

**БИБЛИОТЕКА RUPHARM КАК ИНСТРУМЕНТ ДЛЯ РЕШЕНИЯ ЗАДАЧ
ФАРМАКОКИНЕТИКИ****Р. С. Крашенинников^а, И. И. Митричев^б, М. А. Усова^в**

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский химико-технологический университет имени Д. И. Менделеева»,
г. Москва, Российская Федерация

^а ORCID: <https://orcid.org/0009-0007-3434-031X>, ✉ krasheninnikov.r.s@muctr.ru^б ORCID: <https://orcid.org/0000-0001-5733-3226>, mitrichev.i.i@muctr.ru^в ORCID: <https://orcid.org/0009-0002-9034-7165>, 204137@muctr.ru

Аннотация: в последние годы заметно возросла популярность моделирования при создании медицинской продукции. Особое значение приобретает изучение фармакокинетики разрабатываемых лекарственных средств. Определение параметров таких моделей представляет собой задачу многомерной оптимизации, которую успешно решают эволюционные алгоритмы. В данном исследовании представлена новая открытая библиотека на языке Python, предназначенная для фармакокинетического моделирования и оптимизации параметров фармакокинетических моделей с использованием эволюционных алгоритмов.

Ключевые слова: оптимизация, эволюционный алгоритм, генетический алгоритм, вдохновленный природой алгоритм, фармакокинетика, камерная модель, разработка лекарственной продукции.

Для цитирования: Крашенинников Р. С., Митричев И. И., Усова М. А. Библиотека PyPharm как инструмент для решения задач фармакокинетики. *Успехи кибернетики*. 2025;6(1):158–169.

Поступила в редакцию: 25.11.2024.

В окончательном варианте: 16.12.2024.

THE RUPHARM LIBRARY FOR SOLVING PHARMACOKINETIC PROBLEMS**R. S. Krasheninnikov^а, I. I. Mitrichev^б, M. A. Usova^в**

Mendeleev Russian University of Chemical Technology, Moscow, Russian Federation

^а ORCID: <https://orcid.org/0009-0007-3434-031X>, ✉ krasheninnikov.r.s@muctr.ru^б ORCID: <https://orcid.org/0000-0001-5733-3226>, mitrichev.i.i@muctr.ru^в ORCID: <https://orcid.org/0009-0002-9034-7165>, 204137@muctr.ru

Abstract: recently, simulation has gained significant popularity in the creation of medical products, particularly in the study of pharmacokinetics for developed drugs. Determining the parameters of these models involves a multidimensional optimization problem, which evolutionary algorithms can effectively solve. This study introduces a new open-source Python library designed for pharmacokinetic modeling and optimizing the parameters of pharmacokinetic models using evolutionary algorithms.

Keywords: optimization, evolutionary algorithm, genetic algorithm, nature-inspired algorithm, pharmacokinetics, compartment model, drug design.

Cite this article: Krasheninnikov R. S., Mitrichev I. I., Usova M. A. The PyPharm Library for Solving Pharmacokinetic Problems. *Russian Journal of Cybernetics*. 2025;6(1):158–169.

Original article submitted: 25.11.2024.

Revision submitted: 16.12.2024.

С каждым годом моделирование все чаще применяется в разработке лекарственных препаратов, что подтверждается ростом числа упоминаний о нем в различных источниках [1–2]. Эффективность такого подхода, в частности, подтверждается экономической выгодой [3–5].

Весь процесс разработки лекарственного препарата можно разделить на три этапа: стадия исследования (дизайн молекулы), стадия тестирования (проверка оптимальной дозы и безопасности лекарства) и стадия рынка (коммерциализация лекарства) [6]. Зачастую именно на разработку тратится большая часть бюджета и времени до выхода лекарства на рынок (срок может достигать до 10 лет) [6, 7]. Чтобы ускорить данный процесс применяются методы математического моделирования.

При тестировании лекарственного средства математическое моделирование применяется для двух основных целей: исследования фармакокинетики (ФК) и фармакодинамики (ФД). Фармакокинетика занимается изучением скорости распространения лекарственного средства и/или его метаболитов

в организме и силы, действующей на этот процесс [8]. В свою очередь, фармакодинамика изучает положительное или негативное влияние лекарства на организм [7]. Другими словами, ФК можно описать как «то, что организм делает с лекарством», а ФД — как «то, что лекарство делает с организмом».

В данной работе мы сообщаем о разработке нового программного обеспечения для решения задач фармакокинетики: библиотеки PyPharm с открытым исходным кодом на языке Python. Хотя совместное рассмотрение ФК/ФД дает большее количество информации о применимости и качестве лекарственного средства, на данный момент библиотека ограничена моделированием ФК. ФК модели позволяют определить неизвестные параметры и проводить интерполяцию или экстраполяцию результатов, что, в свою очередь, способствует выбору оптимального режима дозирования препарата [9, 10].

Структура статьи следующая: сначала мы рассмотрим, какие модели вообще используются в ФК-исследованиях, затем приведем обзор программ-аналогов. Далее следует объяснение основного функционала библиотеки PyPharm для программиста (API) и пример ее использования. Заметим, что для использования библиотеки требуется лишь базовое понимание языка Python, который достаточно прост и сейчас широко используется в научных исследованиях.

Типы моделей ФК

В литературе выделяют два основных типа моделей фармакокинетики, а именно камерная (компарментная) модель и физиологически обоснованная фармакокинетическая модель (РВРК) [11].

Камерная модель описывает распространение лекарства через различные камеры. В каждой камере поддерживается состояние кинетической однородности, обеспечивающее равномерное и с постоянной скоростью распределение лекарственного препарата [11].

РВРК модель, в отличие от камерной модели, связывает ткани и органы с кровотоком, основываясь на физиологических и анатомических особенностях организма. Она также подробно описывает структурные элементы, используя такие параметры, как объем ткани, кровоток через ткань, коэффициент распределения между тканью и плазмой и другие [11].

