Биоинформатика.
Некоторые алгоритмические задачи

Поиск сходства

Мотивация 1.

- Похожие последовательности, ***наверное***, выполняют похожие *биологические* функции.
- Поиск сходства может позволить предсказать функции белков (генов)

Два цитохрома С из двух разных бактерий похожи(?)

Flavobacterium johnsoniae
PVK**GK**E**LF**NA**NC**AA**CH**K**LD**A**K**ST**GPALRG**VVH**NS**SDM**I**K**S**
VSA**GK**T**LF**DT**NC**KT**CH**R**LD**T**K**LV**GPALRG**AIK**NS**QAL**I**A**S**
Algoriphagus machipongonensis

Поиск сходства

Мотивация 2.

- Похожие последовательности, ***наверное***, эволюционно связаны (гомолоичны).
- Анализ сходства может позволить исследовать пути эволюции

Эволюцию мы, как правило, не знаем.

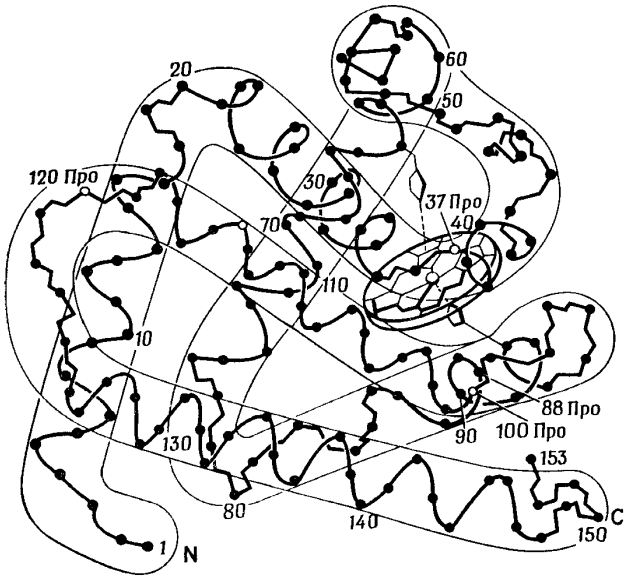
Исключения:

- Вирусы, заразившие одного человека.
- Специальные эксперименты по длительному культивированию бактерий

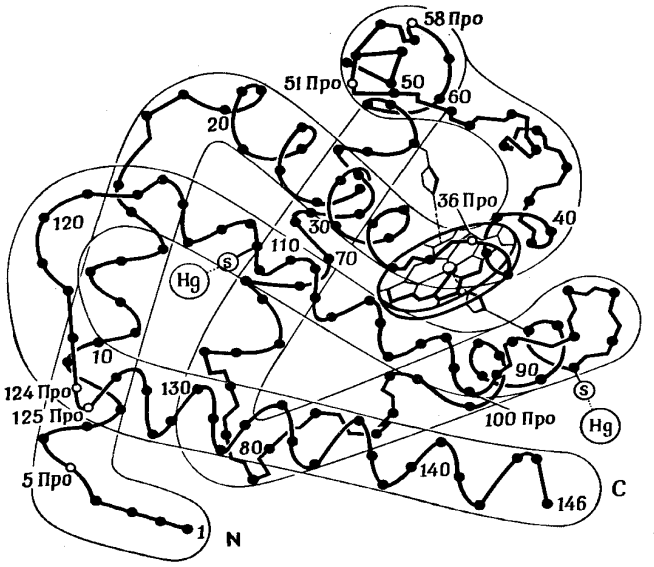
Поиск сходства

Мотивация 3.

- Похожие последовательности, наверное, соответствуют похожим пространственным структурам
- Анализ сходства может позволить предсказывать пространственный структуру



Миоглобин человека



β-глобин человека

Поиск сходства

Определение сходства

- Элементарные операции преобразования текста:
  - заменить символ
  - вставить символ
  - удалить символ
- Редакционное расстояние **минимальное** количество операций, которые преобразуют одну строку в другую (расстояние Левенштейна).

