



# выравнивания

## vol. 2

машаеж копираитед

## что такое score?

```
EMBOSS_001      1 ACGACGGAGC      10
                  |||||
EMBOSS_001      1 ACGACGGAGC      10
```

```
Length: 10
Identity:  10/10 (100.0%)
Similarity: 10/10 (100.0%)
Gaps:      0/10 ( 0.0%)
Score: 63.0
```

```
EMBOSS_001      1 ACGACGGAGC-----      10
EMBOSS_001      1 -----TTTTTTTTTT      10
```

```
Length: 20
Identity:   0/20 ( 0.0%)
Similarity: 0/20 ( 0.0%)
Gaps:      20/20 (100.0%)
Score: 0.0
```

по итогу получаем заполненную матрицу:

		A	T	C	C	G	A	G	T	T
	0	→ -10	→ -20	→ -30	→ -40	→ -50	→ -60	→ -70	→ -80	→ -90
A	↓ -10	↘ 2	→ -8	→ -18	→ -28	→ -38	↘ -48	→ -58	→ -68	→ -78
T	↓ -20	↘ -12	↘ 4	→ -6	→ -16	→ -26	→ -36	→ -46	↘ -56	↘ -66
C	↓ -30	↘ -22	↓ -6	↘ 6	↘ -4	→ -14	→ -24	→ -34	→ -44	→ -54
A	↓ -40	↘ -28	↓ -16	↓ -4	↘ 4	↘ -5	↘ -12	→ -22	→ -32	→ -42
G	↓ -50	↓ -38	↓ -26	↓ -14	↘ -6	↘ 6	→ -4	↘ -10	→ -20	→ -30
T	↓ -60	↓ -48	↘ -36	↓ -24	↘ -15	↓ -4	↘ 4	↘ -6	↘ -8	↘ -18
C	↓ -70	↓ -58	↓ -46	↘ -34	↘ -22	↓ -14	↘ -6	↘ 2	↘ -7	↘ -9

как из заполненной матрицы получить оптимальное выравнивание?

# traceback

идём назад по стрелкам, попутно выписывая получившееся выравнивание

		A	T	C	C	G	A	G	T	T
	0	→ -10	→ -20	→ -30	→ -40	→ -50	→ -60	→ -70	→ -80	→ -90
A	↓ -10	↘ 2	→ -8	→ -18	→ -28	→ -38	↘ -48	→ -58	→ -68	→ -78
T	↓ -20	↘ -12	↘ 4	→ -6	→ -16	→ -26	→ -36	→ -46	↘ -56	↘ -66
C	↓ -30	↘ -22	↓ -6	↘ 6	↘ -4	→ -14	→ -24	→ -34	→ -44	→ -54
A	↓ -40	↘ -28	↓ -16	↓ -4	↘ 4	↘ -5	↘ -12	→ -22	→ -32	→ -42
G	↓ -50	↓ -38	↓ -26	↓ -14	↘ -6	↘ 6	→ -4	↘ -10	→ -20	→ -30
T	↓ -60	↓ -48	↘ -36	↓ -24	↘ -15	↓ -4	↘ 4	↘ -6	↘ -8	↘ -18
C	↓ -70	↓ -58	↓ -46	↘ -34	↘ -22	↓ -14	↘ -6	↘ 2	↘ -7	↘ -9

наше выравнивание:

ATCCGACTT  
AT - C- AGTC

↑  
score выравнивания,  
грубо говоря, то, насколько  
хорошим оно получилось

самостоятельная работа

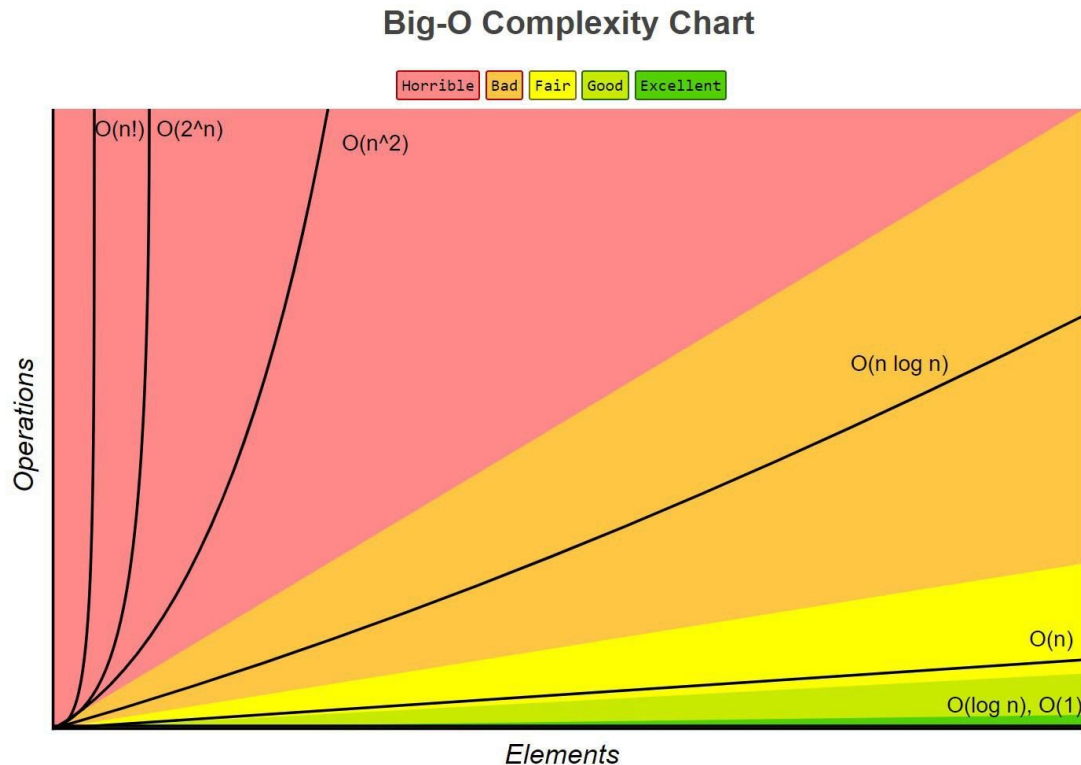
GTCG  
ATGA

матрица замен

	C	T	A	G
C	2	-1	-2	-2
T	-1	2	-2	-2
A	-2	-2	2	-1
G	-2	-2	-1	2

$$S(i, j) = \max \begin{cases} S(i, j-1) - d \\ S(i-1, j-1) + s(x_i, y_j) \\ S(i-1, j) - d \end{cases}$$

сложность алгоритма Нидлмана-Вунша -  $O(m \cdot n)$  - и по скорости, и по памяти






Big-O или какой максимум операций нашего алгоритма

проход по списку	$O(n)$
вложенный цикл	$O(n^2)$
бинарный поиск	$O(\log n)$

# способы оптимизации: по памяти

часто приходится сравнивать большие последовательности - для них не ок хранить в памяти всю таблицу  $m \times n$ . тем более, для подсчета качества выравнивания (score) она нам вся и не нужна. сколько строчек нужно для того, чтобы произвести все наши счетные операции?

$S(i-1, j-1)$ 	$S(i, j-1)$ 
$S(i-1, j)$ 	$S(i, j)$

ответ: 

их мы и будем хранить в двух списках!

## способы оптимизации: по вычислительной сложности

мы не можем потерять вычисления без потери точности – тогда какие-то пути в таблице не будут пройдены. но точность можно потерять аккуратно:

допущение: если наше выравнивание очень плохое, нам не нужно знать, насколько именно – важен сам факт. тогда мы можем ограничить число гэпов так, чтобы такие возможные выравнивания не учитывались:

- - - - -  
ATCAGTC

ATCCGAGTT  
- - - - -

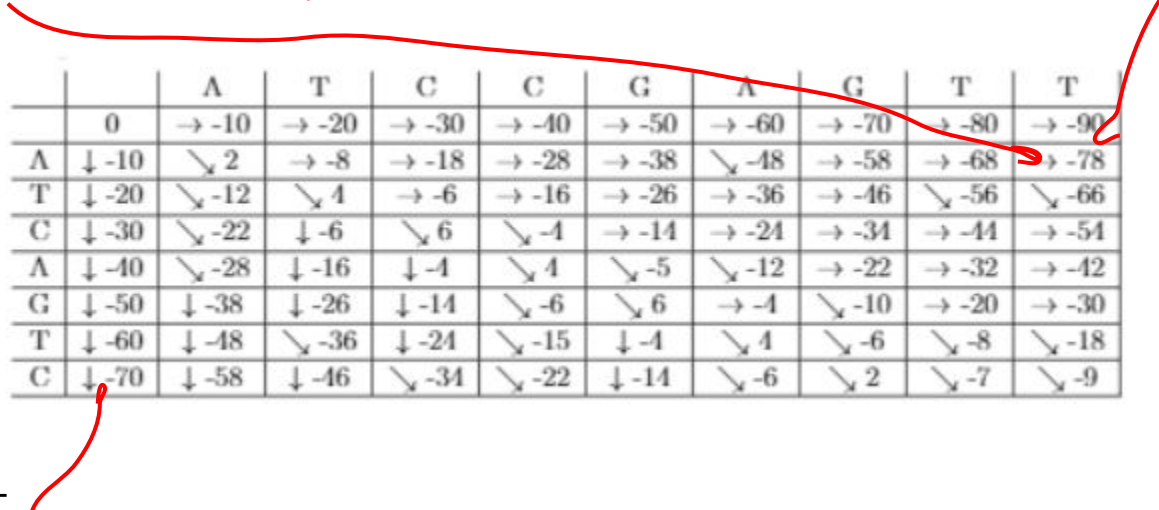
ATCCGAGTT  
- - - - A - - -



заметим, что плохие  
выравнивания встречаются в  
верхнем правом и нижнем  
левом углах таблицы:

ATCCGAGTT  
- - - - - A - - -

ATCCGAGTT  
- - - - - - - - -



		A	T	C	C	G	A	G	T	T
	0	→ -10	→ -20	→ -30	→ -40	→ -50	→ -60	→ -70	→ -80	→ -90
A	↓ -10	↘ 2	→ -8	→ -18	→ -28	→ -38	↘ -48	→ -58	→ -68	→ -78
T	↓ -20	↘ -12	↘ 4	→ -6	→ -16	→ -26	→ -36	→ -46	↘ -56	↘ -66
C	↓ -30	↘ -22	↓ -6	↘ 6	↘ -4	→ -14	→ -24	→ -34	→ -44	→ -54
A	↓ -40	↘ -28	↓ -16	↓ -4	↘ 4	↘ -5	↘ -12	→ -22	→ -32	→ -42
G	↓ -50	↓ -38	↓ -26	↓ -14	↘ -6	↘ 6	→ -4	↘ -10	→ -20	→ -30
T	↓ -60	↓ -48	↘ -36	↓ -24	↘ -15	↓ -4	↘ 4	↘ -6	↘ -8	↘ -18
C	↓ -70	↓ -58	↓ -46	↘ -34	↘ -22	↓ -14	↘ -6	↘ 2	↘ -7	↘ -9

- - - - -  
ATCAGTC

-> можно их просто не учитывать!



все нежеланные значения теперь у нас  
находятся в серых треугольниках – в них  
мы напишем значения  $= -\infty$

алгоритм будет идти в пределах белой  
полосы шириной  $2d + 1$ , где  $d$  – количество  
допустимых гэпов

он не будет  
отфильтровывать все пути с  
количеством гэпов  $> d$ , но  
он фильтрует большую  
часть и оставляет все пути,  
где гэпов  $< d$

		A	T	C	C	G
	0	-10	-20	-30	-40	-50
A	-10					
T	-20					
C	-30					
A	-40					
G	-50					

# какие бывают выравнивания?

1. **глобальные** — выравниваем два слова друг напротив друга целиком

SIMILARITY  
PI-LLAR---

2. **множественные** — глобальные выравнивания нескольких слов

SIMILARITY  
PI-LLAR---  
----LARGE-

3. **локальные** — ищем подпоследовательности внутри слов с наибольшим качеством выравнивания (score)

SIMILARITY  
PI-LLAR---

# алгоритм Смита-Вотермана

или локальное выравнивание

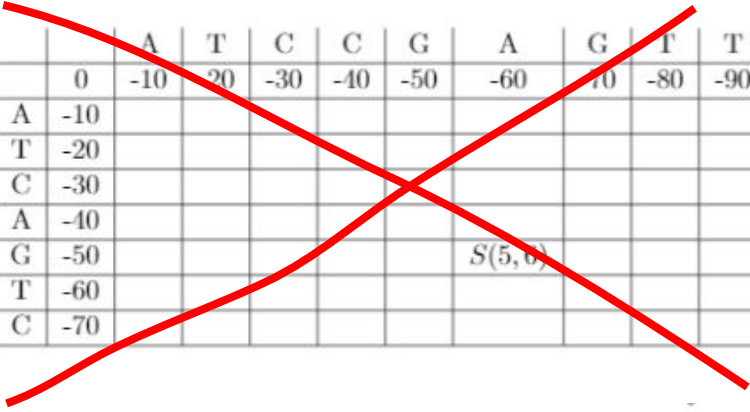
## SIMILARITY

PI-LAR---

## где нам нужно начинать и заканчивать?

матрица замен и  
штраф за гэпы  
сохраняются

[illegible]



