

Выравнивание многих последовательностей

Алгоритмы в биоинформатике

Антон Елисеев

eliseevantoncoon@gmail.com

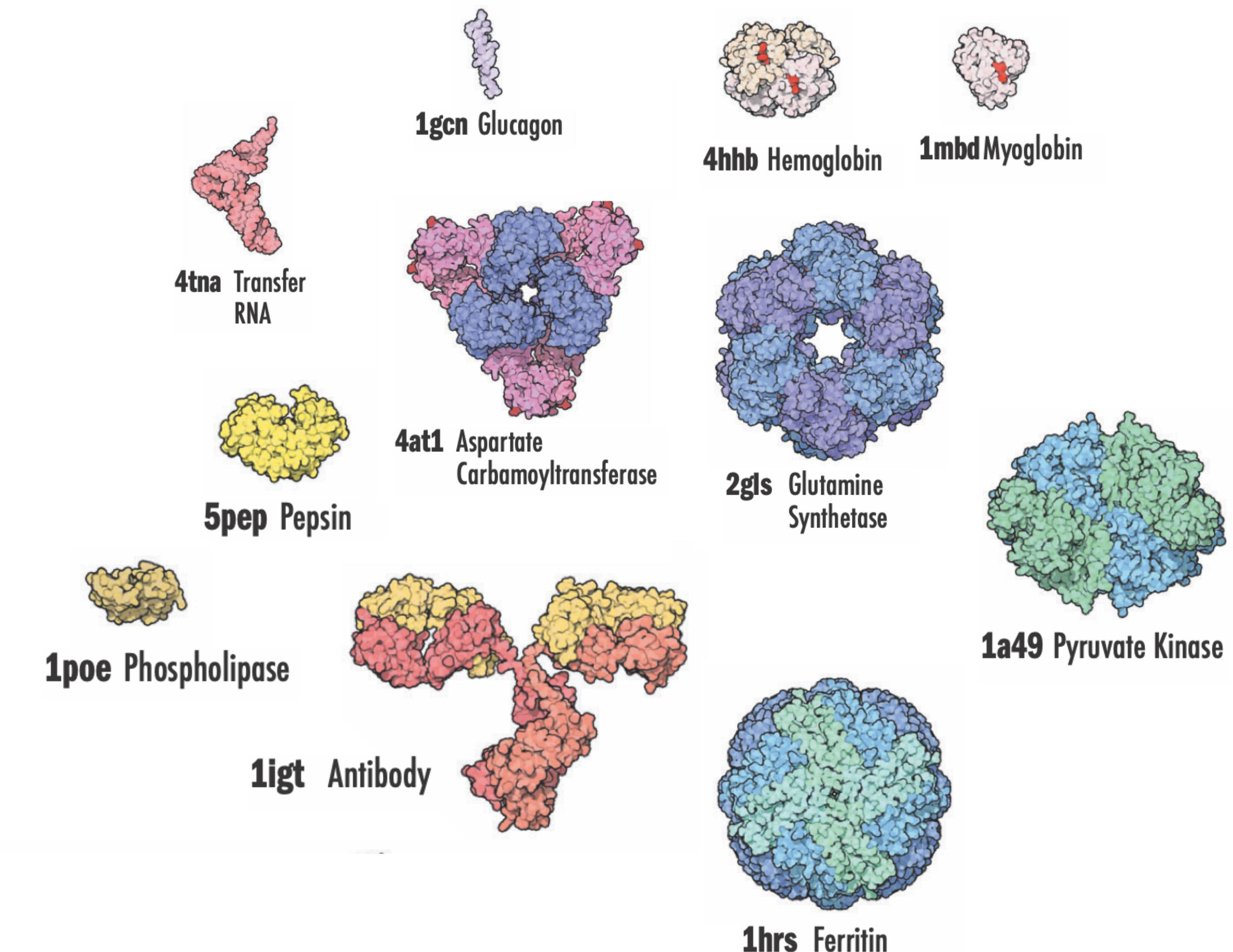
Что было на прошлой лекции

- Выравнивание многих последовательностей как средство поиска устойчивых и меняющихся участков генома.
- Определение множественного выравнивания и его профиля.
- Многомерная динамика и жадный алгоритм поиска множественного выравнивания, ClustalW.

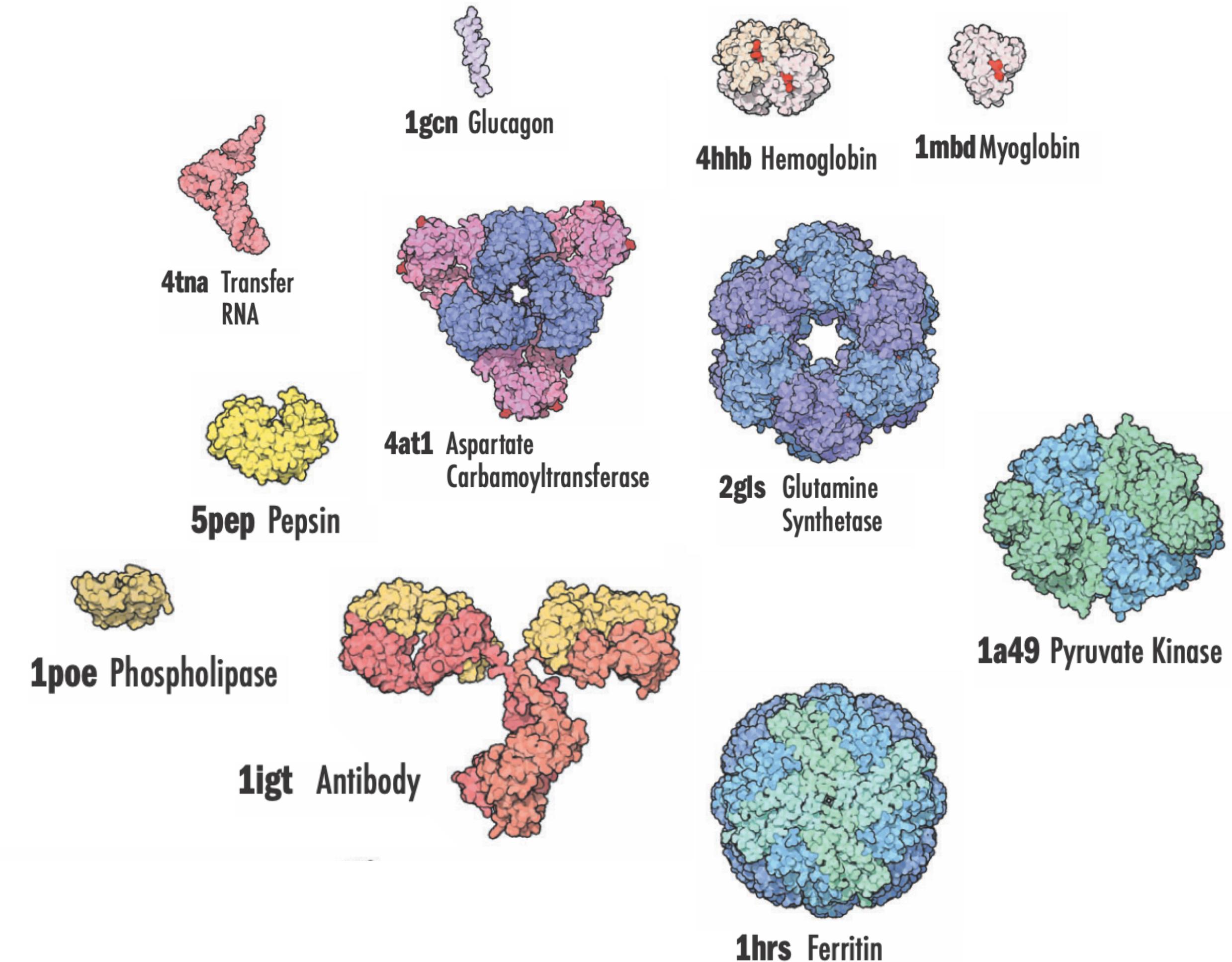
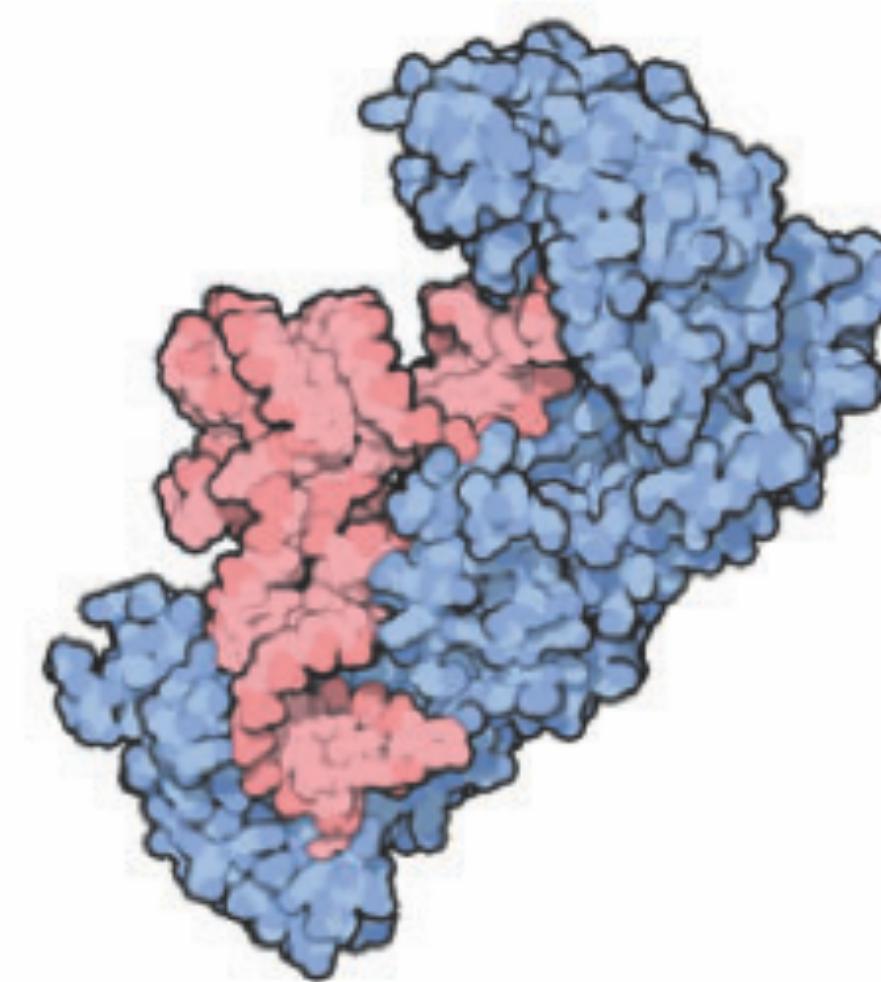
Что будет на этой лекции

- Рассмотрим эффективный алгоритм поиска ближайшего генома генома из множества к заданному
- Разберемся с прямым и обратным преобразованием Барроуза-Уилера
- Поиск совпадений строк при помощи индекса Барроуза-Уилера
- Выравнивание при помощи BWT

