном пространстве.

На рис. 4.22 хорошо видны итеративные действия, связанные с обогащением карты контактов, опираясь на последовательности. Итеративные ходы и редистиллирование приводят к тому, что получается более качественная матрица контактов. Самое важное, что смогли сделать из матрицы попарных контактов и выравнивания можно получить структуру. То есть, это end2end решение, оно хорошо дифференцируется, а именно дифференцируемость по всей цепочке обучения нейронной сети сильно повышает её качество.

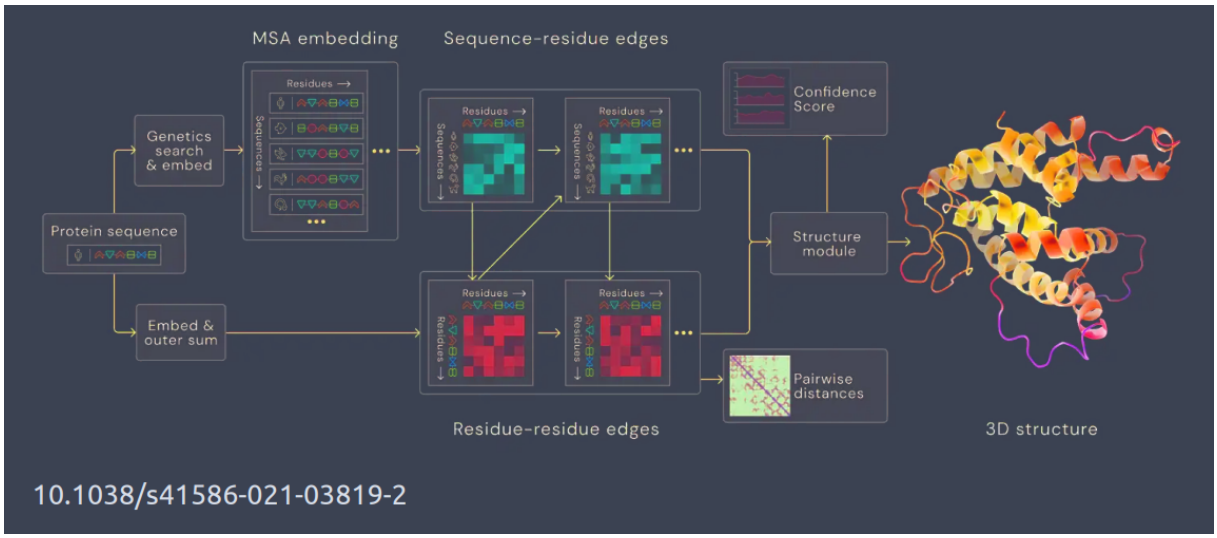

Рис. 4.22. Alphafold 2, метод

Это привело к тому, что Alphafold увеличил количество правильных ответов до 50%, а Alphafold 2 до 85% (рис. 4.23).

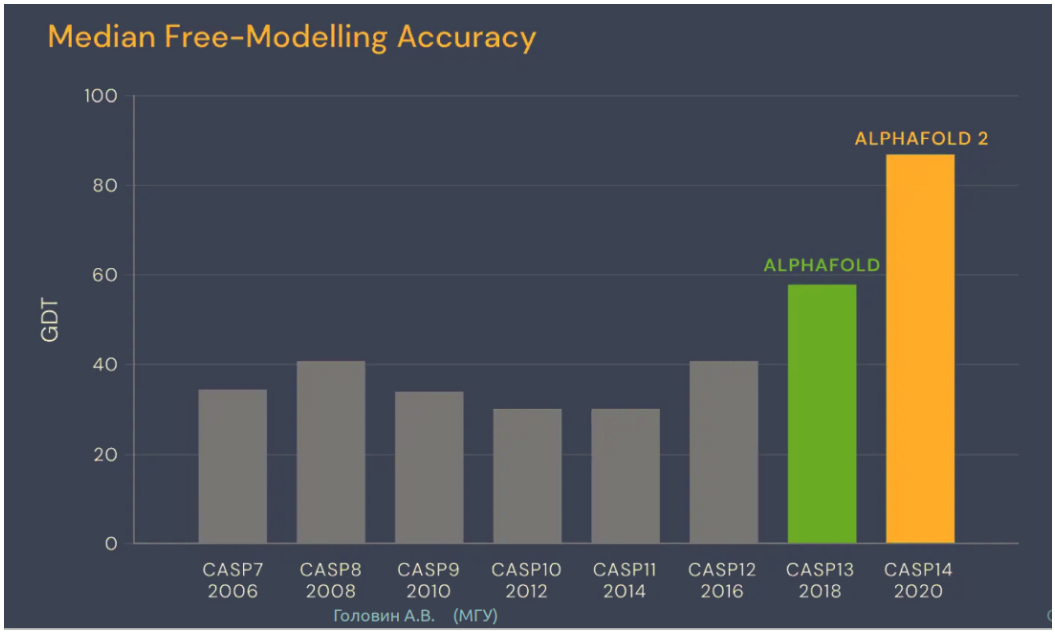

Рис. 4.23. Alphafold 2, результат

В некоторый период значимых улучшений алгоритмики не было: появлялись новые структуры, но они не были похожи на все известные, отсюда падало качество предсказаний. Но потом постепенно стало расти по мере появления методов, связанных с машинным обучением.

Есть и другие группы, которые пытаются разрабатывать подобные алгоритмы. В **trRosetta** похожая идея: трёхмерная структура представляется в виде расстояния и нескольких углов (рис. 4.24).

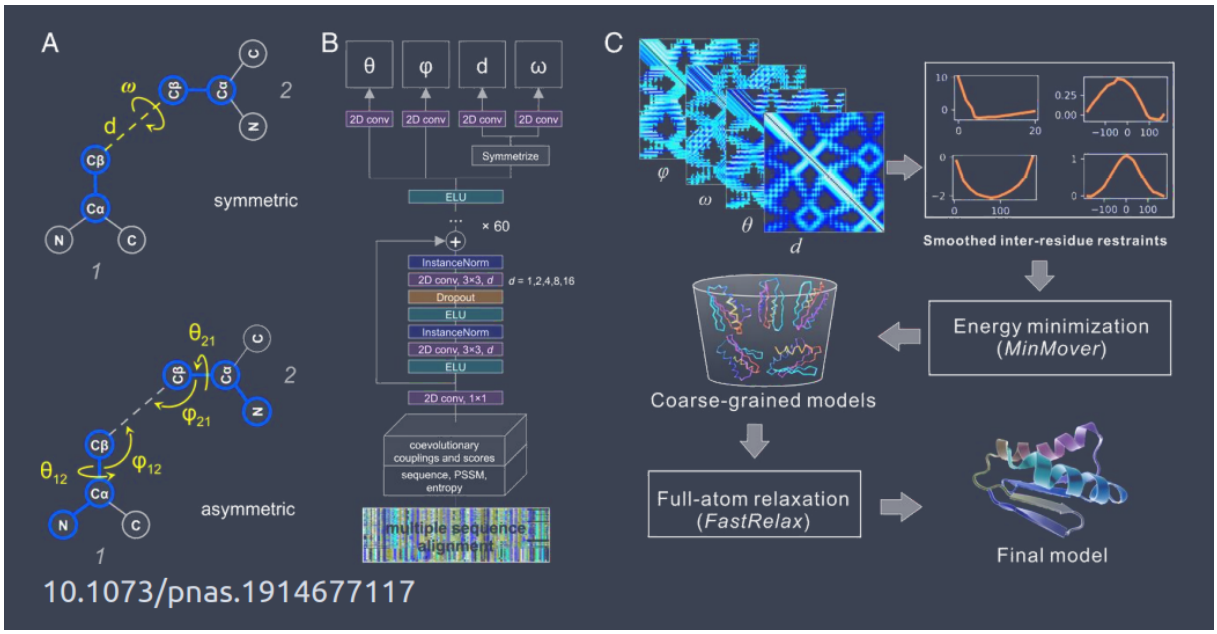

Рис. 4.24. trRosetta, метод

Из этого мы строим конволюционные сети, добавляем информацию о взвешенных матрицах попарных расстояний, пытаемся достать отсюда коэволюционную информацию, scores и т. д. Из этого строятся соответствующие карты. В каждой точке прописываются все потенциалы.

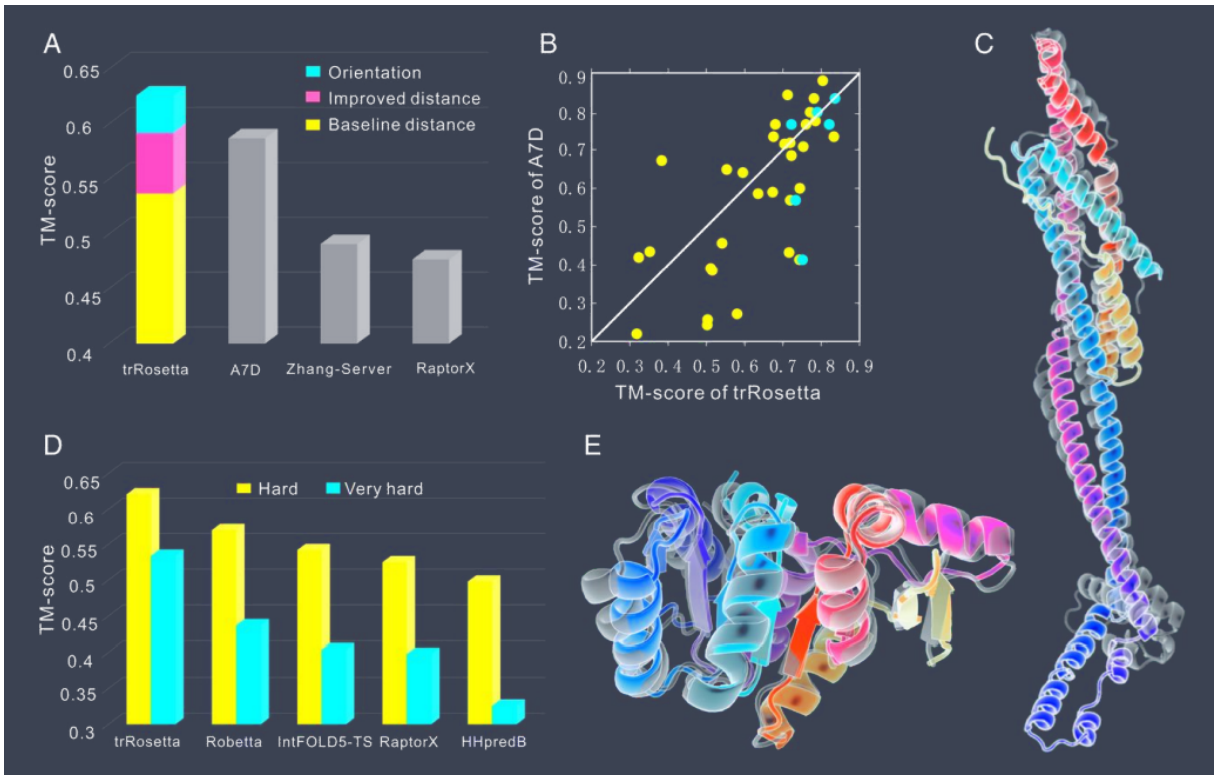

Рис. 4.25. trRosetta, результат

Каждая точка в матрице – это функция, основывающаяся на сплайнах, которые описывают вероятность значения тех или иных углов и расстояний. Дальше применяем традиционные подходы классической молекулярной механики и получаем финальную модель.

В Alphafold берутся не просто попарные расстояния, а вероятности того, что между этими остатками расстояние имеет такое значение. Это позволяет сильно упростить выбор оптимального значения для моделирования.

А в trRosetta решение не выглядит как end2end.

Получается, что результат сравним с Alphafold (рис. 4.25). Для сложных структур достаточно высокий TM-score, явно выше, чем при применении классических механических методов, но пока ещё не слишком революционный.

Но есть нюанс. trRosetta хорошо работает для **предсказания структур искусственных белков,** которые были найдены с помощью Rosetta. Здесь TM-score до 90. Возможно, здесь есть внутренняя обратная связь. Когда в Rosetta получаются белки, они опираются на эмпирически подобранные силовые поля, функции, потом из этого строится новый белок, идёт проверка, может ли он действительно собраться. Оказывается, что набор всех этих функций, особенно в блоке классической молекулярной механики, приводит к тому, что предсказание структур дизайнерских белков лучше, чем белков, которые есть в природе.

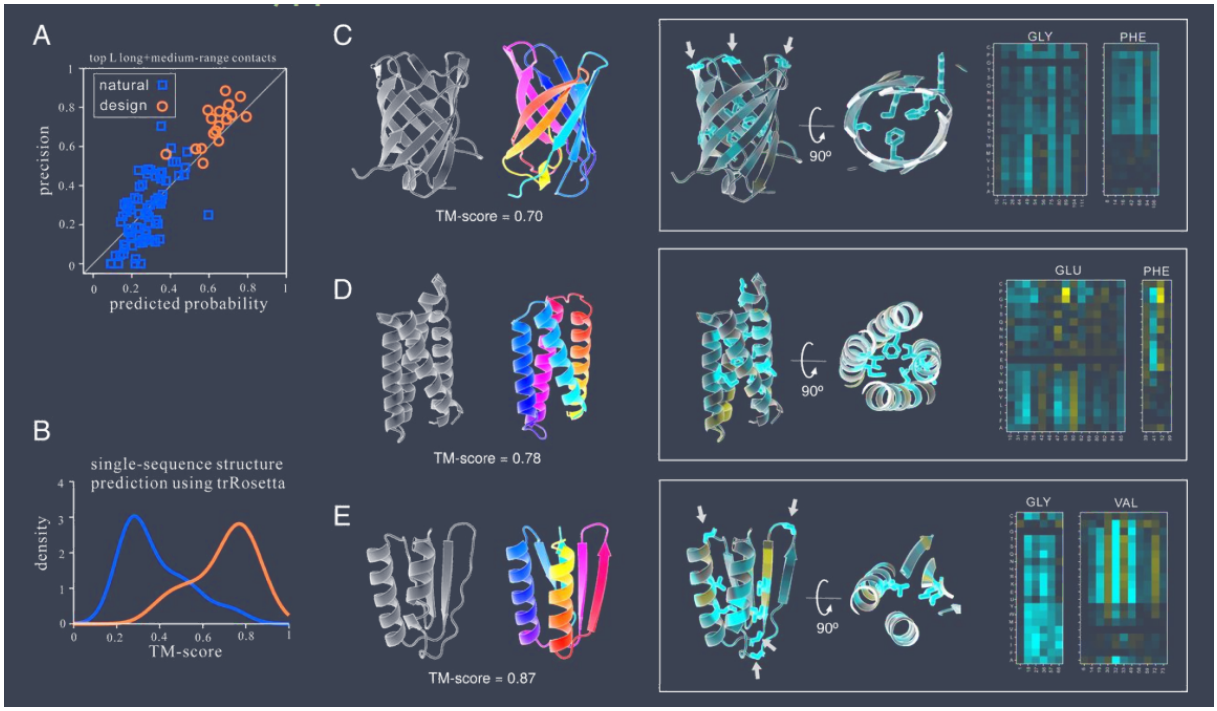

Рис. 4.26. trRosetta, дизайн

Заключение

Сделаем **заключение о перспективах ML** в предсказании структур белков. Сейчас появляется информация, что Alphafold 2 в чём-то хорош, а в чём-то не очень. Так происходит, потому что большинство белков, которые оказываются в базе данных PDB, зачастую в рентгене делаются в комплексе с лигандами. А белок без лиганда, возможно, сложнее кристаллизовать, или это никому не интересно, потому что там нет взаимодействий. И так как есть перевес в PDB, Alphafold в основном предсказывает белки в той форме, которая связана с лигандом.

Все способы предсказания структуры белков, доступные на данный момент, являются сугубо эмпирическими. Они опираются только на те накопленные знания, которые у нас есть, и их достаточно, чтобы предсказывать структуру белков в каком-то приближении.

Надо отметить, что Alphafold чувствителен к одиночным заменам: он выявил изменение конформации при замене в белке. Однако, если эта информация о мутации уже была в PDB, ничего удивительного в этом нет.

Чем больше информации мы используем, тем точнее модель, поэтому было бы интересно добавлять в Alphafold собственные дистанционные ограничения и другие вещи, которые позволили бы эффективно манипулировать сборкой структуры.

Большинство методов имеет свои недостатки. Например, Alphafold 2 не может работать с мультицепочечными молекулами. Здесь критический анализ позволяет избежать заведомо глупых ошибок и улучшить модель.

Лекция 5. Белок-белковые взаимодействия

Interactome – это набор всех взаимодействий между белками в отдельно взятом организме. Давно замечено, что Interactome более развитых организмов гораздо разнообразнее, чем у простых. Например, если у прокариот белок-белковые контакты встречаются в конкретных случаях, например, один белок взаимодействует с одним-двумя партнёрами, то белки человека взаимодействуют со многими партнёрами, и эти взаимодействия часто реализуют явные регуляторные эффекты. В том числе поэтому у человеческих белков очень развито разнообразие петлевых участков. Когда мы начинаем строить по гомологии модель человеческих белков, то видим, это эти белки имеют большое количество вставок в неструктурированных участках. Это объясняет тот факт, что с помощью неструктурированных и частично структурированных участков реализуется большое количество межбелковых взаимодействий в развитых эукариотических клетках.

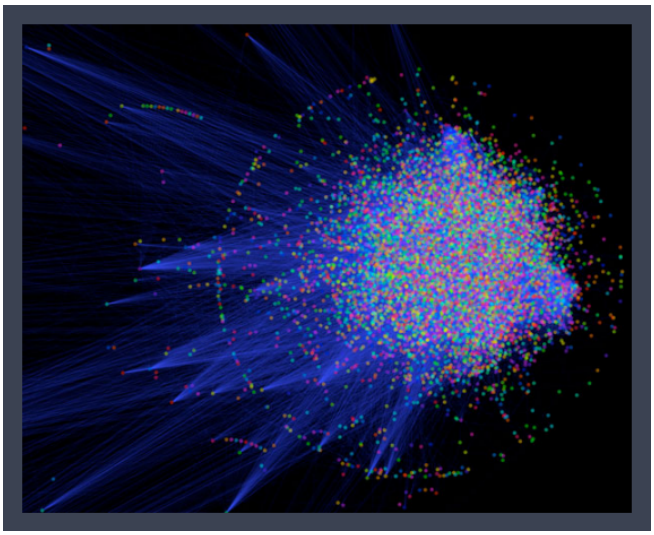

Рис. 5.1. Human «Interactome»

Рассмотрим **особенности белок-белковых взаимодействий.** Когда два белка взаимодействуют, площадь контакта может быть очень большой, 1000-2000 A^2, то есть большая область десольватации. Десольватация может быть выгодна, если поверхность гидрофобная. Поэтому белки будут слипаться частично с использованием гидрофобного эффекта.

