

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени
М.В.ЛОМОНОСОВА**

ФАКУЛЬТЕТ БИОИНЖЕНЕРИИ И БИОИНФОРМАТИКИ

**Разработка вычислительного конвейера для изменения субстратной специфичности
белковой тирозинфосфатазы LWT-PTP**

Дипломная работа студента 6 курса

Выполнил:

Усманов Р. Д.

_____ *подпись*

Научный руководитель:

д.х.н., Головин А. В.

_____ *подпись*

**Москва
2021г.**

Содержание

1	Список сокращений	3
2	Введение	3
2.1	Общие сведения	3
2.2	Изменение субстратной специфичности в природе	3
2.3	Искусственное изменение субстратной специфичности	3
2.3.1	Классификация методов изменения субстратной специфичности	3
3	Обзор литературы	4
3.1	Изменение субстратной специфичности фермента LMW-PTP	5
4	Цели и задачи	7
5	Материалы и методы	7
5.1	Python	7
5.2	PyMol	7
5.3	Фреймворк Rosetta и <i>RosettaDesign</i>	7
5.3.1	Общие сведения	7
5.3.2	Некоторые объекты в программах фреймворка Rosetta	8
5.4	Программа «Relax»	9
5.5	Программа <i>RosettaDesign</i>	9
5.5.1	Типы файлов	9
5.5.2	Алгоритм	9
5.6	Метод молекулярной динамики (МД)	11
5.7	Молекулярно-динамические моделирования	12
5.8	Конвейер SSDO-0.1	12
5.8.1	Реализация конвейера	12
5.8.2	Формализация множеств атомов и остатков, получаемых из литературы	13
5.8.3	Входные данные конвейера	15
5.8.4	Ход работы	15
6	Результаты и обсуждения	21
7	Заключение	22

1 Список сокращений

PDB – Protein Data Bank, банк данных пространственных структур биомолекул

RMSD – root mean square deviation, среднеквадратическое отклонение атомных позиций

2 Введение

2.1 Общие сведения

Ферменты это белки, способные осуществлять химические превращения. Миллиарды лет эволюции понадобилось природе на то, что бы эти молекулярные машины обрели способность превращать множество субстратов, превосходя в скорости методы современной химической промышленности. Эти машины настолько совершенны, что человечество и по сей день вынуждено использовать их в качестве заготовок для создания катализаторов необходимых химических реакций.

Такую модификацию называют редизайном ферментов. Наиболее простой и широко применяемый способ редизайна является изменение аминокислотного состава белка. Целью данного процесса может быть изменение субстратной специфичности, то есть изменение набора возможных лигандов для реакции.

Многие современные подходы редизайна предполагают подробное изучение системы фермент-субстрат а также применение методов, выбор и применение которых требует квалификации и опыта исследователя. Тем временем современные компьютеры могут производить триллионы операций в секунду и в теории могли бы ускорить процесс дизайна на много порядков. Разработка метода, позволяющего автоматически изменять и оптимизировать субстратную специфичность ферментов и стала целью данной работы.

2.2 Изменение субстратной специфичности в природе

Известно множество случаев изменения субстратной специфичности в процессе эволюции. Это может сопровождаться как точечными мутациями [8], происходящими к тому же с разной скоростью, с потерей гена [16] и т.д. В процессе эволюции происходит огромное количество разнообразных мутаций, большинство из которых нейтральны. Большинство из оставшихся приводит к нарушению функций белка и наконец оставшаяся часть мутаций приводит к увеличению приспособленности организма и может закрепиться.

2.3 Искусственное изменение субстратной специфичности

Среди методов изменения субстратной специфичности можно выделить стохастические, использующие перебор возможных вариантов последовательности и рациональные, использующие информацию о работе фермента. Мы выбрали метод компьютерного рационального дизайна, использующий физические модели макромолекул, так как он гораздо дешевле молекулярно-биологических методов. Основное допущение в этой группе методов в том, что наиболее успешный мутантный фермент будет иметь с лигандом наименьшую возможную свободную энергию системы. Ниже для ознакомления представлена классификация известных методов изменения субстратной специфичности.

2.3.1 Классификация методов изменения субстратной специфичности

1. Стохастические подходы Стохастические методы используют перебор возможных вариантов последовательности с целью найти подходящие результаты.

(а) Стохастические ненаправленные подходы

- Описание
 - Вносятся мутации (точечные) случайным образом в последовательность белка.
 - Происходит испытание на сродство к лиганду

- Отбираются лучшие результаты
 - Преимущества
 - Не требуют знаний о работе фермента
 - Недостатки
 - Дорогостоящие
 - (b) Стохастические направленные подходы
 - Описание
 - Используется информация о ферменте, к примеру область связывания
 - Преимущества
 - Более эффективен, чем ненаправленные подходы
 - Недостатки
 - Требуется знание о работе фермента
2. Рациональный дизайн Рациональный дизайн использует информацию о работе фермента, иногда через специальные физические модели.
- (a) Эмпирический рациональный дизайн
 - Описание
 - Используются данные о работе ферментов, полученные из различных экспериментов (из баз данных), а также экспертные оценки.
 - Преимущества
 - Небольшой размер библиотеки мутантов, часто несколько штук
 - Недостатки
 - Глубокое понимание механизмов ферментативной реакции
 - Плохо воспроизводим
 - (b) Систематический рациональный дизайн
 - Описание
 - Может использовать такие методы, как биоинформатический анализ белков, молекулярное моделирование и автоматизированную обработку данных научных статей.
 - Преимущества
 - Знания полученные данным методом могут ускорить дальнейшие исследования
 - (c) Компьютерный рациональный дизайн с использованием физических моделей
 - i. Данный вид дизайна использует компьютерное моделирование структур комплекса лиганд-белок с помощью физических моделей с целью предсказания поведения данной молекулярной системы в реальности.

3 Обзор литературы

Для изменения субстратной специфичности применяются множество компьютерных методов, которые могут применяться отдельно или сочетаться, в зависимости от поставленной задачи. Одним из таких инструментов является фреймворк Rosetta. В основе методологии большинства его программ лежит стохастический подход белковой инженерии, то есть случайный перебор вариантов и их оценка с целью отбора лучших. Подробнее о методах будет изложено в разделе «Материалы и методы»⁵.

По некоторым оценкам [11] одним из самых распространенных способов рационального дизайна ферментов является рациональный дизайн с помощью программы *RosettaDesign* [21] фреймворка Rosetta. На данный момент известно множество примеров успешного дизайна ферментов этим методом [15]. По этим причинам он был выбран в качестве метода сравнения в данной работе.

Для проверки нашего метода необходимо было повторить уже поставленный эксперимент используя метод сравнения и сопоставить результаты. За оценку эффективности было взято количество активных *in vitro* мутантов в выдаче 10 лучших структур по результатам метода.

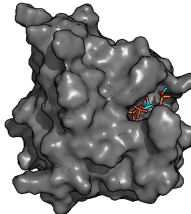
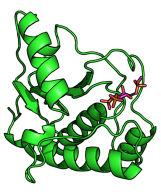
Далее необходимо было проанализировать статьи, ссылающиеся на работу, предлагающую метод сравнения ([21]). Было обнаружено всего 184 таких статьи, из которых только 6 непосредственно использовали алгоритм RosettaDesign [21]. Из этих статей была выбрана работа об изменении субстратной специфичности фермента LMW-PTP, подробнее о котором будет рассказано ниже.

Выбрана она была потому, что в ней представлены данные по активности мутантного фермента с несколькими типами родственных лигандов. Воспроизведение результатов по нескольким лигандам было выбрано, так как оно многократно увеличит качество сравнения методов.

3.1 Изменение субстратной специфичности фермента LMW-PTP

РТР - белки, тирозиновые фосфатазы. Общий вид одного из представителей РТР можно видеть в таблице 1. Имеют каталитический C-XXXXX-R мотив в активном центре (Р-петля) с каталитиче-

Таблица 1: Общий вид белка LMW-PTP из семейства РТР со связанным лигандом $PI[3, 5]P_2$.

LMW-PTP, мол. поверхность	LMW-PTP, проволоочная модель
	

скими остатками цистеина и аргинина а также т.н. D-петлю, содержащую третий каталитический остаток - аспарат (См. Рис. 1).

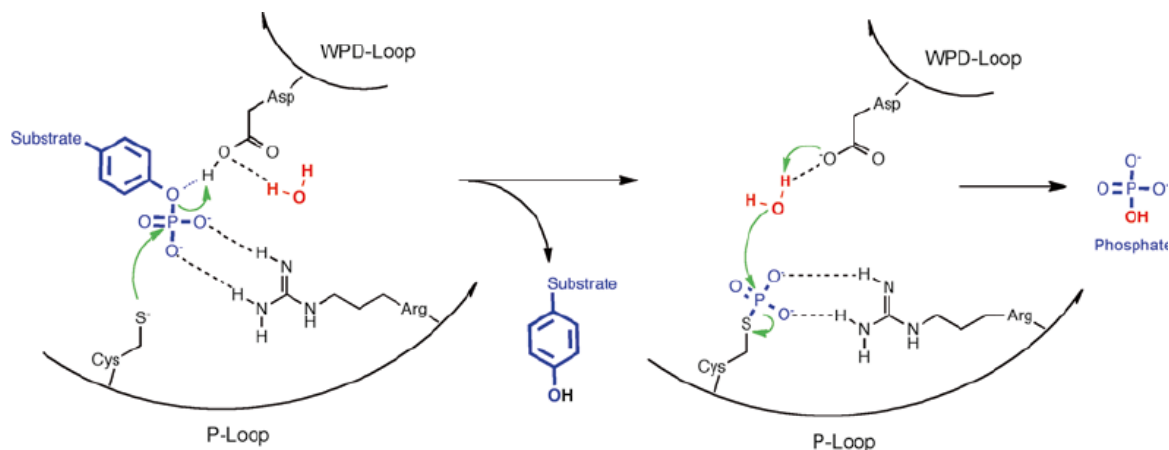

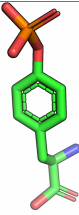
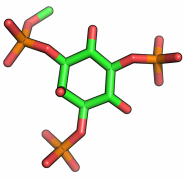


Рис. 1: Предполагаемый каталитический механизм РТР. В механизме реакции белковых тирозин-фосфатаз РТР участвуют три каталитических остатка: цистеин и аргинин на мотиве «Р-петля» снизу, и аспарат сверху.

