ИНСТИТУТ
БИОИНФОРМАТИКИ

УНИВЕРСИТЕТ ИТМО

# Автоматизация процесса вывода совместной демографической истории нескольких популяций из сайт-частотного спектра

Носкова Екатерина

Руководители:
Владимир Ульянцев, Павел Добрынин

# Демографические истории

**Хотим знать как развивались популяции.**

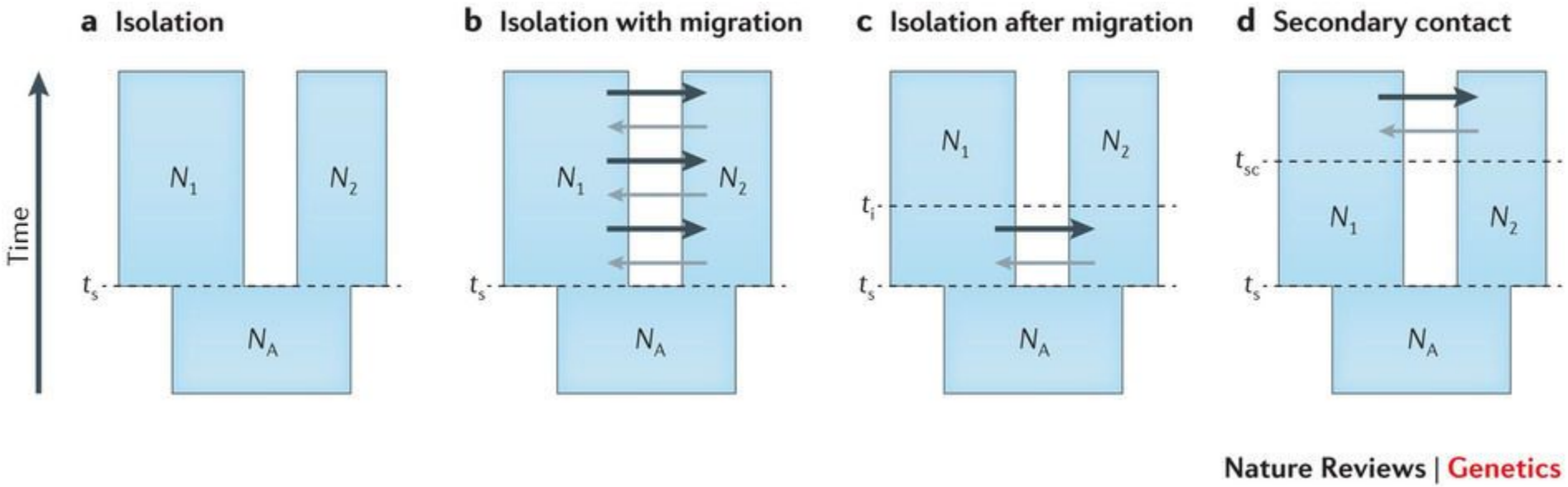
Можно исследовать структуру популяций: сильно ли различаются рассматриваемые популяции, сколько их, были ли admixture events.

Можно искать участки, которые пришли от общих предков.

А можно строить более “сложные” демографические модели: сколько лет назад был какой размер общей популяции, сколько лет назад они разделились, были ли миграции между получившимися популяциями и тд.

Для построения более “сложных” демографических моделей нет “черного ящика”.

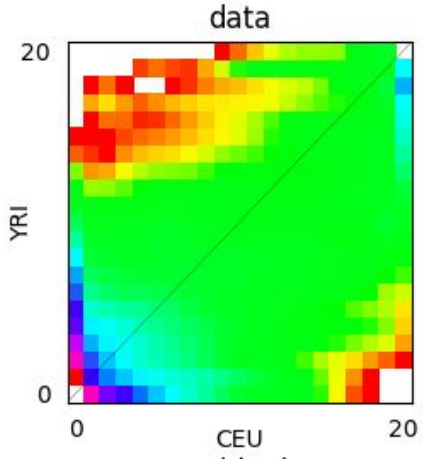
# Простейшие “сложные” демографические модели




Параметры моделей:

- Время (T)
- Размер популяций (N)
- Миграции (m)
- Отбор (h) *не рассматривался*

# Данные для анализа




1) Геномы
2) Снипы
3) Гаплотипы
4) Allele Friequency Spectrum (AFS, Site FS, SFS, FS):

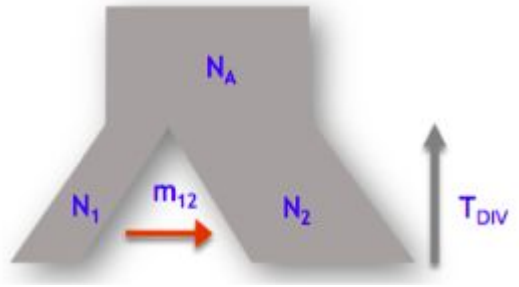
Матрица, которая строится по снипам:

