

Методы работы с динамическими молекулярными моделями, построенными в среде открытого 3D-редактора Blender

Филиппов С.В.

ИМПБ РАН – филиал ИПМ им. М.В. Келдыша РАН

fsv141@mail.ru

В рамках ранее предложенного автором метода построения динамических молекулярных моделей в среде открытой 3D-платформы Blender с использованием Blender Python API на основе данных о молекулярно-динамическом исследовании, предложены специфичные для платформы Blender реализации (на языке Python и средствами графической среды Blender) методов детального исследования элементов динамически смоделированных макромолекулярных структур. Реализован визуальный мониторинг длин межатомных связей, а также построение автоматически адаптирующейся мультикурсовой динамической системы наблюдения за интересующей исследователя группой атомов на протяжении всего молекулярно-динамического исследования. Выполнен ряд сервисных модернизаций базового программного обеспечения, выполняющего автоматическое построение динамических Ван-дер-Ваальсовых молекулярных моделей. Эти изменения включают в себя возможность выборочного импорта (по списку) атомов из серии PDB-файлов – снимков молекулярно-динамических траекторий, а также ввод в число параметров строящейся модели множителя Ван-дер-Ваальсовых радиусов. Все описываемые реализации предлагаемых методов были опробованы на молекулярно-динамической 3D-модели β_2 -адренорецептора. Предъявленные методы подтверждают выводы, сделанные авторами молекулярно-динамического исследования нативной апо формы β_2 -адренорецептора о стабильности межспиральной водородной связи Ser74-Trp158, а кроме того, предоставляют молекулярно-динамическую информацию в «сфокусированном» и «контрастированном» виде, давая обширные возможности для дополнительного анализа.

Ключевые слова: Blender, Python, API, 3D моделирование, молекулярные модели, визуализация, β_2 -адренорецептор, молекулярно-динамическое моделирование.

Methods of working with dynamic molecular models, built in an environment of open 3D editor Blender

Filippov S.V.

IMPB RAS – Branch of KIAM RAS

In the framework of the previously proposed method of constructing dynamic molecular models in an environment of the open 3D Blender platform using the Blender Python API based on data on the molecular dynamics study, Blender-specific implementations (in Python and the Blender GUI) of detailed research methods dynamically modeled macromolecular structures. The implementation of visual monitoring of the lengths of interatomic bonds and the construction of an automatically adapted “multi-angle view” dynamic observation system for a group of atoms of interest to the researcher during the whole molecular dynamic investigation is shown. In addition, a number of service upgrades of the basic software have been performed, which automatically constructs dynamic Van-der-Waals molecular models. These changes include the possibility of selectively importing (from the list) atoms from a series of PDB files – snapshots of molecular dynamic trajectories, as well as entering into the number of parameters for constructing a model of the Van-der-Waals radii multiplier. All described implementations of the proposed methods were tested on the molecular-dynamic 3D model of β_2 -adrenergic receptor. The presented methods confirm the conclusions made by the authors of the molecular dynamics study of the native β_2 -adrenoreceptor apo form on the stability of the inter-helical hydrogen bond Ser74-Trp158, and in addition provide molecular dynamic information in a “focused” and “contrasted” form, giving the researcher extensive opportunities for additional analysis.

Key words: Blender, Python API, 3D modeling, molecular models, visualization, rendering, β_2 -adrenoreceptor, molecular dynamics simulation.

1. Введение

Трёхмерные динамические модели биологических макромолекул, построенные по данным молекулярно-динамического моделирования, содержат в себе огромное количество информации, сложно поддающейся визуальному анализу из-за однородного и нередко монотонного характера динамических данных. Существенно затрудняет визуальный анализ и количество анализируемых объектов (атомов и их групп), исчисляемое тысячами [1–3]. Различные методы селективного представления такого рода информации и несущих её объектов призваны не просто повысить удобство восприятия молекулярно-динамических данных, но и акцентировать внимание исследователя на определённых участках биополимера.

Эти и многие другие сервисные функции обычно встроены в специализированные программы визуализации молекулярных моделей [4–7], но в универсальной графической среде Blender их, конечно же, нет.