Хотя модель РВРК более точно описывает фармакокинетику лекарственного средства в организме, эта модель требует большого количества входных параметров, собранных эмпирически. Также размер модели увеличивает сложность вычислений. В свою очередь, камерная модель создана с большими допущениями и, соответственно, не может идеально моделировать фармакокинетический процесс, но сбор данных, построение и решение модели являются более доступными задачами. Стоит также отметить, что для многих лекарственных препаратов моделирование с помощью камерной модели дает достаточно точные решения [12].

Виды камерных моделей

Простейший вид камерной модели представляет собой однокамерную модель, где весь организм представлен в виде гомогенной камеры, которая обменивается лекарством со внешней средой. Данная модель возможна при больших допущениях и приводит к простому математическому описанию [13].

Более сложные модели строятся из нескольких камер и называются многокамерными моделями, одна из них представлена на рис. 1. Данные камеры в большинстве случаев представляют собой имитацию какого-либо органа или ткани в организме. Однако иногда они выступают в роли абстрактных структур, не имеющих физиологического смысла.

Принцип камерных моделей может быть продемонстрирован на наиболее распространенном их виде — двухкамерной модели (рис. 1). Она может быть описана математически в виде системы дифференциальных уравнений (1)–(2) [14]

$$V_1 \frac{dc_1}{dt} = -k_{12}V_1c_1 + k_{21}V_2c_2 - k_{10}V_1c_1, \quad (1)$$

$$V_2 \frac{dc_2}{dt} = k_{12}V_1c_1 - k_{21}V_2c_2, \quad (2)$$

где c_1 — концентрация вещества в центральной камере (нг/мл), V_1 — объем центральной камеры (мл), c_2 — концентрация в периферической камере (нг/мл), V_2 — объем периферической камеры (мл), k_{12} — константа скорости переноса фармакологического вещества из центральной камеры в периферическую

(мин^{-1}), k_{21} — константа скорости переноса фармакологического вещества из периферической камеры в центральную (мин^{-1}), k_{10} — константа скорости элиминации фармакологического вещества из центральной камеры (мин^{-1}), все константы скорости имеют первый порядок.

Стоит отметить, что основным параметром математического моделирования фармакокинетики является концентрация. Концентрацией называют отношение количества лекарства (измеряется в мг или в молях) к объему камеры. Именно концентрациями оперируют в фармакокинетике из-за легкости их измерения в крови.

Для описания фармакокинетики препарата организм представляют как набор отдельных камер, содержащих часть введенной дозы препарата за исключением элиминированной (выведенной дозы), а скорость перемещения препарата из одной камеры в другую и выведения вещества из организма определяется константами скорости [15].

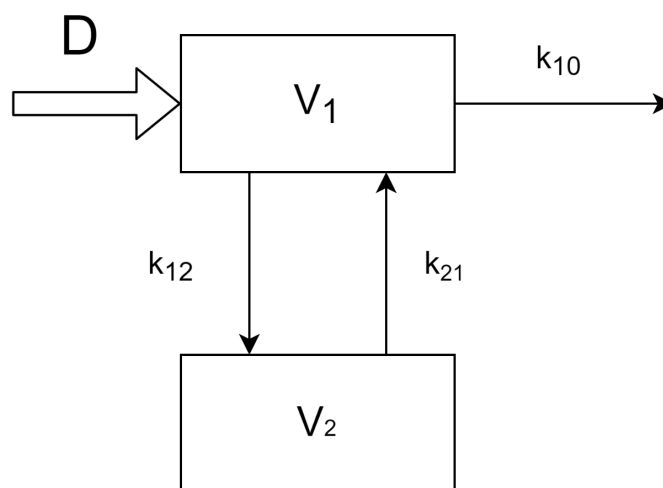


Рис. 1. Двухкамерная модель [14]

Математические выражения, описывающие камерную модель, получаются путем записи соответствующих балансов масс через концентрации и решения результирующей системы дифференциальных уравнений [13]. Решая данную систему, можно получить зависимости концентрации лекарственного средства в камере в зависимости от времени (t) или концентрацию лекарственного препарата в камере в определенный момент t . В камерных моделях рассматривается одно уравнение баланса массы для каждой камеры, охватываемой моделью. Таким образом, в многокамерной модели изначально требуется столько уравнений баланса масс, сколько используется камер.

Программное обеспечение

Камерные модели отличаются простотой, поэтому исследователи часто создают свое внутреннее программное обеспечение для моделирования ФК. Однако существуют и общедоступные коммерческие программы для построения фармакокинетических моделей с высокой точностью, ниже перечислены наиболее известные из них:

Simcyp[®] (www.simcyp.com) — это коммерческая программа для моделирования процессов ADME (всасывания, распределения, метаболизма и выведения лекарств). Simcyp[®] объединяет экспериментальные данные с демографической, физиологической и генетической информацией для прогнозирования взаимодействия лекарств и варибельности в популяциях. Программа умеет моделировать ситуации перорального приема лекарств с учетом таких факторов, как прием пищи и анатомические изменения. Так же Simcyp[®] для моделирования распределения лекарства использует различные модели для учета различных физиологических аспектов. Цель данной программы состоит в том, чтобы поддержать разработку лекарств, предоставляя информацию о том, как лекарства действуют на разных людей, и прогнозируя эффективность и безопасность взаимодействия лекарств. Из-за своего функционала Simcyp[®] является самым используемым программным обеспечением в области фармакокинетики [17, 18].

GastroPlusTM (www.simulations-plus.com) является коммерческим программным обеспечением, которое помогает моделировать, как лекарственные препараты всасываются, распределяются, метабо-

лизируются и выводятся из организма. Данное программное обеспечение известно своим простым в использовании интерфейсом и высококачественной графикой. Функционал программы заключается в способности использовать настраиваемые модели, в том числе модели для детей и различных животных (крысы, собаки, мыши, обезьяны). GastroPlusTM является одним из самых востребованных программных обеспечений в области разработок лекарств [17, 18].