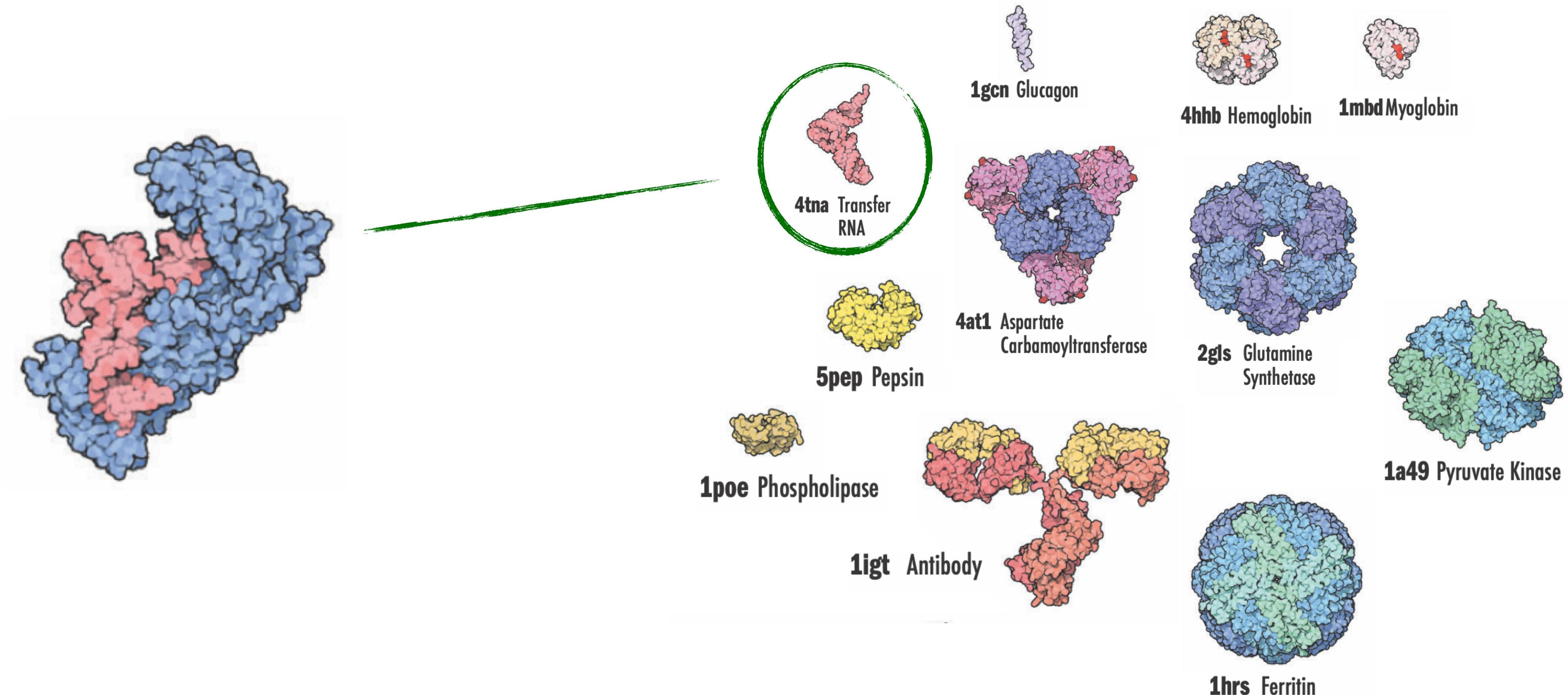
Поиск самого похожего генома



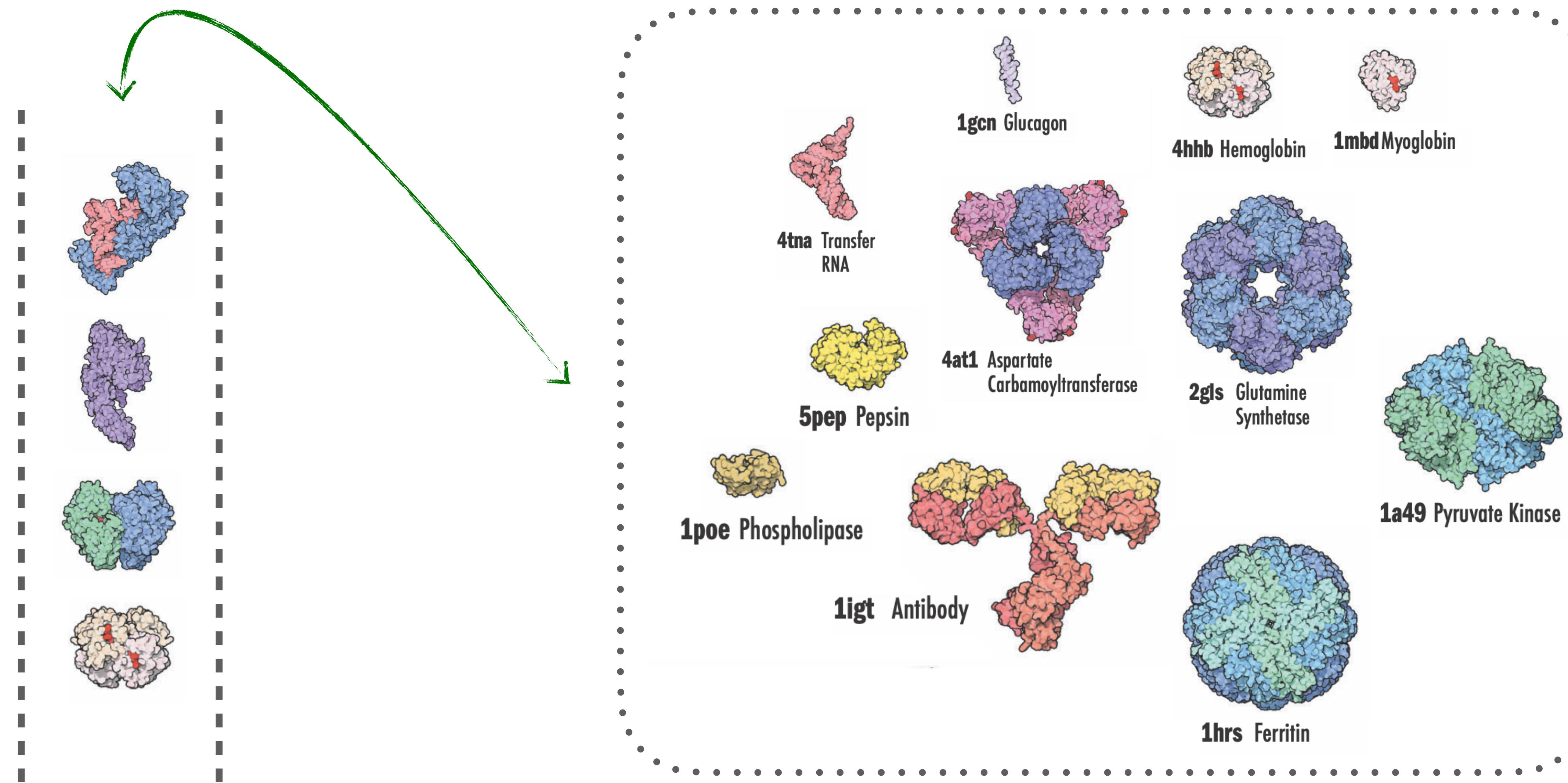
Поиск самого похожего генома



Поиск самого похожего генома



Поиск самого похожего генома



Поиск самого похожего генома

Для ответа на один запрос придется считать все расстояния

$$O(Nn^2)$$

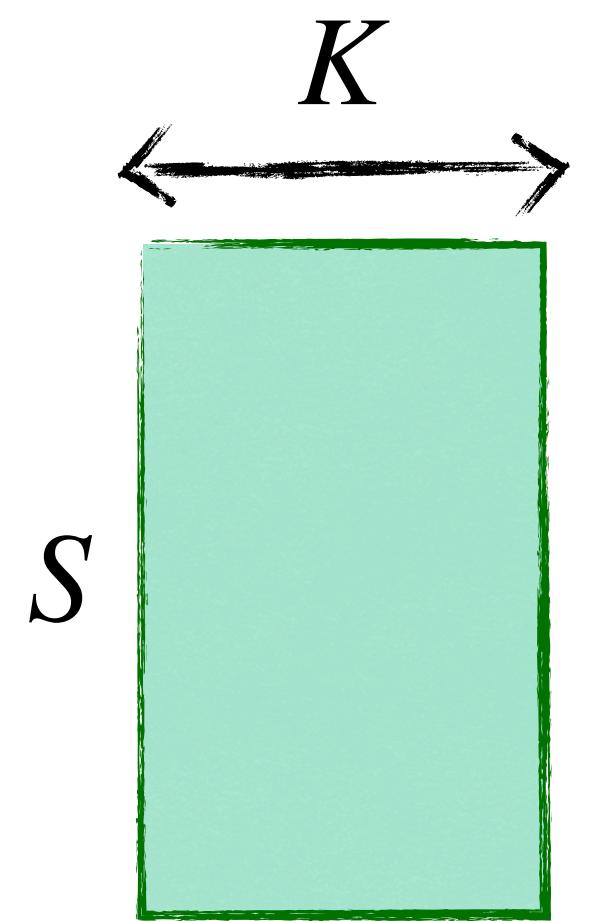
Но N может быть $\geq 10^5$

Поиск самого похожего генома

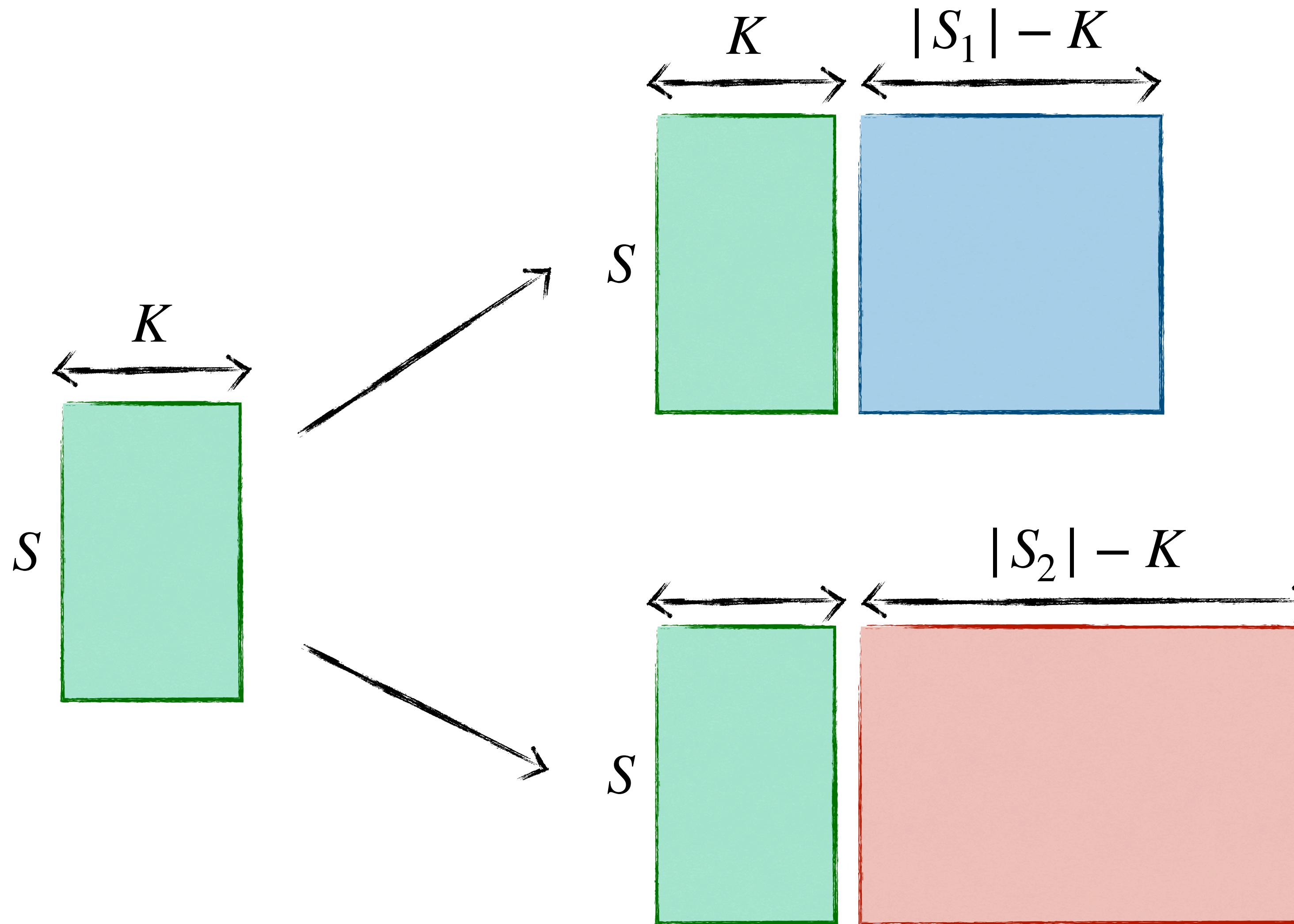
Рассмотрим S_1, S_2 с общим префиксом длины K

Как выравнивать S с S_1, S_2 ?

