Факультет биоинженерии и биоинформатики,
Московский государственный университет имени М. В. Ломоносова

Структурные домены. Пространство укладок

Лекция 5, биоинформатика, 4 курс ФББ МГУ, осенний семестр

Злобин А. С., alexander.zlobin@fbb.msu.ru

Классификации структурных доменов

SCOP (http://scop.mrc-lmb.cam.ac.uk/scop/)

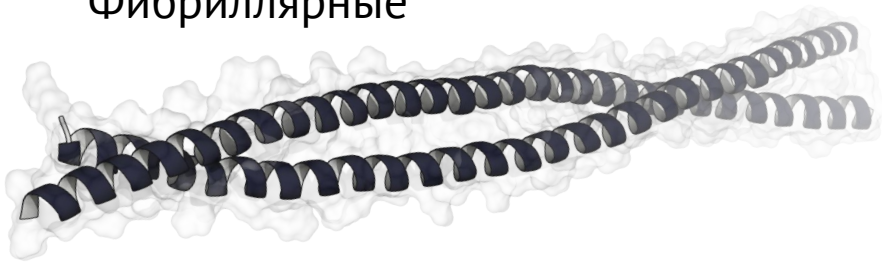
- ручная детекция доменов
- 4 основных уровня классификации (класс, укладка, суперсемейство, семейство), также группирует по физико-химическому типу
- нестрогая иерархичность
- эволюционный контекст

CATH (http://www.cathdb.info/)

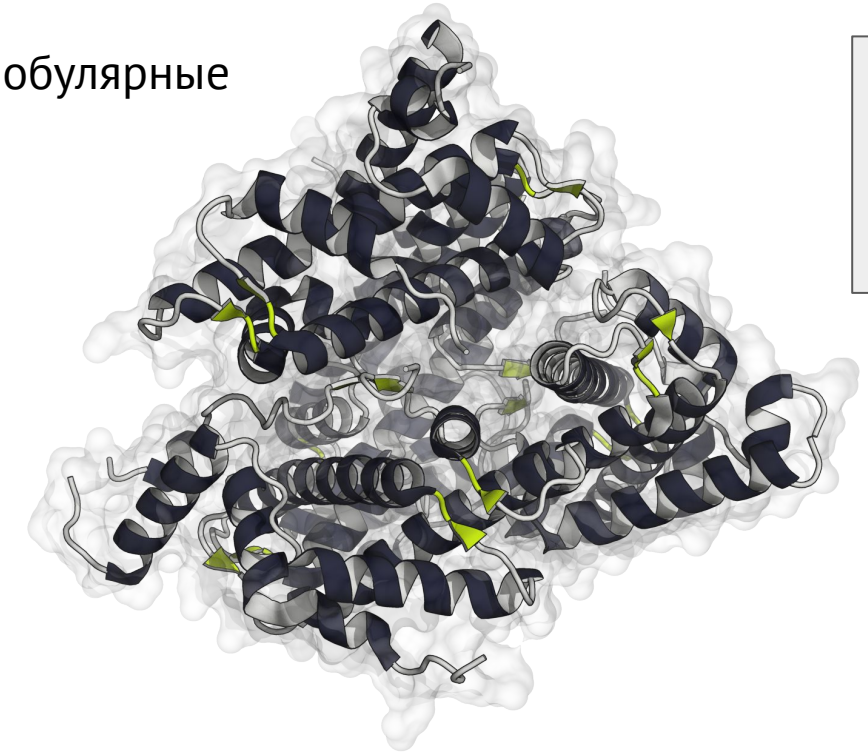
- полуавтоматическая детекция доменов
- 4 основных уровня классификации (класс, архитектура, топология, суперсемейство)
- строгая иерархичность
- без контекста

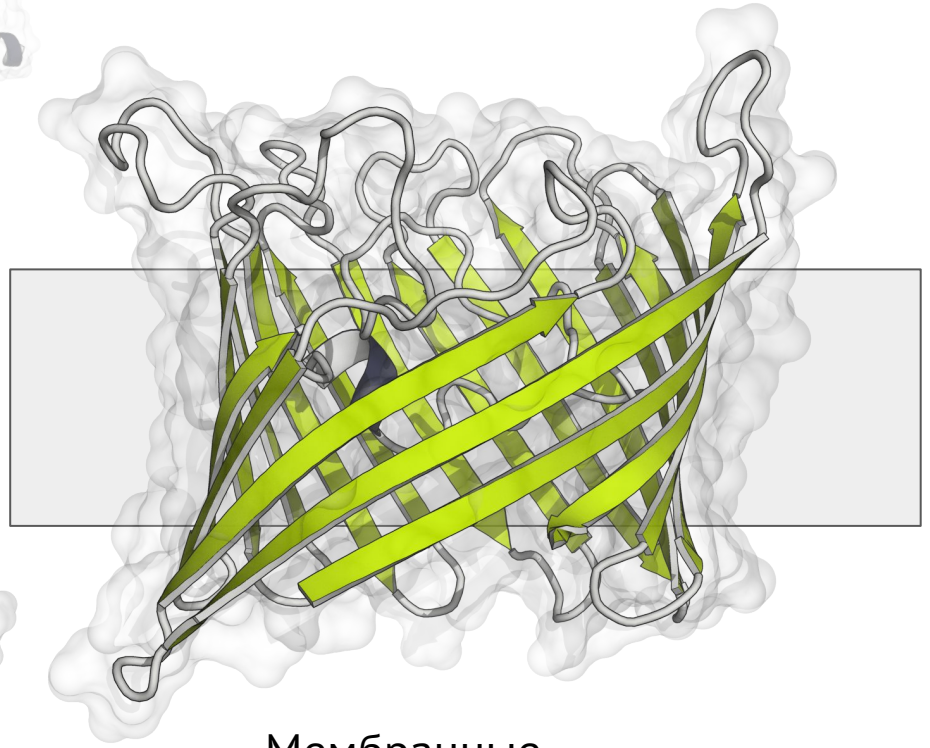
Классификация по физ-хим свойствам

Фибриллярные



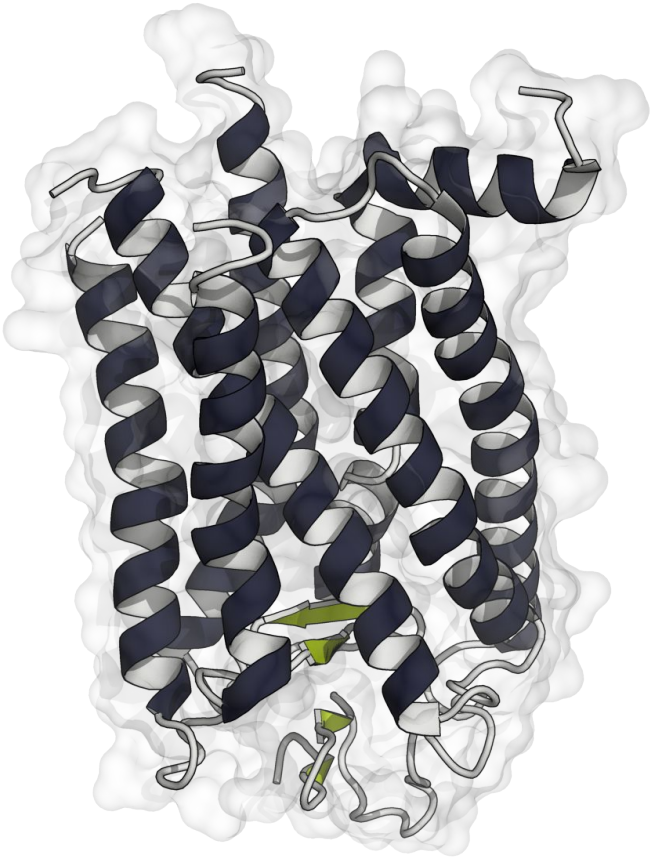
Глобулярные

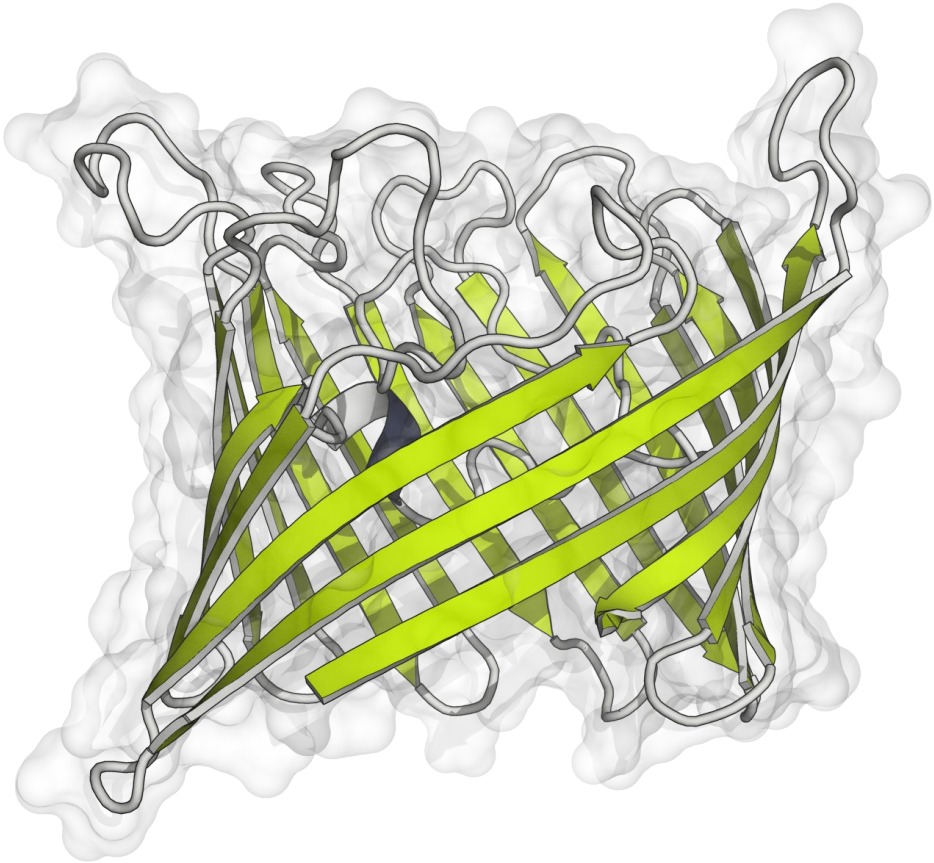




Мембранные

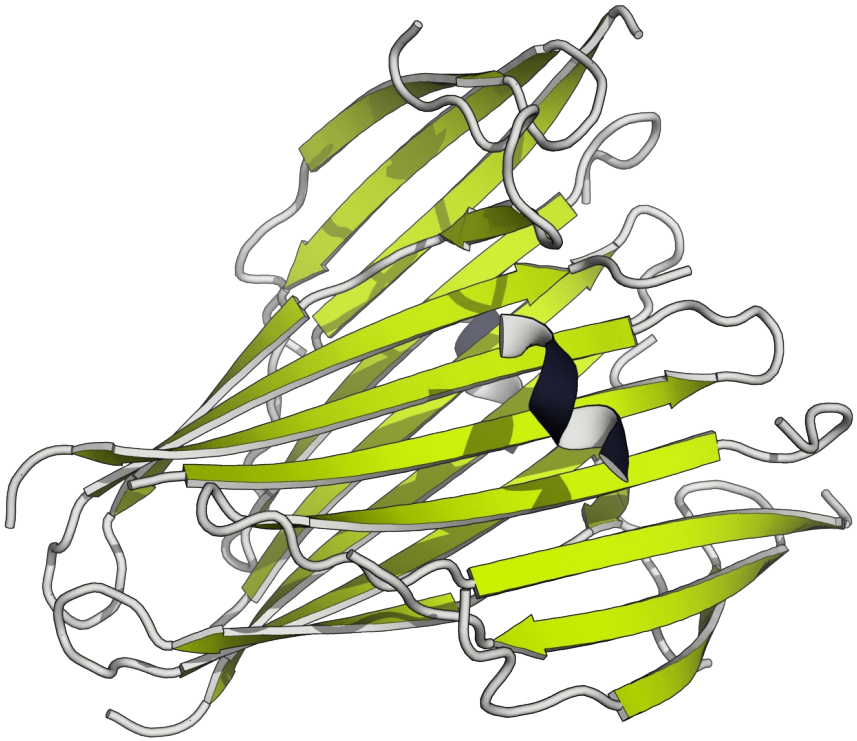
Мембранные белки





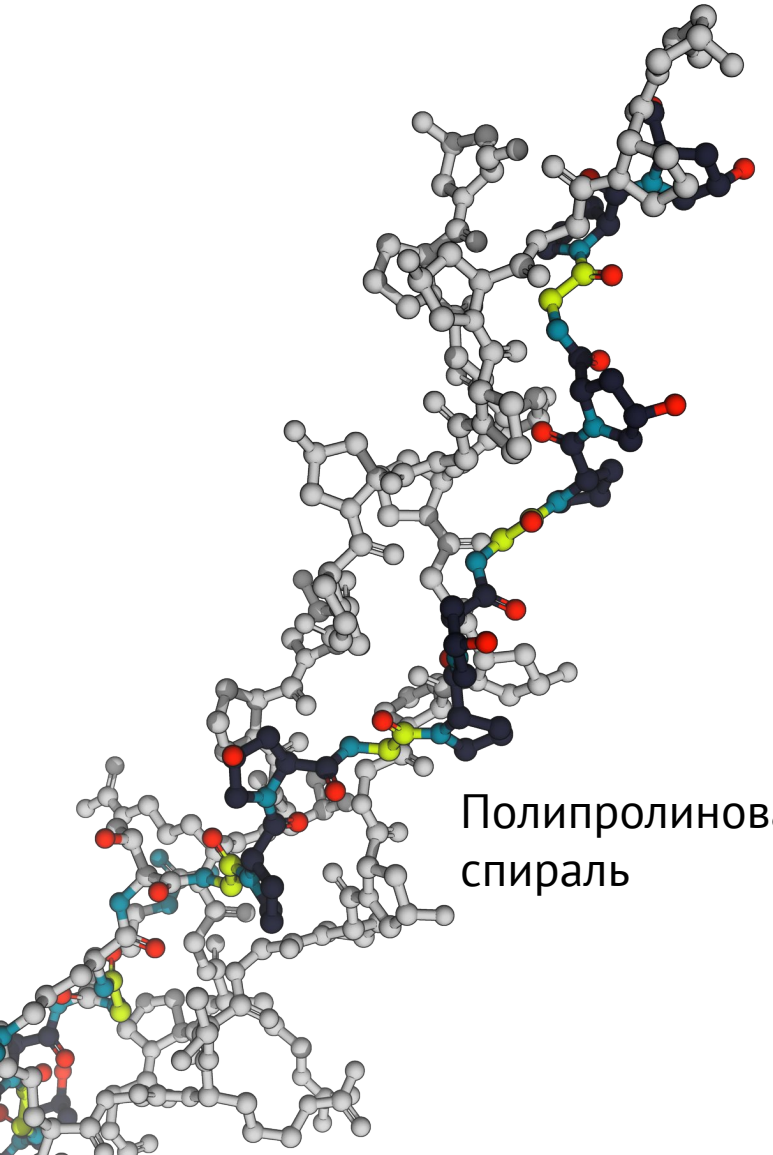
Фибриллярные белки

α-кератин

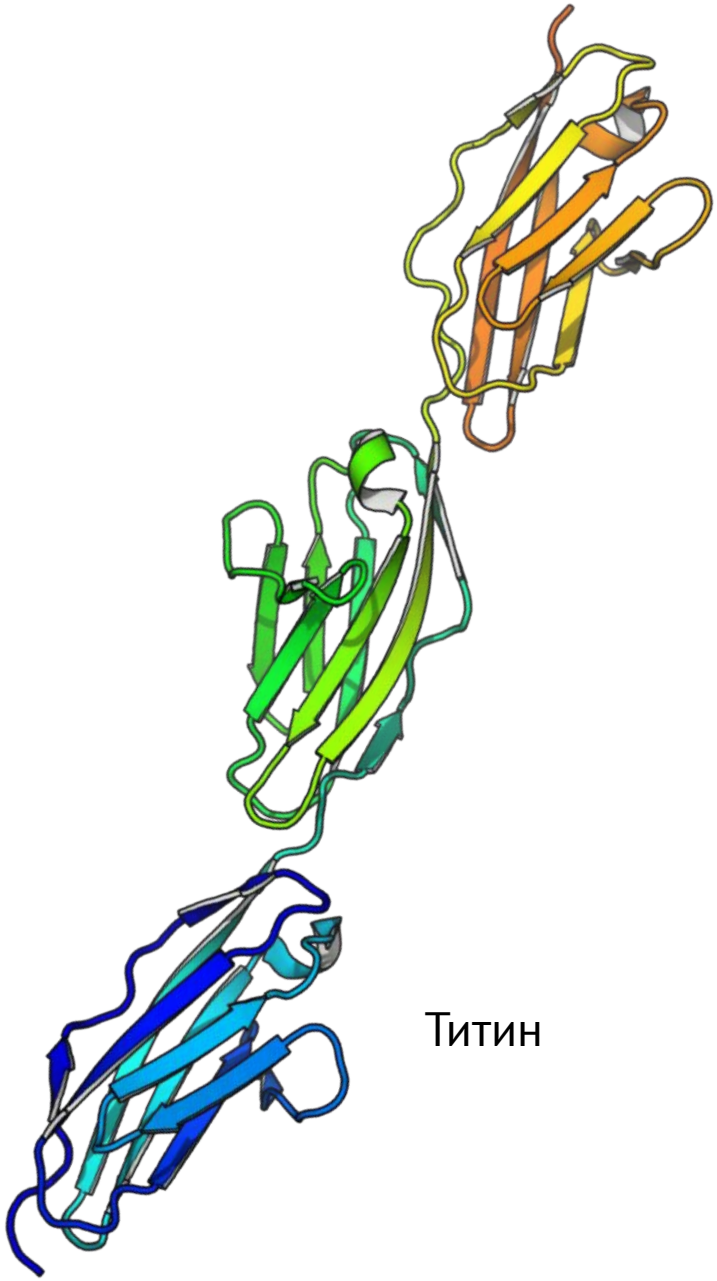


фиброин

Фибриллярные белки

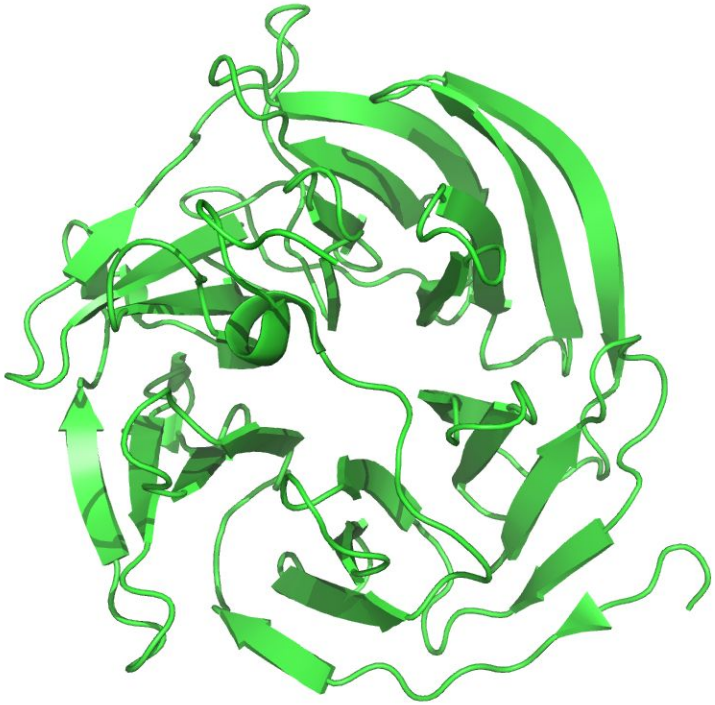


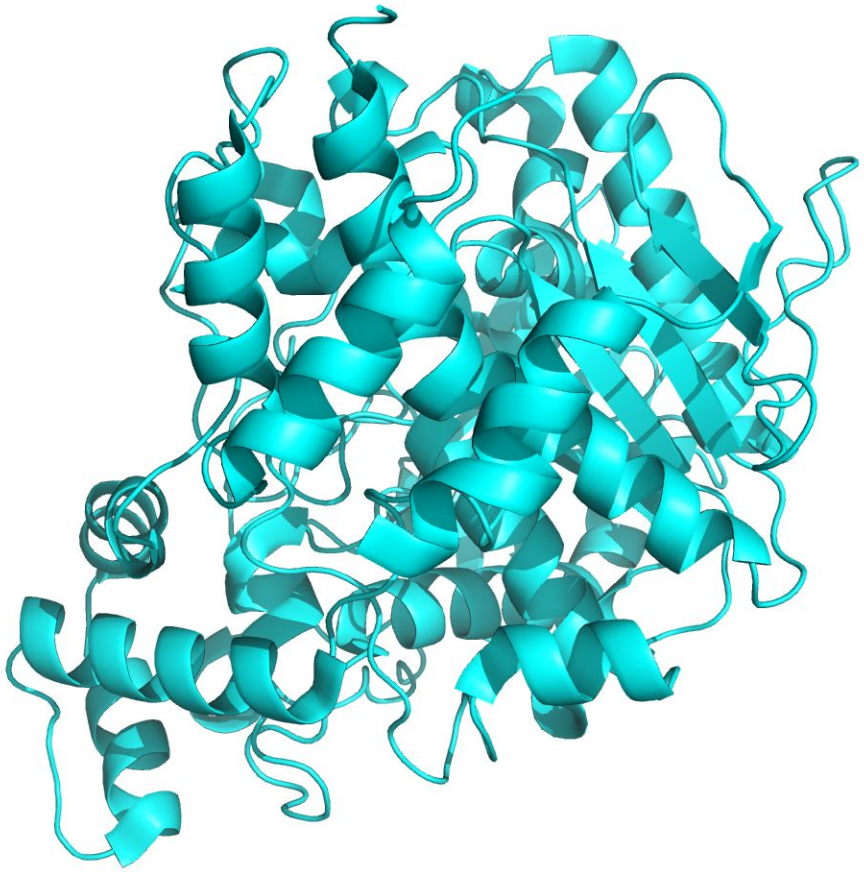
Полипролиновая спираль



Титин

Глобулярные белки





SCOP

В отличие от CATH, ставит своей задачей не просто провести структурную классификацию, а, по-возможности, связать ее с информацией о том, как структуры эволюционируют.