Поиск сходства

Пример

	химия →	биология
1.	имия	х(1)
2.	мия	и(3)
3.	ия	м(5)
4.	я	и(6)
5.		я(7)
6.	б	+б
7.	би	+и
8.	био	+о
9.	биол	+л
10.	биоло	+о
11.	биолог	+г
12.	биологи	+и
13.	биология	+я

	химия →	биология	
1.	бимия	х(1)→	б
2.	биоия	м(3)→	о
3.	биоля	и(4)→	л
4.	биоло	л(7)→	р
5.	биолог	+г	
6.	биологи	+и	
7.	биология	+я	

	химия →	биология	
1.	бимия	х(1)→	б
2.	биомия	+о(3)	
3.	биолмия	+л(4)	
4.	биоломия	+о(5)	
5.	биология	м(6)→	г

- Минимально ли число операций?
- Как мы делали?
- Как это объяснить компьютеру?

Поиск сходства

- ***Префикс*** слова – начальная часть слова

Префиксы

химия — *тоже префикс*

хими

хим

хи

х

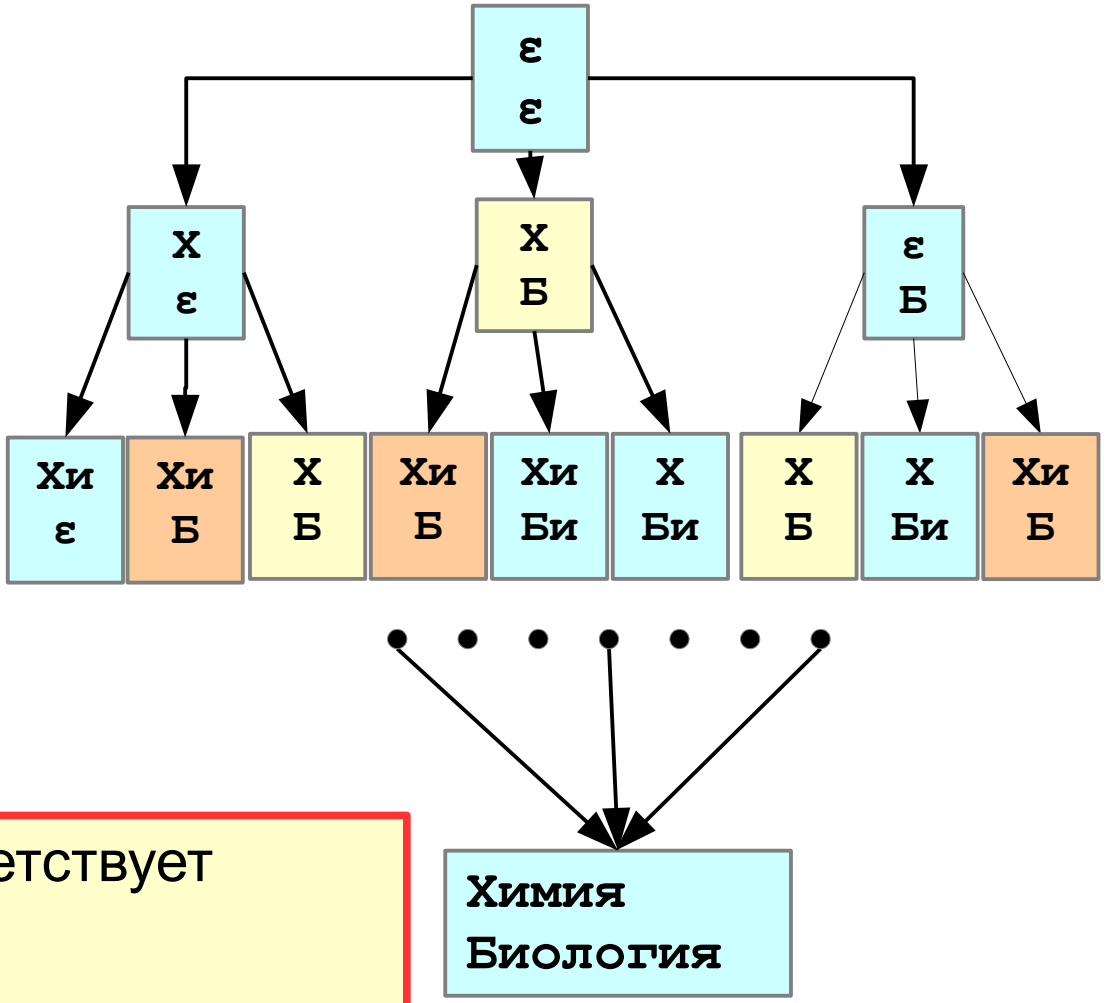
ε — ***пустое слово***

Поиск сходства

Вместо редактирования будем ***порождать*** пару последовательностей.

Операции:

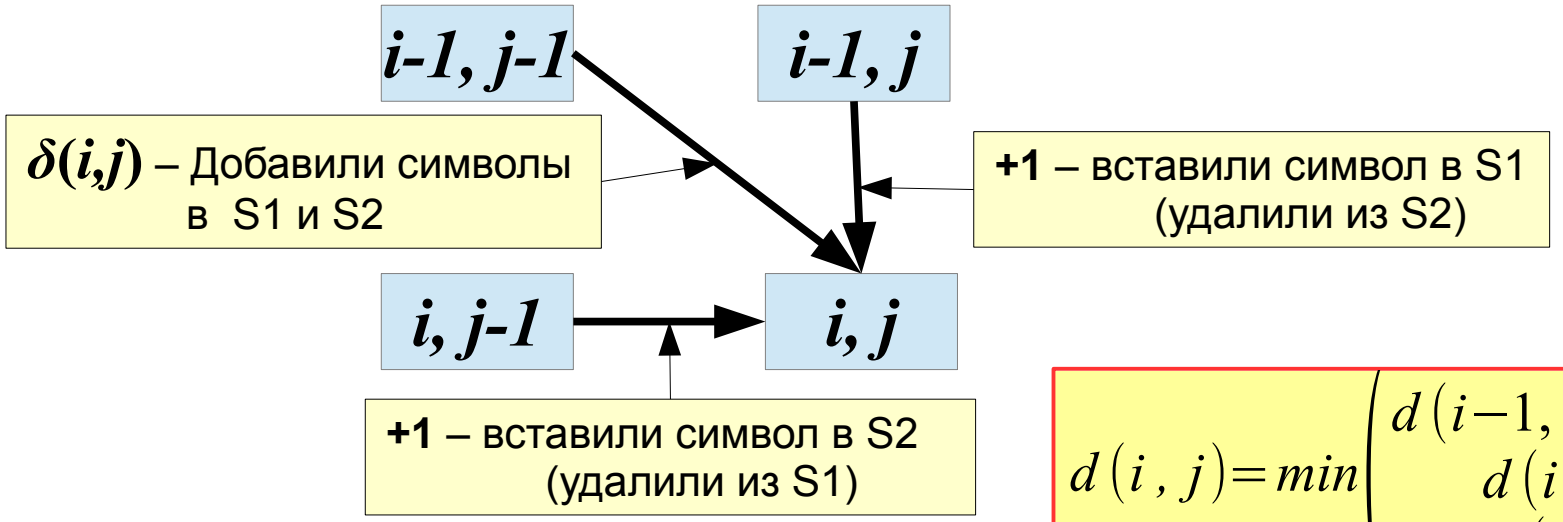
- Добавить очередной символ в первую строку (эквивалентно удалению)
- Добавить очередной символ во вторую строку (эквивалентно вставке)
- Добавить разные очередные символы в две строки (эквивалентно замене)
- Добавить одинаковые очередные символы в две строки (нет редакционной операции)



- Каждому пути по дереву соответствует последовательность операций
- Каждому узлу соответствует пара префиксов