Отсюда вытекает, что только 5% остатков дают ключевой вклад в связывание. Гидрофобные остатки неинвариантны, они могут быть очень разными, но гидрофобный эффект всё равно будет реализовываться. Изменение природы одного остатка не должно сильно влиять на энергию взаимодействия. А вот изолированные водородные связи – те, к которым нет доступа воды, на неё влияют активно.

У нас есть окружение, которое создаёт белок-белковый контакт, и есть большое количество остатков, которые реализуют его взаимодействие. Очевидно, что они дают основной вклад в энергию и специфику. Модификация таких контактов и способов

взаимодействия белков достаточно затруднена экспериментально, потому что их сложно вычленить, а ещё исследование белок-белковых контактов без рентгеноструктурного анализа будет иметь много побочных эффектов. Один из них – концепция молекулярного краудинга, когда мы считаем, что в клетке очень много молекул, а свободной воды как таковой мало, и гидрофобный эффект там гораздо менее выражен. Поэтому попытка прокоррелировать результаты взаимодействия двух белков в пробирке с взаимодействием в клетке может быть заведомо ошибочной.

Белок-белковые взаимодействия очень разнообразны, никаких общих правил нет. А учитывая, что в это может вписываться динамика структуры белка, всё ещё больше усложняется.

Как можно **исследовать белок-белковые контакты**, опираясь на накопленные знания? Одним из таких способов является исследование ко-эволюции. Очевидно, что взаимодействующие белки должны определённым образом коэволюционировать. На прошлой лекции упоминался AlphaFold, который на идее коэволюции научился тренировать нейронные сети так, что они стали хорошо предсказывать структуры. Также это может реализовываться без машинного обучения и структурных данных, например, через поиск пар белковых семейств, которые синхронно эволюционируют. Всё это можно описать с помощью численных методов и использовать для машинного обучения.

Есть предсказание на основе подобия филогенетических деревьев. Филогенетические деревья для отдельных белков могут отличаться от филогенетического дерева для всего организма.

Методы на основе классификации или поиска гомологичных мест контакта имеют невысокую степень точности, потому что если две последовательности близки по составу, то это не всегда будет фактом того, что они в пространстве существуют с одинаковым интерфейсом. Также можно использовать сравнение профилей, мотивов и др., что можно взять из последовательности.

В каких-то случаях это работает, и можно сказать, что этот паттерн взаимодействует с другим паттерном белка. Но даже на примере антител, которые узнают свои антигены, можно увидеть, что нередко они делают это с помощью четырёх петель, которые разнесены в последовательности, но в пространстве сближены. И соотнести последовательность этих петель с последовательностью мишеней нередко бывает достаточно сложно.

Можно попытаться накладывать всё это на структурные данные, пытаться строить библиотеки контактов и из этих библиотек пытаться интерпретировать, может ли эта часть белка с чем-то взаимодействовать.

Методы Байеса используются для того, чтобы вычленить известные данные о том, как изменяется ко-экспрессия белков на данных чип-секвенирования. Здесь высокий уровень шума, и биоинформатика работает только на уровне данных конкретного эксперимента и обучения.

Можно попытаться моделировать комплексы, но это сложная задача. Учитывая разнообразие контактов, которое у нас есть, есть сомнения, что, например, применение молекулярной динамики и её модификаций для сканирования всех возможных взаимодействий данного белка с другими возможно по ресурсам.

Третий вариант – пытаться делать макромолекулярный докинг.

Баз данных достаточно много, они постоянно обновляются. Ниже приведён немного устаревший список, но, тем не менее, он может быть полезен.

String – база данных экспериментальных и предсказанных взаимодействий; отличная графика; http://string-db.org/ IntAct - база данных на основе литературных данных или прямая информация от авторов. http://www.ebi.ac.uk/intact/ iHOP - Информация слинкованая с другими белками. Построена на основе литературных данных. Представление в виде кусочков текста. http://www.ihop-net.org/ BioGRID - Источники: литература и результаты high-throughput экспериментов; http://thebiogrid.org/ MIPS Mammalian Protein-Protein Interaction Database, не работает:). http://mips.helmholtz-muenchen.de/proj/ppi

Макромолекулярный докинг

Перейдём к макромолекулярному докингу. Это процедура, чтобы найти, как **оптимально разместить один белок рядом с другим.** Есть два участника процесса. Мы объявляем центр масс большого белка, который называем рецептором, точкой старта сферических координат (рис. 5.2). Дальше будем изменять положение и ориентацию второго белка относительно первого.

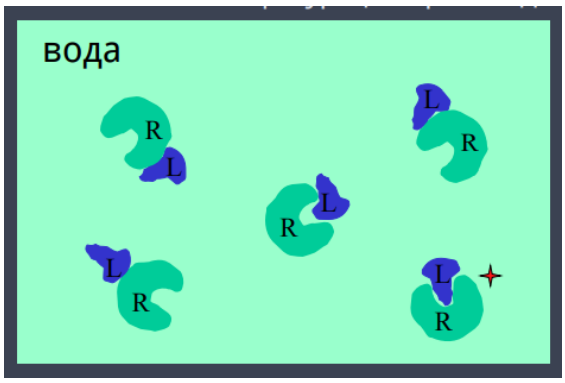

Рис. 5.2. Макромолекулярный докинг

В лекции, связанной с молекулярным докингом, мы пытались воспроизвести поверхность белка в виде решёточных моделей, в виде GRID, в котором проводилось Монте-Карло моделирование динамики низкомолекулярного лиганда. Здесь было бы правильно использовать тот же самый подход, потому что наш объект – жёсткое тело, и в данном случае вряд ли хватит ресурсов, чтобы моделировать молекулярную динамику во время связывания белка. Докинг почти всегда делается, подразумевая, что белок-белковая молекула неподвижна.

Для данной задачи достаточно быстро было найдено хорошее решение – **совмещение решёток,** которые представляют белки, то есть трёхмерное GRID-представление белка (рис. 5.3). С помощью быстрых преобразований Фурье оттуда вылавливались вектор, смещение и трансляционная матрица, которые должны быть для того, чтобы описать смещение белка лиганда относительно своего стартового состояния, чтобы попасть в тот белок, который мы объявили рецептором и центром сферических координат.

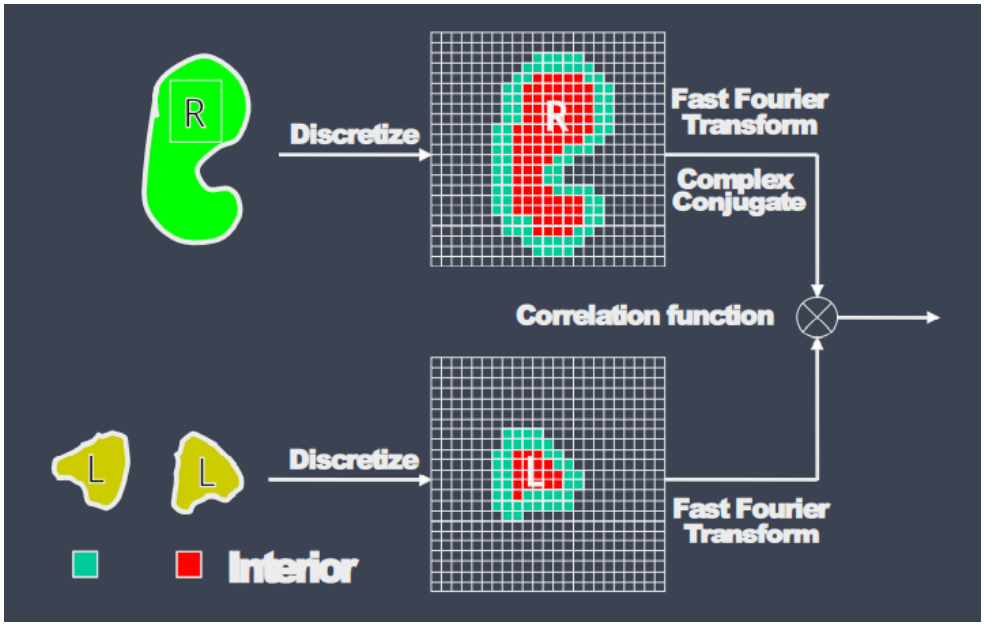

Рис. 5.3. Совмещение решёток

Надо отметить, что у этого белка должны быть две разных области в 3D решётке: внутренняя область и поверхностная область (отмечены на рис. 5.3. красным и зелёным цветом). Идеальным является, когда есть максимальное пересечение зелёных областей, а самым худшим – когда есть пересечение красных, потому что белки не должны входить друг в друга.

Из этого мы получаем матрицы трансляции и вращения, которые говорят о том, как надо расположить один белок относительно другого, чтобы получить умеренно хороший комплекс.

Теперь посмотрим, какие есть численные оценки, чтобы говорить, что хорошо или плохо. Подобный подход будет сильно зависеть от того, насколько эффективно мы сканируем пространство углов, расстояний и т. д. Здесь у нас есть два основных параметра: Success Rate и Hit Count. Success Rate – то значение, которое указывает, что для данного количества предсказаний для каждого комплекса, для какого процента комплексов был найден хотя бы один правильный ответ. То есть берём базу белок-белковых комплексов, для каждого из комплексов делаем по 100, а то и 1000 предсказаний, и говорим, как зависит факт того, что у нас есть хотя бы один правильный ответ, от количества предсказаний по всем белковым комплексам. А в Hit Count мы

учитываем, сколько правильных предсказаний есть в результатах. Но какие предсказания правильные, из численного анализа не очевидно.

Посмотрим на **зависимость Success Rate от шага дискретизации угла**: у нас сферические координаты, сканируем по углам (рис. 5.4). Чем больше считаем, тем меньше зависит от угла и больше от количества предсказаний. Это неудивительно, ведь если угол меняется на большой шаг, всё равно пройдёмся по всем координатам несколько раз.

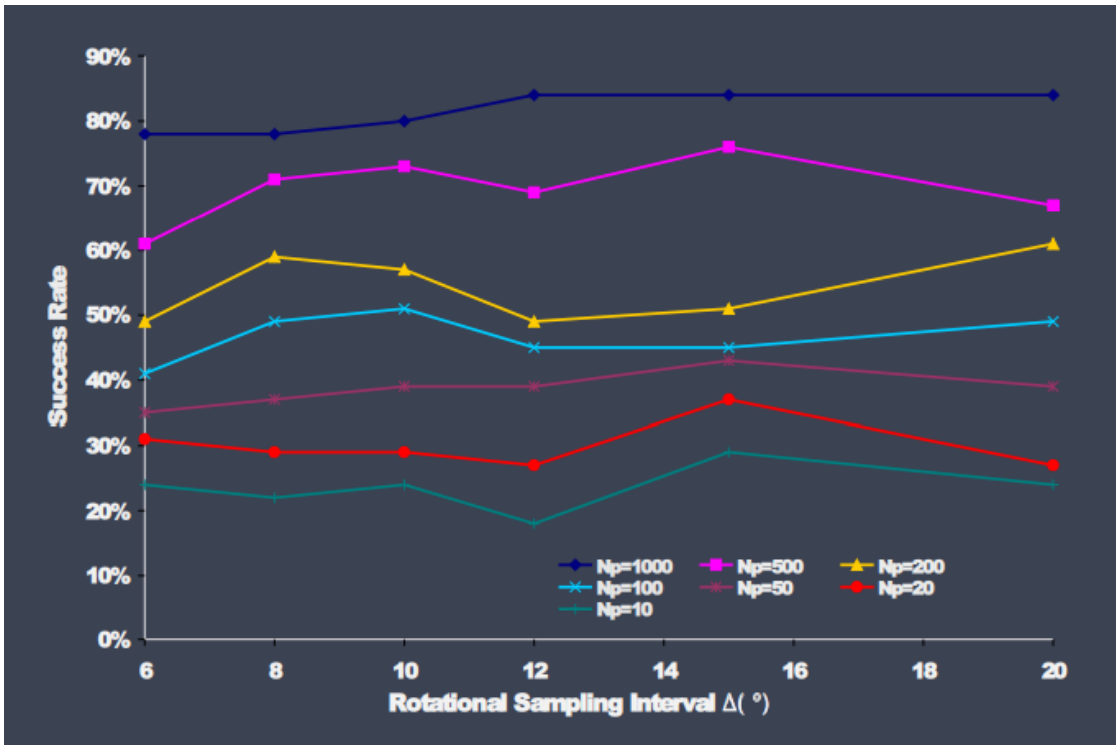

Рис. 5.4. Зависимость Success Rate от шага вращения

А вот в Hit Count количество правильных ответов сильно повышается, если делать сканирование аккуратно: тогда оно растёт почти линейно с ростом количества предсказаний (рис. 5.5.).

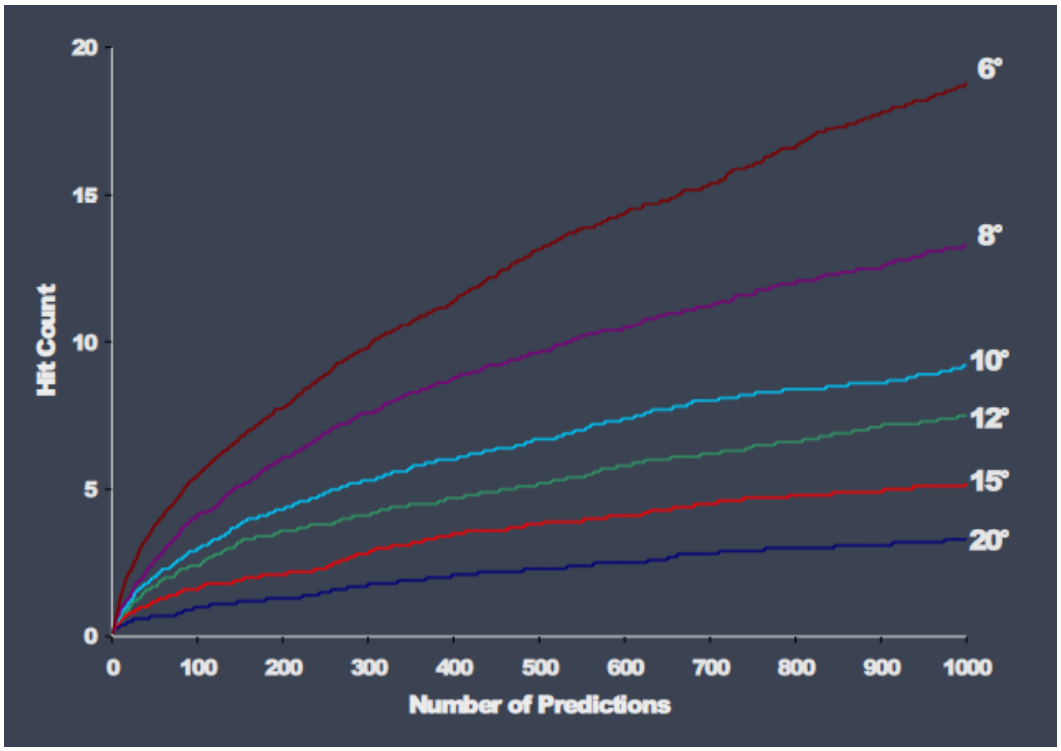

Рис. 5.5. Зависимость Hit count от шага вращения

Как эти решётки совмещаются? Маленькая решётка крутится относительно большой. Самый простой пример реализации совмещения решёток – с помощью правила, что в решётке есть ячейки, которые при пересечении 1 и 1 будут давать какое-то положительное число. Внутри белых шариков на рис. 5.6 тоже написано 9i. Когда пересекается 1 и 9i, при перемножении получается комплексное число, а когда пересекаются 9i и 9i (серые шарики), перемножение даст отрицательное число и сильное падение score. С помощью такой примитивной ранжировки ориентации можно добиться неплохих результатов.

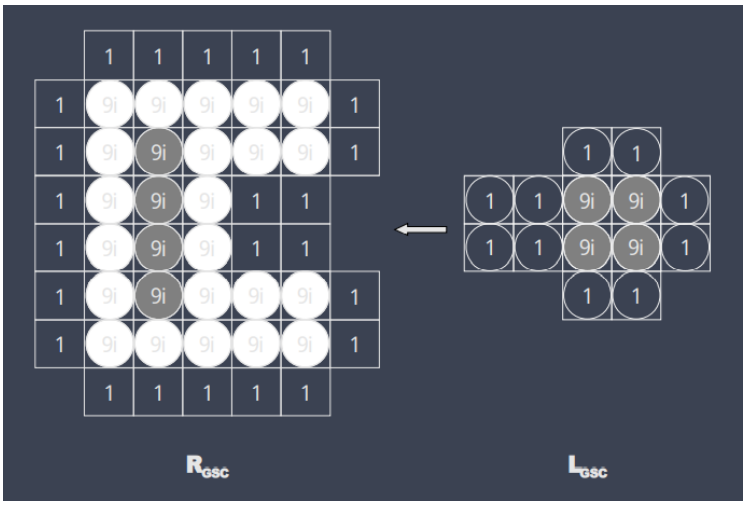

Рис. 5.6. Решеточная комплементарность поверхности

Можно сделать это более аккуратно и учесть ещё и окружение, потому что когда попала одна точка на другую – хорошо, но ещё лучше, если при этом образуется много контактов.

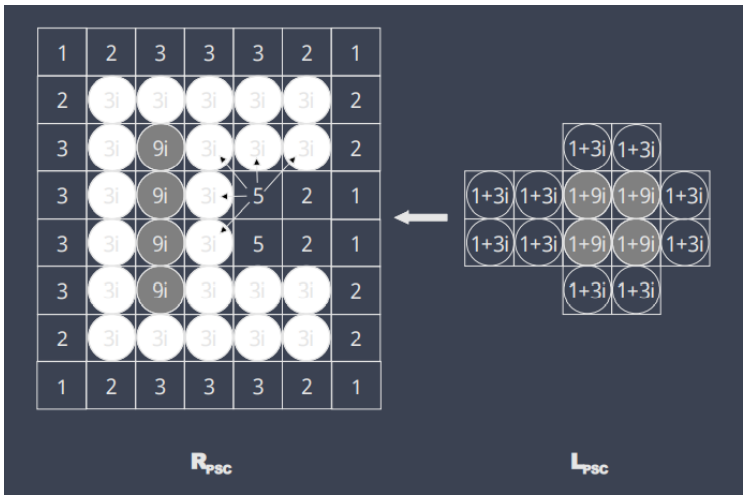

Рис. 5.7. Парная комплементарность поверхности

На рис. 5.7 это реализуется с помощью того же алгоритма, но более продвинутого. Здесь парная комплементарность поверхности, когда есть числа, которые находятся в решётке, принадлежат, допустим, cavity и имеют более высокое значение. Если белок попадает хвостиком или петлёй в это cavity, то получается кумулятивный эффект, потому что мы суммируем ещё и значения из белого окружения, которое находится рядом с этим. Таким образом, мы сильно увеличиваем score. Оптимальное положение находится с помощью фурье-преобразования, когда накапливается определённое количество данных.

