

# Простое уравнение силового поля (СП)

$$\begin{aligned}
 U = & \sum_{bonds} \frac{k_i}{2} (l_i - l_0)^2 + \sum_{angles} \frac{k_i}{2} (\phi_i - \phi_0)^2 + \sum_{torsions} \frac{V_n}{2} (1 + \cos(n\omega - \gamma)) + \\
 & + \sum_{i=1}^N \sum_{j=i+1}^N \left( 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right)
 \end{aligned}$$

Молекулярная динамика

Монте-Карло

$$v(t + \frac{\Delta t}{2}) = v(t - \frac{\Delta t}{2}) + \frac{F(t)}{m} \Delta t$$

$$acc(o \rightarrow \eta) = \min \left( 1, \exp \left\{ -\beta \left[ U(r^N) - U(r^N) \right] \right\} \right)$$

# Метод Монте-Карло

- Основная идея предполагает поиск конформаций с низкой энергией на основе случайного изменения координат.
- Скорости у атомов не рассчитываются

Представим, что канонический ансамбль можно разбить :

$$Q_{NVT} = Q_{NVT}^{ideal} Q_{NVT}^{excess}; \quad Q_{NVT}^{excess} = \frac{1}{V^N} \int dr^N \exp \left[ \frac{-U r^N}{k_b T} \right]$$

Энергия Гемгольца:

$$A = -k_b T \ln Q_{NVT} \rightarrow A = A^{ideal} + A^{excess}$$

# Расчёт термодинамических свойств

Формально нам надо интегрировать по всему пространству конформаций, это невыполнимо.

Приближение:

- 1 Генерируем конформацию
- 2 Считаем потенциальную энергию:  $\exp\left(\frac{U(r^N)}{k_b T}\right)$
- 3 Считаем фактор Больцмана.
- 4 Суммируем известные факторы и находим среднее и переходим к пункту 1.

В результате средняя энергия определяется как:

$$\langle U(r^N) \rangle = \frac{\sum_1^{N_{trial}} U_i(r^N) \exp\left[\frac{U(r^N)}{k_B T}\right]}{\sum_1^{N_{trial}} \exp\left[\frac{U(r^N)}{k_B T}\right]}$$

# Расчёт термодинамических свойств

- Неудобно, что основной счёт приходит на ненужные состояния.
- Метрополис и соавторы предложили использовать цепи Маркова для генерации конформаций.

Пример:

- 1 генерируем:  $x_{new} = x_{old} + (2\xi - 1)\delta r_{max}$
- 2 если новая энергия меньше старой то принимаем конформацию и используем дальше
- 3 если энергия выше то сравниваем фактор Больцмана со случайным числом 0:1 и если фактор Больцмана изменения энергии больше, то принимаем конформацию

# Моделирование жестких молекул (rigid body)

- Так как молекула не точка, мы смещаем центр масс.
- Нам важно не только смещение молекулы, но и её ориентация.
- Удобно использовать сферические координаты и вектор отображающий основную ось молекулы.

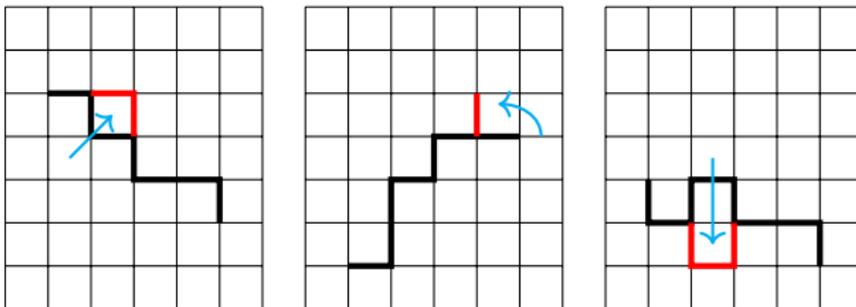
# Моделирование гибких молекул

- Монте-Карло моделирование гибких молекул проводить сложнее.
- Реалистично только для малых молекул и молекул с замороженными степенями свободы.
- Часто замораживают длину валентных связей и валентные углы.
- Для больших молекул незначительное изменение торсионного угла может привести к высоким энергиям.