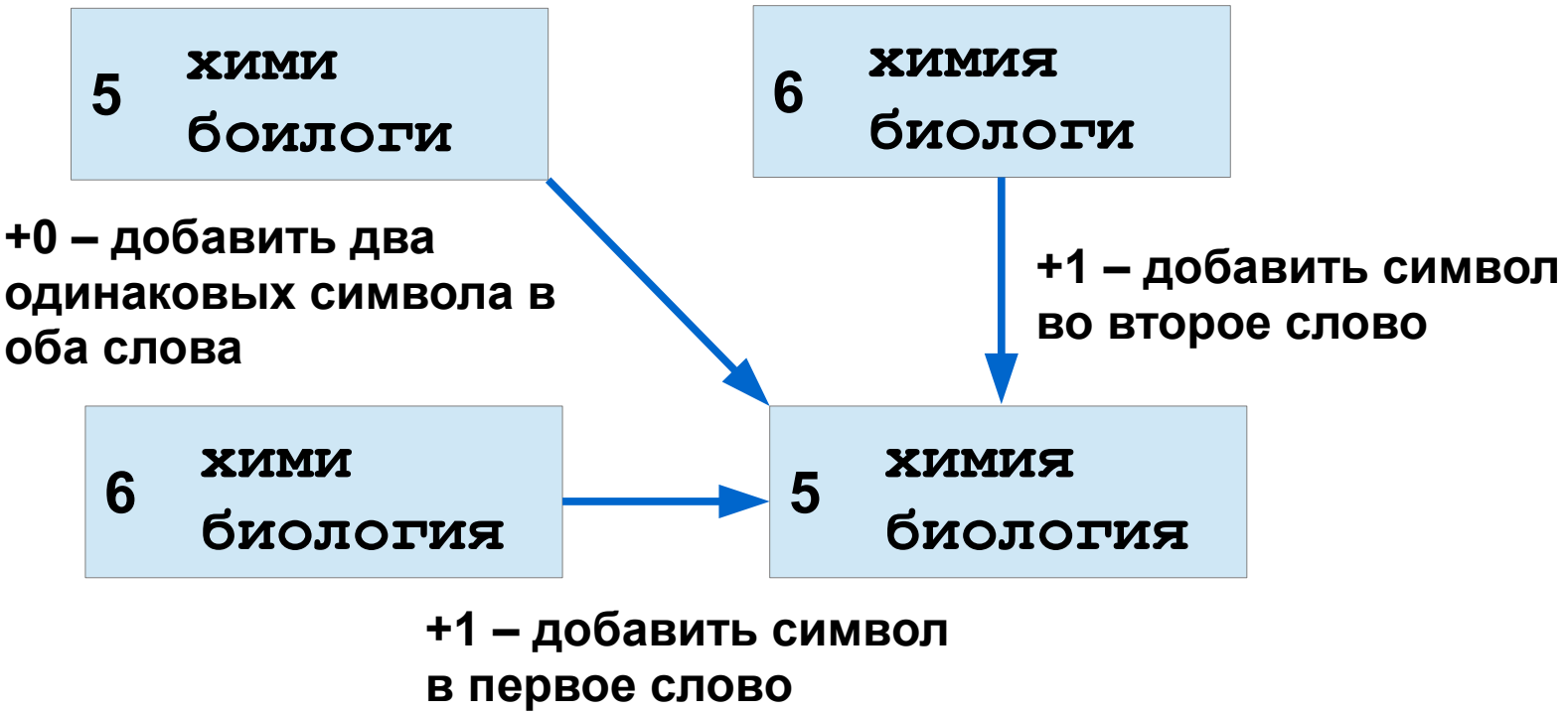
Поиск сходства

- Префиксы можно пронумеровать.
- Пара префиксов — два целых числа ***(i,j)*** → получаем матрицу префиксов.
- В ячейке матрицы пишем редакционное расстояние между префиксами ***(i,j)***
- Зная редакционные расстояния для префиксов меньшего размера можно определить редакционное расстояние для ***(i,j)***



