

MetidaNCA - Руководство по эксплуатации

Владимир Арнаутов

2025-05-28

Содержание

1 Введение	2
1.1 MetidaNCA	2
1.2 Julia	3
2 Требования	4
2.1 Цикл разработки MetidaNCA	4
2.2 Версии	4
2.3 Поддержка	4
3 Установка	6
3.1 Установка Julia	6
3.2 Установка MetidaNCA	6
3.3 Пользовательская валидация	6
3.4 Покрытие	6
3.5 Данные для тестирования	6
4 Описание фармакокинетических параметров	6
5 Работа с пакетом MetidaNCA	8
5.1 Импорт данных	8
5.2 Выполнение вычислений	9
5.3 Построение графиков	10
5.4 Экспорт результатов	10
6 Дополнительные функции	13
6.1 LimitRule	13
6.2 DoseTime	13
6.3 ElimRange	13
7 API	14
8 Вычисление основных параметров	14
8.1 C _{max}	14
8.1.1 T _{max}	14
8.1.2 AUC / AUMC	14
8.2 Пропущенные значения и интерполяция	14

8.2.1	Правило линейной интерполяции	14
8.2.2	Правило лог-интерполяции	15
8.2.3	MRTlast	15
8.2.4	Kel	15
8.2.5	HL	15
8.2.6	AUCinf	15
8.2.7	AUMCinf	15
8.2.8	AUCpct	15
9	Валидация	16
10	Глоссарий	16
11	Ссылки	16
12	Обратная связь	16

1 Введение

1.1 MetidaNCA

MetidaNCA позволяет проводить пакетное вычисление фармакокинетических (ФК) параметров с использованием некомпартментных методов (NCA – Non-Compartmental Analysis). Данный метод обработки данных выполняется в ходе анализа результатов клинических исследований лекарственных средств. MetidaNCA обеспечивает:

- Импорт данных (фармакокинетический профилей);
- Вычисление фармакокинетических параметров: площадь под кривой концентрация-время (AUC), максимальная концентрация (C_{max}), время достижения максимальной концентрации (T_{max}), константа элиминации (Kel), период полувыведения (HL), среднее время удержания (MRT) и др.;
- Вычисление фармакокинетических параметров для исследования концентраций в моче: количество выведенного препарата (AR), площадь под кривой концентрация-время (AUC), максимальная концентрация (C_{max}), время достижения максимальной концентрации (T_{max});
- Вычисление фармакодинамических параметров;
- Вывод графиков;
- Вывод данных в табличном виде.

Пакет MetidaNCA может применяться на всех системах, которые поддерживаются языком Julia: x86-64, ARMv8, i686.

Язык программирования: Julia.

Автоматически обновляемая документация доступна по ссылкам:

- Версия в разработке: <https://pharmcat.github.io/MetidaNCA.jl/dev/>
- Стабильный релиз: <https://pharmcat.github.io/MetidaNCA.jl/stable/>

Сборка руководства по эксплуатации выполнена в окружении:

- Julia version: v"1.11.5"
- Current machine: "x86_64-linux-gnu"
- MetidaNCA: ver

1.2 Julia

Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений, имеет встроенную поддержку многопоточности и распределённых вычислений, реализованные в том числе в стандартных конструкциях. Julia применяется как язык программирования и среда вычисления на различных платформах, таких как Российская платформа математических вычислений и динамического моделирования Engее и др.

Сайт: <https://julialang.org/>

2 Требования

- Установленная Julia 1.8 (и выше) для Операционной системы/Архитектуры в Тип 1 (Tier 1) списке (см. Supported platforms).

Tier 1: Julia is guaranteed to build from source and pass all tests on these platforms when built with the default options. Official binaries are always available and CI is run on every commit to ensure support is actively maintained.

Для использования MetidaNCA могут потребоваться пакеты для работы с табличными данными (DataFrames), CSV таблицами (CSV) и графиками (Plots). При этом пакеты DataFrames и CSV не являются зависимостями для MetidaNCA. Предполагается, что MetidaNCA работоспособна с пакетами, которые поддерживают интерфейс Tables (для таблиц) и RecipeBase (для графиков).