PK-Sim[®] (www.open-systems-pharmacology.org) — это программное обеспечение с открытым исходным кодом, используемое для комплексного моделирования, как лекарственные препараты перемещаются и взаимодействуют в организме. Оно объединяет анатомические и физиологические параметры человека и животных для создания надежных моделей. PK-Sim[®] работает совместно с другим программным обеспечением под названием MoVi[®] для расширенного физиологического моделирования, которое помогает определить воздействие лекарств на организм. Это позволяет исследователям прогнозировать, как лекарства поведут себя в различных сценариях, в том числе у разных видов животных и при различных условиях (например, голодание или кормление). PK-Sim[®] известен своей способностью изучать поведение лекарств у пациентов с особыми состояниями, такими как цирроз печени [17, 18].

Intellipharm[®] PK (www.intellipharm.com) является коммерческим программным обеспечением, разработанным для моделирования концентрации лекарственных средств в плазме крови при пероральном приеме препарата. В Intellipharm[®] PK используется модель, в которой желудок и кишечник рассматриваются как один большой резервуар. Программное обеспечение связывает эту модель всасывания с более простыми фармакокинетическими моделями (одно- или двухкамерными), что помогает симулировать, как лекарства растворяются и перемещаются с течением времени. Но данная модель не полностью учитывает сложную природу ЖКТ и не может учитывать различия между отдельными людьми в всасывании, распределении, метаболизме и выведении лекарственных средств без клинических данных [17].

Cloe[®] PK (www.cyprotex.com) — это коммерческий инструмент моделирования, способный прогнозировать всасывание при приеме внутрь, а также распределение и выведение из организма. Эта модель представляет просвет желудочно-кишечного тракта в виде ряда из пяти отделов. Cloe[®] PK объединяет физико-химические данные и данные ADME для прогнозирования продолжительности действия лекарств в крови, тканях и органах [17].

PKQuest (www.pkquest.com) является бесплатным программным обеспечением, используется в образовательных целях и для исследований в области PK. PKQuest разработан на языке программирования Java, и включает в себя возможности различных методов моделирования на основе клинических данных ФК человека или крысы. Основное преимущество программы — это минимальное количество вводимых пользователем параметров, по средствам классификации лекарственных препаратов и предварительно запрограммированные оптимизированные данные ФК человека или крысы [18].

MaxSim2 — программа так же разработана в образовательных целях для обучения ФК и ФД, а также для доклинического моделирования. MaxSim2 предоставляет распространенные модели ФК и ФД и позволяет изменять параметры моделей в режиме реального времени, чтобы визуализировать графики концентрации и времени отклика [18].

Большинство из представленных программ являются дорогостоящими и зарубежными, что ограничивает возможности исследователей на территории России. Единственным отечественным продуктом в этой области является программное обеспечение «Симург» (Simurg) от компании «Симург-Фарм» (simurgpharm.ru). Оно предназначено для математического моделирования доклинических и клинических исследований лекарственных средств.

Кроме того, некоторые лекарственные средства вводятся с использованием носителя, и вещество постепенно высвобождается из него уже в организме. Описание ФК при внутривенном введении наночастиц с лекарством представляет собой значительный интерес, а большая часть программных продуктов не имеет возможностей моделирования ФК в таком случае.

В связи с этим было принято решение создать продукт с открытым исходным кодом, который позволит создавать и оптимизировать параметры фармакокинетических моделей. В основу ПО была заложена созданная для этой цели общедоступная библиотека на языке Python. А поиск параметров ФК моделей был произведен при помощи вдохновленных природой эволюционных алгоритмов оптимизации.

Библиотека PyPharm

Созданная в ходе работы библиотека *PyPharm* [19] способна производить моделирование ФК с использованием камерных моделей, а также производить поиск параметров модели по экспериментальным данным.

Основным модулем библиотеки является модуль *models*, который на текущий момент имеет в себе три типа моделей, отраженных в соответствующих классах:

- *BaseCompartmentModel* — классическая камерная модель;
- *MagicCompartmentModel* — классическая камерная модель с дополнительно введенным параметром *magic_coefficient*, отвечающим за преобразование единиц измерения входной концентрации в единицы измерения выходной концентрации. Этот параметр необходим, в тех случаях, когда доза препарата задана в одних единицах измерения, а концентрация в органах в других, и нет явного способа преобразования;

- *ReleaseCompartmentModel* — камерная модель, учитывающая высвобождение лекарственного вещества из носителя по заданному закону.

Библиотека построена на принципах объектно-ориентированного программирования (ООП) и подразумевает наследование моделей друг от друга для избежания повторного написания кода при разработке. Так два последних класса наследуются от *BaseCompartmentModel*, поэтому основные методы классов подобны и отличаются лишь набором входных параметров. Далее будет описана суть их работы.

Метод `__init__`

Данный метод отвечает за инициализацию параметров модели, и в базовом варианте принимает следующие аргументы:

- *configuration_matrix* — настроечная матрица модели, отображающая константы перехода между камерами. Значение a_{ij} отражает константу перехода вещества из i -й камеры в j -ю;
- *outputs* — вектор констант перехода во вне камер. В данном случае a_i отражает константу элиминации из i -й камеры;
- *volumes* — вектор объемов камер;
- *numba_option* — флаг необходимости использования библиотеки *numba* при оптимизации параметров;
- *use_shared_memory* — флаг необходимости использования общей памяти при оптимизации параметров.

Все неизвестные параметры модели в настроечных матрицах и векторах, пользователь может задать значением *None*, тогда их поиск будет производиться при оптимизации модели.

Для класса *MagicCompartmentModel* добавляются дополнительные параметры:

- *magic_coefficient* — параметр, отвечающий за преобразование данных. Простое число, на которое необходимо умножить входные данные, чтобы преобразовать их в выходные. Так же может быть задан *None*, если неизвестен;
- *exclude_compartments* — список номеров камер, которые нужно исключить из процесса преобразования.