2. Постановка задачи

Таким образом, для надления графического окружения Blender некоторыми сервисными функциями, присущими специализированным программным пакетам визуализации молекулярных моделей, нами было принято решение реализовать нижеперечисленные методы и необходимые для их создания программные компоненты, а также продемонстрировать работоспособность предлагаемых методов на динамической модели реальной биомолекулы.

В конечном итоге интегрированная среда 3D-моделирования, анимации и рендеринга Blender должна обеспечить следующие возможности:

1. Произвольный выбор атомов при построении трёхмерной динамической молекулярной модели.
2. Определение множителя Ван-дер-Ваальсовых радиусов для строящейся модели.
3. Автоматизированный выбор атомов в уже построенной модели по таким специфическим для этих объектов свойствам, как имя и номер остатка (аминокислоты или нуклеотида), заряд атома и т. п.
4. Мониторинг интересующих исследователя межатомных расстояний на любом временном участке молекулярно-динамического исследования.
5. Построение визуализирующего окружения (виртуальные камеры, источники света и т.п.), автоматически подстраивающегося под непрерывно меняющееся положение наблюдаемых объектов.

Последняя из вышезаявленных целевых функций не имеет прямых аналогов в существующих специализированных пакетах визуализации молекулярных моделей [4–7] и, будучи перспективной формой визуального представления молекулярно-динамических данных,

способна обеспечить более информативный показ таких областей особого внимания, как активные центры биомолекул.

3. Исходные данные, ПО и методы

Работа проводилась исключительно в среде Blender 2.79b с интегрированным интерпретатором Python v.3.5.3. Эта же среда была использована и в качестве интегрированной среды разработчика (IDE).

Тестовый объект – динамическая молекулярная модель свободного (АПО) β_2 -адренорецептора (β_2 -АР), полученная нами из 425 PDB-файлов, представляющих собой снимки траекторий атомов, рассчитанных методом молекулярно-динамического моделирования Т.В. Богдан и Е.С. Алексеевым в работе о роли в конформационной стабильности β_2 -АР консервативной межспиральной водородной связи Ser74-Trp158 в сайте связывания холестерина [8] по исходным данным рентгеноструктурного анализа с наилучшим разрешением PDB ID: 2RH1 [9, 10].

Ранее построенная нами и используемая в данной работе динамическая молекулярная модель β_2 -АР содержит молекулу рецептора без молекул воды и ионов NaCl. Соответственно, размер тестового объекта составил 4 666 атомов. Объём геометрических данных при этом составил 755 892 вершины и 1 493 120 треугольных граней. Динамика макромолекулы была представлена в виде траекторий всех атомов β_2 -АР, позиции которых зафиксированы в виде ключевых кадров с интервалом 20 кадров. «Масштаб» временной шкалы: 1 кадр – 1 пс.

4. Результаты и обсуждение

Созданный нами ранее Python-скрипт для построения динамических молекулярных моделей в среде Blender, использующий Blender Python API [11] был модифицирован для того, чтобы у создаваемых с его помощью моделей можно было определять диаметр сфер, обозначающих атомы в виде множителя, применяемого к Ван-дер-Ваальсовым [12–14] радиусам атомов модели.

Также в программный код была внесена возможность импорта атомов, номера и имени которых (относительно имён и номеров атомов во входных PDB-файлах) перечислены в специальном текстовом файле. Эта модификация существенно повышает удобство и гибкость работы с программой, а в ряде случаев и существенно повышает скорость построения динамических молекулярных моделей.

Однако приводить детальное описание вышеперечисленных изменений нет необходимости, поскольку все они не Blender-специфичны. Заметим лишь, что в качестве файла-списка выборочно импортируемых атомов было принято решение

использовать обычный текстовый файл в общепринятом формате PDB [15] из соображений удобства подготовки таких файлов – простым редактированием любого из исходных PDB-файлов.

Скрипт, выполняющий поиск и выделение заданных атомов в уже построенной молекулярной модели, показан на рисунке 1.

```

1 import bpy
2
3 for i in bpy.data.objects: # перебор всех объектов сцены
4     # рассматриваем лишь объекты, имеющие пользовательское
5     # свойство 'GRP_NUM'
6     if 'GRP_NUM' in bpy.data.objects[i.name]:
7         # выделение групп с номерами 74 (Ser) и 158 (Trp)
8         if (bpy.data.objects[i.name]['GRP_NUM'] == 158 or
9             bpy.data.objects[i.name]['GRP_NUM'] == 74):
10            # выделение объекта
11            bpy.data.objects[i.name].select = True

```

Рис. 1. Python-скрипт, выполняющий автоматическое выделение атомов, наделённых заданными свойствами.