В работе (См. [12]) была предложена последовательность мутанта легкой тирозиновой фосфатазы LMW-PTP, способной реагировать с лигандом $PI[3, 5]P_2$ вместо фосфорилированного остатка тирозина с помощью программы RosettaDesign.

В своей работе коллеги использовали алгоритм RosettaDesign для LMW-PTP с фосфатидилинозитол 3,4-бисфосфатом (См. Табл. 2) и получили 20 лучших (См. Табл. 3). Для этих результатов они получили распределение мутаций по типам и позициям и рационально выбрали нужные комбинации для проверки. Выбранные мутации были введены в последовательность и проверены *in vitro* (См. Рис. 6).

Таблица 2: Создание фосфоинозитидной с. специфичности в LMW-PTP E.Egbe и коллегами: молекулы *E*, *S* и *X*.

Обозначение	Структура	Наименование
<i>E</i>		LMW-PTP
<i>S</i>		Фосфотирозин
<i>X</i>		$PI[3, 5]P_2$

Имя	LEU13	ILE16	TRP49	ASN50	TYR131	Значение скор-функции (меньше - лучше)
DE ₁₉₄₈	ARG	THR	LYS		ALA	-10.14
DE ₇₄₁	ARG	THR	LYS		ALA	-9.84
DE ₂₅₁	ARG	THR	LYS		ALA	-9.83
DE ₈₃₇	SER	THR	LYS		ALA	-9.52
DE ₁₂₆₉	SER	ALA	LYS		ALA	-9.44
DE ₁₃₂₉	SER	ASN	LYS	SER		-9.33
DE ₇₃₅	SER	THR	ARG		ALA	-9.14
DE ₁₁₂₅	SER	THR	LYS		ALA	-9.11
DE ₈₈	ARG	THR	LYS			-9.11
DE ₈₇₃	ARG	THR	SER		ALA	-8.99
DE ₁₀₈₀	ARG	THR	ASN		ALA	-8.95
DE ₁₅₄₁	ARG	THR	ASN		ALA	-8.95
DE ₁₂₂	SER	THR	LYS			-8.92
DE ₁₂₉₉	SER	THR	LYS		ALA	-8.9
DE ₃₅₉	SER	THR	LYS		ALA	-8.8
DE ₁₃	ARG	THR	LYS		ALA	-8.78
DE ₆₁₂	SER	THR	LYS		ALA	-8.77
DE ₃₆₉	ARG	THR	ARG		SER	-8.76
DE ₁₈₃₃	ARG	THR	ASN		ALA	-8.67
DE ₁₈₈₃	ARG	THR	LYS		ALA	-8.67
...
#2000

Таблица 3: Изменение субстратной специфичности фермента LMW-PTP E.Egbe и коллегами: выдача алгоритма *RosettaDesign*.

Фермент	$pNPP$	$PI(3)P$	$PI(4)P$	$PI(5)P$	$PI(3,4)P_2$	$PI(3,5)P_2$	$PI(4,5)P_2$	$PI(3,4,5)P_3$
WT	1303±6	0.53±0.06	1.12±0.01	нет	нет	нет	нет	нет
L13R	1215±17	4.43±0.21	2.16±0.12	4.44±0.13	нет	0.33±0.01	нет	нет
I16K	1711±25	4.66±0.16	5.04±0.17	1.5±0.08	0.05±0	0.65±0.01	нет	нет
W49K	1639±23	2.67±0.04	нет	нет	нет	0.33±0.02	нет	нет
L13R-I16T (RT)	1423±20	4.43±0.09	3.82±0.26	17.96±0.76	нет	2.5±0.09	нет	нет
L13R-I16K (RK)	940±26	11.99±0.41	15.79±0.46	42.48±3.11	нет	12.77±0.67	нет	нет
L13R-W49K	1385±24	2.94±0.03	нет	5.66±0.17	нет	нет	нет	нет
I16K-W49K	1408±18	3.77±0.19	1.02±0.05	2.41±0.19	нет	0.6±0.05	нет	нет
I16T-W49K	1415±16	0.48±0.03	5.15±0.07	1.75±0.06	0.04±0	2.48±0.14	1.24±0.04	нет
L13R-I16T-W49K (RTK)	1360±20	2.02±0.12	3.18±0.24	11.87±0.24	нет	2.14±0.2	нет	нет
L13R-I16K-W49K (RKK)	858±27	2.37±0.19	3.83±0.17	14.21±0.6	нет	16.94±1.02	нет	нет

Таблица 4: Изменение субстратной специфичности фермента LMW-PTP E.Egbe и коллегами: активность (нмоль/мин/мг) мутантных ферментов с различными субстратами.

4 Цели и задачи

Целью данной работы являлось разработка эффективного конвейера изменения и оптимизации субстратной специфичности SSDO (Substrate Specificity Design and Optimization pipeline) белковой тирозинфосфатазы LMW-PTP. Для этого были поставлены следующие задачи:

1. Проанализировать известные способы изменения и оптимизации субстратной специфичности, изучить преимущества и недостатки.
2. Разработать конвейер SSDO.
3. Реализовать его версию на языке Python.
4. Использовать запуск конвейера для предсказания мутантной LMW-PTP, которая бы реагировала с лигандом $PI[3,5]P_2$.

5 Материалы и методы

5.1 Python

Конвейер был реализован на языке программирования Python [25].

5.2 PyMol

Визуализация структур была произведена с помощью программы PyMol [22].

5.3 Фреймворк Rosetta и *RosettaDesign*

5.3.1 Общие сведения

Фреймворк Rosetta была изначально разработана для предсказания структуры белков de novo. Достигнув в этой области результатов, она стала применяться для решения других задач: de novo дизайн белков, докинг и моделирование петель. Предсказание и дизайн пространственных структур макромолекул основывается на предположении о том, что в естественных условиях молекула пребывает почти всегда в конформации с наименьшей свободной энергией [6]. Для количественной оценки энергии моделируемой системы вводится т.н. энергетическая функция (скоровая функция, скор-функция, оценочная функция). Она отображает фазовое пространство возможных конформаций (конформационное пространство) на действительную прямую. Получается, что конформации, обладающие наименьшей свободной энергией должны (в случае идеальных расчетов энергии) соответствовать глобальному минимуму скор-функции (далее, конформация глобального минимума: КГМ).

Так как подвижность белка в основном обусловлена вращением вокруг торсионных углов, то в программах Rosetta другие степени свободы не учитываются. Для нахождения глобального минимума скор-функции необходимо использовать метод семплирования, выбирая на поверхности

энергии случайные точки, а затем применять методы локальной минимизации энергии, основанные на подсчете частных производных первого и второго порядков.

5.3.2 Некоторые объекты в программах фреймворка Rosetta

1. *Поза* Объект *Поза* в Rosetta3 хранит информацию о структуре и состоянии моделируемой молекулярной системы (к примеру белок и лиганд). Он хранит информацию о составе и степенях свободы системы (ССС). Степени свободы делятся на внешние (смещение и вращение) и внутреннее (изменения торсионных углов). Он может получать инструкции о том, какие изменения должны быть сделаны в системе и соответственно изменять позиции нужных атомов. Для того, чтобы не пересчитывать по несколько раз значения энергии вводится объект Кэш. *Поза* «знает», как эффективно оценивать энергию системы через данные Кэша, основываясь на обновленных положениях атомов. *Поза* обычно инициализируется из входной структуры.
2. *Конформация* Объект *Конформация* отвечает за преобразование любых изменений по СССР в фактические изменения декартовых координат атомов. Этот объект имеет три категории данных:
 - (a) Химическое представление системы: Каждая *Конформация* разбита на структурные единицы (напр. аминокислоты и лиганд), представляемые объектами типа *Остаток*. Объекты типа *Остаток* содержат информацию о химическом составе и типе атомов, их связности, так и пространственных положениях. Информация представляется в виде файлов с расширением *.params*.
 - (b) Кинематическое представление системы: Вводится древовидная топология, объединяющая внутренние и внешние СССР: объекты *Fold-tree* и *Atom-tree*. Используя эти объекты любые изменения в СССР могут быть транслированы как в декартовы координаты, так и во «внутренние» координаты.
3. *Энергия* Объект *Энергия* производит оценку энергии *Позы* (используя данные Кэша) пользуясь скор-функцией.
4. *Скор-функция* Вследствие возможности вращения по одинарным связям молекула может принимать различные конформации. Конформерами далее будем называть только относительно низко-энергетические такие конформации. Ротамерами будем называть изомеры, полученные путем вращения только по торсионным углам некоторой исходной структуры. Объект *Скор-функция* рассчитывает значение энергии Конформации через скор-функцию. Последняя представляет собой взвешенную сумму компонент энергии. К таким компонентам можно отнести ленард-джонсовские, сольватационные, компоненту водородных взаимодействий, компоненту вероятности встречи данного ротамера, компоненты вандер-ваалсовых взаимодействий и многие другие. Значение всех параметров компонент и значения весов подгоняются эмпирически, часто используется банк PDB.
5. *Минимизатор* Объект *Минимизатор* производит поиск локального минимума скор-функции *Позы*. Данную процедуру будем называть *локальной минимизацией*.
6. *Паковщик* Для каждой аминокислоты есть набор типичных ротамеров, которые к примеру, могут наиболее часто встречаться в PDB. Такие наборы называются ротамерными библиотеками. Они используются *Паковщиком* для оптимизации скор-функции путем семплирования торсионного пространства *Позы*. Последнее обычно очень велико, в связи с чем используются различные стохастические алгоритмы. Изменение ротамера остатка (остатков) в процессе моделирования будем называть *перепакровкой*.