		A	T	C	C	G	A	G	T	T
	0	-10	-20	-30	-40	-50	-60	-70	-80	-90
A	-10									
T	-20									
C	-30									
A	-40									
G	-50									
T	-60									
C	-70									

мы можем выйти из любой точки и  
прийти в любую точку, поэтому нету  
смысла штрафовать за гэпы в начале!

		A	T	C	C	G	A	G	T	T
	0	0	0	0	0	0	0	0	0	0
A	0									
T	0									
C	0									
A	0									
G	0									
T	0									
C	0									

из-за этого же глупо  
писать в таблицу  
значения меньше нуля -  
можно же просто начать  
с начала! (то есть с нуля)

=> в нашу формулу добавляется  
еще одно слагаемое

$$S(i, j) = \max \begin{cases} 0 \\ S(i, j - 1) - d \\ S(i - 1, j - 1) + s(x_i, y_j) \\ S(i - 1, j) - d \end{cases}$$

итоговая таблица:

		A	T	C	C	G	A	G	T	T
	0	0	0	0	0	0	0	0	0	0
A	0	↘ 2	0	0	0	0	↘ 2	0	0	0
T	0	0	↘ 4	0	0	0	0	0	↘ 2	↘ 2
C	0	0	0	↘ 6	↘ 2	0	0	0	0	↘ 1
A	0	↘ 2	0	0	↘ 4	↘ 1	↘ 2	0	0	0
G	0	0	0	0	0	↘ 6	0	↘ 4	0	0
T	0	0	↘ 2	0	0	0	↘ 4	0	↘ 6	0
C	0	0	0	↘ 4	0	0	0	↘ 2	0	↘ 5

где же в ней score?



находим в таблице максимум и по  
стрелкам идем до нуля

		A	T	C	C	G	A	G	T	T
	0	0	0	0	0	0	0	0	0	0
A	0	↘ 2	0	0	0	0	↘ 2	0	0	0
T	0	0	↘ 4	0	0	0	0	0	↘ 2	↘ 2
C	0	0	0	↘ 6	↘ 2	0	0	0	0	↘ 1
A	0	↘ 2	0	0	↘ 4	↘ 1	↘ 2	0	0	0
G	0	0	0	0	0	↘ 6	0	↘ 4	0	0
T	0	0	↘ 2	0	0	0	↘ 4	0	↘ 6	0
C	0	0	0	↘ 4	0	0	0	↘ 2	0	↘ 5

AGT

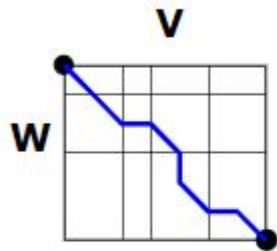
AGT

оптимальных решений несколько:

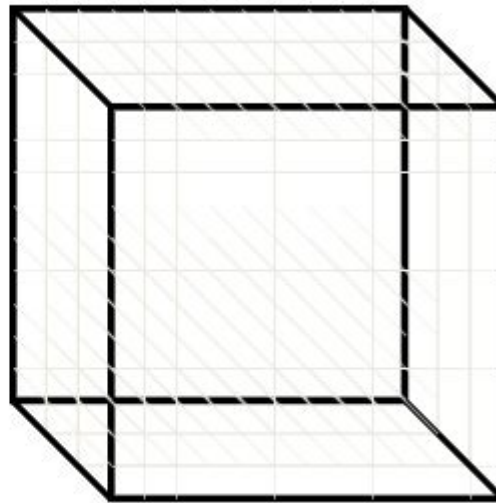
ATCAG

ATCCG

# множественные выравнивания



2D table



3D graph

Время выполнения алгоритма с определённой сложностью в зависимости от размера входных данных при скорости  $10^6$  операций в секунду