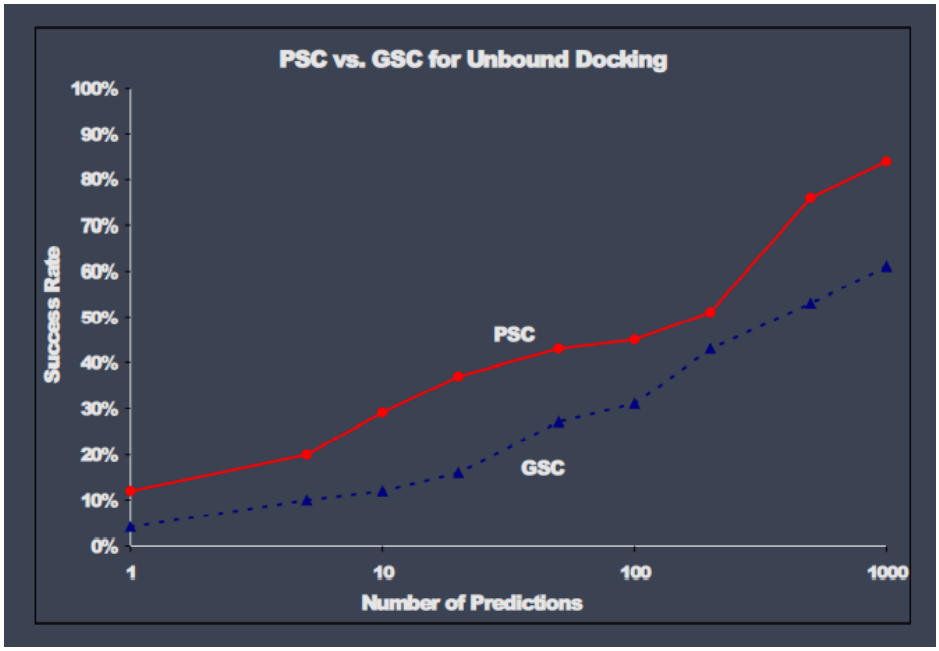

Рис. 5.8. PSC vs. GSC и Success Rate

На рис. 5.8 видно, что в целом при 1000 предсказаний парная комплементарность всегда выигрывает в Success Rate. Практически в 90% случаев мы можем найти хотя бы 1 правильный ответ среди 1000 предсказаний.

Понятно, почему так реализуется. В природе есть гидрофобная поверхность, контактирующие остатки. И если всё это сконцентрировано в одном месте пространства на поверхности, это гораздо более выгодно, чем если есть множество одинарных контактов, которые разделены водой, пространством или чем-то ещё (рис. 5.9). Поэтому первый тип предсказаний даёт больше правильных ответов, чем второй (примерно 85 против 60).

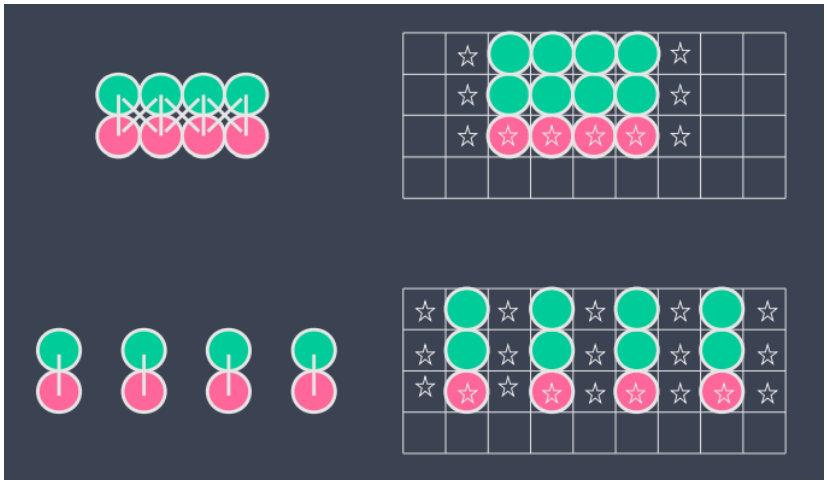

Рис. 5.9. Объяснение двух типов предсказаний

После того, как мы совместили решётки, мы имеем матрицу вращений и смещений. И мы можем применить эту матрицу к белку-лиганду, который был исходно, то есть все его атомы сместить и повернуть в пространстве. А дальше на основе простоых методов молекулярной механики пытаться **посчитать энергию**, которая у нас есть. Для этого в программе ZDoc используется стандартное силовое поле, в котором прописаны заряды, радиусы Ван-дер-Ваальса и т. д. Отсюда рассчитывается энергия Ван-дер-Ваальса, причём часто с поправками, потому что, когда мы хорошо расположили белки, можно ожидать, что аминокислоты могут локально подвинуться, и не будет сильных клэшей. Поэтому можно ожидать, что у находящихся на поверхности аминокислот будут поправки радиуса Ван-дер-Ваальса, чтобы энергия отталкивания ($\Delta E_{V\,dW}$) не была запредельной, если два остатка неожиданно попали в одно и то же место пространства.

Гидрофобика оценивается с помощью энергии десольватации (ΔG_{desol}), потому что биополимеры находятся в воде, и чтобы белок стал контактировать с белком надо, чтобы он перестал контактировать с водой, и для этого нужна энергия. Так рассчитывать гидрофобный эффект не очень честно, потому что он не всегда опирается только на энергию десольватации: десольватация связана с энтальпией, которую можно посчитать как количество энергии, необходимой для отдирания воды от белка, а энтропийный

эффект - количество воды, которое теряет свою подвижность при контакте с этой поверхностью белка.

В электростатике (ΔE_{el}) мы берём заряды силового поля и считаем кулоновское взаимодействие.

Ещё есть константа, которая описывает изменение вращательных и поступательных энтропий (ΔG_{const}), которые теряет молекула белка, образуя комплекс.

$$\Delta G = \Delta E_{V\,dW} + \Delta E_{el} + \Delta G_{desol} + \Delta G_{const}, \tag{5.1}$$

$$\Delta G_{desol} = \sum_i \sum_j N_{ij} \Delta G_{ij} \tag{5.2}$$

Влияние каждой из компонент существенное. Но, если много предсказаний, большого эффекта от дополнительного расчёта ΔG по отношению к score, поступающим от решётки, нет (рис. 5.10). С другой стороны, на Hit Count это влияет (рис. 5.11), потому что Success Rate – это факт наличия хотя бы одного правильного ответа, а это почти всегда так, если сделать много расчётов. Однако надо ещё понять, как вычленить этот правильный результат. А вот Hit Count может сильно увеличиться, если мы начинаем применять и рассчитывать энергии, учитывая при этом десольватацию и электростатику.

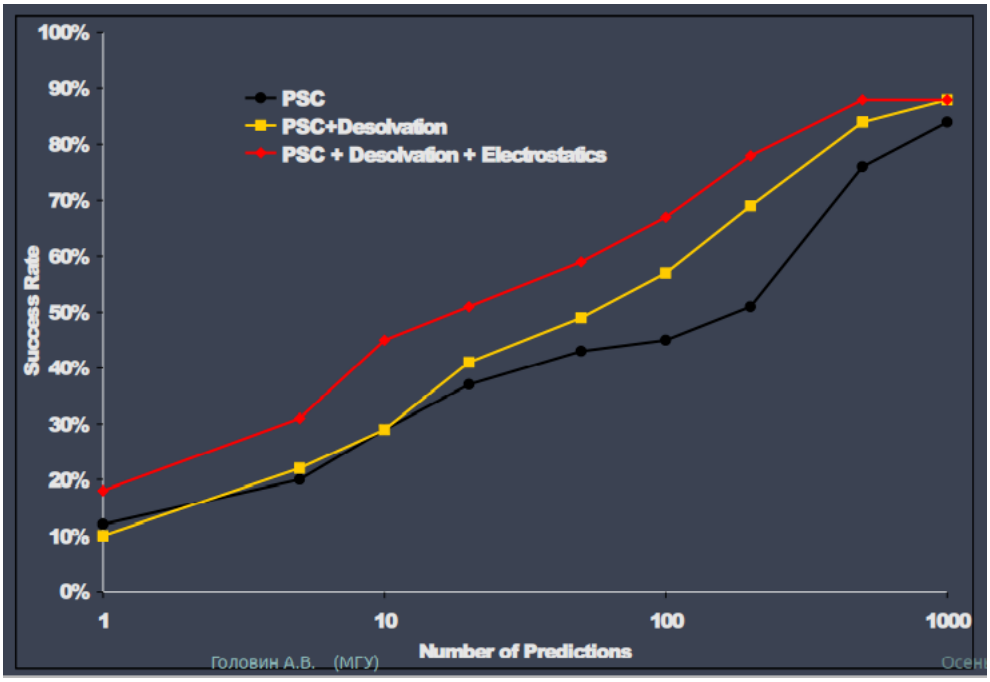

Рис. 5.10. Влияние на Hit Count

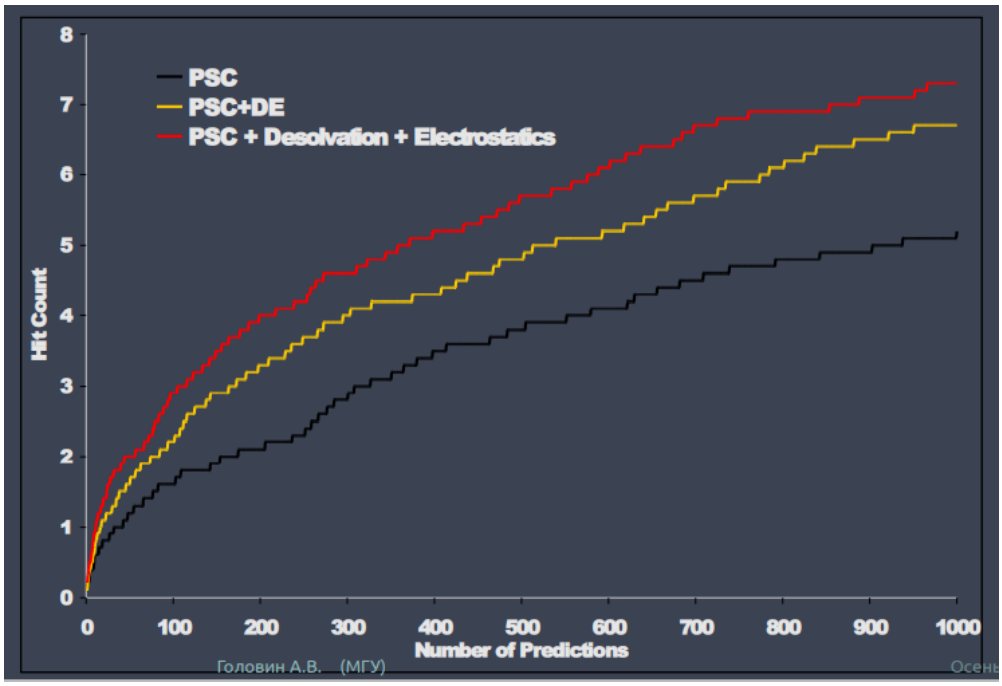

Рис. 5.11. Влияние на Hit Count

Rosetta

Альтернативным классическим подходом для определения способа взаимодействия двух белков является **алгоритм Rosettadock** (рис. 5.12), который является прямой реализацией докинга по отношению к белкам. Он вычислительно затратный, но зато позволяет учитывать подвижность боковых радикалов, а в одной из модификаций – и подвижность остова.

В начале есть случайное расположение, потом мы грубо совмещаем два белка, и если это совмещение даёт чем-то выгодную поверхность контакта, на основе фильтров переходим к докингу с высоким разрешением, тогда мы начинаем хорошо упаковывать боковые радикалы и после кластеризации получаем результат.

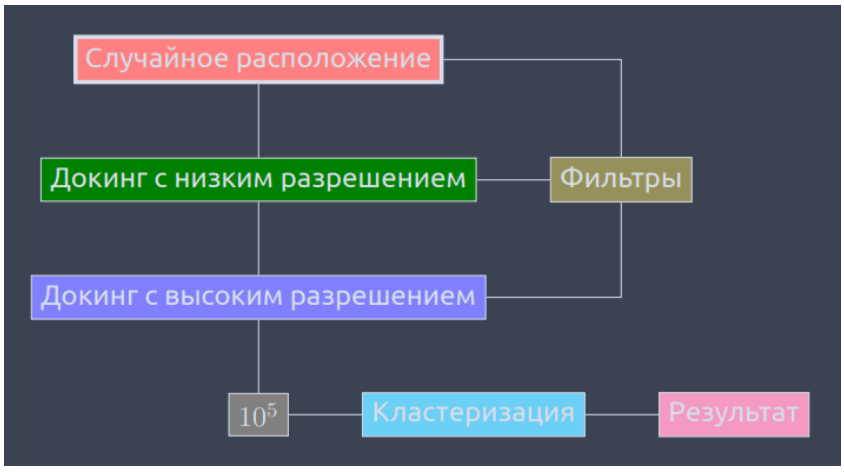

Рис. 5.12. Алгоритм Rosettadock

Это работает, но надо понимать, что **низкое разрешение** – это когда есть остов в явном виде, а боковые радикалы представлены в виде одной частицы (рис. 5.13). Эта частица находится на каком-то расстоянии, она неподвижна, у неё есть какой-то заряд, и тем самым делается общее представление о боковом радикале. Оно оптимизировано под то, чтобы у нас получались правильные результаты, но никогда не смещается: точка находится на относительно константном месте от остова. Счёт становится гораздо быстрее, потому что становится мало переменных, но мы таким образом просто пытаемся имитировать физическую диффузию одного белка относительно другого.

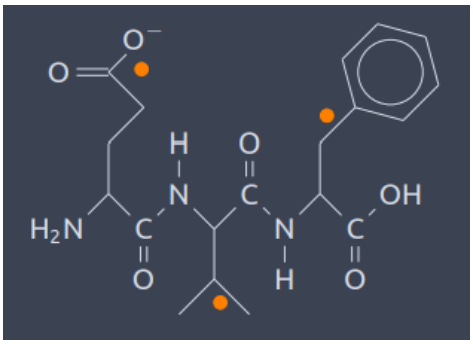

Рис. 5.13. Поиск с низким разрешением

После того, как нашли хорошие результаты в докинге с низким разрешением, можно переходить в докинг с высоким разрешением. Это означает, что мы должны восстановить боковые радикалы из библиотеки ротамеров. Дальше их надо оптимизировать так, чтобы они хорошо располагались и взаимодействовали между собой. Это делается стандартным методом молекулярной механики, итеративные шаги зачастую делаются с помощью Монте-Карло и оптимизации геометрии. Мы добиваемся того, чтобы у энергии, которая опирается на молекулярную механику, значение максимизировалось, то есть становилось всё более отрицательным. Дальше используется циклическая оптимизация положения белка, небольшое смещение и прочие движения. Это похоже на эмпирически найденный путь, но других путей в Rosettsa нет.

Перейдём к интерпретации результатов (рис. 5.14). Если мы получаем очень много результатов, например, 10^5 конформаций комплекса белков, полезно построить фунал. Фунал – это когда есть некое место, в котором мы строим зависимость энергии комплекса от геометрического параметра, описывающего взаимное расположение лигандов или лиганда и рецептора и видим, что при каких-то значениях геометрии у нас есть некое количество точек, которые становятся всё меньше и меньше по энергии. Это создаёт воронку почти оптимальных расстояний и показывает одно из лучших.

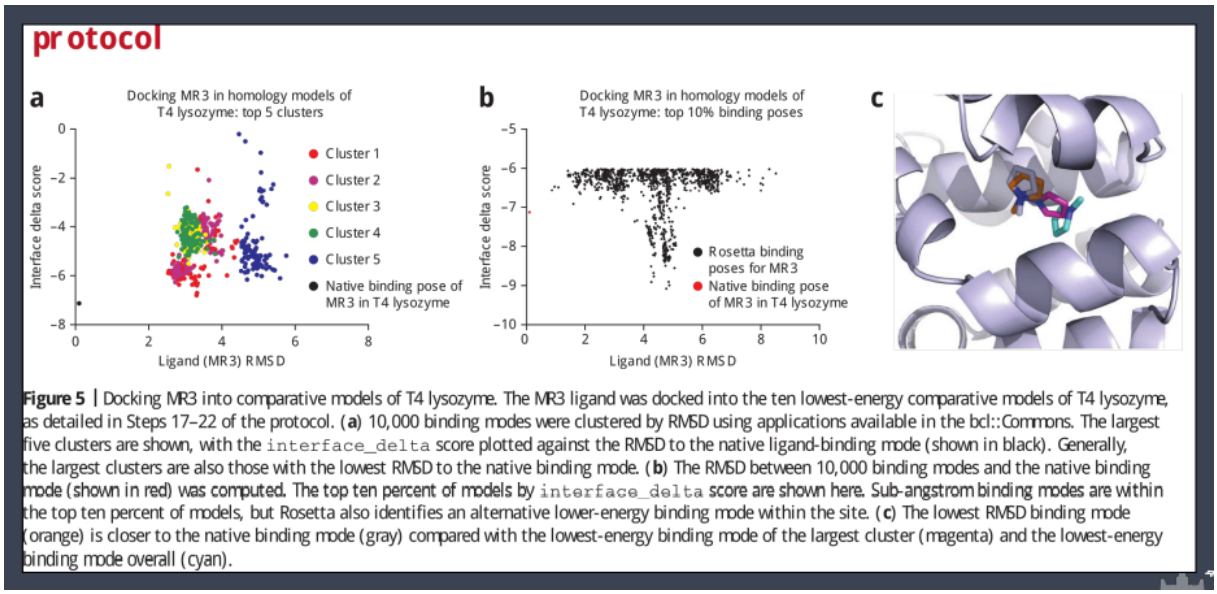

Figure 5 | Docking MR3 into comparative models of T4 lysozyme. The MR3 ligand was docked into the ten lowest-energy comparative models of T4 lysozyme, as detailed in Steps 17–22 of the protocol. (**a**) 10,000 binding modes were clustered by RMSD using applications available in the bcl::Commons. The largest five clusters are shown, with the `interface_delta` score plotted against the RMSD to the native ligand-binding mode (shown in black). Generally, the largest clusters are also those with the lowest RMSD to the native binding mode. (**b**) The RMSD between 10,000 binding modes and the native binding mode (shown in red) was computed. The top ten percent of models by `interface_delta` score are shown here. Sub-angstrom binding modes are within the top ten percent of models, but Rosetta also identifies an alternative lower-energy binding mode within the site. (**c**) The lowest RMSD binding mode (orange) is closer to the native binding mode (gray) compared with the lowest-energy binding mode of the largest cluster (magenta) and the lowest-energy binding mode overall (cyan).