Поиск самого похожего генома



Поиск самого похожего генома



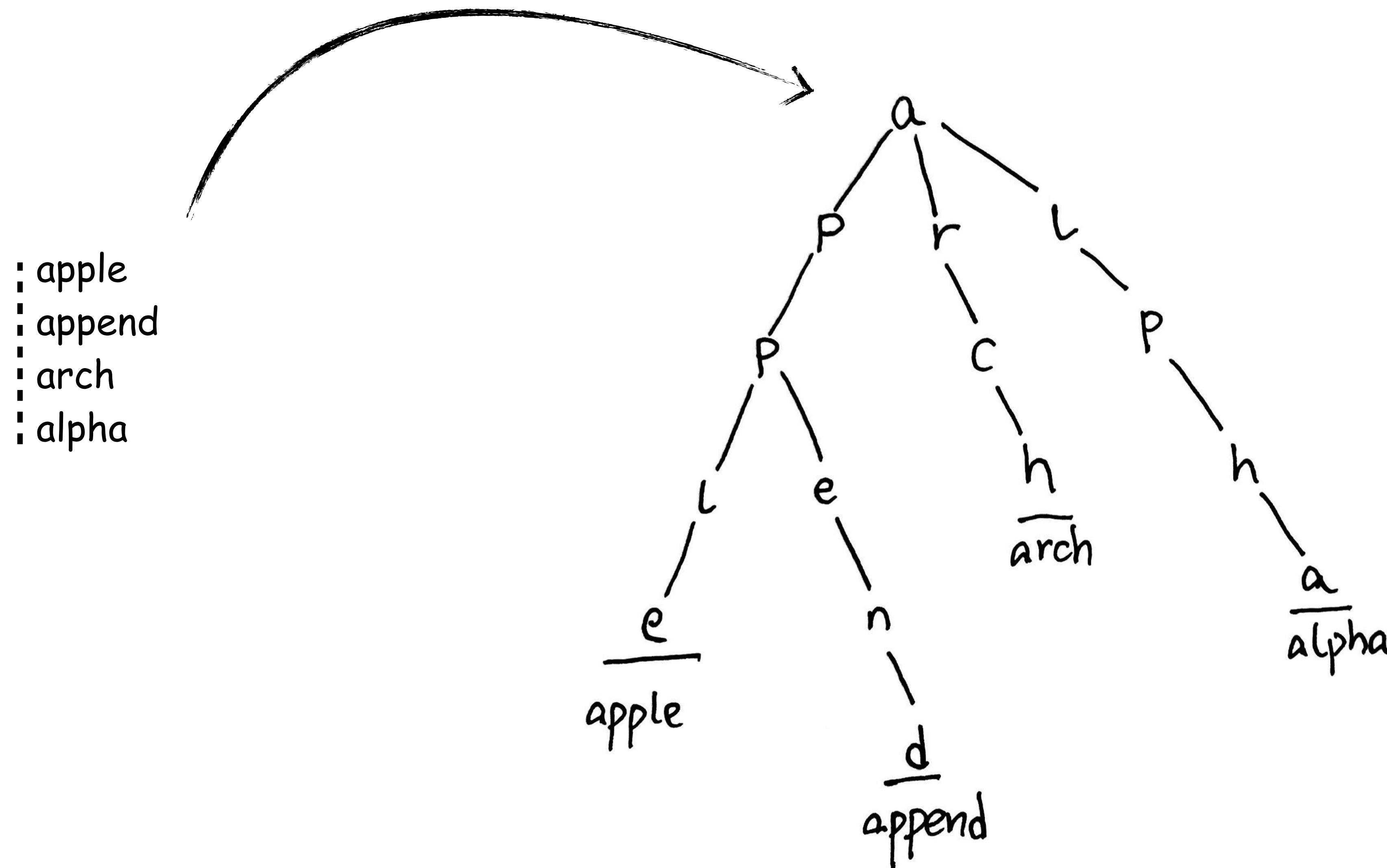
Поиск самого похожего генома

Рассмотрим S_1, S_2 с общим префиксом длины K

Как выравнивать S с S_1, S_2 ?

Если сперва выровнять префикс, а потом суффиксы S_1, S_2 то суммарно потратим $O((N_1 + N_2 - K) |S|)$

Поиск самого похожего генома



Поиск самого похожего генома

Префиксное дерево:

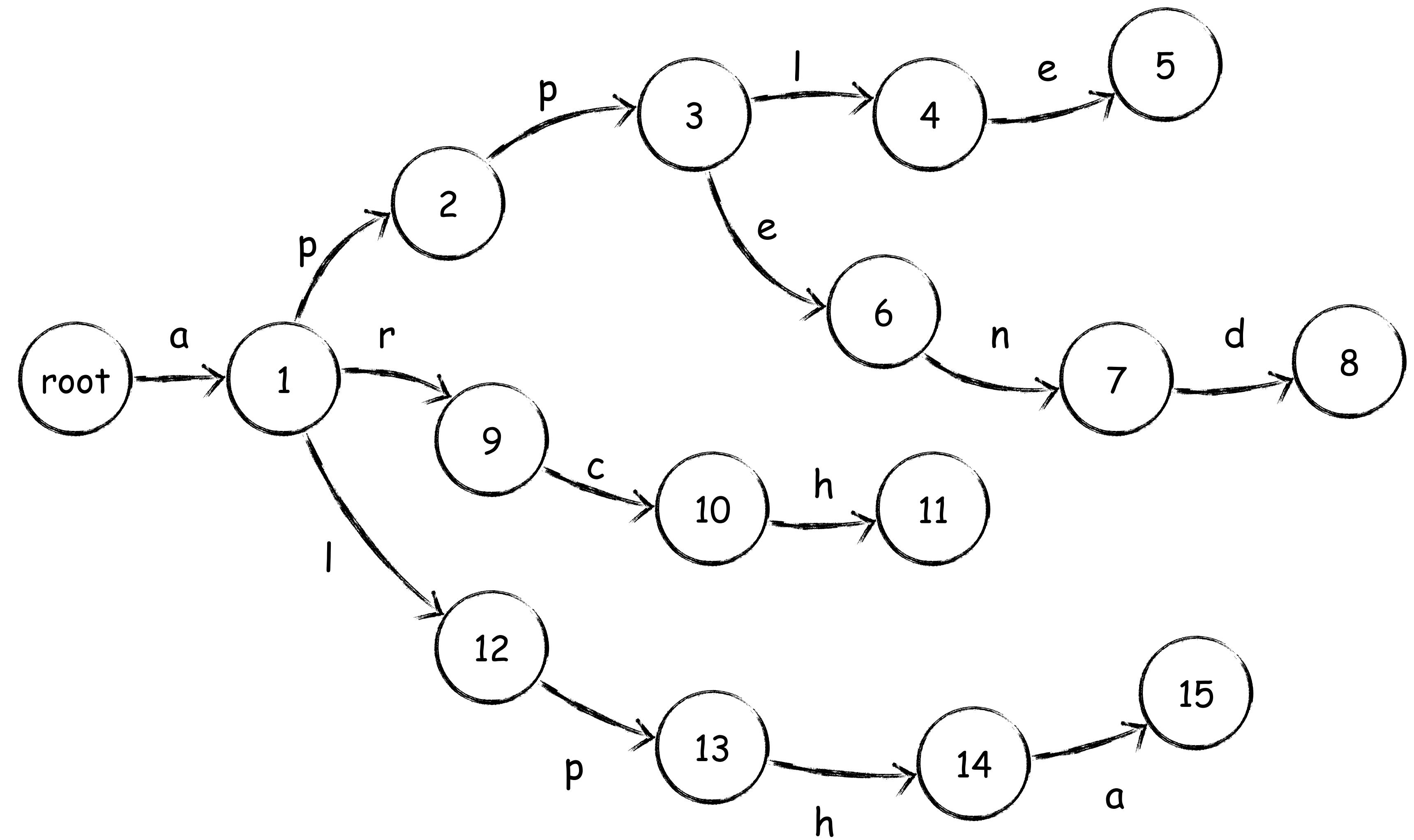
Ребра - символы последовательности

Вершины - индексы, описывающие порядок добавления элемента

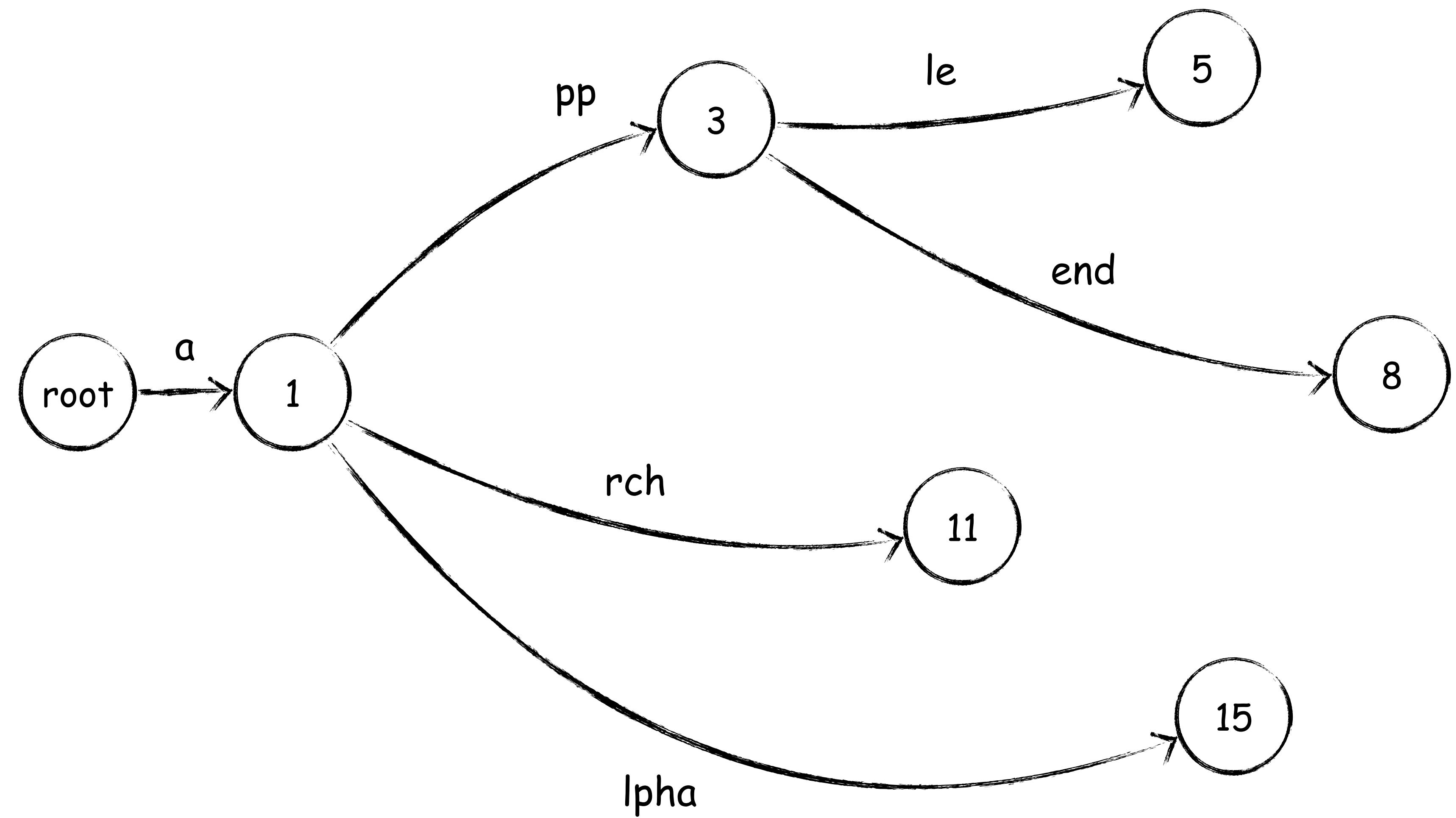


Поиск самого похожего генома

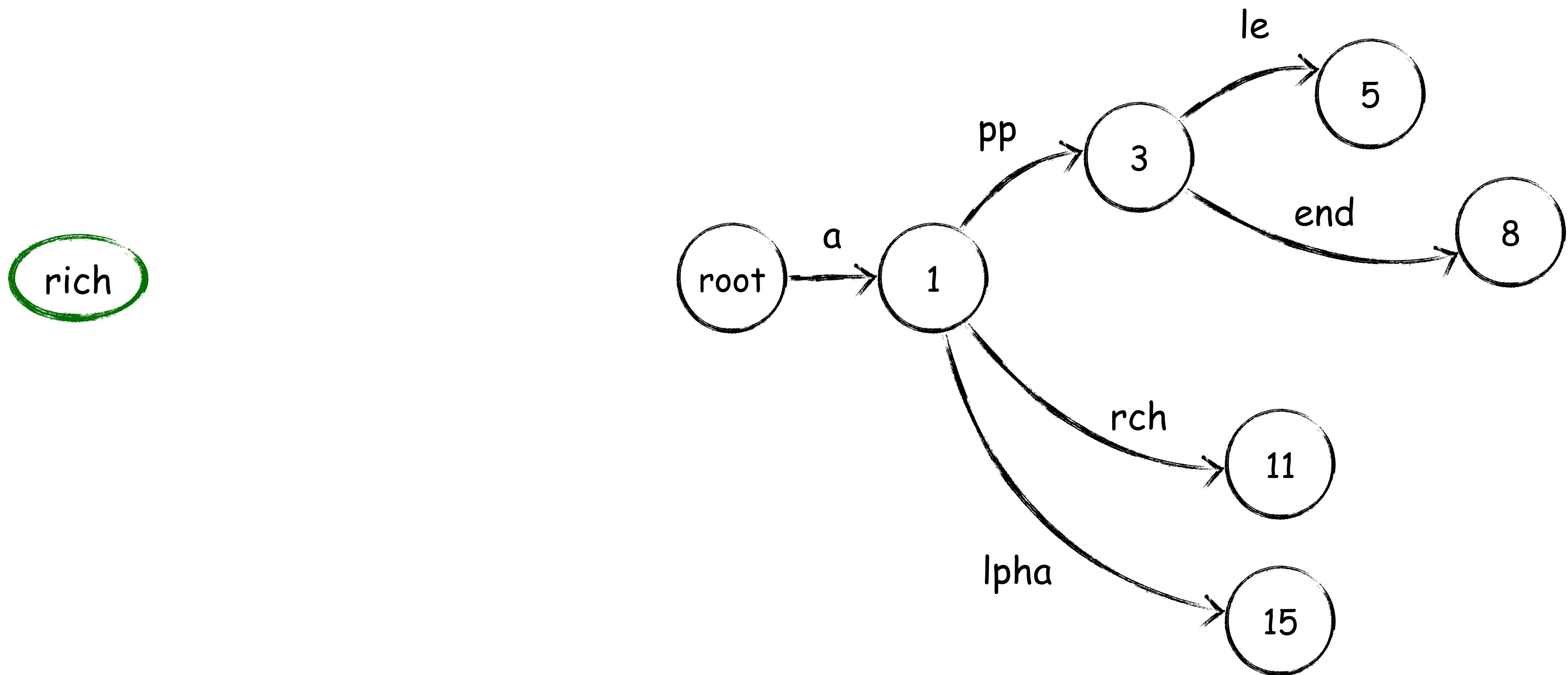
apple
append
arch
alpha



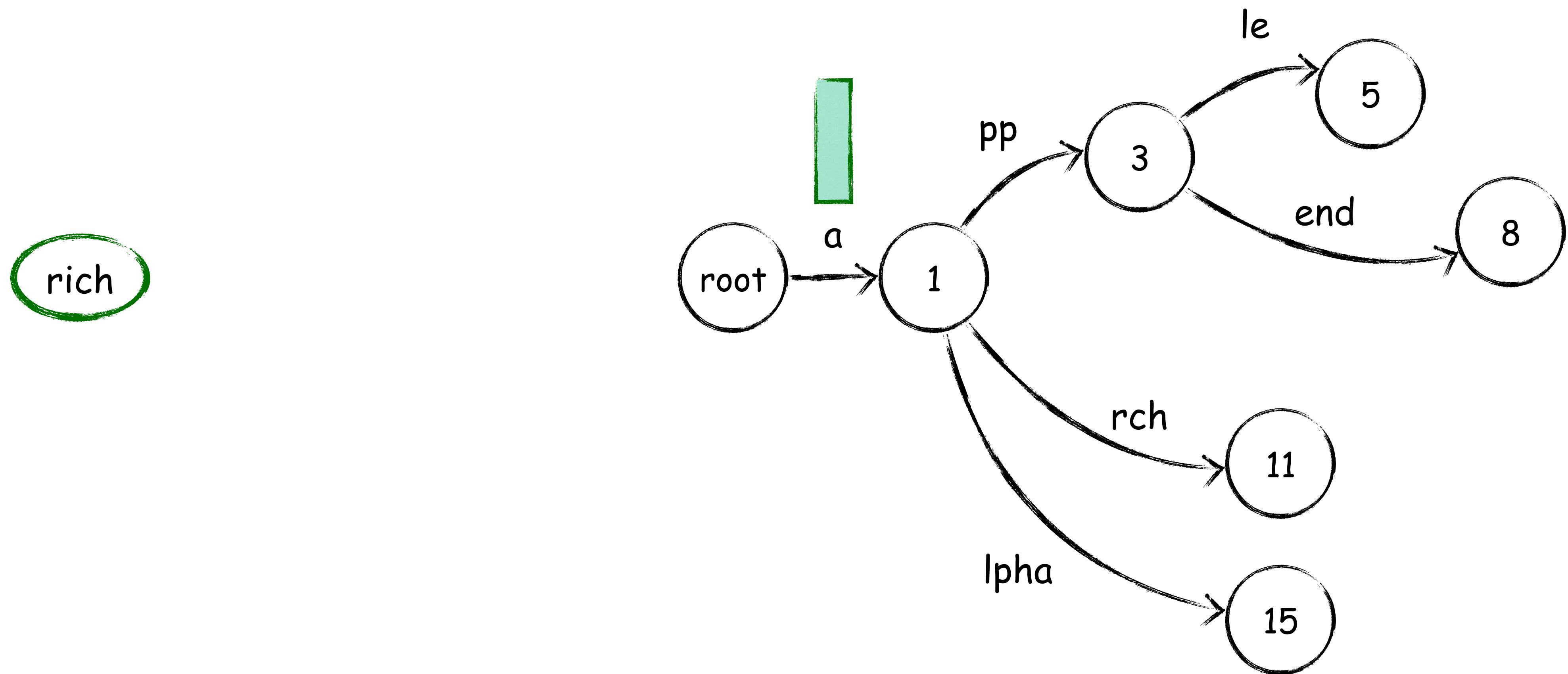
Поиск самого похожего генома



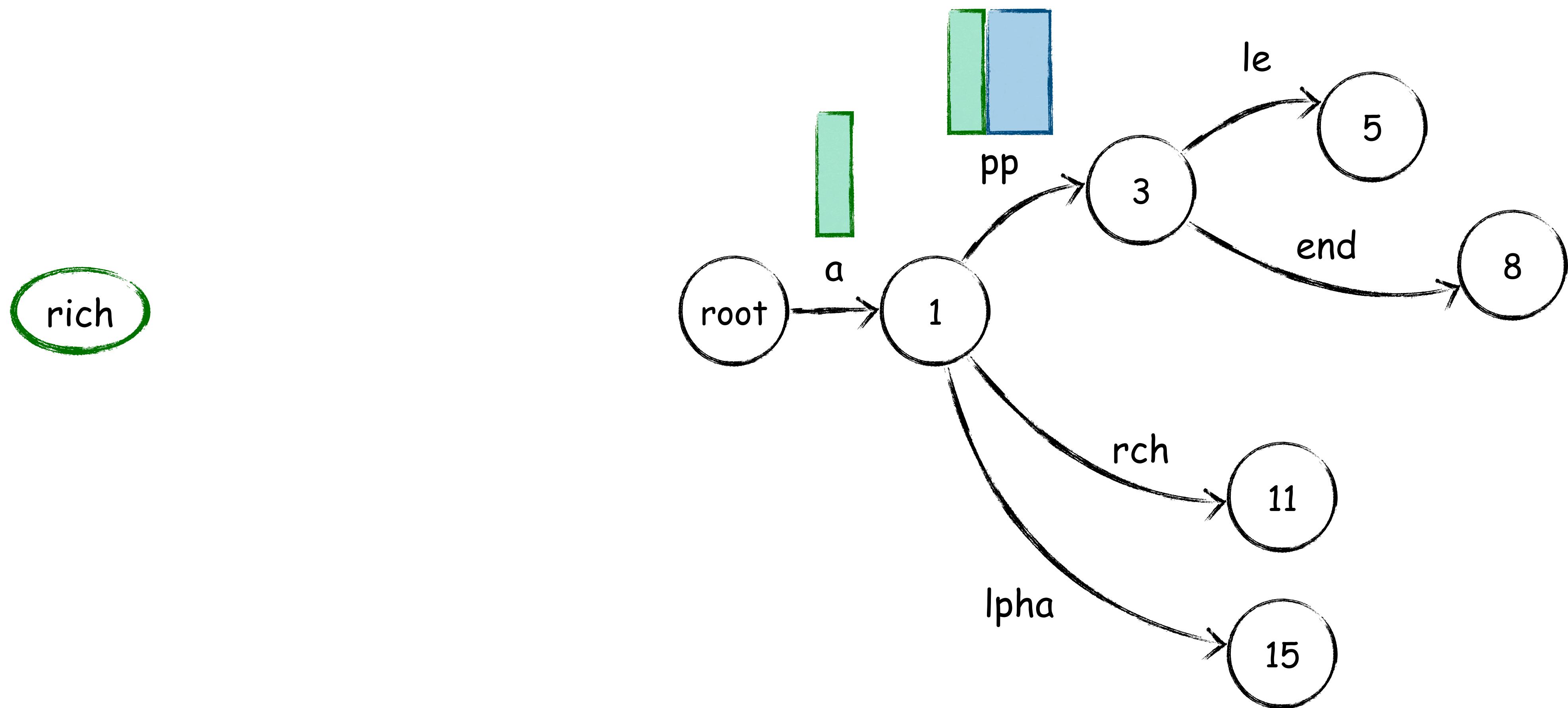
Поиск самого похожего генома



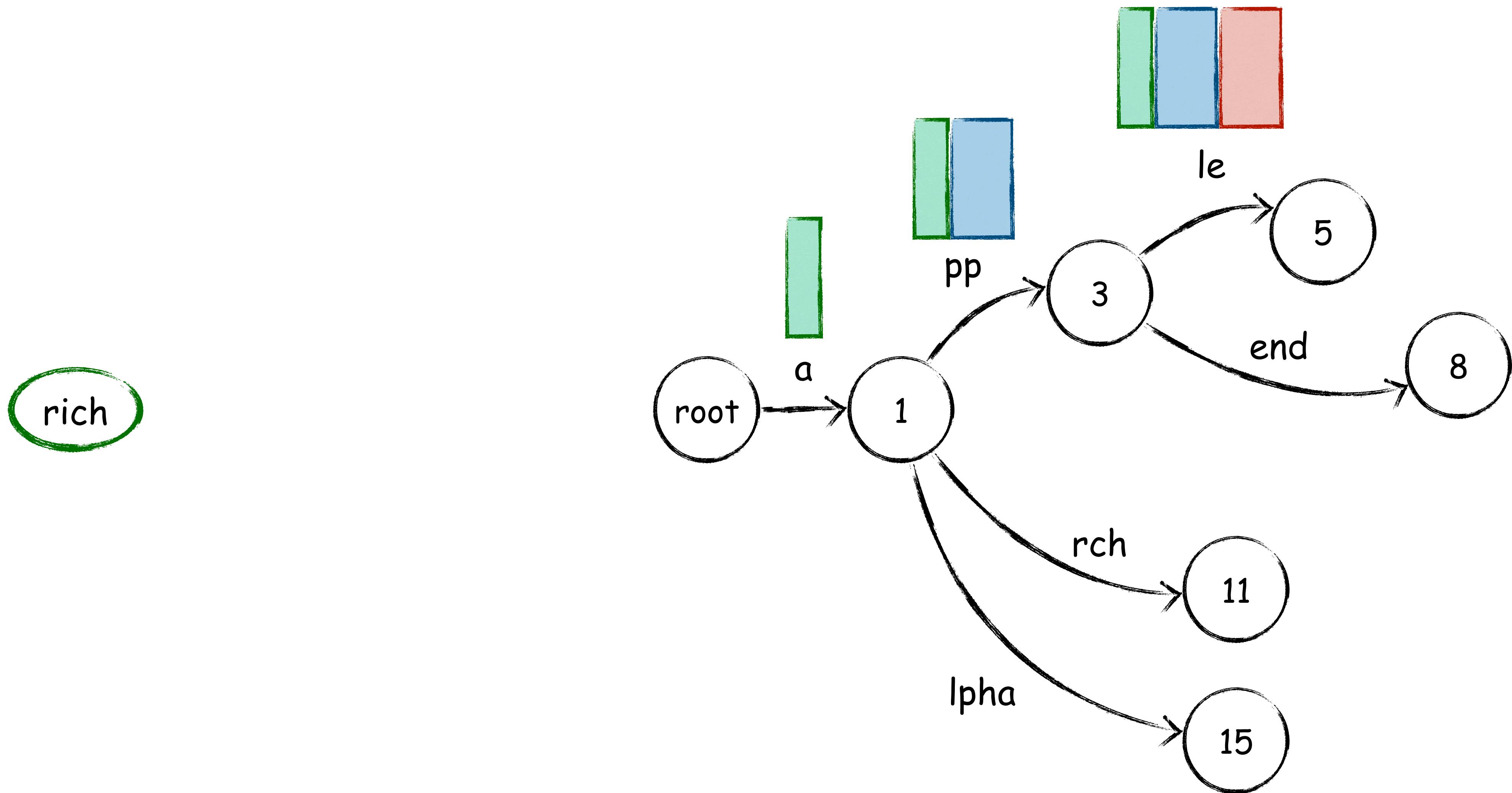
Поиск самого похожего генома



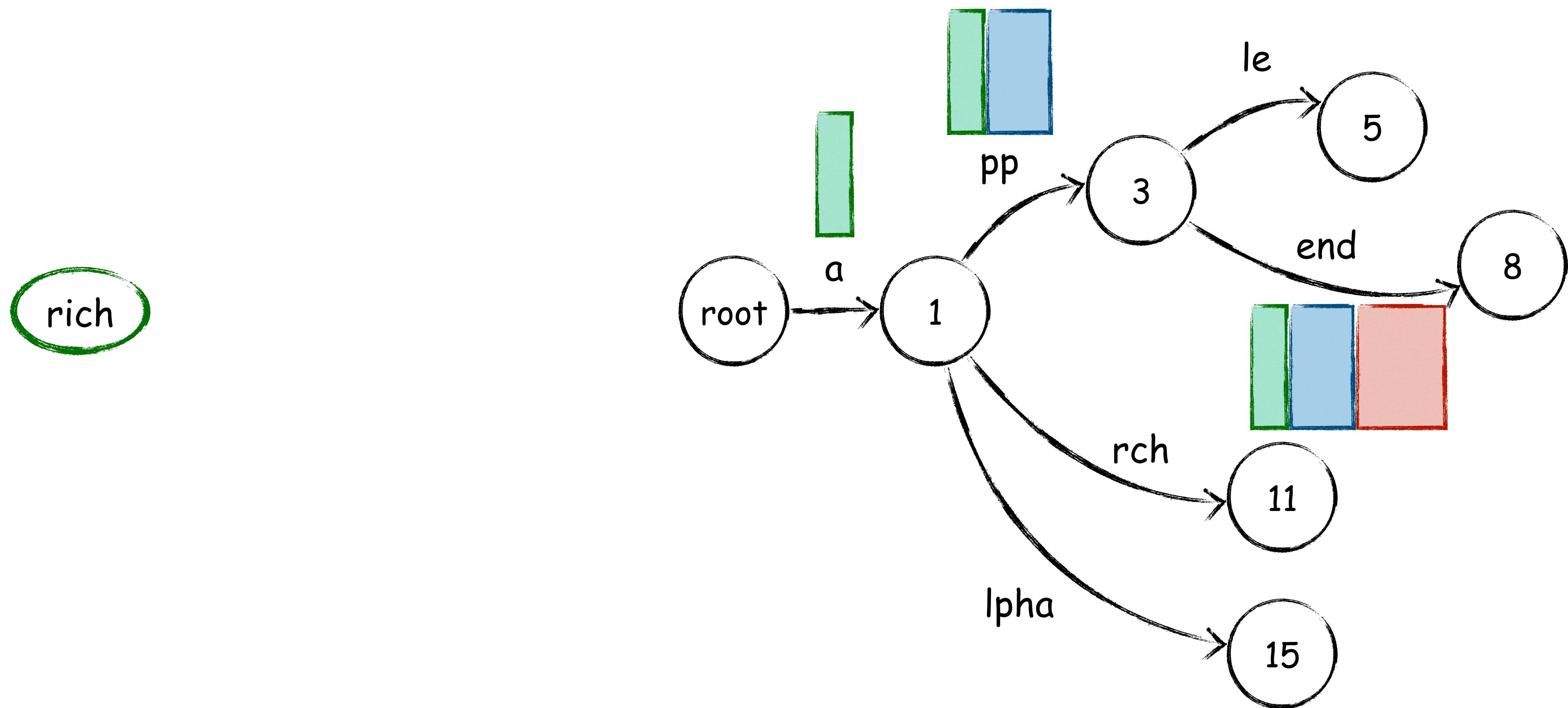
Поиск самого похожего генома



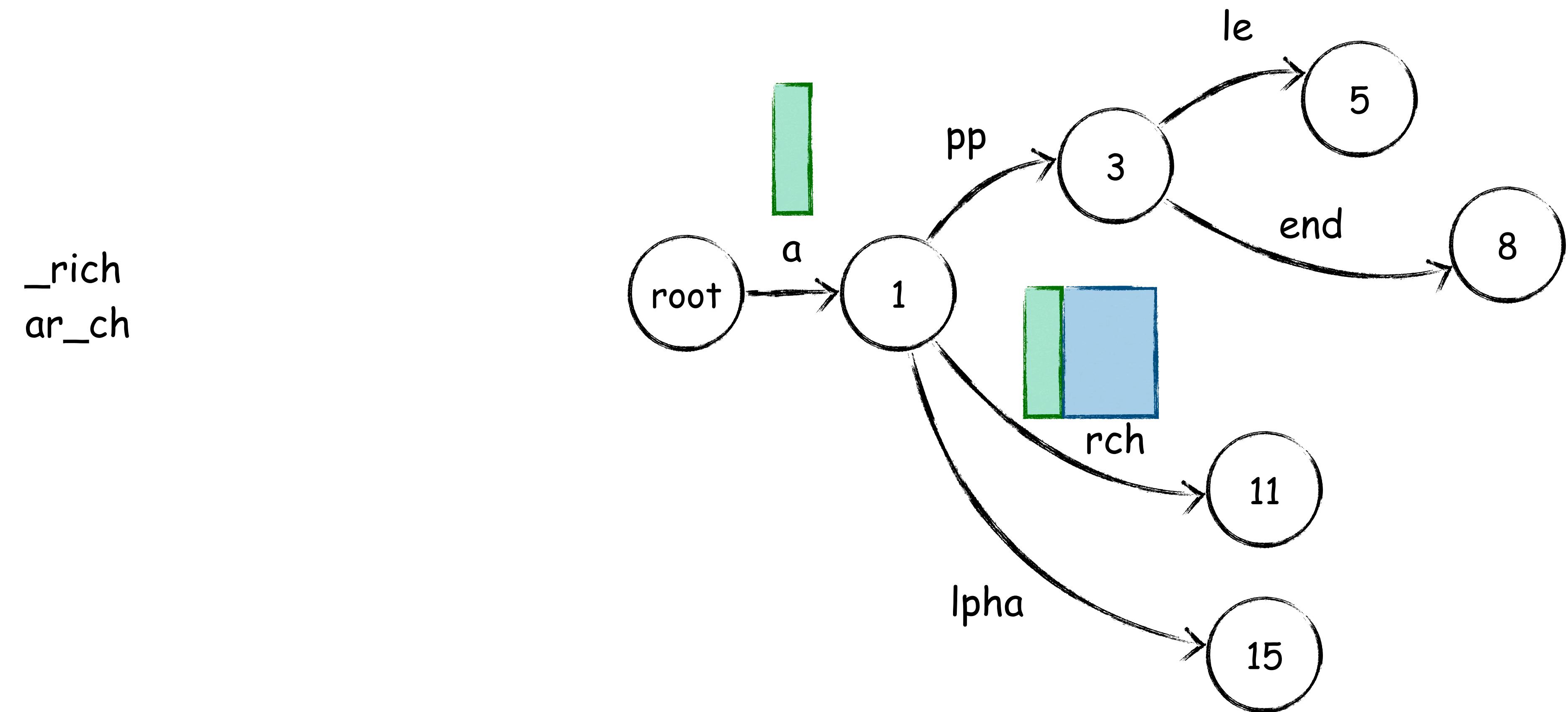
Поиск самого похожего генома