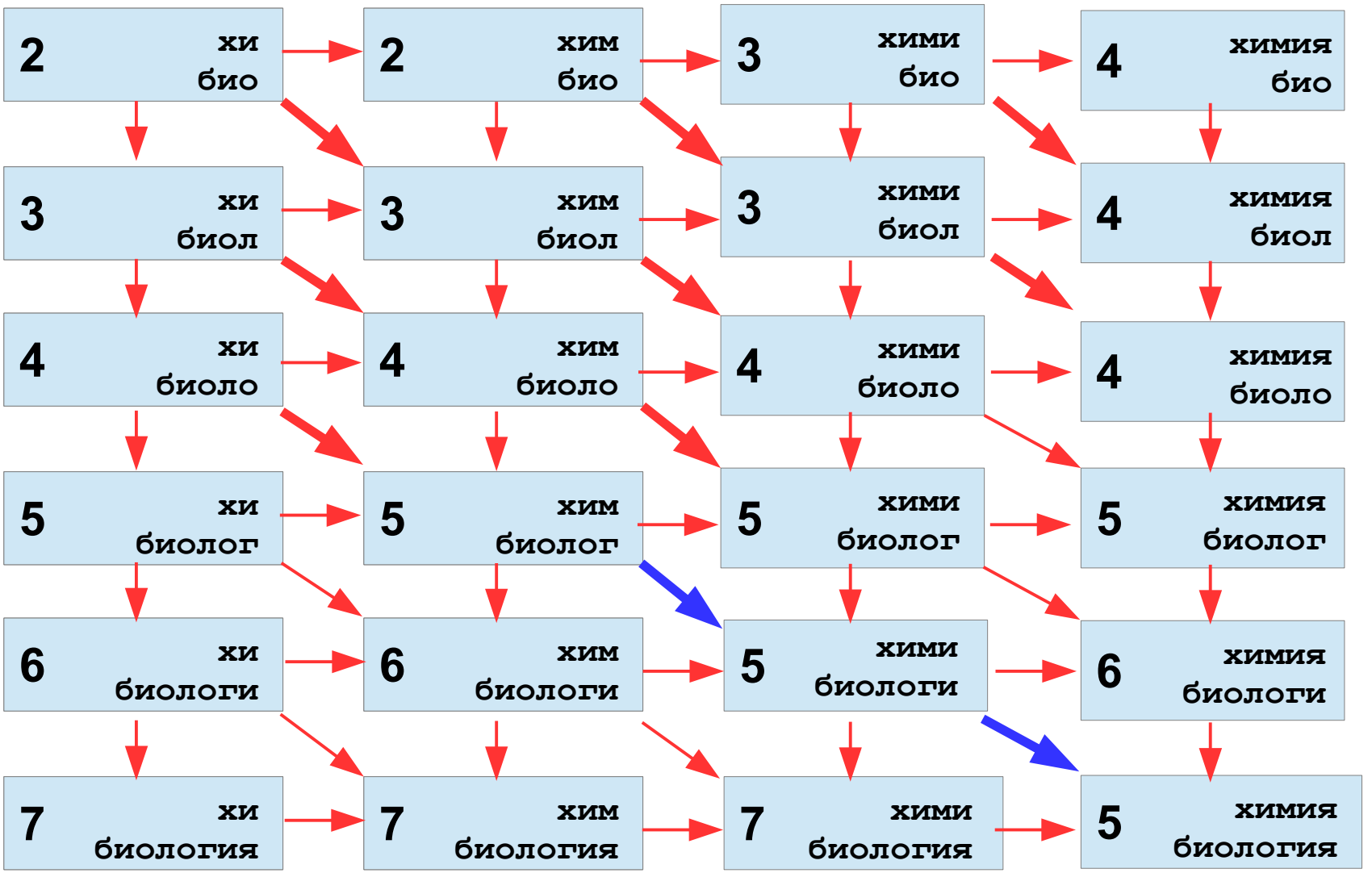
$$d(i,j) = min\begin{pmatrix} d(i-1,j-1)+\delta(i,j) \\ d(i,j-1)+1 \\ d(i-1,j)+1 \end{pmatrix}$$

