

Оглавление

1	Введение	2
2	Описание инструмента	5
2.1	Перед использованием	5
2.2	Входные и выходные данные	6
2.3	Препроцессинг	9
2.4	Постпроцессинг. Кластеризация	9
3	Заключение	11
	Список литературы	12

1 Введение

Знание пространственной структуры белковых комплексов очень важно для понимания механизмов их функционирования.

Основным методом определения структуры белка был и является рентгеноструктурный анализ (РСА)[1][2]. По картине дифракции рентгеновских лучей на кристаллах можно изучать их свойства, определять координаты атомов, однако, получение кристаллов довольно сложная задача, успех которой очень сильно зависит от условий, и для получения кристалла могут уйти годы. Еще одним минусом является то, что кристалл не естественное состояние белка. В кристаллической структуре форма белка изменена, так как во время образования кристалла кристаллические силы искажают структуру белка.

В 1975 году впервые удалось получить структуру мембранного белка при помощи электронной микроскопии[3]. Электронный микроскоп использует пучок быстрых электронов вместо света, благодаря чему разрешающая способность повышается до 0.05 нм (для обычных микроскопов около 200 нм). Для фиксации молекулы объект замораживают[4], причем делая это довольно быстро, можно получить макромолекулу в естественном окружении.

Метод спектроскопии ядерного магнитного резонанса[5] используется с 1980х. Этот микроскопический метод использует явление ядерного магнитного резонанса. К его преимуществам относят возможность изучать молекулы в природном окружении — растворе, — а также возможность исследовать не только структуру, но и подвижность биомолекул на различных участках. Однако проведение таких экспериментов стоит больших денег. Также есть ограничения по размеру объектов исследования.

В 1971 году в Брукгейвенской национальной лаборатории (США) был создан архив данных структур биологических макромолекул The Protein Data Bank (PDB)[6]. Вначале архив содержал всего несколько структур, но каждый год база значительно пополнялась. В 1980х количество добавляемых структур начало стремительно расти. Это связано с улучшением технологий кристаллографии, появлением новых методов получения структур (спектроскопия ядерного магнитного резонанса) и изменениями в представлении общества об обмене данными. Сейчас база активно используется и пополняется самыми разными группами исследователей в области биологии, химии и компьютерных наук, преподавателями и учащимися всех уровней со всего мира.

Большая часть структур получена при помощи рентгеноструктурного анализа, около 15 % — при помощи спектроскопии ядерного магнитного резонанса, и лишь малая часть (около 1 %) — при помощи крио-электронной микроскопии.

На практике белки часто выполняют свои функции не в одиночку. Для участия в различных внутриклеточных и внешних процессах они с помощью белок-белковых взаимодействий собираются в целый комплекс. Именно в белок-белковом

взаимодействии выполняются такие важные процессы как передача сигналов, транспортная и защитная функции, поэтому перед нами встает задача определения структуры целого белкового комплекса.

Белковые комплексы часто недостаточно устойчивы для образования кристалла, поэтому определить структуру комплекса при помощи рентгеноструктурного анализа еще сложнее.

В настоящее время для предсказания белок-белковых взаимодействий широко используются вычислительные методы. Одним из самых быстрых среди них является молекулярный докинг, который, исходя из трехмерной структуры компонентов (отдельных белков), позволяет определить трехмерную структуру комплекса[7][8][9].

Большинство современных методов докинга полагают взаимодействующие белки твердыми телами. Они генерируют огромное количество конформаций с хорошей комплементарностью поверхности. Корреляционный подход с быстрым преобразованием Фурье (БПФ) [10], введенный в 1992 году Качальски-Кациром (Katchalski-Katzir) и его коллегами, произвел революцию. Благодаря численной эффективности этого алгоритма впервые стало возможным систематически исследовать конформационное пространство белок-белковых комплексов, оценивая энергии для миллиардов конформаций в сетке, и, таким образом, находить структуру комплекса без какой-либо априорной информации по ожидаемой структуре. Другие подходы, в первую очередь Монте-Карло, работают хорошо, если поиск ограничен областями конформационного пространства, но становятся вычислительно затратными, если такие ограничения отсутствуют. По этой причине докинг на основе быстрого преобразования Фурье является первым шагом во многих методах, которые хорошо зарекомендовали себя в CAPRI (Critical Assessment of Predicted Interactions). Хотя основанный на быстром преобразовании Фурье метод представляет собой значительный прогресс в белковом докинге, к его минусам можно отнести довольно неточную оценку свободной энергии связи.