# Моделирование полимеров

## Решёточные модели полимеров.

Представим что один мономер занимает один кубик (аналог крупно-зернистого моделирования)



# Методы выборочного поиска (Biased Monte-Carlo)

**Суть идеи:** сканировать не все пространство конформаций, а только объект интереса.

**Пример:** Нас интересует поведение молекулы в воде. Поведение воды в далеке от молекулы не важно.

Существуют:

- Силовое предпочтение — направляет движение молекулы вдоль действия силы.
- Умное (Smart) предпочтение — тоже использует силы для определения движения. В случае силового предпочтения есть ограничения на движение. В большинстве случаев оба метода очень похожи.

# Проблема квази-эргодичности

**Суть проблемы:** При наличии высокоэнергичных барьеров метод Метрополиса оказывается в ловушке локальных минимумов энергии. Происходит накопление данных для весьма ограниченной области фазового пространства.

Существуют два подхода для решения этой проблемы:

- Подход “прыжок-ход” (Jump-walking или J-walking).
- Мультиканонический подход.

# J-walking

**Суть идеи:** При движении в пространстве при заданной температуре, разрешены переходы в пространство с более высокой температурой.

**Простейшее решение:** проводить 2 и более моделирования параллельно при разных температурах. Моделирование при высокой температуре делают с предпочтением в фазовое пространство при низкой температуре.

# Мультиканонический ансамбль

При каноническом ансамбле вероятность ( $P$ ) оказаться в точке с энергией  $E$ , пропорциональна фактору Больцмана умноженному на плотность состояний между точками  $E_1$  и  $E_2$ .

Фактор Больцмана:  $w_B = \exp\left[-\frac{E}{k_B T}\right]$

$$P_{\text{canon}}(T, E) \sim n(E)w_B(E)$$

Этот вариант предполагает существенные скачки энергии при переходах.

# Мультиканонический ансамбль

В мультиканоническом ансамбле вероятность не зависит от энергии в определённых диапазонах энергии. Это равносильно замене фактора Больцмана на некий мультиканонический фактор  $w$ .

$$P_{mu}(E) \sim n(E)w_{mu}(E) = constant$$

Определение этого фактора не тривиальная задача. Сначала делают поиск при высокой температуре. На основе результатов определяют состояния близкие к наименьшей энергии и наоборот.

$$w_{mu}(E) = exp[-\beta(E)E - \alpha(E)]$$

# Большой канонический ансамбль

Большой канонический ансамбль предполагает постоянство химического потенциала.

$$\mu = k_B T \ln(\Lambda^3 z); \quad \Lambda = \sqrt{\frac{h^2}{2\pi m k_B T}}$$

Ключевые особенности большого канонического ансамбля:

- Смещение частицы происходит методом Метрополиса
- Частица может быть уничтожена
- Частица может появиться в случайном положении
- Необходимо аккуратно подобрать частоту уничтожений/появлений.

# Что выбрать молекулярную динамику или Монте-Карло?

**Молекулярная динамика:** незаменима когда нужно найти свойство базирующиеся на эволюции системы во времени.

**Монте-Карло:** гораздо удобнее для работы с заданным ансамблем. Работать с большими молекулами не удобно, но возможность «нефизических” переходов позволяет проводить эффективный поиск в фазовом пространстве.

**Методы комплиментарны:** не удивительно, что существуют подходы с использованием обоих методов одновременно.

# Фреймворк Rosetta

- Rosetta это унифицированный пакет для предсказания структуры белка и функционального дизайна.
- Для решения этих задач надо:
  - Исследовать конформационное пространство белка
  - Для дизайна надо еще исследовать пространство последовательностей
  - Используется эмпирически адаптированный алгоритм Монте-Карло и эмпирические функции для расчёта энергии (PDB)