размер сложность	10	20	30	40	50	60
$n$	0,00001 сек.	0,00002 сек.	0,00003 сек.	0,00004 сек.	0,00005 сек.	0,00005 сек.
$n^2$	0,0001 сек.	0,0004 сек.	0,0009 сек.	0,0016 сек.	0,0025 сек.	0,0036 сек.
$n^3$	0,001 сек.	0,008 сек.	0,027 сек.	0,064 сек.	0,125 сек.	0,216 сек.
$n^5$	0,1 сек.	3,2 сек.	24,3 сек.	1,7 минут	5,2 минут	13 минут

чтобы хорошо написать ср дальше читать не обязательно



Q5E940	BOVIN	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	HUMAN	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	MOUSE	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	RAT	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	CHICK	-----MPREDRATWKSNYFMKIIQLDDYPKCFVVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	RANSY	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
Q7ZUG3	BRARE	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	ICTPU	-----MPREDRATWKSNYFLKIIQLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	DROME	-----MVRENKAAWKAQYFIKVFELDFPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLGMGKNTMMRKAIRGHLENN--PALE	76
RLA0	DICDI	-----MSGAG-SKRKKLFIEKATKLTFTYDKMIVAEADFVGSQLOKIRKSIRGI-GAVLMGKKTMRKVIRDLADSK--PELD	75
Q54LP0	DICDI	-----MSGAG-SKRKNVFIEKATKLTFTYDKMIVAEADFVGSQLOKIRKSIRGI-GAVLMGKKTMRKVIRDLADSK--PELD	75
RLA0	PLAF8	-----MAKLSKQKKQMYIEKLSSLIQQYSKILIVHVDNVGSKOMASVRKSIRGK-ATILMGKNTIRIATLAKKNLQAV--PQIE	76
RLA0	SULAC	-----MTGLAVTTTKIAKWKVDEVAELTEKLKTHKTIIANIEGFPADKLHEIRKKLRGK-ADIKVTKNLNFIALKNAG----YDK	79
RLA0	SULTO	-----MRIMAVITQERKIAKWKIEEVKELEKLRKYHTIIIANIEGFPADKLHDIRKKMRGM-AEIKVTKNLTFGIAAKNAG----LDVS	80
RLA0	SULSO	-----MKRLALALKQRKVASWKEEVKELTELIKNSNTILIGNLEGFPADKLHEIRKKLRGK-ATIKVTKNLTFKIAAKNAG----IDIE	80
RLA0	AERPE	MSVVSIVGQMYKREKPIPEWKTLMLEELFSEKIRRVVFLADLTGTPTFVVRVRKKLWKK-YPMMVAKKRIILRAMKAAGLE---LDDN	86
RLA0	PYRAE	MMLAIGKRRYVRTQYPARKVKIVSEATELQKYPVFLDLHGLSRILHEYYRRLRY-GVIKIAPKTLFKIAFTKVYGG---IPAE	85
RLA0	METAC	-----MAEERHHTEHIPQWKDEIENIKELIQSHKVFQMGVIEGILATKIQKIRRDLDKV-AVLKVSRTNLTIERALNQLG----ETIP	78
RLA0	METMA	-----MAEERHHTEHIPQWKDEIENIKELIQSHKVFQMGVIEGILATKIQKIRRDLDKV-AVLKVSRTNLTIERALNQLG----ESIP	78
RLA0	ARCFU	-----MAAVRGS---PPEYKVRAVEEIKRMISSKPVVAIVSFRNVAGOMQKIRREFRGK-AEIKVVKNTLLERDALG----GDYL	75
RLA0	METKA	MAVKAKGQPPSGYEPKVAEWKRREVKELKELMDEYENVGLVDLEGIAPAPLOEIRAKLRERDTIIRMSRNTLMRIALEEKLDER--PELE	78
RLA0	METTH	-----MAHVAEWKKKEVEELAKLHDLIKGYEVVGLNADLAPARQLQKMRQTLDLS-ALIRMSKKTLLISLAEKAGREL--ENVY	84
RLA0	METTL	-----MITAESEHKIAPWKIEEVNKLKELLKNGQIVALVDMMEVPARLOEIRDKIR-GTMTLKMSRNTLIERAIEKVAEETGNPEFA	82
RLA0	METVA	-----MIDAKSEHKIAPWKIEEVNKLKELLKNSANVIALIDMMEVPARLOEIRDKIR-DQMTLKMSRNTLIERAIEKVAEETGNPEFA	82
RLA0	METJA	-----METKVKAHVAPWKIEEVKTLKGLIKSKPVVAIVDMMDVPAPLOEIRDKIR-DKVKLRMSRNTLIERALKEAAEELNNPKLA	81
RLA0	PYRAB	-----MAHVAEWKKKEVEELANLKSYPVIALVDVSSMPAYPLSQMRRILRENGCLLRVSRTNLTIELAIKKAQELGKPELE	77
RLA0	PYRHO	-----MAHVAEWKKKEVEELAKLKSYPVIALVDVSSMPAYPLSQMRRILRENGCLLRVSRTNLTIELAIKKAQELGKPELE	77
RLA0	PYRFU	-----MAHVAEWKKKEVEELANLKSYPVIALVDVSSMPAYPLSQMRRILRENGCLLRVSRTNLTIELAIKKAQELGKPELE	77
RLA0	PYRKO	-----MAHVAEWKKKEVEELANLKSYPVIALVDVAGVPAYPLSKMRDKLR-GKALLRVSRTNLTIELAIKKAQELGQPELE	76
RLA0	HALMA	MSAESERKTETIPEWKQEEVDVAIVMIESYESVGVVNIAGIPSRLOQDMRRDLHGT-AELRVSRTNLTIERALDDVD---DGLT	79
RLA0	HALVO	MSSEVRQTEVIPWKREVEDELVDVIESYESVGVVVGAGIPSRLOQDMRRDLHGS-AAVRMSRNTLVNRALDEVN---DGFE	79
RLA0	HALSA	MSAEEQRTTEEVPEWKQREVAELVDLEITYDSVGVVNVGTGIPSRLOQDMRRDLHGQ-AALRMSRNTLLVRALDEEAG----DGLD	79
RLA0	THEAC	-----MKEVSQKKELVNEITRIKASRSVAIVDTAGIRITROIQDIRGKNRGK-INLKVIKKTLLFKALENLGD---EKLS	72
RLA0	THEVO	-----MRKINPKKEIVSELAQDITKSKAVIVDIKGVITROMODIRAKNRDK-VKIKVVKKTLLFKALDSIND---EKLT	72
RLA0	PICTO	-----MTEPAQWKIDFVKNLNEINSRKVAIVSIKGLRNNEFQKIRNSIRDK-ARIKVSRRALLRLAIENTGK----NNIV	72
ruler 1.....10.....20.....30.....40.....50.....60.....70.....80.....90			