Методом докинга, который решает эту проблему, является Piper[11]. Он использует парные потенциалы, которые оказались мощным инструментом нахождения почти нативных конформаций в наборах структур, генерируемых алгоритмами поиска при макромолекулярном моделировании. Это внесло существенный вклад в повышение точности предсказания структуры белка.

В методах докинга, основанных на быстром преобразовании Фурье, ускорение достигается только в трехмерном подпространстве полного шестимерного вращательно-поступательного пространства, а оставшиеся компоненты должны быть подобраны с использованием обычных медленных вычислений. С учетом этого замечания алгоритм Piper был улучшен, и создан FMFT (Fast Manifold Fourier Transform) алгоритм[12]. Теперь выборка происходит в пятимерном вращающемся пространстве, скорость расчета повышается более, чем на порядок при той же точности. В результате

появляется возможность решать новые классы задач докинга, включая докинг больших комплексов белков, а не только одной пары белков.

Код, реализующий алгоритм FMFT, доступен для скачивания на https://bitbucket.org/abc-group/fmft_suite

При желании можно протестировать данный алгоритм. Для удобства использования был реализован плагин ppfmft (<https://github.com/aziza-calm/ppfmft>), о котором и пойдет речь.

2 Описание инструмента

2.1 Перед использованием

Для успешного запуска плагина необходимо предварительно установить пакет PyMOL[13] и FMFT suite(Fast Manifold Fourier Transform suite)[12].

PyMOL один из немногих инструментов визуализации молекул с открытым исходным кодом. Написав собственные скрипты на языке Python, легко можно расширить функционал его, чем обусловлен наш выбор.

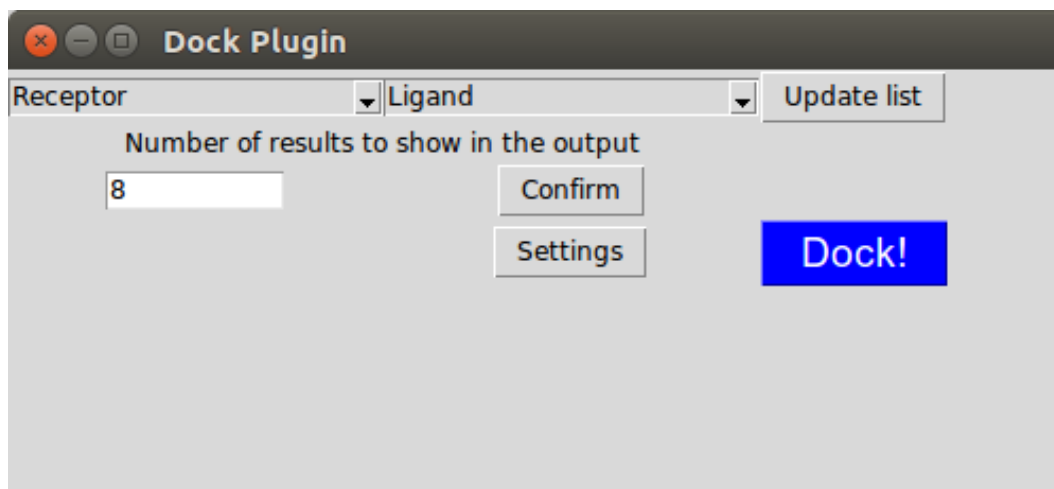
Скачивание и установка PyMOL задача весьма тривиальная и не займет много времени. Пакет доступен для всех популярных операционных систем.