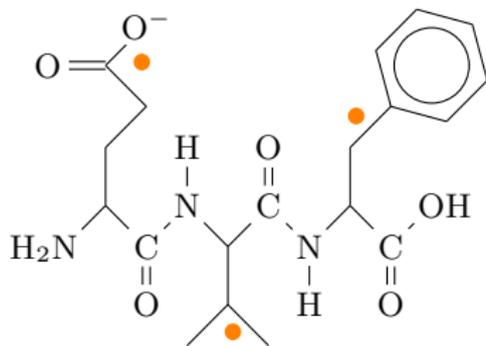
# Исследование конформационного пространства остова

- Изменения в конформации остова белка делятся на локальные и глобальные
- Глобальные изменения описываются обменом конформаций между фрагментами определенной длины (3-6-9). Конформации фрагментов получают из PDB
- Локальная оптимизация достигается перебором  $\phi, \psi$  углов, так, чтобы не происходило глобального изменения фолда.

# Конформации боковых радикалов

- Для значительного числа аминокислот перебрать все конформации боковых цепей невозможно.
- Rosetta предлагает значительно сократить количество конформаций состояниями из PDB - rotamers
- Используется Монте-Карло моделирование отжига для поиска глобального минимума

# Функция для расчёта энергии, centroid



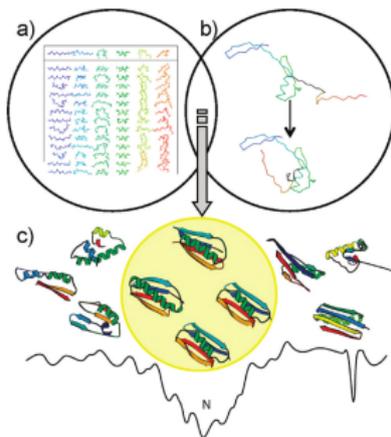
- Сольватация описывается на основе вероятности такой конформации в PDB
- Электростатика описывается вероятностью расстояния между центроидами в PDB
- Водородные связи между  $\beta$  тяжами описываются на основе геометрии расположения тяжей

# Полноатомные функции энергии

- VdW : Lennard-Jones
- Гидрофобика: SAS
- Эмпирический потенциал для водородной связи на основе конформации
- Эмпирический потенциал для электростатики (заряды)
- Важная особенность: потенциалы попарно разложимы (оптимизация)

# De Novo Folding Simulation

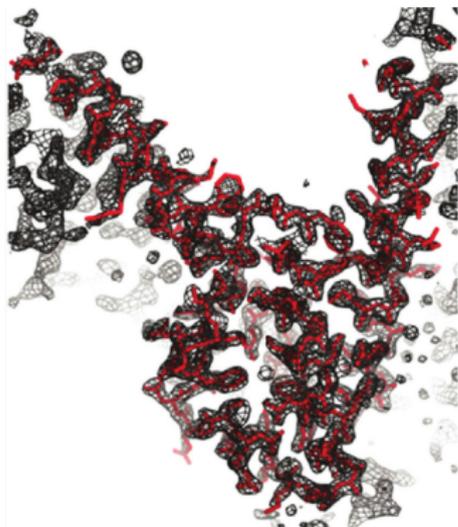
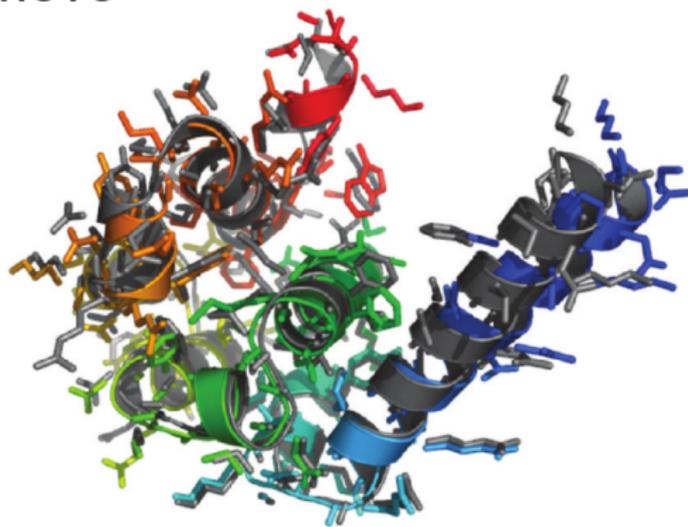
- В линейную конформацию вставляют фрагменты и быстро сканируют пространство укладывая остов белка. 9-мер 30000 раз затем 10000 раз 3-мер. Таким образом собирают 25000-50000 моделей.
- Выбранные модели оптимизируют на полноатомном уровне.