5.4 Программа «Relax»

Программа «Relax» проводит поиск глобального минимума скор-функции. В основе алгоритма лежит чередование локальной минимизации *Позы* и изменения ротамеров её остатков на случайно выбранные. Данную процедуру принято называть *релаксацией*.

5.5 Программа *RosettaDesign*

RosettaDesign (*RosettaDesign*) — программа фреймворка Rosetta, алгоритм создания нового фермента с новыми функциями основываясь на данных о структуре некоторого исходного фермента (ферментов) или его части (частей) [19].

5.5.1 Типы файлов

Для запуска *RosettaDesign* требуются следующие файлы:

- *.params* файл содержит химическую информацию о лиганде
- *.resfile* файл описывает множества остатков трех типов:
 1. остатки, которые можно мутировать на любую каноническую аминокислоту кроме цистеина в процессе моделирования (1)
 2. остатки, которые нельзя мутировать, но можно паковать (изменять их ротамер) в процессе моделирования (2)
 3. остатки, которые фиксируются в процессе моделирования (3)
- *.cst* файл — файл геометрических ограничений. Остатки активного центра обычно можно разделить на два типа: остатки, отвечающие за химические стадии реакции (так называемые каталитические остатки) и остатки, ответственные за связывание субстрата. Оценка того, насколько благоприятна для катализа некоторая конформация активного центра, требует расчетов на квантовом уровне.

Эмпирическая скор-функция Розетты не может выделить конформации, благоприятствующие катализу. Для решения этой проблемы вводятся ограничения, то есть правила, по которым можно будет узнать, находится ли данный каталитический остаток в некаталитической конформации.

На практике каталитические ограничения представляют собой набор расстояний, углов и торсионных углов между каталитическим остатком и субстратом (См. Рис. 2). Эти значения могут быть получены из квантово-механических расчетов реакции, кристаллических структур фермента в комплексе с субстратом или аналогом переходного состояния. В третьей, четвертой и пятой колонке нотации файла ограничений можно видеть x_0 , x_{tol} (возможное отклонение) и k параметры соответственно.

Энергетический штраф в *RosettaDesign* будет наложен так: 0 , если $|x - x_0| < x_{tol}$ и $k * (|x - x_0| - x_{tol})$ в противном случае. Таким образом, применяются гармонические ограничения.

5.5.2 Алгоритм

Вариант алгоритма *RosettaDesign*, в котором остов *Позы* остается неподвижным состоит из 4 этапов:

1. Определение к какому из трех типов по классификации из раздела 5.5.1 можно отнести каждый остаток
2. Оптимизация каталитических взаимодействий
3. Циклы дизайна/минимизации
4. Релаксация структуры

1. Определение к какому из трех типов нужно отнести каждый остаток Для этого есть два способа: использование стандартного файла *.resfile* для Rosetta или же автоматическое определение. В модели фреймворка Rosetta остатки белка могут быть поделены на 5 групп по расстоянию от лиганда (в порядке возрастания). Им соответствуют параметры cut1, .. cut4 программы:

- Все аминокислоты боковой цепи, C α атом которых находился ближе cut1Å от любого атома лиганда будут отнесены к типу №1 (1)
- Все аминокислоты боковой цепи, C α атом которых находился ближе cut2Å от любого атома метки, а вектор C α - C β указывает в сторону лиганда будут отнесены к типу №1 (1)
- Все аминокислоты боковой цепи, C α атом которых находился ближе cut1Å от любого атома лиганда будут отнесены к типу №2 (2)
- Все аминокислоты боковой цепи, C α атом которых находился ближе cut2Å от любого атома метки, а вектор C α - C β указывает в сторону лиганда будут отнесены к типу №2 (2)
- Остальные остатки будут отнесены к типу №3 (3)

CST::BEGIN

TEMPLATE:: ATOM_MAP: 1 atom_name: N2 C18 N3

TEMPLATE:: ATOM_MAP: 1 residue3: RRF

TEMPLATE:: ATOM_MAP: 2 atom_name: OD1 CG CB

TEMPLATE:: ATOM_MAP: 2 residue3: ASN

CONSTRAINT:: distanceAB: 3.05 10.0 50.00 0 0

CONSTRAINT:: angle_A: 120.22 100.00 50.00 360.00 1

CONSTRAINT:: angle_B: 118.93 100.00 50.00 360.00 2

CONSTRAINT:: torsion_A: -14.77 100.00 50.00 360.00 3

CONSTRAINT:: torsion_B: 98.45 100.00 50.00 360.00 4

CONSTRAINT:: torsion_AB: 172.36 100.00 50.00 360.00 5

CST::END

Рис. 2: Пример файла геометрических ограничений.

2. Оптимизация каталитических взаимодействий Оптимизация каталитических взаимодействий может быть проведена двумя способами:

- А Перед дизайном проходит градиентная минимизация структуры. Во время этой минимизации все не-каталитические остатки активного центра мутируются в аланин, а также применяется энергетическая функция без компонент (т.н. термы) ван-дер-ваалсового отталкивания и сольватации. Целью этой стадии является перемещение субстрата в положение, при котором каталитические взаимодействия становятся наиболее идеальными.
- В Альтернативой градиентной оптимизации является метод твердотельного семплирования типа Монте-Карло. Флаги *trans_{magnitude}* (в Å) и *rot_{magnitude}* (в градусах) задают максимально возможное смещение и поворот лиганда соответственно.

3. Циклы дизайна/минимизации Для позиций типа №1 5.5.1 применяется Монте-Карло алгоритм выбора последовательности, отвечающей более низкоэнергетической конформации. Каталитические ограничения применяются на протяжении всей моделирования. После чего структура подвергается минимизации. Процедуры повторяются несколько раз.

4. Релаксация структуры На этой стадии каталитические ограничения выключаются. Проводится релаксация.
5. Выходные файлы *RosettaDesign* Структуры, получаемые после применения вышеуказанного алгоритма называются *декоями* и сохраняются программой в формате PDB в указанную директорию. Если указан флаг `-out:file:o scorefile.txt`, то в файл *scorefile.txt* будут сохранена таблица, строкам которой соответствуют файлы декоев, а столбцам — значения членов энергетической функции (также некоторые их суммы), номера каталитических остатков и некоторые другие данные. С помощью скор-функции можно подсчитать попарные значения скор-функции во всевозможных парах X, Y , где X — остаток множества R_1 , а Y — остаток множества R_2 . Складывая полученные значения можно получить величину, называемую *интерфейсовой энергией* для множеств R_1 и R_2 . Далее этот термин будет употребляться в значении интерфейсовой энергии для множества всех остатков фермента (R_1) и лиганда (R_2). Все данные, приводимые о результатах *RosettaDesign* будут представлять собой обработанные данные файлов декоев и *scorefile.txt*.

5.6 Метод молекулярной динамики (МД)

Метод молекулярной динамики — метод, в котором временная эволюция системы взаимодействующих атомов или частиц отслеживается интегрированием их уравнений движения [14].

Силы межатомного взаимодействия представляются в форме градиента потенциальной энергии системы. Согласно модели МД совокупность конформаций, генерируемых в ходе работы алгоритма МД распределены в соответствии с некоторой статистической функцией распределения (напр. микроканонический ансамбль).

Для учета энергообмена с внешней средой используются другие алгоритмы: т.н. термостаты соответственно для каждого типа термодинамических ансамблей. Взаимодействия между атомами моделируются с помощью потенциала Леннарда-Джонса и кулоновского потенциала.

Внутримолекулярные взаимодействия моделируются как сумма вкладов растяжения валентных связей, изменения валентных углов, торсионных взаимодействий, изменения двугранных углов и проч. Электростатические взаимодействия моделируются методом Эвальда. Гамильтонианом называют сумму значений кинетической и потенциальной энергий всех частиц некоторой системы.

Гамильтониан для системы атомов может быть приближен различными способами. В методе МД гамильтониан представлен суммой с коэффициентами, составляющие которой отвечают за различные взаимодействия (см. выше). Сами взаимодействия также моделируются при помощи потенциалов, в которые входят коэффициенты.

Данные коэффициенты оптимизируются для конкретных типов систем (к примеру нуклеиновых кислот). Также данные коэффициенты могут подвергаться масштабированию (т.н. скалирование) для управления величинами вкладов тех или иных взаимодействий. Система в методе МД — модель реальной молекулярной системы, имеющей пространство возможных конформаций (имеется некоторое распределение). Каждая точка данного пространства обладает некоторой свободной энергией, которую достаточно тяжело вычислить на прямую.