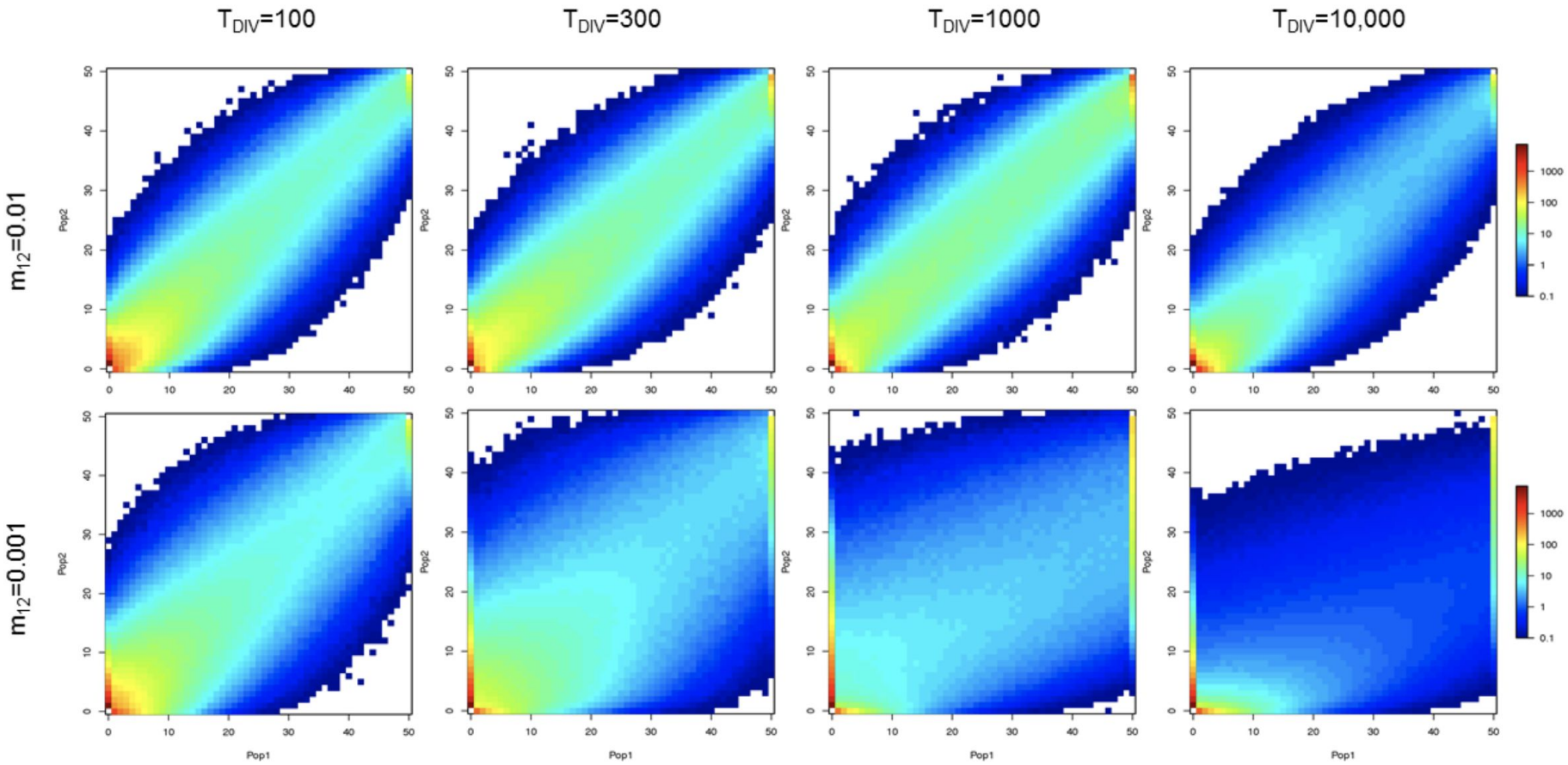
**У нас есть P популяций, n1 хромосом 1 популяции, n2 хромосомы 2 популяции и т.д.**

**AFS - P-мерная матрица такая, что каждая клетка под индексом d1, d2,.., dP содержит количество снипов (аллелей), которые появились d1 раз в 1 популяции, d2 раз во 2, … dP раз в P популяции.**

*-Теряется информация о связанности и расположении.*

# Зависимость от параметров



# ∂a∂i

Пакет питона, который использует AFS для выявления демографической истории.

Пользователь дает AFS и также задает **демографическую  модель(!)**

Эта модель написана **на питоне**.

Пакет выясняет насколько модель “подходит” для данного AFS:

Возвращает log likelihood - логарифм вероятности того, что модель соответствует данному спектру.