Изначально написание графического интерфейса пользователя планировалось на PyQt, его главное преимущество - кроссплатформенность. Однако, данный фреймворк поддерживается версиями PyMOL начиная с 2.0, что значительно сужает круг пользователей, так как последняя версия вышла сравнительно недавно. Также, новый PyMOL поддерживает плагины, написанные для старых версий (1.7), поэтому выбор был сделан в сторону Tkinter.

Tkinter входит в стандартную библиотеку Python. Следует отметить, что плагин был написан на Python 2.7, так как PyMOL 1.7 не поддерживает версии Python 3.0 и выше. Надо быть готовым, что установка FMFT suite может потребовать значительных усилий и времени.

2.2 Входные и выходные данные

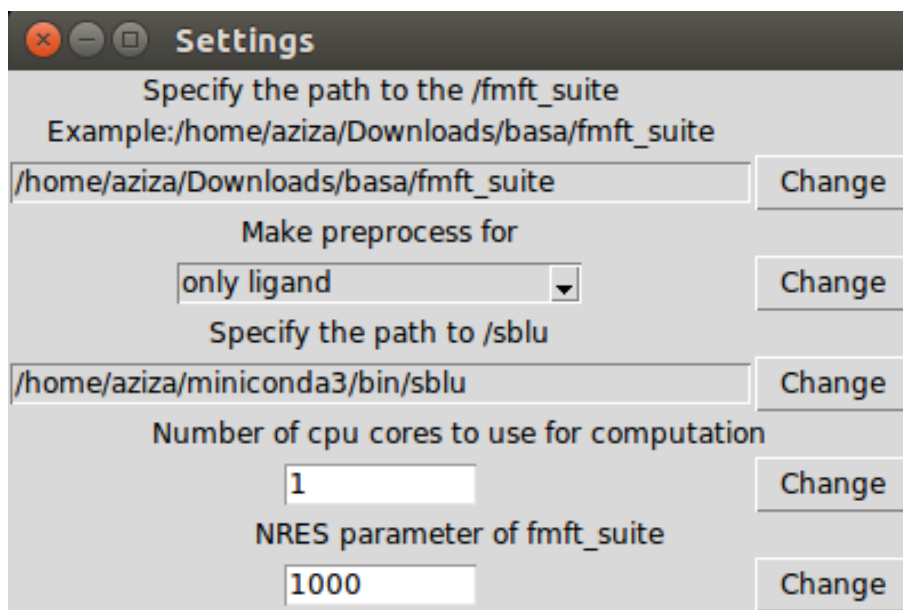
После установки всего необходимого и открытия плагина вы увидите следующее окно.



В выпадающих списках выбираются рецептор и лиганд соответственно. Список формируется из уже загруженных в PyMOL молекул. При необходимости его можно обновить нажатием кнопки Update list. Файлы рецептора и лиганда должны быть в формате pdb, иначе поведение не определено.

Далее указывается желаемое количество выводимых результатов.

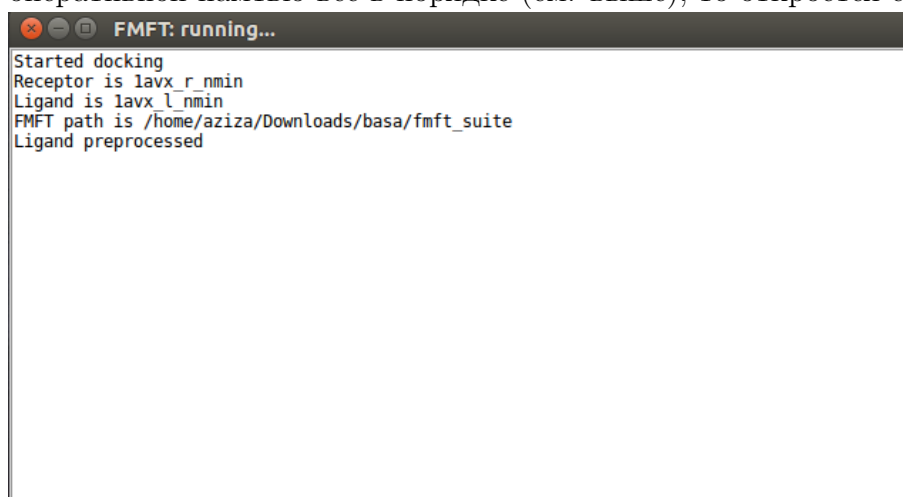
В настройках обязательно нужно указать путь до FMFT suite. Здесь же регулируются его параметры, а также параметры препроцессинга (подробнее об этом см.2.3).