Вышеприведённый программный код примечателен своей простотой, наглядностью и доступностью для использования.

В нашей работе с его помощью были выделены интересующие нас аминокислоты Ser74 и Trp158, которые были перенесены на отдельный слой сцены Blender штатным для этого редактора способом.

Для отслеживания расстояний интересующей нас водородной связи между атомами кислорода № 734 (Ser74) и азота № 2074 (Trp158) был написан скрипт, приведённый на рисунке 2, в задачу которого входило вычисление межатомных расстояний и назначение их в виде длины цилиндрическому столбику гистограммы с фиксацией этого параметра в ключевых кадрах.

```

15 first_frm = 0 # начальный кадр анимации
16 last_frm = 8500 # конечный кадр анимации (8500)
17 frm_step = 20 # расстояние между ключевыми кадрами
18 atm_name_1 = "734_OG" # Имена атомов, связь (расстояние)
19 atm_name_2 = "2074_NE1" # между которыми нас интересует
20
21 # Создаём "столбик гистограммы" с именем "GRAPH" и высотой 1
22 # единичная высота важна для способа трансформации - масштабирования
23 bpy.ops.mesh.primitive_cylinder_add(vertices = 32, radius = 0.25,
24     depth = 1, location = (0, 0, 0.5))
25 bpy.context.active_object.name = "GRAPH"
26 grph = bpy.data.objects['GRAPH'] # получим ссылку на объект-столбик
27 # установим центр объекта (origin) в центр нижней грани, совпадающий
28 # с центром координат, простейшим способом - через 3D курсор
29 bpy.context.scene.cursor_location = (0.0, 0.0, 0.0)
30 bpy.ops.object.origin_set(type='ORIGIN_CURSOR')
31
32 # Идём по таймлайну в диапазоне анимации с шагом, равным
33 # расстоянию между ключевыми кадрами
34 for frm in range(first_frm, last_frm, frm_step):
35     bpy.data.scenes['Scene'].frame_set(frm) # уст. тек. кадра
36     # вычисляем расстояние, соединяющее два атома,
37     print(frm) # номер кадра - индикатор прогресса
38     dst = distance(bpy.data.objects[atm_name_1].location,
39                 bpy.data.objects[atm_name_2].location)
40     grph.scale[2] = dst
41     # фиксируем высоту столбика гистограммы
42     grph.keyframe_insert(data_path = "scale", index = 2, frame = frm)

```

Рис. 2. Основная часть Python-скрипта, вычисляющая расстояние между заданными атомами на протяжении всего молекулярно-динамического моделирования – 8.5 н.с.

Поскольку результаты измерения межатомных расстояний фиксировались в сцене в форме анимации, то и отобразить их можно в стандартном редакторе Blender–Graph Editor, предназначенном для работы с анимационными кривыми.

Из снимков экрана Graph Editor, отображающих участки молекулярно-динамического моделирования длительностью чуть более 1 н.с. были составлены итоговые графики изменения длины водородной связи Ser74-Trp158, показанные на рисунках 3 и 4.

Одной из главных целей данной работы была разработка методики построения адаптирующегося окружения, визуализирующего динамику аминокислот Ser74 и Trp158. Для решения этой задачи было решено применить подход, использовавшийся при построении графика изменений длины водородной связи между атомами кислорода № 734 (Ser74) и азота № 2074 (Trp158) – то есть предварительный расчёт положения нескольких камер, обеспечивающих круговой обзор интересующего нас участка макромолекулы относительно атомов, образующих водородную связь, и фиксация положения камер в пространстве в ключевых кадрах на протяжении всего молекулярно-динамического моделирования.