# Пример кода модели

```
def IM(params, ns):
    s, nu1, nu2, T, m12, m21 = params

    sts = moments.LinearSystem_1D.steady_state_1D(ns[0] + ns[1])
    fs = moments.Spectrum(sts)
    fs = moments.Manips.split_1D_to_2D(fs, ns[0], ns[1])

    nu1_func = lambda t: s * (nu1/s)**(t/T)
    nu2_func = lambda t: (1-s) * (nu2/(1-s))**(t/T)
    nu_func = lambda t: [nu1_func(t), nu2_func(t)]

    fs.integrate(nu_func, T, dt_fac=0.01,
                 m=numpy.array([[0, m12], [m21, 0]]))

    return fs
```

Listing 5: **Two-population isolation-with-migration:** The ancestral population splits into two, with a fraction `s` going into pop 1 and fraction `1-s` into pop 2. The populations then grow exponentially, with asymmetric migration allowed between them.

# Постановка задачи

Задача: автоматический подбор демографической истории нескольких популяций.

Мотивация:

- Биологи не хотят писать код на питоне.
- Никто не хочет вручную подбирать модели. Это неудобно и долго.

Решение:

Напишем тул, который будет делать все автоматически: рассматривать различные структуры моделей, варьировать их параметры и выбирать оптимальную модель.

# ∂a∂i Что внутри?

Диффур:

$$\frac{\partial}{\partial \tau}\phi = \frac{1}{2}\sum_{i=1,2,\ldots,P}\frac{\partial^2}{\partial^2 x_i}\frac{x_i(1-x_i)}{\nu_i}\phi$$

$$-\sum_{i=1,2,\ldots,P}\frac{\partial}{\partial x_i}\left(\gamma_i x_i(1-x_i)+\sum_{j=1,2,\ldots,P}M_{i\leftarrow j}(x_j-x_i)\right)\phi.$$

где кси - эволюция (плотность накопленных мутаций),

x_i - вероятность мутации в популяции i.

Теперь по нашим x_i мы можем построить AFS модели как матожидание числа снипов с определенными частотами в популяциях

И затем посчитать likelihood двух матриц.

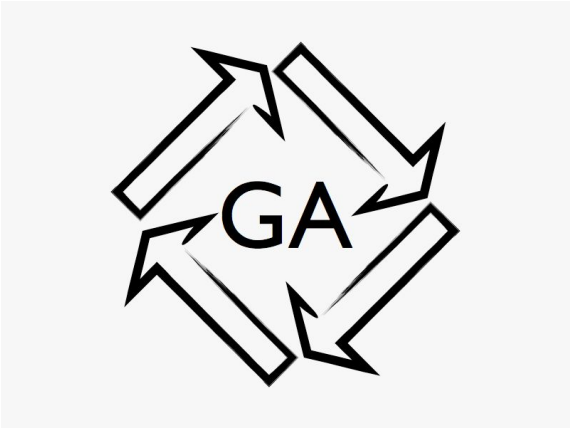
# Внутренняя оптимизация *д***а***д***i**

Можно попросить dadi оптимизировать параметры модели.

+ У пакета есть и своя оптимизация: различные варианты BFGS.
+ Есть выбор оптимизаций.
- **Структура модели фиксирована.**
- Оптимизация долгая.
- **Приходит к нужной модели только из близкой по структуре.**
- Нужно давать нижнюю и верхнюю границу всех параметров.
- O(n^2), где n - количество параметров

Вопрос: можно ли использовать для подбора части параметров?