Семейство: есть четкие свидетельства общего происхождения, чаще всего из высокого сходства последовательностей.

Суперсемейство: предположительно общее происхождение, сходство в отдельных структурных деталях, архитектуре активного центра, карманов связывания.

Укладка (фолд): глобальные структурные особенности, топология.

Класс: пропорция элементов вторичной структуры.

Тип: растворимый, мембранный, фибриллярный, неструктурированный.

About Contact Download Search SCOP by text or ID

SUPERFAMILY Osmotin/thaumatin-like
SUPERFAMILY GH64 beta-glucanase insert domain-like

FAMILY

Pfam

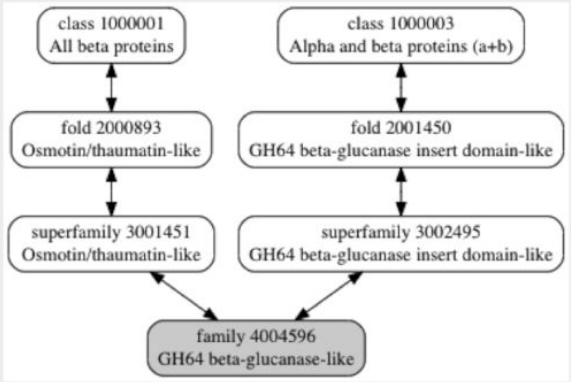
GH64 beta-glucanase-like

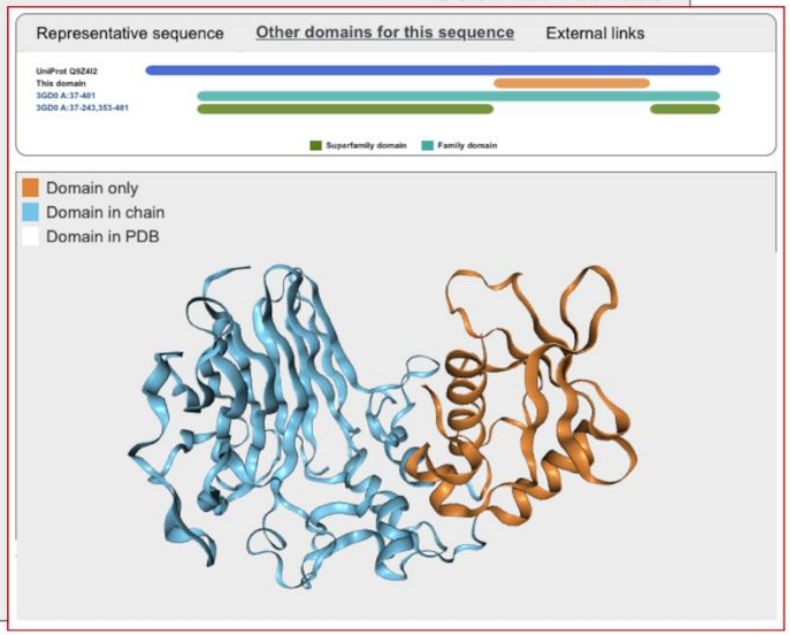
SCOP ID: 4004596

PF16483, comprises two domains from different superfamiles

Function functionally relevant complex structure(s) determined

Show ancestry





Domains [2 entries]	ID	Region	Links
Protein **Laminaripentaose-producing beta-1,3-guluase (LPHase)**			
Species ***Streptomyces matensis***	Q9Z4I2	37-401	UniProt
Representative domain 8045656	3GD0	A:37-401	PDBe
Represented structures [1]			RCSB PDB
3GD9 A			

FOLD PqqD-like WHD fold
SUPERFAMILY Winged helix DNA-binding domain

FAMILY

Pfam

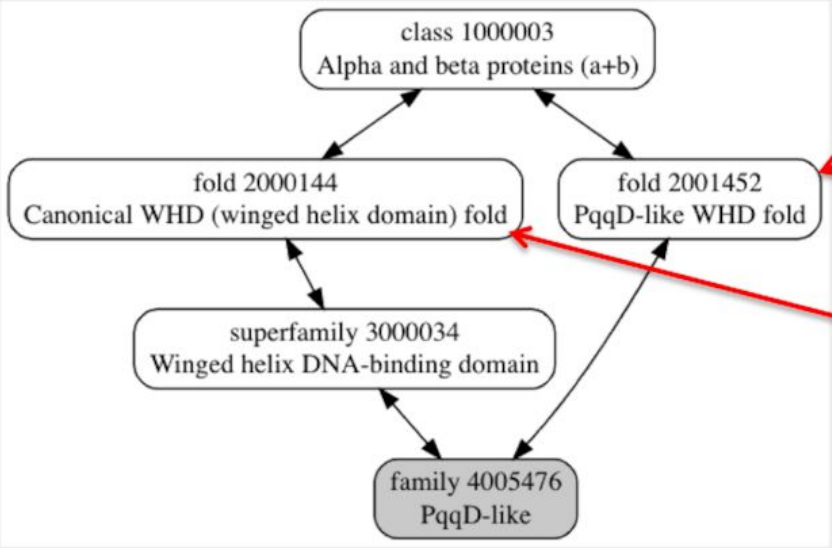
PqqD-like

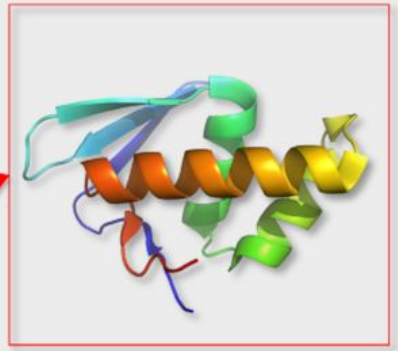
SCOP ID: 4005476

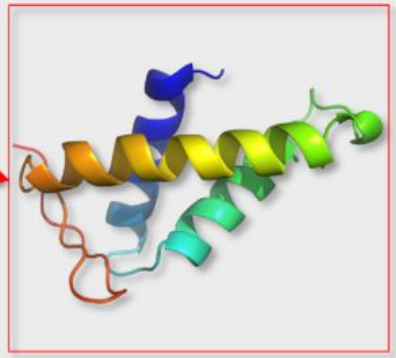
PF05402

Function known the process in which is implicated

Show ancestry







CATH: уровни классификации

Class - structures are classified according to their secondary structure composition (mostly alpha, mostly beta, mixed alpha/beta or few secondary structures).

Architecture - structures are classified according to their overall shape as determined by the orientations of the secondary structures in 3D space but ignores the connectivity between them.

Topology (fold family) - structures are grouped into fold groups at this level depending on both the overall shape and connectivity of the secondary structures.

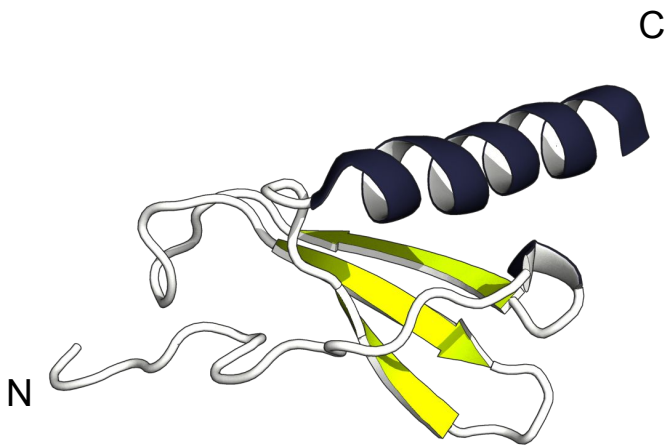
Homologous superfamily - this level groups together protein domains which are thought to share a common ancestor and can therefore be described as homologous.

Class	Architecture	Topology	Homology
1 Mainly α	10 Non-bundle	490 Globin-like	10 1hlm
2 Mainly β	20 Bundle		20 1cpc chain A
3 α–β	30 Few SS		30 1col chain A
4 Few SS			40 1ddt domain 2

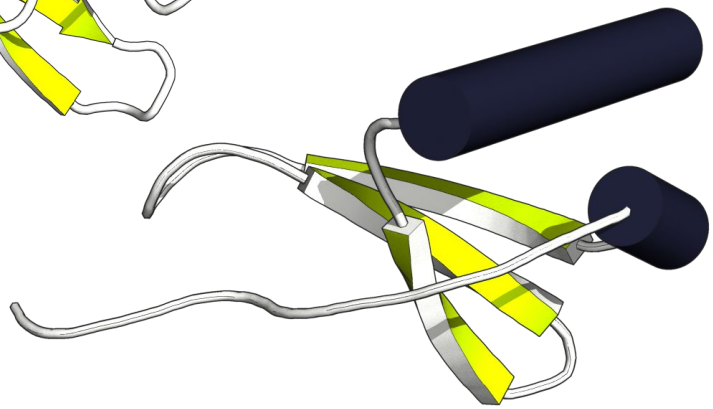
460 ...
470 Variant surface glycoprotein...
480 Glucoamylase, domain 2
500 βlactamase, domain 2
510 Casein kinase δ ..
520 ...

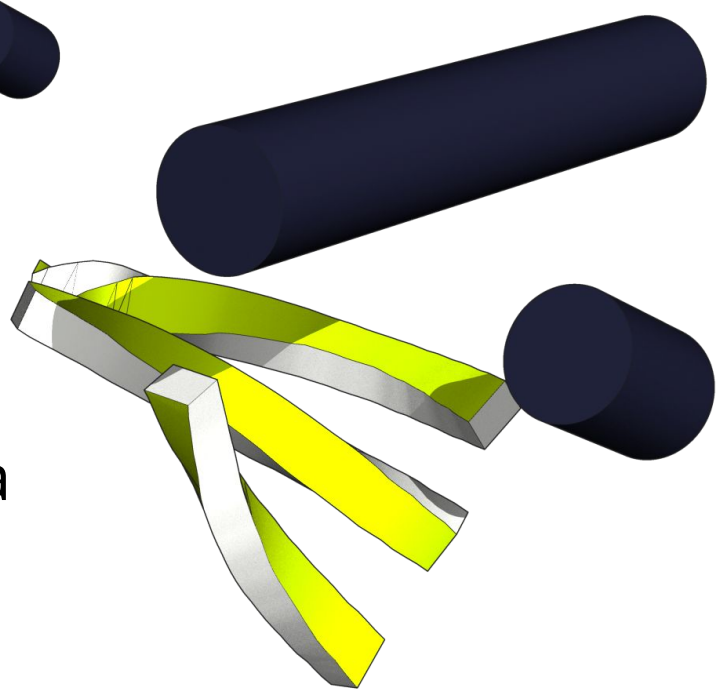
1.10.490.20

Mainly α.Non-bundle.Globin-like.1cpc chain A

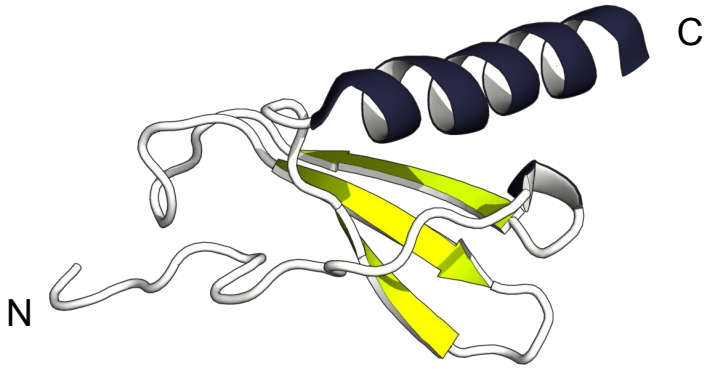


Топология

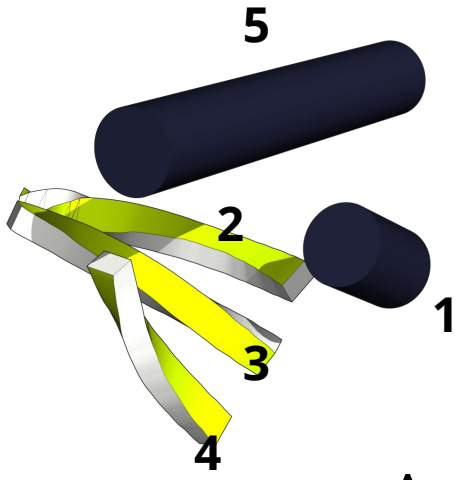




Архитектура



Топология



Архитектура

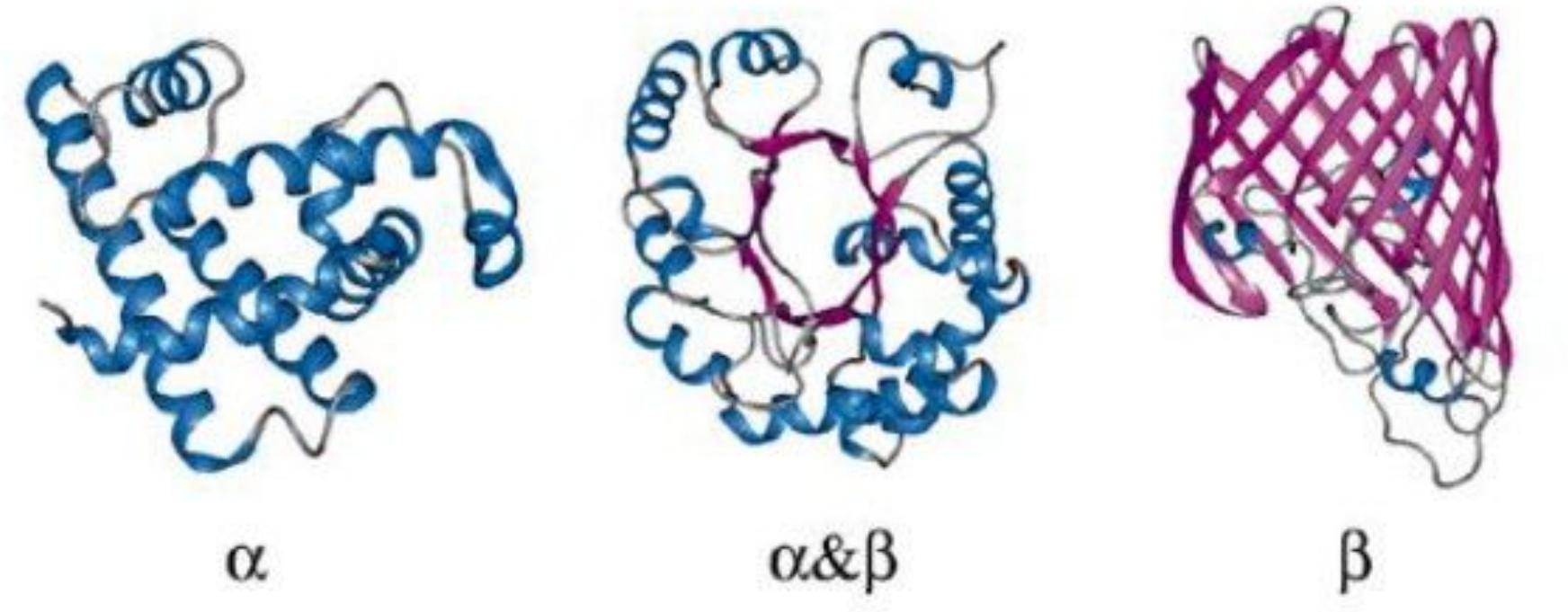
Данная топология: 1-2-3-4-5
Однако мы вполне можем представить себе случай, когда она будет 1-4-3-2-5

CATH: протокол

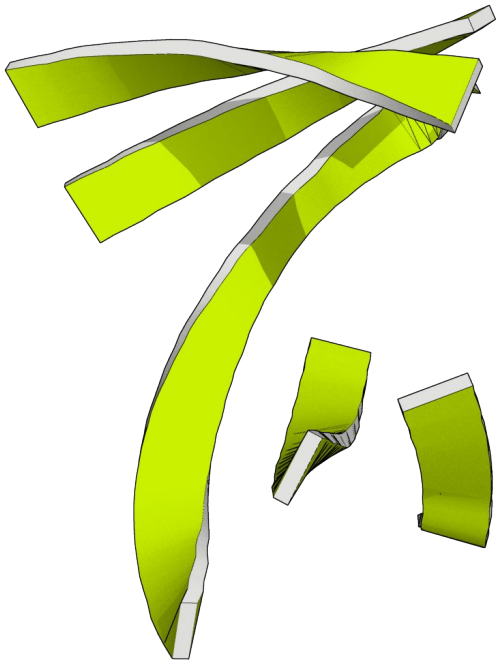
1. S-level. Группировка по последовательностям (non-redundancy)
2. Выделение доменов - консенсус по DOMAK, DETECTIVE, PUU. Если консенсус 85% и больше, то присваивается разбиение по DETECTIVE. В противном случае вручную
3. C-level. Подсчет содержания вторичных структур
4. H- и T-level. Структурное выравнивание SSAP. 70% структурного сходства - один T-level. 80% - один H-level. H-levels аннотируются по функции (по возможности)
5. A-level. Вручную

Class

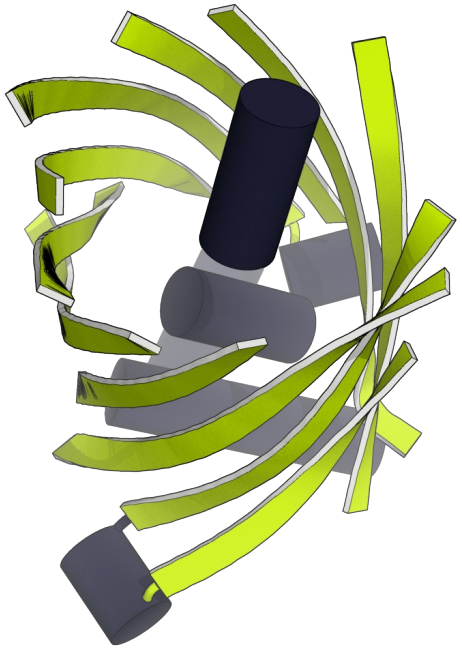
- Преимущественно альфа-спиральные
- Преимущественно бета-листовые
- Смешанные
- Неструктурированные