Дополнительные параметры имеются и для класса *ReleaseCompartmentModel*:

- *v_release* — объем гипотетической камеры, из которой происходит высвобождение. Может быть задан *None*, если неизвестен;
- *release_parameters* — список параметров функции высвобождения. Данные параметры передаются в функцию высвобождения и могут быть заданы *None*, если неизвестны;
- *release_compartment* — номер камеры, в которую происходит высвобождение;
- *release_function* — функция высвобождения;
- *with_accumulate* — флаг необходимости учесть аккумуляцию вещества;
- *accumulation_parameters* — список параметров функции высвобождения.;
- Данные параметры передаются в функцию высвобождения и могут быть заданы *None*, если неизвестны;
- *accumulation_function* — функция аккумуляции.

Метод `__call__`

Метод `__call__` позволяет вызывать объект, классов моделей, как функцию и на выходе дает численное решение системы дифференциальных уравнений, описывающей камерную модель. В базовом варианте функция принимает следующие аргументы:

- *t_max* — предельное время расчета;
- *c0* — вектор начальных концентраций (может не передаваться, если передано *d* и *compartment_number*);
- *d* — вводимая доза (может не передаваться, если передано *c0*);
- *compartment_number* — номер камеры, в которую вводится доза (может не передаваться, если передано *c0*);
- *max_step* — максимальный шаг при решении СДУ;
- *t_eval* — вектор временных точек, в которых необходимо получить решение.

На выходе метод возвращает результат работы функции *solve_ivp* [20] библиотеки `numpy`.

Для класса *ReleaseCompartmentModel* смысл параметров немного видоизменяется, и *compartment_number* больше не требуется, а *c0* обозначает начальную концентрацию в гипотетической камере.

Методы `load_optimization_data` и `optimize`

Данные методы используются для поиска параметров камерной модели и используются в связке.

Метод *load_optimization_data* позволяет загрузить в модель экспериментальные данные, для ее дальнейшей оптимизации. Он принимает следующие параметры:

- *teoretic_x* — вектор значений времени замера экспериментальных данных;
- *teoretic_y* — матрица с экспериментальными значениями по известным камерам;
- *know_compartments* — вектор с номерами камер, по которым есть данные;
- *c0* — вектор начальных концентраций (может не передаваться, если передано *d* и *compartment_number*);
- *d* — вводимая доза (может не передаваться, если передано *c0*);
- *compartment_number* — номер камеры, в которую вводится доза (может не передаваться, если передано *c0*);
- *w* — матрица весов, отражающая достоверность данных, по которым производится оптимизация.

Для класса *ReleaseCompartmentModel*, передаваемые параметры меняются по аналогии с методом `__call__`.

Метод *optimize* производит непосредственный поиск неизвестных параметров с использованием нескольких алгоритмов. Для всех классов моделей он принимает одни и те же параметры:

- *method* — метод, который будет использован для оптимизации. Может принимать строковые значения: `country_optimization` (использование метода взаимодействующих стран [10, 15, 16]), `GA` (использование генетического алгоритма) или иные, тогда оптимизация производится функцией *minimize* [20] библиотеки `scipy`;
- *user_method* — принимает объект функции или класс, метод, который пользователь хочет использовать для оптимизации;
- *method_is_func* — флаг является ли пользовательский метод функцией (по умолчанию истина);
- *optimization_func_name* — метод, который нужно вызвать у объекта пользовательского класса (по умолчанию `__call__`);
- *max_step* — максимальный шаг при численном решении системы дифференциальных уравнений;
- *metric* — метрика, определяющая целевую функцию. Может принимать строковые значения: `R2` (используется показатель R^2), `norm` (используется норма матрицы отклонений);
- `**kwargs` — именованные аргументы, которые необходимо передать для настройки оптимизационного алгоритма.

Выбранный в методе *optimize* подход позволяет оптимизировать данные не только способами, встроенным библиотеку, но и выбранными пользователем, что значительно повышает исследовательский потенциал.

Методы `plot_model` и `get_kinetic_params`

Два данных метода позволяют вывести информацию о модели.

Метод `plot_model` позволяет вывести графики распределения вещества в камерах и в базовом варианте принимает аргументы:

- `compartment_numbers` — список камер, в которых нужно построить графики распределения;
- `compartment_names` — словарь имен камер, где ключ — это номер камеры, а значение — отображаемое имя;
- `left` — левая граница по времени на графиках;
- `right` — правая граница по времени на графиках;
- `y_lims` — словарь ограничений по оси y для камер. Ключом в данном случае является номер камеры, а значением кортеж (y_{min}, y_{max}) ;
- `**kwargs` — именованные аргументы, передаваемые в метод `__call__`.

Для класса `ReleaseCompartmentModel` добавляется флаг `plot_accumulation` отражающий необходимость построения данных в теоретической камере.

Дополнительные особенности библиотеки

Как упоминалось ранее, библиотека `PyPharm` опционально способна использовать библиотеку `numba` для своих вычислений. `Numba` [21] — это JIT-компилятор для `CPython 2`, самой популярной реализации языка `Python`, который, как следует из названия, написан на `C`. `Numba` реализован в виде библиотеки, которую можно загрузить в программы, запущенные в интерпретаторе `CPython`, без необходимости обновления или замены самого интерпретатора. Изначально он был создан для языка `Python`, активно использующего массивы и числовые скаляры в циклах, чтобы пользователям не приходилось переписывать свой код на низкоуровневых языках для повышения производительности. В рамках библиотеки `PyPharm` `numba` позволяет ускорить работу кода за счет компиляции такого функционала, как вычисление целевой функции.

Необходимо также упомянуть и о целевой функции, которая лежит в основе оптимизации фармакокинетических моделей в рамках библиотеки. Для количественной оценки точности модели, а также для подбора параметров используется функция (3):

$$F = \sum_{i=1}^k \sum_{j=1}^n \frac{(C_{ij}^{calc} - C_{ij}^{ex})^2}{(\bar{C}_k^{ex} - C_{ij}^{ex})^2} \quad (3)$$

где k — количество известных наборов данных, n — количество данных в наборе, C_{ij}^{calc} — вычисленное значение для i -го набора в j -й точке, C_{ij}^{ex} — экспериментальное значение для i -го набора в j -й точке, \bar{C}_k^{ex} — среднее значение по набору экспериментальных данных.

