

Колебательная и конформационная подвижность полипептидной цепи

Задача направлена на знакомство с азами методов молекулярного моделирования. Основной ее целью является знакомство студентов с методом молекулярной динамики на примере пакета программ GROMACS и освоение ими базовых операций, необходимых для успешной постановки молекулярно-динамических расчетов, а также обработки и визуализации полученных данных.

Объектом исследования в рамках задачи был выбран случайный полипептид, состоящий из 7-ми аминокислотных остатков, индивидуально сгенерированный каждым из студентов. На примере модельного полипептида будут освоены следующие операции: 1. Создание молекулярной модели; 2. Оптимизация геометрии системы; 3. Расчет молекулярной динамики; 4. Анализ полученных траекторий и построение графиков.

Создание рабочей модели.

На первом этапе работы должна быть создана модель молекулы небольшого полипептида в вакууме. Аминокислотный состав пептида не имеет решающего значения, так как эта молекула будет использована в качестве примера для отработки рутинных операций молекулярного моделирования.

В связи с тем, что в процессе выполнения задачи будет создано несколько десятков файлов следует обратить особое внимание на их названия. Они должны быть осмысленными и содержать достаточно информации для того, что бы Вы смогли правильно идентифицировать свои файлы. В этом пособии для обозначения имени файла мы будем использовать звездочку, вместо которой Вам следует подставить какое-либо свое имя. Например, *.trg означает какой-то файл с расширением trg.

1. Запустите PyMol. Программу предназначенную для просмотра и редактирования молекулярных структур.
2. Создайте произвольный полипептид состоящий из 7-ми аминокислот. В состав пептида должны войти хотя бы по одному остатку аланина, валина, лизина и аспартата (глутамата). Сохраните полученную структуру в формате *.pdb.

***.pdb** – от Protein Data Bank, файлы предназначенные для хранения координат атомов молекул, а также вспомогательной информации о кристалле белка.

3. При помощи утилиты pdb2gmx создайте файл топологии (*.top) и структурный файл (*.gro) в формате GROMACS.

***.gro** – от GROmacs, структурный файл, предназначенный для хранения координат и скоростей атомов системы в формате пакета программ GROMACS.

***.top** – от TOPology, файл топологии, в котором хранится информация о всех связях, валентных и двугранных углах молекулы, а также о типах атомов молекулы.

Для этого используйте утилиту pdb2gmx, которая читает файл *.pdb и, используя информацию о силовых параметрах молекулы, взятые из файлов интерактивно выбранного силового поля, создает искомый файл топологии:

```
$ pdb2gmx -f *.pdb -p *.top -o *.gro -ignh -ter
```

В качестве силового поля укажите OPLS. Для концов полипептида выберете простой ионизированный вариант (COO⁻ и NH₃⁺).

Справка по любой из утилит GROMACS доступна из командной строки, например:

```
$ pdb2gmx -h
```

или через соответствующую закладку браузера (<http://manual.gromacs.org/>). Обратите внимание на то, какое число полипептидных цепей обнаружила программа в Вашем *.pdb-файле. При необходимости отредактируйте исходный файл при помощи утилиты gedit.

Оптимизация геометрии системы.

Первым этапом молекулярно динамических расчетов как правило является оптимизация геометрии системы. Прежде всего это связано с низким, порядка ангстрема, разрешением метода рентгеноструктурного анализа, в основном используемого для получения координат атомов молекулы белка. 1 ангстрем – это почти длина ковалентной связи, поэтому перед началом расчетов молекулярной динамики необходимо «отпустить» систему. Наиболее простым и распространенным методом минимизации чего-либо является метод градиентного спуска. Он заключается в том, что систему последовательно смещают вдоль каждой из координат в направлении обратном частной производной минимизируемого параметра (в нашем случае потенциальной энергии) по данной координате.

1. Скачайте с сайта нашей лаборатории (<http://erg.biophys.msu.ru/student/>) файлы em.mdp и run.mdp, с молекулярно-динамическими параметрами.

***.mdp** – от Molecular Dynamics Parameters, файл с инструкциями о том что и как считать. Справка доступна по адресу: http://manual.gromacs.org/online/mdp_opt.html.