Рис. 5.14. Интерпретация результатов

Но здесь зачастую желательно иметь какой-то контроль или хотя бы дополнительную информацию, потому что на рисунке видно, что положение небольшого лиганда в этом эксперименте сильно отличалось от положения фунала. Поэтому данный метод не является полностью самостоятельным.

Дополнительной информацией может быть любая информация о взаимодействии двух белков. Её сейчас очень легко получить экспериментально. Например, сшив белки глутаровым альдегидом и отнеся их на масс-спектр, можно узнать, в каком месте произошла сшивка. Значит, можно ожидать, что эти места в комплексе сближены. Эта информация позволяет строить очень качественные модели белковых комплексов.

ML походы к PPI

Апофеозом данного подхода является программа **HADDOCK**, которая позволяет на основе дистанционных ограничений, полученных из экспериментальных методов, качественно строить комплексы, удовлетворяющие этим ограничениям (рис. 5.15).

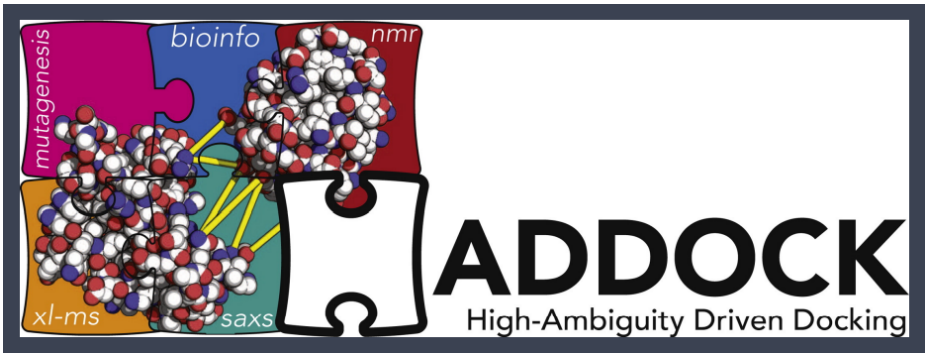

Рис. 5.15. HADDOCK

Перейдём к **машинным методам** в данной области. Методы поиска белок-белковых контактов с помощью анализа последовательностей использовались, и на них тренировались нейронные сети, с 2001 года. Типы представлений, использовавшихся для обучения нейросетей, были совершенно разные: аминокислотный состав, слова, которые можно изъять из последовательности, доменный состав, мотивы, профили гидрофобности, генетические и филогенетические особенности и т. д. Естественно, что не всё это давало правильные ответы.

В основном использовались нейтронные сети, Баесовские методы, SVM, RF и даже простая кластеризация. Всё работало, но было сделано заключение, что все эти представления не могут полностью охватить динамически сложные явления, происходящие при образовании белок-белковых комплексов. А также сложно однозначно сказать, где true positive ответы, а где false positive, аналогично для negative. Без дополнительной информации это работало не очень хорошо.

Первым хорошим шагом в данном направлении была попытка научить нейронную сеть определять **hot spots** (рис. 5.16). Это именно то небольшое количество остатков, которые реально отвечают за эффективное связывание рецептора с лигандом. Какие дескрипторы здесь могут быть использованы? Структура, энергии связывания, а также всё, что касается эволюционно накопленного материала. Все эти описания использовались для того, чтобы строить модели по предсказанию, и на текущий момент с переменной эффективностью такой подход работает.

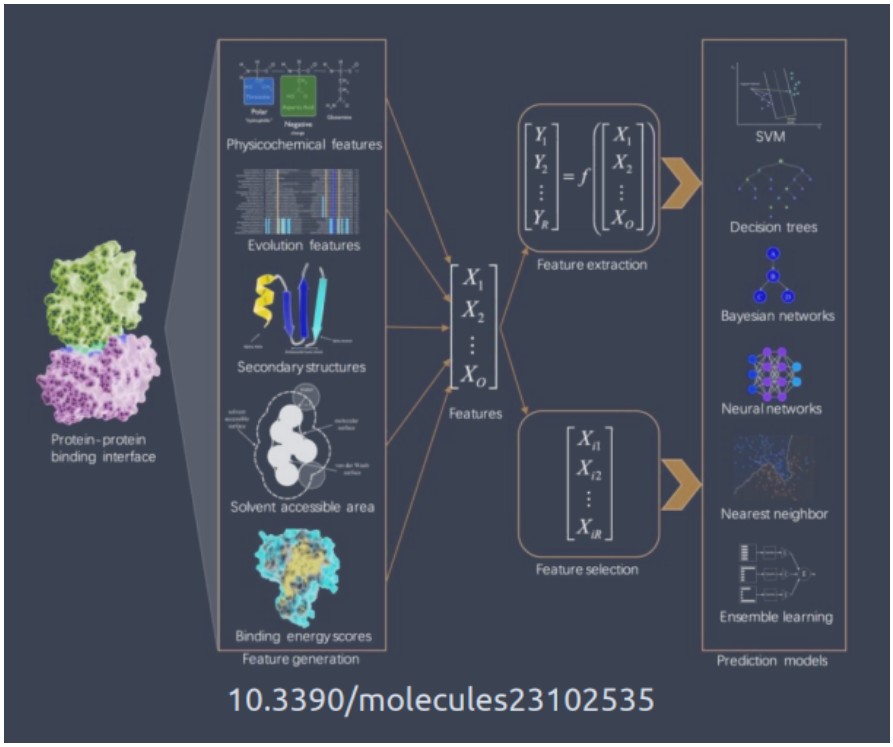

Рис. 5.16. Hot spots

Но у этого подхода есть один большой **минус**. Как у нас обычно происходит обучение? Есть выборка того, что связывается, значит, белок, структура и ΔG к этому. А выборки того, с чем белок не связывается, у нас нет. Выборки ddG, например, при введении мутаций, очень редки. Поэтому и обучение весьма ограничено. Ну и само определение изменения энергии связывания производится экспериментально и зачастую не унифицировано, и разумного способа привести разные эксперименты к одному значению ddG связывания нет. Все эти причины приводят к переобучению, поэтому при работе в данной области надо быть осторожными: с маленьким data set будет много проблем, а больших пока нет. Сложно придумать, как сделать так, чтобы было мутаций, и мы на их основе на интерфейсах разных белков умели бы хорошо дискриминировать их по ddG.

Естественно, 3D структуры используются не полностью. Было бы полезно интегрировать методы оценки, дополняя конформационным разнообразием, которое можно получить для белков с помощью молекулярной динамики, либо получить предварительные оценки о контактах с помощью маркомолекулярного докинга. То есть hot spots могут лучше заработать, у нас будет дополнительная массовая информация о том, как данные белки взаимодействуют хотя бы в каких-то моделях и какая динамика есть у этих белков, чтобы это можно было оценить.

Однако пока хорошего способа соединить макромолекулярный докинг и молекулярную динамику нет, потому что когда делается динамика белка в растворе, он оптимизирован на то, чтобы быть один. У него есть давление гидрофобного эффекта, и он достаточно компактный. Когда же он начинает взаимодействовать с другим белком, могут отходить петли, становится выгодным более раскрытое состояние, которого в динамике в воде просто нет.

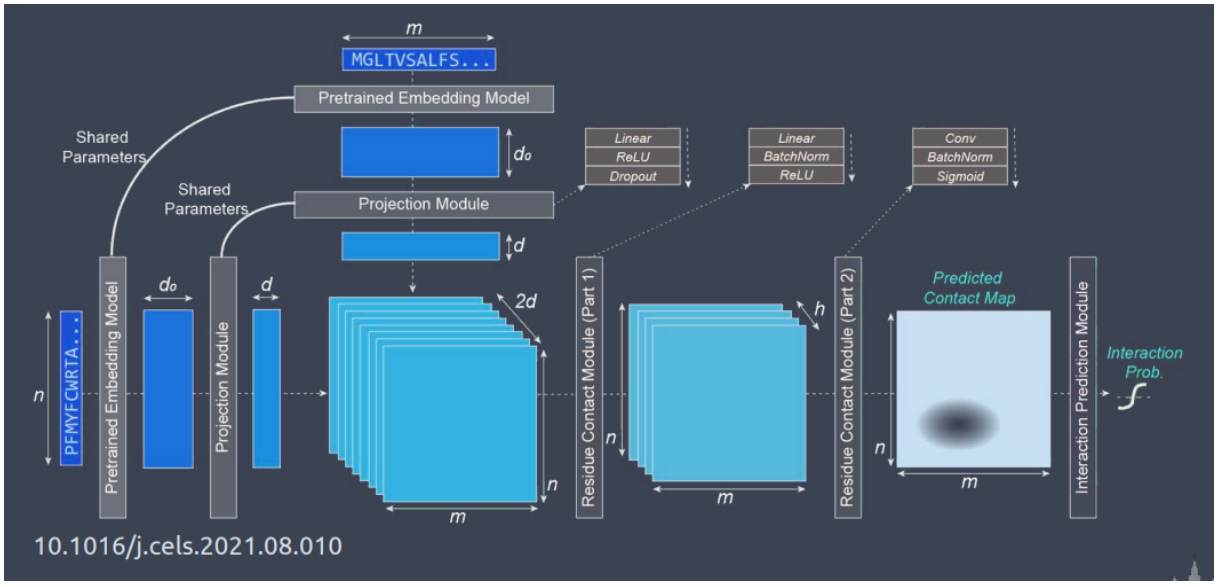

Рис. 5.17. D-SCRIPT

Ещё есть обученная сеть, которая пытается **предсказывать контактирующие аминокислоты в 3D** (рис. 5.17). Это обученная сеть на основе последовательностей.

Она хорошо предсказывает парные контакты и у каждого контакта выставляет уровень вероятности (рис. 5.18). Именно эти контакты, судя по выравниваниям, относятся к «горячим точкам», то есть консервативны, не относятся к гидрофобике в явном виде или относятся к гидрофобике, которую нельзя заменить.

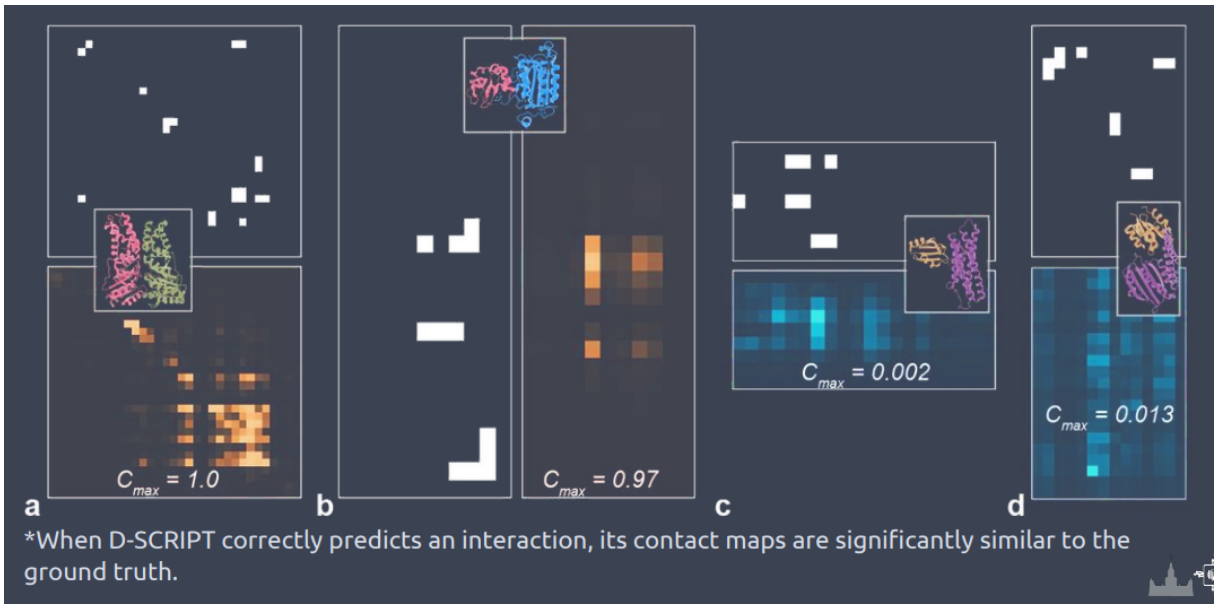

Рис. 5.18. D-SCRIPT, результат

Это сложно назвать сильно автоматизированным методом. Если общее предсказание является в целом верным, то горячие точки идентифицируются хорошо, если нет – плохо.

У нас уже было обсуждение про **MASIF**. Это Interaction fingerprints, которые строятся на основе геодезических карт и описания точек, придания им химического смысла типа заряда, гидрофобики и др., придания им молекулярно-механического смысла на основе контактирования.

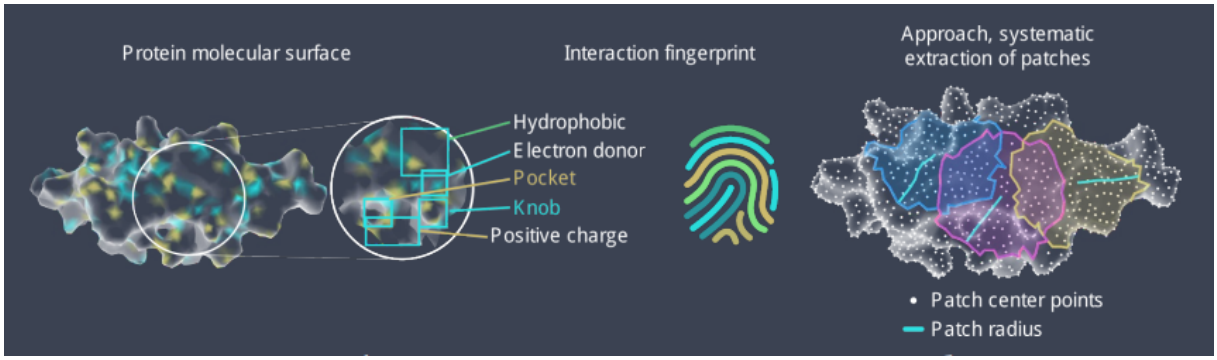

Рис. 5.19. MASIF

Из этого был построен классификатор, который позволяет узнать, есть ли теоретически на поверхности у белка участки, которые могут взаимодействовать с другим белком (рис. 5.19). Формально сетка помнит все белок-белковые контакты и смотрит, какая часть поверхности этого белка может участвовать в белковом контакте.

Мы разбили все известные комплексы на обучающую, проверяющую выборку, и метим поверхность белка, которая могла бы потенциально участвовать в белок-белковых взаимодействиях. Было показано, что введение любой химической информации сильно повышает факт предсказания (рис. 5.20).

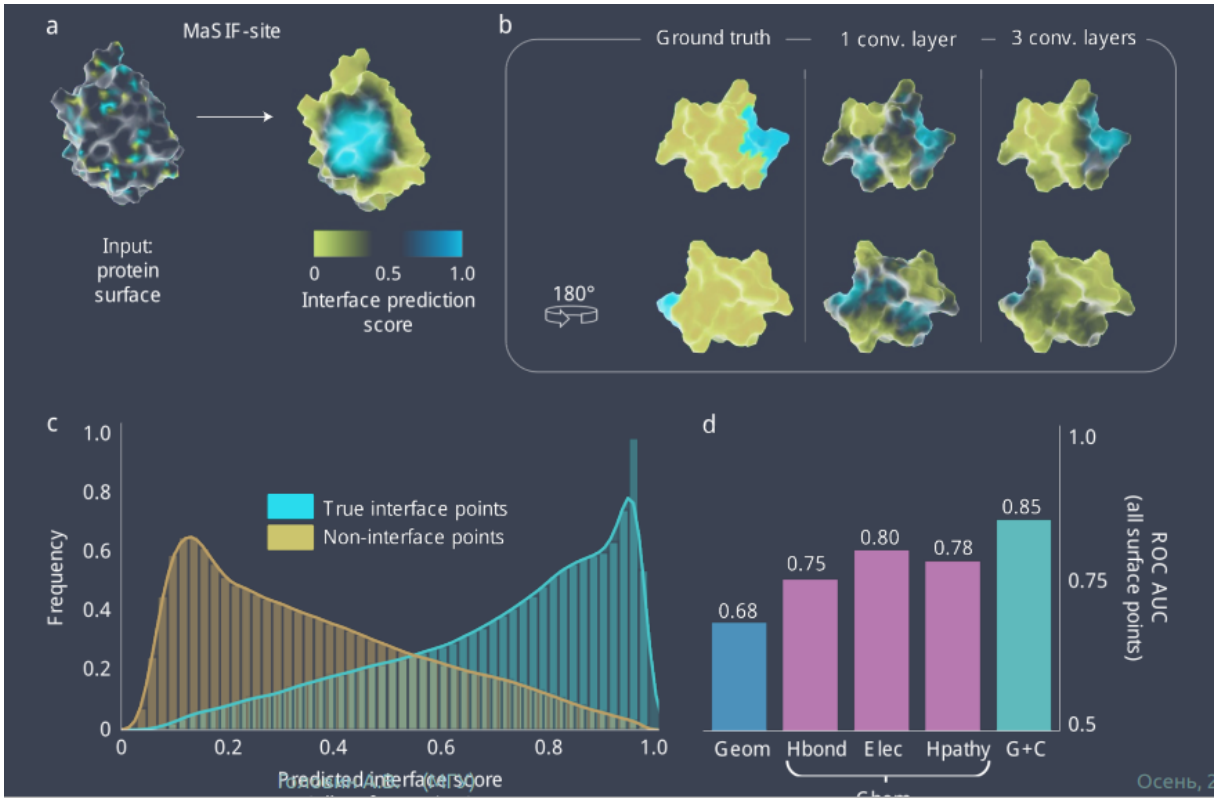

Рис. 5.20. MASIF, предсказание участков

Вопрос – как определить, чтобы белок может взаимодействовать с чем-то? Если у каждого белка есть поверхности, которые могут быть объявлены как участники взаимодействия, для каждого белка есть вектор, который описывает его потенциальную поверхность контакта. Дальше надо сравнить эти векторы друг с другом. И там, где расстояние между векторами маленькое, делаем заключение, что это может быть белковой парой.

Есть и другие алгоритмы, которые могут предсказывать поверхность контакта с белком. И на рис. 5.21 видно, что MASIF имеет преимущество. Однако преимущество не очень большое (16%), если сравнивать с достаточно простым подходом, который опирается на множественное выравнивание и описание поверхности.