Один из параметров FMFT suite - количество ядер, используемых для расчета. Конечно, чем больше ядер вы используете, тем быстрее выполнится расчет, но здесь

следует учесть, что каждый процесс занимает примерно 1.5 гигабайта оперативной памяти, поэтому нужно следить, хватает ли ее на устройстве пользователя. В плагине реализована подобная проверка. Прежде чем запустить расчет, проверяется остаток свободной оперативной памяти, сравнивается с предполагаемым количеством памяти, которое займет расчет (количество ядер, указанное в параметрах, умноженное на 1.5 гигабайта), и если свободной памяти оказывается недостаточно, то выводится сообщение с соответствующим предупреждением и просьбой уменьшить количество ядер в параметрах. Далее пользователь сам для себя решает, действительно ли ему нужно снижать количество ядер. Эта возможность дается, потому что оценка довольно грубая.

Когда рецептор и лиганд выбраны, все желаемые параметры настроены, все готово для запуска докинга. Процесс запускается при нажатии синей кнопки Dock. Если с оперативной памятью все в порядке (см. выше), то откроется следующее окно.



```
FMFT: running...
Started docking
Receptor is lavx_r_nmin
Ligand is lavx_l_nmin
FMFT path is /home/aziza/Downloads/basa/fmft_suite
Ligand preprocessed
```

В этом окне будет появляться лог процесса, а название окна показывает статус процесса. Если он идет, то в названии мы видим "FMFT: running...", если процесс завершен, то - "FMFT: finished". В случае ошибки - "FMFT: failed".

Успешно заверченный расчет выдаст несколько файлов - выходные данные. Для отображения результатов используется два из них (не считая файлы постпроцессинга см. 2.4) - ft.000.0.0 и gm.000.0.0.

ft.000.0.0 выглядит примерно следующим образом.

0	7.57	-2.51	-10.30	-1036.788452148
1	7.26	-1.27	-7.55	-1031.609252930
2	-0.25	2.19	1.12	-1025.497192383
3	-0.31	1.51	1.84	-1024.613403320
4	9.78	-0.51	-7.55	-1022.117126465
5	2.00	-0.10	-1.24	-1005.038635254
6	12.06	0.83	0.05	-998.006408691
7	-0.49	-0.79	-0.59	-997.780761719
8	7.47	-2.13	-7.08	-997.116638184
9	7.57	-2.51	-10.30	-993.983886719
10	-0.31	1.51	1.84	-992.668579102
11	9.67	-0.18	0.05	-990.718627930
12	9.78	-0.51	-7.55	-987.676025391
13	2.00	-0.10	-1.24	-982.701625077

Первый столбец просто номер, второй, третий и четвертый - координаты вектора трансляции, пятый - соответствующая энергия.