Стоит упомянуть о возможности работы библиотеки с общей памятью (`shared memory`) [22]. Данная технология доступна в библиотеке начиная с версии 1.3.0, она удобна для получения информации о ходе процесса оптимизации извне запущенного кода. Данное действие возможно при помощи обращения к памяти по имени, которое хранится в поле `memory_name` объекта модели. Обращение может быть осуществлено при помощи стандартного класса `ShareableList` библиотеки `multiprocessing`.

Пример: поиск параметров модели, описывающей ФК $[^{25}\text{Mg}^{2+}]$ порфириновых наночастиц в организме крысы при однократном, внутривенном введении

Далее будет рассмотрен пример использования библиотеки для поиска параметров модели, описывающей ФК $[^{25}\text{Mg}^{2+}]$ порфириновых наночастиц в организме крысы при однократном, внутривенном введении. Данный лекарственный препарат представляет интерес в роли нейропротектора, способного помочь при ишемическом инсульте.

В статье [23], авторами предлагается пятикамерная (рис. 2) модель для описания распространения данного лекарственного вещества в организме человека.

В оригинальной статье не учитывался объем отдельных камер, однако для демонстрации воз-

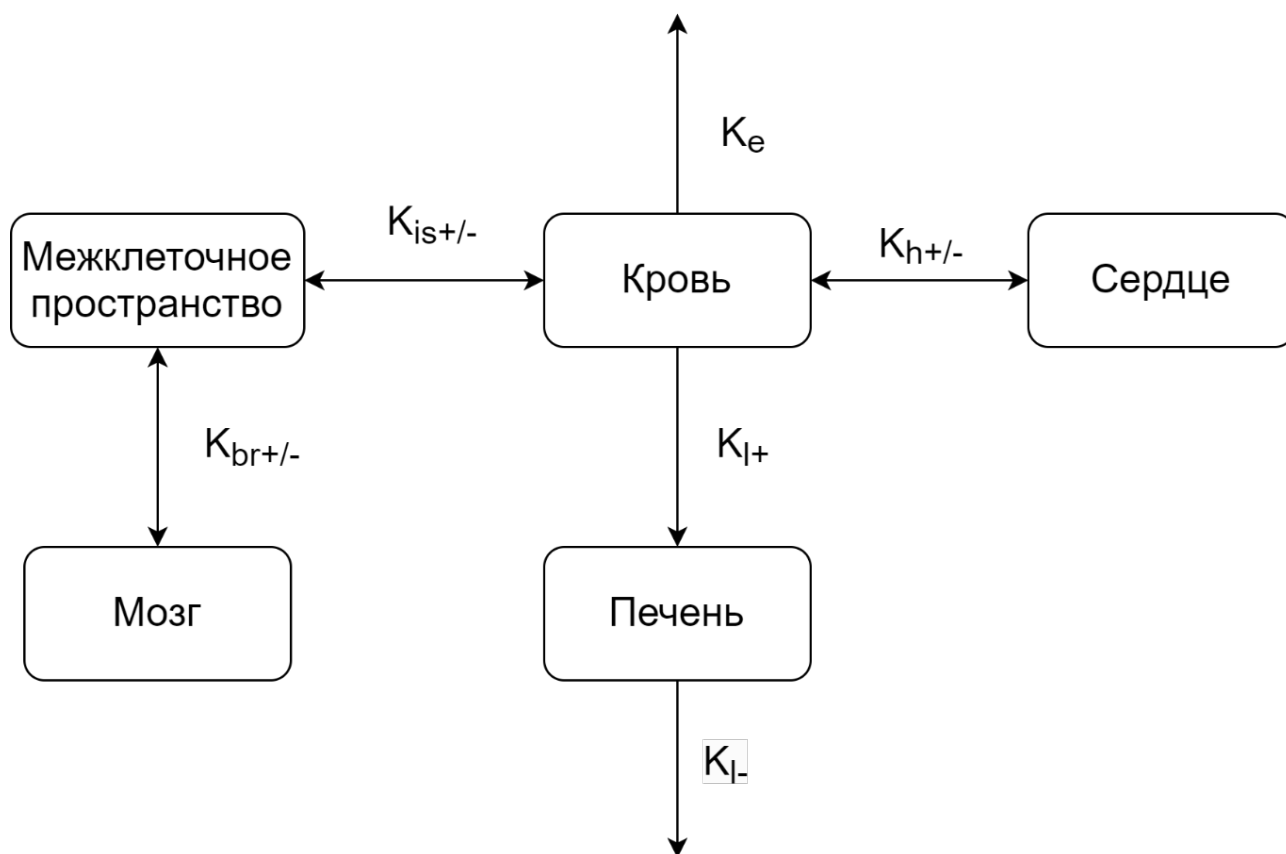


Рис. 2. Схематическое отображение пятикамерной модели

возможностей библиотеки, можно учесть его, тогда модель может быть записана в виде системы ОДУ (4):

$$\begin{aligned}
 V_b \frac{dC_b}{dt} &= -k_e C_b V_b - k_{l+} C_b V_b - k_{h+} C_b V_b - k_{is+} C_b V_b + k_{h-} C_h V_h + k_{is-} C_{is} V_{is} \\
 V_l \frac{dC_l}{dt} &= k_{l+} C_b V_b - k_{l-} C_l V_l \\
 V_h \frac{dC_h}{dt} &= k_{h+} C_b V_b - k_{h-} C_h V_h \\
 V_{is} \frac{dC_{is}}{dt} &= k_{is+} C_b V_b - k_{br+} C_{is} V_{is} - k_{is-} C_{is} V_{is} + k_{br-} C_{is} V_{is} \\
 V_{br} \frac{dC_{br}}{dt} &= k_{br+} C_{is} V_{is} - k_{br-} C_{br} V_{br}
 \end{aligned} \tag{4}$$

где C_b , C_l , C_h , C_{is} , C_{br} отражают концентрации лекарственного вещества в крови, печени, сердце, межклеточном пространстве мозга и клетках мозга соответственно.