# Генетический алгоритм



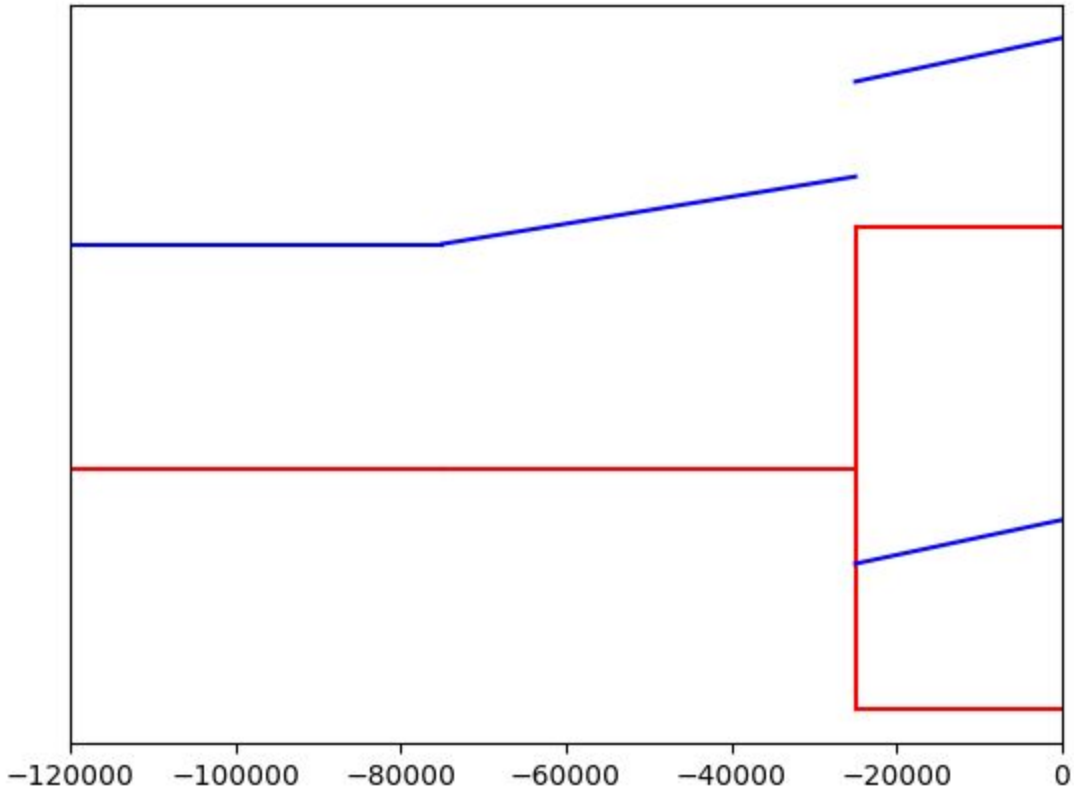

Используем генетику для генетики!

Модель представляет собой список периодов.

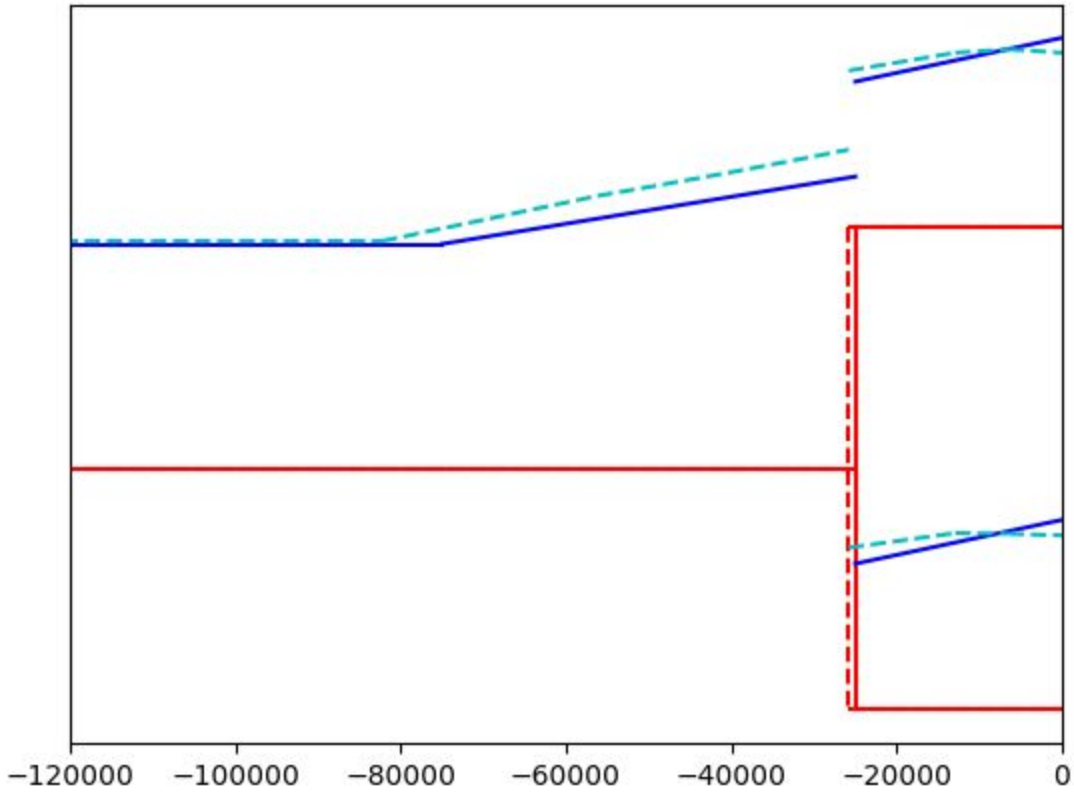
Схема генетического алгоритма:

- Задать начальную популяцию из K особей (демографических моделей)
- Пока не сойдется:
  - Скрестить особей: выбрать две модели и обменять их периодами
  - Мутации особей: поменять у модели один случайный параметр
  - Сгенерировать случайных особей
  - Посчитать значение целевой функции на особях: наш log likelihood модели
  - Отсортировать особей по приспособленности (по целевой функции)
  - Выбрать первые K особей