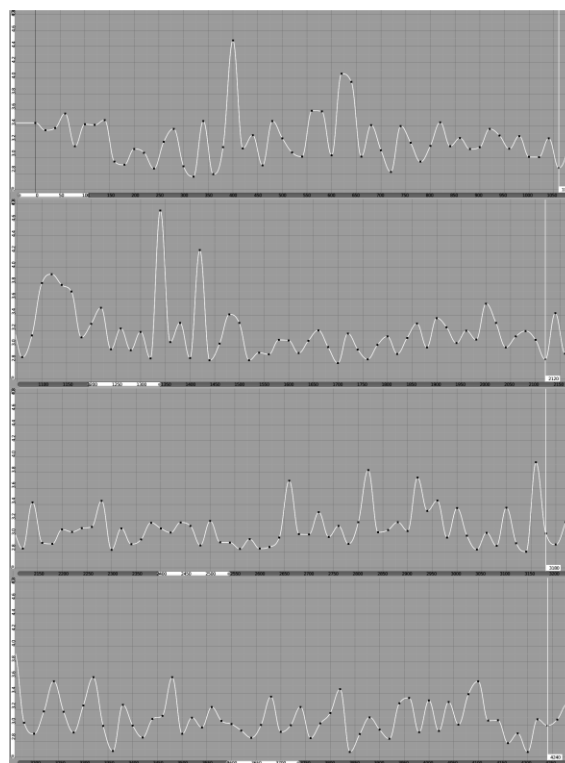


Рис. 3. Анимационная кривая, отображающая динамику водородной связи Ser74-Trp158 на участке молекулярно-динамического моделирования 0–4.25 н.с.

Конструктивно визуализирующее окружение (рис. 5) было решено оформить в виде окружности RIG, несущей четыре камеры и объекта типа Empty с именем TARGET, предназначенного для нацеливания камер. Положение этих двух объектов должно рассчитываться так, чтобы они всегда располагались в центре условной линии, соединяющей атомы кислорода № 734 (Ser74) и азота № 2074 (Trp158).

Нацеливание и привязка камер к несущей их окружности выполнялась штатными для Blender средствами – с помощью ограничителей (Constraint) типа Track To и Follow Path соответственно (рис. 6). Позиция каждой из камер относительно несущей их окружности RIG определялась параметром Offset ограничителя Follow Path.

Для автоматического расчёта позиций окружности RIG и объекта нацеливания TARGET относительно непрерывно меняющихся позиций атомов аминокислот Ser74-Trp158 был написан Python-скрипт, основная часть которого продемонстрирована на рисунке 7.

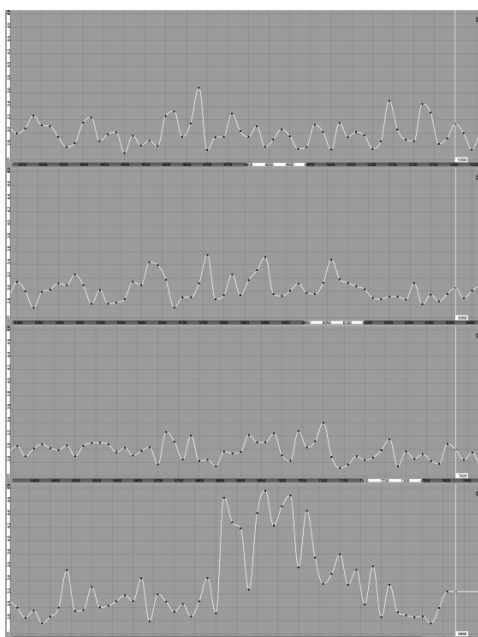


Рис. 4. Анимационная кривая, отображающая динамику водородной связи Ser74-Trp158 на участке молекулярно-динамического моделирования 4.25–8.5 н.с.

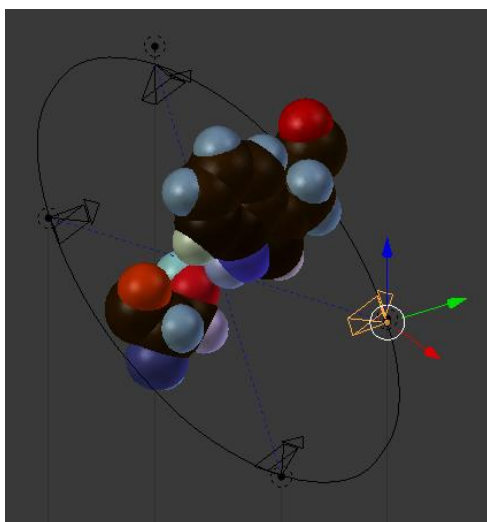


Рис. 5. Конструкция адаптирующейся к динамике молекулярной модели визуализирующей мультиракурсной системы.