Поиск самого похожего генома



Поиск самого похожего генома



Поиск самого похожего генома

Обход префиксного дерева в глубину + заполнение матрицы расстояний

На каждом шаге выравниваем префикс на ребре с заданной строкой при прямом ходе.

При обратном ходе очищаем неактуальные столбцы матрицы

Поиск самого похожего. Замечания

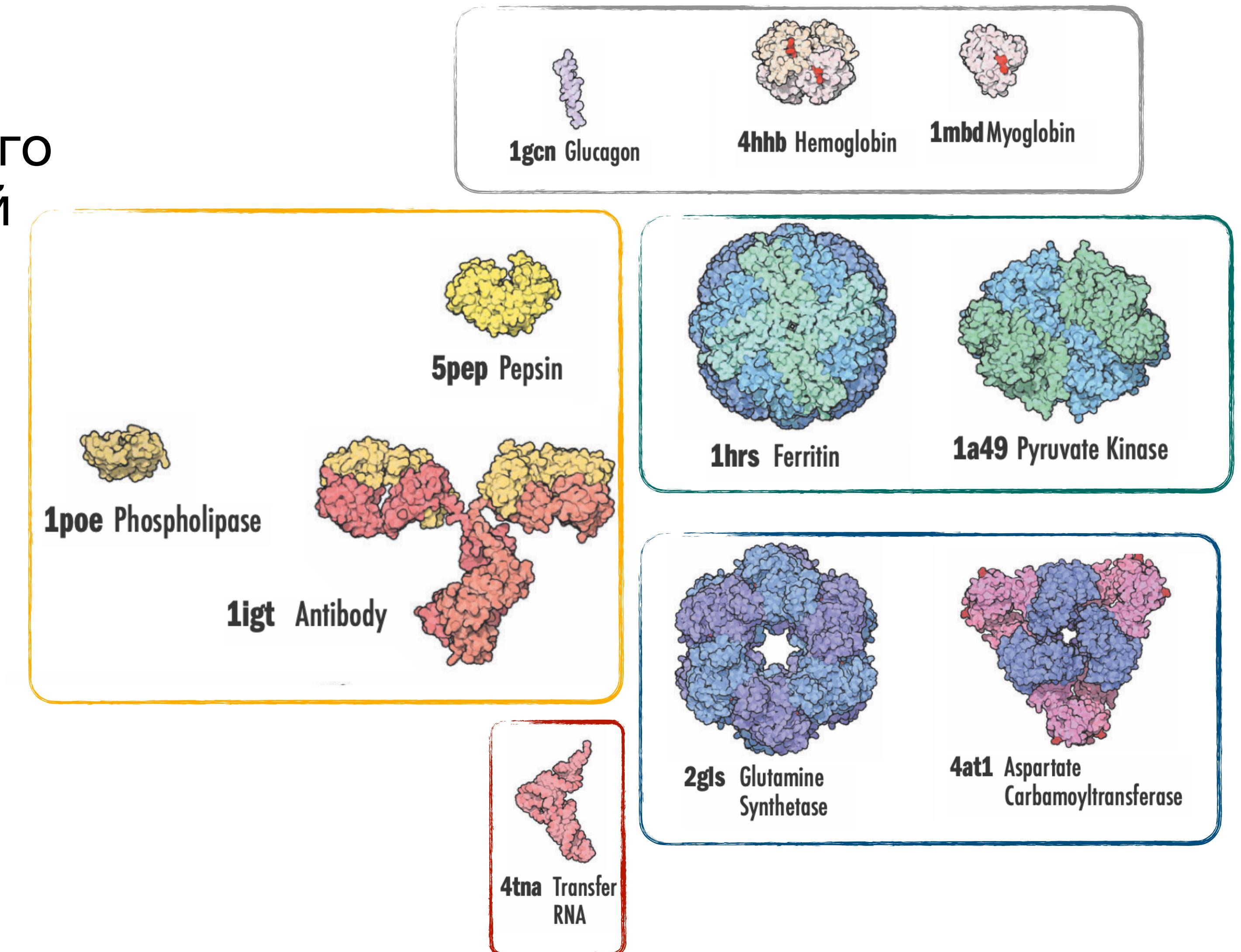
- На каждом шаге можем получить и расстояние и выравнивание
- В узлах можно хранить полученное выравнивание и последний столбец матрицы
- Не зависит от модели выравнивания
- Последовательность запросов тоже можно предварительно превратить в бор