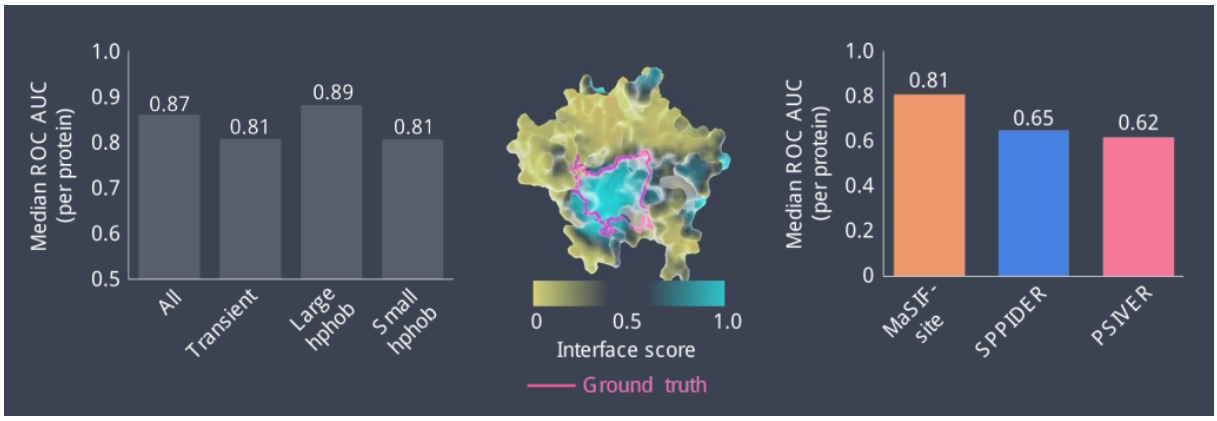

Рис. 5.21. MASIF, сравнение

До сих пор множественное выравнивание мы использовали только один раз, когда сравнивали с другой программой. Исходно MASIF не использует информацию о консервативности и ко-эволюции остатков. Поэтому можно было ожидать, что этот метод должен работать не только на природных белках, но и на искусственных. И действительно, MASIF смог хорошо предсказать поверхности контакта для искусственных белков. Это означает, что искусственные белки не так уж сильно отличаются от нативных с точки зрения контактов.

Это хорошо, потому что открываются перспективы использования **MASIF для дизайна** (рис. 5.22).

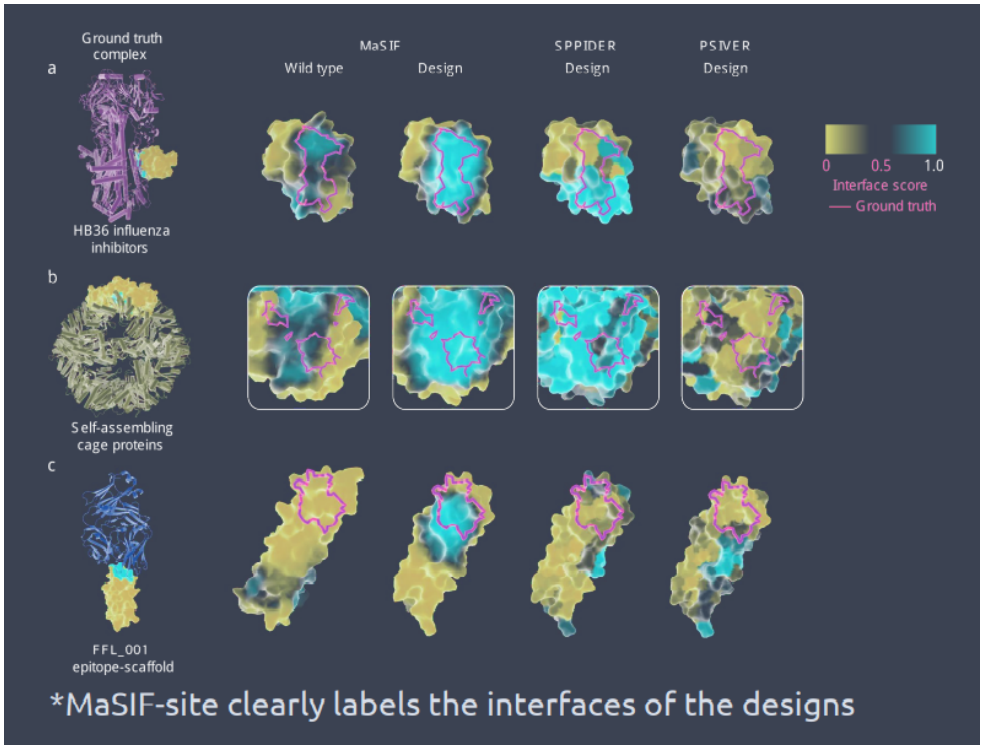

Рис. 5.22. MASIF, предсказание для новых белков

Например, мы нашли на белке поверхность, которая может образовывать контакт с другим белком. Дальше задача найти белок, который бы идеально воспроизводил обратный вектор к этой поверхности. Для белков это сделать сложнее, чем для низкомолекулярных лигандов, но возможно. С другой стороны, т. к. MASIF не опирается на информацию о последовательностях, мы не используем всю информацию о белках. Но это также и хорошо, потому что здесь больше рассуждений из области простой физики.

Можно придумать очень быстрый сканер на попарные белок-белковые взаимодействия, просто опираясь на то, что мы можем для каждого белка построить вектор, который описывает потенциальную поверхность связывания с другим белком (рис. 5.22). У некоторых белков может быть несколько таких поверхностей и несколько векторов. Дальше умножаем на минус один, чтобы перевернуть геодезию поверхности в виде вектора, отражаем химию кроме гидрофобики и сравниваем все вектора против всех. По рис. 5.23 видно, что это эффективно находит взаимодействующие пары.

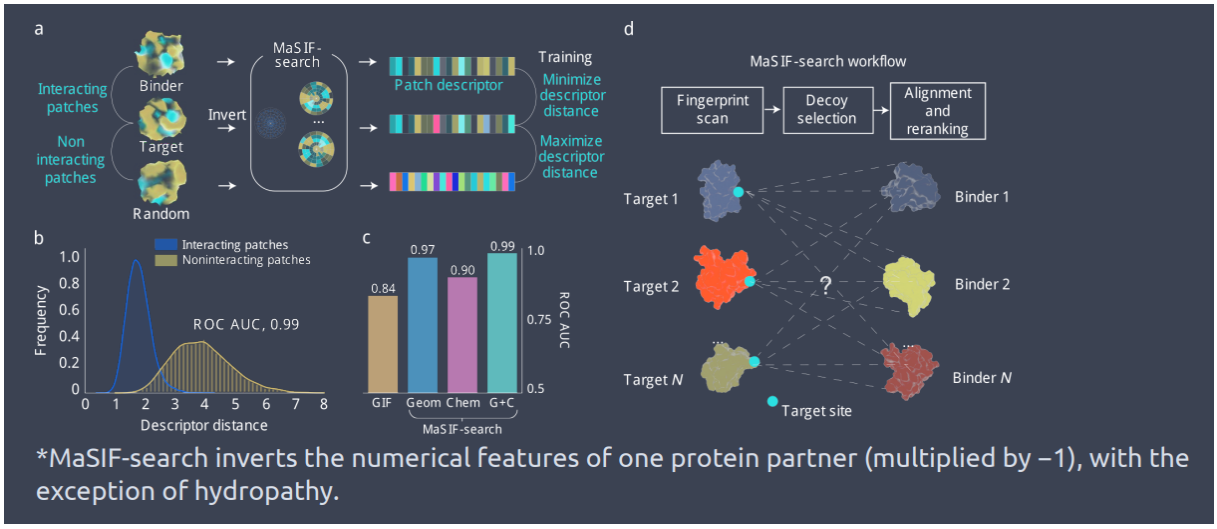

Рис. 5.23. Ultrafast scanning with MASIF

Так ли хорош MASIF? Теоретически да, но ZDock лучше. Однако, если посмотреть на скорость счёта (рис. 5.24), станет понятно, что никто не будет делать ZDock все против всех, а MASIF можно. Методы машинного обучения не дают сильного преимущества в точности предсказаний, кроме Alphafold, но большинство из них хороший выигрыш при умеренной точности предсказаний в скорости расчётов. Это важно для дизайна, потому что там возникает очень много вариантов.

Ещё было бы хорошо добавить к этим методам эволюционную информацию, чтобы улучшить качество предсказаний.

Table 1 | **Results for large-scale docking benchmark benchmark for PatchDock, MaSIF-search (with multiple numbers of decoys), ZDock and ZDock+ ZRank2 on bound (holo) complexes**

Method	Number of solved complexes in the top			t ime (min)
	100	10	1	
MaSIF-search decoys = 100	37	36	30	4
MaSIF-search decoys = 2,000	67	56	43	39
PatchDock	43	32	21	2,743
ZDock	58	36	18	134,934
ZDock+ ZRank2 decoys = 200,000	77	63	45	159,902

No. of solved complexes in the top, number of target–binder complexes within 5 Å iRMSD found in the top 100, top ten or top one (for holo cases) or top 1,000, top 100 and top ten (for apo cases). Time (min), CPU time in minutes for each program, which excludes precomputation time for MaSIF-search.

Table 2 | **Results for large-scale docking benchmark benchmark for PatchDock, MaSIF-search (with multiple numbers of decoys), ZDock and ZDock+ ZRank2 on unbound (apo) complexes**

Method	Number of solved complexes in the top			t ime (min)
	1,000	100	10	
MaSIF-search decoys = 2,000	17	7	2	16
PatchDock	11	4	1	560
ZDOCK	17	13	5	13,174
ZDock+ ZRank2 decoys = 80,000	23	12	5	16,866

*Moreover, all these methods could benefit from sequence evolutionary data to improve their predictive capabilities.

Рис. 5.24. Сравнение MASIF и ZDock

Переходим к биоархиву октября 2021 года. Рассмотрим **ColabFold** – реализацию Alfafold Овчиниковым (рис. 5.25). Исходно Alfafold предполагает, что мы даём на вход последовательность и на выходе получаем структуру. А как можно получить комплекс? Сначала решили вставить в последовательность много глицинов (примерно 20). Вставка такого длинного хвоста, а потом присоединение следующего белка, который должен был образовывать комплекс, давала с помощью Alphafold приличные ответы.

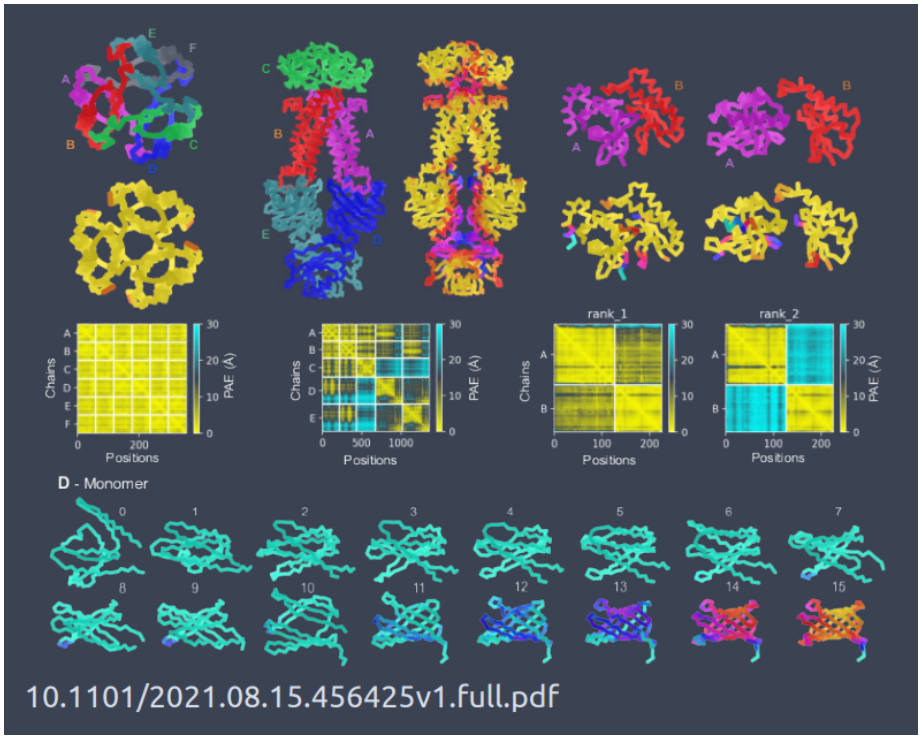

Рис. 5.25. ColabFold

А Овчиников научился эффективно делать выравнивание с гэпом. Он вставлял много гэпов выравнивания, чтобы разделить цепи белка. В итоге получилось, что для гомодимеров с высоким качеством локального сходства получалось сформировать мультимерные комплексы. Жёлтое на рис. 5.25 – это ошибка расстояния на интерфейсе, и она достаточно низкая.

Когда мы начинаем работать с мультигетерокомплексами, всё делится на квадратики, потому что, когда мы попарно сравниваем, некоторые квадратики могут быть близко, а некоторые нет. Видно, что добиться высокого локального сходства можно, но всё равно остаются проблемные места, которые не очень хорошо работают. Ещё правильным эффектом является то, что увеличение количества гэпов приводит к более дискретному разделению на два белка, которые в рамках данной модели являются двумя доменами одного большого белка. Если они разделяются плохо, получается ошибочный комплекс.

Вообще это является не очень хорошим решением, потому что исходно Alphafold обучался на то, что есть одна цепь белка, там попарные контакты, которые коэволюционируют, сохраняются, и из этого строилась модель. Даже если закладывались многоцепочечные молекулы, они разделялись на разные цепи. Эволюционная информация, заложенная в модель Alphafold, справедлива не только для внутрибелковых контактов, но и для межбелковых.

Дальше создатели Alphafold сделали **Alphafold-Multimer**. Он появился 4 октября. В нём модифицировали функцию потерь, чтобы учесть симметрии перестановок идентичных цепей: цепи можно переставить по выравниванию, и от этого должна быть минимальная функция потери качества. Дальше сделали автоматизированное совмещение двух выравниваний в одну модель для того, чтобы мы могли уже при обучении напрямую использовать информацию о том, что две цепи взаимодействую друг с другом, утилизирую при этом эволюционную информацию. Также там применяется новый способ выборки набора остатков для обучения, смотрят не только на взаимодействие в цепи, но и между цепями. И другие мелкие оптимизации.

Этот подход даёт существенное преимущество в качество расположения двух цепей друг относительно друга. Однако качество всё-таки ниже качества исходного Alfafold, скорее всего, потому что информация о межмолекулярных взаимодействиях в PDB не очень хорошая (рис. 5.26). В PDB изрядная доля белок-белковых контактов вызвана кристаллической упаковкой. Поэтому, если не использовать дополнительные базы данных о реальных взаимодействиях белков, обучение будет работать плохо.

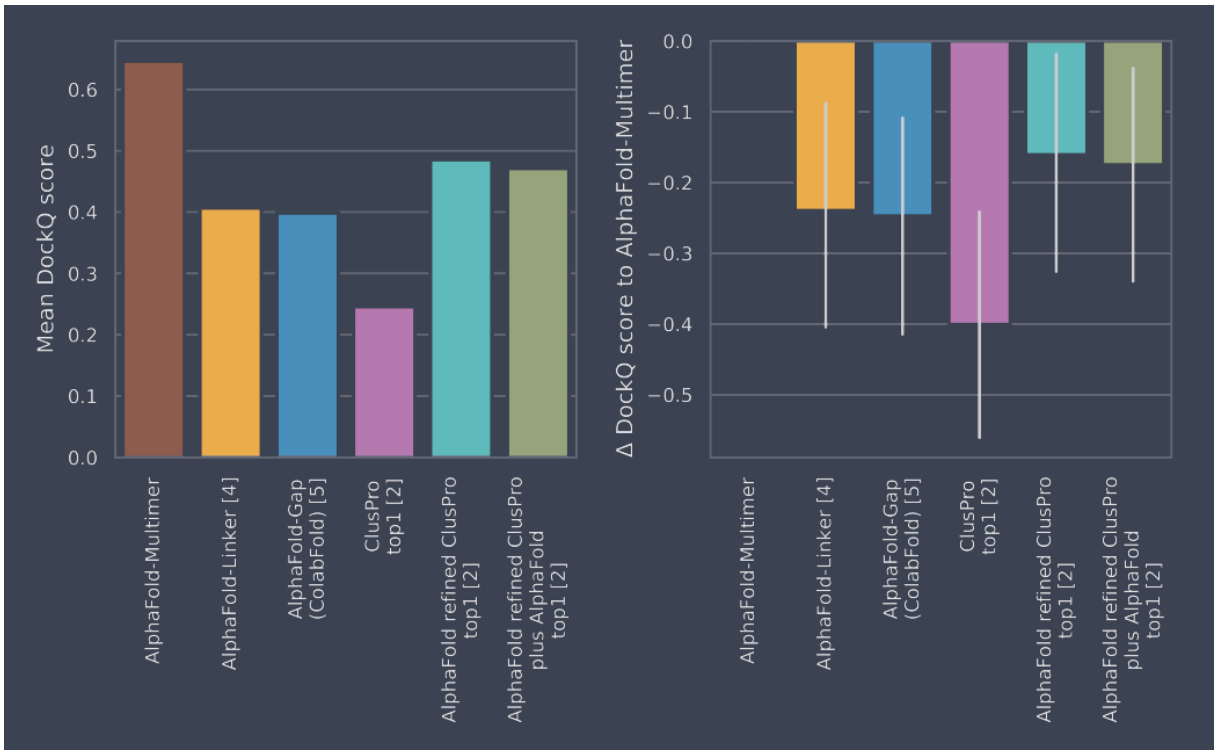

Рис. 5.26. Сравнение Alphafold-Multimer с другими методами

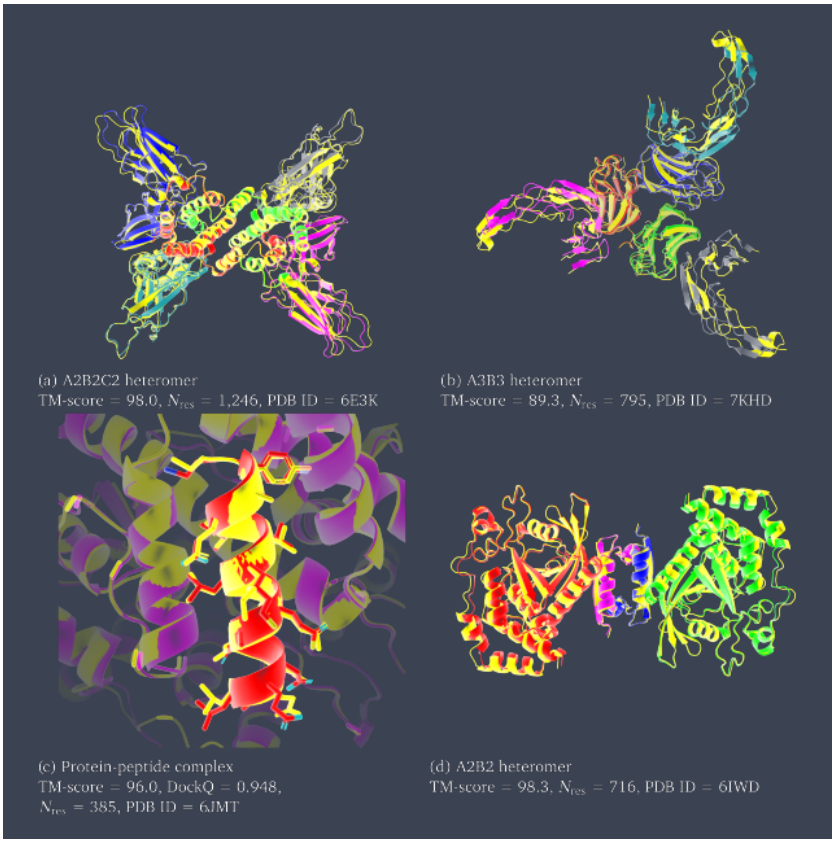

Рис. 5.27. Результат для пептид-белкового комплекса

Всё хорошо совмещается, ошибки, особенно в районе интерфейса, минимальны (рис. 5.27). Очень хороший результат получен для пептид-белкового комплекса. Это может быть важно в медицине для исследования аутоиммунных заболеваний.

