

# ProtGen

магистрант Диевский А. В.

# Задача

- Изучить движение белка между известными конформациями
- Движение белка осуществляется изменением торсионных углов
- Хотелось применить генетические методы для нахождения глобально оптимальной траектории

# Незадача

- Техническая часть оказалась существенно сложнее, чем ожидалось
- Имеющаяся программа поиска локально оптимальной траектории пока что непригодна для использования в качестве подзадачи

# Что сделано

- Разработан инструментарий из нескольких функций и процедур на языке **R** (встроенная поддержка векторных вычислений)
- В будущем планируется оформить этот набор как библиотеку
- Используются некоторые функции из библиотеки **bio3d**

# УТИЛИТЫ

Несколько вспомогательных функций:

- вычисление длин связей
- обработка PDB-данных
- построение XYZ-координат по плоским и торсионным углам и длинам связей
- случайное искажение набора торсионных углов
- стохастическая линейная интерполяция двух наборов торсионных углов

# mutate.pdb

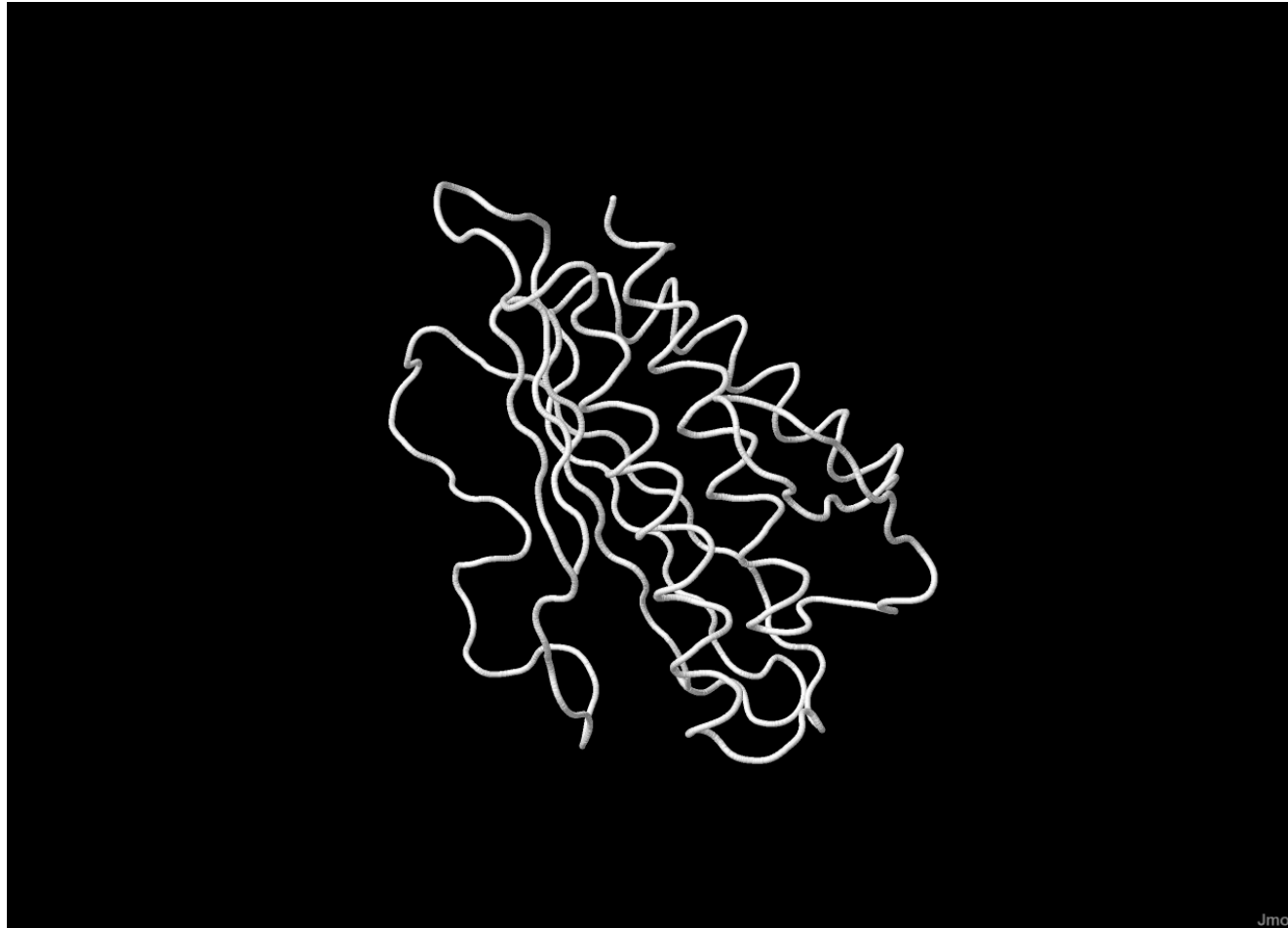
- **.protgen.mutate.pdb** (*pdb*, *sd=1*, *conserve.helix=FALSE*, *conserve.sheet=FALSE*,...)
- Возвращает PDB-структуру искажённого белка.
- *sd* – стандартное отклонение
- при *sd = 1* распределение почти равномерное
- при *sd = 0* углы сохраняются