# Релаксация и улучшение моделей

- В ходе релаксации незначительно изменяются углы остова.
- Боковые радикалы укладываются с помощью Монте-Карло моделирования отжига.
- В конце применяется градиентная оптимизация по всем торсионным углам.
- Возможно последовательное увеличение коэффициента при потенциале отталкивания
- Есть работы где в ходе релаксации накладывались ограничения из ЯМР или РСА

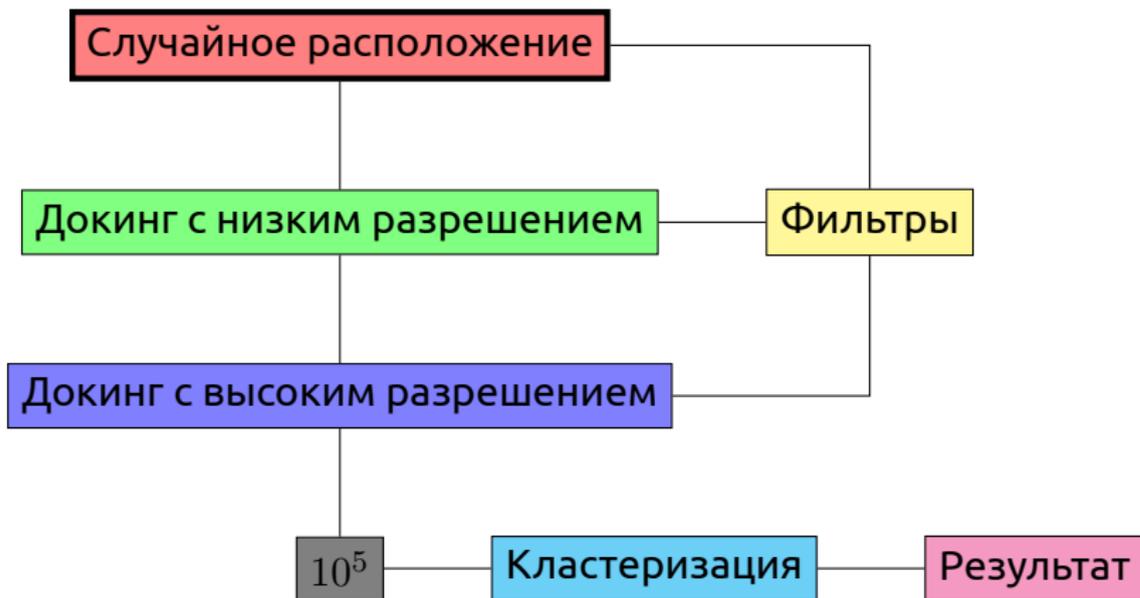
# Использование экспериментальных данных в de novo



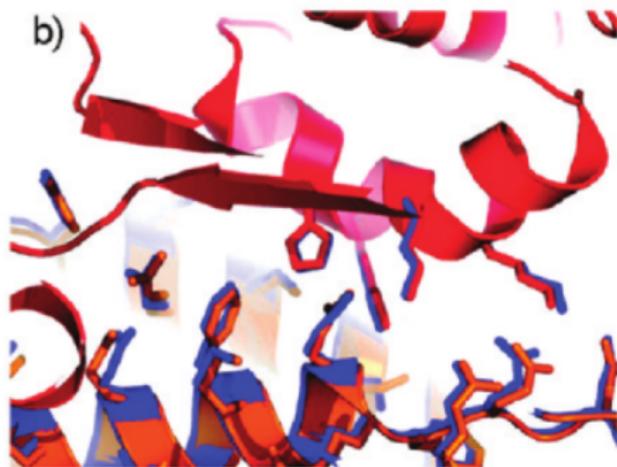
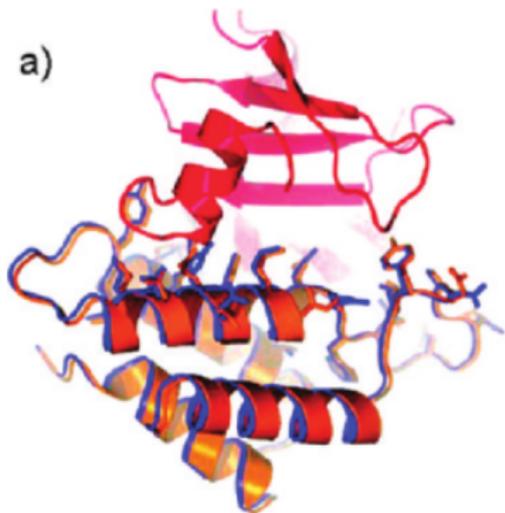
DOI: 10.1021/bi902153g