2.1 Цикл разработки MetidaNCA

Обычный выпуск обновлений выполняется следующим образом:

- Разработка программы (дополнений)
- Разработка тестировочных процедур
- Проверка на локальной машине
- Релиз ветви разработчика
- Pull request в основную ветвь
- Проведение тестировочных процедур при помощи GitHub Actions
- При успешном тестировании слияние с основной ветвью (при необходимости)
- Релиз новой версии (при необходимости)

2.2 Версии

- X.Y.Z - patch release (без критических изменений, нет значимых функциональных изменений если $X \neq 0$)
- X.Y.0 - minor release (возможны критические изменения если $X = 0$)
- X.0.0 - major release (критические изменения, возможны изменения API)
- 0.#.# - данные версии не гарантируют неизменность API
- 1.#.# or higher - стабильный публичный API (переход на следующую версию не гарантирует полное сохранение API, тем не менее стабильность API является приоритетом при разработке последующих релизов)

2.3 Поддержка

Тестировочные процедуры с использованием GitHub Actions выполняются для следующих комбинаций версий Julia/Операционной системы/Архитектуры:

- julia-version: 1.8, 1
- julia-arch: x64
- os: ubuntu-latest, macOS-latest, windows-latest

Где “1” - последняя стабильная версия Julia. Проведение тестирования процедур для версии “1.8” не исключает, что MetidaNCA не будет работоспособна на более ранних версиях. Однако, тестирование MetidaNCA для систем не включенных в список полностью в зоне ответственности пользователя.

3 Установка

3.1 Установка Julia

Установка Julia зависит от операционной системы пользователя согласно руководству: [Install julia](#)

3.2 Установка MetidaNCA

MetidaNCA устанавливается путем выполнения следующей команды в REPL:

```
import Pkg; Pkg.add("MetidaNCA")
```

3.3 Пользовательская валидация

MetidaNCA поставляется с набором валидационных (тестировочных) процедур, которые направлены на проверку работоспособности программы в конкретном окружении.

При проведении тестирования выполняются стандартные функции пакета MetidaNCA с последующим сравнением с заранее известными значениями (значения вычисленных параметров получены с использованием других программных продуктов предназначенных для вычисления фармакокинетических параметров: Phoenix WinNonlin и др.).

Для запуска валидационных процедур выполнить:

```
import Pkg; Pkg.test("MetidaNCA")
```

3.4 Покрытие

Покрытие кода тестами контролируется с использованием ресурса: [Codecov.io](#)

- Цель: покрытие $\geq 90.0\%$

3.5 Данные для тестирования

Поставляются с пакетом (`./test/csv/*.csv`).

4 Описание фармакокинетических параметров

Следующие фармакокинетические параметры могут быть получены в ходе анализа ФК профилей:

Название	Описание(EN)	Описание(RU)
Cmax	Maximum concentration	Максимальная концентрация
Tmax	Time at Cmax	Время достижения Cmax
Cdose	Concentration at dose time	Концентрация в момент дозирования
Clast	Last non-zero concentration	Последняя ненулевая концентрация
AUClast	AUC to Clast	AUC до Clast
AUMClast	AUMC to Clast	AUMC до Clast
AUCall	AUC with all values	AUC с включением всех концентраций
Rsq	r square	Коэффициент r квадрат
ARsq	Adjusted r square	Коэффициент r квадрат с поправкой
Kel	Terminal elimination constant	Терминальная константа элиминации
HL	Half live or T1/2	Период полувыведения
LZint	Intercept	Intercept (свободный член)
Clast_pred	Predicted Clast	Предсказанная Clast
AUCinf	AUC extrapolated to infinity	AUC экстраполированная до бесконечности
AUCpct	Percentage AUClast from AUCinf	Доля AUClast в AUCinf (%)
MRTlast	Mean Residence Time (last)	Среднее время удержания (Clast)
MRTinf	Mean Residence Time (inf)	Среднее время удержания (inf)
Clinf	Clearance	Клиренс (кажущийся)
Vzinf	Volume of distribution	Объем распределения (кажущийся)
AUCtau	AUC in Tau range	AUC в границах Tau
AUMCtau	AUMC in Tau range	AUMC в границах Tau
MRTtauinf	MRT based on Tau	MRT в границах Tau
Cltau	Clearance in Tau range	Клиренс в границах Tau
Vztau	Volume of distribution in Tau range	Объем распределения Tau