Данная энергия может быть приближенно вычислена используя встречаемость данной конфигурации в распределении. Чем выше встречаемость, тем ниже свободная энергия. Пространство конформаций континуальное (его дискретизируют) и обладает большой размерностью, соответствующей количеству степеней свободы. Вторую проблему решают введением 2-3 переменных (т.н. коллективных переменных). Система МД попадает в локальные минимумы свободной энергии в вышеуказанном пространстве и долгое (экспоненциальное от величине минимума) время в нем остается. Данная задержка сильно замедляет процесс исследования пространства конформаций.

Эволюция системы в среднем моделируется на временных отрезках порядка микросекунд, в то время как важные с биологической точки зрения события могут происходить на интервалах порядка миллисекунд. Поэтому было предложено множество модификаций алгоритма, ускоряющих получение необходимой информации о модели. К одному из методов относится МД с обменом ре-

пликами (REMD, Replica Exchange Molecular Dynamics), в котором МД реализуется в нескольких системах сразу (т.н. репликах).

Между репликами возможен обмен конформациями с некоторой вероятностью, зависящей от взвешенной разности значений соответствующих гамильтонианов. Также существуют модификации REMD, в которых изменяются т.н. виртуальные температуры частей системы (T-REMD). Это происходит за счет изменения некоторых параметров слагаемых функции потенциальной энергии атомов модели. В методе REST2 (Replica Exchange with Solute Tempering) виртуальная температура изменяется только у некоторых атомов растворенных молекул системы [17]. Метод MD был использован посредством программы `mdrun` пакета GROMACS [5]. В работе использовалась реализация метода REST2 в плагине Gromacs PLUMED [9], называемая HREX.

5.7 Молекулярно-динамические моделирования

Общая схема молекулярно-динамических симуляций в данной работе представлена ниже. Она реализована с использованием кода консорциума BioExcel [7] и основана на обучающих материалах этого проекта в сети интернет [3].

- В директории запуска оказывались структуры лиганда *lig.mol2* и фермента *prot.pdb*.
- Файл лиганда *lig.mol2* подавался на вход скрипту *ascpure.py* [10] для расчета зарядов и подготовки файла топологии *unl_GMX.itp* и координатного файлов лиганда *UNL_GMX.gro*.
- С помощью функции *pdb2gmx* из *prot.pdb* генерировался файл *prot_pdb2gmx.gro*.
- Координатные файлы и файлы топологии лиганда и фермента затем объединялись.
- Затем генерировалась додекаэдрическая ячейка с отступом 10Å с помощью *gmx editconf*.
- В систему добавлялась вода (*gmx solvate*). Молекулы воды имели множественные клеши ферментом, поэтому скриптом удалялись все молекулы воды в радиусе 2.9Å от молекулярной поверхности фермента.
- Моделирование с минимизацией энергии с помощью метода градиентного спуска (1200 итераций).
- Наложение потенциалов удержания на лиганд
- NVT-симуляция (500 итераций)
- NPT-симуляция (500 итераций)
- Удаление потенциалов удержания
- Запуск молекулярно-динамической моделирования
- Центрирование полученной траектории (*gmx trjconv*) по координатам белка
- Анализ RMSD остатков активного центра а также анализ водородных взаимодействий на протяжении всей траектории были проведены с помощью библиотеки MDAnalysis [20].
- Визуальный анализ полученной траектории

5.8 Конвейер SSDO-0.1

5.8.1 Реализация конвейера

Конвейер реализован в виде скрипта на языке python 3 [25] в объектно-ориентированном стиле. Код выложен на github пока как частный репозиторий и доступен по [ссылке](#), за доступом необходимо обращаться к автору.

Конвейер может быть запущен только на операционной системе из семейства **Unix**. Для мониторинга прогресса по ступеням конвейера был написан скрипт на языке Emacs lisp. Запуская его в программе **Emacs** на компьютере пользователь сможет в режиме реального времени следить за статусом директорий-узлов конвейера, как показано на рис. 3. Скрипт доступен через ссылку на частный репозиторий на сайте [github](#), за доступом необходимо обратиться к автору.

```

STARTUP: content
* HR_1_1 [run_spores] [__] [ENDED] [__] [2021-05-28 10:47:55.091408] [07:0:00:02.976863]
* status_history...
* HR_2_1 [make_vdw] [__] [ENDED] [__] [2021-06-02 15:23:59.782929] [07:0:00:22.867295]
* status_history...
* HR_4_1 [prep_complex_structure] [__] [STARTED] [__] [2021-05-31 08:52:00.328326]
* status_history...
* HR_8_1 [heat_atoms] [__] [STARTED] [__] [2021-05-31 09:57:15.369247]
* status_history...
* HR_5_1 [gpus] [__] [ENDED] [__] [2021-05-31 09:16:26.245030] [07:0:00:00.800337]
* status_history...
* HR_1_2 [prep_lig_restraints] [__] [STARTED] [__] [2021-05-31 10:40:46.954966]
* status_history...
* HR_2_2 [make_and_analyze_cv_table] [__] [STARTED] [__] [2021-05-31 08:50:52.609628]
* HR_2_2 [prep_complex_structure] [__] [STARTED] [__] [2021-05-31 08:49:25.154883]
* status_history...
* HR_7_1 [lpt] [__] [ENDED] [__] [2021-05-31 09:37:29.256552] [07:0:00:00.519342]
* status_history...
* HR_1_3 [make_vdw] [__] [ENDED] [__] [2021-06-02 15:20:47.174332] [07:0:00:00.185571]
* status_history...
* HR_2_3 [gpus] [__] [STARTED] [__] [2021-05-31 09:57:41.169718]
* status_history...
* HR_10_1 [gpus] [__] [STARTED] [__] [2021-05-31 10:36:43.403894]
* status_history...
* HR_8_2 [make_md] [__] [STARTED] [__] [2021-05-31 09:26:33.348241]
* status_history...
* HR_12_1 [stop_srv] [__] [ENDED] [__] [2021-05-31 11:51:46.458520] [07:0:00:00.008220]
* status_history...
* HR_1_4 [append_ligand] [__] [STARTED] [__] [2021-05-31 01:45:11.571533]
* status_history...

```

Рис. 3: Инструмент для контроля и визуализации прогресса по конвейеру в программе Emacs.

5.8.2 Формализация множеств атомов и остатков, получаемых из литературы

Координацией молекулой M1 молекулой M2 будем называть совокупность полярных взаимодействий между атомами этих двух систем.

Существуют ферменты, в которых ускорение реакции достигается за счет создания определенной геометрии атомов участников реакции, без образования ковалентных взаимодействий между атомами белка и лиганда. Фермент связан с лигандами S , $L-1$... $L-N$. Происходит химическая реакция, в течение которой координация атомов S атомами E остается постоянной. Успех реакции зависит от многих факторов.

Для понимания роли различных факторов в успехе реакции проведем мысленный эксперимент: К полному отсутствию реакции будет приводить:

- замена некоторых аминокислот E
- замена некоторых атомов S
- аминокислот E и замена некоторых атомов S

Реакция сможет продолжаться (скорость может измениться):

- замена некоторых аминокислот E
- замена некоторых атомов S
- замена некоторых аминокислот E и замена некоторых атомов S

В теории путем перебора здесь мы можем получить новый фермент, катализирующий реакцию с другим субстратом по другому механизму, что нам не требуется. Для сужения рассматриваемой картины начнем учитывать предполагаемый механизм реакции. Часто там приведена следующая информация:

- предполагаемые полярные взаимодействия между некоторым набором атомов S и некоторым набором атомов E . В случае водородных взаимодействий учитываются тяжелые атомы. Возьмем все такие пары атомов и обозначим их как $E_S_пол_вз_пары$.

Обозначим множество атомов $E_S_пол_вз_пары$, относящиеся только к ферменту, как $E_S_кат_поляр_пары_E_ат$, а только к лиганду S как $E_S_кат_поляр_пары_S_ат$. Обозначим остатки белка, содержащие $E_S_пол_вз_пары_E_ат$ как $E_S_кат_поляр_пары_E_ост$.

- изменения электронной плотности у некоторого набора атомов S и некоторого набора атомов E . К примеру нуклеофильная атака одного атома на другой, часто обозначаемая стрелкой.

Обозначим $E_кат_ат$ вовлеченные атомы белка

Обозначим $S_кат_ат$ вовлеченные атомы лиганда

Аналогично предыдущему пункту обозначим $E_S_кат_пары$ за множество взаимодействующих атомов в парах из $E_кат_ат$ и $S_кат_ат$.

Обозначим $E_кат_ост$ аминокислоты E , содержащие $E_кат_ат$.

Если в механизме приведена дополнительная информация (к примеру гидрофобные взаимодействия), то данный конвейер не сможет это учесть, что неизбежно скажется на качестве результатов.

Также часто приведена информация о том, что мутации в некоторых позициях приводили к получению неактивного фермента. Также в эту группу для удобства можно включить остатки, мутации в которых могли бы нарушить катализ по любым причинам. Обозначим остатки в данных позициях как $E_мут_искл_ост$. Они не будут подвергаться мутациям так как мутационное пространство аминокислотных остатков обычно очень велико.