Маркировку атомов аминокислот выполнили с помощью текстурирования.

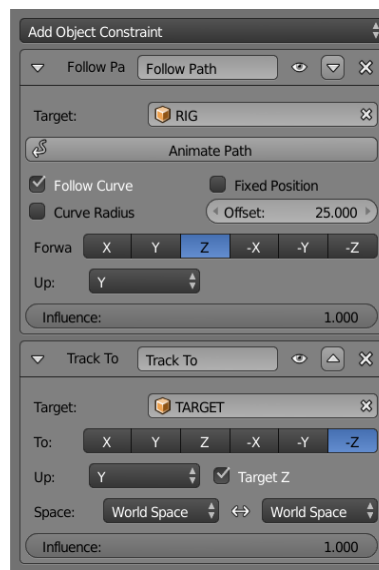


Рис. 6. Параметры ограничителей одной из камер, привязывающих её к окружности RIG (Follow Path) и нацеливающих на объект TARGET (Track To).

```

32 for frm in range(first_frm, last_frm, frm_step):
33     bpy.data.scenes['Scene'].frame_set(frm) # устанавливаем текущий кадр
34     # вычисляем середину отрезка, соединяющего два атома,
35     # расстояние между которыми нас интересует и сразу перемещаем
36     # в эту позицию "пустышку"
37     print(frm)
38     trgt = bpy.data.objects['TARGET']
39     rig = bpy.data.objects['RIG']
40     trgt.location = midpnt(bpy.data.objects[atm_name_1].location,
41                          bpy.data.objects[atm_name_2].location)
42     # центр окружности - траектории для камер помещаем в середину
43     # отрезка, "связывающего" атомы и совпадающего с "пустышкой"
44     rig.location = trgt.location
45
46     # фиксируем позицию "пустышки" в ключевом кадре
47     trgt.keyframe_insert(data_path = "location", frame = frm)
48     # фиксируем позицию траектории камер в ключевом кадре
49     rig.keyframe_insert(data_path = "location", frame = frm)
50
51     # поворачиваем плоскость окружности и "пустышку"
52     # перпендикулярно оси, соединяющей атомы
53     trgt.rotation_mode = 'QUATERNION'
54     rig.rotation_mode = 'QUATERNION'
55     direction = (bpy.data.objects[atm_name_2].location -
56                bpy.data.objects[atm_name_1].location)
57     trgt.rotation_quaternion = direction.to_track_quat('Z', 'Y')
58     rig.rotation_quaternion = direction.to_track_quat('Z', 'Y')
59     # фиксируем вращение в ключевом кадре
60     trgt.keyframe_insert(data_path = "rotation_quaternion", frame = frm)
61     rig.keyframe_insert(data_path = "rotation_quaternion", frame = frm)

```

Рис. 7. Основная часть Python-скрипта, отслеживающая непрерывно меняющиеся позиции целевых атомов и выполняющая синхронные перемещения объектов TARGET и RIG.

В конечном итоге после рендеринга видеопотоков всех четырёх ракурсных камер, а также камеры, нацеленной на индикатор длины водородной связи, мы получили 5 синхронных видеопоследовательностей длительностью 8500 кадров (длина всего молекулярно-динамического исследования) с разрешением Full HD, которые были сведены во встроенном видеоредакторе Blender (Video Sequence Editor) в единый видеофрагмент с разделённым на 4 равные области кадром для каждого из ракурсов и демонстрацией столбика гистограммы, динамически отображающего длину межспиральной водородной связи Ser74-Trp158 [17].

5. Выводы

Представленные методы работы с динамическими молекулярными моделями способны, по нашему мнению, обеспечить основные потребности исследователя в средствах анализа таких моделей.

Использование Blender в работе по визуальному анализу динамики водородной связи Ser74-Trp158 продемонстрировало мощь и самодостаточность данной среды для молекулярного моделирования и анализа динамических молекулярных моделей.

Необходимый при этом объём программирования в решении типовых задач визуализации молекулярных моделей не превышает такового у таких специализированных пакетов как PyMol [5], а такие Blender-специфичные аспекты, как использование Blender Python API, нивелируются справочной системой, глубоко интегрированной в пакет.