Разнообразие фолдов: преимущественно β

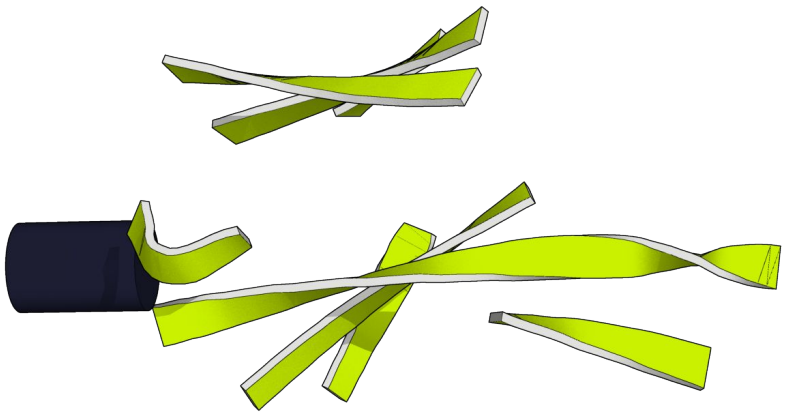


β-сверток
β-roll

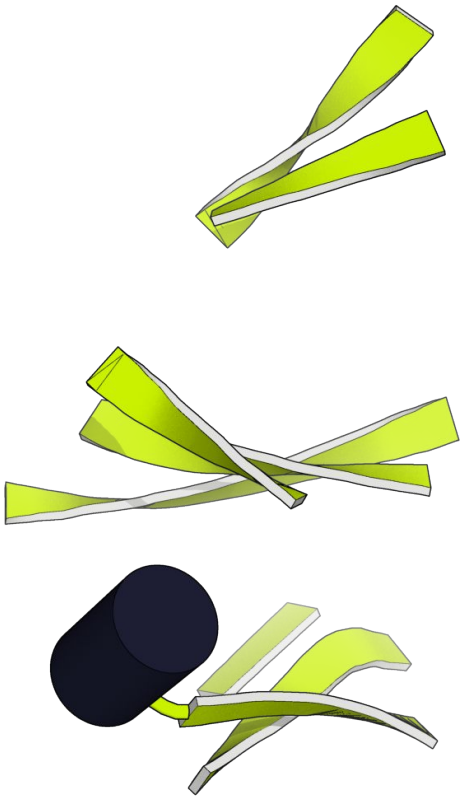


β-бочка
β-barrel

Разнообразие фолдов: преимущественно β

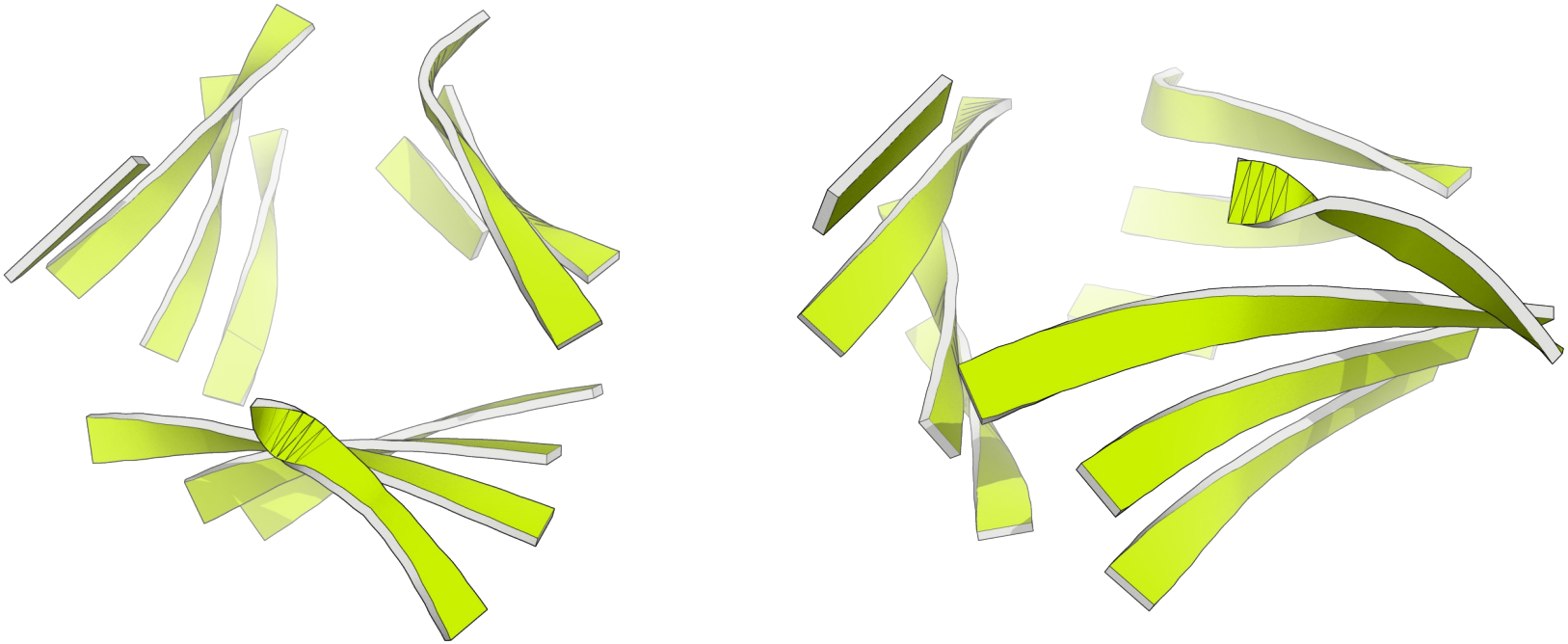


β-сендвич
β-sandwich



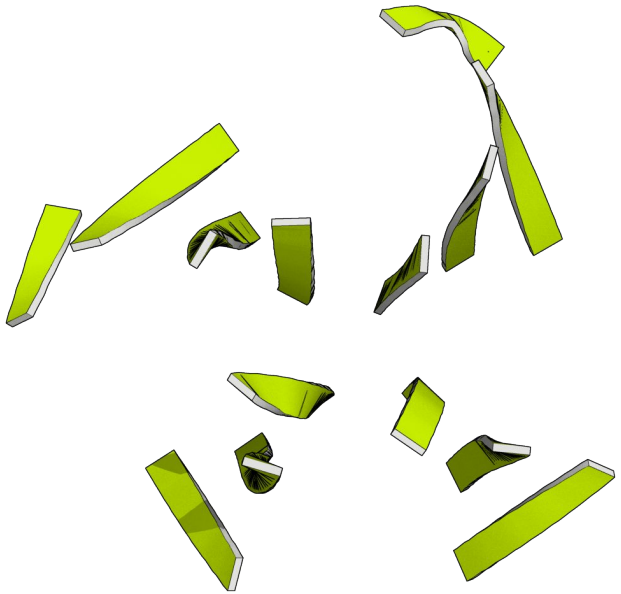
Тройной β-сендвич
3-layer sandwich

Разнообразие фолдов: преимущественно β

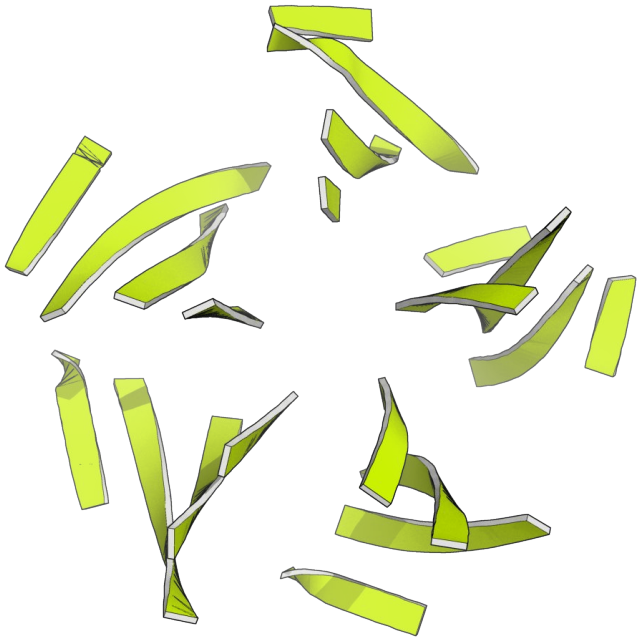


β-призма
β-prism

Разнообразие фолдов: преимущественно β

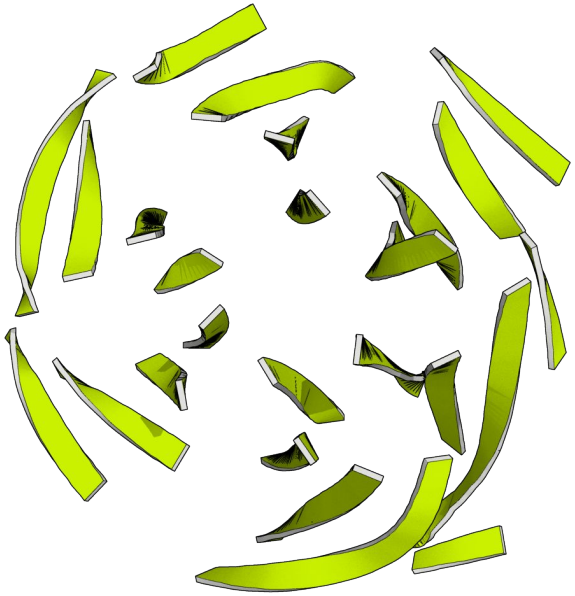


4-пропеллер
4-propeller



5-пропеллер
5-propeller

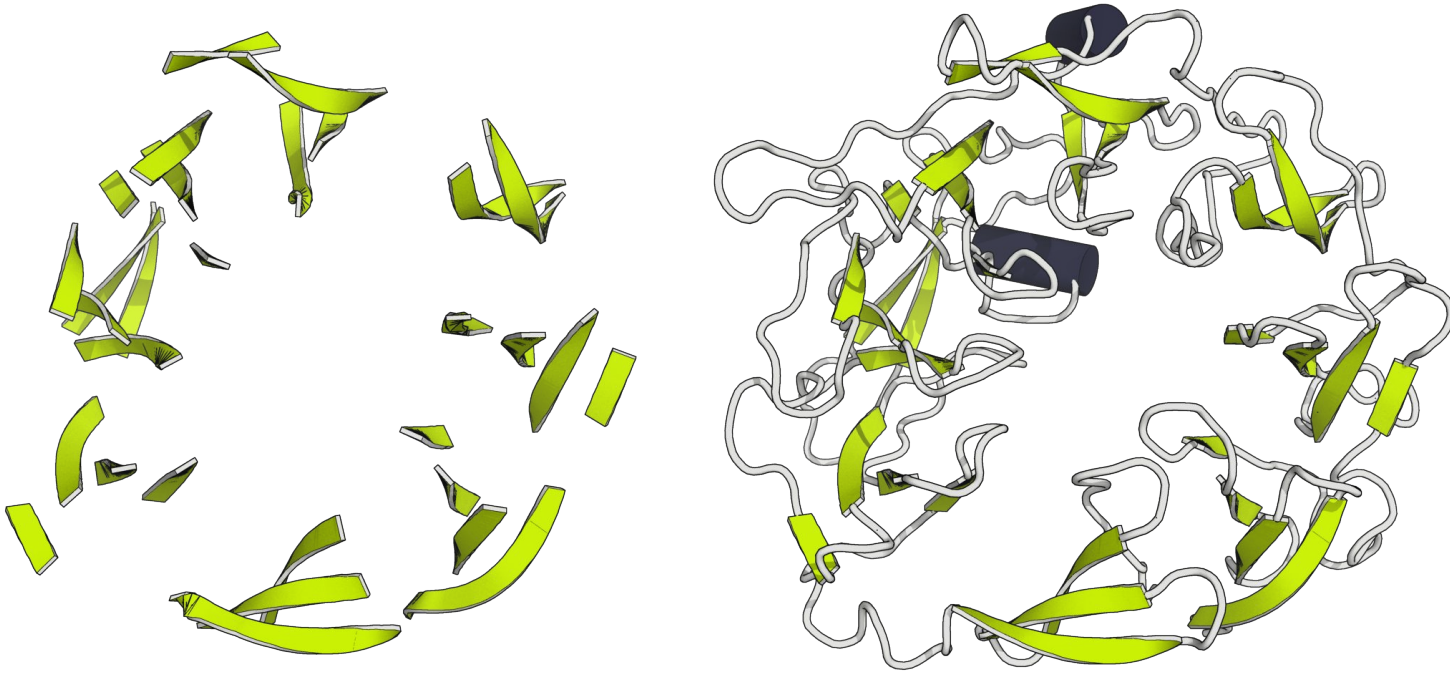
Разнообразие фолдов: преимущественно β



6-пропеллер
6-propeller

7-пропеллер
7-propeller

Разнообразие фолдов: преимущественно β



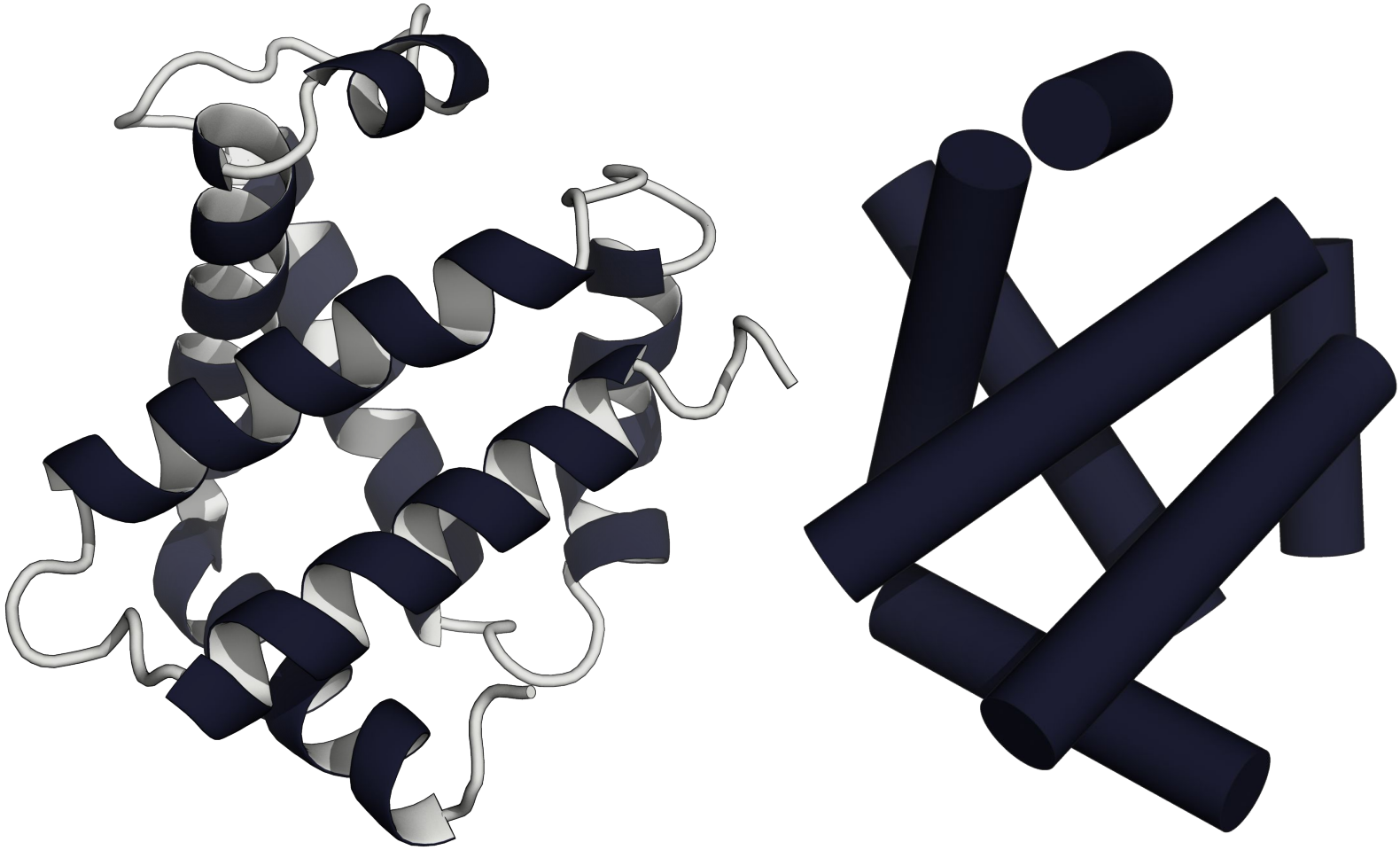
8-пропеллер
8-propeller

α-фолды: α-пучки

alpha-bundles

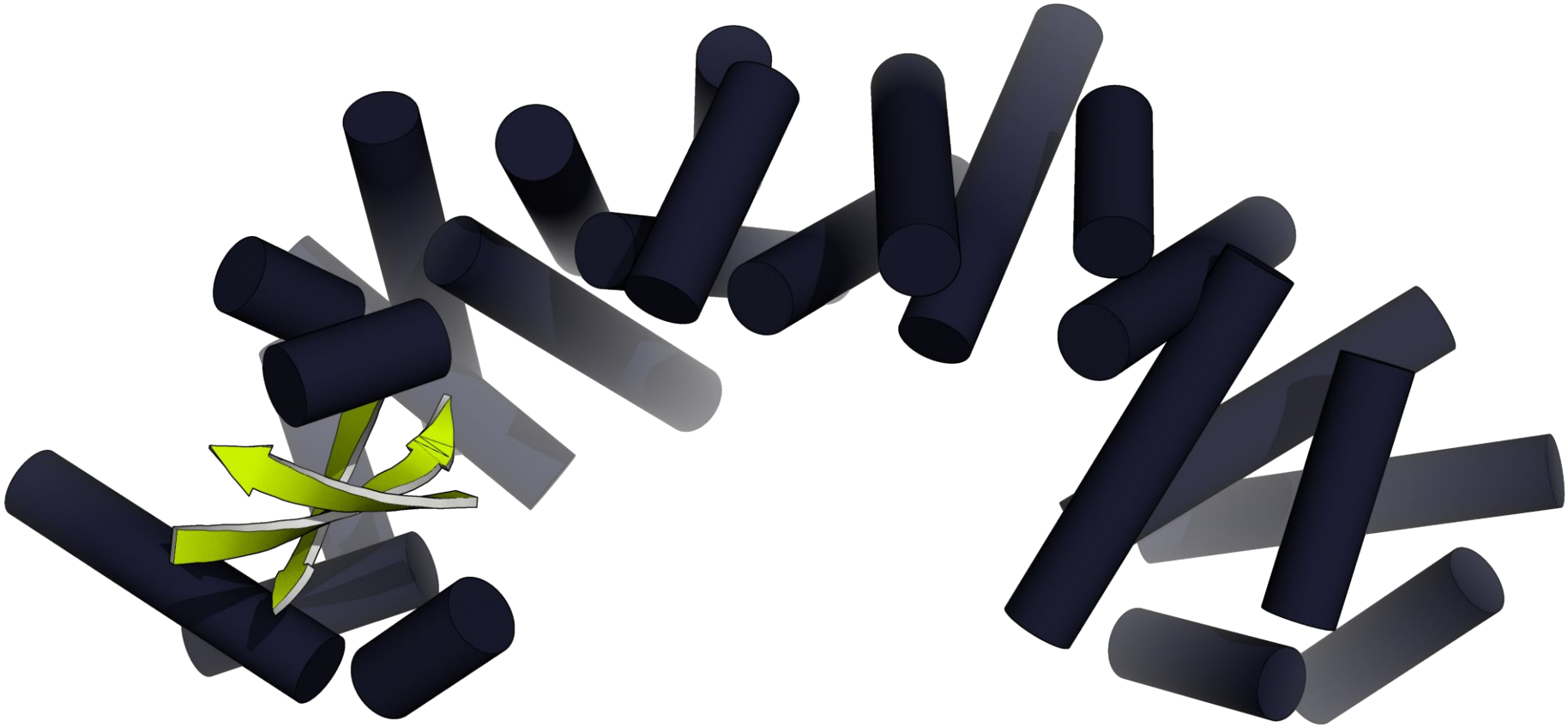
α-фолды: α-многогранники

Alpha solenoid



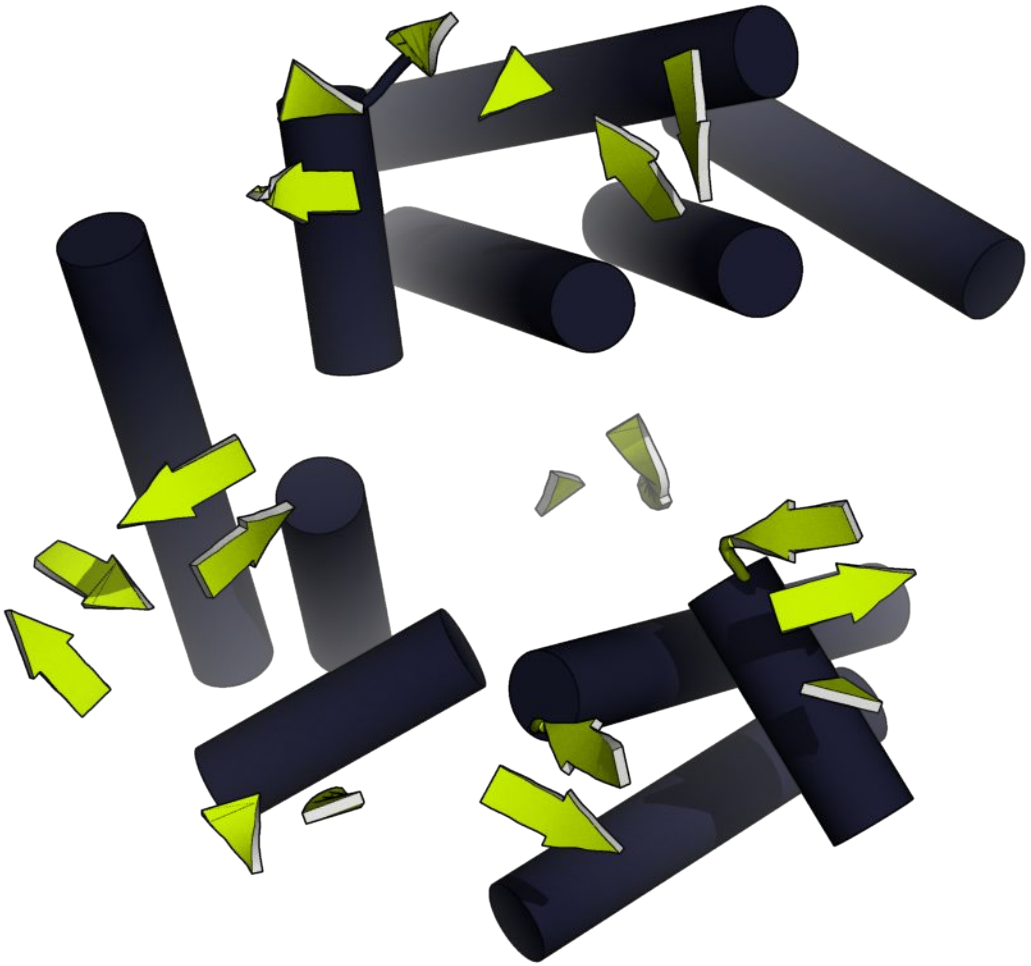
α-фолды: α-подковы

Alpha horseshoe

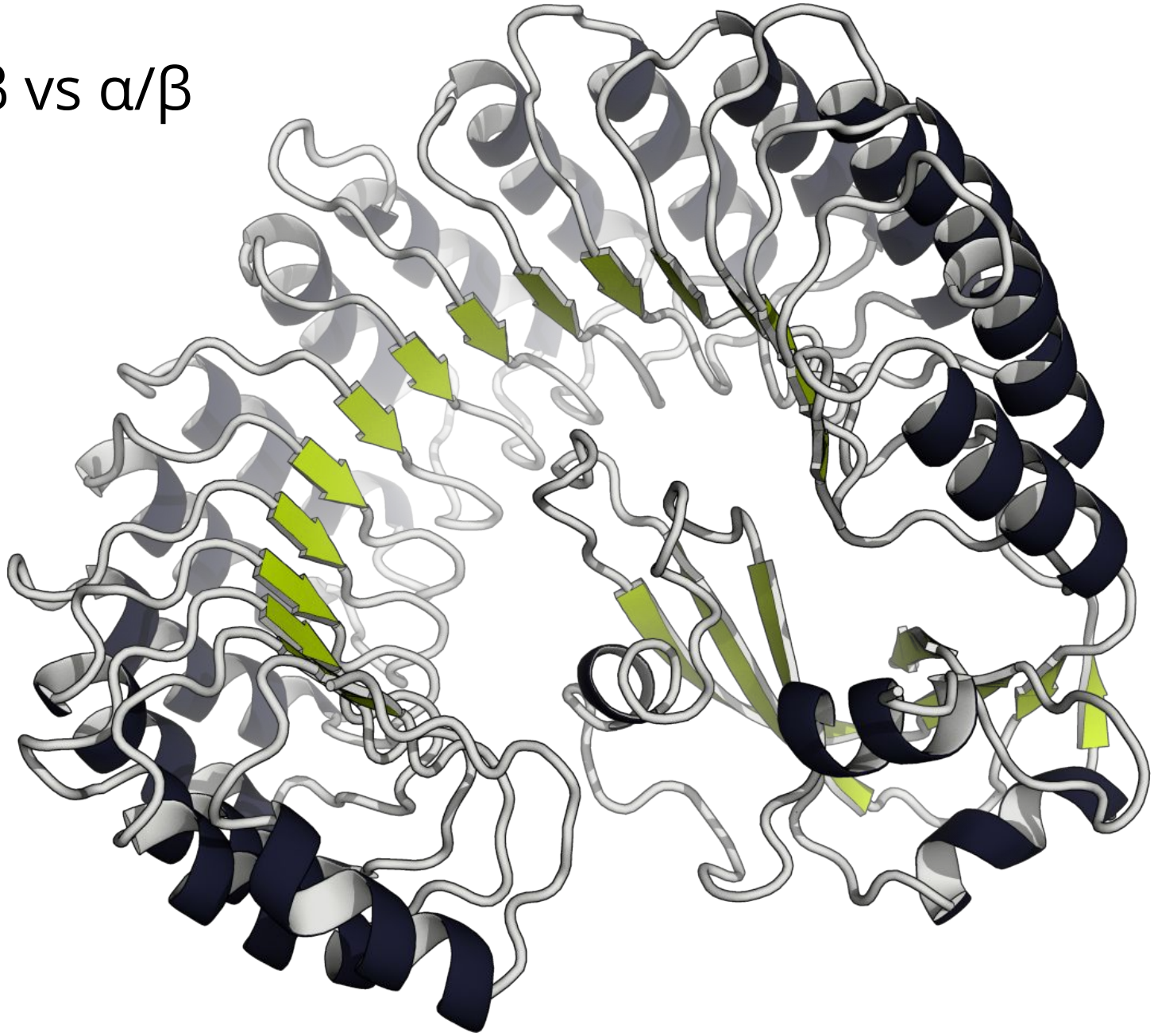


α-фолды: α-бочонки

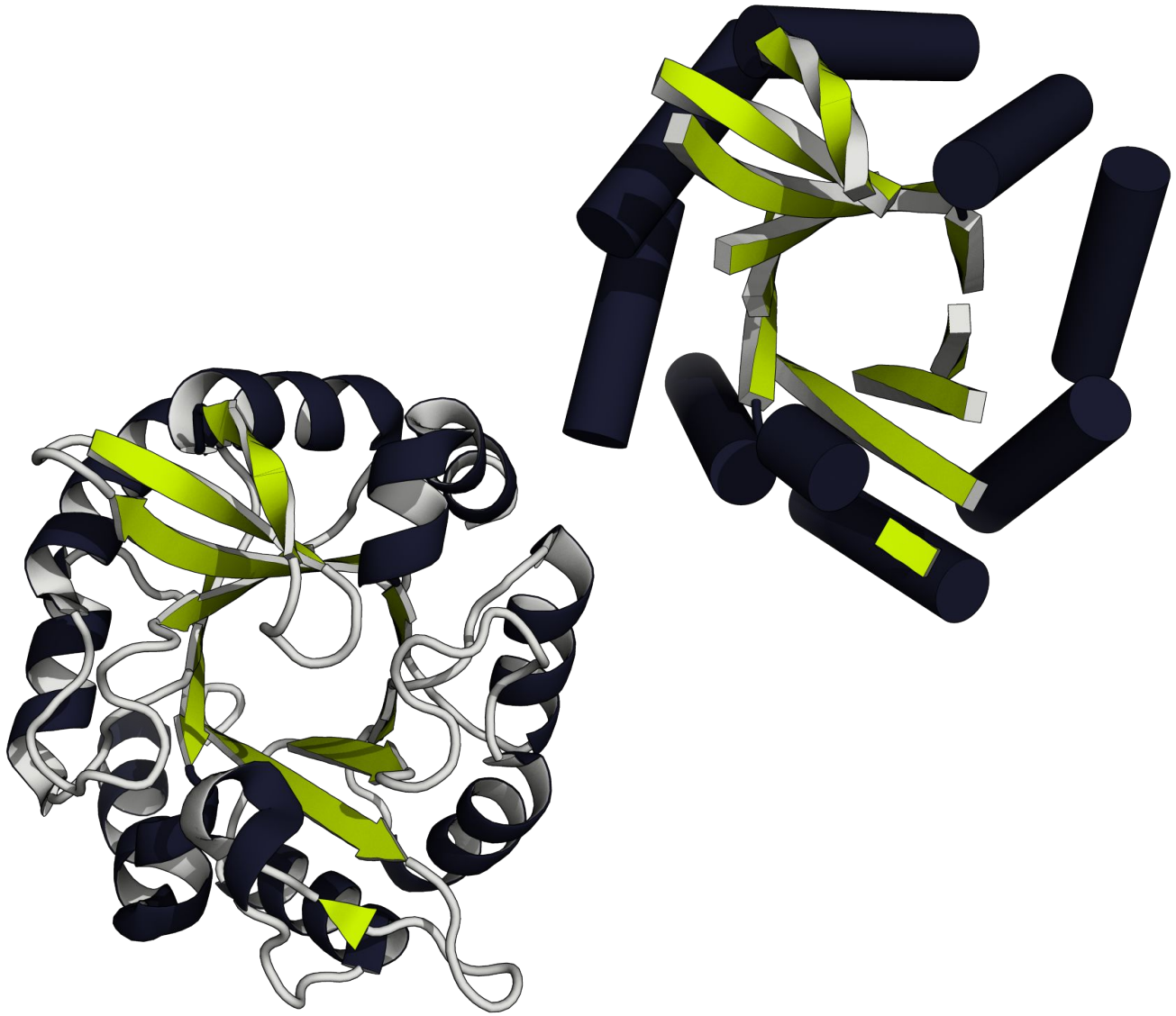
Alpha-barrels



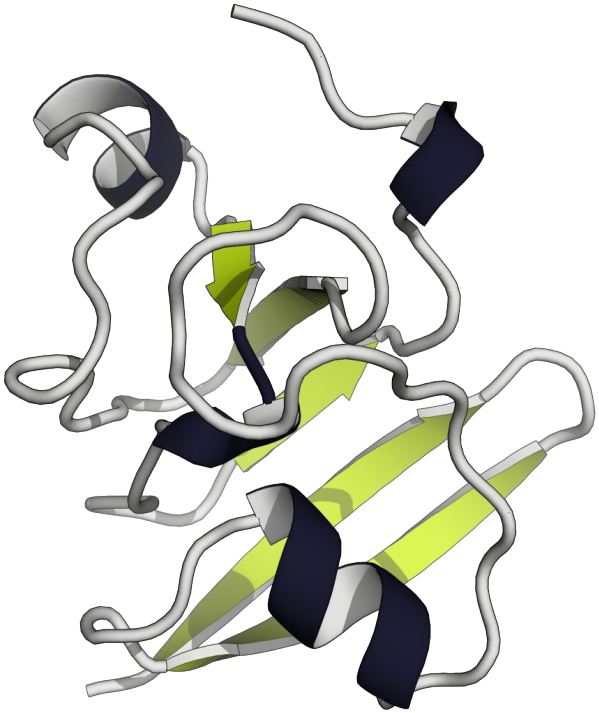
α+β vs α/β

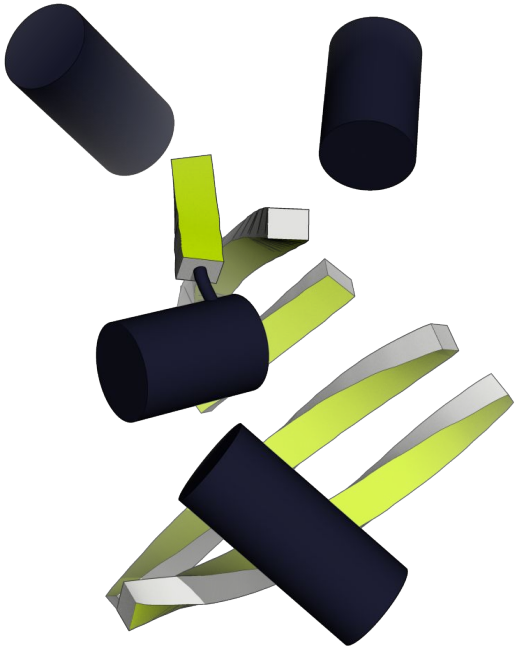


α/β

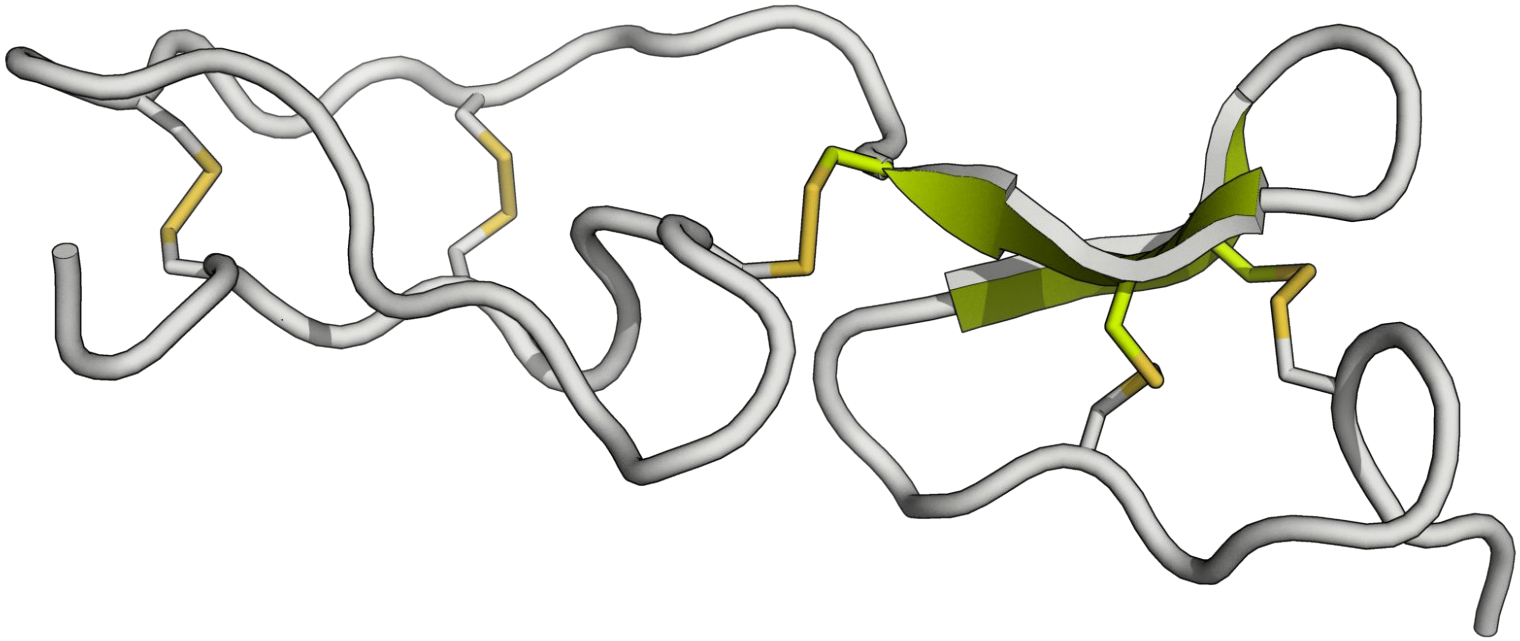


α+β





Глобулярные белки с нестандартной укладкой



Не защищено | www.cathdb.info/browse/tree

CATH Home Search Browse Download About Support Search CATH by keywords or ID

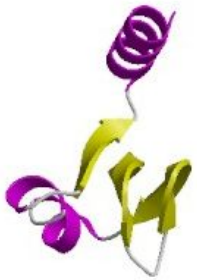
Superfamily Comparison

Select a CATH node...

A 3-Layer(aba) Sandwich

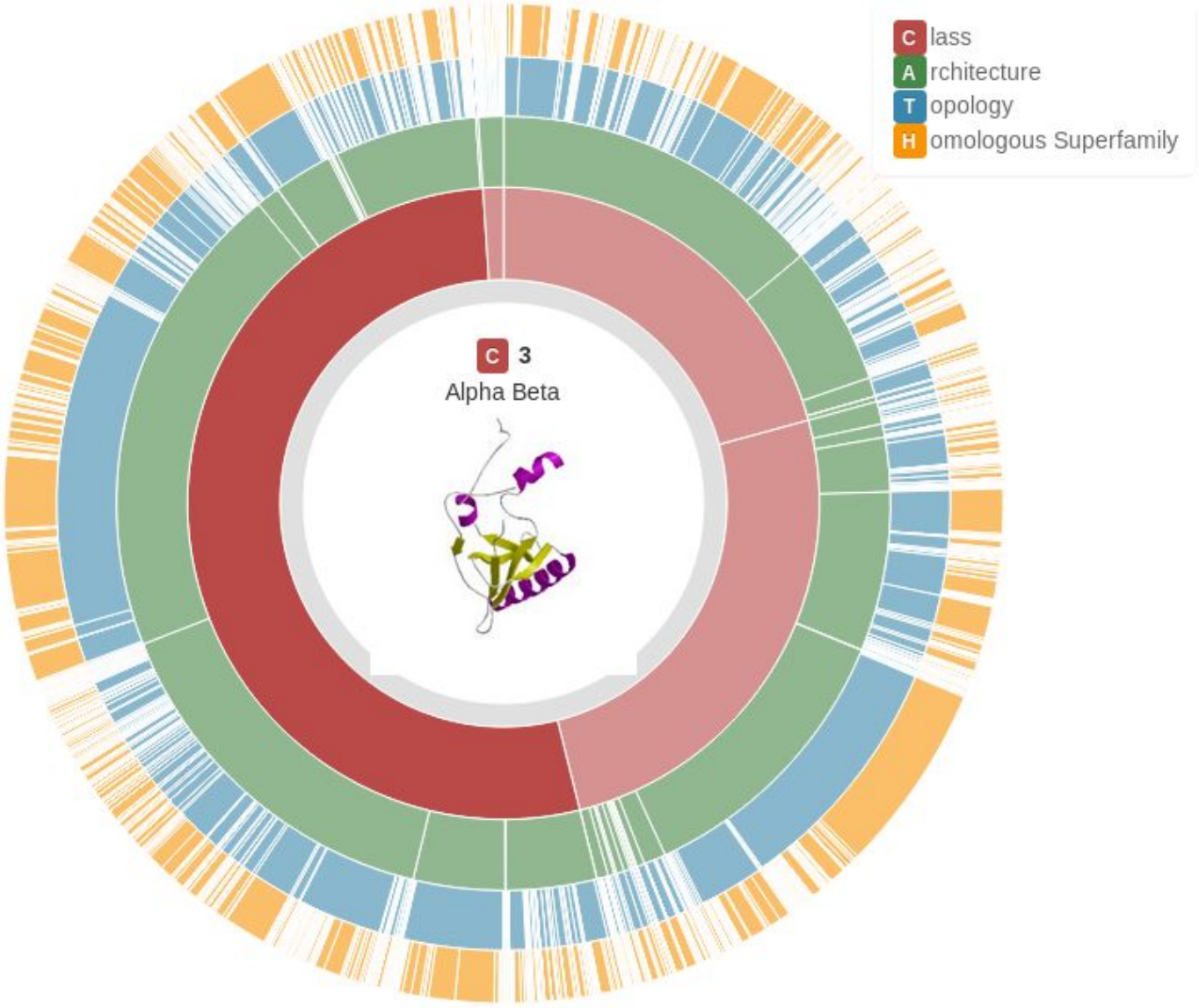
3.40

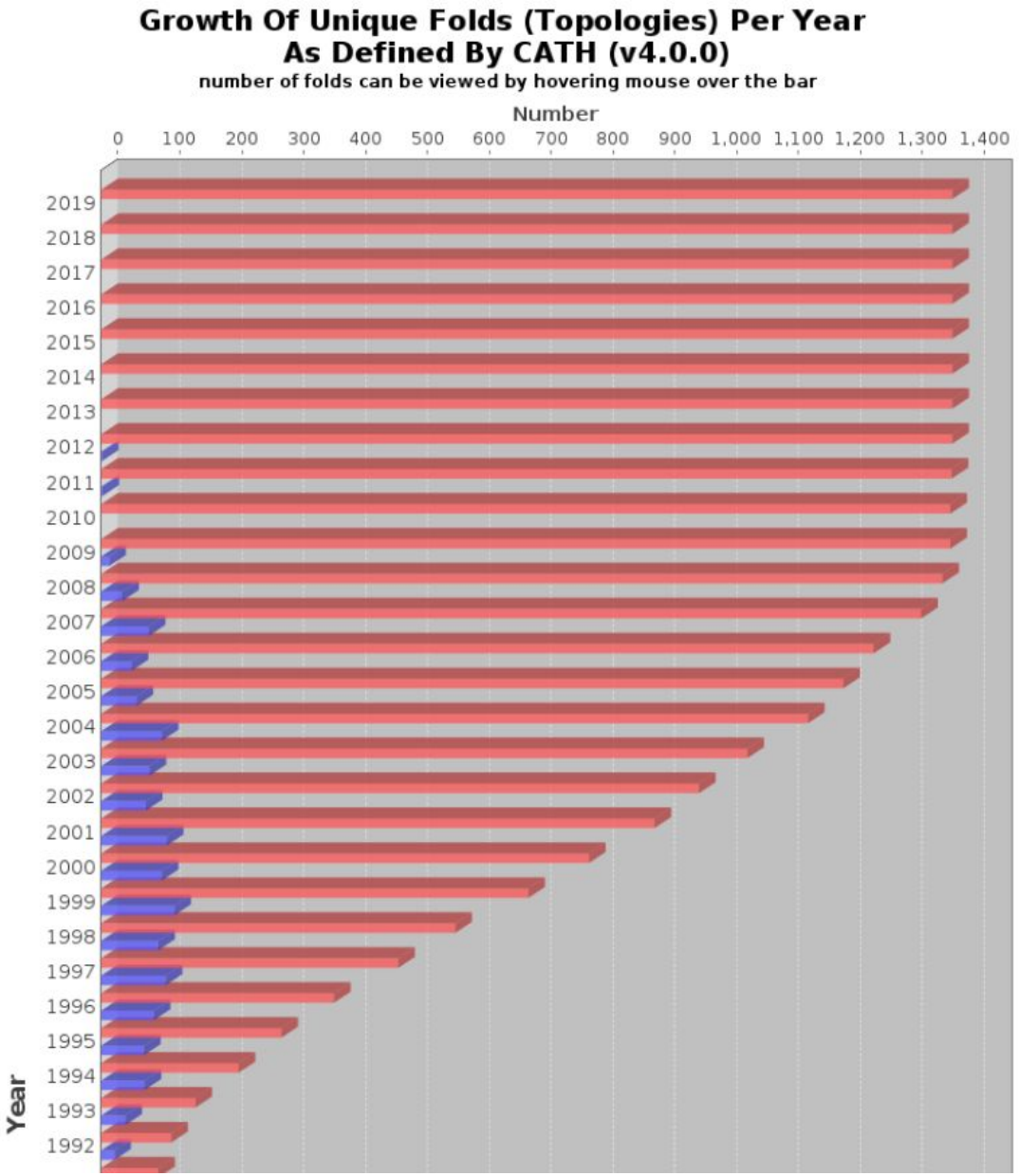
CATH ID	3.40
Topologies	126
Superfamilies	577
Domains	86984
Example Domain	2hbaA00 [PDB]



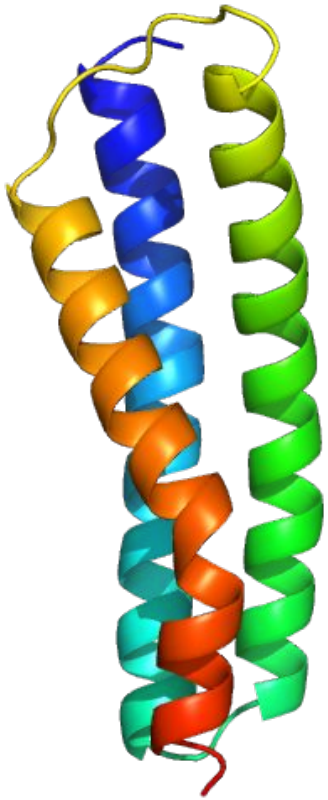
Top of CATH Hierarchy (4 Classes)

- C 1 Mainly Alpha — *5 Architectures, 405 Folds, 2174 Superfamilies, 90302 Domains*
- C 2 Mainly Beta — *21 Architectures, 244 Folds, 1395 Superfamilies, 110260 Domains*