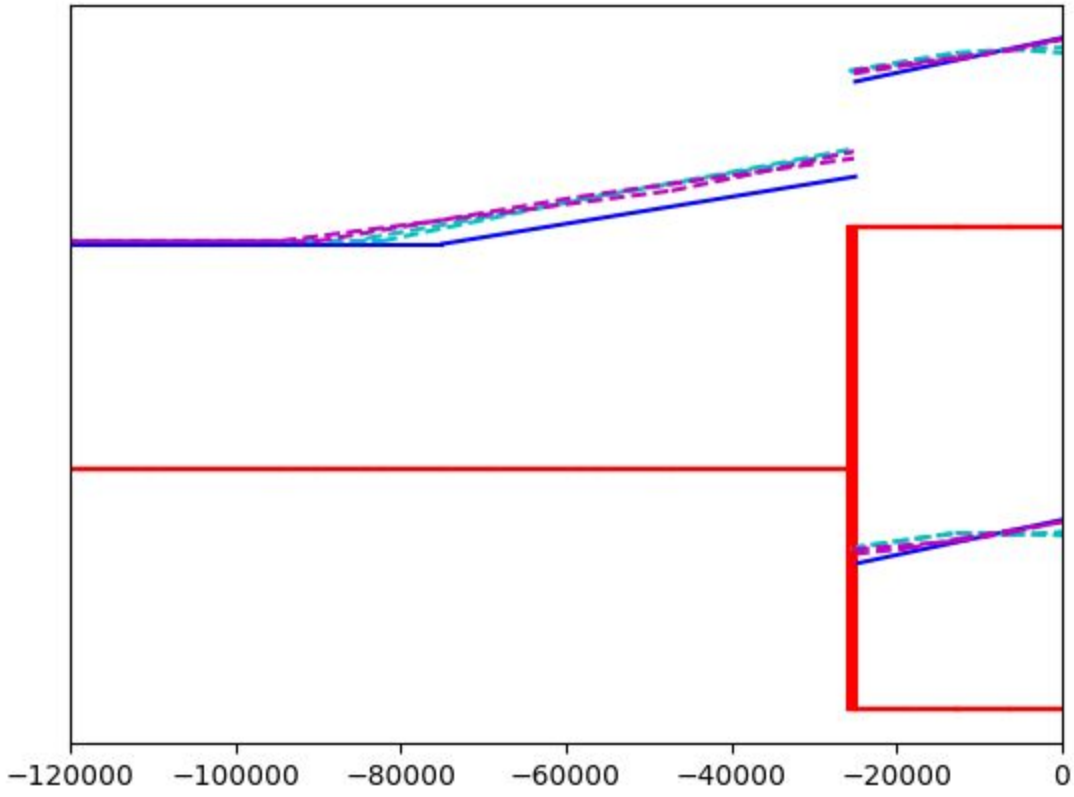
# Результаты: симулированные данные



# Результаты: симулированные данные



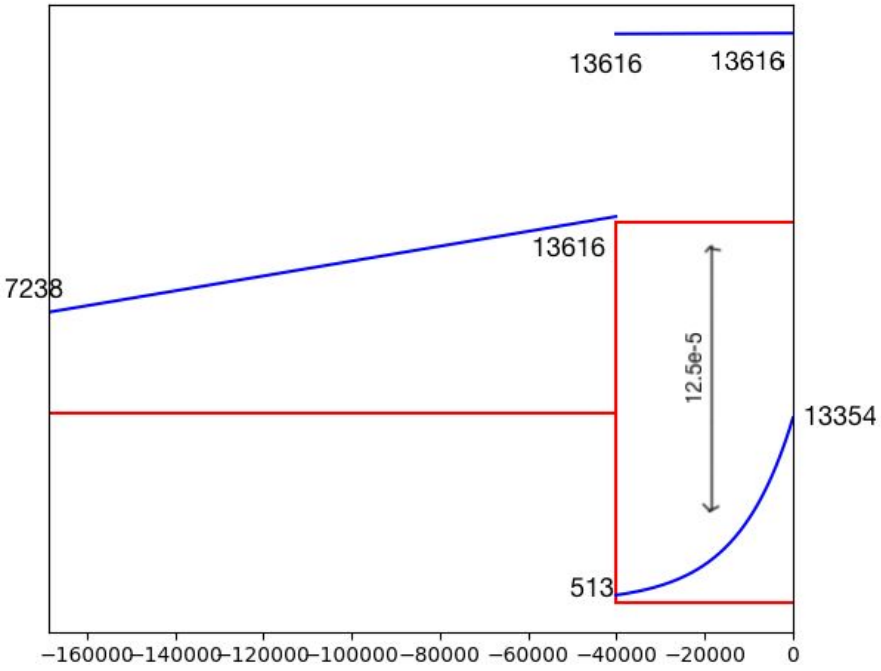
# Результаты: симулированные данные

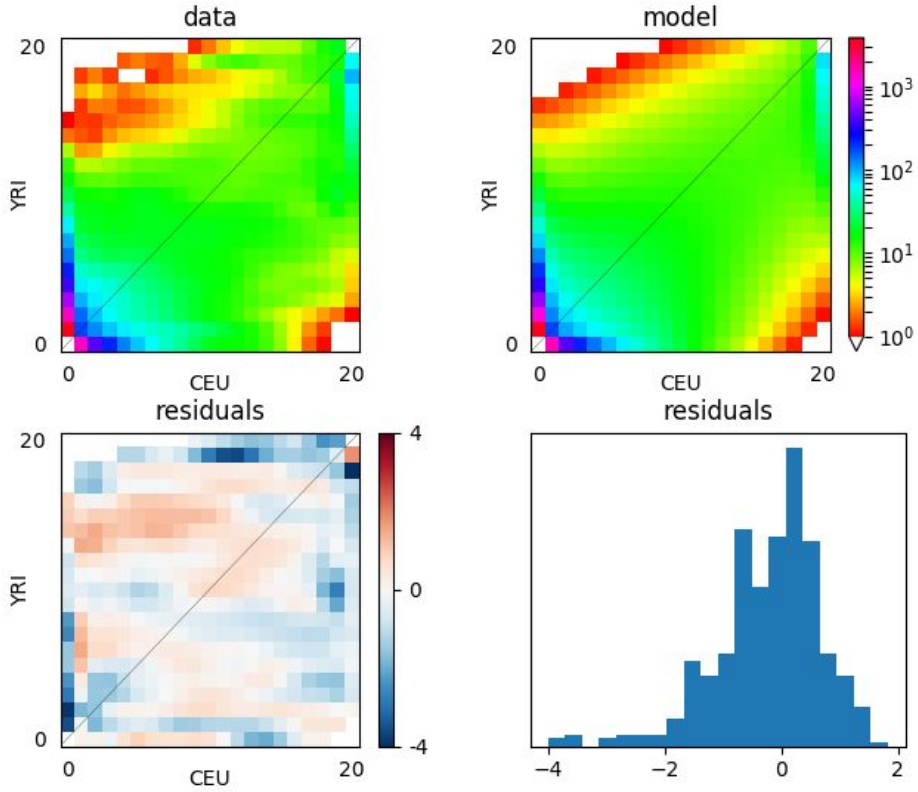


# Демографическая история людей

- Данные AFS были взяты те же что и в статье про dadi.
- YRI - люди народа Йоруба из Нигерии.
- CEU - жители штата Юта с предками из северной и западной Европы.