rm.000.0.0 выглядит примерно следующим образом.

```

0.191252055 0.439085796 0.877853242 0.209577670 0.855409649 -0.473549236 -0.958905549 0.274545702 0.071587751
0.216077213 0.433852252 0.875080502 0.107962891 0.880166263 -0.462224365 -0.970388918 0.194353021 0.143430300
-0.267110511 -0.713580044 -0.647645025 0.848485745 0.144465229 -0.509118589 0.456863125 -0.685508498 0.566881102
-0.267110511 -0.713580044 -0.647645025 0.848485745 0.144465229 -0.509118589 0.456863125 -0.685508498 0.566881102
0.107962891 0.273561356 0.955775884 0.051818684 0.958540231 -0.280260684 -0.992803193 0.079779761 0.089314103
0.994347251 -0.106137932 0.002878185 0.105825942 0.988493516 -0.108080702 0.008626395 0.107774336 0.994137957
0.566017707 0.790907419 -0.232571299 0.112205438 0.205577251 0.972187191 0.816721430 -0.576370929 0.027616250
-0.295671281 -0.699622057 -0.650467118 0.753740421 0.247477951 -0.608793924 0.586981927 -0.670286238 0.454161301
0.216077213 0.433852252 0.875080502 0.107962891 0.880166263 -0.462224365 -0.970388918 0.194353021 0.143430300
0.236840534 0.416269440 0.877853242 0.299321221 0.828346502 -0.473549236 -0.924290738 0.374915759 0.071587751
-0.364289885 -0.670570758 -0.646241238 0.826644701 0.086747478 -0.555997674 0.428895579 -0.736756224 0.522722535
0.574737811 0.808278293 -0.127916571 0.051405777 0.120344308 0.991400370 0.816721430 -0.576370929 0.027616250
0.203377340 0.337042984 0.919260401 0.130318860 0.921210579 -0.366589777 -0.970388918 0.194353021 0.143430300
0.980726972 -0.182364596 -0.070126741 0.173868604 0.978316246 -0.112547902 0.089130883 0.098185925 0.991168608
0.980493486 -0.182493823 -0.072996773 0.182636419 0.983168934 -0.004773349 0.072639266 -0.008651631 0.997328754
0.105806039 0.495874114 0.861924559 0.180960207 0.842714467 -0.507636223 -0.977782433 0.096215441 -0.000569203
0.203377340 0.337042984 0.919260401 0.130318860 0.921210579 -0.366589777 -0.970388918 0.194353021 0.143430300
0.311683819 0.344712548 0.885452684 0.331467074 0.833888212 -0.441316245 -0.890495803 0.431049544 0.145648603
0.191673350 0.216089042 0.957374980 0.090769816 0.967377573 -0.236519496 -0.977252356 0.132352355 0.165806137
0.280611269 0.271124040 0.920732898 -0.172984718 0.929281587 -0.328361793 -0.944104657 0.250853517 0.213866570
0.107962891 0.273561356 0.955775884 0.051818684 0.958540231 -0.280260684 -0.992803193 0.079779761 0.089314103
0.955531407 -0.256708624 -0.145015874 0.257337390 0.966202888 -0.015147501 0.144004159 -0.022844093 0.989313373
0.409479145 -0.394509616 -0.822582305 -0.901424199 -0.313909839 -0.298152688 -0.140574688 0.863582904 -0.484214132
0.280611269 0.271124040 0.920732898 -0.172984718 0.929281587 -0.328361793 -0.944104657 0.250853517 0.213866570
0.967236457 -0.238455287 0.087136175 0.246646903 0.963936319 -0.09960378 0.060157643 0.118177189 0.991168608
0.949397608 -0.314076712 0.000000000 0.314076712 0.949397608 0.000000000 0.000000000 0.000000000 1.000000000
0.194673933 -0.73257958 -0.652273637 0.766774737 0.301045311 -0.566946402 0.011085919 -0.610516633 0.503181261
0.131105105 0.491470505 0.86090934 0.077405618 0.860739400 -0.503126281 -0.988341956 0.132606287 0.074804751
0.973825919 -0.161671207 0.159767019 0.178589686 0.979047052 -0.097839634 -0.140601577 0.123811513 0.982294205
0.297408099 0.368364647 0.880826630 0.142177575 0.895198302 -0.422380797 -0.944104657 0.250853517 0.213866570
0.487608081 0.864187010 0.120344308 0.74339487 0.311155720 0.807658684 0.875701173 0.305807667 0.785376678

```

Каждая строчка содержит 9 элементов матрицы поворота 3x3.