Есть и негативные результаты (рис. 5.28). Например, ориентация белка получалась плохой, когда зоны контактов были очень подвижными. Сама по себе зона контакта получилась неплохо, но её смещение привело к значительному изменению геометрии всего белка. Таких результатов очень много при моделировании с помощью ZDock. Среди них можно найти структуру, максимально близкую к правильной, если мы знаем правильный ответ. Без дополнительных данных выбрать правильные результаты из 1000 при работе с комплексом, о котором мало что известно, очень сложная задача.

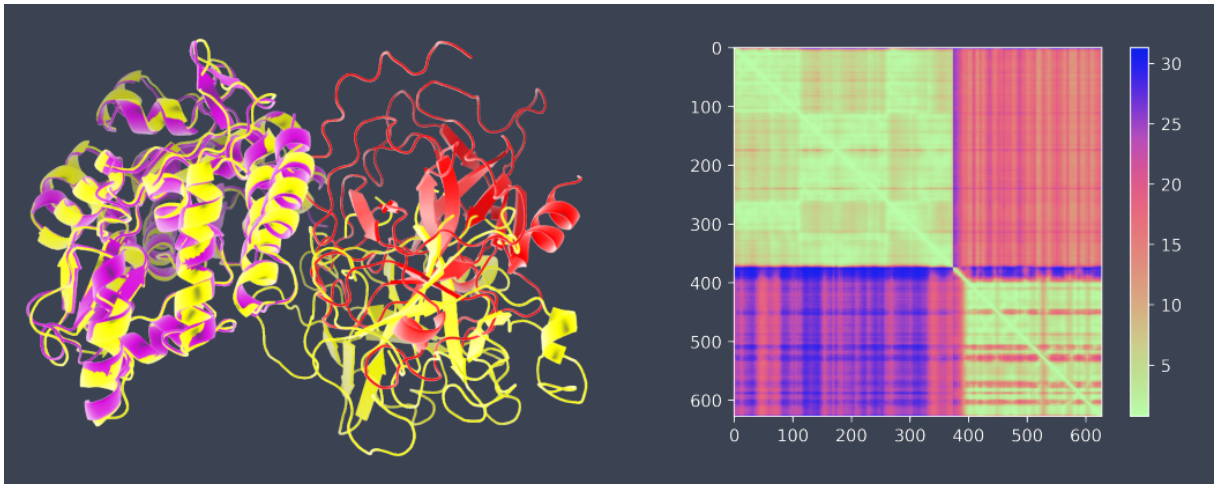

Рис. 5.28. Негативные результаты

Резюмируя, можно сказать, что есть **три основных типа достижений**. Это поиск hot spots – важных остатков, контактов на поверхности белков, имеющих большое значение для энергии связывания. Второе – обнаружение самих мест контактов на поверхности белка без построения моделей комплексов. А сами модели комплексов можно построить, используя методологию Alphafold, которая опирается на большой пласт знаний об эволюции и хорошо обработанные модели по парным взаимодействиям из PDB.

Лекция 6. Машинные модели для расчёта свойств электронной структуры молекул

Введение

Какие у нас задачи? Что такое **химическое разнообразие**? Оно не такое большое, как пространство разнообразия белков, но тоже большое. Мы не всегда можем найти корреляции между строением молекулы и её свойствами.

Рассмотрим, какие у нас есть **подходы для работы с молекулами**. QM методы – те, которые основываются на расчётах квантовой химии, а именно на ряде приближений к уравнению Шрёдингера, из которого получаются результаты. На сегодняшний день этим методы имеют два основных направления – расчёты ab initio и расчёты теории функционала плотности. Если первые расчёты опираются только на теорию и являются самодостаточными, то второй метод гораздо быстрее, но является отчасти эмпирическим.

Но всё равно все эти расчёты достаточно дороги. Это связано с тем, что само представление об электроне, о его положении, смещении напрямую связано с интегралами и большими вычислительными матрицами, да и электронов в молекулах много. Но для небольших молекул можно эффективно предсказывать их свойства, не зная практически ничего кроме химического строения.

Статистическая физика позволяет эффективно исследовать конформационное разнообразие и получать информацию о макроскопических свойствах. Это более характерно для исследования белков и крупных систем, где макроскопические свойства сильно отличаются от микроскопических.

Первый метод зачастую опирается на квантовую химию, второй – на моделирование в силовых полях. Но ни тот, ни другой не даёт возможности эффективно сканировать всё химическое разнообразие. Идеи о том, как найти **взаимосвязь между строением молекул, свойствами** и прочим мы уже обсуждали. Но люди до сих пор пытаются использовать более детальное описание в явной попытке учесть электронную плотность. Эти идеи могут дать преимущество, сохраняя идею о физичности модели.

Часто можно сделать так: есть набор соединений, для него строим таблицу, то есть рисуем SMILES, напротив ставим свойства и пытаемся манипуляциями со SMILES добиться правильной регрессии, которая позволила бы подтверждать, что такой комбинации SMILES соответствует такое число во втором столбике. Это хорошо, но представляет из себя чёрный ящик.

Было бы гораздо более научно внести в это основы физики, попытаться увидеть электроны или их аппроксимации какими-то описаниями. Тогда при построении модели мы могли бы делать заключения, которые точно не получится сделать, используя методологию чёрного ящика.

Эта область является такой же молодой, как и всё, связанное с ML. **QML** – это комбинация методов квантовой механики и статистической механики для того, чтобы добиться рационального и реалистичного описания молекул. Сам по себе QML – абстрактный набор методов, который ссылается на современные методы статического обучения, чтобы предсказать атомистические и молекулярные свойства процессов и молекул, которые в них вовлечены, в том числе и в материалах.

Надо хорошо понимать, что здесь легко перепутать данную область с машинным обучением в квантовых расчётах. Квантовые компьютеры работают на других принципах, для них надо придумывать новую модель машинного обучения, и это не связано с квантовой химией, хотя и может быть использовано для методов квантовой химии. У QML основная цель – описывать электронную структуру и получать из этого описания достаточно качественные модели, которые позволяют предсказывать на их основе некоторые свойства молекул. Здесь самый важный шаг – наличие описания именно электронной структуры, которая порождает эти свойства, а не просто прямая экстраполяция SMILES с какими-то свойствами.

Основная **цель методов QML** в том, чтобы эффективно экстраполировать найденные модели на всё многообразие химической выборки, а это примерно 10^{43} соединений. Если модель работает на всей выборке, это позволяет считать, что мы построили её эффективно. Мы можем быть уверены, что это сработает, потому что строим модель на основе электронной плотности, а это универсальное описание, ведь электронная плотность в молекулах строится по совершенно прозрачным физическим законам. Поэтому мы вправе ожидать, что, если экстраполируем электронную плотность моделями, то всё равно продолжаем идеологию прозрачности это описания для выявления свойств.

Здесь надо уже всегда бороться за эффективность, можно в ущерб универсальности. Есть мы работаем с биологическими объектами, нет смысла исследовать всё химическое разнообразие. Есть смысл рассмотреть некоторые металлы и элементы второго ряда таблицы Менделеева.

Ну и надо реализовывать эти методы. Опираясь на электронную плотность, мы можем пытаться генерировать молекулы, которые могли бы связаться с белком в нужном месте, уже непосредственно учитывая межмолекулярные взаимодействия, основанные на электронной плотности.

Наборы данных

Какие у нас есть наборы данных? Один из самых известных в данной области – **набор GDB**, который является генерированным. Всего для 11 тяжёлых атомов строятся графы, генерируются стереоизомеры, которые могут породить 13,9 млн соединений. Данный набор данных предполагает, что для этих наборов соединений мы уже строим 3D структуры, проводим фильтрацию и получаем из этого набор соединений, которые похожи на лекарства. Они могут не существовать в природе на текущий момент, но

являются химически осмысленными, то есть могут теоретически существовать и потенциально их можно сделать и использовать для поиска новых лигандов, соединений и т. д.

GDB развивался и сейчас уже достиг размера 166 млн. соединений (рис. 6.1). Их так много, потому что сначала мы делаем 114 млн. графов, потом убираем в них заведомо глупые соединения, дальше из этого получаются вариации, куда мы добавляем двойные связи, начинаем их перетасовывать. Всё это порождает огромного количество молекул, которые мы видим в GDB.

Для того, чтобы объяснить актуальность такого набора данных, можно посмотреть на сравнение представленности разных молекул в разных базах данных. На рис. 6.1 видно, что **GDB-17** покрывает то же самое пространство, что и другие продвинутые базы. Это говорит о том, что больше расширять её пока не надо, однако если мы хотим прийти в область всего химического разнообразия, не надо ориентировать на известные базы данных. Но уже сейчас полезно иметь базу данных сгенерированных молекул, которая профилирована как база известных молекул с точки зрения молекулярного веса и др. А вот её содержание гораздо больше.

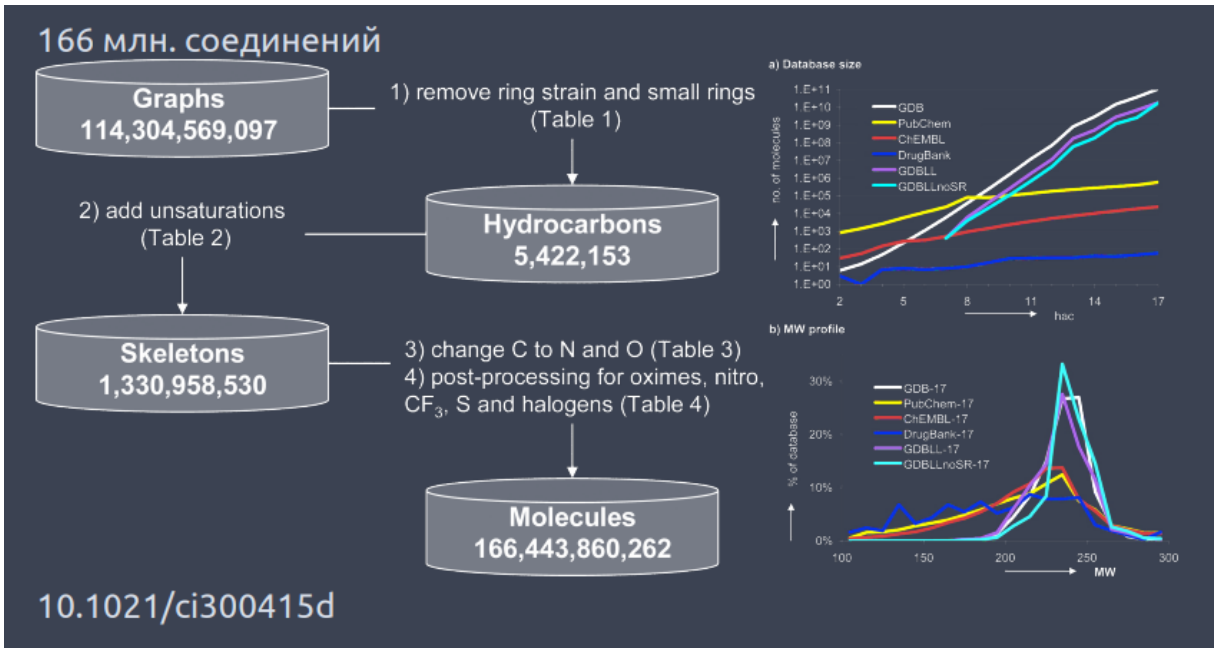

Рис. 6.1. GDB-17

Но надо понимать, что в химическом синтезе ещё много неоткрытых реакций, и мы можем придумать молекулу, для которой нет явного известного способа синтеза. Надо придумывать что угодно, а потом думать, можно это синтезировать или нет.

Есть **разные варианты подобных баз данных**, когда есть выборки, содержащие не только оптимизированные молекулы, но и рассчитанные для них, исходя из квантовохимических расчётов, энергии: HOMO, LUMO, E, E_i. Такие базы гораздо меньше, потому что там надо очень много всего считать.

База QM9 содержит в себе 134 тыс. соединений, для которых рассчитано много квантовохимических чисел. База MD17 содержит всего 10 соединений, но зато мы получаем не равновесные геометрии этих соединений, а пытаемся накопить всевозможные конформации, которые могут быть у атомов в этих соединениях. Это делается методами молекулярной динамики на основе ab initio расчётов, то есть не в силовых полях, а в квантовой химии. Это вычислительно дорого, но позволяет получить очень много конформаций, и из них можно получить много разных геометрий и использовать для обучения.

В качестве отдельного достижения надо отметить набор ANI1. Это идеологическое продолжение набора MD17. Там есть 57 тыс. соединений, и в сумме они дают около 20 млн. конформаций.

Посмотрим детально, как это работает (рис. 6.2). У нас есть 57 тыс. соединений. Когда они выбирались, скорее всего, было ограничение на молекулярный вес, разветвлённость и т. д., чтобы был более-менее равномерный data set. Дальше для этих соединений генерируется 3D структура. Сначала идёт оптимизация с помощью силовых полей, потом оптимизация на DFT. Если молекула не сходится, она выбрасывается. Дальше проводится анализ нормальных мод – расчёт гессиана. Это попытка выделить силы, действующие внутри молекулы, чтобы описать, как она колеблется при данной температуре. Из этого мы тоже можем породить конформации, учитывая, что, если данную связь надо растянуть с данной силой, можно оценить, как будет меняться и колебаться связь при данной температуре.

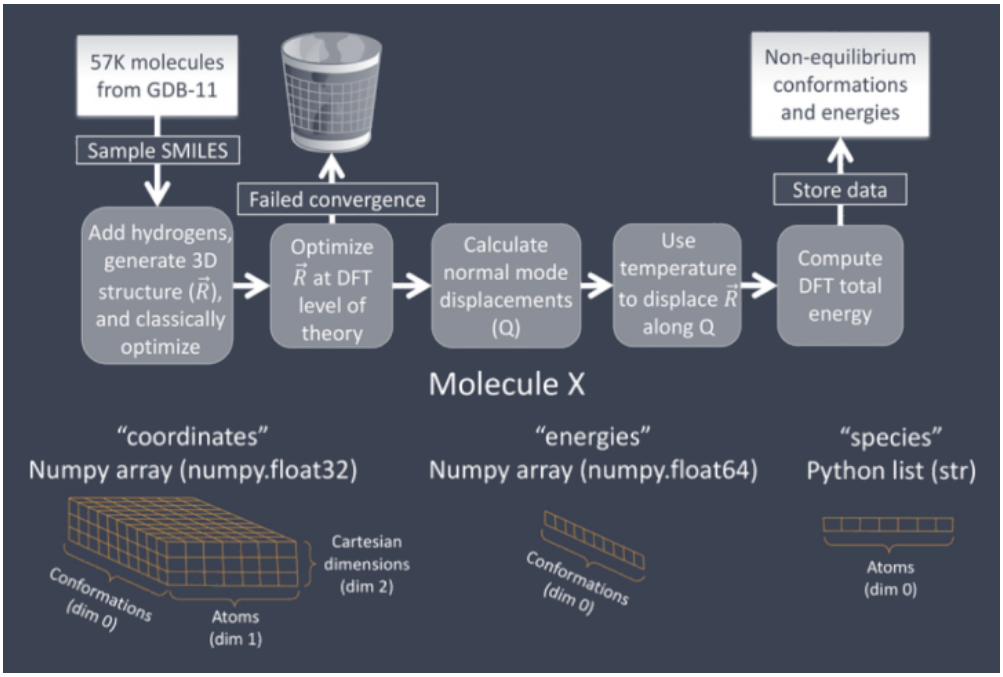

Рис. 6.2. ANI-1

Можно провести аналогию связи с пружинкой: кинетическая энергия при данной температуре одна и та же, но в зависимости от жёсткости пружинки амплитуда

колебаний будет разной. Дальше температура используется, чтобы оценить смещение атомов. А потом для всех этих потенциальных смещений высчитывается значение энергии температурного колебания в этой точке, и из этого сохраняются неравновесная энергия и неравновесные координаты, что и порождает наш набор, на котором можно научиться, чтобы рассматривать вещества не только в состоянии глобального минимума, но и при какой-то температуре.

Если посмотрим на распределение углов, видно, что данный набор данных уверенно выходит за пределы равновесных состояний. Это хорошо, ведь если мы собираемся с помощью данного набора научиться моделировать молекулярную динамику веществ и реакции, нам нужны именно неравновесные состояния.

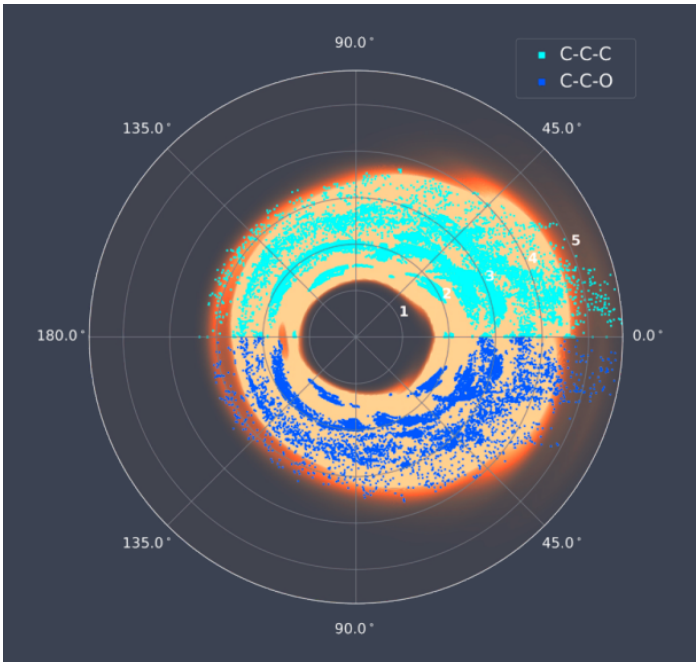

Рис. 6.3. ANI-1, сравнение с равновесными наборами

Сколько нам теоретически нужно данных, чтобы попытаться понять, хорошо ли работает наша модель? На рис. 6.4 по оси x – количество соединений в обучающей выборке, по оси y – средняя ошибка. Видно, что уже после 10 тыс. мы уходим в достаточно низкие значения ошибки, причём это происходит линейно для разных представлений. Для сферического перекрывания атомных позиций ошибка может упасть до 0,005 eV.