Таблица: Описание параметров

5 Работа с пакетом MetidaNCA

Стандартные сценарии использования пакета MetidaNCA обычно предполагают следующие этапы:

- Импорт данных
- Модификация импортированных данных (при необходимости)
- Получение параметров
- Построение графиков
- Экспорт результатов

Пример загрузки табличных данных:

```
pkdata2 = CSV.File(joinpath(dirname(pathof(MetidaNCA)), "..", "test", "cs
```

Подробная информация о работе с табличными данными при помощи пакета DataFrames: [Codecov.io](https://codecov.io)

Пример табличных данных:

```
pkdata2[1:15, :]
```

Error: UndefVarError: `pkdata2` not defined in `Main.var`##WeaveSandBox#230"
Suggestion: check for spelling errors or missing imports.

5.1 Импорт данных

Для пакетного импорта данных таблица должна содержать следующие поля:

- Время (Float64) - уникальное в пределах SubjectID;
- Концентрация (Float64);
- Уникальный идентификатор субъекта (SubjectID) (если субъектов > 1) - одно или несколько полей.

```
# Загрузка данных
pkdata2 = CSV.File(
  joinpath(dirname(pathof(MetidaNCA)), "..", "test", "csv", "pkdata2.csv")
) |> DataFrame
```

```
# Импорт данных
ds = pkimport(pkdata2, :Time, :Concentration, [:Subject, :Formulation];
dosetime = DoseTime(dose = 100, time = 0))
```

```
DataSet: Pharmacokinetics subject
Length: 10
Subject 1: Subject => 2, Formulation => R,
Subject 2: Subject => 10, Formulation => R,
Subject 3: Subject => 9, Formulation => T,
Subject 4: Subject => 8, Formulation => R,
Subject 5: Subject => 3, Formulation => R,
Subject 6: Subject => 1, Formulation => T,
Subject 7: Subject => 5, Formulation => T,
```


Subject 8: Subject => 4, Formulation => R,
 Subject 9: Subject => 6, Formulation => T,
 Subject 10: Subject => 7, Formulation => R,

5.2 Выполнение вычислений

Для одного субъекта:

```
dsnca1 = nca!(ds[1], adm = :ev, calcm = :lint, verbose = 1)
```

Non-compartmental Pharmacokinetic Analysis

Subject: Subject => 2; Formulation => R;

Settings:

Method: lint; Dose: 100; Dose time: 0

Time	Conc.	AUC	AUC (cum.)	AUMC	AUMC (cum.)	Info
0	0	0	0	0	0	D
0.5	62.22	15.56	15.56	7.778	7.778	
1	261.2	80.85	96.41	73.07	80.85	
1.5	234.1	123.8	220.2	153.1	233.9	
2	234.1	117	337.3	204.8	438.7	
2.5	222.9	114.2	451.5	256.3	695.1	
3	213.9	109.2	560.7	299.7	994.8	
4	196	205	765.7	712.9	1708	
5	199.6	197.8	963.5	891.1	2599	
6	196	197.8	1161	1087	3686	
8	213.4	409.4	1571	2883	6569	
10	200.1	413.4	1984	3708	1.028e+04	E
12	196	396.1	2380	4353	1.463e+04	E
24	160.3	2138	4519	3.72e+04	5.183e+04	E
48	110.3	3247	7766	1.097e+05	1.615e+05	E
72	85.24	2346	1.011e+04	1.372e+05	2.987e+05	E

Cdose: 0.0, Dose time: 0

Kel start: 10.0; end: 72.0

PK/PD subject NCA result

Parameter	Value
Rsqn	3.0
Vzlast	0.701038
Tmax	1.0
ARsq	0.990351
AUClast	10112.2
MRTinf	71.9379
AUCinf_pred	15963.2

```

      LZ      -0.0141063
    LZint      5.4289
  Obsnum      16.0
    Clast      85.241
    Dose      100.0
    Tlag       0.0
    Cdose       0.0
  Vssinf      0.4453
  AUCall      10112.2
  []          []

```

```

-----
              15 rows omitted

```

Для всего набора данных:

```
dsnca = nca!(ds, adm = :ev, calcm = :lint)
```

```
dsnca[3:4]
```

DataSet: PK/PD NCA result

Length: 2

Subject 1: Subject => 9, Formulation => T,

Subject 2: Subject => 8, Formulation => R,

5.3 Построение графиков

Построение графиков выполняется с использованием пакета RecipesBase. Это значит, что для построения могут быть использованы пакеты совместимые с RecipesBase.jl - Plots.jl и .

