ПРИМЕНЕНИЕ ИНСТРУМЕНТОВ DATA-SCIENCE ДЛЯ ПРЕДСКАЗАТЕЛЬНОГО МОДЕЛИРОВАНИЯ И ПОСТРОЕНИЯ 3D-ВИЗУАЛИЗАЦИИ СТРУКТУР НА ОСНОВЕ PYTHON И ДОПОЛНЕННОЙ СИМУЛЯЦИИ ASE

Кудаева И. В., студентка группы ИТб-161, IV курс
Пылов П. А., студент группы ИТб-162, IV курс
Протодьяконов А.В., [1] к.т.н., доцент
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Рассматриваемая в статье задача направлена на программную реализацию модели предсказания и взаимодействия атомов в синтезе органических соединений. Особые технологии визуализации, такие как магнитно-резонансная томография, позволяют представить для изучения в графическом виде молекулярный состав тканей. Ядерный магнитный резонанс (далее ЯМР) является тесно связанной технологией, которая использует те же принципы, чтобы понять структуру и динамику соединений белков и молекул химических элементов.

По состоянию на сегодняшний день[2] исследователи во всем мире проводят эксперименты по ЯМР для лучшего понимания структуры и динамики молекул в таких областях, как экология, фармацевтика и материаловедение. Анализирование и поиск всех возможных решений рассматриваемой в статье задачи осуществляется сотрудниками лабораторий химии и математики в Университете Бристоля, Университете Кардиффа, Имперском колледже и Университете Лидса[3].

В статье разрабатывается алгоритм, который моделирует предсказательное магнитное взаимодействие между двумя атомами в молекуле (то есть скалярную константу связи).

Использование визуализации ЯМР для представления структуры и динамики молекулы необходимо, чтобы точно прогнозировать так называемые «скалярные связи». Это эффективно отображает магнитные взаимодействия между парой атомов. Сила этого магнитного взаимодействия зависит от промежуточных электронов и химических связей, которые составляют трехмерную структуру молекулы.

В языке программирования высокого уровня Python для работы с атомами и молекулами существует множество библиотек. Одной из самых распространённых библиотек является библиотека *ase*, которая предоставляет

[1] Научный руководитель
[2] Февраль 2020 года
[3] https://research-information.bris.ac.uk/en/publications/a-semipredictive-molecular-model-of-1dh-substrate-specificity(c308b782-e508-4e00-be52-4d69b1effbab).html

разработчику методы и классы для реализации атомной симуляции различных элементов и соединений.

Первое, что нужно сделать, – это установить библиотеку *ase*. Установка пакета осуществляется выполнением команды «*pip install ase*» в терминале командной строки (рисунок 1)

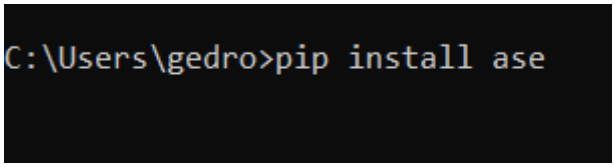

Рисунок 1. Установка библиотеки.

После установки требуемых библиотек в среде разработки осталось только подключить данную библиотеку к проекту решения с помощью ключевого слова *import* (рисунок 2)

```
import ase
```

Рисунок 2. Импорт библиотеки в проект решения.

Теперь следует загрузить набор необходимых данных[4] для создания и 3D-визуализации молекул (рисунок 3)

```
molecule_structure_data = pd.read_csv('structures.csv')
```

Рисунок 3. Загрузка данных из набора.