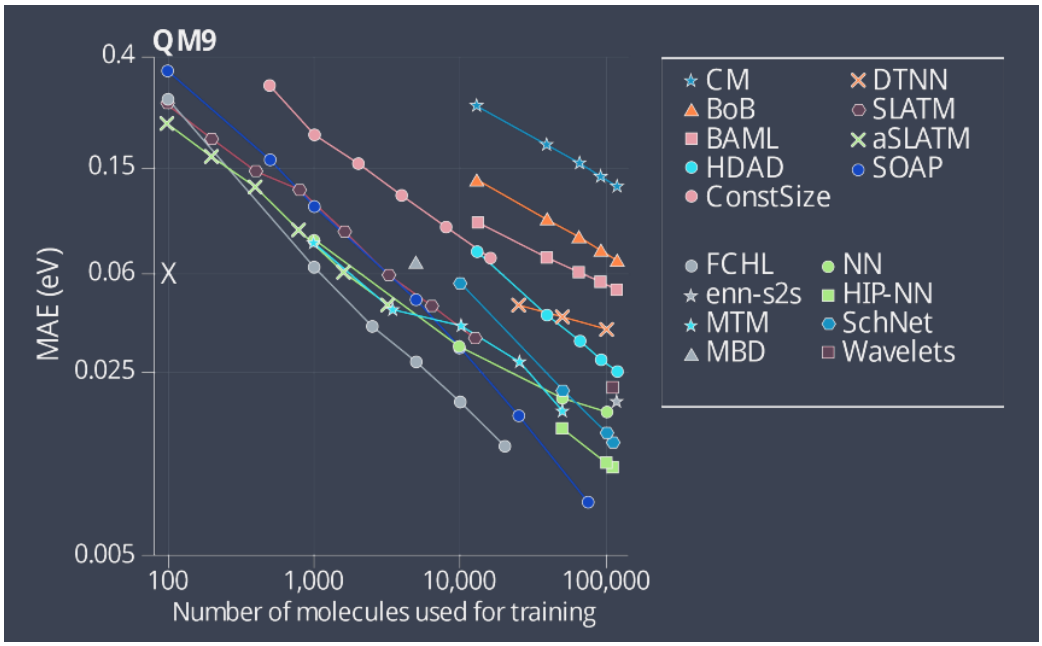

Рис. 6.4. Размер набора данных

Представление молекул

Обратимся к **представлению атомов и молекул**. На данный момент нет представления, которое удовлетворяет всем желаемым свойствам одновременно. Да и не так уж много людей в мире занимается этой проблемой, и они делятся на два основных лагеря: те, кого интересует биологическое применение подобных подходов, и те, кого интересует применение подобного в материалах. Оба подхода ориентированы на явно не газовую фазу, но материалы здесь – периодические или околопериодические системы, а белковые и молекулярно-биологические системы совсем другие – скорее водные, растворные. Поэтому надо понимать, что каждое из этих направлений пытается оптимизировать под себя и модель, и представление. Мы в основном будем говорить о моделях и представлениях, которые ориентированы на биологические системы.

На сегодняшний момент есть ряд представлений, которые удовлетворяют части общих требований. Все существующие представления опираются на то, что есть атом, у него определённые заряд ядра и положение в пространстве. Отсюда пытаются экстраполировать состояние электронной плотности и из этого высчитывать свойства.

На старте можно использовать не только представления, сделанные человеком, но и генерировать их в процессе обучения. Это делают энкодеры, но здесь мы не будем о них говорить, потому что это опять переход в систему чёрного ящика, а мы хотим, чтобы данный подход позволил быстро и эффективно экстраполировать свойства с электронным строением вещества.

В чём идея **атомцентрированных функций симметрии**? На первой лекции мы решали уравнение Шрёдингера для водорода. Мы представили всю систему в сферической системе координат, дальше было разделение переменных, появлялись

радиальная составляющая и две угловых. Всё это реализовано в атомцентрированных функциях симметрии, которые часто используются в том же ANI, и является физически осмысленным представлением.

$$G_i^1 = \sum_i f f_c(R_{ij}) \tag{6.1}$$

$$G_i^2 = \sum_i e^{n\left((R_{ij}-R)s\right)^2} f_c(R_{ij}) \tag{6.2}$$

$$G_i^3 = \sum_i cos\left(kR_{ij} f_c(R_{ij})\right) \tag{6.3}$$

Если посмотреть на рис. 6.5 для водорода, можно увидеть, что многие из них визуализируются очень похоже на волновые функции для атомов. Ожидаемый недостаток данного подхода – очень быстрый рост сложности вычислений с ростом числа атомов. Самая сложная задача в данной области – промоделировать с квантовой точностью динамику белка.

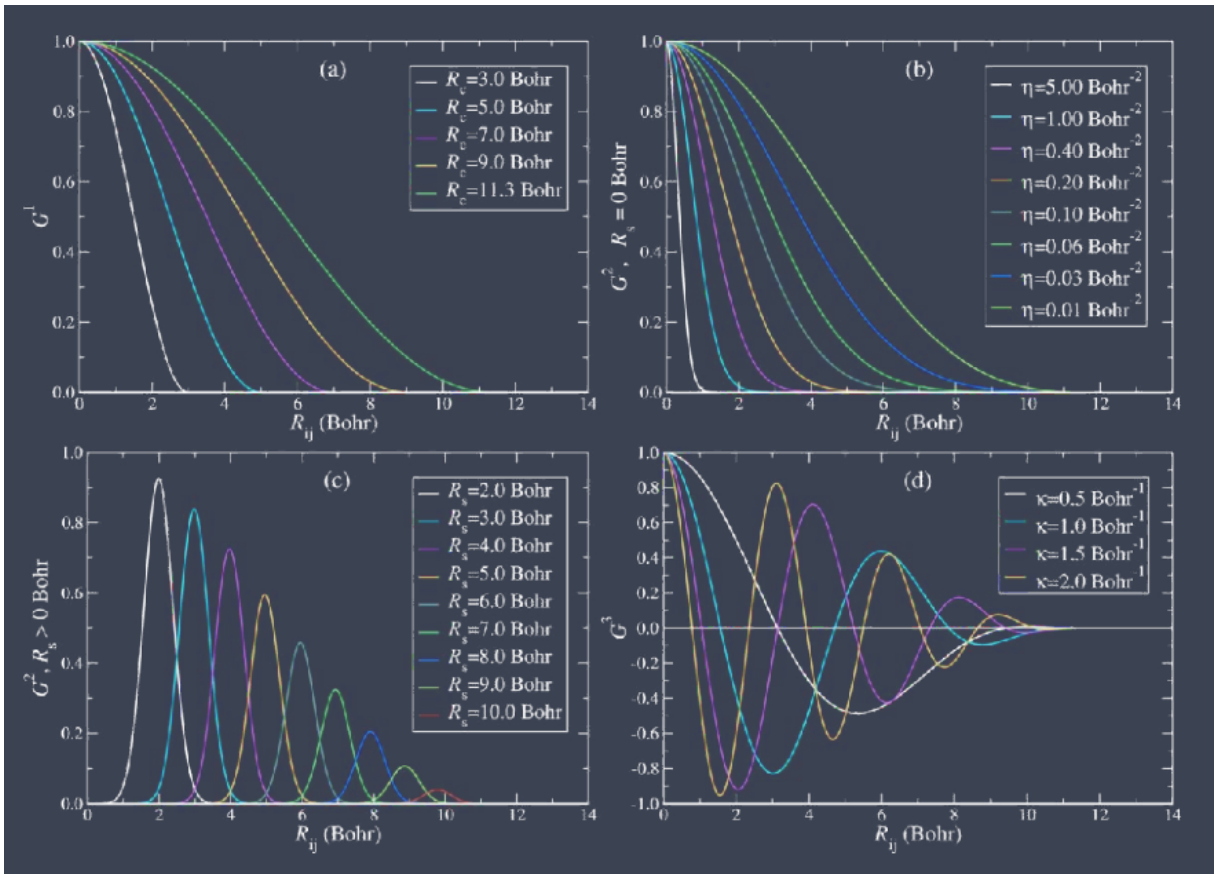

Рис. 6.5. Атомцентрированные функции симметрии

Альтернативное представление – **кулоновская матрица**. Это набор обратного значения попарных расстояний, которые имитируют кулоновское отталкивание между

атомами. Это тоже напоминает квантовую химию, потому что там есть радиальная составляющая не для электронной плотности, а для атом-атомного взаимодействия, в котором и есть это отталкивание. Отталкивание описывается электронной плотностью, но тоже отображает не в явном виде, и может быть использовано для обучения. То есть здесь задача – придумать описание, которое идеологически близко к электронной плотности, но в реальности её не описывает. Мы рассматриваем только ядра, но считаем, что они на таком расстоянии, потому что у них есть какая-то спрятанная в машинном обучении электронная плотность. Это хорошо, но не удовлетворяет симметрии перестановки. А вот это уже не очень хорошо, потому что мы считаем, что химически идентичные углероды могут быть переставлены.

$$\hat{V}_C = \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J}{|R_I - R_J|} \tag{6.4}$$

Посмотрим визуализацию этой кулоновской матрицы (рис. 6.6). Эта картинка не очень хорошая, потому что здесь показаны корреляции машинного обучения с энергиями, которые были получены квантовомеханическими полуэмпирическими методами. Это большой недостаток, потому что данный метод может сильно ошибаться, а учить нейронную сеть на ошибающемся квантовомеханическом методе неверно.

$$d(\mathrm{M},\mathrm{M}') = d(\epsilon,\epsilon') = \sqrt{\sum_I |\epsilon_0 - \epsilon_I'|^2}\,, \tag{6.5}$$

где ϵ собственные значения M.

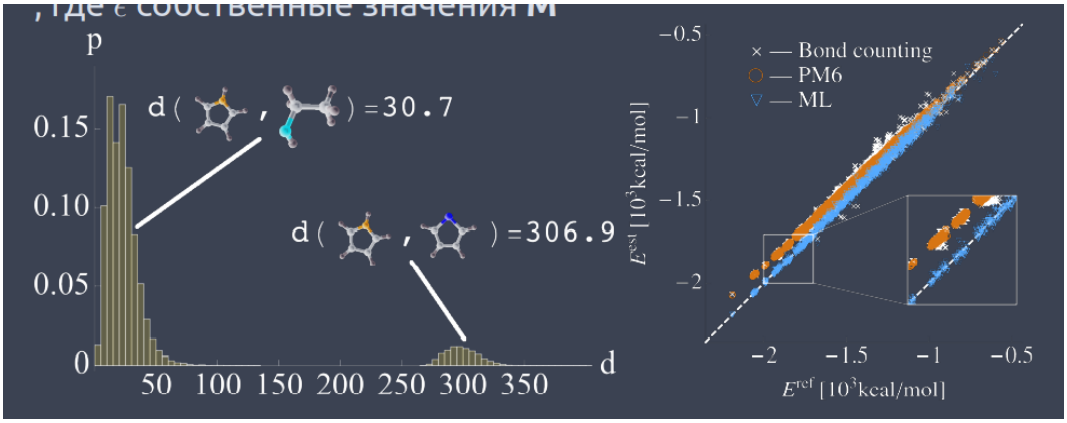

Рис. 6.6. Кулоновская матрица

Оказалось, что расстояние от пиррола до этанола меньше, чем если бы у пиррола вместо азота был атом серы. Здесь расстояния становятся слишком большими, потому что в формуле (6.4) в явном виде учитываются заряды ядер, а заряд ядра кислорода от азота отличается гораздо меньше, чем заряд ядра кислорода от серы.

На рисунке 6.6. слева – расстояние, которое рассчитывается как кулоновская матрица. Почему оно такое большое (300 усл. ед.)? Как различить две молекулы? Любую

молекулу можно представить в виде кулоновской матрицы, а матрицу схлопнуть в вектор. У этого вектора можно взять длину. Поэтому в таком виде пирол гораздо ближе к этанолу, чем к сере – похожему геометрически вещество, но сильно отличающемуся по элементному составу.

Наша задача – сделать обучающий набор, в котором написано, есть вещество, у него такое состояние, такая конформация, у неё такая кулоновская матрица. И набор содержит много миллионов веществ. А дальше мы по этим матрицам пытаемся искать корреляции с определёнными свойствами, потому что матрицы отображают электронное строение данного вещества. Используя эти корелляции, дальше для любого вещества, которое можем построить, для любой матрицы, можно предсказать его свойства.

Мы уже сказали, что матрица векторизируется. Это называется **Bag of bonds** – мешок связей (рис. 6.8). Получается немного лучше, потому что никуда не исчезает симметрия перестановок.

Рис. 6.7. Bag of bonds

Если говорить о **многотельном представлении молекулы**, то здесь уже попытка использовать расширение для мешка связей, в котором появляются дополнительные термины для того, чтобы рассматривать взаимодействия более чем двух тел. Если мешок связей – набор парных взаимодействий, да и кулоновская матрица парная, то переход к тройным взаимодействиям – более сложная вещь, в том числе и в вычислениях, но зато она решает симметрию перестановок. Но когда мы используем компоненты уравнения, которые могут учитывать более чем трёхтельные члены, скорость сильно падает.

Перейдём к одному из самых современных представлений **– Smooth overlap of atomic positions**. Здесь мы пытаемся использовать функции Гаусса, которые рассказывают о том, как перекрываются атомцентрированные радиальные функции. И при этом ещё строим гауссовское распределение и получаем зависимости, представленные на рис. 6.8 – сумму гауссовских функций с разницами координат.

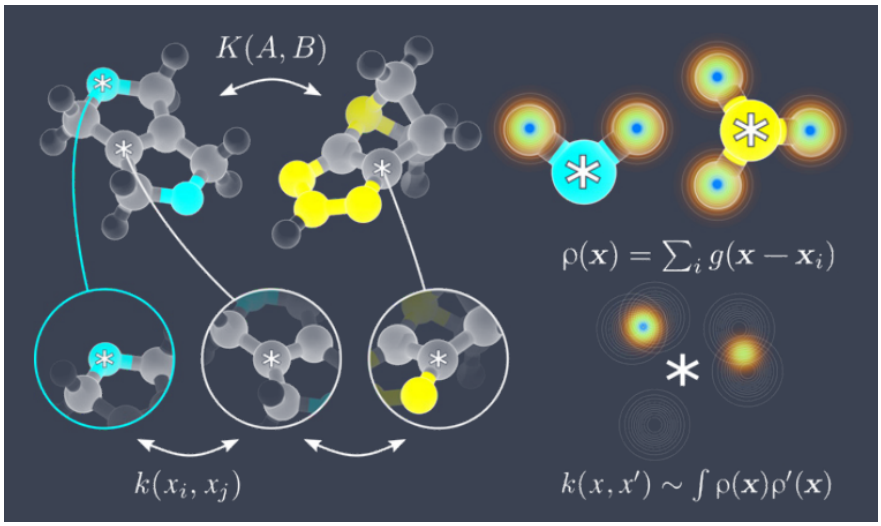

Рис. 6.8. Smooth overlap of atomic positions, метод

Это тоже похоже на попытку реализовать электронную плотность достаточно простыми функциями. Мы хорошо знаем, что электронные облака перекрываются, образуются связи. Чем ближе находятся атомы, тем лучше перекрываются облака. В некоторых случаях перекрывания и гибридизации нет, в некоторых есть.

Здесь хорошо, что функции плавные – потому что гауссиановские. Здесь может не быть разрывов, так как гаусс может тянуться далеко в спадающей части.

Данный метод строго относится к вращению и перестановочным симметриям. Это позволяет дискриминировать разные молекулы, их изомеры и т. д. Однако он может стать дорогим, как только начинаем говорить о большом количестве типов атомов в системе.

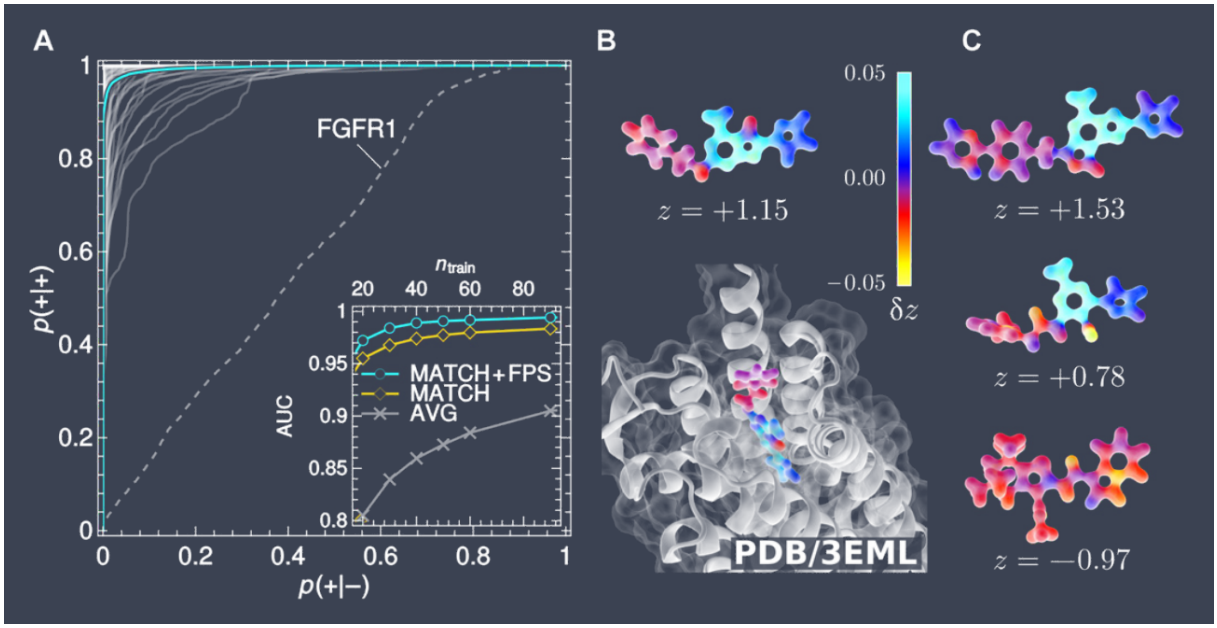

Рис. 6.9. Smooth overlap of atomic positions, связывание веществ с белками

Люди попытались с помощью данного метода научиться смотреть, как вещества связываются с белками. В зависимости от того, как происходят подобные пересечения гауссиановских функций, центрированных на атомах, получаются разные эффективности взаимодействий лиганда с белком. На рис. 6.9 видно, что не каждая конформация лиганда одинаково хороша для взаимодействия с белком.