# mutate.pdb

**.protgen.mutate.pdb** (*pdb*, *sd=1*,  
*conserve.helix=FALSE*,  
*conserve.sheet=FALSE*,...)

- *sd = c(1, 0, 1)* –  $\omega$ -углы зафиксированы
- *conserve.helix* и *conserve.sheet*  
позволяют зафиксировать углы  
альфа-спиралей и бета-листов

$$sd = 0$$

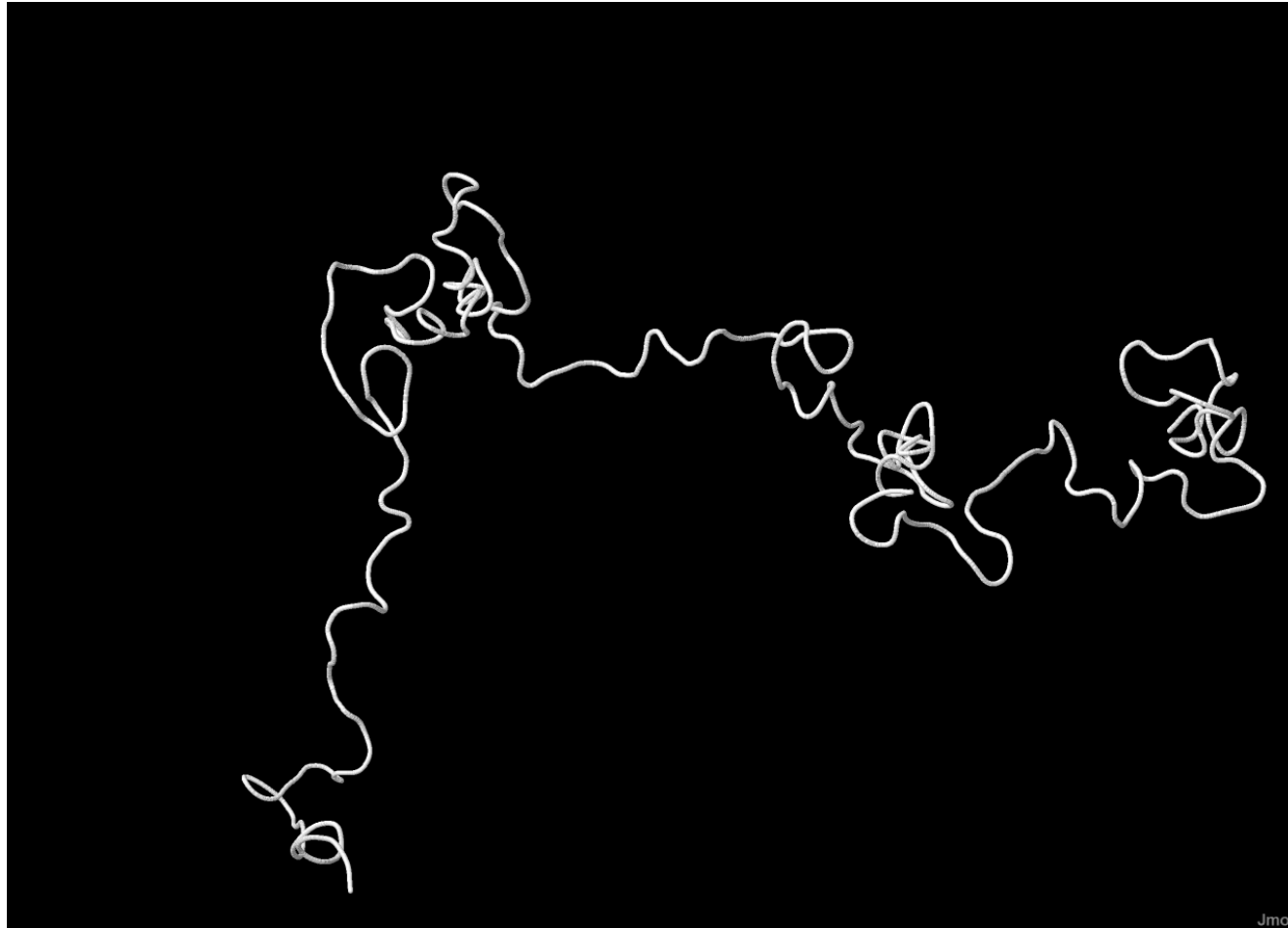




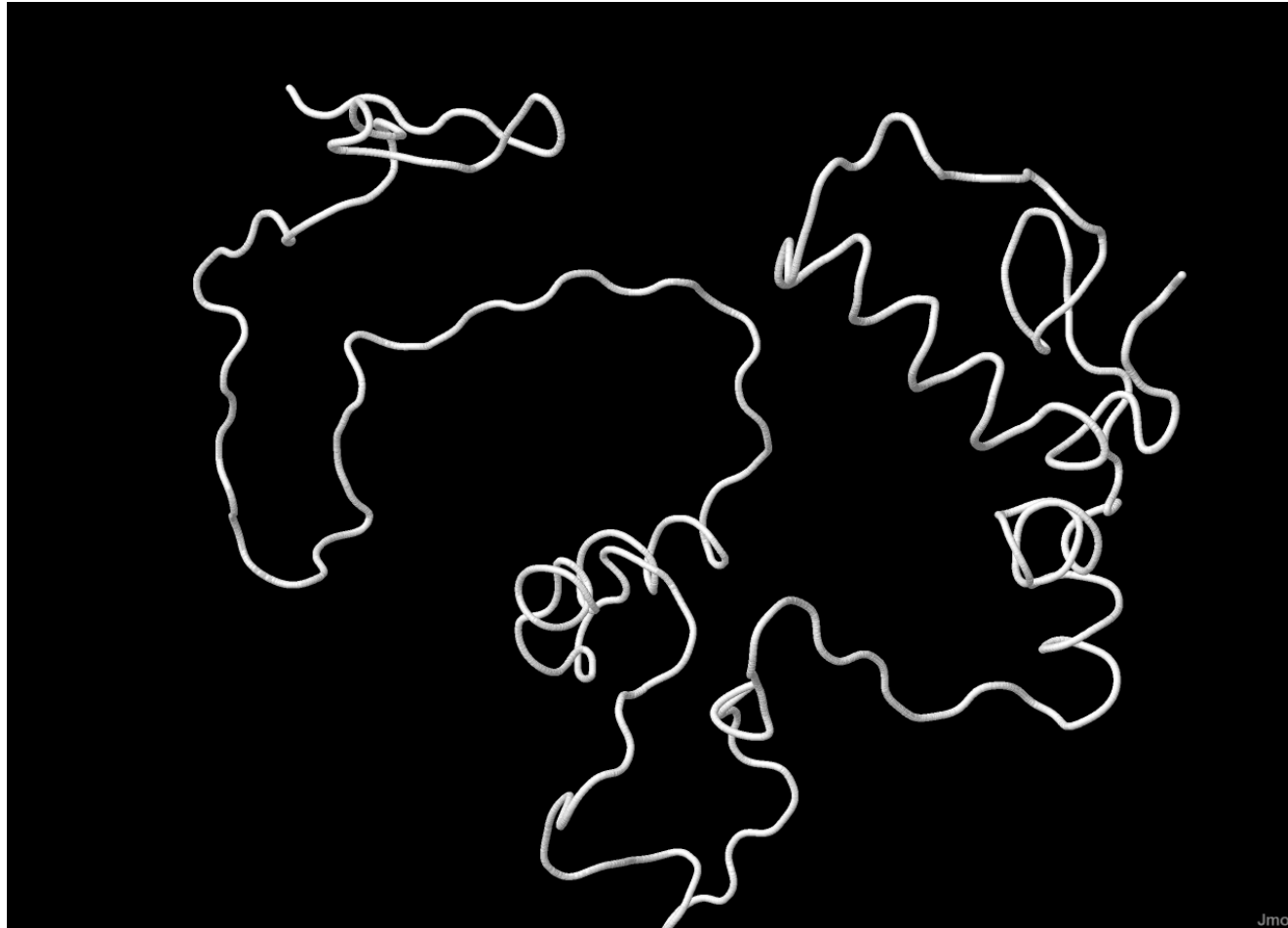
$sd = 0.1$



sd = 1



sd=1, conserve.helix=TRUE,  
conserve.sheet=TRUE



# randomized.linear.trajectory

## **.protgen.randomized.linear.trajectory**

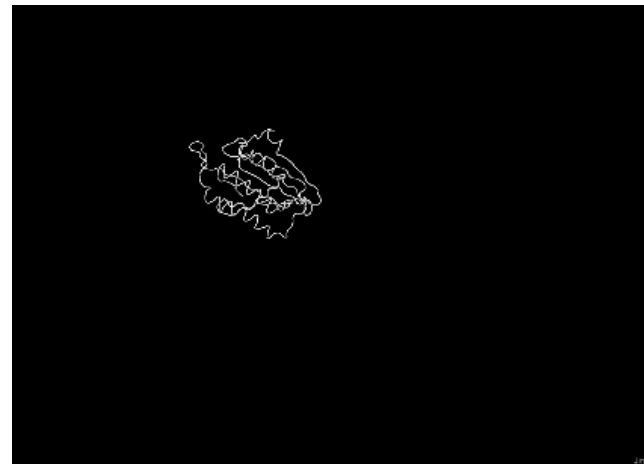
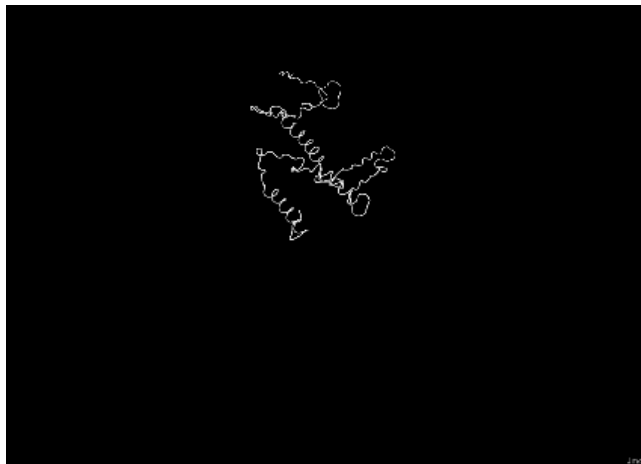
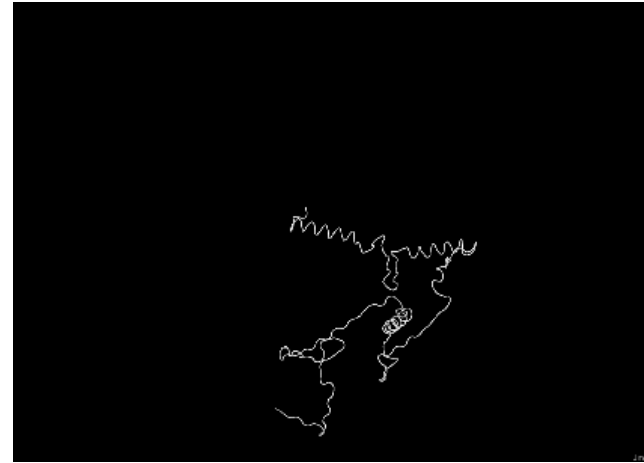