Итак, в результате анализа литературы мы получили следующую информацию (из которой можно вывести другие обозначенные множества):

- $E_S_пол_вз_пары$
- $E_S_кат_вз_пары$
- $E_мут_искл_ост$

Были введены следующие ограничения:

- Атомы S , которые должны сохраняться ($S_сохр_ат$):
 - $S_кат_ат$
 - $E_S_пол_вз_пары_S_ат$
- Остатки E , которые должны сохраняться ($E_сохр_ост$):
 - $E_кат_ост$
 - $E_мут_искл_ост$

Таким образом, вводимый лиганд X должен содержать атомы, играющие каталитическую роль в нативном лиганде. В таком случае, обозначим их соответственно:

- $X_кат_ат$, входящие в $E_X_кат_пары$
- $E_X_пол_вз_пары_X_ат$ Заметим, что наличие модели структуры комплекса ES не требуется. В файле нативного комплекса конвейер удалит все структуры, кроме структуры фермента.

В ходе работы конвейера различий в обработке данных между $E_X_кат_пары$ и

$E_X_пол_вз_пары_X_ат$ принципиально нет, поэтому далее они будут объединены под названием $E_X_пол_кат_пары$. Данное множество содержит пары атомов, атомы которых находятся в момент начала реакции на некотором расстоянии друг от друга. Это расстояния ($E_X_пол_кат_пары_дист$) суть одни из входных параметров конвейера. По умолчанию они будут приняты за **3.0**. Расстояние нуклеофильной атаки или длина водородной связи в реальности могут несколько отличаться от оценки.

5.8.3 Входные данные конвейера

Для работы конвейера необходимы структуры фермента и лиганда в форматах *.pdb*. Также необходимо указать пары атомов, взаимодействие которых важно для катализа и остатки, которые нельзя подвергать мутациям. Еще необходимо указать заряд вводимого лиганда. Остальные параметры имеют значения по умолчанию.

- Модель фермента E_{pdb} - файл фермента в формате *.pdb*
- Модель лиганда X_{pdb} - файл лиганда в формате *.pdb*
- Информация из литературы

— *$E_X_пол_кат_пары$* Пары удобно вводить в следующем виде:

```
E_X_пол_кат_пары = [  
    [[<Номер остатка 1-го атома (E)>, '<Имя 1-го атома (E)>' ],  
     [<Номер остатка 1-го атома (X)>, '<Имя 1-го атома (X)>' ]],  
    ...  
    [[<Номер остатка N-го атома (E)>, '<Имя N-го атома (E)>' ],  
     [<Номер остатка N-го атома (X)>, '<Имя N-го атома (X)>' ]],  
]
```

- *$E_мут_искл_ост$*

```
E_мут_искл_ост = [  
    <Номер исключаемого 1-го остатка (E)>,  
    ...  
    <Номер исключаемого N-го остатка (E)>,  
]
```

- Заряд вводимого лиганда (обозн. *X*) Число, показывающее суммарный заряд молекулы *X*
- (Опционально) SMILES нотация лиганда Используется для проверки правильности протонирования модели *X*
- Входные параметры Различные параметры для корректной работы программ. Имеются значения по умолчанию.

5.8.4 Ход работы

Работа конвейера сопровождается 5 крупномасштабными запусками программ с последующим отбором результатов по некоторым параметрам (далее: ступени). Данные запуски производятся в директориях файловой системы *Linux*.

Эти директории представлены в программе как объекты (далее: узлы), содержащие атрибуты и методы. Один из методов **self.launch()** отвечает за запуск одной конкретной ступени.

В результате получаем древовидную структуру. В корне стоит корневой узел, соответствующий директории, в которой находятся исходные структуры. Его ступень принята за ноль. Каждый узел кроме корневого имеет ровно один материнский узел и хотя бы один (кроме конечных) дочерний узел.

Имена узлов и соответствующих директорий определяются с помощью нотации:

<Мнемоника узла>_<Порядковый номер узла>__<Ступень узла>

Для каждой ступени были придуманы мнемоники:

Степень узла	Мнемоника узла	Описание узла
1	PL	Докинг программой PLANTS
2	HR	REST2 молекулярная динамика
3	ED	Дизайн фермента программой <i>RosettaDesign</i>
4	HR	REST2 молекулярная динамика
5	PL	Докинг программой PLANTS

Так как количество директорий растёт экспоненциально с увеличением номера конечной ступени конвейера (См. Рис. 4), а вычислительные ресурсы ограничены была создана система общей очереди с приоритетом.

Приоритет был указан как номер ступени. Чем меньше номер ступени, тем выше соответствующий приоритет.

Детали работы конвейера подробно описаны в главе о результатах (6).

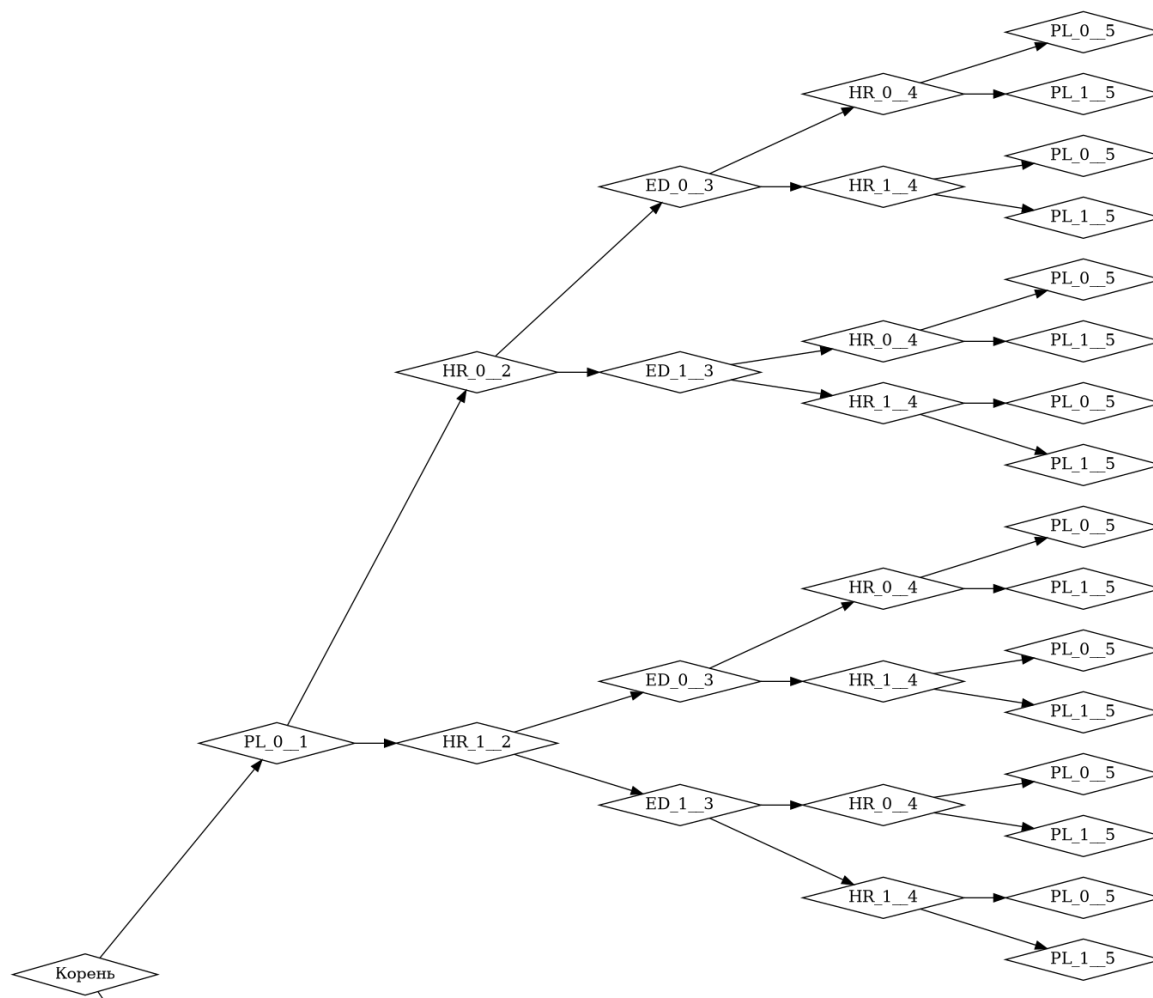


Рис. 4: Визуализация количества директорий в конвейере SSDO-0.1

1. Степень №0 Подготовка реагентов Структуры поданные на вход протонируются. Для лиганда также производится расчет зарядов. Полученные структуры подаются в алгоритм докинга (См. Рис. 5).
2. Степень №1 Докинг и внесение мутаций Полученные структуры подаются в алгоритм докинга. Затем программа PLANTS [18] используется для докинга протонированной структуры лиганда в протонированную структуру белка, полученных из прошлого шага (См. Рис. 6).

На этом этапе применяются ограничения на расстояния между атомами пар *E_X_пол_кат_пары* типа *protein_ligand_distance_constraint*.