- Слева ЭПР данные + Rosetta, серым отмечена структура по данным PCA
- Справа: решение проблемы фазы с помощью моделей Rosetta и автоматического алгоритма молекулярного

## Белок-Белок докинг: Rosettadock



# Rosettadock, результат



модель окрашена синим

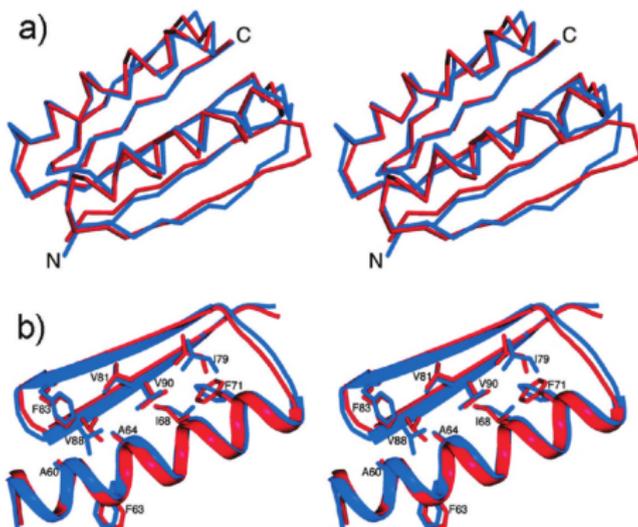
DOI: 10.1021/bi902153g

# RosettaLigand

- Основное преимущество это одновременный учет подвижности лиганда и белка
- Надо знать конформеры лиганда
- Можно учитывать не только подвижность боковых цепей, но и остова
- Надо использовать в системах где предполагается индуцированное конформационное изменение

# Дизайн de novo

- Надо менять не только конформацию в ходе моделирования но и последовательность.
- Была построена модель белка, укладки которого не было в PDB



# Другие способы дизайна

- Редизайн известных белков для увеличения устойчивости структуры
- Изменение места контакта с другими биополимерами полимерами (нуклеаза-ДНК)
- Дизайн ферментов, оптимизация структуры и последовательности белка для улучшения заданного переходного состояния.

# Использование Rosetta

AbinitioRelax AnchoredDesign  
antibody\_graft BuildPeptide  
docking\_protocol enzyme\_design  
FlexPepDocking FloppyTail  
hbs\_design homodimer\_design  
ligand\_dock loopmodel  
loops\_from\_density

membrane\_abinitio2 relax  
remodel rna\_denovo rna\_design  
rosettaDNA rosetta\_scripts  
rotamer\_recovery SymDock  
UnfoldedStateEnergyCalculator  
zinc2\_homodimer\_design

Подробнее:

<https://www.rosettacommons.org/docs/latest/Application-Documentation.html>

# Разработка Rosetta, PyRosetta

- Язык программирования в Rosetta C++
- К основным функциям и протоколам есть Python bindings

```
In [1]: import sys
import os
import operator
from itertools import *

sys.path.append('/home/anur/pyrosetta')

from rosetta import *
from toolbox import *
from rosetta.utility import vector1_bool
init()
#from prody import *
#from pylab import *
import numpy as np

from numpy.random import randint

from joblib import Parallel, delayed
```

let use pymol for visual anlysis and debugging

```
In [2]: pymover = PyMOL_Mover()
```