Это похоже на хорошо устроенную функцию для докинга. К сожалению, она всё-таки не учитывает в явном виде подвижность белка, но может хорошо оценивать эффективность взаимодействий, именно энтальпийную составляющую. Также можно увидеть, что кривые достаточно хорошо идут по верхнему левому краю, то есть при небольшом изменении хорошая корреляция с наблюдаемыми данными. Значит, функция чувствительная и не сильно отклоняется от ожидаемых значений, которые были в обучающей выборке. Даже при небольшом обучении получаются хорошие значения, и при улучшении они тоже улучшаются.

Когда мы говорим, что есть вклад в связывание, должно произойти взаимодействие атомов, значит, мы можем накопить те scores, которые дают хорошее взаимодействие и те, которые дают плохое или вообще его не дают. Каждая кривая – свой рецептор и свой лиганд. Для них старались накопить конформаций, и для них высчитывали score по пересечению гауссианов. Из этого пытались показать, что метрика коррелирует с ответами, которые можно получить, исходя из знания о правильном связывании лиганда. Для каждого белка (для набора конформаций) есть n взаимодействий: какие-то правильные, какие-то неправильные. Проверка правильности проводится с помощью рентгена и конформаций. Дальше строим зависимость доли правильных взаимодействий от доли неправильных.

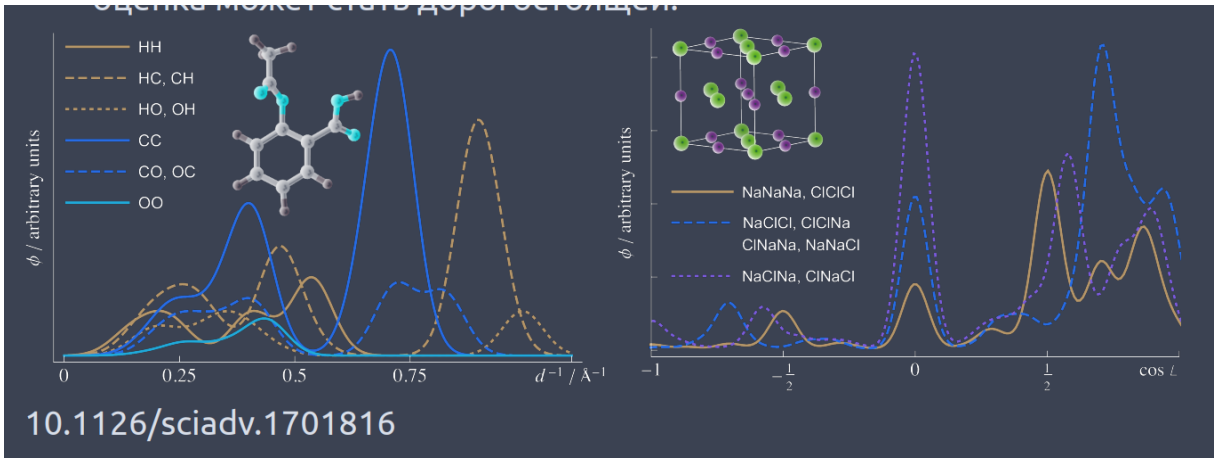

Рис. 6.10. Многотельное тензорное представление, кристаллы NaCl

Рассмотрим **многотельное тензорное представление**. Это попытка совместить несколько векторов (сил, свойств), которые между собой коррелируют (рис. 6.10). Их объединение позволяет более аккуратно представлять систему. Это хорошо работает по отношению к твёрдым телам, потому что это повторяющиеся системы, и у них много

корреляций, т. к. в кристалле все действия сбалансированы, тепловое движение минимально, и мы знаем, как сбалансировать все подобные коррелирующие вектора.

То же самое можно сделать и для молекул. Видно, что поведение группировки на рисунке 6.10 ограничено вокруг связи с кольцом. Значит, здесь тоже есть некое корреляционное движение.

Расчёты в таких случаях дорогие, но эффективные.

Вейвлет – это функция для описания затухающей волны. Это описание исходит из плосковолнового описания электронных орбиталей. Всё это больше ориентировано на материалы, но и для белков может быть хорошо использовано. Самое важное – не навязывается локализация данных свойств, потому что локализация всегда имеет свои недостатки, однако это привлекает идеологию компонент чёрного ящика. Здесь надо уметь делать хорошие электронные свойства, фиттить описания в простые системы, а потом экстраполировать на более сложные.

Можно сделать вывод, что представления всё ещё находятся в зачаточном состоянии. Всегда есть стремление найти компромисс между производительностью и физичностью.

Тензорные нейронные сети могут в ходе процесса привести к появлению многомасштабного представления, то есть некоторые вещи оцениваются грубо как удалённые взаимодействия, более близкие – более тонко. Это идёт из самого процесса обучения. Пока сложно сказать, хорошо это или плохо, потому что мало публикаций с полноценными тестированиями моделей.

Типы ML методов

Приведём пример, как может сильно **измениться архитектура сети при обучении**, когда начинают эволюционировать требования к сети (рис. 6.11).

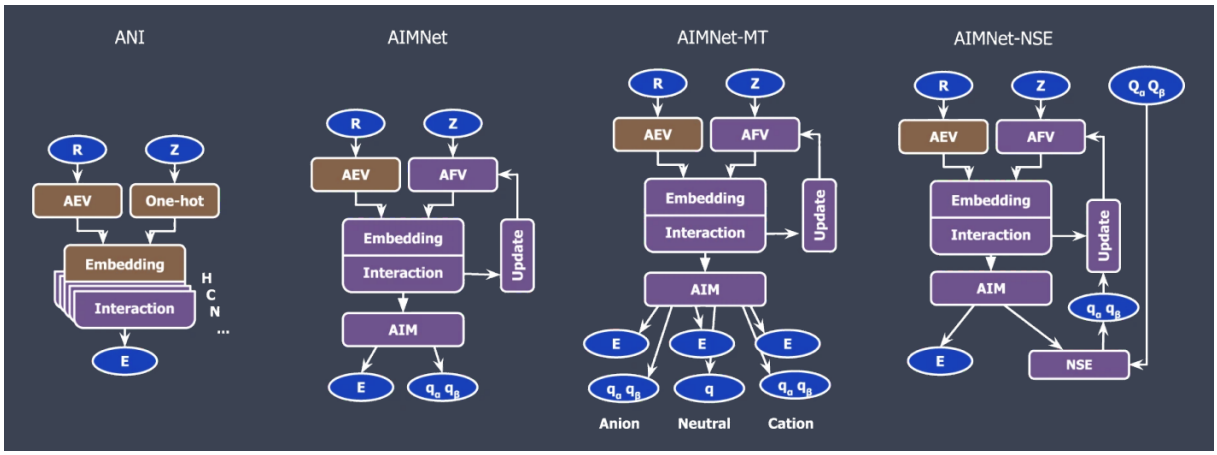

Рис. 6.11. Архитектуры NN

Исходно у нас был набор ANI, который предполагал, что есть молекула, координаты и заряды ядер, дальше мы всё это коррелирует с энергией. Получается нейронная сеть, которая работает так, что для каждого набора координат мы порождаем

энергию. А если мы пытаемся найти заряды? Когда у нас молекула начинает участвовать в реакциях, начинают образовываться катионы и анионы. В гидрофильной атаке образуются нуклеофилы, атакующие нужные атомы, и происходит реакция через образование зарядовых катионов и анионов.

Как только выяснилось, что это надо делать, оказалось, что учёт зарядов и перераспределения электронной плотности не попадал в обучающую выборку. Набор ANI генерировался так: была молекула, для молекулы делали анализ, для него генерировали конформации, для них считали энергии и на этом обучали сеть. Места депротонированным, протонированным, заряженным соединениям места в этом наборе не было, потому что они опирались на SMILES, а SMILES органических соединений зачастую нейтральны.

Сетку необходимо было усложнять, она эволюционировала, и на данный момент есть публикации, когда ANI модифицирована таким образом, что потенциально обучена для работы с анионами, катионами, неправильными состояниями и т. д. Значит, мы может использовать подобные QML-потенциалы для моделирования реакций в биологических системах. Это сильно увеличивает скорость счёта.

Мы даём координаты активного центра, там происходит реакция. Также даются названия атомов и конформации, а в результате получаем разделение. Сама сетка дискриминирует, какая часть атомов является заряженной, какая незаряженной. На выходе получается энергия, которая учитывает образование катионов, анионов и т. д.

Поясним подробнее. Как вычисляются реакции с помощью квантовой химии? Есть стартовое состояние и конечное. Можно сделать релакс скан в поисках переходного состояния. Накапливаем 20 разных состояний, которые идут через это переходное из реагента в продукт. Для каждого можно посчитать энергию. По оси x будет 20 реакций, где будут отличаться конформации, а по оси y – разные энергии. После этого рисуем профиль.

Можно сделать сложнее. Можно сделать релакс скан разными способами и получить разные профили, и уже из них выбрать тот, у которого минимальная высота барьера.

Ещё можно включить динамику. Всё шевелится с температурой, вся система подогревается, пока не получится переходное состояние. Она нагревается в результате до такой степени, что начинает реагировать: температура позволяет перевалить барьер. В этом случае на каждом шаге молекулярной динамики мы для каждого набора координат вычисляем энергию. И эта энергия суммируется с энергией системы, которая у нас есть для всего белка и растворителя. В результате, учитывая статистику, можно построить профили реакции: на сколько нужно нагреть систему, чтобы реакция прошла.

Силовые поля на основе ML потенциалов

Альтернативное применение моделей – попытка **построить силовые поля**. Они описываются простыми уравнениями: законом Гука, Кулона, потенциалом Леннарда-

Джонса и т. д. Есть sGDML модель, которая позволяет построить хорошее силовое поле, используя небольшое количество выборки. При этом можно получить точность на уровне теории CCSD(T).

Было показано, что данная модель позволяет получать качественные данные, коррелировать геометрии с ЯМР и QM свойствами для различных систем, где много тяжёлых атомов.

Исследование химического разнообразия

Одно из значимых **достижений в этой области** – на уровне точности CCSD(T) машинным обучением был описан инфракрасный спектр аспирина (рис. 6.12). Это обеспечивает хорошую точность при незначительных вычислительных затратах.

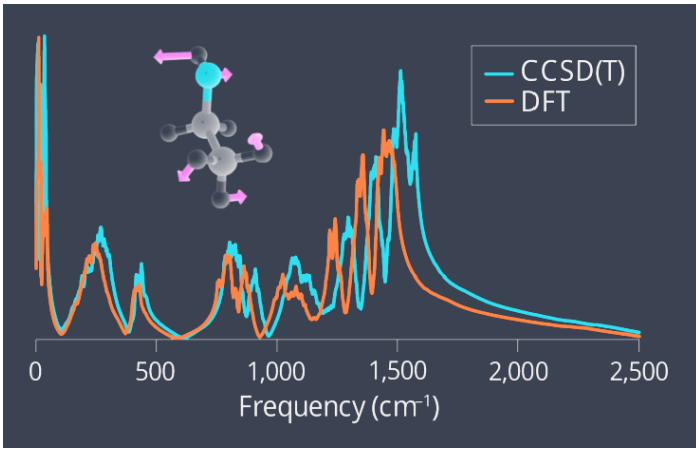

Рис. 6.12. Новые insights-1

Сканируя пространство электронных свойств, которое получается из разных описаний, можно двигаться вверх по наблюдаемому свойству и из этого вычислять, какое должно быть описание, и из этого описания генерить молекулу (рис. 6.13).

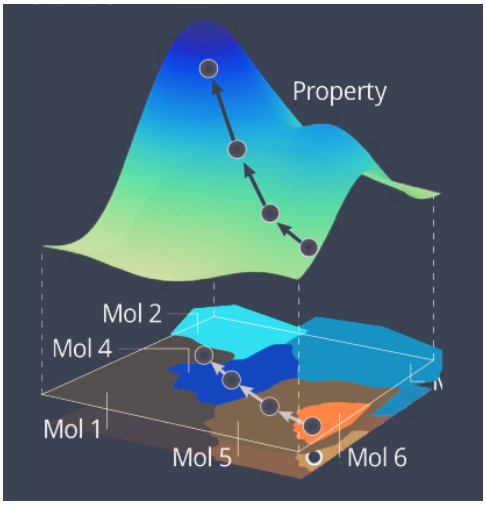

Рис. 6.13. Новые insights-2

В результате были предсказаны новые кандидаты на светопоглощающие материалы. Это типичное применение, потому что поглощение света связано с высвобождением электрона на внешних оболочках. Это можно оптимизировать с помощью QML подходов.

Ещё один успешный вариант применения QML в области материалов представлен на рис. 6.14.

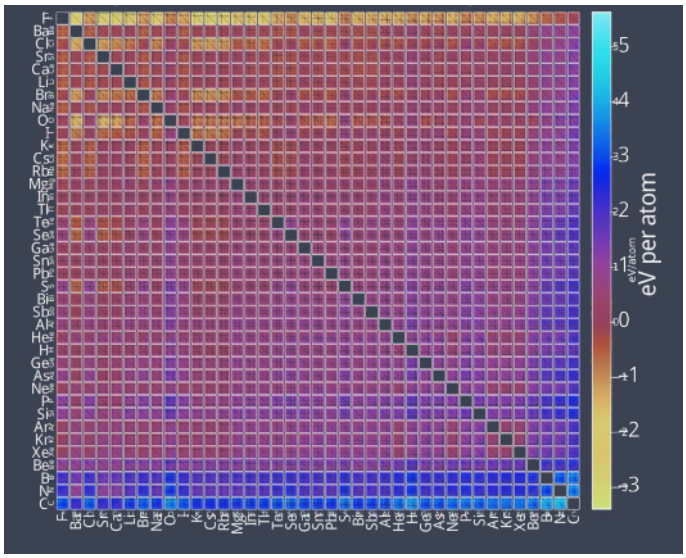

Рис. 6.14. Новые insights-3

Можно генерировать комбинации, сканировать их энергии, получать геометрии и т. д., потому что у нас высокая вычислительная эффективность. В итоге были найдены новые стабильные кристаллы, экспериментально показано, что они существуют, и даже был найден экзотический кристалл, в котором алюминий несёт отрицательную степень окисления.

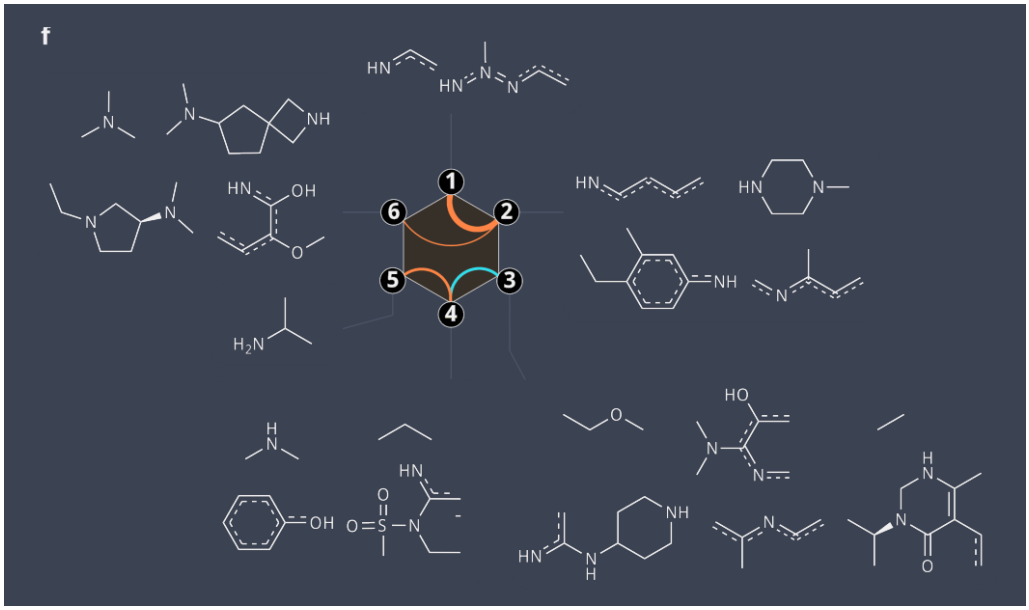

Рис. 6.15. Новые insights-4

Ещё один пример – попытка с помощью машинного обучения определить, какие группировки чаще дают хороший вклад в энергию связывания лиганда с белком, чем другие (рис. 6.15). Отсюда можно сделать целенаправленные заключения для дизайна. Группировки имеют взаимодействия, а они основаны на электронных свойствах.

А ещё молекулярная механика часто не может описывать геометрию некоторых соединений. В этом случае QML может сильно помочь, потому что он подразумевает наличие явной электронной плотности, и это позволяет получать хорошие геометрии.

Перспективы

А **финальная цель** – объединить квантовую механику, статистическую механику и машинное обучение, чтобы реализовать глобальное и универсальное исследование, которое позволит решать очень разные задачи с разным уровнем точности.

Как **QML можно применить к дизайну**? Напрямую это сделать сложно, потому что надо будет генерировать ещё один элемент – генерирующий, так как любой дизайн опирается либо на перебор вариантов, который был сделан в случае кристаллов, либо на абстракцию, которая позволяет генерировать случайные значения и проверять их.

QML можно использовать в дизайне при переборе для оценки наблюдаемых статистической механикой значений. Для экспериментального дизайна QML может давать хорошие идеи, но не давать готовых ответов.

Самое эпическое применение – попытки находить новые реакции с помощью QML, потому что, если мы можем моделировать реакции быстро и качественно с точки зрения оценки теплоты, мы можем перебрать большое количество компонентов в поисках комбинаций, которые приведут к появлению новых реакций со специальными условиями. Это пытаются реализовать в 1960-х годов, но современные скорости и точности вычисления предсказания реакций пока не позволяют использовать данный метод эффективно.

Можно представить в будущем вычислительный реактор, в который помещаем компоненты, и наблюдаем, какие реакции там могут происходить. Такие статьи уже были, они были связаны с метадинамикой на основе ab initio квантовой молекулярной динамики, но подобные вещи работают только для очень небольшого количества атомов. А гораздо интереснее было бы проводить такие исследования с большими молекулами.

ФАКУЛЬТЕТ
БИОИНЖЕНЕРИИ И
БИОИНФОРМАТИКИ
МГУ ИМЕНИ
М.В. ЛОМОНОСОВА

teach-in
ЛЕКЦИИ УЧЕНЫХ МГУ