Полученная нами аналитическая визуализация динамики межспиральной водородной связи Ser74-Trp158 [17] в динамической модели β_2 -АР подтвердила стабильность этой связи в рамках молекулярно-динамического исследования у нативной безлигандной апо-формы β_2 -АР [8, 16].

6. Благодарности

Работа выполнена при поддержке РФФИ, проект 18-07-00354.

7. Список литературы

1. Филиппов С.В., Соболев Е.В., Использование технологий профессиональной компьютерной графики для визуализации результатов научных исследований. *Компьютеры и суперкомпьютеры в биологии*. Под ред. Лахно В.Д., Устинина М.Н. Москва–Ижевск: Институт компьютерных исследований, 2002. С. 476–497.
2. Филиппов С.В. *Методы и алгоритмы визуализации структурных и динамических данных, характеризующих макромолекулярные структуры*: дис. ... канд. физ.-мат. наук: Пушино, 2014.
3. Филиппов С.В., Сивожелезов В.С., Ким В.Л., Сычев В.В., Устинин М.Н. Программа трехмерного моделирования и визуализации конформационной динамики биомолекул Maya-K-PDB. *Математическая биология и биоинформатика*. 2015. Т. 10. № 1. С. 260–282. doi: [10.17537/2015.10.260](https://doi.org/10.17537/2015.10.260).
4. *RasMol and OpenRasMol – Molecular Graphics Visualisation Tool*. URL: <http://www.openrasmol.org> (дата обращения: 29.06.2018).
5. *PyMol: a molecular visualization system on an open source foundation, maintained and distributed by Schrödinger*. URL:

- <http://www.pymol.org> (дата обращения: 29.06.2018).
6. *SWISS-MODEL*. URL: <https://swissmodel.expasy.org> (дата обращения: 29.06.2018).
7. *VMD (Visual Molecular Dynamics)*. URL: <http://www.ks.uiuc.edu/Research/vmd/> (дата обращения: 29.06.2018).
8. Богдан Т.В., Алексеев Е.С. Роль консервативной межспиральной водородной связи Ser74-Trp158 в сайте связывания холестерина в конформационной стабильности β_2 -адренорецептора (молекулярно-динамическое моделирование). *Журнал структурной химии*. 2017. Т. 58. № 2. С. 402–409. doi: [10.1134/S002247661702024X](https://doi.org/10.1134/S002247661702024X).
9. Cherezov V., Rosenbaum D.M., Hanson M.A., Rasmussen S.G., Thian F.S., Kobilka T.S., Choi H.J., Kuhn P., Weis W.I., Kobilka B.K., Stevens R.C. *Science*. 2007. V. 318. № 5854. P. 1258.
10. *High resolution crystal structure of human B2-adrenergic G protein-coupled receptor*. URL: <http://pdb.org/pdb/explore/explore.do?structureId=2rh1>. (дата обращения: 29.06.2018).
11. *Blender Documentation Contents*. URL: https://docs.blender.org/api/blender_python_api_current/contents.html (дата обращения: 29.06.2018).
12. Зефилов Ю.В., Зоркий П.М. *Журнал структурной химии*. 1974. Т. 15. № 1. С. 118–122.
13. Зефилов Ю.В., Зоркий П.М. *Журнал структурной химии*. 1976. Т. 17. № 4. С. 745–746.
14. *LANL Periodic Table of Elements*. URL: <http://periodic.lanl.gov/26.shtml> (дата обращения: 29.06.2018).
15. *PDB. File Format Documentation*. URL: <http://www.wwpdb.org/documentation/file-format> (дата обращения: 29.06.2018).
16. Novikov G.V., Sivozhelezov V.S., Kolesnikov S.S., Shaitan K.V. Investigation of the influence of external factors on the conformational dynamics of rhodopsin-like receptors by means of molecular dynamics simulation. *J. Recept. Signal Transduct.* 2014. V. 34. № 2. P. 104–118. doi: [10.3109/10799893.2013.863918](https://doi.org/10.3109/10799893.2013.863918).
17. *Визуализация (Blender) мол. динамики Ser74-Trp158 B2-адренорецептора АРО*. URL: <https://youtu.be/uQwWV1xm3xk> (дата обращения: 29.06.2018).