Предполагается, что рецептор неподвижен, меняется расположение лиганда относительно рецептора. Имея вектор трансляции и матрицу поворота, мы двигаем и поворачиваем лиганд. В PyMOL для этого есть две команды - translate и rotate. Чтобы воспользоваться командой rotate, пришлось из матрицы поворота получить ось и угол поворота, так как эта команда принимает именно ось и угол.

Для каждого положения лиганда создавалась временная копия, которая записывалась в результирующую молекулу result как одно из состояний. Далее с помощью команды mplay() смотрим на result в разных состояниях. Таким образом получаем анимацию, в которой лиганд перемещается вокруг рецептора.

2.3 Препроцессинг

Как уже было сказано, рецептор и лиганд должны быть загружены в формате pdb. PDB-файл представляет собой текстовый файл в ASCII кодировке.

Пример pdb-файла

```
HEADER      VIRUS                               13-MAR-16   SIRE
TITLE       THE CRYO-EM STRUCTURE OF ZIKA VIRUS
COMPND      MOL_ID: 1;
COMPND      2 MOLECULE: E PROTEIN;
COMPND      3 CHAIN: A, C, E;
COMPND      4 SYNONYM: CAPSID E PROTEIN;
COMPND      5 MOL_ID: 2;
COMPND      6 MOLECULE: M PROTEIN;
COMPND      7 CHAIN: B, D, F;
COMPND      8 FRAGMENT: UNP RESIDUES 179-253
SOURCE      MOL_ID: 1;
SOURCE      2 ORGANISM_SCIENTIFIC: ZIKA VIRUS;
SOURCE      3 ORGANISM_TAXID: 64320;
SOURCE      4 MOL_ID: 2;
SOURCE      5 ORGANISM_SCIENTIFIC: ZIKA VIRUS;
SOURCE      6 ORGANISM_TAXID: 64320
KEYWDS      ZIKA VIRUS, VIRUS
EXPDTA      ELECTRON MICROSCOPY
AUTHOR      D. SIROHI, Z. CHEN, L. SUN, T. KLOSE, T. PIERSON, M. ROSSMANN, R. KUHN
REVDAT      6 18-JUL-18 SIRE 1 REMARK
REVDAT      5 13-SEP-17 SIRE 1 REMARK
REVDAT      4 04-MAY-16 SIRE 1 JRNL LINK
REVDAT      3 13-APR-16 SIRE 1 REVDAT JRNL
REVDAT      2 06-APR-16 SIRE 1 JRNL
REVDAT      1 30-MAR-16 SIRE 0
ATOM        10000  C  STRUT 7 CHEN L SUN T KLOSE T PIERSON M ROSSMANN
```

Хотя существует некая общепринятая структура pdb-файлов и стандартизованная номенклатура для типов атомов, все равно довольно часто эти файлы могут быть записаны не по стандартам по усмотрению авторов.

Для корректной работы FMFT suite файлы рецептора и лиганда следует проверить на "испорченность". Эта проверка (и исправление в случае необходимости) называется препроцессингом.

В плагине для препроцессинга используется пакет sb-lab-utils. Он написан как модуль Python, и планировалось его импортирование и использование как модуля. Однако, PyMOL использует собственный набор питоновских модулей, поэтому препроцессинг пришлось запускать с помощью subprocess.Popen (так же как и сам FMFT suite).

Препроцессинг является опциональным параметром. Можно препроцессить только лиганд или только рецептор, или и лиганд, и рецептор, или вообще не делать препроцессинг.

Если препроцессинг используется, то необходимо установить пакет sb-lab-utils и в настройках (settings) указать путь до файла sblu.

2.4 Постпроцессинг. Кластеризация

Основными выходными файлами FMFT suite являются ft.000.0.0 и rm.000.0.0. Они содержат в себе несколько тысяч положений лиганда. Многие из этих положений очень близки, поэтому нет смысла отображать абсолютно все. Гораздо удобнее сначала сгруппировать данные, а потом выводить центры каждой из групп, то есть провести кластеризацию.