- C 3 Alpha Beta — *14 Architectures, 634 Folds, 2428 Superfamilies, 229776 Domains*
  - A 3.10 Roll — *60 Folds, 248 Superfamilies, 16507 Domains*
  - A 3.15 Super Roll — *3 Folds, 7 Superfamilies, 56 Domains*
  - A 3.20 Alpha-Beta Barrel — *18 Folds, 63 Superfamilies, 16668 Domains*
  - A 3.30 2-Layer Sandwich — *224 Folds, 1178 Superfamilies, 66583 Domains*
  - A 3.40 3-Layer(aba) Sandwich — *126 Folds, 577 Superfamilies, 86984 Domains*
    - T 3.40.5 Ribosomal Protein L9; domain 1 — *10 Superfamilies, 292 Domains*
    - T 3.40.10 DNA Methylphosphotriester Repair Domain — *1 Superfamilies, 5 Domains*
    - T 3.40.20 Severin — *2 Superfamilies, 290 Domains*
    - T 3.40.30 Glutaredoxin — *12 Superfamilies, 4013 Domains*
    - T 3.40.33 Pathogenesis-related Protein p14a — *1 Superfamilies, 62 Domains*
    - T 3.40.35 Fructose Permease — *1 Superfamilies, 15 Domains*
    - T 3.40.47 Peroxisomal Thiolase; Chain A, domain 1 — *2 Superfamilies, 1877 Domains*
    - T 3.40.50 Rossmann fold — *298 Superfamilies, 52880 Domains*
    - T 3.40.80 Lysozyme-like — *1 Superfamilies, 257 Domains*
    - T 3.40.91 Restriction Endonuclease — *9 Superfamilies, 265 Domains*
    - T 3.40.109 NADH Oxidase — *3 Superfamilies, 288 Domains*
    - T 3.40.120 Alpha-D-Glucose-1,6-Bisphosphate; Chain A, domain 3 — *1 Superfamilies, 221 Domains*
    - T 3.40.140 Cytidine Deaminase; domain 2 — *7 Superfamilies, 420 Domains*
    - T 3.40.190 D-Maltodextrin-Binding Protein; domain 2 — *18 Superfamilies, 5295 Domains*
    - T 3.40.198 Delta-endotoxin CytB — *1 Superfamilies, 16 Domains*
    - T 3.40.210 PvuII Endonuclease; Chain A — *3 Superfamilies, 35 Domains*
    - T 3.40.220 Leucine Aminopeptidase, subunit E; domain 1 — *3 Superfamilies, 484 Domains*
    - T 3.40.225 L-fuculose-1-phosphate Aldolase — *1 Superfamilies, 110 Domains*
    - T 3.40.228 Dimethylsulfoxide Reductase; domain 2 — *1 Superfamilies, 104 Domains*