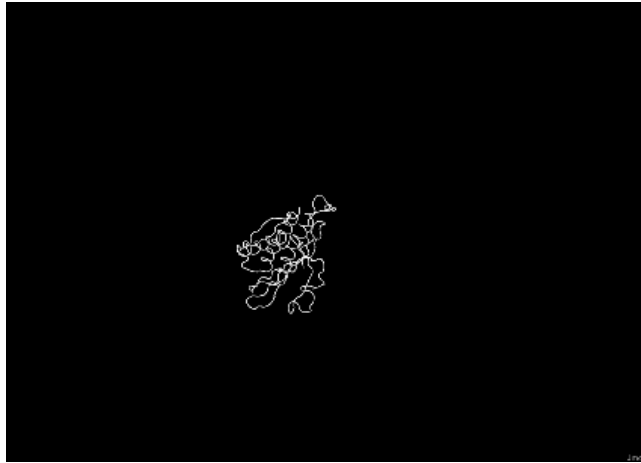
*(pdb1, pdb2, lambda=1:9/10,  
prob=function(x) (x/180)^2/2, ...)*

- Движение из *pdb1* в *pdb2* с помощью стохастической линейной интерполяции торсионных углов
- Возвращает список из  $length(lambda)+2$  PDB-структур

# randomized.linear.trajectory

- Линейную интерполяцию между двумя углами можно осуществлять по любой из двух дуг между ними
- Чем ближе эти дуги к  $180^\circ$ , тем меньше оснований выбирать меньшую
- Поэтому при интерполяции дуга выбирается случайно (вероятность зависит от длины дуги)

# randomized.linear.trajectory



# mutate.trajectory

**.protgen.mutate.trajectory** (*trajectory*, ...)

- Искажает среднюю конформацию траектории и проводит к ней стохастические линейные интерполяции из начала и из конца траектории
- Можно передавать параметры к функциям **mutate.pdb** и **randomized.linear.trajectory**

# write.trajectory

**.protgen.write.trajectory** (*trajectory*, *file* = *"Rmotion.pdb"*)

- Записывает траекторию как мультимодельный PDB-файл
- Может быть прочитан Jmol и другими приложениями



# Заключение

- Разработан инструментарий для порождения большого количества случайных траекторий
- Применение к ним метода градиентного спуска должно помочь в поисках глобального минимума