В плагине для кластеризации используется пакет sb-lab-utils (тот же, что и для

препроцессинга). Сначала считаются попарные (PairWise) расстояния (RMSD) между всеми результирующими структурами, результаты записываются в файл `pwrmsd.000.0.0`. На его основе делается уже кластеризация. После кластеризации получаем файл `clusters.000.0.0.json`, в котором перечислены центры и члены кластера. Centers - строки из `ft`-файла, которые соответствуют центрам кластеров, members - список членов кластера. Именно centers мы выводим как финальный результат.

3 Заключение

Итак, в ходе работы был создан плагин, который позволит широкому пользователю взаимодействовать с инструментом FMFT suite. Теперь не нужно вводить длинные команды в консоли и разбираться с документацией. Все необходимые манипуляции можно произвести с помощью плагина с интуитивно понятным интерфейсом.

Результат загружается прямо в RuMOL в виде анимации, что очень наглядно и позволит детально рассмотреть каждое положение лиганда. Пользователь не запутается в выходных файлах FMFT suite, все уже сделал плагин.

Подобный подход, как уже было упомянуто, расширяет круг потенциальных потребителей, и способствует популяризации науки в целом.

В дальнейшем плагин будет дорабатываться, планируется добавление опции запуска докинга на удаленном сервере и др.

Список литературы

- [1] M.S. Smyth and J.H.J. Martin *x Ray crystallography* Mol Pathol. 53(1): 8–14 (2000)
- [2] M.W. Parker *Protein Structure from X-Ray Diffraction* J Biol Phys. 29(4): 341–362 (2003)
- [3] Lyman Monroe, Genki Terashi and Daisuke Kihara *Variability of Protein Structure Models from Electron Microscopy* Structure 25(4): 592–602.e2 (2017)
- [4] Marta Carroni and Helen R. Saibil *Cryo electron microscopy to determine the structure of macromolecular complexes* Methods 95: 78–85 (2016)
- [5] Andrea Cavalli, Xavier Salvatella, Christopher M. Dobson, and Michele Vendruscolo *Protein structure determination from NMR chemical shifts* Proc Natl Acad Sci U S A. 104(23): 9615–9620 (2007)
- [6] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne *The Protein Data Bank* Nucleic Acids Res. 28(1): 235–242 (2000)
- [7] Smith, G.R., Sternberg, M.J. *Prediction of protein-protein interactions by docking methods.* Curr. Opin. Struct. Biol. **12**, 28–35 (2002).
- [8] Halperin, I., Ma, B., Wolfson, H. Nussinov, R. *Principles of docking: an overview of search algorithms and a guide to scoring functions.* Proteins **47**, 409–443 (2002).
- [9] Ritchie, D.W. *Recent progress and future directions in protein-protein docking.* Curr. Protein Pept. Sci. **9**, 1–15 (2008).
- [10] Katchalski[U+2010]Katzir E, Shariv I, Eisenstein M, Friesem A, Aflalo C, Vakser I. *Molecular surface recognition—determination of geometric fit between proteins and their ligands by correlation techniques* Proc Natl Acad Sci USA 89: 2195–2199 (1992)
- [11] Kozakov D, Brenke R, Comeau SR, Vajda S *Piper: An FFT-based Protein Docking Program with Pairwise Potentials.* Proteins **65(2)**, 392–406 (2006).
- [12] Padhorny, D., Kazennov, A.M. Brandon S. Zerbe, Kathryn A. Porter, Bing Xia, Scott E. Mottarella, Yaroslav Kholodov, David W. Ritchie, Sandor Vajda, and Dima Kozakov *Protein-protein docking by fast generalized Fourier transforms on 5D rotational manifolds* PNAS 113 (30) E4286-E4293 (2016).
- [13] Shuguang Yuan, H.C. Stephen Chan, Zhenquan Hu *Using PyMOL as a platform for computational drug design* Wiley Interdiscip Rev: Comput Mol Sci. 7:e1298 (2017)