Для субъекта:

```
pkplot(ds[1])
```

Для набора данных:

```
pkplot(ds; typesort = :Subject, pagesort = MetidaNCA.NoPageSort(), sort =
```

5.4 Экспорт результатов

Для экспорта таблиц может быть использован пакет DataFrames.jl:

```
dsnca = DataFrame(dsnca)
```

```
dsnca[1:5, 1:7]
```

5×7 DataFrame

Row	Subject	Formulation	Tmax	ARsq	AUClast	LZint	Obsnum
	Int64	String1	Float64	Float64	Float64	Float64	Float64
1	2	R	1.0	0.990351	10112.2	5.4289	16.0
2	10	R	2.0	0.863912	5620.89	4.89636	16.0
3	9	T	3.0	0.921307	8315.08	4.94726	16.0

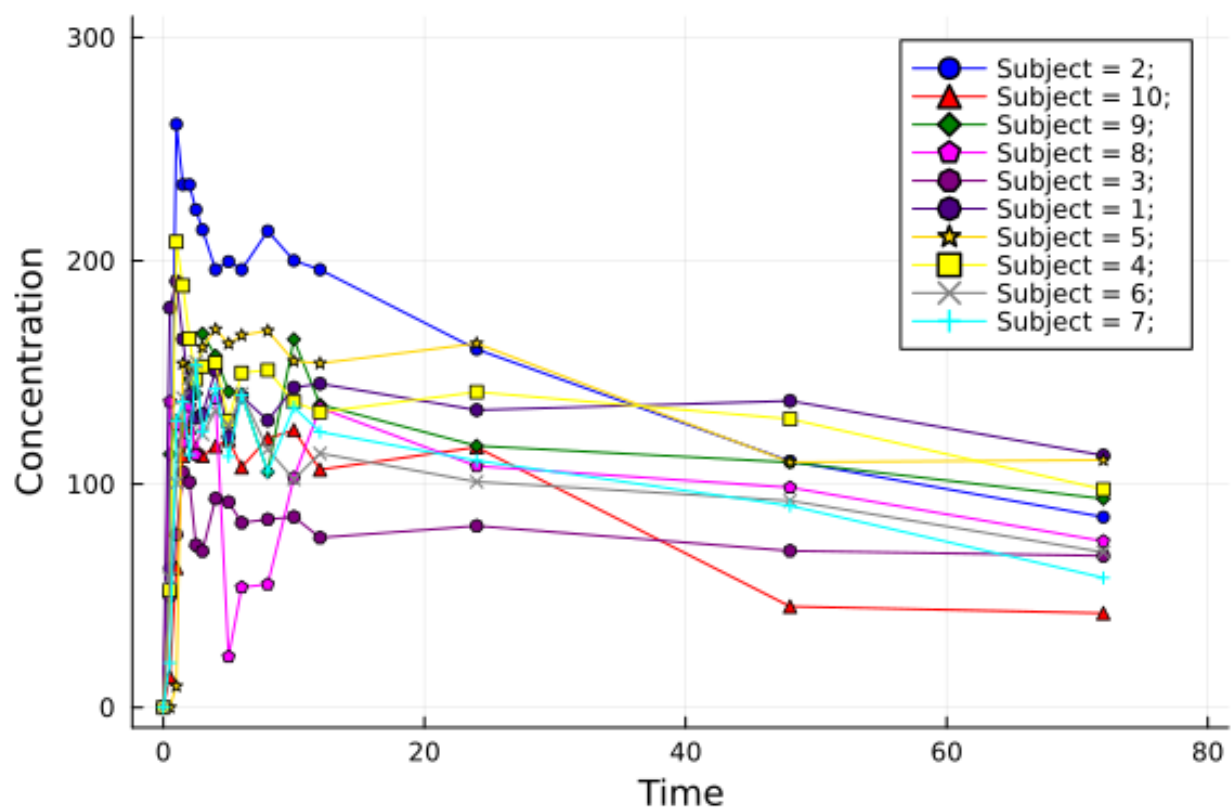


Рис. 1: A plot

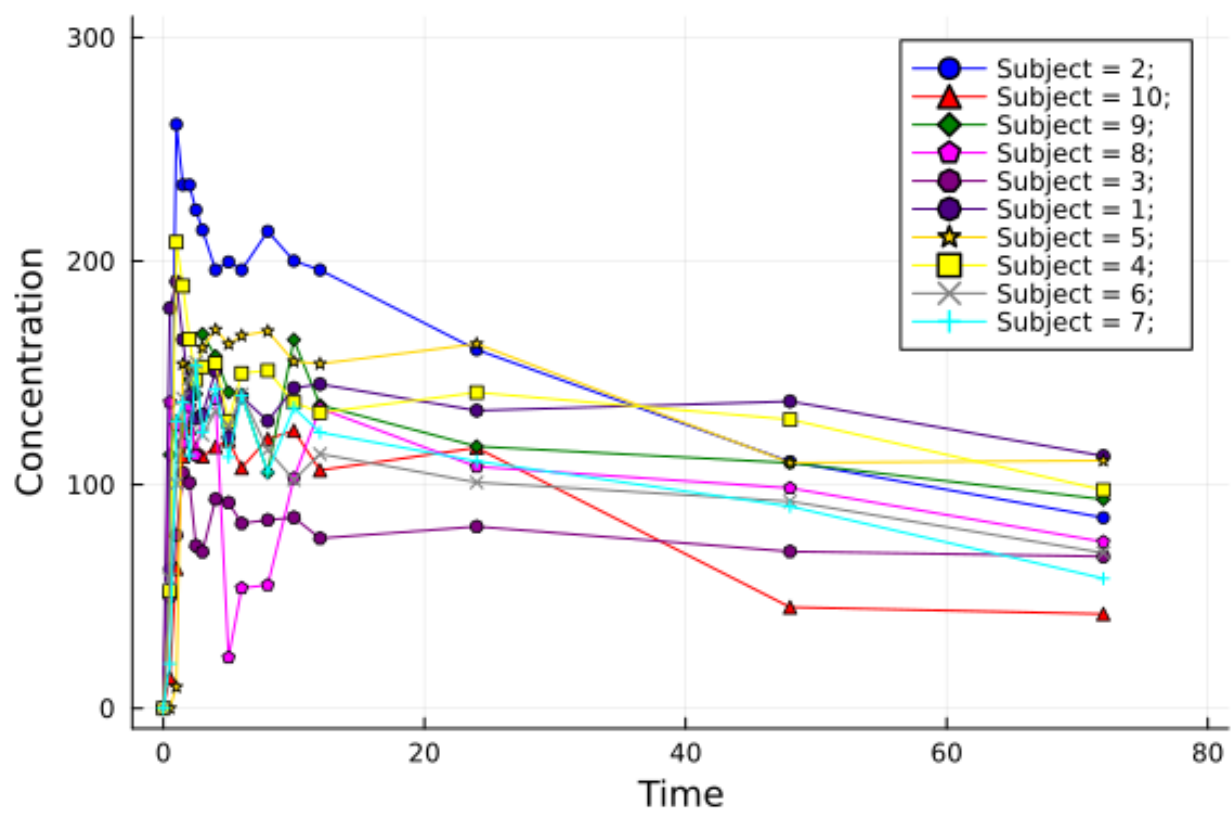


Рис. 2: B plot

4		8	R	4.0	0.921954	7110.67	4.96848	16.0
5		3	R	1.5	0.776307	5396.55	4.44065	16.0

С последующим сохранением в виде CSV или XLSX файла (CSV.jl, XLSX.jl)

```
CSV.write("nca_export.csv", dsnca)
```

6 Дополнительные функции

6.1 LimitRule

Правила для замены наблюдений ниже LLOQ и неопределенных значений (NaN).

```
lr = LimitRule(;lloq = 0.5, btmx = 0.0, atmax = NaN, nan = NaN, rm = true)
applylimitrule!(ds, lr)
```

6.2 DoseTime

Установка дозы и времени дозирования.

```
dt = DoseTime(dose = 110, time = 2.1, tau = 10)
setdosetime!(ds[1], dt)
```

6.3 ElimRange

Установка границ для расчета константы элиминации:

```
kr = ElimRange(kelstart = 4, kelend = 12)
setkelrange!(ds, kr; kelauto = true)
```

7 API

Подробное описание API составляется автоматически при релизе новой версии и доступно по адресу: API.