Проведем анализирование столбцов исходного набора данных. Столбец «molecule_name» содержит в себе идентификатор уникального химического соединения. Столбец «atom_index» представляет собой итератор числа элементов, входящих в состав соединения, стоит отметить, что отсчёт ведется с нуля. Столбец «atom» содержит символ химического элемента, например, «H» – водород. Столбцы «x», «y», «z» заполняются координатами соответствующих элементов (рисунок 4)

[4] Набор данных представлен для выполнения конкурсных компетенций официальном сайтом Университета Бристоля, город Бристоль, Соединенное Королевство Великобритании и Северной Ирландии https://www.bristol.ac.uk/study/postgraduate/2020/eng/msc-data-science

	molecule_name	atom_index	atom	x	y	z
2146835	dsgdb9nsd_120257	0	C	-0.137966	1.535818	-0.306223
2146836	dsgdb9nsd_120257	1	C	-0.000669	0.036478	-0.077058
2146837	dsgdb9nsd_120257	2	C	1.070680	-0.684256	-0.911770
2146838	dsgdb9nsd_120257	3	O	2.343527	-0.154410	-0.517098
2146839	dsgdb9nsd_120257	4	C	1.754547	-1.185875	0.286569
2146840	dsgdb9nsd_120257	5	C	0.734804	-0.525093	1.224799
2146841	dsgdb9nsd_120257	6	C	1.282955	0.409617	2.298188
2146842	dsgdb9nsd_120257	7	C	0.218311	0.927494	3.258338
2146843	dsgdb9nsd_120257	8	O	0.865175	1.743086	4.223472
2146844	dsgdb9nsd_120257	9	H	0.837333	2.026882	-0.265178
2146845	dsgdb9nsd_120257	10	H	-0.570998	1.731750	-1.293712
2146846	dsgdb9nsd_120257	11	H	-0.791637	2.002715	0.437085
2146847	dsgdb9nsd_120257	12	H	-0.971363	-0.457998	-0.185460
2146848	dsgdb9nsd_120257	13	H	0.998544	-1.098791	-1.911205
2146849	dsgdb9nsd_120257	14	H	2.343810	-2.082644	0.444398
2146850	dsgdb9nsd_120257	15	H	0.103298	-1.294895	1.684106
2146851	dsgdb9nsd_120257	16	H	2.034808	-0.127513	2.890736
2146852	dsgdb9nsd_120257	17	H	1.799853	1.257528	1.839802
2146853	dsgdb9nsd_120257	18	H	-0.548643	1.495665	2.708667
2146854	dsgdb9nsd_120257	19	H	-0.292841	0.074570	3.736450
2146855	dsgdb9nsd_120257	20	H	0.197331	2.088149	4.822953

Рисунок 4. Образец набора данных.

Любую химическую молекулу можно представить в виде совокупности составляющих её атомарных атомов. Поэтому следующим шагом в решение задачи визуального представления молекулы соединения будет определение координат и символов химических элементов, из которых в последующем будет смоделировано соединение. Загрузка соответствующих координат атомов и их химических символов представлено на рисунке 5

```
atoms_coordinates = molecule.iloc[:, 3:].values
print(atoms_coordinates)
```

```
[[-1.37966489e-01  1.53581847e+00 -3.06223240e-01]
 [-6.68589500e-04  3.64783321e-02 -7.70577519e-02]
 [ 1.07067957e+00 -6.84255729e-01 -9.11769641e-01]
 [ 2.34352726e+00 -1.54409940e-01 -5.17098092e-01]
 [ 1.75454685e+00 -1.18587498e+00  2.86569431e-01]
 [ 7.34804370e-01 -5.25093296e-01  1.22479939e+00]
 [ 1.28295538e+00  4.09616565e-01  2.29818767e+00]
 [ 2.18310900e-01  9.27493505e-01  3.25833835e+00]
 [ 8.65174630e-01  1.74308619e+00  4.22347154e+00]
 [ 8.37332910e-01  2.02688227e+00 -2.65178006e-01]
 [-5.70998355e-01  1.73174969e+00 -1.29371178e+00]
 [-7.91636713e-01  2.00271453e+00  4.37084934e-01]
 [-9.71363017e-01 -4.57997679e-01 -1.85459728e-01]
 [ 9.98543861e-01 -1.09879088e+00 -1.91120475e+00]
 [ 2.34381031e+00 -2.08264420e+00  4.44397746e-01]
 [ 1.03298143e-01 -1.29489511e+00  1.68410636e+00]
 [ 2.03480782e+00 -1.27513067e-01  2.89073626e+00]
 [ 1.79985252e+00  1.25752767e+00  1.83980200e+00]
 [-5.48642810e-01  1.49566497e+00  2.70866663e+00]
 [-2.92840775e-01  7.45698285e-02  3.73645032e+00]
 [ 1.97330965e-01  2.08814887e+00  4.82295290e+00]]
```

```
symbol_of_atoms = molecule.iloc[:, 2].values
print(symbol_of_atoms)
```

```
['C' 'C' 'C' 'O' 'C' 'C' 'C' 'C' 'O' 'H' 'H' 'H' 'H' 'H' 'H' 'H' 'H' 'H'
 'H' 'H' 'H']
```