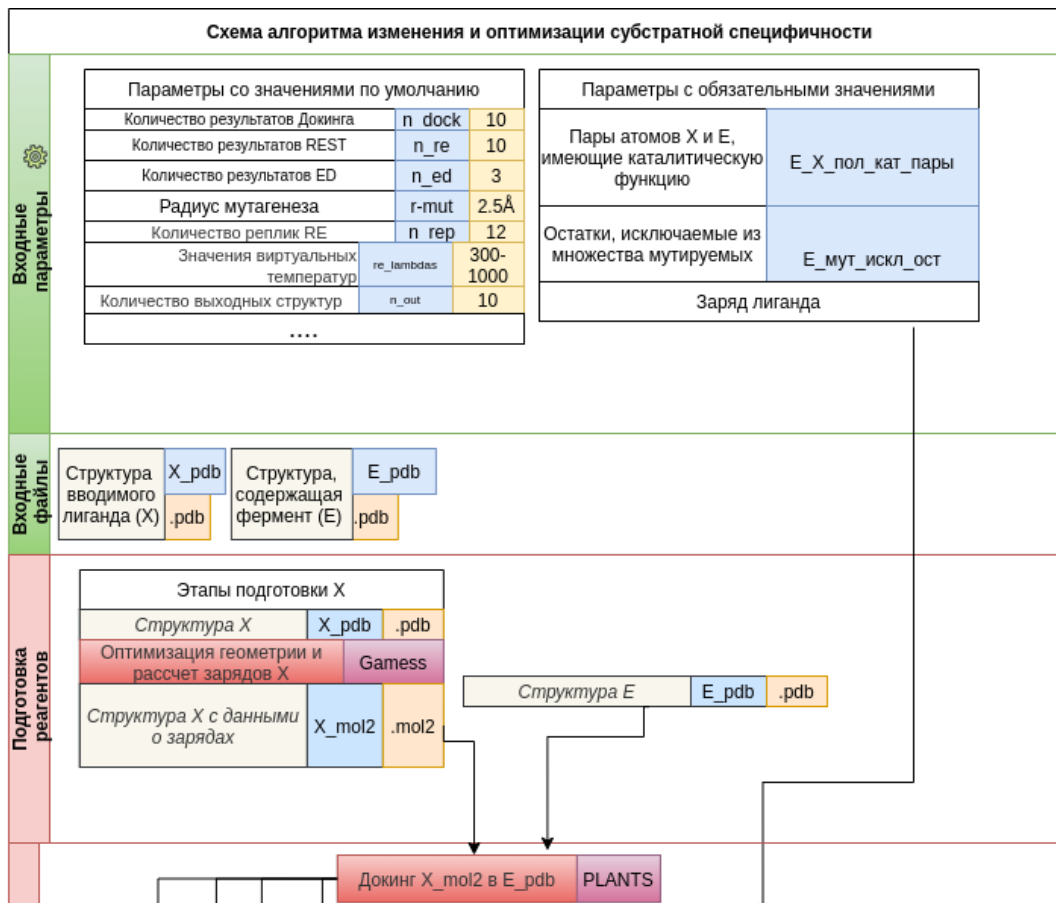


Рис. 5: Схема конвейера SSDO-0.1: подготовка реагентов.

Данные ограничения являются штрафом, добавляемым к скоровой функции системы при нарушении условий (расстояния между атомами).

Расстояние констрейнтов берется за *E_X_пол_кат_пары_дист*.

В SSDO-0.1 PLANTS запускается в директории узла с количеством результатов 10. В 10 лучших структурах происходит замена на остатки аланина остатков вокруг задаваемого порога, если они не входят в запрещенное множество. Полученные структуры подаются на вход алгоритму молекулярной динамики REST2. 10 лучших результатов переходят на следующий этап. Мутации вносятся по следующей схеме:

```
{Мутируемые остатки} =
{
  Остатки атомов
  {Атомы ВОКРУГ 2.5Å ОТ {множество атомов X}}
  И НЕ
  (PRO или ALA или GLY)
}
```

3. Ступень №2 Молекулярная динамика REST2 На второй ступени проводится молекулярная динамика типа REST2 для результатов первой ступени (См. Рис. 7). Для сохранения предреакционной геометрии вводятся потенциалы удержания.

В процессе подготовки записываются т.н. дистанционные рестрейнты, добавляющие потенциал на взаимодействующие атомы [2] (См. Табл. 5). Параметр *type* был указан как **2**, что гарантирует действие потенциала на протяжении всей траектории. Параметры *low*, *up1* и *up2* отвечают за границы допустимых значений расстояния между атомами. Для них были выбраны значения, близкие к возможным длинам водородных связей.

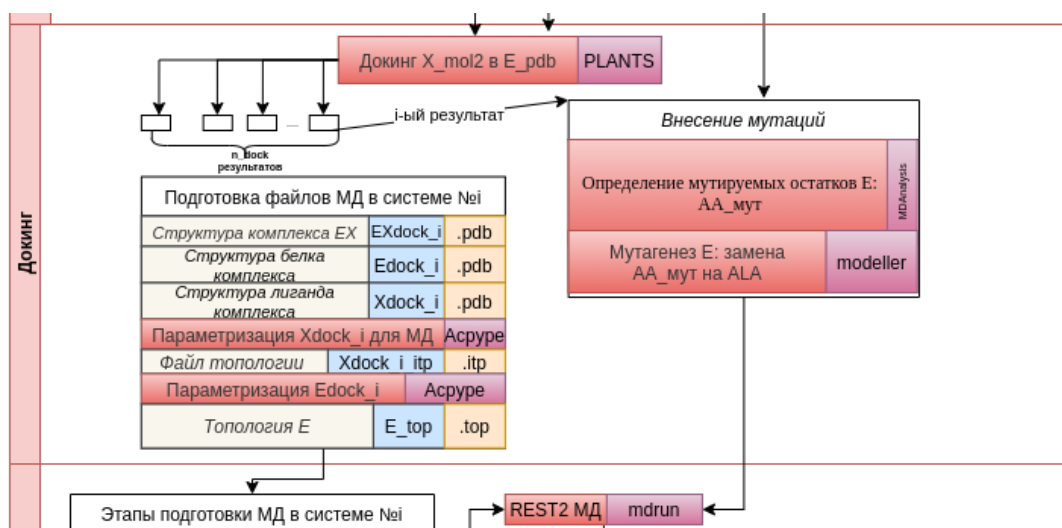


Рис. 6: Схема конвейера SSDO-0.1: Докинг и внесение мутаций.

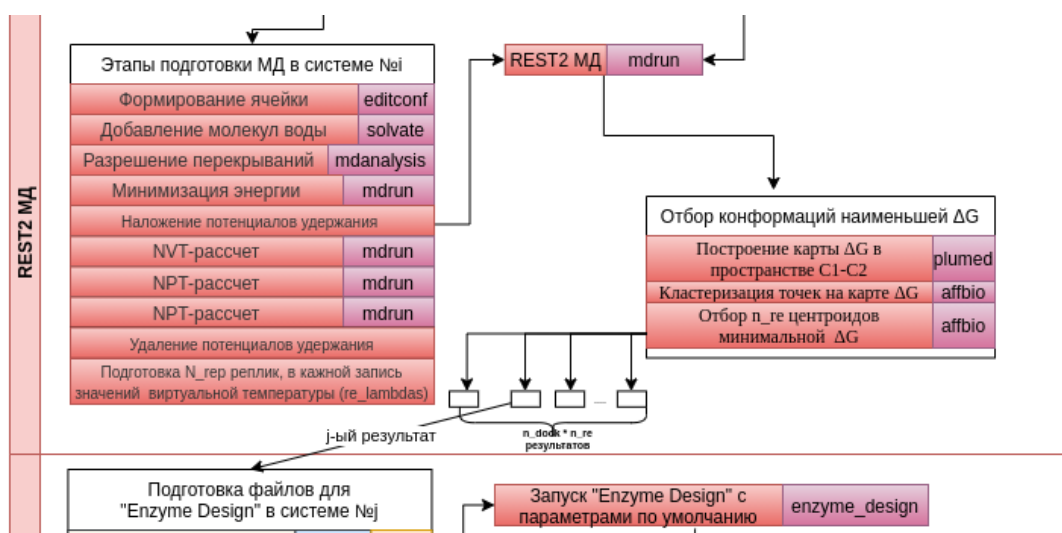


Рис. 7: Схема конвейера SSDO-0.1: Молекулярная динамика REST2 второй ступени

Таблица 5: Пример дистанционных рестреинтов.

ai	aj	type	index	type'	low	up1	up2	fac
249	2462	1	0	2	0.28	0.3	0.35	1.0
2025	2483	1	1	2	0.28	0.3	0.35	1.0
269	2482	1	2	2	0.28	0.3	0.35	1.0
272	2484	1	3	2	0.28	0.3	0.35	1.0

Молекулярная динамика запускается в 4 поддиректориях со значениями параметра виртуальной температуры от 300 до 1000. Для анализа берется директория с минимальной виртуальной температурой. Далее вычисляются значения двух коллективных переменных: CV1 (среднеквадратическое отклонение атомных (RMSD позиций атомов лиганда после выравнивания позиций атомов белка относительно стартовой структуры) и CV2 (мера координации, COORDINATION, атомов лиганда атомами фермента) инструментом PLUMED [9]. Также PLUMED моделирует поверхность свободной энергии для системы лиганд-белок (См. Рис. 8).