2. Отредактируйте файл параметров em.mdp. Порог по максимальной силе (emtol), после достижения которого оптимизация останавливается, должен быть 100 кДж/моль нм.
3. Запустите препроцессор GROMACS. Который создаст основной входной файл для запуска молекулярной динамики и при этом проверит правильность всех исходных файлов и условий. На вход препроцессор получает уже знакомые файлы *.gro, *.top и *.mdp. На выходе получается один единственный файл *.tpr, содержащий в себе всё необходимое для запуска расчета молекулярной динамики.

***.tpr** – от Topology Portable binary Run. Файл с исходными данными для молекулярно-динамического расчета. Создается препроцессором grompp:

```
$ grompp -f em.mdp -c *.gro -p *.top -o *.tpr -v
```

Внимательно изучите полученную ошибку. Размеры бокса в *.gro файле меньше удвоенного радиуса обрезания невалентных взаимодействий. Исправьте исходный *.gro файл при помощи утилиты editconf:

```
$ editconf -f *.gro -o *.gro -box x y z -c
```

вместо x, y и z следует подставить числа, каждое из которых больше чем удвоенный параметр rlist в Вашем файле em.mdp. Затем перезапустите препроцессор:

```
$ grompp -f em.mdp -c *.gro -p *.top -o *.tpr
```

3. Запустите оптимизацию геометрии системы. Для этого служит основная утилита пакета GROMACS, mdrun:

```
$ mdrun -s *.tpr -deffnm * -v
```

На вход mdrun получает только один файл *.tpr, а на выход отдает три типа файлов, имена которых задаются ключевым словом deffnm (DEfault File Name):

***.trr** – бинарный файл, в котором содержатся координаты и скорости всех атомов системы на каждом из записанных шагов расчета. Частота записи данных в файл *.trr регулируется параметром nstxout и nstvout в файле *.mdp.

***.edr** – бинарный файл, в котором содержатся энергетические параметры молекулярной системы на каждом из записанных шагов. Частота записи в *.edr регулируется параметром nstfout в файле *.mdp.

*.log — текстовый файл, в который записываются данные о состоянии системы на каждом из выделенных шагов. Частота записи регулируется параметром nstlog *.mdp файла.

Просмотр полученной траектории.

После завершения процедуры оптимизации геометрии необходимо оценить ее результаты. Если с количественной оценкой все более или менее понятно, максимальная сила составила 100 кДж/моль нм, то какие именно изменения произошли с молекулой в процессе оптимизации необходимо выяснять отдельно. Наиболее простой способ - это посмотреть на происходящее глазами. Особое внимание обратите на масштабные структурные перестройки, а так же на образование и разрыв водородных и других невалентных связей.

Для визуализации молекулярной системы мы воспользуемся программой PyMOL, которая, как это следует из названия, написана на языке программирования python и обладает всеми его преимуществами. В программе доступна командная строка, позволяющая выполнять множество полезных действий не мышкой, а командой, что часто быстрее и удобнее. Прежде чем перейти к описанию наиболее востребованных команд условимся давать имена переменных, которые, как и имена файлов, Вам предстоит дать самостоятельно.

При работе с PyMOL имеют большое значение следующие команды:

1. Команда load, которая позволяет загрузить структуру из файла *.pdb в объект с заданным именем:

```
PyMol> load *.pdb, OBJ_NAME
```

2. Команда delete, позволяющая удалить объект, выделение или все сразу, как в нижеследующем примере:

```
PyMol> delete all
```

3. Команда select, которая, в частности, позволяет выделить атомы, остатки или молекулы по их именам или номерам. Первый аргумент команды (*SEL_NAME*) - имя выделения, а второй описывает что, собственно, выделяется. Так, например, ключевое слово id обозначает номер атома, name - его имя, resi - номер остатка, а resn - его имя. Ниже приведена пара примеров использования команды select:

```
PyMol> select SEL_NAME, resn ALA
```

```
PyMOL> select SEL_NAME, id 10 or id 18
```

Первая из них выделит все атомы остатков аланина, а вторая два атома с номерами 10 и 18.