Поиск самого похожего. Замечания

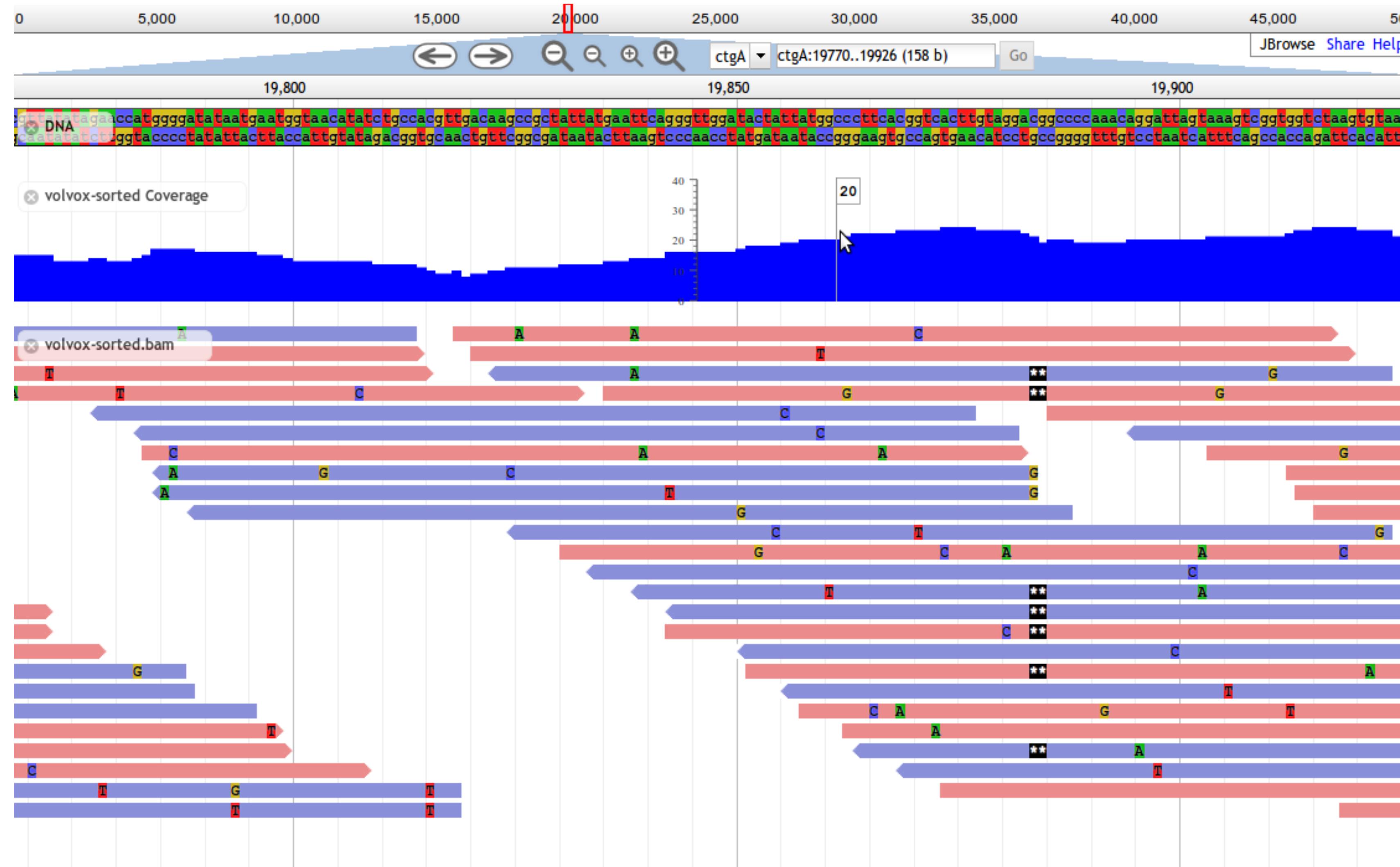
Можно строить бор для каждого кластера последовательностей

Поиск:

- Определение кластера
- Выравнивание на бор



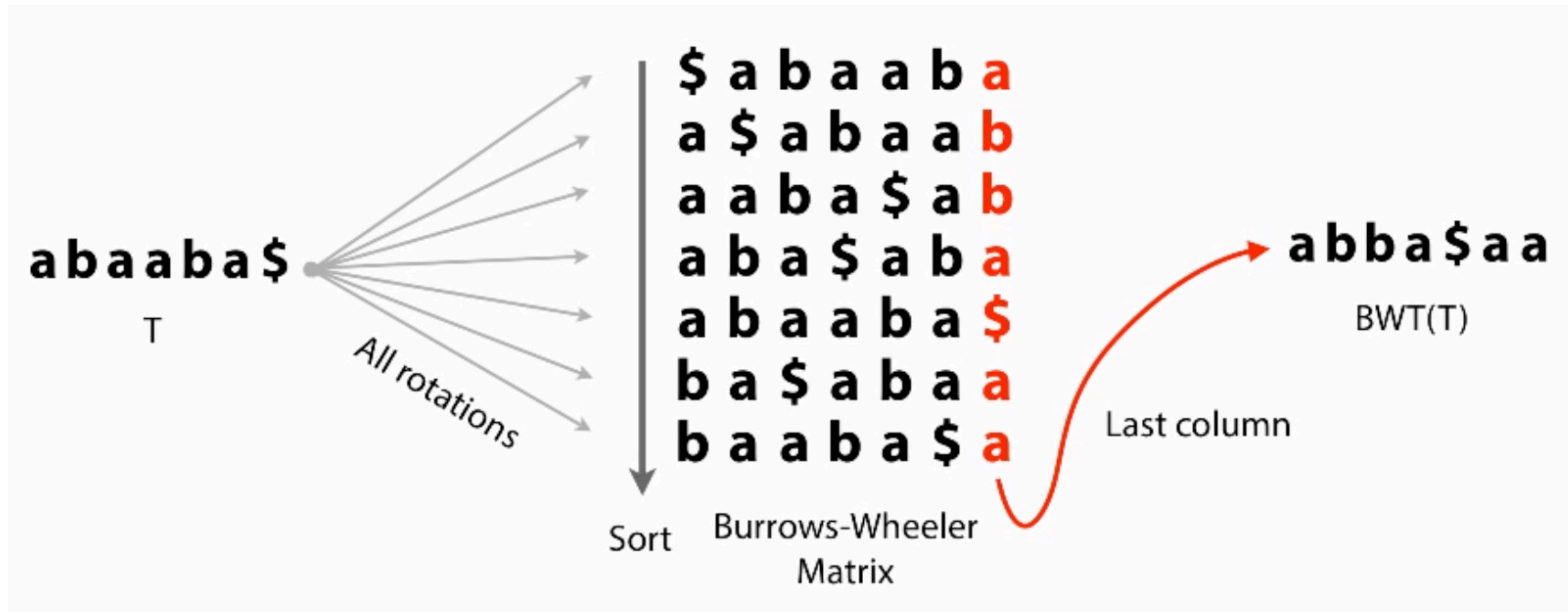
Выравнивание многих строк на одну



Задача поиска подстроки

- Скорость
Достаточную чтобы искать в геноме человека $3 \cdot 10^9$ бп
- Размер
Структура при помощи которой совершаем поиск должна быть компактной

Преобразование Барроуза-Уилера



Преобразование Барроуза-Уилера

Возможный алгоритм получения BWT

```
rotations:: Str -> [Str]
...
bwm:: [Str] -> [Str]
bwm strs = sort strs

bwt:: Str -> Str
bwt str = bwt_help (bwm str) where
    bwt_help (x:xs) = ((last x):(bwt_help xs))
    bwt_help [] = ""
```

BWT

\$ a b a a b a
a \$ a b a a b
a a b a \$ a b
a b a \$ a b a
a b a a b a \$
b a \$ a b a a
b a a b a \$ a

BWM(T)

6	\$
5	a \$
2	a a b a \$
3	a b a \$
0	a b a a b a \$
4	b a \$
1	b a a b a \$

SA(T)

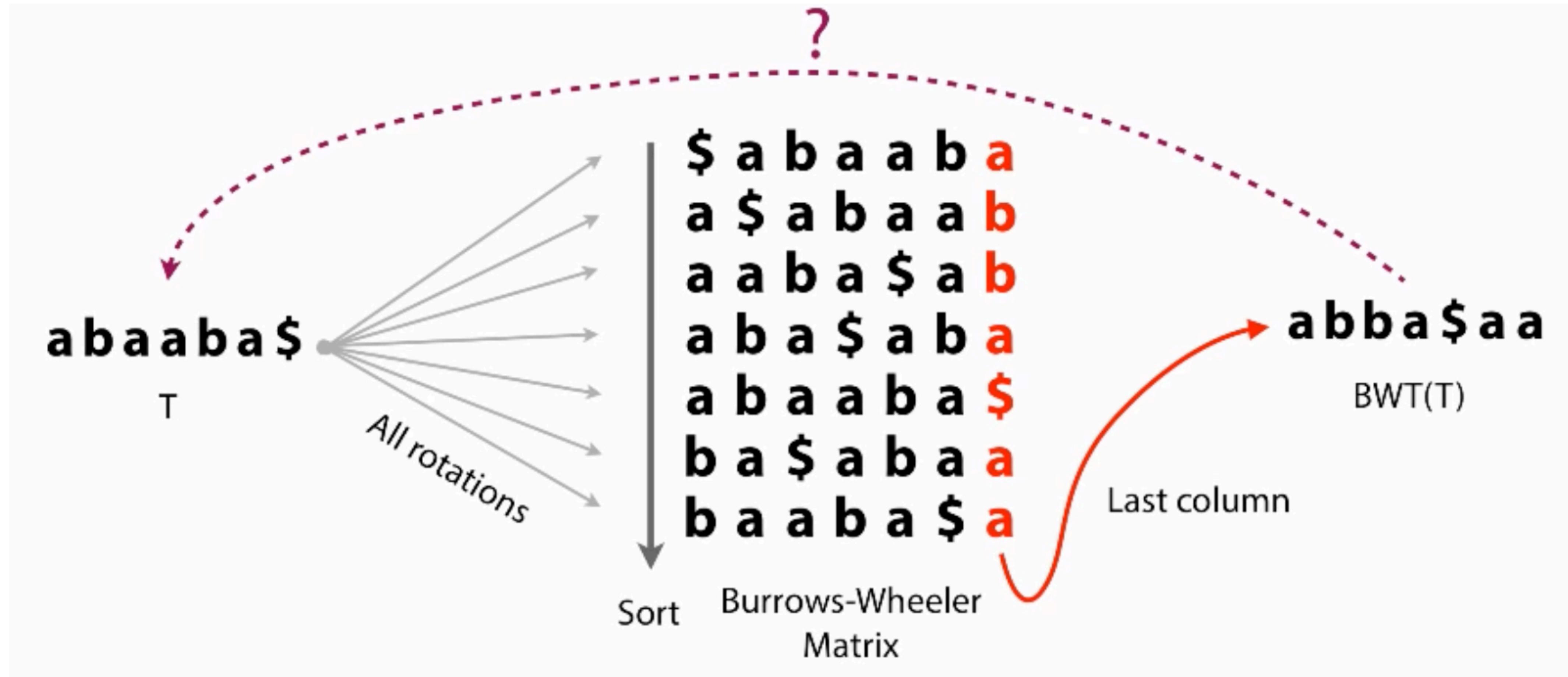
BWT

\$ a b a a b a	6 \$
a \$ a b a a b	5 a \$
a a b a \$ a b	2 a a b a \$
a b a \$ a b a	3 a b a \$
a b a a b a \$	0 a b a a b a \$
b a \$ a b a a	4 b a \$
b a a b a \$ a	1 b a a b a \$

BWM(T) SA(T)

$$BWT_T = \begin{cases} T[SA[i] - 1] : & \text{if } SA[i] > 0 \\ \$: & \text{otherwise} \end{cases}$$

BWT



BWT

\$ a ₀ b ₀ a ₁ a ₂ b ₁ a ₃
a ₃ \$ a ₀ b ₀ a ₁ a ₂ b ₁
a ₁ a ₂ b ₁ a ₃ \$ a ₀ b ₀
a ₂ b ₁ a ₃ \$ a ₀ b ₀ a ₁
a ₀ b ₀ a ₁ a ₂ b ₁ a ₃ \$
b ₁ a ₃ \$ a ₀ b ₀ a ₁ a ₂
b ₀ a ₁ a ₂ b ₁ a ₃ \$ a ₀

BWT matrix + rank

BWT

F	L
\$	a₃
a₃	b₁
a₁	a₀
a₂	a₁
a₀	\$
b₁	a₂
b₀	a₃

BWT matrix + rank

BWT

F	L
\$ a ₀ b ₀ a ₁ a ₂ b ₁	a₃
a₃ \$ a ₀ b ₀ a ₁ a ₂ b ₁	
a₁ a ₂ b ₁ a ₃ \$ a ₀ b ₀	
a₂ b ₁ a ₃ \$ a ₀ b ₀	a₁
a₀ b ₀ a ₁ a ₂ b ₁ a ₃ \$	
b ₁ a ₃ \$ a ₀ b ₀ a ₁	a₂
b ₀ a ₁ a ₂ b ₁ a ₃ \$	a₀

BWT matrix + rank

BWT

	\$	a	b	a	a	b	a ₃
T	a ₃	\$	a	b	a	a	b ₁
	a ₁	a	b	a	\$	a	b ₀
	a ₂	b	a	\$	a	b	a ₁
	a ₀	b	a	a	b	a	\$
	b ₁	a	\$	a	b	a	a ₂
	b ₀	a	a	b	a	\$	a ₀

	\$	a	b	a	a	b	a ₃
	a ₃	\$	a	b	a	a	b ₁
	a ₁	a	b	a	\$	a	b ₀
	a ₂	b	a	\$	a	b	a ₁
	a ₀	b	a	a	b	a	\$
	b ₁	a	\$	a	b	a	a ₂
	b ₀	a	a	b	a	\$	a ₀

Почему ранк действительно совпадает в L и F?