В целом, механизм использования библиотеки PyPharm может быть описан в виде схемы (3):

Работа начинается с получения экспериментальных данных по ФК вещества в организме человека. Далее выбирается камерная модель, которая по мнению исследователя наилучшим образом будет описывать экспериментальные данные. Затем неизвестные параметры модели подбираются исходя из минимизации отклонения рассчитанных на модели данных от экспериментальных значений. Результатом работы библиотеки является массив данных, описывающий изменение концентрации лекарственного вещества в той или иной камере во времени. Данные для удобства могут быть представлены в виде графиков.

Если поэтапно рассматривать описанный выше процесс для модели $[^{25}\text{Mg}^{2+}]$ порфириновых наночастиц, работа с библиотекой PyPharm будет выглядеть следующим образом (рис. 4–5):

- 1) так как входные и выходные концентрации в источнике данных выражены в разных единицах

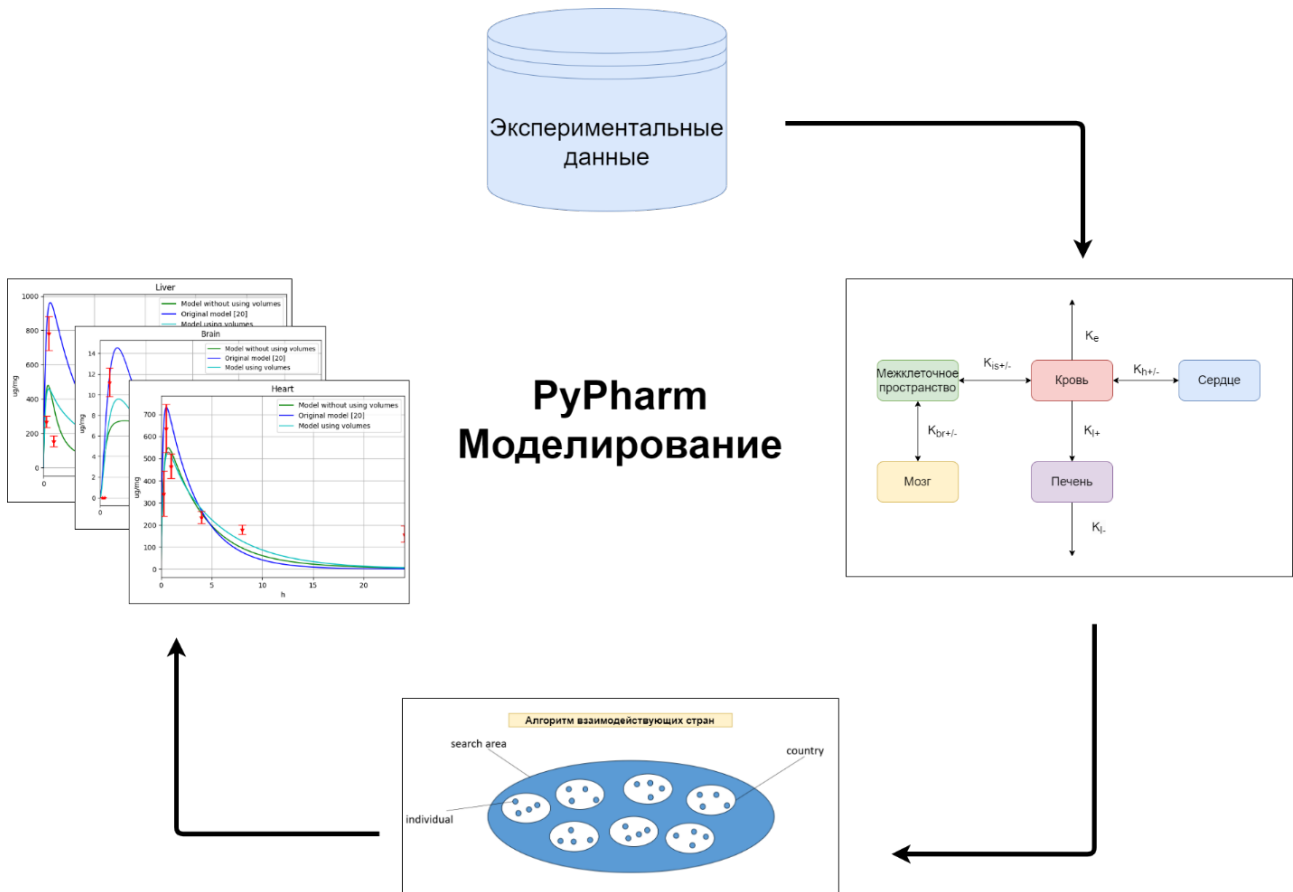


Рис. 3. Схематическое описание работы библиотеки PyPharm

измерения (таких, что нельзя явно преобразовать одни в другие), необходимо создать объект класс *MagicCompartmentModel*;

2) на этапе инициализации объекта необходимо задать неизвестные объемы и коэффициенты как None для их дальнейшего поиска. Кроме того, нужно задать параметр *exclude_compartments*, исключив в данном случае камеру, отвечающую за кровь, так как концентрация лекарственного вещества в ней выражена в других единицах измерения. При этом переменная *matrix* отвечает за конфигурацию модели, а *outputs* за элиминацию, в данном случае она будет происходить только через печень;

3) далее необходимо произвести загрузку экспериментальных данных, используя *load_optimization_data* и запустить оптимизацию функцией *optimize*. Параметры M , N , n , p , m , k , l и ep являются настроечными параметрами для алгоритма взаимодействующих стран [21];

4) в завершении можно использовать *plot_model* для построения графиков высвобождения ЛВ в отдельных камерах, либо же использовать полученные данные для дальнейшего форматирования и вывода.