Поиск сходства



$$d\begin{pmatrix}химия\\биология\end{pmatrix}=min\begin{pmatrix}d\begin{pmatrix}хими\\биологи\end{pmatrix}+0\\d\begin{pmatrix}хими\\биология\end{pmatrix}+1\\d\begin{pmatrix}химия\\биологи\end{pmatrix}+1\end{pmatrix}=min\begin{pmatrix}5\\7\\7\end{pmatrix}=5$$

Поиск сходства



Поиск сходства

	ε	б	и	о	л	о	г	и	я
ε	**0**	1	2	3	4	5	6	7	8
х	1	**1**	2	3	4	5	6	7	8
и	2	2	**1**	2	3	4	5	**6**	7
м	3	3	3	2	3	4	**5**	6	7
и	4	4	**3**	4	3	4	5	**5**	6
я	5	5	5	3	3	4	5	6	**5**

$$d(i,j)=min\begin{pmatrix} d(i-1,j-1)+\delta(i,j) \\ d(i,j-1)+1 \\ d(i-1,j)+1 \end{pmatrix}$$

```
хи---мия
 |    ||
биология
```

Выравнивание

Поиск сходства

- Выравнивание – способ написать последовательности друг под другом так, чтобы ***гомологичные*** буквы были сопоставлены
- ***Гомологичные*** буквы — буквы с общим происхождением
- ~~***Процент гомологии***~~ = процент беременности

```
QEGQEIFNKSCIGCHAVGSNDSRPPSARIAPNLANFADRDMVAGIAENNEEN
  ** **   * *****             ** *     *   *  *
LRGQRIFADRCAGCHAVRGTGAAGTQ...APDLTHVGARRLLAAGALANTPD
```

- Мы не знаем наверняка историю.
- Золотой стандарт — выравнивание трехмерных структур (***гипотеза*** — во время эволюции аминокислоты не меняют своего положения на структуре)

Поиск сходства

Задача

- Придумать алгоритм, который лучше всего воспроизводит золотой стандарт

Качество выравнивания

- Редакционное расстояние — уровень различия
- Степень сходства — удобнее и допускает обобщения

Алгоритм

- Поиск выравнивания с наибольшей степенью сходства

Поиск сходства

Матрица замен – отражает частоту фиксации замен

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

$$m(a,b) = \log\left(\frac{p(a,b)}{p(a)\,p(b)}\right)$$

Поиск сходства

Needleman–Wunsch - Почти настоящий алгоритм выравнивания

	ε	M	A	G	G	F	I	M	K
ε	**0**	-1	-2	-3	-4	-5	-6	-7	-8
M	-1	3	-2	-3	-4	-5	-4	-4	-5
A	-2	-4	5	4	3	2	1	0	-1
L	-3	-1	-2	3	3	4	5	4	3
L	-4	-2	-3	-4	1	4	5	7	6
R	-5	-3	-3	-4	-5	3	4	6	9

$$d(i,j) = max \begin{pmatrix} d(i-1,j-1) + M(S_i^1, S_j^2) \\ d(i,j-1) - d \\ d(i-1,j) - d \end{pmatrix}$$

$$T = O(n \cdot m)$$

$M(a,b)$ – матрица замен

d — штраф за делецию

```
MAGGFIMK
**    ||*
MA---LLR
```

выравнивание

Поиск сходства

Варианты

1. Локальное выравнивание такое выравнивание, где нет штрафов за концевые делеции и несовпадения

Локальное выравнивание = выравнивание фрагментов последовательностей с наибольшим весом

2. Штрафы за делеции

- Линейные $\Delta = n_d \cdot d$
- Аффинные $\Delta = d_0 + n_d \cdot d$

  Выравнивание имеет блочную структуру

Поиск сходства

Быстрый поиск сходства BLAST

- Забудем (пока) про делеции.
- Составим индексную таблицу, куда положим все возможные слова данной длины

“банк последовательностей”

```
accgattgctgacttgtacgtgtttaacgtgcattaaaaacgcgc
12345678901234567890123456789012345678901234 5
```

Запрос

```
aacgc
```

индексная таблица

aa	26,36,37,38,39
ac	1,12,27,40
ag	-
at	5,33
ca	32
cc	2,18
cg	3,19,28,41,43
ct	9,13,
...	