БВТ

<i>F</i>	<i>L</i>
\$ a ₃ b ₁ a ₁ a ₂ b ₀ a ₀	
a ₀ \$ a ₃ b ₁ a ₁ a ₂ b ₀	
a ₁ a ₂ b ₀ a ₃ \$ a ₃ b ₁	
a ₂ b ₀ a ₀ \$ a ₃ b ₁ a ₁	
a ₃ b ₁ a ₁ a ₂ b ₀ a ₀ \$	
b ₀ a ₀ \$ a ₃ b ₁ a ₁ a ₂	
b ₁ a ₁ a ₂ b ₀ a ₀ \$ a ₃	

Поправим ранк чтобы он монотонно возрастал в L

BWT

<i>F</i>	<i>L</i>
\$ a ₃ b ₁ a ₁ a ₂ b ₀	a ₀
a ₀ \$ a ₃ b ₁ a ₁ a ₂	b ₀
a ₁ a ₂ b ₀ a ₃ \$ a ₃	b ₁
a ₂ b ₀ a ₀ \$ a ₃ b ₁	a ₁
a ₃ b ₁ a ₁ a ₂ b ₀ a ₀	\$
b ₀ a ₀ \$ a ₃ b ₁ a ₁	a ₂
b ₁ a ₁ a ₂ b ₀ a ₀ \$	a ₃

Поправим ранк чтобы он монотонно возрастал в *L*

Преобразование Барроуза-Уилера

Как теперь имея L определить, где будет стоять ее символ в F?

L

a₀

b₀

Найдем b_1

b₁

Пропустим все \$

a₁

Пропустим все a

\$

И пропустим 1 b

a₂

a₃

Преобразование Барроуза-Уилера

Как теперь имея L определить, где будет стоять ее символ в F?

F	L	
\$	a_0	
a_0	b_0	Найдем b_1
a_1	b_1	Пропустим все \$
a_2	a_1	Пропустим все a
a_3	\$	И пропустим 1 b
b_0	a_2	
 b_1	a_3	

Преобразование Барроуза-Уилера

Но как пользуясь всем этим получить исходную строку обратно?

F	L
\$	a_0
a_0	b_0
a_1	b_1
a_2	a_1
a_3	\$
b_0	a_2
b_1	a_3

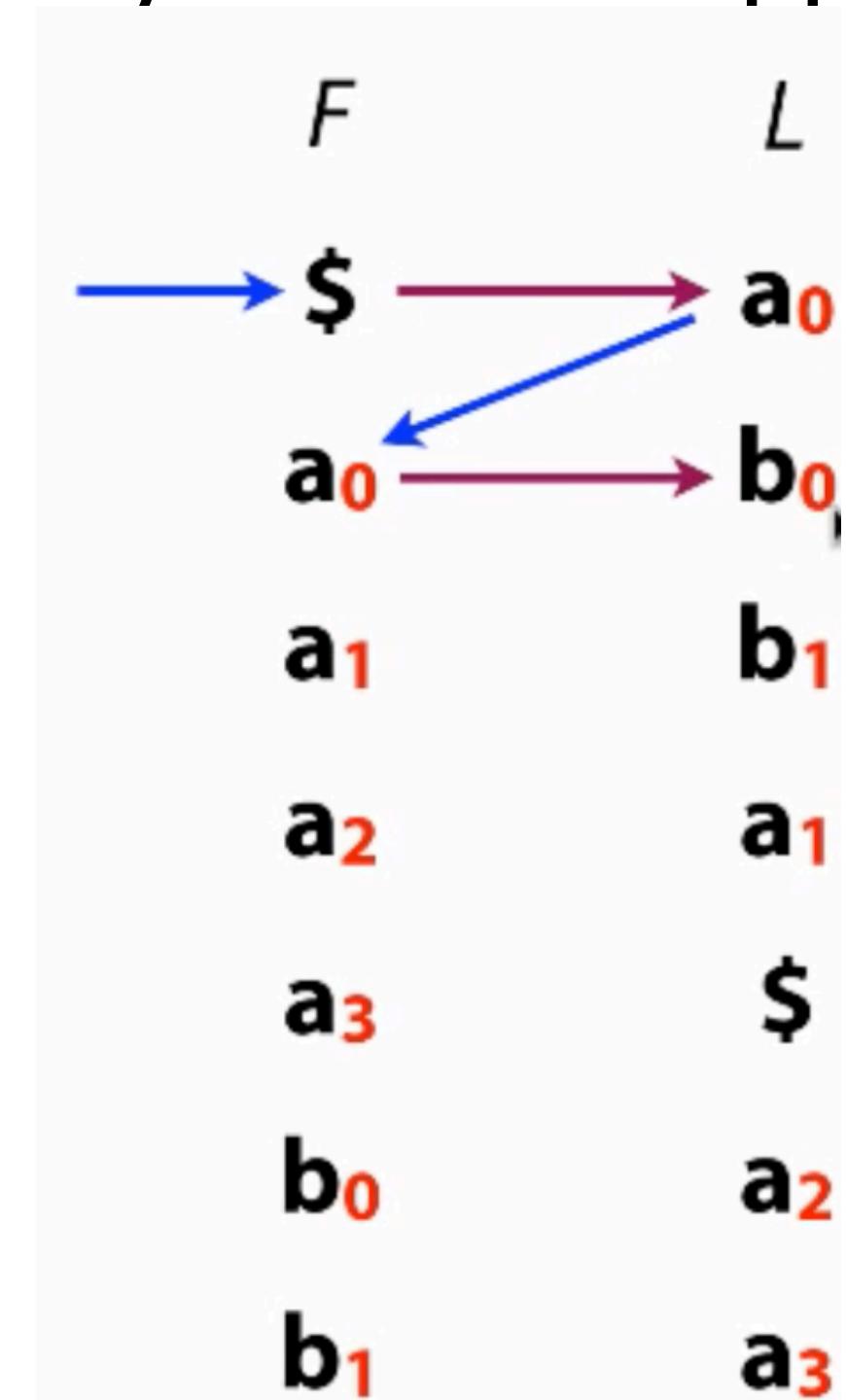
Преобразование Барроуза-Уилера

Но как пользуясь всем этим получить исходную строку обратно?

F	L
→ \$	→ a_0
a_0	b_0
a_1	b_1
a_2	a_1
a_3	\$
b_0	a_2
b_1	a_3

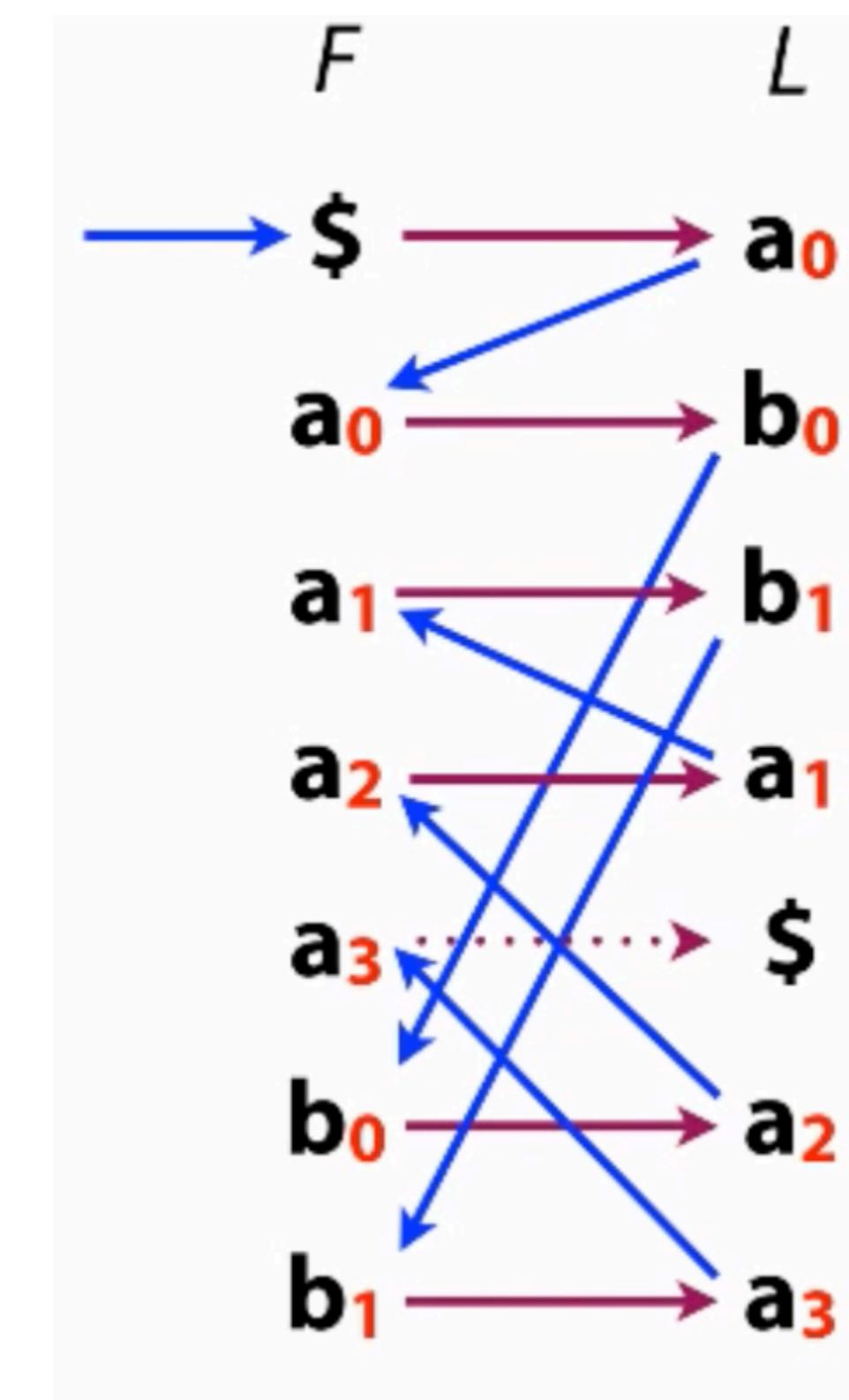
Преобразование Барроуза-Уилера

Но как пользуясь всем этим получить исходную строку обратно?



Преобразование Барроуза-Уилера

Но как пользуясь всем этим получить исходную строку обратно?



$$a_3 b_1 a_1 a_2 b_0 a_0 \$ = T$$

Индекс Барроуза-Уилера

Как пользоваться ВWT как индексом?

Найдем все вхождения строки в заданную

$P = \mathbf{aba}$	
F	L
\$	a b a a b a₀
a₀	\$ a b a a b₀
a₁	a b a \$ a b₁
a₂	b a \$ a b a₁
a₃	b a a b a \$
b₀	a \$ a b a a₂
b₁	a a b a \$ a₃

Индекс Барроуза-Уилера

Как пользоваться ВWT как индексом?