4. Команда center, помещающая в начало координат центр масс выбранного объекта или выделения:

```
PyMOL> center SEL_NAME
```

5. Команда ray, предназначенная для качественной отрисовки текущей трёхмерной сцены. Рекомендуем использовать ее перед сохранением изображения понравившейся Вам сцены для отчета.

```
PyMol> ray
```

Прежде чем загрузить структуру в PyMOL необходимо пересоздать ее на основании *.gro файла, использованного для расчета, так как в исходном *.pdb файле и созданном pdb2gmx *.gro файле скорее всего не совпадают имена и номера атомов, что может привести к путанице.

1. При помощи утилиты editconf пересоздайте исходный структурный файл *.pdb на основании выданного pdb2gmx файла *.gro:

```
$ editconf -f *.gro -o *.pdb
```

Затем откройте вновь созданный *.pdb файл при помощи PyMol, для этого в командной строке PyMol выполните:

```
PyMol> load *.pdb, OBJ_NAME
```

2. Загрузите траекторию в PyMol. Для этого в командной строке PyMol выполните команду:

```
PyMol> load_traj *.trr, OBJ_NAME
```

где *OBJ_NAME* - имя объекта, в который на предыдущем шаге был загружен *.pdb файл.

3. Просмотрите полученную траекторию пользуясь кнопками внизу справа, в окне просмотра. Удостоверьтесь в правильности наблюдаемых процессов. Опишите изменения, произошедшие с полипептидом в процессе оптимизации геометрии. Особое внимание обратите на разрыв и образование водородных связей, если таковые имели место. Эта информация потребуется для отчета.

4. Подготовьте несколько иллюстраций процесса оптимизации геометрии для отчета. Например, общий вид молекулы до и после оптимизации. Для этого разверните молекулу подходящим образом, в командной строке PyMOL выполните команду "ray" и сохраните полученное изображение. Перед началом рендеринга (ray) лучше сменить цвет фона в PyMOL с черного на белый (Display/Background - White).

Запуск расчета молекулярной динамики.

Как следует из названия задачи, нас в основном будут интересовать движения внутри молекулы полипептида. Эти движения можно условно разделить на две группы: колебания относительно положений равновесия валентных связей и углов и вращения вокруг связей, т. е. изменение двугранных углов, что по сути своей и является собственно конформационным изменением. Важно понимать, что характерные времена колебательных движений лежат в области десятков или сотен фемтосекунд, в то время как характерное время поворота вокруг валентной связи часто составляет десятки и сотни пикосекунд.

Для того чтобы с достаточной точностью оценить параметры колебательных движений необходимо провести расчет и записать в файл траектории *.trj координаты атомов примерно раз в одну фемтосекунду. При этом достаточно провести относительно короткий расчет, длиной до 100 пс. В то же время, для того чтобы двугранные углы успели более или менее "замести" доступный им конформационный объем необходимо провести расчет как минимум длиной в 100 наносекунд. Нетрудно подсчитать, что при такой длине траектории и частоте записи в 1 фс в траекторный файл будет записано 10^8 шагов, что выльется в много гигабайт дискового пространства, а для обработки потребуется большой объем оперативной памяти. При этом для адекватного описания поведения двугранных углов вовсе не требуется знать значение угла каждую фемтосекунду, вполне достаточно одной пикосекунды.

В связи с этим мы проведем два расчета. Один короткий и подробно записанный – для анализа колебательных движений валентных связей и углов, а второй длинный и записанный менее подробно для анализа конформационных движений.

1. Создайте два новых файла *.mdp, отредактировав файл параметров run.mdp. Один файл должен быть составлен так, чтобы длина траектории составила 100 нс, при записи данных в выходные файлы каждую пс, а во втором файле длина траектории должна составить 100 пс, а период записи данных - 1 фс. В обоих случаях шаг интегрирования должен быть равен одной фс, интегратор - sd, временные константы термо- и баростатирования (tau_t и tau_p) - 0.1 пс. Полученным файлам необходимо дать вразумительные имена.

2. Запустите препроцессор GROMACS для обоих случаев:

```
$ grompp -f *.mdp -c *.gro -p *.top -o *.tpr -v
```

3. Запустите два расчета молекулярной динамики. Сначала короткий, а по его окончании - длинный:

```
$ mdrun -s *.tpr -deffnm * -v
```

Не закрывайте окно терминала из которого был запущен расчет! Это приведет к его остановке!!! Расчет займет около 10-12 часов, поэтому его необходимо запустить в конце рабочего дня.