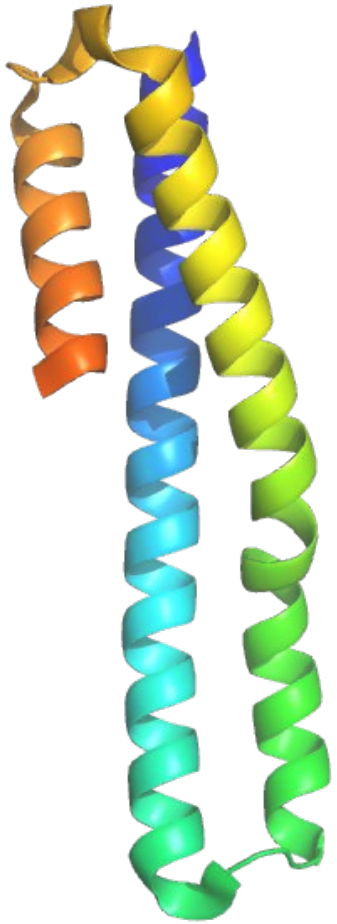

Class
Architecture
Topology
Homologous Superfamily
C 3
Alpha Beta


Growth Of Unique Folds (Topologies) Per Year
As Defined By CATH (v4.0.0)
number of folds can be viewed by hovering mouse over the bar
Number
0 100 200 300 400 500 600 700 800 900 1,000 1,100 1,200 1,300 1,400
Year
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
2006
2005
2004
2003
2002
2001
2000
1999
1998
1997
1996
1995
1994
1993
1992

Разные белки - одинаковые укладки

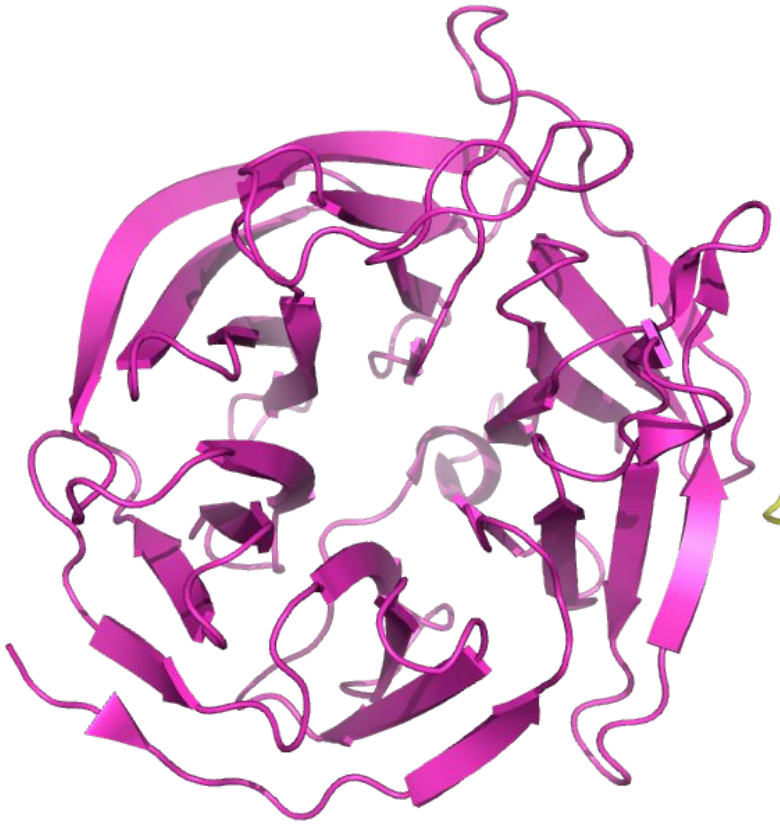


Лактаза бактерии

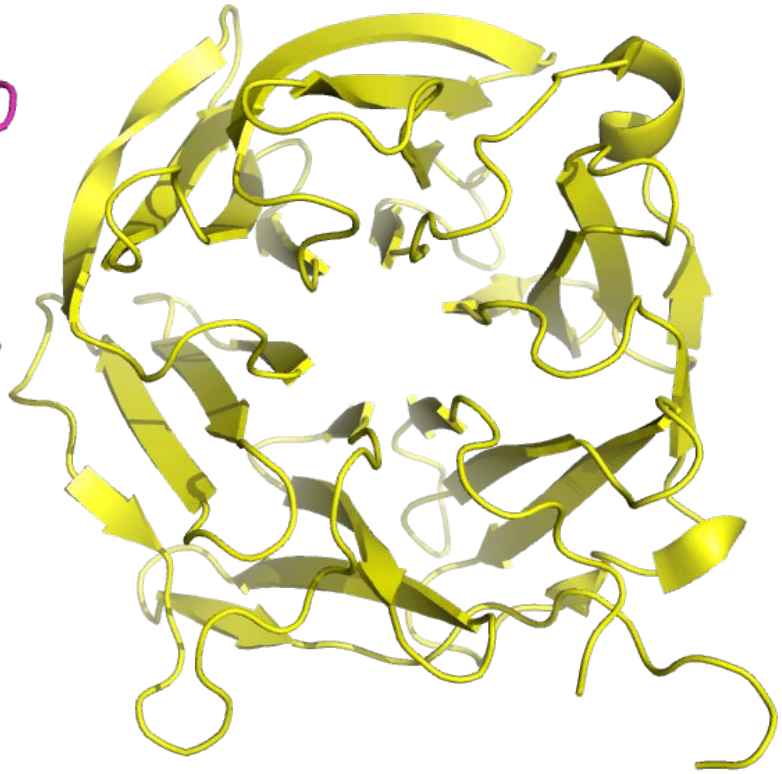


Шаперон человека

Разные белки - одинаковые укладки



DFP-аза каракатицы

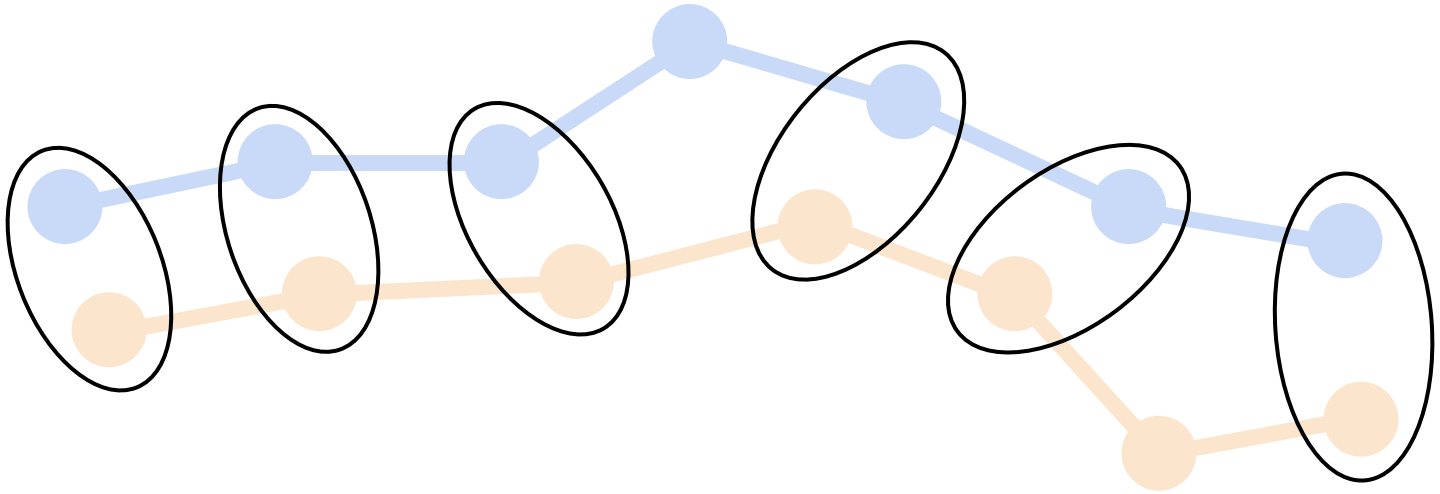


Лектин гриба

Структурное выравнивание/ пространственное совмещение

Структурное выравнивание – сопоставление пар остатков исходя из структурных критериев

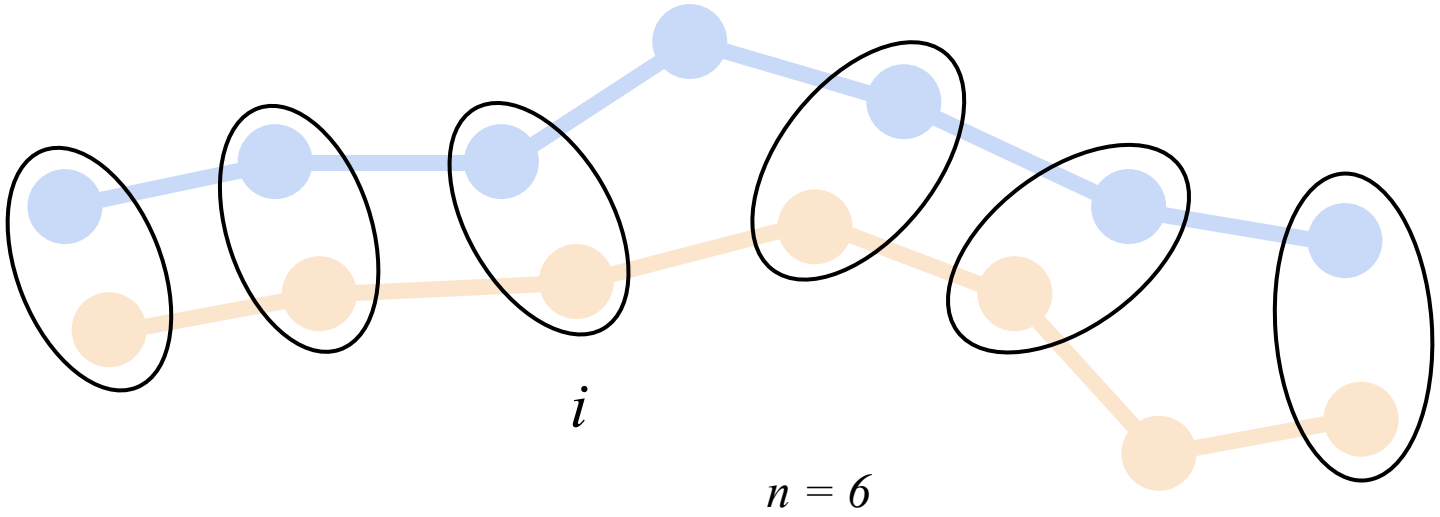
Пространственное совмещение – трансформация координат, приводящая к наилучшему наложению сопоставленных атомов (чаще всего Calpha)



Способы измерить качество наложения: RMSD

$$\mathrm{RMSD}(\mathbf{v}, \mathbf{w}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|v_i - w_i\|^2} \tag{1}$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n} ((v_{ix} - w_{ix})^2 + (v_{iy} - w_{iy})^2 + (v_{iz} - w_{iz})^2)} \tag{2}$$

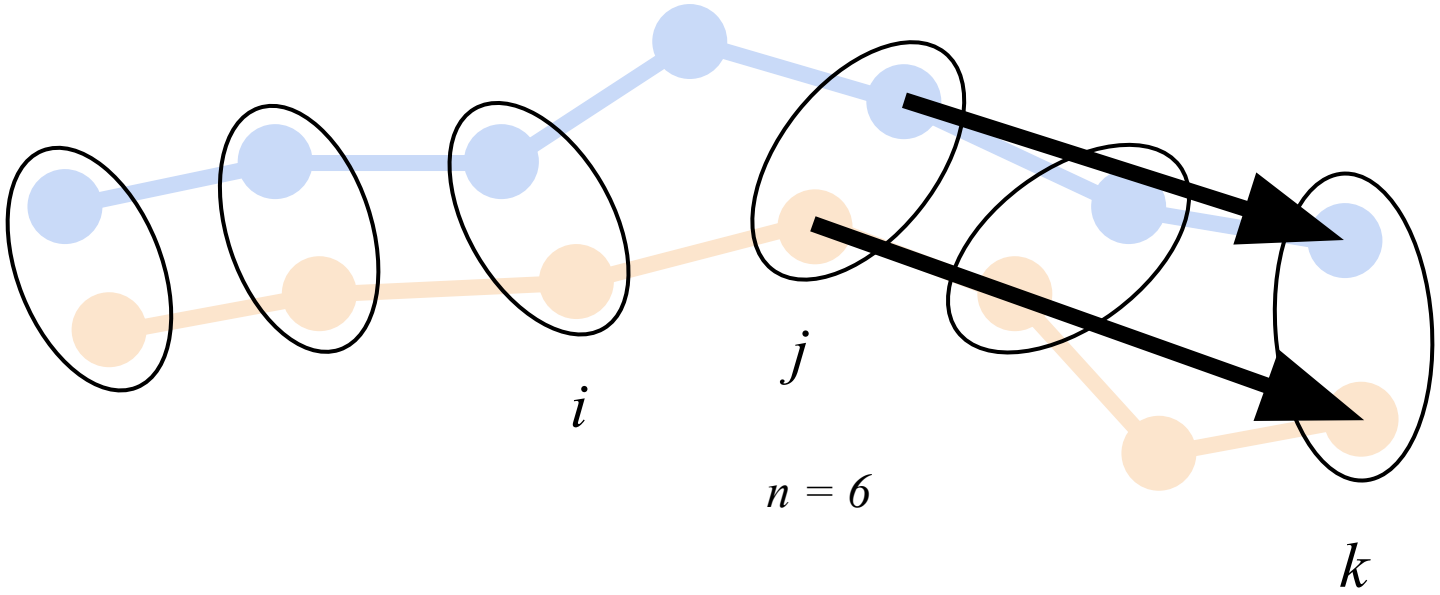


Способы измерить качество наложения: eRMSD

Идея:

- Описать каждую пару оснований внутри структуры как вектор контакта
  - Вектор контакта - вектор между центрами масс оснований
- Оценить сходство структур как разницу векторов

$$eRMSD = \sqrt{\frac{1}{N}\sum_{j,k} |\vec{G}(\tilde{\vec{r}}^{\alpha}_{jk}) - \vec{G}(\tilde{\vec{r}}^{\beta}_{jk})|^2}$$



Способы измерить качество наложения: TM-score

Идея:

- Избавиться от влияния длины

$$\text{TM-score} = \max\left[\frac{1}{L_{\text{target}}}\sum_{i}^{L_{\text{aligned}}}\frac{1}{1+\left(\frac{d_i}{d_0(L_{\text{target}})}\right)^2}\right]$$

$$d_0(L_{\text{target}}) = 1.24\sqrt[3]{L_{\text{target}}-15} - 1.8$$

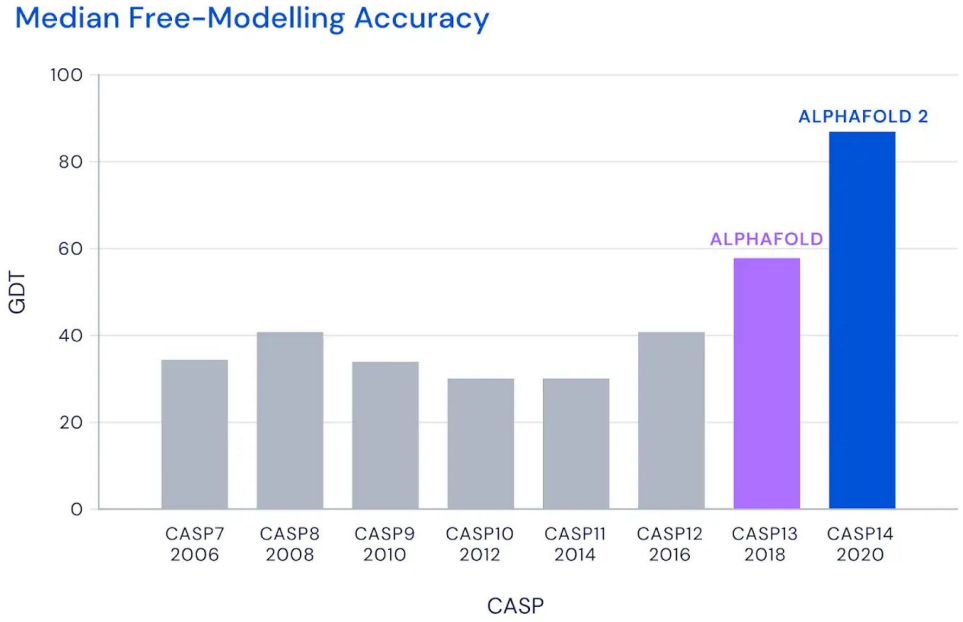
Способы измерить качество наложения: GDT_TS

Идея:

- Оценивать наложение по похожим участкам, не учитывать непохожие
- Применяется в оценке качества моделирования структур по последовательностям в соревновании CASP

$$GDT\_TS = N(\text{пар } i \text{ для которых } d(i^A, i^B) < threshold)$$

где $d(i^A, i^B)$ это расстояние между Calpha атомами



Как получить структурное выравнивание?

- Не получать, взять выравнивание по последовательностям (PyMol align)
- Сделать вручную (бенчмарки для инструментов)
- Сделать алгоритмически

Для написания алгоритма нам нужно определиться со способами

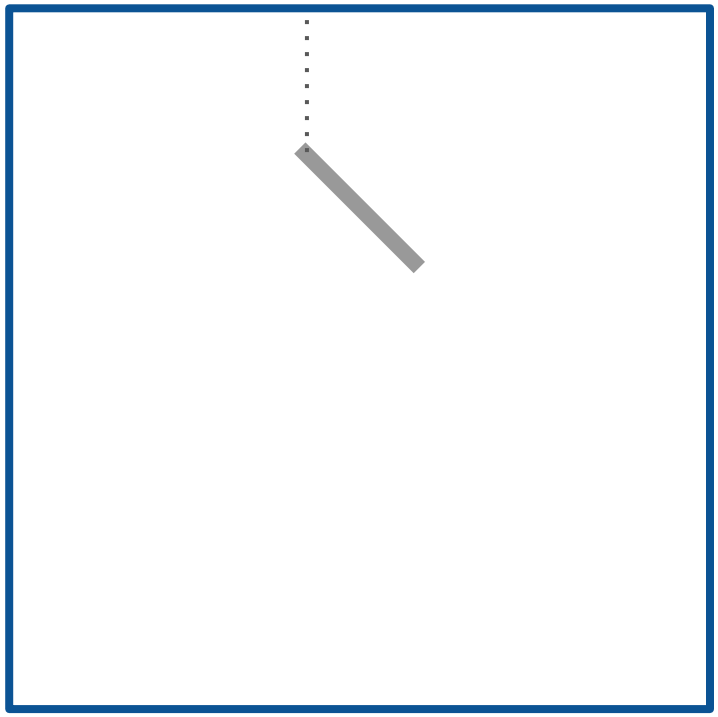
- описать локальную геометрию участка белка
- решить, похожи ли геометрии двух участков
- найти по возможности наилучшее объединение похожих участков

AFP (aligned fragment pairs)

Очень распространенный концепт в алгоритмах структурного выравнивания. AFP это пара подструктур из последовательно идущих остатков белков A и B небольшой фиксированной длины (часто L=8).