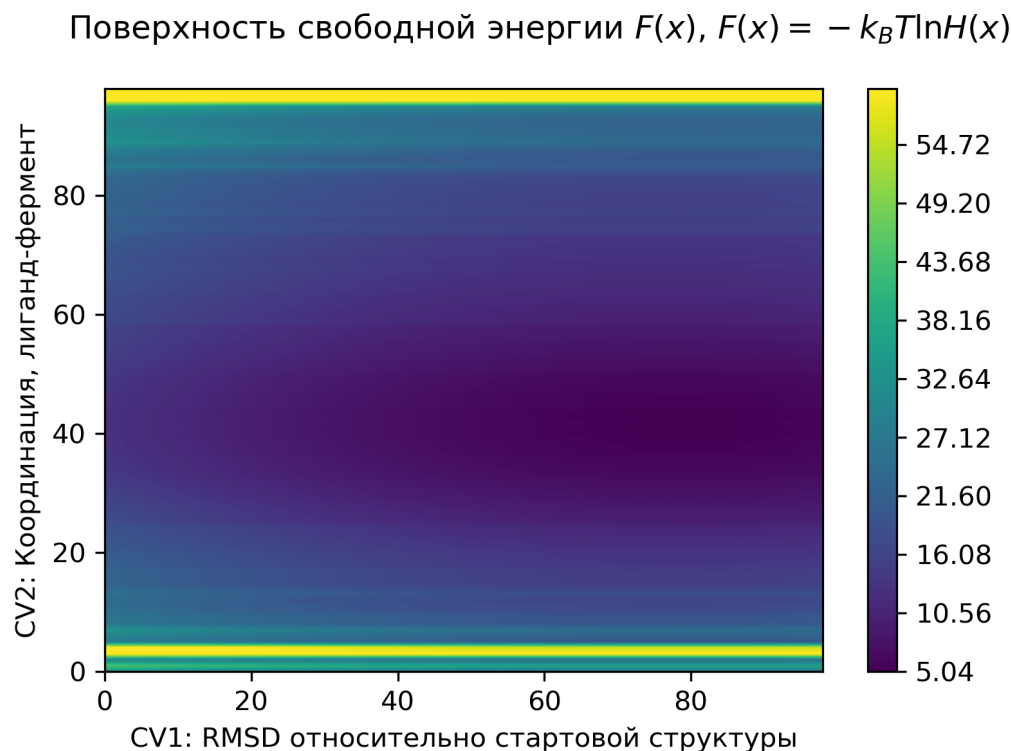


Рис. 8: Смоделированная функция свободной энергии на основе гистограммы в координатах CV1-CV2

Через некоторые отрезки времени траектории проводится построение функции свободной энергии конформаций из всей траектории. Из всех конформаций выбирается набор из максимум 10шт, обладающих наименьшей (оценкой) свободной энергией. Если этот набор останется неизменным на протяжении 100 нс траектории, то расчет останавливается.

В конвейере SSDO-0.1 для каждой конформации траектории создается файл структуры комплекса *ES*. Полученные 10 структур наименьшей свободной энергии подаются на вход программе affbio [1], использующий алгоритм Affinity Propagation для разделения на кластеры.

В полученных кластерах находятся лидеры (структуры с наименьшей $G(x)$). Данные структуры дают начало дочерним узлам.

4. Ступень №3 *RosettaDesign*

- Модель комплекса подается на вход программе *RosettaDesign* с параметрами по умолчанию (описание подготовки системы приведена на рис. 9).
- Файл геометрических ограничений (*.cst*) накладывает штраф и уменьшает вероятность попадания соответствующей структуры в набор 10-ти лучших по значению скоровой функции (параметр $total_{score}$).
- При запуске специальная функция следит за набором 10 лучших.

- Если он не изменяется после добавления очередных 300 структур (итого получается, что в лучшем случае будет тестировано 600 структур), то запуск признается удачным.
- Если процесс продолжается до 10 000-го результата, то узел удаляется.
- В случае удачного запуска 10 лучших структур превращаются в 10 дочерних узлов.

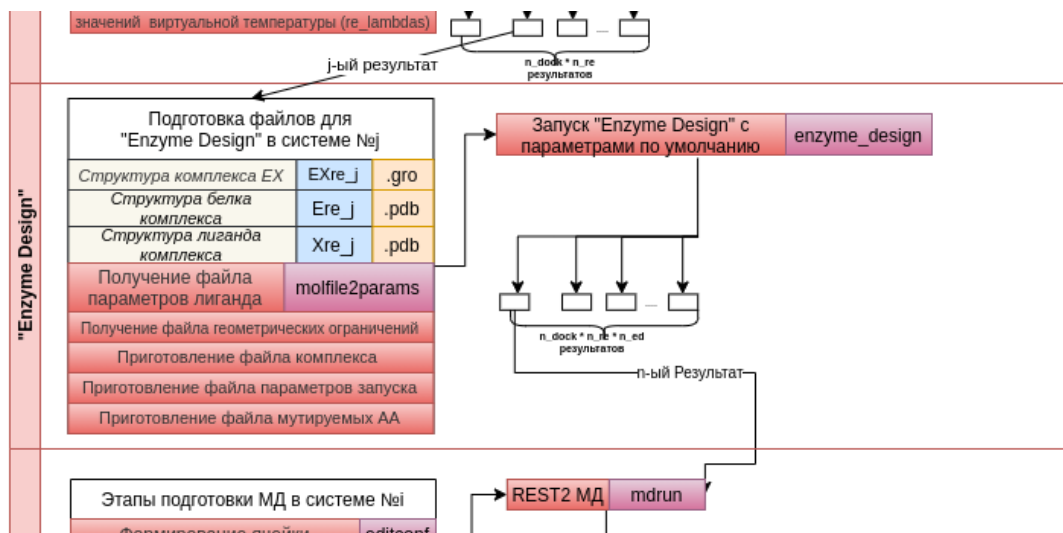


Рис. 9: Схема конвейера SSDO-0.1: *RosettaDesign*

5. Ступень №4 Молекулярная динамика REST2 На четвертой ступени для результатов третьей ступени проводится аналогичный запуск молекулярной динамики REST2 также как во второй ступени, но без потенциалов удержания (См. Рис. 10).

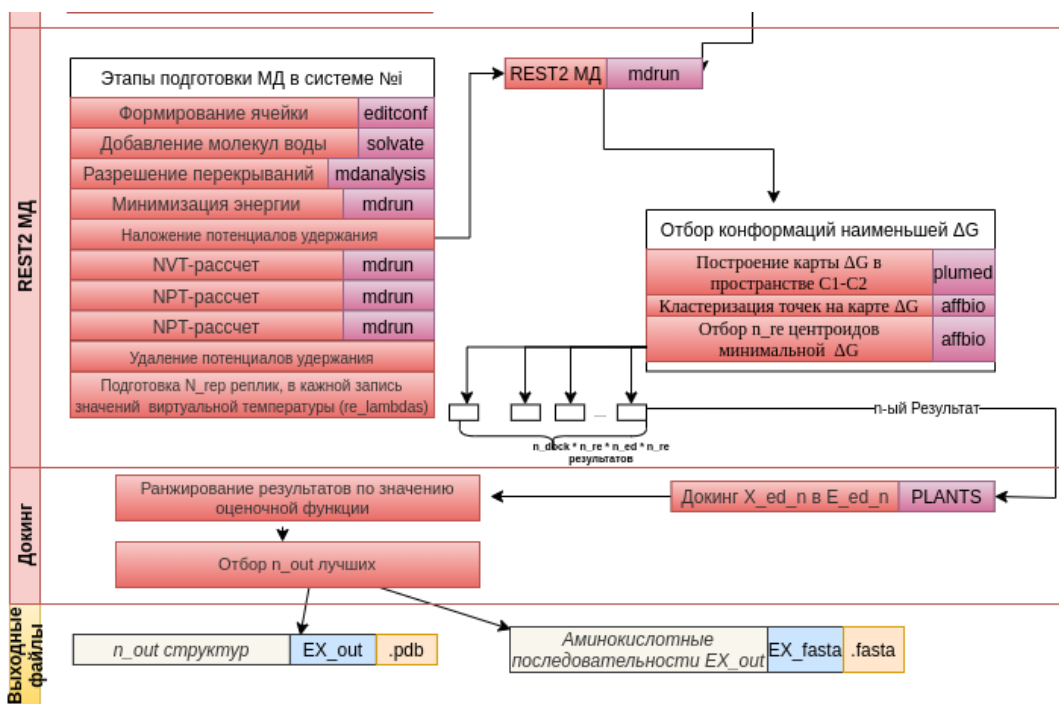


Рис. 10: Схема конвейера SSDO-0.1: Молекулярная динамика REST2 четвертой ступени.

6. Ступень №5 Докинг На пятой ступени для результатов прошлой проводится докинг аналогично первой ступени, но в этот раз геометрические ограничения отключены и выводится одна структура (См. Рис. 10). Эта одна структура и есть результат узла.

7. Оценка результатов

- Все результаты докинга пятой ступени сортируются по значению сора.
- Топ 10 лучших выводится из конвейера в качестве итоговых результатов.

6 Результаты и обсуждения

В рамках решения тестирования конвейера было решено изменить и оптимизировать субстратную специфичность фермента LMW-PTP для связывания лиганда $PI[3,5]P_2$ с помощью конвейера SSDO-0.1. Для проверки работоспособности конвейера были проведены множества тестовых запусков с параметрами, обеспечивающими высокую скорость счета. В запуске количество «потомков» каждого узла (кроме конечных) было указано как 1. Но эти результаты не представлены, потому что не имеют большого физического смысла. Далее был запущен производящий эксперимент, но были пройдены стадии №1 и №2.


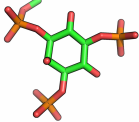
1. На вход конвейеру были поданы следующие данные (См. Рис. 6).