1. Разбиваем запрос на слова:

```
aacgc
aa 1
 ac 2
  cg 3
   gg 4
```

2. Находим в таблице слова, получаем пары :

```
aacgc
aa 1:26; 1:36;..
ac 2:1; 2:12;..
cg 3:3; 3:19;..
gc 4:31; 4:42;...
```

3. **Расширяем пары**

```
cg 3:19
aaCGc
taCGc
```

Поиск сходства

Статистическая значимость выравнивания

p-value — вероятность увидеть вес выравнивания такой же или больше при выравнивании ***случайных последовательностей***

e-value – ожидаемое количество локальных выравниваний с весом таким, или больше в банке ***случайных последовательностей***

Сборка геномов

Секвенирование

- Разбиваем (физически) геном на фрагменты
- Читаем фрагменты

Задача:

- по фрагментам собрать полную последовательность генома

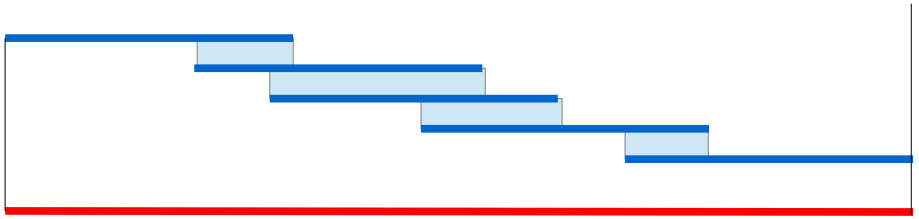
Особенности:

- Платформа *Illumina* Размер фрагментов = 100 нуклеотидов
- Есть ошибки, но мало
- Число фрагментов 5 млн.
- Полиморфизм (гетерозиготность)

- ***или*** (платформа PacBio) размер фрагментов 10 тыс. нуклеотидов. Есть ошибки и много

Сборка геномов

Сборка по перекрытиям



Фрагменты

Контиг

Задача 1.

Найти перекрытия

Задача 2.

По перекрытиям построить контиги

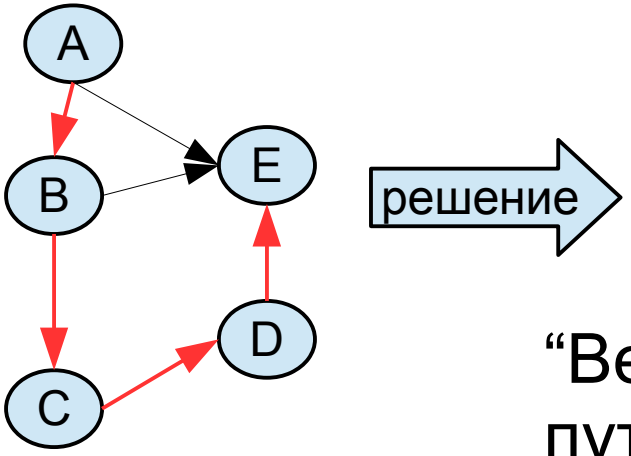
Сборка геномов

Допустим, перекрытия мы нашли

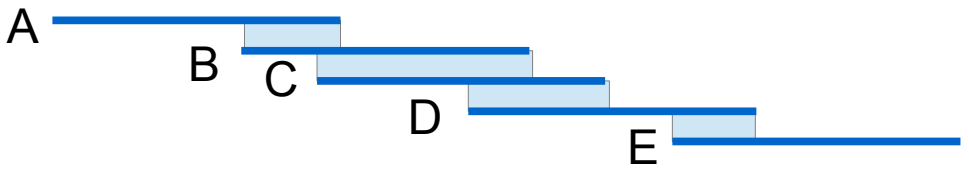
Граф:

вершины = фрагметны; ребра = перекрытия

Объяснить данные — пройти по всем вершинам графа



решение



Возможны случайные перекрытия фрагментов (AE, BE)

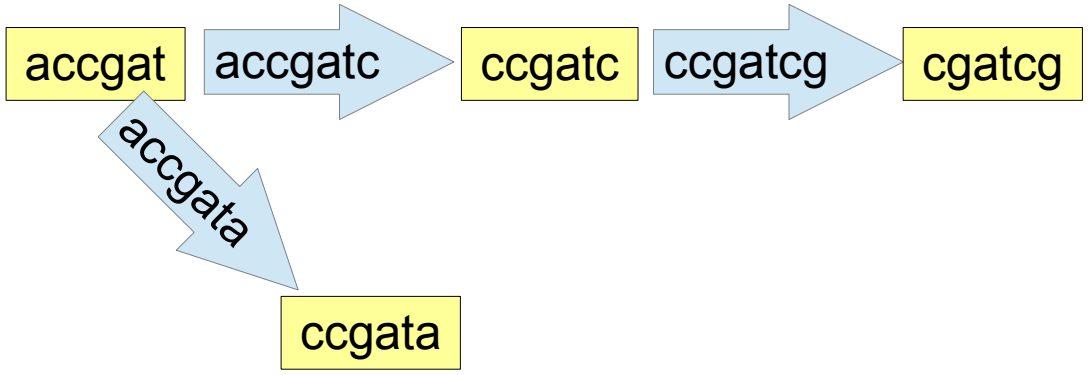
“Вершинная задача (гамильтонов путь)” - нет эффективных алгоритмов

Сборка геномов

1. Разобьем все фрагменты на слова заданной длины и сваливаем все в одну кучу. Повторяющиеся слова учитываем один раз. Собираем контиги из этих коротких слов, а не из фрагментов.

2. Переопределим граф (граф де-Бреина):

ребра = слова; вершины префиксы и суффиксы слов



Достижения:

- Задача стала “реберной” - есть эффективные алгоритмы
- Перекрытия сами собой определились — слово, принадлежащее сразу двум фрагментам — есть перекрытие

Потери:

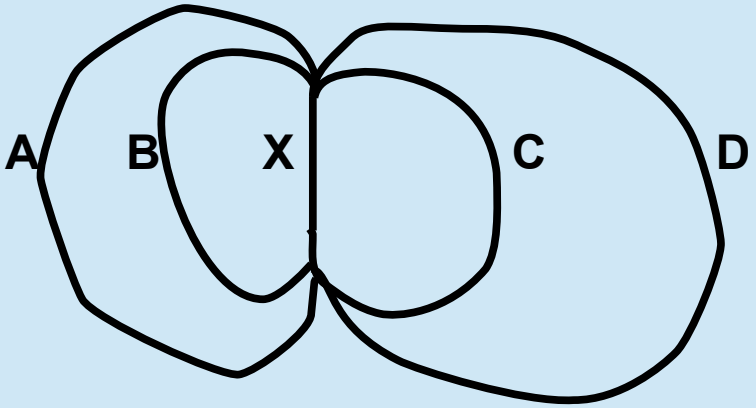
- Вместо фрагментов длиной 100 работаем с фрагментами длиной слово

Сборка геномов

Алгоритм:

- Разбиваем фрагменты на слова заданной длины (например,15)
- На этих словах строим граф де-Бреина
- На полученном графе ищем пути без развилок — это мини-контиги
- Вспоминаем, что слова пришли из фрагментов. Если два мини-контига имеют слова из одного фрагмента, то объединяем в контиг

Проблема для любого алгоритма сборки генома — повторы, которые длиннее фрагмента



Возможные прочтения:

A-X-B-X-C-X-D

A-X-C-X-D-X-B

A-X-D-X-B-X-C

.....

Сборка геномов