Рисунок 5. Загрузка данных химических элементов.

Для моделирования атомов в графическом виде потребуется загрузка модуля *Atoms* из библиотеки *ase*. После этого приступаем к написанию программного кода для создания соединения из представленных в наборе данных атомов. Для решения этой части наиболее оптимальным вариантом будет написание функции, которая принимает в качестве аргумента название набора данных из загруженного списка химических элементов. После этого, для соответствующего имени набора соединения, функция получает координаты атомов и символы названий химических атомов, из которых состоит синтезируемое соединение (рисунок 6)

```
from ase import Atoms
import ase.visualize

system = Atoms(positions=atoms, symbols=symbols)
```

```
def view(molecule):
    # Выбор молекулы для 3D-визуализации
    molecule_draw = molecule_structure_data[molecule_structure_data['molecule_name'] == molecule]

    # загружаем координаты для переменной mol
    xcart = molecule_draw.iloc[:, 3:].values

    # получаем набор химических символов элементов
    symbols = molecule_draw.iloc[:, 2].values

    # визуализация модели молекулы
    system = Atoms(positions=xcart, symbols=symbols)
    print('Уникальный код синтезируемой молекулы: %s.' %molecule)
    return ase.visualize.view(system, viewer="x3d")

random_molecule = random.choice(molecule_structure_data['molecule_name'].unique())
view(random_molecule)
```

Рисунок 6. Функция создания 3D-модели молекулы.

Осуществляя вызов функции с требуемым именем набора компонент химических элементов, создаём 3D-модель молекулы. В приведенном примере имя выбирается случайным образом из всего набора загруженных данных (рисунок 7-8)

Уникальный код синтезируемой молекулы: dsgdb9nsd_041008.