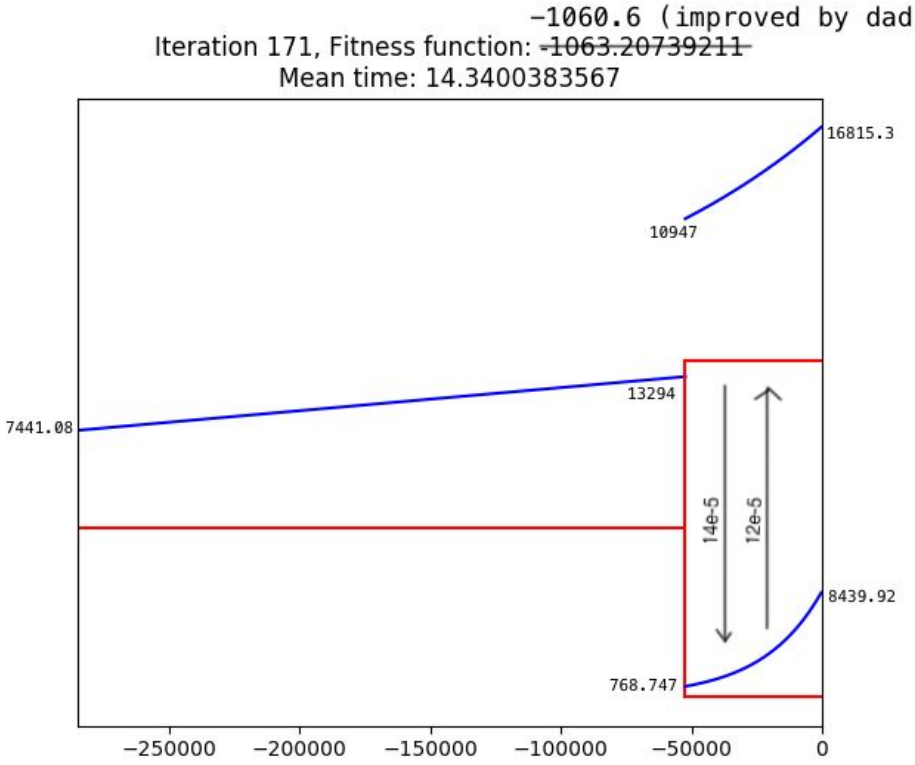
likelihood = -1066

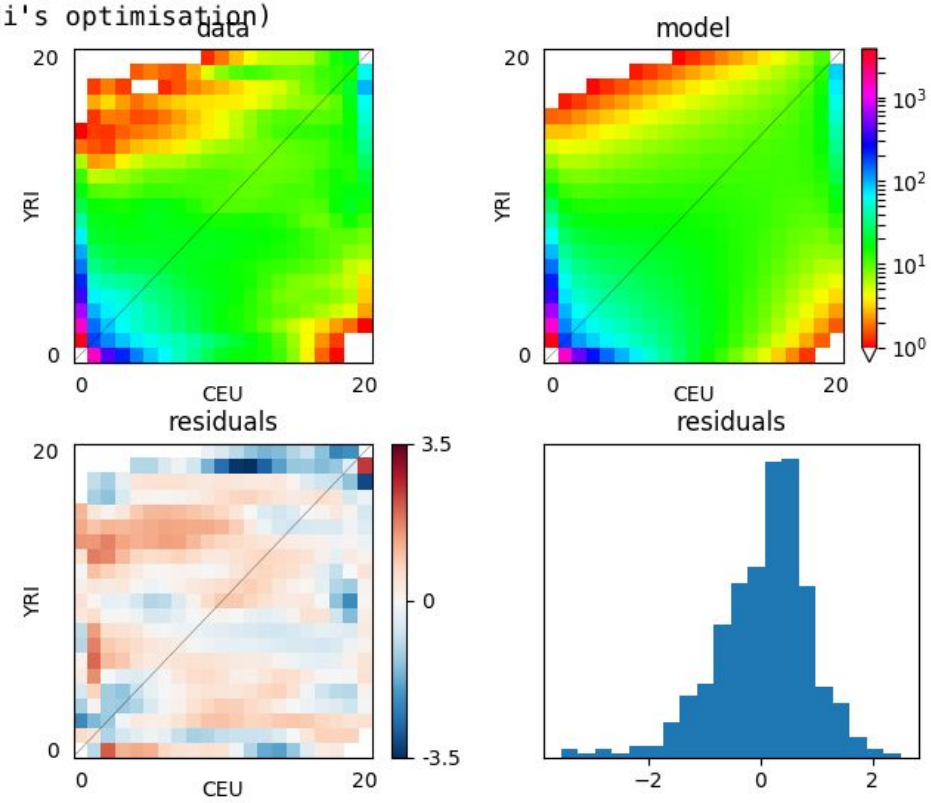


# Полученная модель

- Данные AFS были взяты те же что и в статье про dadi.
- YRI - люди народа Йоруба из Нигерии.
- CEU - жители штата Юта с предками из северной и западной Европы.