Создание индекс-файлов.

Основные файлы, в которые mdrun записывает результаты расчета (*.trr и *.edr) имеют бинарный формат записи (нечитабельны текстовыми редакторами) и содержат в себе всю возможную информацию о системе. В то же время в большинстве случаев исследователю требуется получить какие-то конкретные данные лишь о части системы. В пакете программ GROMACS есть немало специальных программ, предназначенных для извлечения информации из бинарных файлов в текстовые. Однако всем им необходимо как-то передать что же именно мы хотим извлечь из траекторного файла. Для этого служат индекс файлы.

***.ndx** – от iNDeX. Текстовые файлы, предназначенные для произвольной группировки атомов. Служат для передачи дополнительных данных о системе препроцессору и программам обработки готовых траекторий.

Индекс файл содержит в себе собственно группы атомов. Каждая группа имеет имя, с которого начинается ее описание. Имя не должно содержать пробелов, записывается на отдельной строке, в квадратных скобках и пробелах: [GROUP_NAME]. Далее, с новой строки следуют номера атомов, входящих в группу, разделенные пробелами, знаками табуляции и конца строки. Для связей номера атомов лучше группировать в пары, для углов в тройки, для двугранных углов в четверки.

Для создания индекс-файлов мы воспользуемся программами PyMOL и gedit. Дальнейшие действия будут направлены на создание индексов, в которых будут перечислены номера атомов, образующих C-H связи метильного радикала, C=O и N-H связи пептидной группы; валентный угол C-C-C (для алифатического углеводорода), O-C-O угол карбоксильной группы и угол C-N-H (в пептидной группе); двугранные углы ϕ и ψ , а также углы χ (X-CA-CB-X) - углы поворота боковой цепи аминокислоты.

1. Откройте свой *.pdb файл при помощи пакета PyMol и выведите номера атомов молекулы. Для этого в командной строке PyMOL выполните команды:

```
PyMol> select SEL_NAME, all
```

```
PyMol> label SEL_NAME, id
```

2. Метильный радикал есть в аланине, валине, лейцине и изолейцине, а также в треонине и метионине. Следовательно в соответствующую группу должны попасть по три пары номеров атомов на каждый остаток аминокислоты, для того чтобы помянуть все три пары C-H связей метильного радикала. В каждой паре один номер – номер атома углерода, а второй – водорода.

3. C=O и N-H связи пептидной группы встречаются в полипептиде в количестве на единицу меньше, чем количество остатков. Соответствующие им группы должны содержать, соответственно, по шесть пар атомов каждая (в том случае, если аминокислот 7).

4. Угол C-C-C, для алифатической углеводородной цепочки встречается в остатках аргинина и лизина, причем во втором остатке таких угла два. Соответственно, следует перечислить в группе номера соответствующих атомов, сгруппировав их по три в строке. Порядок атомов в индексе принципиально важен: тот атом, который в молекуле в середине

должен быть упомянут вторым. В противном случае программа рассчитает значения не валентного угла, а нечто странное.

5. Угол O-C-O карбоксильной группы встречается в остатках аспартата и глутамата, а также на C-конце полипептидной цепи. Соответствующие номера следует записать в индекс файл.

6. Угол C-N-H пептидной группировки встречается в 7-ми аминокислотном полипептиде 6 раз. В PyMOL достаточно просто найти соответствующие номера атомов и записать их в индекс.

7. Угол ϕ - угол образованный атомами C-N-CA-C, ψ - атомами N-CA-C-N. Порядок упоминания атомов в индекс файле должен быть именно таким, так как в противном случае мы получим отрицательные (если изменить порядок) или смещенные на некоторую константу (если поменять концевые атомы, скажем, на атомы водорода) углы. В индексе углы ϕ и ψ следует поместить в разные группы, а атомы лучше сгруппировать по четыре в строке.

8. Двугранный угол, описывающий поворот боковой цепи остатка аминокислоты (χ) в именах атомов записывается так: X-CA-CB-Y, можно выбрать любые X и Y. Но лучше чтобы это были тяжелые атомы, если это возможно. Углы χ в индекс файле также следует поместить в отдельную группу.

Извлечение данных из траектории.