8 Вычисление основных параметров

8.1 C_{max}

Максимальная концентрация измеренная после дозирования, используется первое значение.

8.1.1 T_{max}

Время достижения максимальной концентрации, используется первое значение.

8.1.2 AUC / AUMC

Area under Curve / Area under the Moment Curve:

$$AUC = \sum_{n=1}^N AUC_{\{n\}}$$

$$AUMC = \sum_{n=1}^N AUMC_{\{n\}}$$

$$AUC_{\text{mid}_{\{t_1\}}^{\{t_2\}}} = \Delta t \times \frac{C_1 + C_2}{2}$$

$$AUMC_{\text{mid}_{\{t_1\}}^{\{t_2\}}} = \Delta t \times \frac{t_1 \times C_1 + t_2 \times C_2}{2}$$

8.1.2.1 Правило линейных трапеций

$$AUC_{\text{mid}_{\{t_1\}}^{\{t_2\}}} = \Delta t \times \frac{C_2 - C_1}{\ln(C_2/C_1)}$$

$$AUMC_{\text{mid}_{\{t_1\}}^{\{t_2\}}} = \Delta t \times \frac{t_2 \times C_2 - t_1 \times C_1}{\ln(C_2/C_1)}$$

8.1.2.2 Правило лог-трапеций

8.2 Пропущенные значения и интерполяция

8.2.1 Правило линейной интерполяции

$$C_x = C_1 + \frac{(t_x - t_1) \times (C_2 - C_1)}{t_2 - t_1}$$

8.2.2 Правило лог-интерполяции

$$C_x = \exp\left(\ln(C_1) + \frac{(t_x - t_1) \times (\ln(C_2) - \ln(C_1))}{t_2 - t_1}\right)$$

8.2.3 MRTlast

Среднее время удержания:

$$MRT_{\text{last}} = AUMC_{\text{last}} / AUC_{\text{last}}$$

8.2.4 Kel

λ_z - константа элиминации. Для расчета используется линейная регрессия для логарифмированных данных на терминальном участке.

8.2.5 HL

Время полувыведения; $T_{1/2}$

$$HL = \ln(2) / \lambda_z$$

8.2.6 AUCinf

$$AUC_{\infty} = AUC_{\text{last}} + \frac{C_{\text{last}}}{\lambda_z}$$

8.2.7 AUMCinf

$$AUMC_{\infty} = AUMC_{\text{last}} + \frac{t_{\text{last}} \times C_{\text{last}}}{\lambda_z} + \frac{C_{\text{last}}}{\lambda_z^2}$$

8.2.8 AUCpct

$$AUC_{\text{pct}} = (AUC_{\infty} - AUC_{\text{last}}) / AUC_{\infty} \times 100 \%$$

9 Валидация

Информация о валидации программного обеспечения представлена в отчете: MetidaNCA Validation Report.

10 Глоссарий

- Repository - GitHub repository: <https://github.com/PharmCat/MetidaNCA.jl>
- Master branch - main branch on GitHub (link).
- Current machine - pc that used for validation report generating.
- AUC - площадь под фармакокинетической кривой — кривой «концентрация—время».
- AUMC - площадь под кривой «произведение времени на концентрацию препарата tC».
- HL - период полувыведения.
- LLOQ - Нижний предел количественного определения (Lower Limit of Quantification).
- REPL (от англ. read-eval-print loop — «цикл “чтение — вычисление — вывод”») — форма организации простой интерактивной среды программирования в рамках средств интерфейса командной строки.
- Тау - интервал дозирования (время).
- тестовое покрытие (test coverage): Степень, с которой данный тест проверяет требования для программного продукта

11 Ссылки

- Gabrielsson J, Weiner D. Non-compartmental analysis. Methods Mol Biol. 2012;929:377-89. doi: 10.1007/978-1-62703-050-2_16. PMID: 23007438.
- ГОСТ Р ИСО/МЭК 12207-2010
- MetidaNCA Documentation EN
- MetidaNCA Validation Report

12 Обратная связь

Сообщения об ошибках, а также сообщения для улучшения программного обеспечения может быть направлена:

- по адресу mail@pharmcat.net
- В разделе Issues на сайте github.com