Идея:

- Рассмотреть все AFP, отфильтровать по похожести их локальных геометрий
- Из оставшихся выбрать “затравку”
- Достраивать “затравку”, стыкуя ее с новыми AFP
  Комбинаторно, динамически или Монте-Карло
- Преобразовать цепочку AFP в черновое выравнивание
- Оптимизировать выравнивание
  Чтобы добиться наилучшего RMSD



CE (Combinatorial Extension)

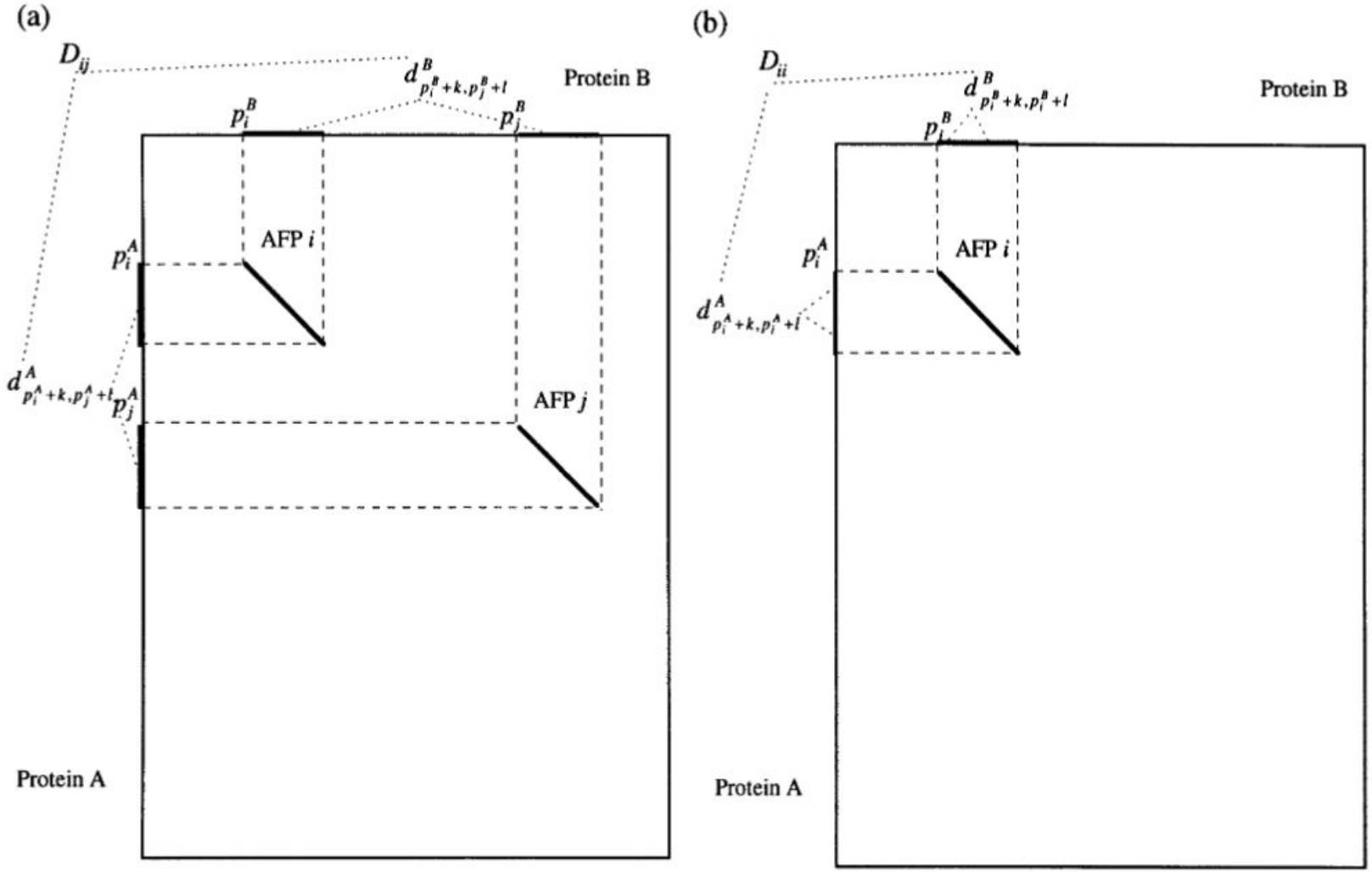
Мера сходства геометрий – функции над попарными расстояниями
Способ продления цепочки – переборный с отбраковкой согласно эвристикам

1. **Выбор затравки.**
   Для всех октамеров из белка A, для всех октамеров из белка B попарно определяются AFP. Для *i*-го AFP:

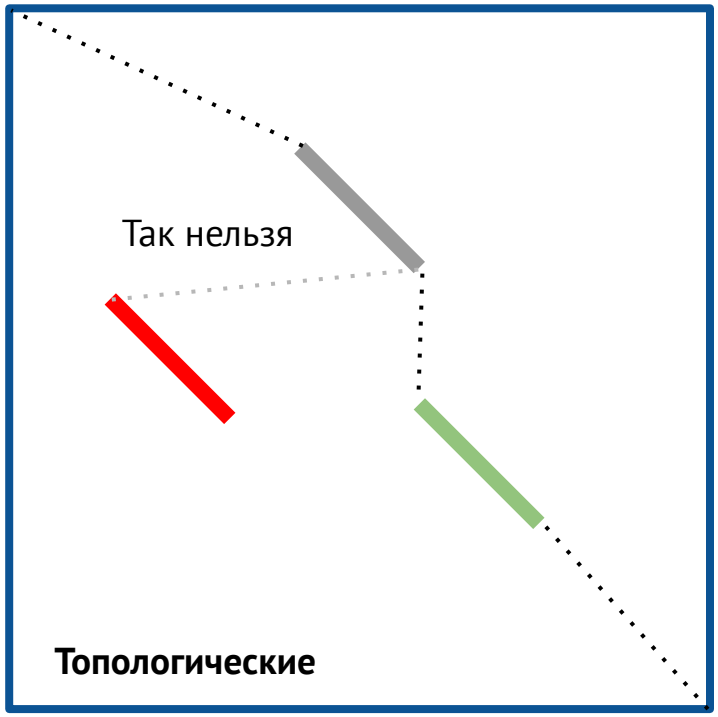
$$D_i = \frac{1}{m^2} \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} \left| d_{p_i^A+k, p_i^A+l} - d_{p_i^B+k, p_i^B+l} \right|$$

Где m=8, p_i^A номер остатка белка A, с которого начинается октамер AFP$_i$ в A, аналогично p_i^B
Затравки принимаются, если $D_i < 3A$

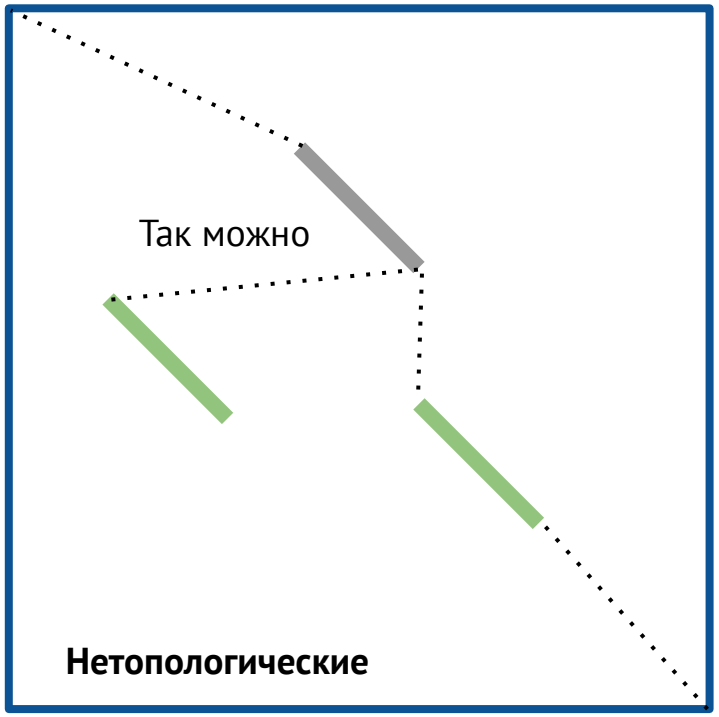
CE (Combinatorial Extension)



Топологические и нетопологические выравнивания

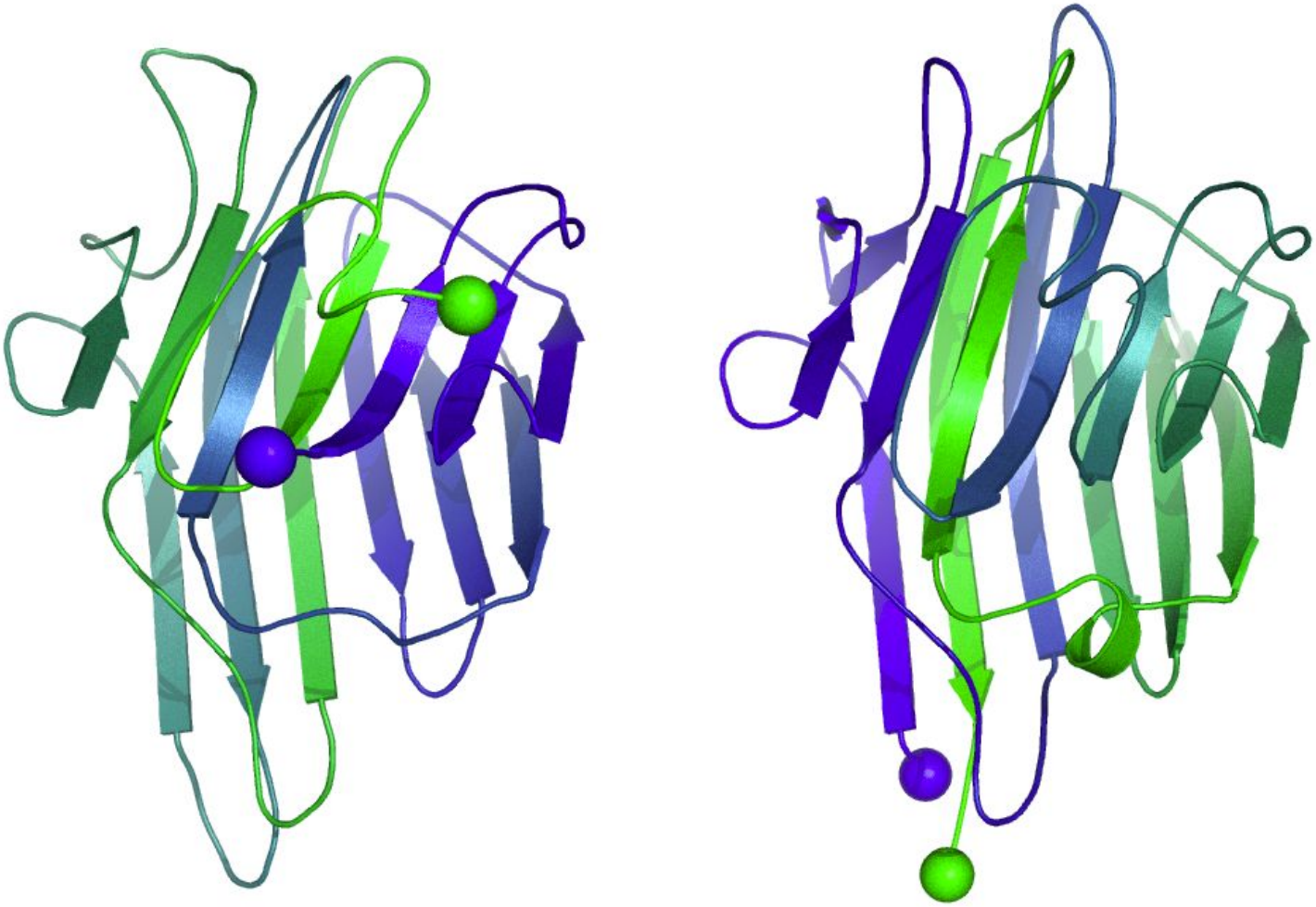


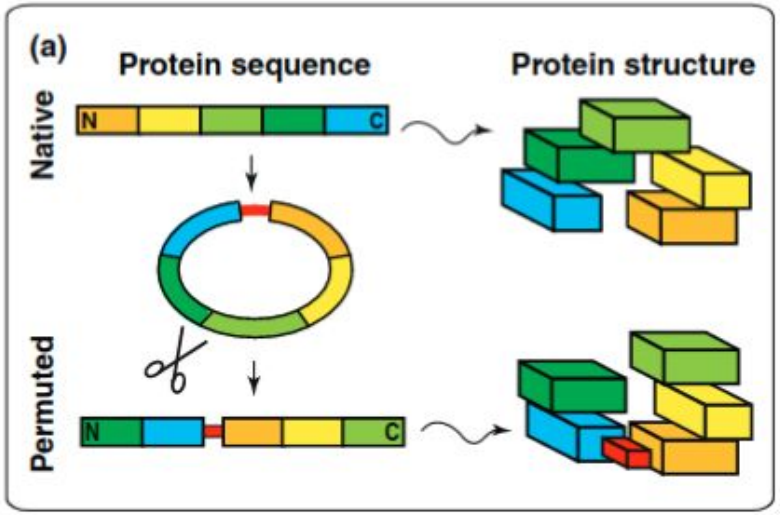
CE
MATT
FatCat

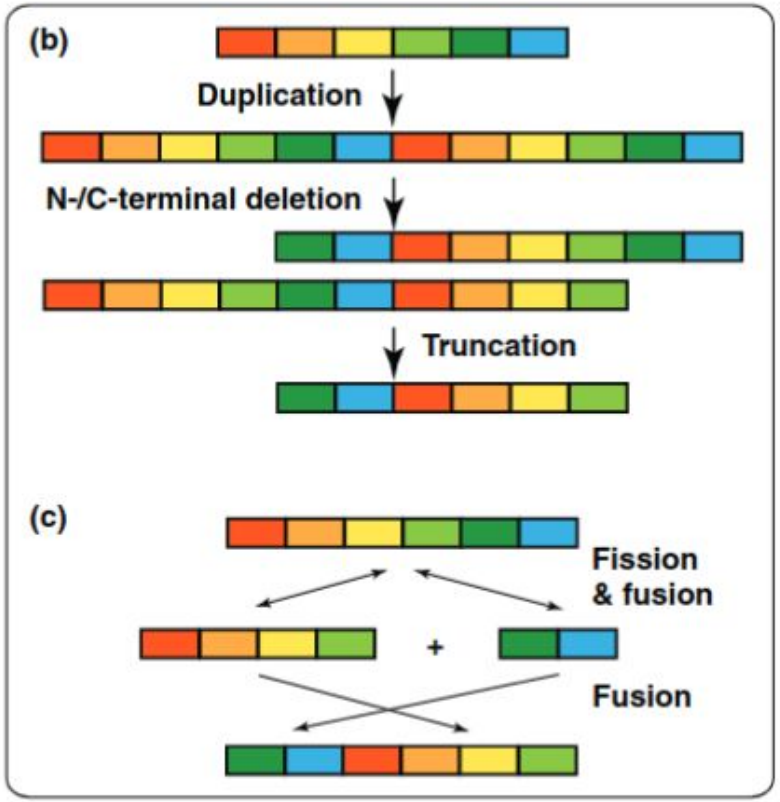


DALI
SSM

Что не может топологическое выравнивание: круговые перестановки







Круговые перестановки

Не редкость в природе

Можно проектировать:

- Изменить место окончания цепи – расширить набор областей пространства, куда можно продлить цепочку, например, пришивая репортерный белок
- Зимогены активируются путем разрезания – можно сделать конститутивно активированный фермент

CE (Combinatorial Extension)

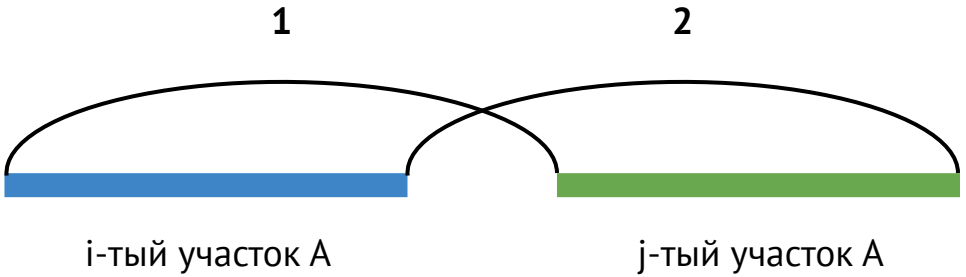
2. **Продление цепочки**

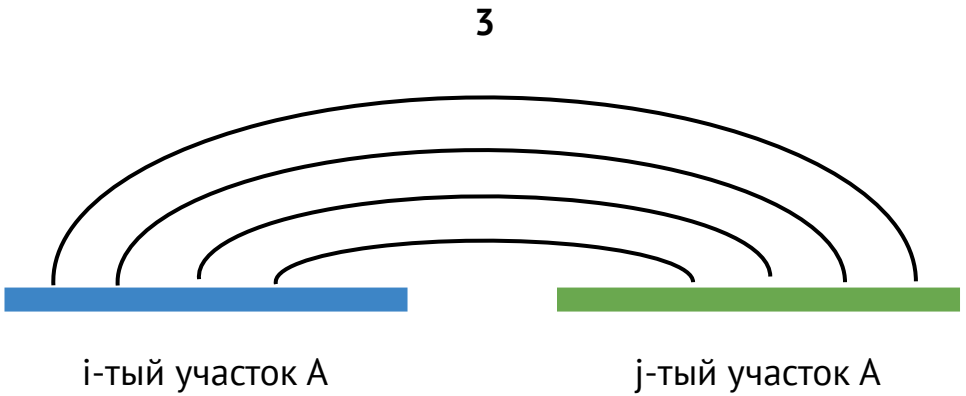
Для всех AFPj, которые не наслаиваются на данный AFPi (и остальные AFP в цепочке), решается, встроить ли их в цепочку

1) Dj < 3A
2) Вычисляется Dij:

$$D_{ij} = \frac{1}{m}(|d^A_{p_i^A p_j^A} - d^B_{p_i^B p_j^B}| + |d^A_{p_i^A+m-1,p_j^A+m-1} - d^B_{p_i^B+m-1,p_j^B+m-1}| + \\ + \sum_{k=1}^{m-2} |d^A_{p_i^A+k,p_j^A+m-1-k} - d^B_{p_i^B+k,p_j^B+m-1-k}|)$$

$$D_{ij} = \frac{1}{m}(|d^{A}_{p_i^A p_j^A} - d^{B}_{p_i^B p_j^B}| + |d^{A}_{p_i^A+m-1,p_j^A+m-1} - d^{B}_{p_i^B+m-1,p_j^B+m-1}| + \\ + \sum_{k=1}^{m-2} |d^{A}_{p_i^A+k,p_j^A+m-1-k} - d^{B}_{p_i^B+k,p_j^B+m-1-k}|)$$





CE (Combinatorial Extension)

Условия для продления цепочки путем добавления AFPn:

$$D_n < D_0$$

$$\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1$$

$$\frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} D_{ij} < D_1$$

Где D_0 = 3A, D_1 = 4A

CE (Combinatorial Extension)

3. **Выбор лучшей финальной цепочки**

Идея: использовать такую меру сходства структур, которая отражала бы некий баланс между требованием к максимизации длины выравнивания и минимизации RMSD.

Решение: вероятность случайно получить выравнивание такой же длины с таким же или меньшим числом гэпов и с таким же или меньшим RMSD.

Авторы взяли много пар структур (какие-то из них похожи, какие-то нет) и построили выравнивания при помощи своего алгоритма. Для каждого выравнивания можно посчитать длину и RMSD. Для каждой длины выравнивания можно получить распределение RMSD и числа гэпов по такой случайной выборке.

CE (Combinatorial Extension)

4. **Оптимизация**

Если есть черновое выравнивание, то его можно оптимизировать. А именно, можно:

- Удлинять выровненные участки.
- Укорачивать их.
- Сдвигать гэпы.

Каждое такое изменение соответствует некоторому новому выравниванию. И, следовательно, некоторому совмещению. И у него есть длина, RMSD и Z-score. Можно посмотреть, увеличился ли он от каждого возможного изменения.

Алгоритм CE рассматривает последовательности таких изменений, выбираемые случайным образом. Те последовательности, которые ведут к сильному ухудшению Z-score отбраковываются.

Похожие идеи

Комбинаторный перебор продлений может быть заменен на **динамическое программирование**. В таком случае процедура становится похожей на алгоритм Нидлмана-Вунша, только S_{ij} вместо значения из матрицы аминокислотных замен становится величиной соответствия AFP_{ij}.

Старт алгоритма динамического программирования с самого раннего по ходу последовательностей белков A и B подходящего AFP.

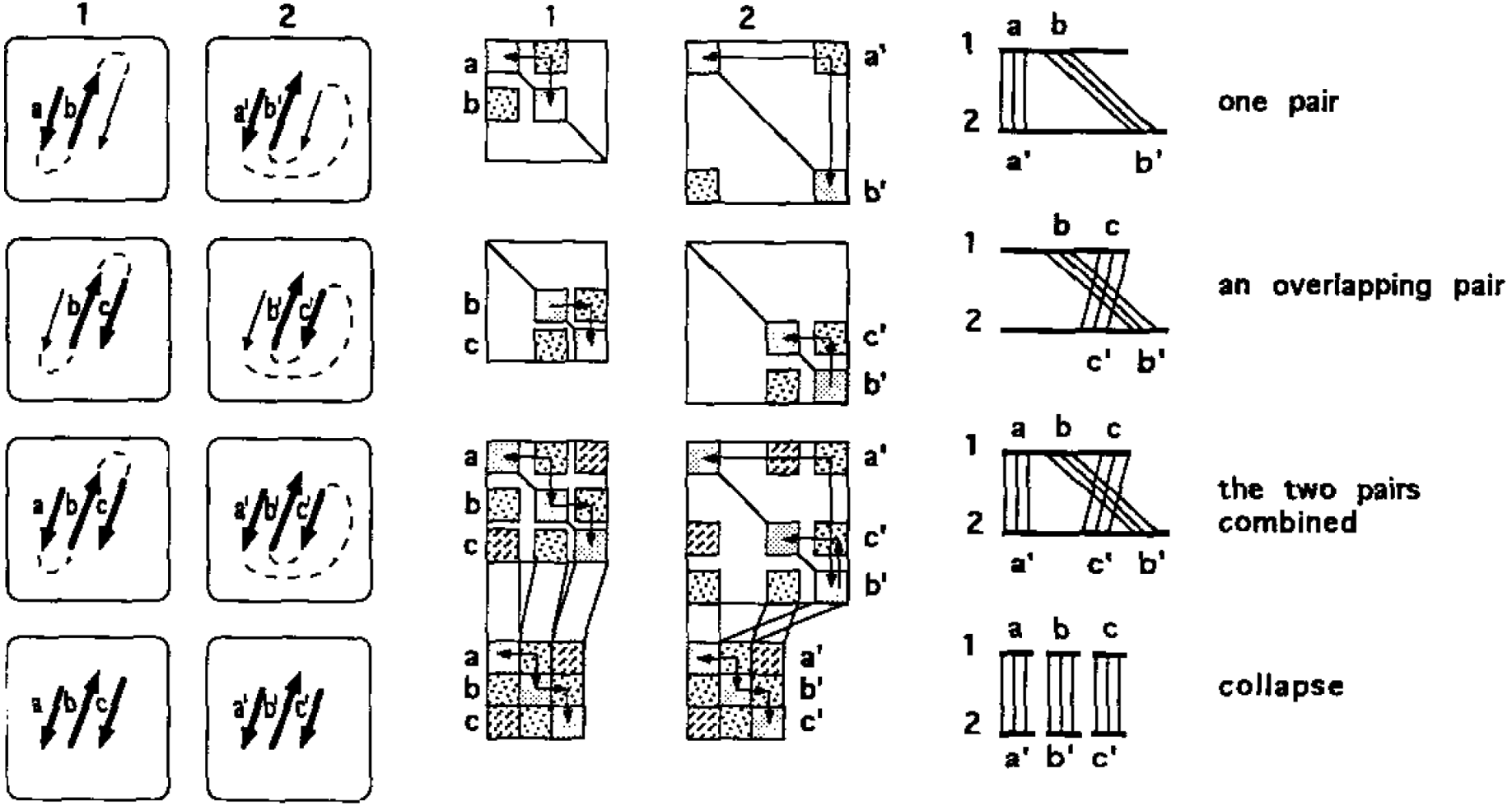
S(k) – лучший скор для цепочки, оканчивающейся на AFPk

$$S(k) = D_k + \max(\max_m(S(m) + a(m \to k)), 0)$$

Качество самого AFPk

“Цена” включения AFPk в цепочку

DALI – матрицы расстояний 6x6



DALI

Целевая функция

$$S = \sum_{i=1}^{L} \sum_{j=1}^{L} \phi(i,j)$$, где i и j – координаты в матрице (= в выравнивании)

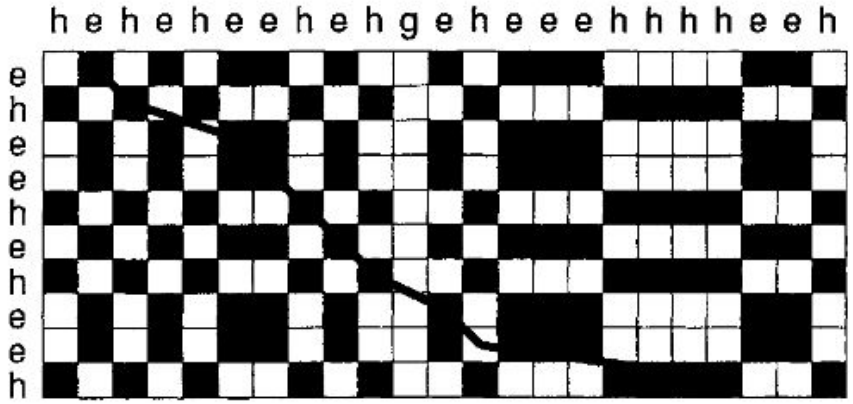
$$\phi^R(i,j) = \theta^R - |d_{ij}^A - d_{ij}^B|$$

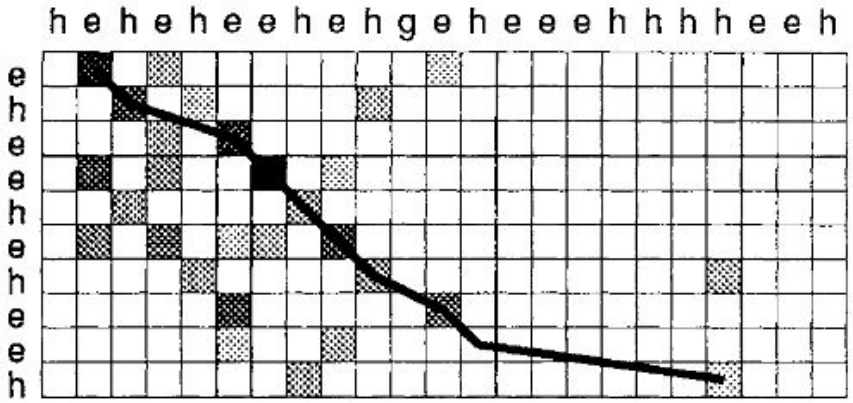
$$\phi^E(i,j) = \left\{ \begin{array}{l} (\theta^E - \frac{|d_{ij}^A - d_{ij}^B|}{d_{ij}^*})\omega(d_{ij}^*),\ i \neq j \\ \theta^E,\ i = j \end{array} \right\}, \omega(r) = e^{-\frac{r^2}{400}}$$

Смысл: сильно непохожие расстояния вообще не надо учитывать.