После того, как сформированы индекс файлы можно приступить извлечению данных из траектории. Для этого служат утилиты `g_bond` и `g_angle`. Они получают на вход траекторию, топологию (*.tpr) и индекс файл, а на выходе создают текстовый файл с расширением `xvg`. Это основной формат файлов, с результатами, который используется всеми без исключения утилитами пакета программ GROMACS.

*.**xvg** – текстовый файл, содержащий таблицу с данными, сформированную одной из программ обработки GROMACS. Не следует открывать эти файлы блокнотом или иными стандартными текстовыми редакторами `windows`, так как файлы эти часто имеют большой объем и блокнот просто повиснет. Для их просмотра служит `gedit`.

1. Утилита `g_bond` предназначена для создания усредненной гистограммы длин связей, содержащихся в группе индекс-файла, а также для построения зависимости длин связей от времени. Группу, для которой следует рассчитать длины связей пользователь выбирает интерактивно, после запуска программы. Кроме того, она способна вывести в *.xvg файл таблицу, в первой колонке которой будет записано время, а в последующих значение длин связей, записанных в соответствующей группе индекс файла:

```
$ g_bond -f *.trr -s *.tpr -n *.ndx -d *.xvg -noaverdist -noaver
```

2. Утилита `g_angle` предназначена для анализа валентных и двугранных углов. Принцип работы такой же как и у `g_bond`, только не требует на вход *.tpr. Нужно помнить о том, что в *.xvg файле, сформированном `g_angle` вторая колонка - это среднее значение всех (!!!) углов. Для запуска утилиты выполните команды:

```
$ g_angle -f *.trr -n *.ndx -ov *.xvg -type angle -all
```

```
$ g_angle -f *.trr -n *.ndx -ov *.xvg -type dihedral -all
```

Анализ полученных данных

Обработка данных данных ведется в среде PyLAB. Вызов соответствующего интерпретатора команд на машинах с установленной ОС Windows осуществляется из системного меню: Пуск->Программы->IPython->pylab, а на машинах с ОС GNU/Linux - из командной строки:

```
$ ipython --pylab
```

Примем для среды PyLAB такую же систему обозначения переменных, как и для PyMOL, т. е. Имена переменных, которые определяет пользователь, будем давать курсивом: *NAME*.

Загрузка файлов с XVG в среду PyLAB

Прежде чем приступить к анализу данных необходимо импортировать данные из файлов в формате xvg в среду PyLAB. Для этого необходимо выяснить с какой строки файла начинаются собственно числовые данные. Выяснить это можно отобразив файл при помощи программы less с ключом -N, при этом в начале каждой строки будет выведен ее номер:

```
$ less -N bond.xvg
```

Также можно открыть файл при помощи gedit и установив курсор на последнюю незначащую строку посмотреть на ее номер (справа внизу). После этого можно импортировать соответствующий файл в PyLAB:

```
IPython> BONDS = loadtxt('*.xvg', skiprows = N)
```

В качестве N следует указать количество строк в начале файла, не содержащих числовых данных. В результате будет создана переменная, содержащая числовые данные из файла.

Построение графиков

Загруженные данные можно визуализировать при помощи команды plot:

```
IPython> plot(BONDS[:,0], BONDS[:,1], '-r', label='C-H bond')
```

В результате будет построен график зависимости величины приведенной во второй колонке файла (длина связи, угол, двугранный угол) от первой колонки (время). Аргумент '-r' - задаёт формат отображения данных, сплошная линия красного цвета. Именной аргумент label задает название ряда данных в легенде. Легенду можно отобразить при помощи команды legend:

```
IPython> legend()
```

Подписи к осям могут быть добавлены при помощи команд xlabel и ylabel:

```
IPython> xlabel('Time, ps')
```

Название графика можно вывести при помощи команды title:

```
IPython> title('Bonds Length')
```

Полученный график можно интерактивно масштабировать и сохранить в виде файла в формате PNG. Справка по аргументам plot доступна из командной строки IPyton:

```
IPython> help(plot)
```

Построение распределения вероятности величины, определение параметров распределения

Для построения гистограммы распределения числового ряда из второй колонки переменной BONDS воспользуемся функцией hist:

```
IPython> hist(BONDS[:,1], 20, normed=1)
```