- Файл фермента: **E.pdb** Структура белка была взята из банка PDB с PDB_{ID}: 1xww.
- Файл лиганда: **X.pdb** Структура лиганда была взята из структуры фермента MTMR2 (PDB_{ID}: 1zvr).
- **E_X_пол_кат_пары** Атомы пары представлены списком из номера остатка и имени атома в исходных файлах:

```
E_X_пол_кат_пары = [  
    [[17, 'SG' ], [3632, 'P1']],  
    [[18, 'NH1'], [3632, 'O11']],  
    [[18, 'NH2'], [3632, 'O13']],  
    [[129, 'OD1'], [3632, 'O12']],  
]
```

- **E_мут_искл_ост:** Отсутствуют
- Заряд системы: -5

Таблица 6: Тестовый запуск SSDO-0.1: Входные данные

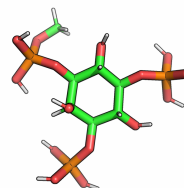
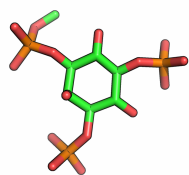
Имя	Значение
E _{pdb}	LMW-PTP (PDB ID: 1xww)
	
X _{pdb}	PI[3,5]P ₂
	

2. Ступень №0 Подготовка реагентов

(а) Протонирование лиганда

В ходе работы конвейера структура лиганда и белка были протонированы а также были рассчитаны заряды их атомов (См. Табл. 7). Структура лиганда была протонирована программой Rmол [22]. Лишние атомы водорода были удалены вручную.

Таблица 7: Тестовый запуск SSDO-0.1: Входные данные
Деротонированная Протонированная



- (b) Расчет зарядов атомов лиганда: MOPAC В SSDO-1.0 для расчета зарядов по умолчанию используется программа RED-III [4], которая в свою очередь использует программу GAMESS [13].

Программа MOPAC [23] используется для получения протонированной структуры лиганда в формате mol2. Данный файл содержит информацию о зарядах атомов, которые рассчитываются программой семиэмпирическими методами квантовой химии.

- (c) Протонирование белка: SPORES Программа SPORES [24] была использована получения протонированной структуры белка в формате mol2.

3. Ступень №1: докинг PLANTS В данной работе был проведен запуск с количеством результатов в количестве 1 штуки для визуальной оценки (См. Рис. 11). Такое упрощение было сделано, т.к. на этом этапе разработки было важно показать работоспособность кода конвейера.

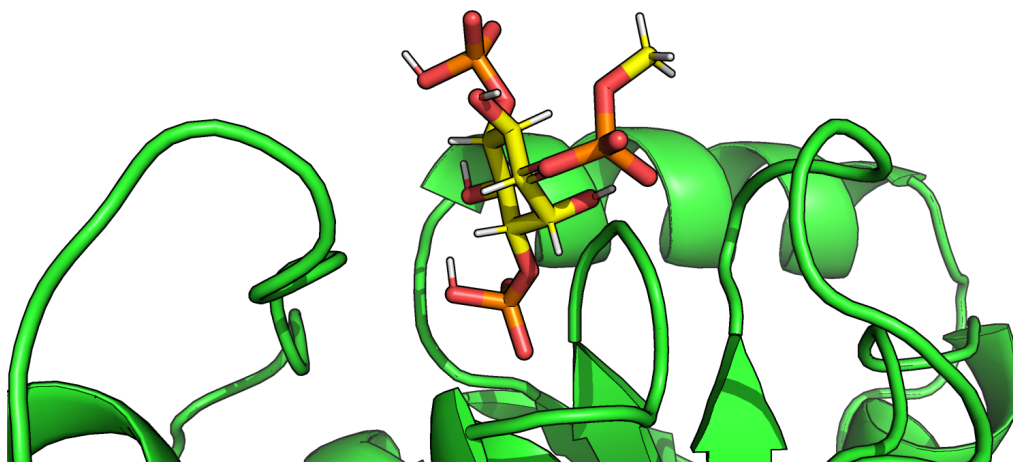


Рис. 11: Результат докинга протонированной структуры лиганда в протонированную структуру белка инструментом PLANTS 11.

4. Ступень №2: Молекулярная динамика REST2 Далее структура была внесена в систему для молекулярной динамики типа REST2 с четырьмя репликами длиной 103нс каждая. В реплике с параметром виртуальной температуры 300K была смоделирована функция плотности вероятности, показанная на рис. 8. Далее были проведены шаги согласно описанному алгоритму.

7 Заключение

Таким образом был предложен и частично протестирован конвейер изменения и оптимизации субстратной специфичности для LMW-PTP.

Список литературы

- [1] affbio · pypi. <https://pypi.org/project/affbio/>. (Accessed on 12/09/2020).
- [2] Distance restraints - gromacs. <http://www.gromacs.org/h34/35>. (Accessed on 12/09/2020).
- [3] Protein-complex md setup tutorial - bioexcel building blocks. https://mmmb.irbbarcelona.org/biobb/availability/tutorials/protein-complex_md_setup. (Accessed on 12/10/2020).
- [4] R.e.d. iii faq. <https://upjv.q4md-forcefieldtools.org/RED/FAQ-III.php>. (Accessed on 06/01/2020).
- [5] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.
- [6] Paul D Adams, David Baker, Axel T Brunger, Rhiju Das, Frank DiMaio, Randy J Read, David C Richardson, Jane S Richardson, and Thomas C Terwilliger. Advances, interactions, and future developments in the cns, phenix, and rosetta structural biology software systems. *Annual review of biophysics*, 42:265–287, 2013.
- [7] Pau Andrio, Adam Hospital, Javier Conejero, Luis Jordá, Marc Del Pino, Laia Codo, Stian Soiland-Reyes, Carole Goble, Daniele Lezzi, Rosa M Badia, et al. Bioexcel building blocks, a software library for interoperable biomolecular simulation workflows. *Scientific data*, 6(1):1–8, 2019.
- [8] Kristian B Axelsen and Michael G Palmgren. Evolution of substrate specificities in the p-type atpase superfamily. *Journal of molecular evolution*, 46(1):84–101, 1998.
- [9] Massimiliano Bonomi, Davide Branduardi, Giovanni Bussi, Carlo Camilloni, Davide Provati, Paolo Raiteri, Davide Donadio, Fabrizio Marinelli, Fabio Pietrucci, Ricardo A Broglia, et al. Plumed: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications*, 180(10):1961–1972, 2009.
- [10] Alan W Sousa da Silva and Wim F Vranken. Acypype-antechamber python parser interface. *BMC research notes*, 5(1):367, 2012.
- [11] Jiri Damborsky and Jan Brezovsky. Computational tools for designing and engineering enzymes. *Current opinion in chemical biology*, 19:8–16, 2014.
- [12] Eyong Egbe, Colin W Levy, and Lydia Taberner. Computational and structure-guided design of phosphoinositide substrate specificity into the tyrosine specific lmw-ptp enzyme. *PloS one*, 15(6):e0235133, 2020.
- [13] Martyn F Guest*, Ian J Bush, Huub JJ Van Dam, Paul Sherwood, Jens MH Thomas, Joop H Van Lenthe, Remco WA Havenith, and John Kendrick. The gamess-uk electronic structure package: algorithms, developments and applications. *Molecular physics*, 103(6-8):719–747, 2005.
- [14] James M Haile. Molecular dynamics simulation. *Elementary methods*, 1992.
- [15] Lin Jiang, Eric A Althoff, Fernando R Clemente, Lindsey Doyle, Daniela Röthlisberger, Alexandre Zanghellini, Jasmine L Gallaher, Jamie L Betker, Fujie Tanaka, Carlos F Barbas, et al. De novo computational design of retro-aldol enzymes. *science*, 319(5868):1387–1391, 2008.
- [16] Ana Lilia Juárez-Vázquez, Janaka N Edirisinghe, Ernesto A Verduzco-Castro, Karolina Michalska, Chenggang Wu, Lianet Noda-García, Gyorgy Babnigg, Michael Endres, Sofia Medina-Ruiz, Julián Santoyo-Flores, et al. Evolution of substrate specificity in a retained enzyme driven by gene loss. *Elife*, 6:e22679, 2017.

- [17] Motoshi Kamiya and Yuji Sugita. Flexible selection of the solute region in replica exchange with solute tempering: Application to protein-folding simulations. *The Journal of chemical physics*, 149(7):072304, 2018.
- [18] Oliver Korb, Thomas Stützle, and Thomas E Exner. Plants: Application of ant colony optimization to structure-based drug design. In *International workshop on ant colony optimization and swarm intelligence*, pages 247–258. Springer, 2006.
- [19] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In *Methods in enzymology*, volume 487, pages 545–574. Elsevier, 2011.
- [20] Naveen Michaud-Agrawal, Elizabeth J Denning, Thomas B Woolf, and Oliver Beckstein. Mdanalysis: a toolkit for the analysis of molecular dynamics simulations. *Journal of computational chemistry*, 32(10):2319–2327, 2011.
- [21] Florian Richter, Andrew Leaver-Fay, Sagar D Khare, Sinisa Bjelic, and David Baker. De novo enzyme design using rosetta3. *PloS one*, 6(5):e19230, 2011.
- [22] Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 1.3r1. The PyMOL Molecular Graphics System, Version 1.3, Schrödinger, LLC., Aug 2010.
- [23] James JP Stewart. Mopac: a semiempirical molecular orbital program. *Journal of computer-aided molecular design*, 4(1):1–103, 1990.
- [24] T ten Brink and TE Exner. Structure protonation and recognition system (spores): version 1.28. *Universität Konstanz, Fachbereich Chemie und Zukunftskolleg D-78457 Konstanz*, 2012.
- [25] G. van Rossum and J. de Boer. Interactively Testing Remote Servers Using the Python Programming Language. *CWI Quarterly*, 4(4):283–303, December 1991.