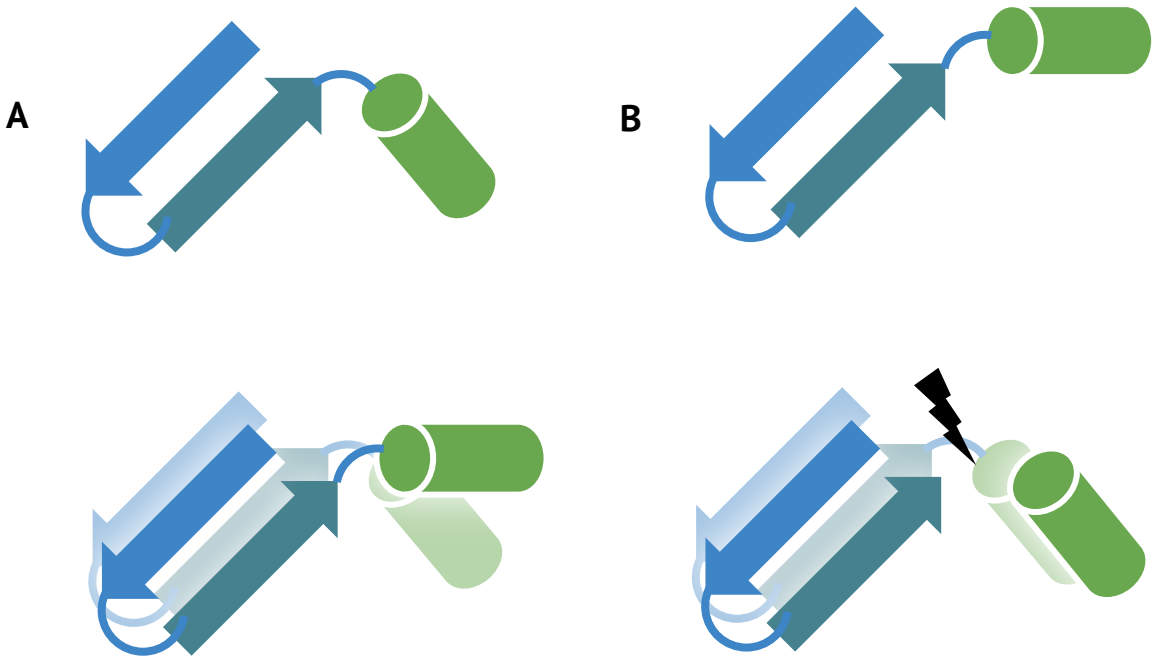
DALI решает задачу поиска лучшей цепочки выровненных пар гексамеров с помощью алгоритма Монте-Карло

Вторичная структура как AFP





Гибкое выравнивание



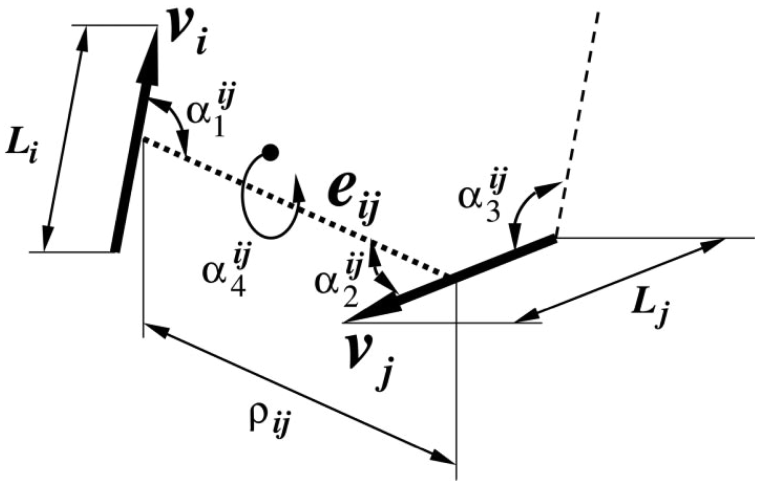
CE не примет продление цепи через добавление зеленого AFP

FatCat и MATT допускают изгибы при оценке возможности добавления AFP
NB: финальный RMSD все равно рассчитывается по интактным структурам

Алгоритм PDBeFold (SSM)

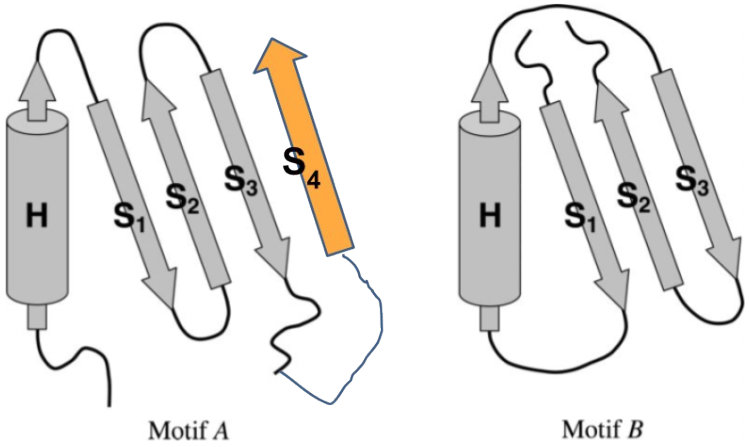
Сравниваем между собой взаиморасположения пар элементов вторичной структуры

- Отдаленно напоминает DALI – там мы сравнивали взаиморасположения пар гексамеров
- Однако в DALI мы описывали его как матрицу расстояний
- А в SSM мы присвоим каждому элементу вторичной структуры вектор из начала в конец и зададим их взаиморасположение набором чисел



Будем использовать эти наборы, чтобы решать, сопоставляется ли пара элементов из белка А с парой элементов из белка В, т.е. похожее ли между элементами взаиморасположение в двух структурах

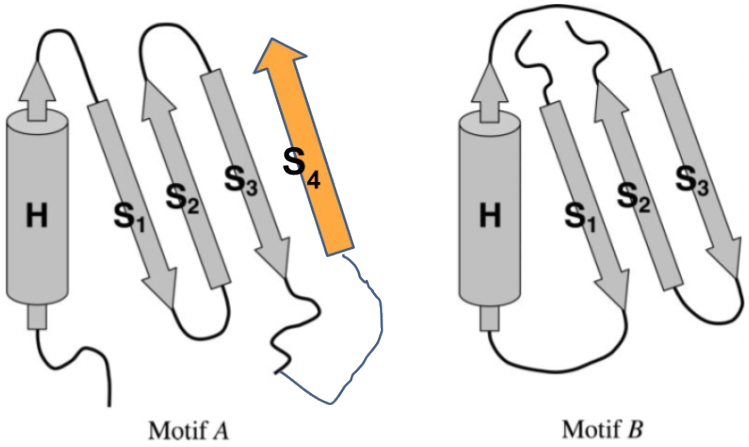
Алгоритм PDBeFold (SSM)



Это модифицированный пример из оригинальной статьи. Я добавил к первой структуре тяж S_4. Вопрос: какие элементы вторичной структуры тут расположены примерно одинаково и в A, и в B?

Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)

Алгоритм PDBeFold (SSM)



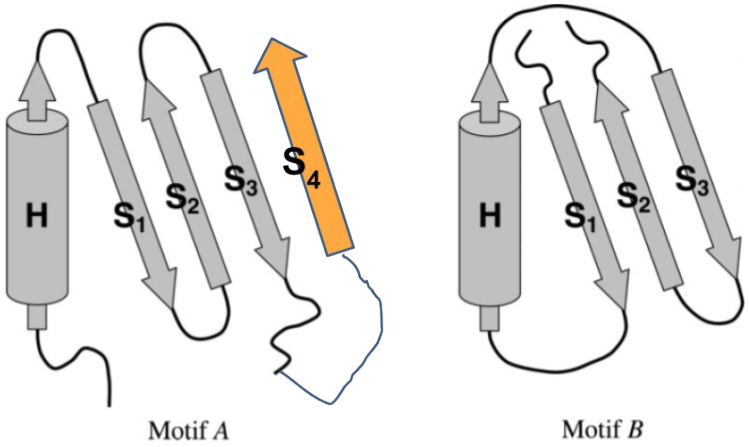
Motif *A* Motif *B*

Это модифицированный пример из оригинальной статьи. Я добавил к первой структуре тяж S_4. Вопрос: какие элементы вторичной структуры тут расположены примерно одинаково и в A, и в B?
Ответ: H, S_1, S_2, S_3.

1. Рассмотрим все пары элементов из первой структуры:
   (H, S_1), (H, S_2), (H, S_3), (H, S_4), (S_1, S_2), (S_1, S_3), (S_1, S_4),
   (S_2, S_3), (S_2, S_4), (S_3, S_4)
   + еще 10 (S_1, H), (S_2, H) и т.д.
2. Рассмотрим все пары элементов из второй структуры:
   (H', S_1'), (H', S_2'), (H', S_3'), (S_1', S_2'), (S_1', S_3'),
   (S_2', S_3')
   + еще 6 (S_1, H), (S_2, H) и т.д.
3. Рассмотрим двойки пар из п. 1 и 2 (например, (H, S_1) + (H', S_3') и проч.)
   Вопрос: сколько всего таких двоек?
   Ответ: 20 x 12 = 240.
   Вопрос: Все ли такие двойки пар соответствуют парам элементов, которые одновременно могут быть выровнены?

Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)

Алгоритм PDBeFold (SSM)



Рассмотрим двойки пар (например, (H, S_1) + (H', S_3') и проч.) – это гипотезы об одновременном выравнивании пар элементов

Вопрос: Все ли такие двойки пар соответствуют парам элементов, которые одновременно могут быть выровнены?

(H, S_1) + (H', S_3')
Может, т.к. спираль соответствует спирали, а тяж - тяжу

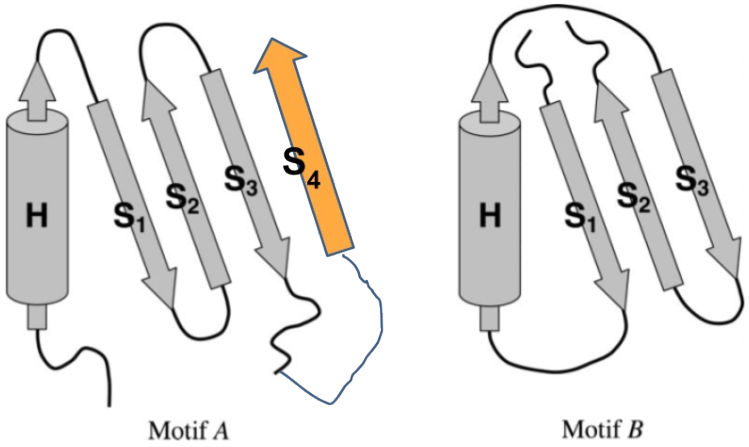
(H, S_1) + (S_1', S_3')
Не может, т.к. спираль не может быть выровнена с β-тяжем

(S_1, S_2) + (S_1', S_3')
Может

Из рисунка видно, что не все эти двойки реально выровнены. Но алгоритм этого пока не знает – это надлежит установить.

Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)

Алгоритм PDBeFold (SSM)



$(H, S_1) + (H', S_1')$

$(H, S_2) + (H', S_2')$

$(H, S_3) + (H', S_3')$

$(S_1, S_2) + (S'_1, S'_2)$

$(S_1, S_3) + (S'_1, S'_3)$

$(S_2, S_4) + (S'_1, S'_3)$

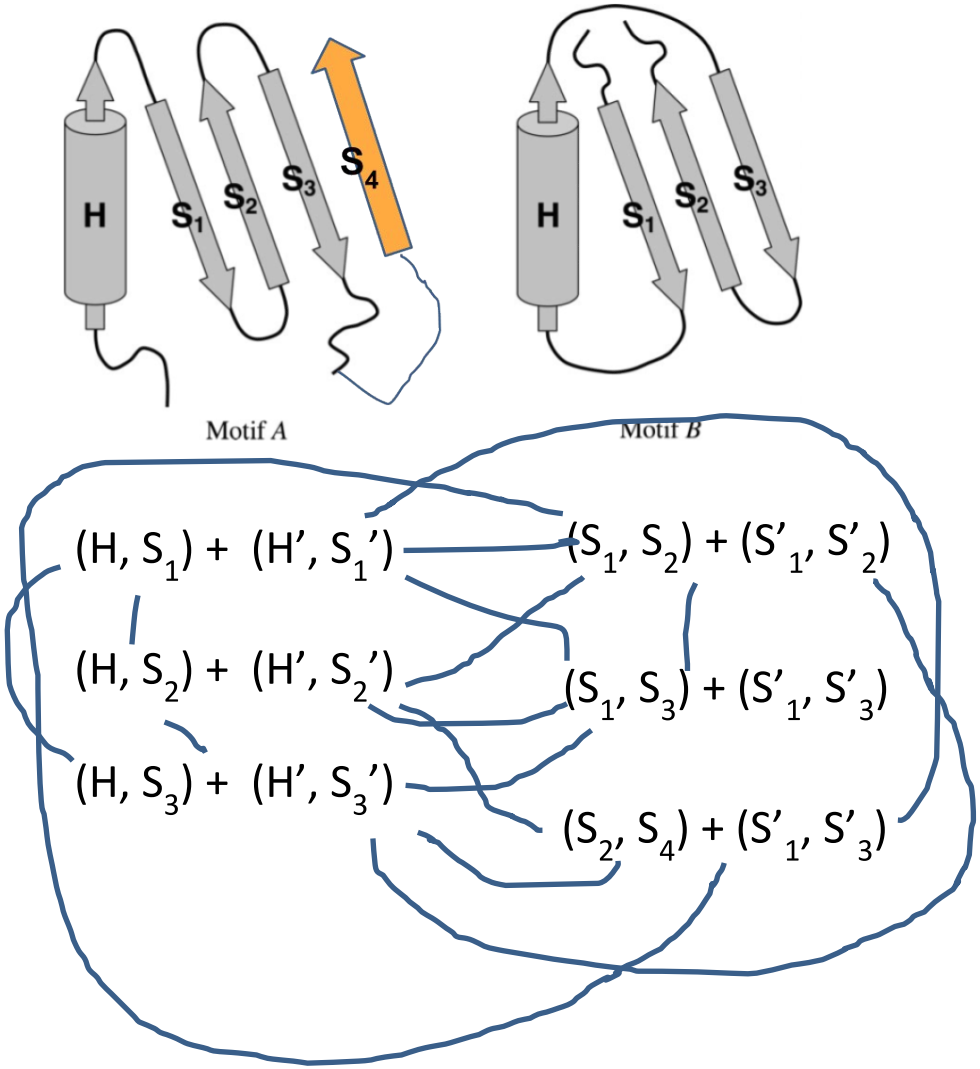
Отберем из всех двоек вида
те, которые потенциально могут быть выровнены. Это будут вершины графа. (На рисунке ниже показаны далеко не все вершины.)

Часть вершин соответствует правильному выравниванию. Часть – нет, но мы этого пока не знаем. Такова, например, двойка $(S_2, S_4) + (S'_1, S'_3)$. Как видно из рисунка в правильном выравнивании такой двойки нет. Но вообще говоря, это две пары тяжей, идущих в одном направлении, примерно с одинаковым расстоянием между ними.

Теперь соединим ребрами те вершины, между которыми нет противоречия.

Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)

Алгоритм PDBeFold (SSM)



Теперь соединим ребрами те вершины, между которыми нет противоречия.

Например, двойки
$(H, S_1) + (H', S_1')$ и
$(H, S_3) + (H', S_3')$
вполне могут быть в одном выравнивании.

А двойки
$(S_1, S_3) + (S'_1, S'_3)$ и
$(S_2, S_4) + (S'_1, S'_3)$
- нет. Действительно, S'_1 не может быть одновременно выровнен
и с S_1, и с S_2.

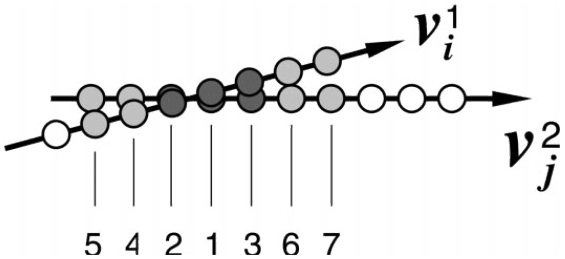
Клика в таком графе соответствует максимальному множеству выровненных элементов.

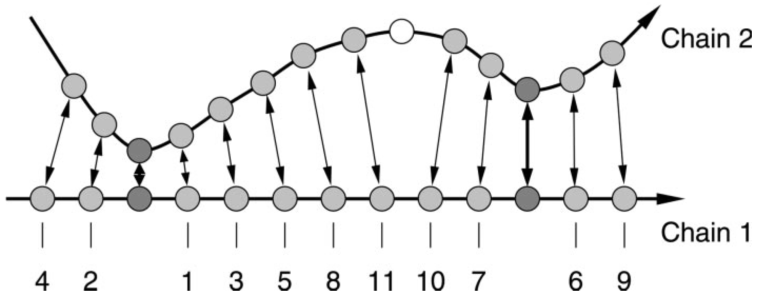
Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)

Алгоритм PDBeFold (SSM)

Далее, заменим каждый SSE на точку (его центр). Поскольку у нас уже есть списки совпадающих элементов, эти точки можно совместить. Получится черновое совмещение структур.

Теперь надо перейти от выравнивания элементов вторичной структуры к выравниванию последовательностей.





Для совпадающих SSE (см. предыдущий этап) выбирает 3-4 аминокислоты, наиболее близких друг к другу. Затем выравнивание расширяется на весь тяж или спираль.

Для всех остальных алгоритм находит взаимно наиболее близкие C_α атомы. Соседние с ними пары атомов также считаются выровненными.

Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)

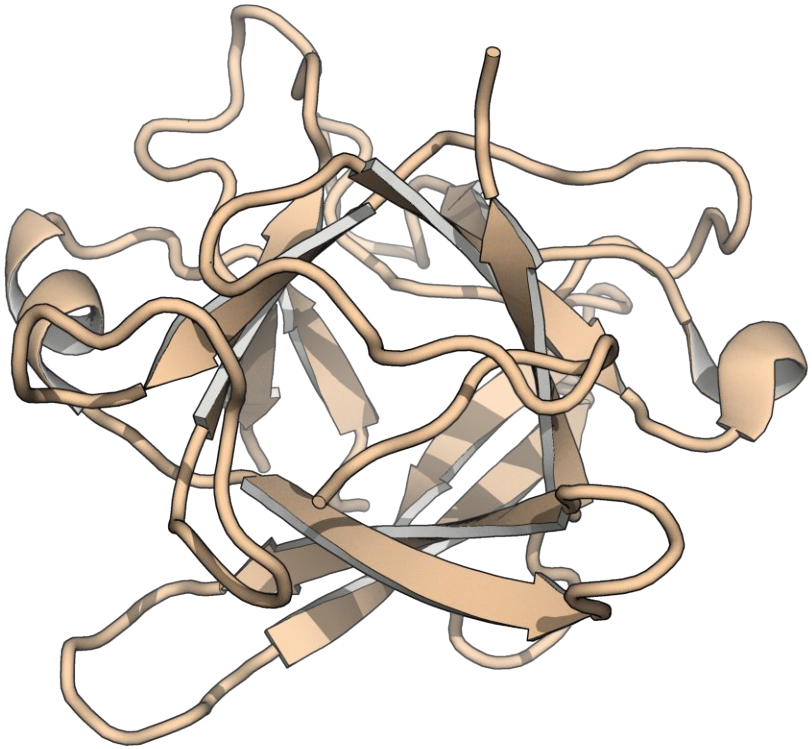
Алгоритм PDBeFold (SSM)

Далее, заменим каждый SSE на точку (его центр). Поскольку у нас уже есть списки совпадающих элементов, эти точки можно совместить. Получится черновое совмещение структур.