Пример результатов, полученных для данной модели с использованием библиотеки PyPharm приведен на рис. 6. Стоит заметить, что по результатам моделирования можно заключить, что модели, настроенные при помощи библиотеки PyPharm существенно превосходят авторскую [23] модель (значение целевой функции уменьшилось примерно на 67%).

Таким образом, мы сообщаем о разработке библиотеки *PyPharm*, способной решать задачи построения фармакокинетических моделей. На данный момент библиотека имеет версию 1.3.6 и доступна в репозитории PyPi <https://pypi.org/project/pypharm/>. Там же имеются примеры использования библиотеки и описание ее возможностей.

Библиотека способна строить три вида камерных моделей и уже была опробована и хорошо показала себя в различных исследованиях [14, 19]. В дальнейшем планируется расширение PyPharm, с добавлением в нее методов машинного обучения и расширением списка доступных моделей, например, РВРК-моделей.

```

from PyPharm import MagicCompartmentModel

# Задание матрицы, определяющей конфигурацию модели
matrix = [[0, None, 0, 0, 0],
          [None, 0, None, 0, 0],
          [0, None, 0, None, None],
          [0, 0, 0, 0, 0],
          [0, 0, None, 0, 0]]

# Задание вектора, определяющего конфигурацию выводов
outputs = [0, 0, 0.077, None, 0]

model = MagicCompartmentModel( #Создание объекта модели
    matrix, outputs,
    volumes=[None, None, None, None, None], #Неизвестные параметры, везде
    magic_coefficient=None, #задаются None
    exclude_compartments=[2],
    numba_option=True
)
model.load_optimization_data( #загрузка данных для оптимизации
    theoretic_x=[0.25, 0.5, 1, 4, 8, 24],
    theoretic_y=[[0, 0, 11.2, 5.3, 5.42, 3.2],
                [268.5, 783.3, 154.6, 224.2, 92.6, 0],
                [342, 637, 466, 235, 179, 158]],
    know_compartments=[0, 3, 4],
    d=20 * 0.2 * 1000,
    compartment_number=2
)

```

Рис. 4. Использование библиотеки *PyPharm* для построения конфигурации модели и загрузки экспериментальных данных

```

model.optimize( #Оптимизация параметров методом взаимодействующих стран
    method='country_optimization',
    Xmin=[0.5, 0.001, 0.001, 0.00001, 0.01, 0.01, 1, 20, 20, 20, 20, 20, 1],
    Xmax=[5, 2, 2.5, 0.3, 1, 1, 10, 100, 100, 100, 100, 100, 10],
    M=10,
    N=25,
    n=[1, 10],
    p=[0.00001, 2],
    m=[1, 8],
    k=8,
    l=3,
    ep=[0.2, 0.4],
    tmax=300,
    max_step=0.1,
    printing=True,
)

model.plot_model( #Построение кривых распределения
    d=20 * 0.2 * 1000,
    compartment_number=2,
    t_max=24,
    compartment_numbers=[3, 4, 0],
    compartment_names={3: "Печень", 4: "Сердце", 0: "Мозг"}
)

```

Рис. 5. Использование библиотеки *PyPharm* для оптимизации параметров модели и построения кривых распределения