Найдем все вхождения строки в заданную

		$P = \mathbf{aba}$
F	L	
\$	a b a a b	a₀
a₀	\$ a b a a	b₀
a₁	a b a \$ a	b₁
a₂	b a \$ a b	a₁
a₃	b a a b a	\$
b₀	a \$ a b a	a₂
b₁	a a b a \$	a₃

Индекс Барроуза-Уилера

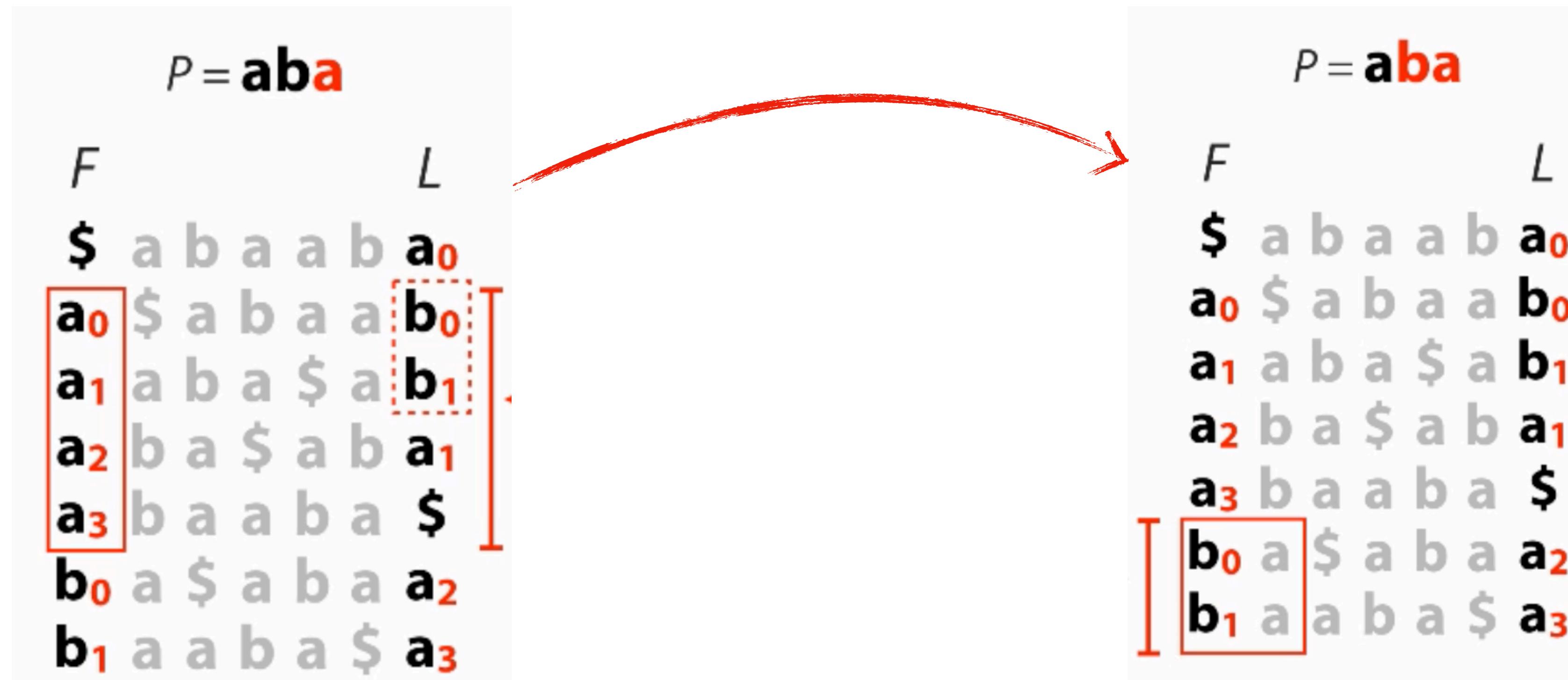
Как пользоваться ВWT как индексом?

Найдем все вхождения строки в заданную



Индекс Барроуза-Уилера

Как пользоваться BWT как индексом?
Найдем все вхождения строки в заданную



Индекс Барроуза-Уилера

Как пользоваться ВWT как индексом?

Найдем все вхождения строки в заданную

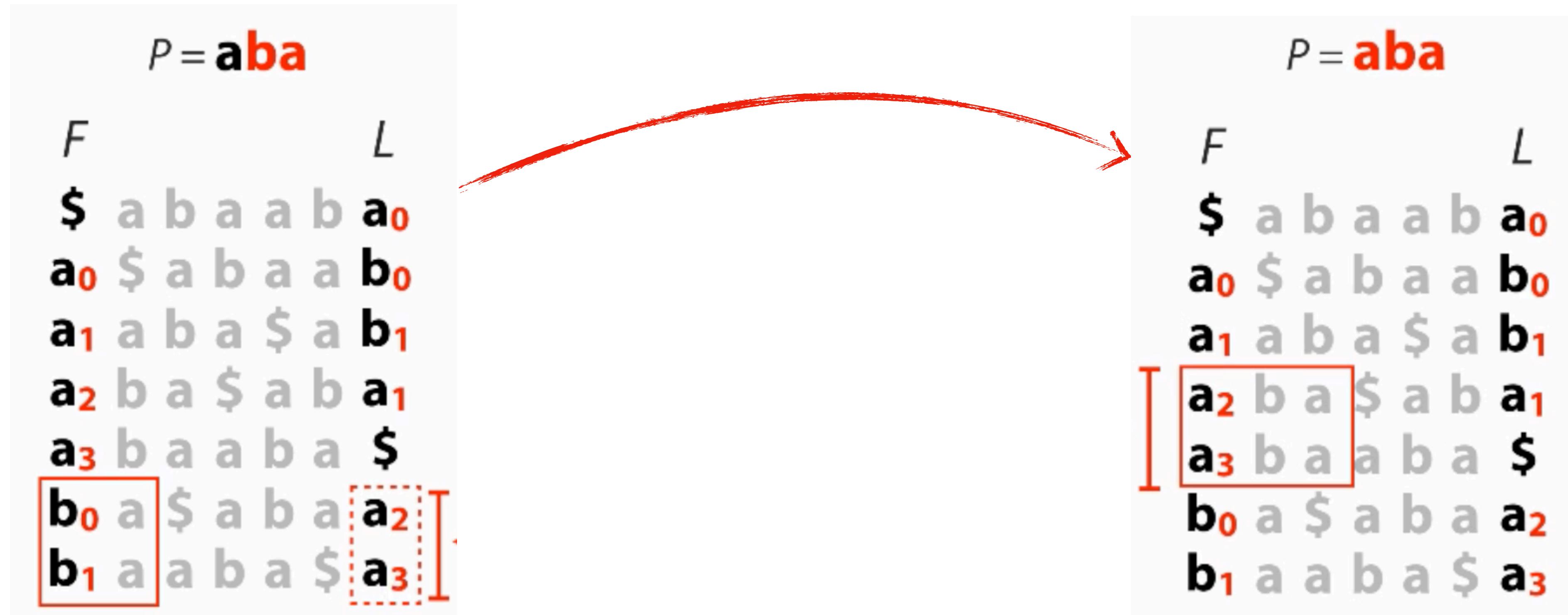
$$P = \mathbf{aba}$$

F	L
\$	a b a a b a ₀
a ₀	\$ a b a a b ₀
a ₁	a b a \$ a b ₁
a ₂	b a \$ a b a ₁
a ₃	b a a b a \$
b ₀	a \$ a b a a ₂]
b ₁	a a b a \$ a ₃]

Индекс Барроуза-Уилера

Как пользоваться ВWT как индексом?

Найдем все вхождения строки в заданную



Индекс Барроуза-Уилера

Но у нас есть 2 проблемы

- Искать в окне колонки F долго
- Хотелось бы знать где именно находятся совпадения

Индекс Барроуза-Уилера

F	L	a	b
\$	a	1	0
a	b	1	1
a	b	1	2
a	a	2	2
a	\$	2	2
b	a	3	2
b	a	4	2

Достаточно проверить эти строки матрицы, чтобы понять что в окно попали b_0, b_1

Индекс Барроуза-Уилера

- Сложность по времени:

$$482 + 2 - 1 = 483$$

482 — чекпоинт

2 — # а

-1 — т.к. ранк с 0

$$439 - 2 - 1 = 436$$

L	a	b
:	:	
a	482	432
b		
b		
a		
a		
a		
b		
a		
a		
b		
b		
a		
a		
a		
b		
b		
b		
a	488	439
a		

Индекс Барроуза-Уилера

- Сложность по времени:
 $O(1)$
- По памяти:

$$482 + 2 - 1 = 483$$

482 — чекпоинт

2 — # а

-1 — т.к. ранк с 0

$$439 - 2 - 1 = 436$$

L	a	b
:	:	
a	482	432
b		
b		
a		
a		
a		
b		
a		
a		
a		
b		
b		
b		
b		
a		
a		
a		
b		
b		
b		
488	439	
a		

Индекс Барроуза-Уилера

- Сложность по времени:
 $O(1)$
- По памяти:
 $O(n)$ но константа < 1 :)

$$482 + 2 - 1 = 483$$

482 — чекпоинт

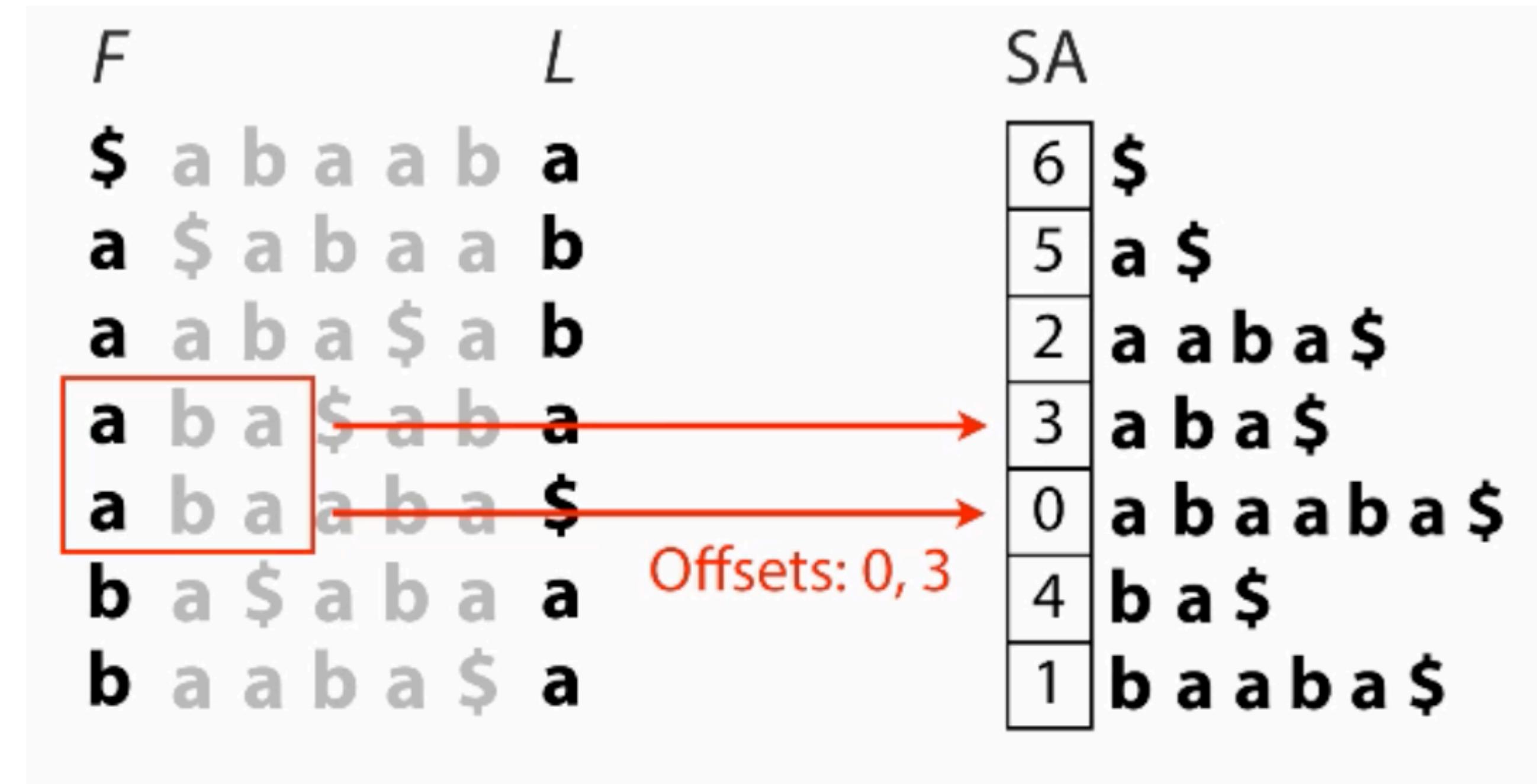
2 — # а

-1 — т.к. ранк с 0

$$439 - 2 - 1 = 436$$

L	a	b
:	:	
a	482	432
b		
b		
a		
a		
a		
b		
a		
a		
b		
b		
a		
a		
a		
b		
b		
a		
488	439	
a		

Индекс Барроуза-Уилера



Индекс Барроуза-Уилера

Что в итоге хранится в индексе?

- F – храним как словарь $|\Sigma|$
- L – m символов
- SA – m чисел
- Чекпоинты - $m |\Sigma| b$, где $b \leq 1$

Индекс Барроуза-Уилера

Что в итоге хранится в индексе?

- F – храним как словарь $|\Sigma|$
- L – m символов
- SA – m чисел
- Чекпоинты - $m |\Sigma| b$, где $b \leq 1$

Геном человека, пусть $a = 1/128$

- 16 bytes
- $2bits * 3 * 10^9 chars = 750 MB$
- $3 * 10^9 chars * 4bytes/char = 12800 MB$
- $3 * 10^9 * 4bytes/128 = 100 MB$

Менее 15 GB

Идея выравнивания Барроуза-Уилера

Как воспользоваться для выравнивания?

- Найдем достаточно большие сиды
- Расширим их за счет обычных алгоритмов выравнивания

Наиболее известная реализация – bwa (Burrows-Wheeler Aligner)

Резюмируем

- Для поиска гомологичных белков важно иметь предпосчитанные структуры данных по их последовательностям.
- Для хранения индекса Барроуза-Уилера требуется не много места
- При помощи индекса Барроуза-Уилера можно быстро искать все вхождения
- ВWT можно использовать для выравнивания