Второй аргумент функции задает количество столбцов в гистограмме. Третий аргумент означает, что распределение будет нормировано. Определим также параметры распределения в предположении, что оно нормально, а именно среднее:

```
IPython> ba = average(BONDS[:,1])
```

и стандартное отклонение:

```
IPython> bs = std(BONDS[:,1])
```

и построим соответствующую функцию распределения на одном графике с гистограммой:

```
IPython> x = linspace(min(BONDS[:,1]),max(BONDS[:,1]),100)
```

```
IPython> plot(x,exp(-(x-ba)**2/bs**2)/sqrt(2*pi*bs**2),'-r')
```

Построение автокорреляционных функций

Для построения автокорреляционных функций воспользуемся функцией `acorr` (слэш в конце первой строки означает перенос строки и писать его не обязательно!!!):

```
IPython> X,Y,L1,L2 = acorr(BONDS[:,1], \
detrend=mlab.detrend_linear, maxlags=len(BONDS[:,1])-1)
```

Теперь в переменных `x` и `y` записаны данные, собственно гистограмма. Для ее построения выполните команду:

```
IPython> plot(X, Y, '-r')
```

Сглаженная кривая при помощи скользящего среднего может быть получена и построена следующим образом:

```
IPython> Y_AVER = movavg(x, 10)
```

```
IPython> plot(x[4:-5], Y_AVER)
```

Второй аргумент функции `movavg` задает ширину окна скользящего среднего.

Построение Фурье спектров

Теорема Фурье гласит, что любой сигнал конечной длины можно сколь угодно точно аппроксимировать достаточно длинным рядом гармонических функций. Этот математический факт является основой спектрального анализа. Собственно спектром называют зависимость модуля вклада соответствующей гармоники от её номера. Для того чтобы провести разложение сигнала в ряд Фурье выполните:

```
IPython> BOND_FOUR = rfft(BOND[:,1])
```

Первое значение в полученном ряду данных соответствует нулевой частоте, второе - частоте, соответствующей полной длине сигнала, третье - половине и так далее, до максимальной частоты, соответствующей половине частоты дискретизации. Для того чтобы правильно задать диапазон частот создадим при помощи команды `linspace` переменную `FREQ`:

```
IPython> FREQ = linspace(BOND[0,1]/len(BOND[:,1]), \
1.0/2*BOND[0,1], len(BOND_FOUR))
```

Теперь можно построить Фурье спектр и определить частоту колебаний соответствующей, например, длины связи.

```
IPython> plot(FREQ, BOND_FOUR)
```

Построение двумерных карт распределений (карт Рамачандрана)

Чтобы создать двумерную карту распределения плотности вероятности для двух величин, хранящихся в переменных, например, `PHI` и `PSI` можно воспользоваться функцией `histogram2d`:

```
IPython> F,X,Y = histogram2d(PHI, PSI, 50)
```

Третий аргумент задает количество карманов гистограммы по каждой из осей. Затем для правильного построения карты следует уменьшить количество точек в массивах `x` и `y`:


```
IPython> X = movavg(X,2)
```

```
IPython> Y = movavg(Y,2)
```

Кроме того для правильного отображения карты Рамачандрана необходимо транспонировать выдачу `histogram2d`:

```
IPython> F = transpose(F)
```

Теперь в переменных `x`, `y` и `p` хранится собственно карта, которую можно построить при помощи функции `contourf`:

```
IPython> countourf(X, Y, F, 100)
```

Четвертый аргумент задает количество цветовых градаций на карте.

Подготовка отчета о проделанной работе.

Отчет должен представлять собой **связный текст**, разделы которого позволяют понять **ЧТО** Вы делали и **ЗАЧЕМ**? В отчет не следует включать много теории, основной объем должен быть посвящен ходу работы и описанию полученных результатов.

В отчете должны быть даны ответы на следующие вопросы:

1. Какие изменения претерпевает молекула пептида в процессе оптимизации геометрии? Почему происходят эти изменения?
2. Сравните друг с другом частоты и амплитуды колебаний валентных связей, углов и двугранных углов. Сделайте выводы.
3. Заметны ли различия в конформационной подвижности между разными аминокислотами? Как это можно показать?
4. Каким конформациям соответствуют наблюдаемые в модельной системе карты Рамачандрана?