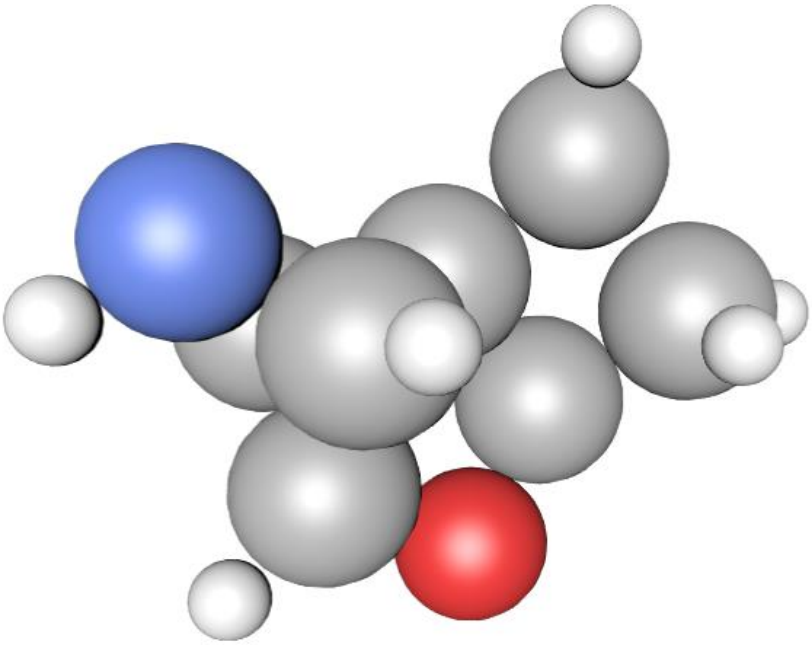

Рисунок 7. Модель синтезированной молекулы из представленных атомов.

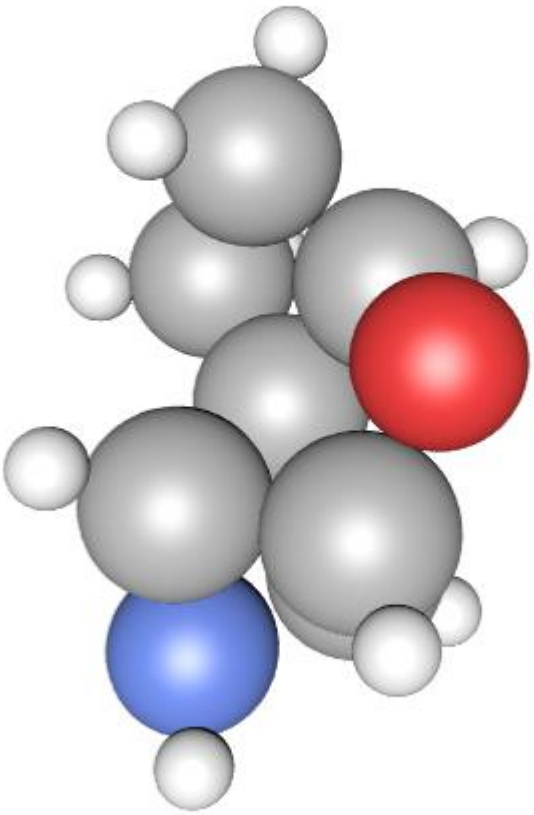

Рисунок 8. Модель синтезированной молекулы из представленных атомов.

Разработанная модель позволяет смоделировать крайне сложные органические молекулы, которые могут содержать в своем составе атомы углерода различных конфигураций (sp3, sp2, sp) и малые циклы. Так, например, на рисунках 7-8 получена модель и структурная формула молекулы C_7H_9NO, которая называется[5] *6-оксо-3-азотетрацикло-[5,2,0,0(1,4),0(2,5)]-нонан* (рисунок 9).

[5] Название соединения и его структурная формула получены на основе международной базы данных химических молекул Spectral Database for
Organic Compounds SDBS https://sdbs.db.aist.go.jp/sdbs/cgi-bin/direct_frame_top.cgi

6-oxa-3-azatetracyclo[5.2.0.$0^{1,4}$.$0^{2,5}$]nonane

Рисунок 9. Структурная формула

В настоящий момент возможность практического синтеза некоторых соединений остается неизученной, поскольку синтез отдельных молекул может быть нереализуем из-за отсутствия методов стабилизации нескольких сопряженных малых циклов. Однако, стоит отметить, что этот недостаток можно доработать путём серии лабораторных экспериментов и уточнением модели на основе квантовой химии.

Существенное достоинство реализованной модели состоит том, что она может предсказывать модели с учетом валентности атомов.

Список литературы:

1. Christopher Bishop. Pattern Recognition and Machine Learning, Plenum press, New York – London, 1971
2. M. Narasimha Murty, V. Susheela Devy. Introduction to pattern recognition and machine learning, IISc press, New Jersey – London, 2015
3. S. Theodoridis, A. Pikrakis. Introduction to Pattern Recognition: A MATLAB Approach, Elsevier press, 2010