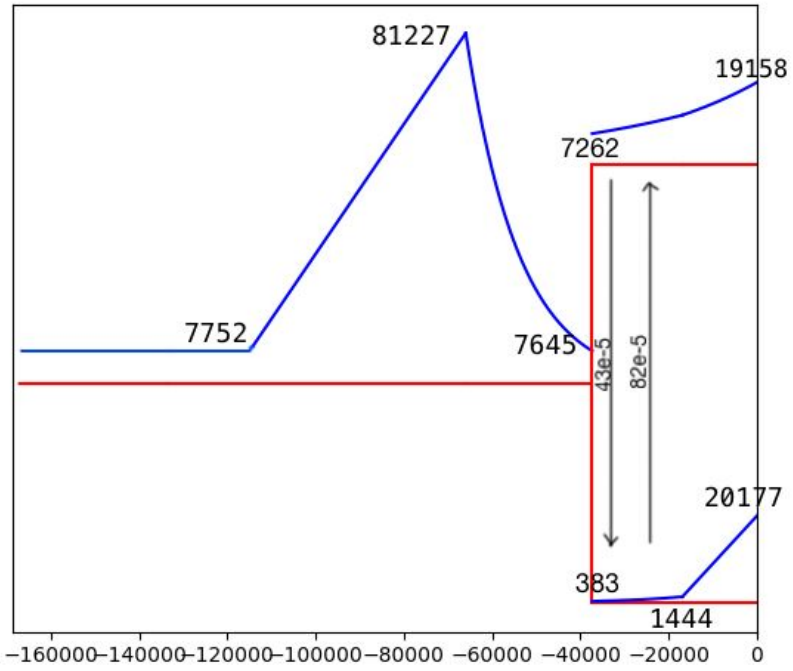


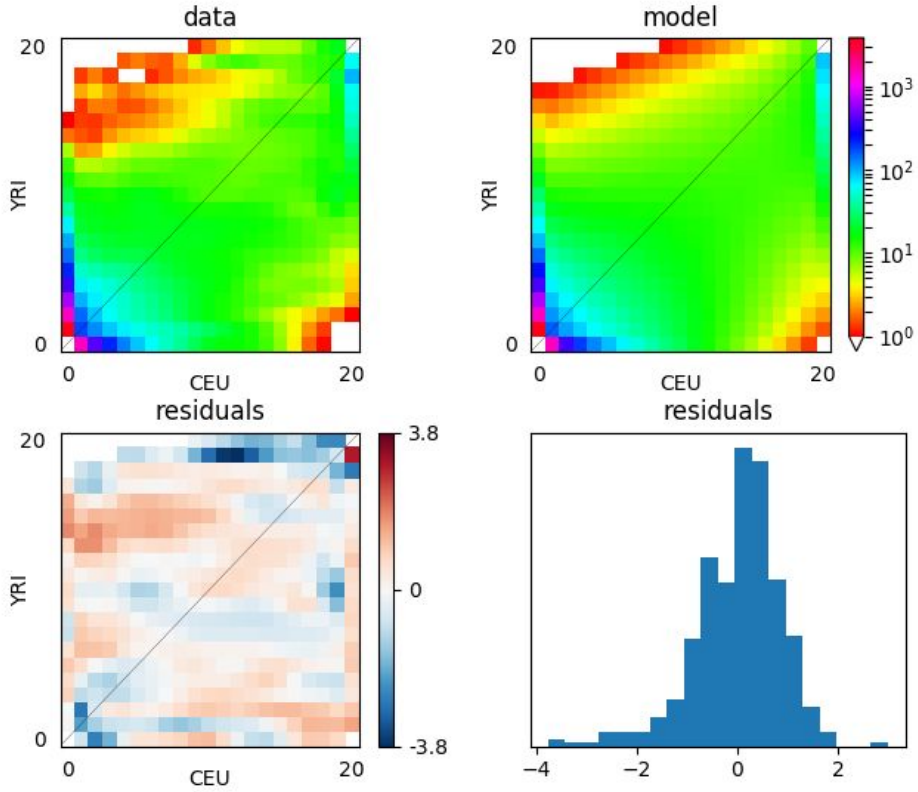



# Полученная модель

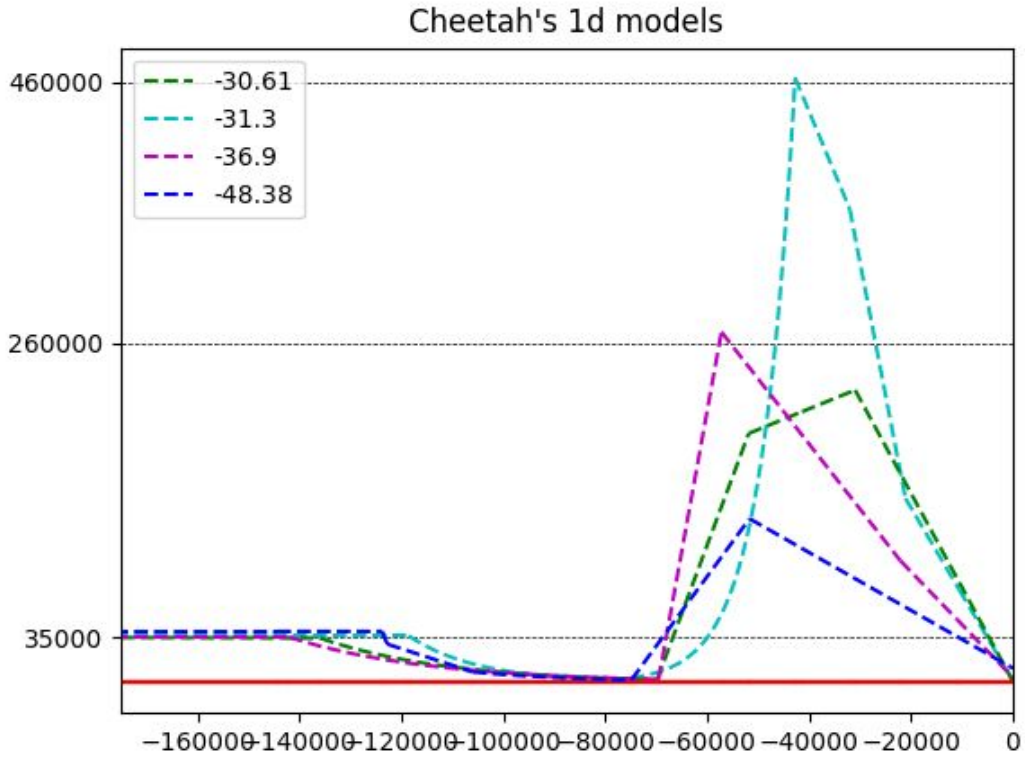
- Данные AFS были взяты те же что и в статье про dadi.
- YRI - люди народа Йоруба из Нигерии.
- CEU - жители штата Юта с предками из северной и западной Европы.

likelihood = -1053



# С чего все начиналось или демографическая история гепардов

# Результаты

- Был разработан и реализован генетический алгоритм для построения демографических историй
- Были симулированы данные
- Алгоритм был проверен на различных симулированных данных
- Алгоритм был проверен на демографической истории людей из статьи
- Были получены новые результаты по демографической истории гепардов

Спасибо за внимание!