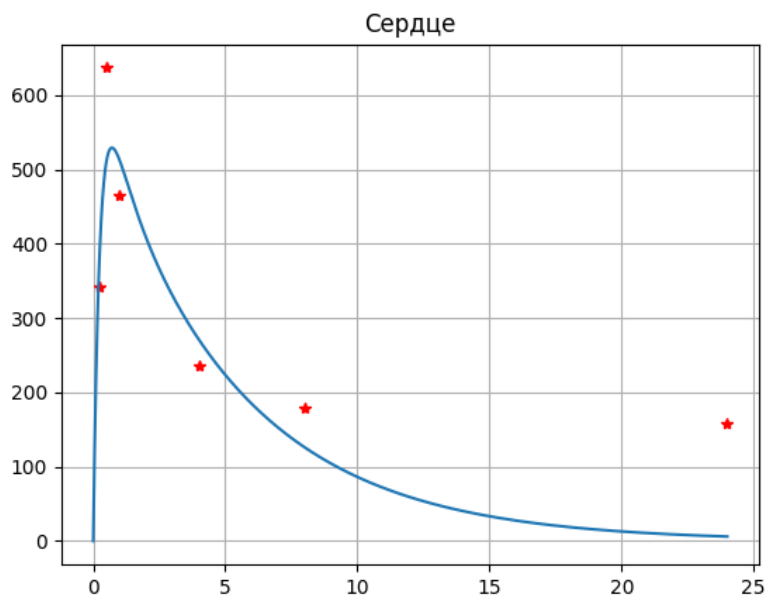


Рис. 6. Пример результатов работы

ЛИТЕРАТУРА

1. *Guideline on the Qualification and Reporting of Physiologically Based Pharmacokinetic (PBPK) Modelling and Simulation*. Режим доступа: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-reporting-physiologically-based-pharmacokinetic-pbpbk-modelling-and-simulation_en.pdf.
2. *Physiologically Based Pharmacokinetic Analyses – Format and Content Guidance for Industry*. Режим доступа: <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/physiologically-based-pharmacokinetic-analyses-format-and-content-guidance-industry>.
3. Ivy S. P., Siu L. L., Garrett-Mayer E., Rubinstein L. Approaches to Phase 1 Clinical Trial Design Focused on Safety, Efficiency, and Selected Patient Populations: a Report from the Clinical Trial Design Task Force of the National Cancer Institute Investigational Drug Steering Committee. *Clin Cancer Res.* 2010;16(6):1726–1736. DOI: 10.1158/1078-0432.CCR-09-1961.
4. Hill-McManus D., Marshall S., Soto E., Hughes D. A. Integration of Pharmacometrics and Pharmacoeconomics to Quantify the Value of Improved Forgiveness to Nonadherence: A Case Study of Novel Xanthine Oxidase Inhibitors for Gout. *Clin Pharmacol Ther.* 2019;106(3):652-660. DOI: 10.1002/cpt.1454.
5. Hill-McManus D., Marshall S., Soto E., Lane S., Hughes D. Impact of Non-Adherence and Flare Resolution on the Cost-Effectiveness of Treatments for Gout: Application of a Linked Pharmacometric/Pharmacoeconomic Model. *Value Health.* 2018;21(12):1373–1381. DOI: 10.1016/j.jval.2018.06.002.
6. Singh N., Vayer P., Tanwar S., Poyet J.-L., Tsaioun K. and Villoutreix B. O. Drug Discovery and Development: Introduction to the General Public and Patient Groups. *Frontiers in Drug Discovery.* 2023; 3:1201419. DOI: 10.3389/fddsv.2023.1201419.
7. Mortensen S. B., Jónsdóttir A. H., Klim S. and Madsen H. *Introduction to PK/PD Modelling - with Focus on PK and Stochastic Differential Equations*. Lyngby: Technical University of Denmark, DTU Informatics; Building 321. Technical Report, No. 2008-16. Режим доступа: <https://accio.github.io/AMIDD/assets/2019/08/2008-Mortensen-IntroductionToPKPDModelling.pdf>.
8. Сергиенко В. И., Джеллифф Р., Бондарева И. Б., *Прикладная фармакокинетика: основные положения и клиническое применение*. М.: Изд-во РАМН; 2003. 208 с.
9. Bhavatharini P., Deepalakshmi M., Arun K. Pharmacometrics: The Science Applied from Bench to Bedside. *J Appl Pharm Sci;* 2022;12(1):55–64. DOI: 10.7324/JAPS.2021.120104.
10. Yoon E., Babar A., Choudhary M., Kutner M., Pysopoulos N. Acetaminophen-Induced Hepatotoxicity:

- a Comprehensive Update. *J Clin Transl Hepatol*. 2016;4(2):131-142. DOI: 10.14218/JCTH.2015.00052.
11. Ryu H.-J., Kang W.-H., Kim T., Kim J. K., Shin K.-H., Chae J.-W. and Yun H.-Y. A Compatibility Evaluation between the Physiologically Based Pharmacokinetic (PBPK) Model and the Compartmental PK Model Using the Lumping Method with Real Cases. *Front. Pharmacol*. 2022;13:1–12. DOI: 10.3389/fphar.2022.964049.
 12. Talevi A., Bellera C. L. One-Compartment Pharmacokinetic Model. *The ADME Encyclopedia*. Springer, Cham. 2021:1–8. DOI: 10.1007/978-3-030-51519-5_58-1.
 13. Talevi A., Bellera C. L. Compartmental Pharmacokinetic Models. *ADME Processes in Pharmaceutical Sciences*. Springer, Cham. 2024:173–192. DOI: https://doi.org/10.1007/978-3-031-50419-8_8.
 14. Крашенинников Р. С., Митричев И. И. Применение алгоритма взаимодействующих стран для поиска констант фармакокинетической модели. *Математические методы в технологиях и технике*. 2023;4:12–15. DOI: 10.52348/2712-8873_ММТТ_2023_2_12.
 15. Самородов П. О. Особенности построения математической модели фармакокинетики при многократном введении препарата. *Теоретико-методологические аспекты преподавания математики в современных условиях: материалы III Международной научно-практической конференции (1–7 июня 2020 г., г. Луганск)*. Луганск: Книта; 2020:78–83.
 16. Jamei M., Dickinson G. L., Rostami-Hodjegan A. A Framework for Assessing Inter-Individual Variability in Pharmacokinetics Using Virtual Human Populations and Integrating General Knowledge of Physical Chemistry, Biology, Anatomy, Physiology and Genetics: a Tale of ‘bottom-up’ vs ‘top-down’ Recognition of Covariates. *Drug Metabolism and Pharmacokinetics*. 2009;24(1):53–75. DOI: 10.2133/dmpk.24.53.
 17. Rowland M. Physiologically-Based Pharmacokinetic (PBPK) Modeling and Simulations Principles, Methods, and Applications in the Pharmaceutical Industry. *CPT Pharmacometrics Syst Pharmacol*. 2013;2(7):e55. DOI: 10.1038/psp.2013.29.
 18. Rostami-Hodjegan A., Bois F. Y. Opening a Debate on Open-Source Modeling Tools: Pouring Fuel on Fire Versus Extinguishing the Flare of a Healthy Debate. *CPT Pharmacometrics Syst Pharmacol*. 2021;10(5):420–427. DOI: 10.1002/psp4.12615.
 19. Крашенинников Р. С., Митричев И. И. Разработка библиотеки Python для решения задач фармакокинетики. *Свободное программное обеспечение в высшей школе: материалы девятнадцатой конференции, Переславль-Залесский, 28–30 июня 2024 года*. Москва: ООО «МАКС Пресс»; 2024:166–170.
 20. *Документация метода solve_ivp*. Режим доступа: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html.
 21. Крашенинников Р. С., Митричев И. И. Оптимизационный алгоритм взаимодействующих стран. *Информационные технологии в моделировании и управлении: подходы, методы, решения: материалы VI Всероссийской научной конференции с международным участием, Тольятти, 18–20 апреля 2023 года*. Тольятти: Тольяттинский государственный университет; 2023:184–191.
 22. Крашенинников Р. С., Митричев И. И. Оптимизационный алгоритм взаимодействующих стран и его применение для решения задач фармакокинетики. *Прикладная математика и вопросы управления*. 2023;4:26–135. DOI: 10.15593/2499-9873/2023.4.08.
 23. Фурсов В. В., Зинченко Д. И., Наместникова Д. Д., Кузнецов Д. А. In silico-моделирование в оптимизации алгоритмов фармакокинетических исследований [$^{25}\text{Mg}^{2+}$] порфирина-фуллереновых наночастиц. *Вестник Российского государственного медицинского университета*. 2022;4:65–71. DOI: 10.24075/vrgmu.2022.037.