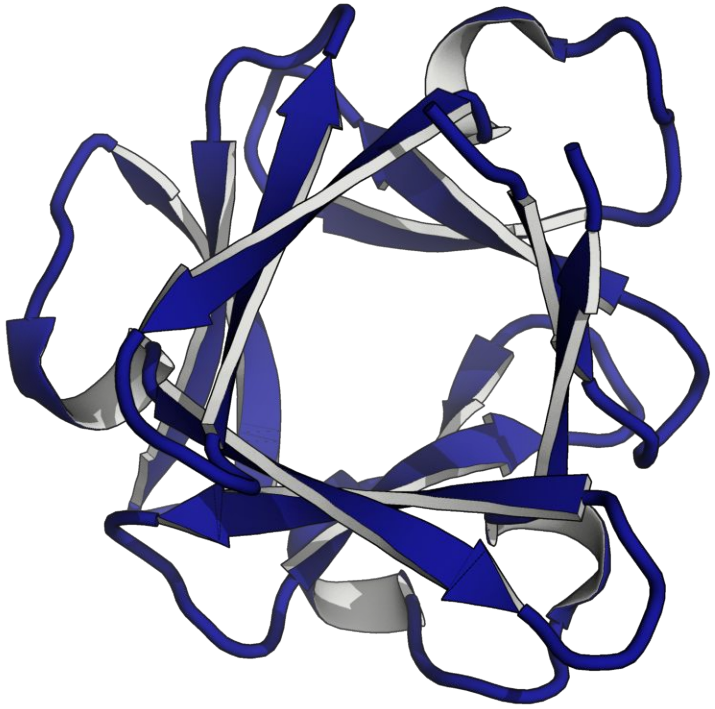
Теперь можно перейти от выравнивания элементов вторичной структуры к выравниванию последовательностей. При этом близко расположенные C_α атомы считаются выровненными и это выравнивание распространяется на их соседей по полипептидной цепочке.

Теперь у нас есть **черновое выравнивание** последовательностей. Ему соответствует некое совмещение структур, длина выравнивания, RMSD и Q-score. Его можно **оптимизировать**: менять разными способами, строить новое совмещение и добиваться увеличения Q. И так много-много раз, пока **решение не стабилизируется**.

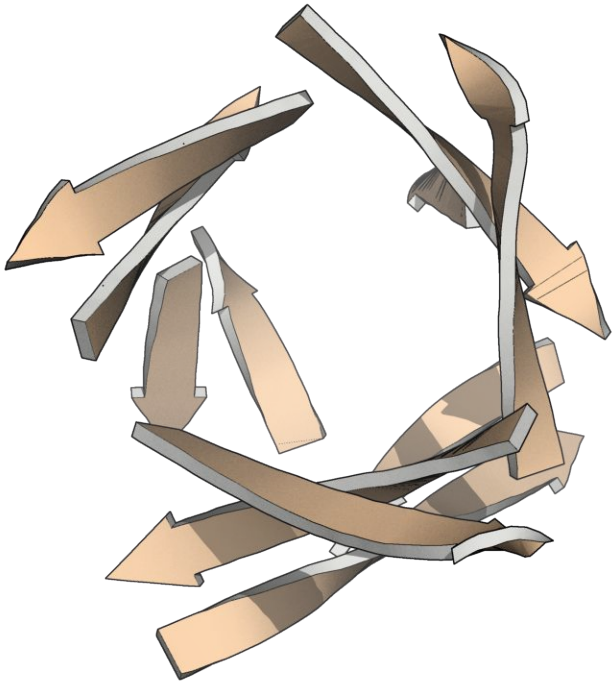
Krissinel E, Henrick K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1)



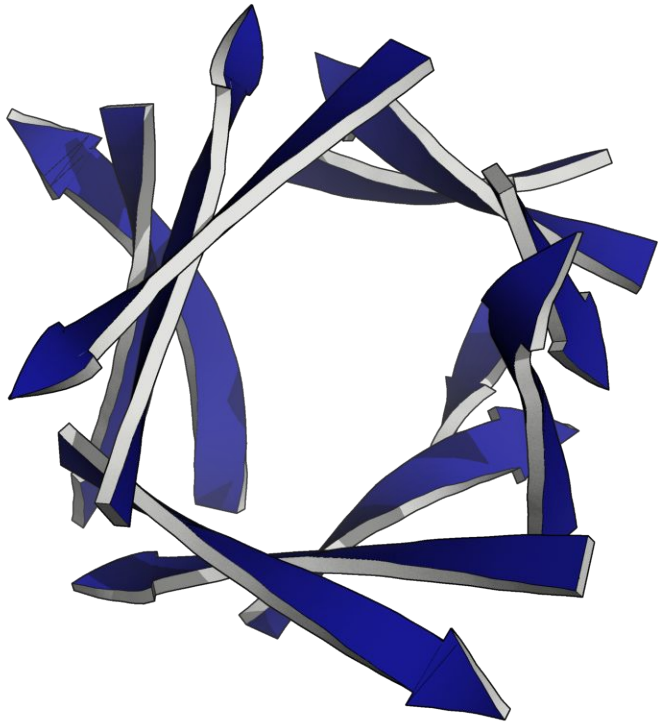
1TIE



4FGF

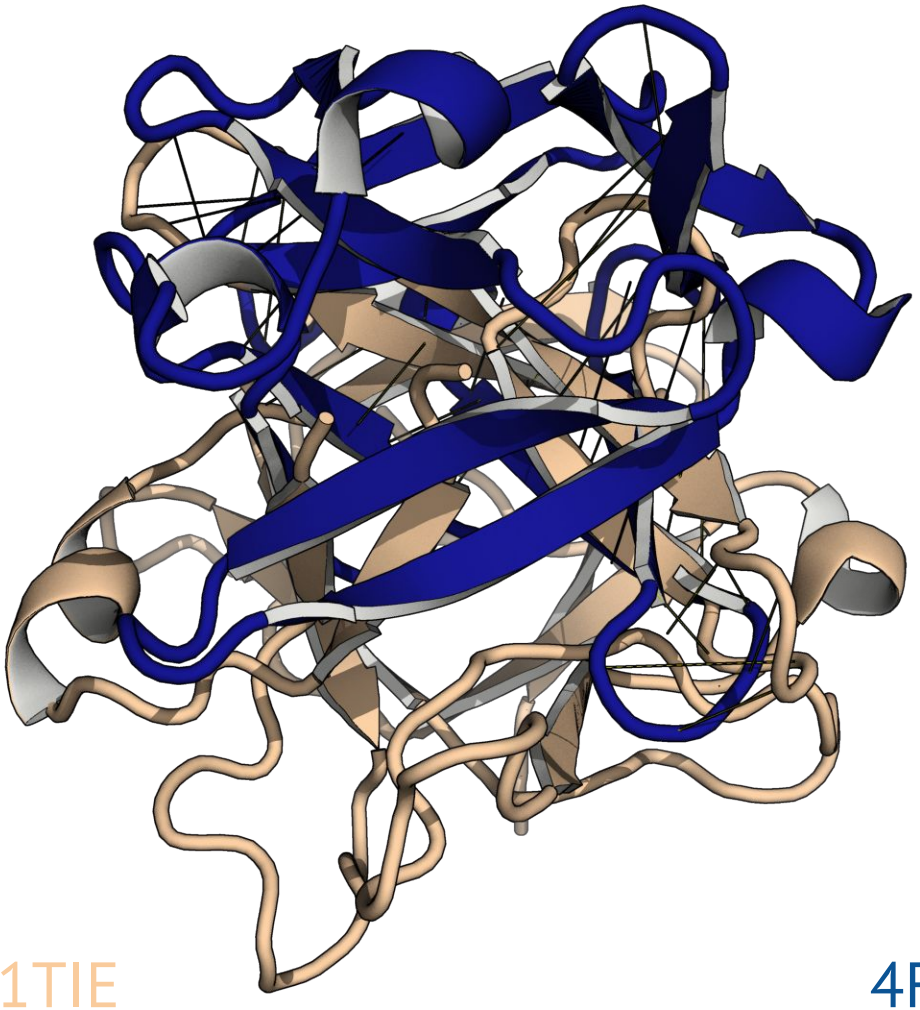


1TIE



4FGF

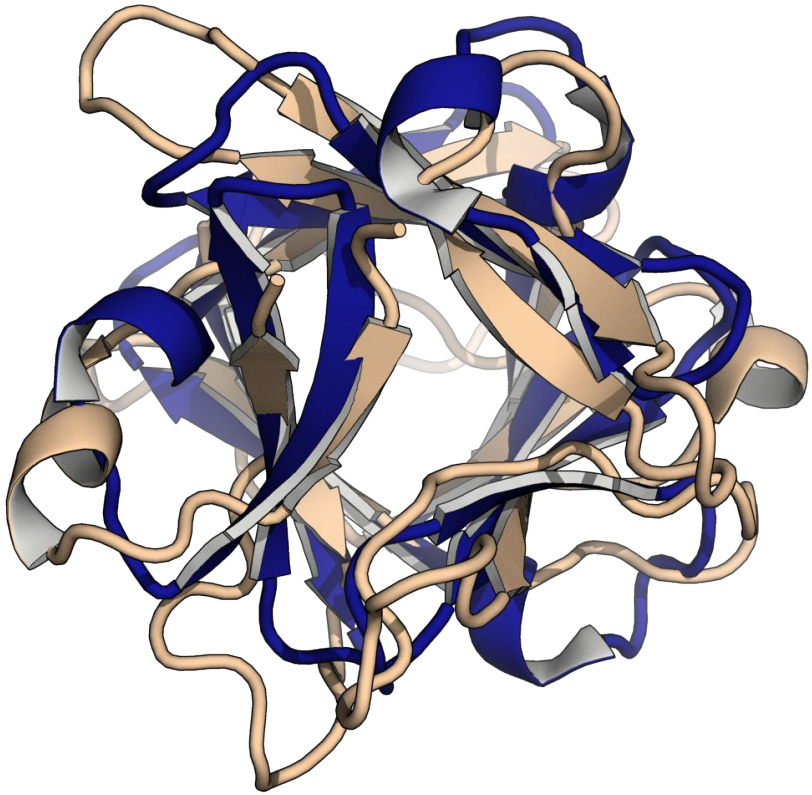
Align

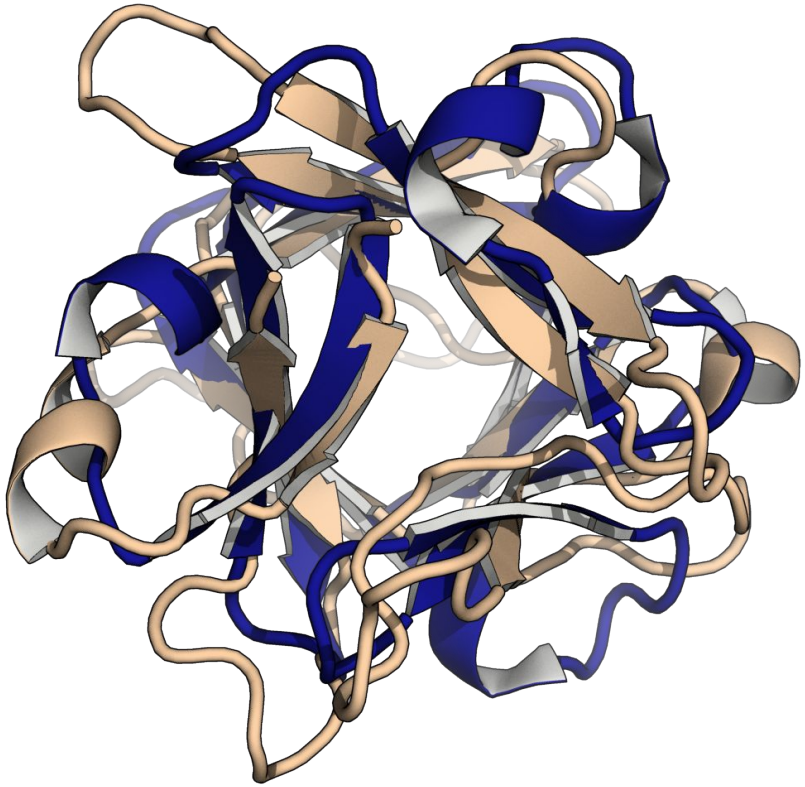


1TIE

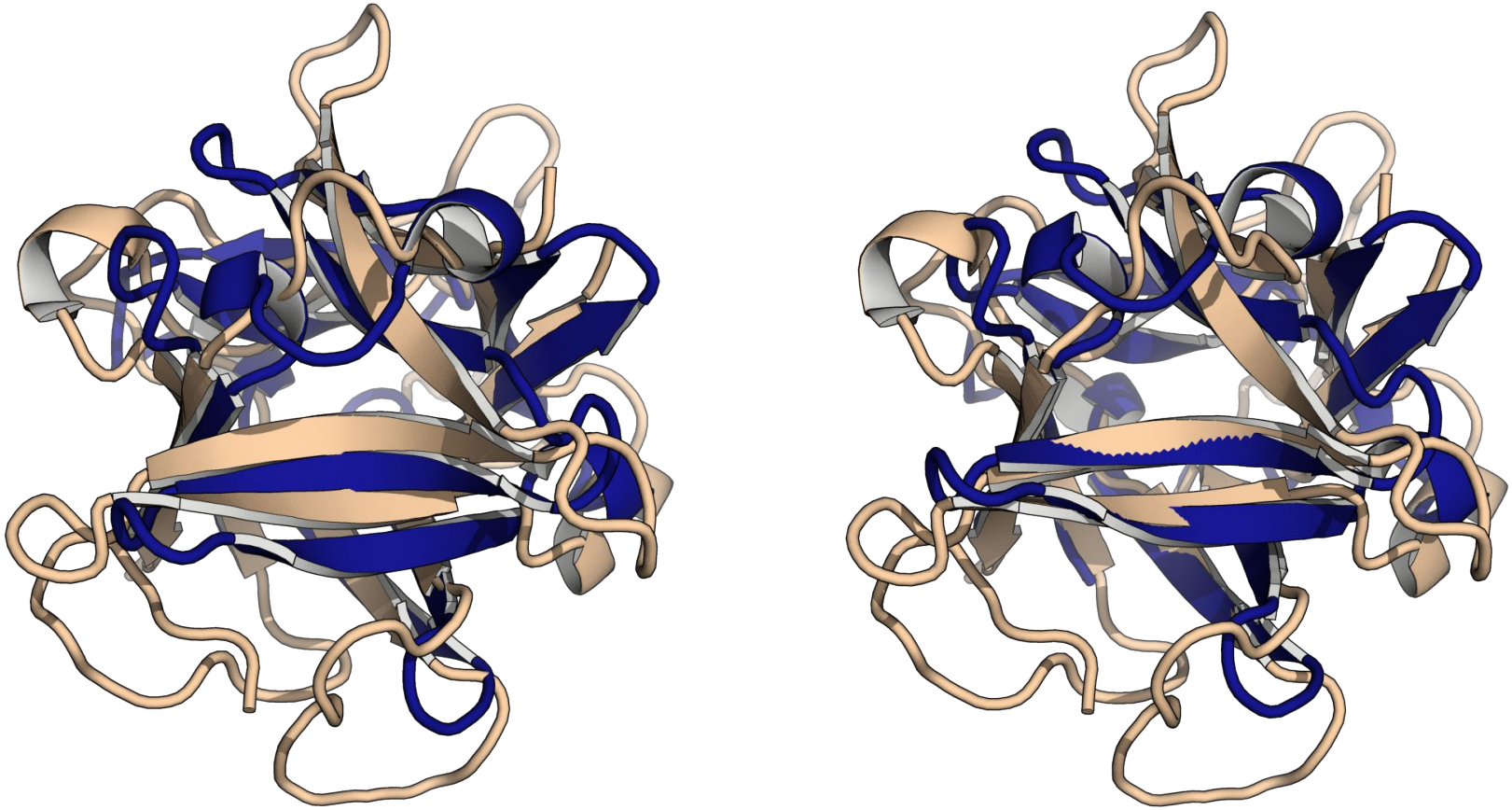
4FGF

Super

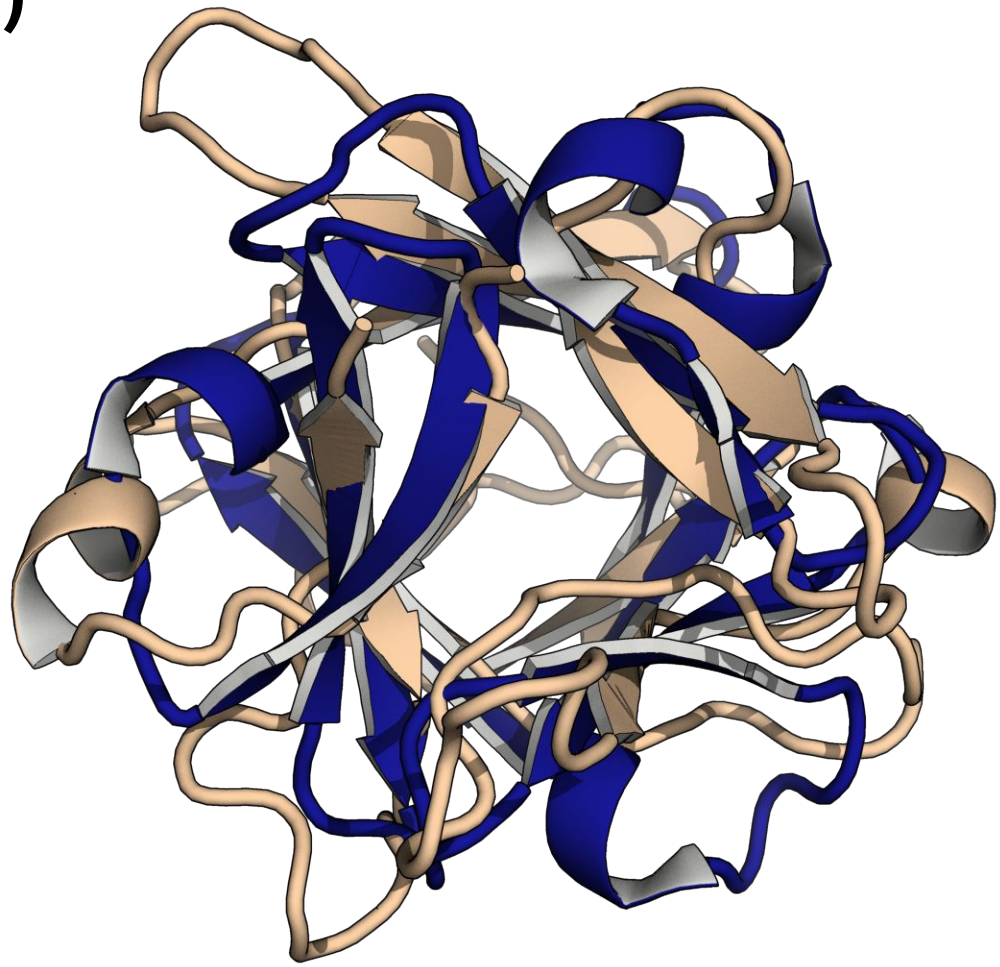




Super



CE (cealign)

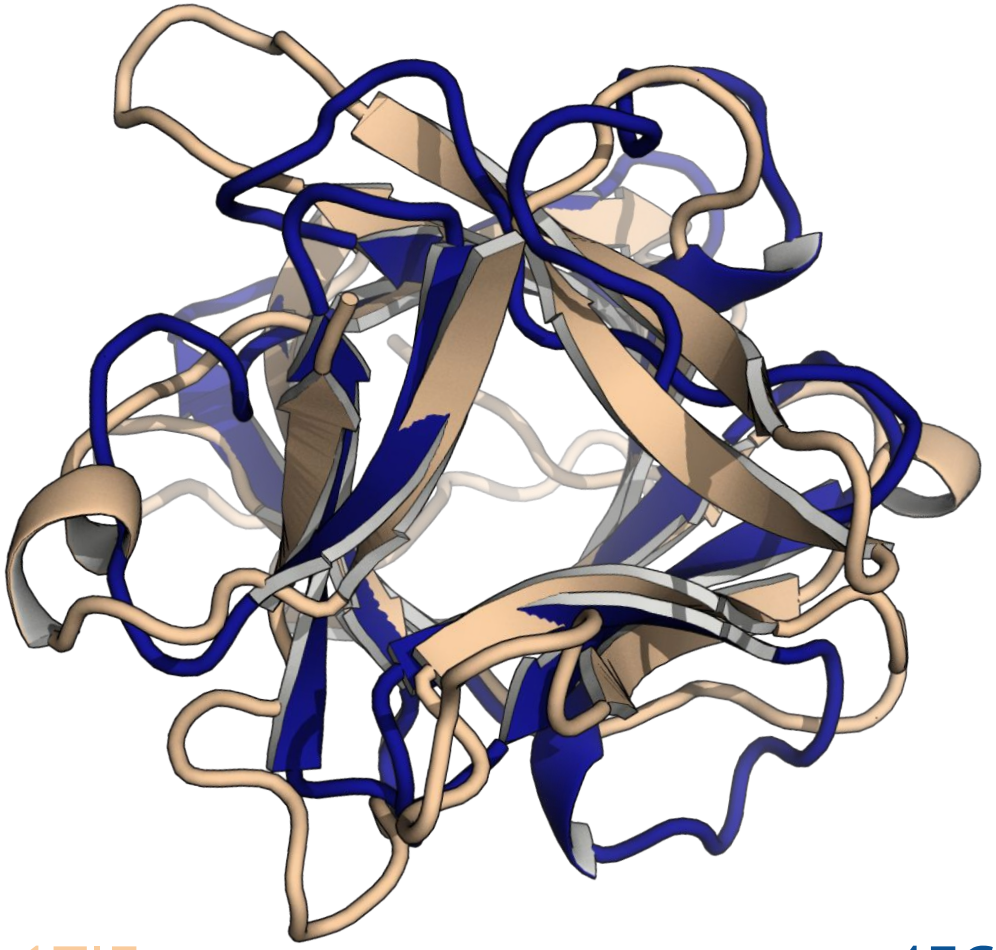


1TIE

4FGF

TM-align

(нет в PyMol)



1TIE

4FGF