Нам дали последовательность  $A$ , у нас есть база данных из кучи генов, надо найти кусочки из базы данных, похожие на последовательность  $A$ .

**Давайте просто построим все возможные локальные выравнивания ?**



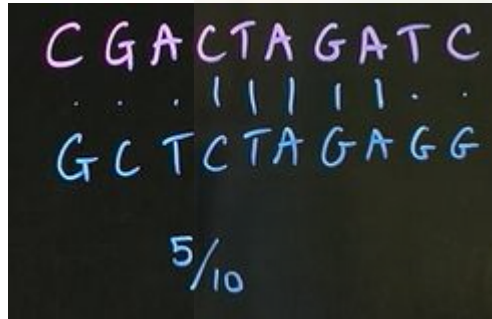
Нам дали последовательность  $A$ , у нас есть база данных из кучи генов, надо найти кусочки из базы данных, похожие на последовательность  $A$ .

**Давайте просто построим все возможные локальные выравнивания ?**

- + получим самые оптимальные выравнивания
- это долго, и чем больше база данных – тем дольше

Баланс между точностью и скоростью - BLAST

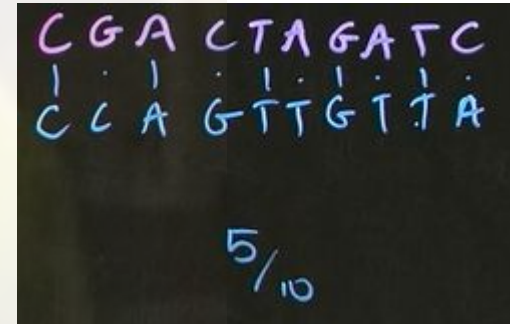
Хорошее выравнивание должно содержать хороший цельный кусок



CGACTAGATC  
.. .| | | |.  
GCTCTAGAGG  
5/10

This image shows a handwritten DNA sequence alignment on a black background. The top sequence is 'CGACTAGATC' in purple. The bottom sequence is 'GCTCTAGAGG' in blue. There are five vertical lines connecting the two sequences at positions 4, 5, 6, 7, and 8, indicating matches. The score '5/10' is written in blue at the bottom.

vs



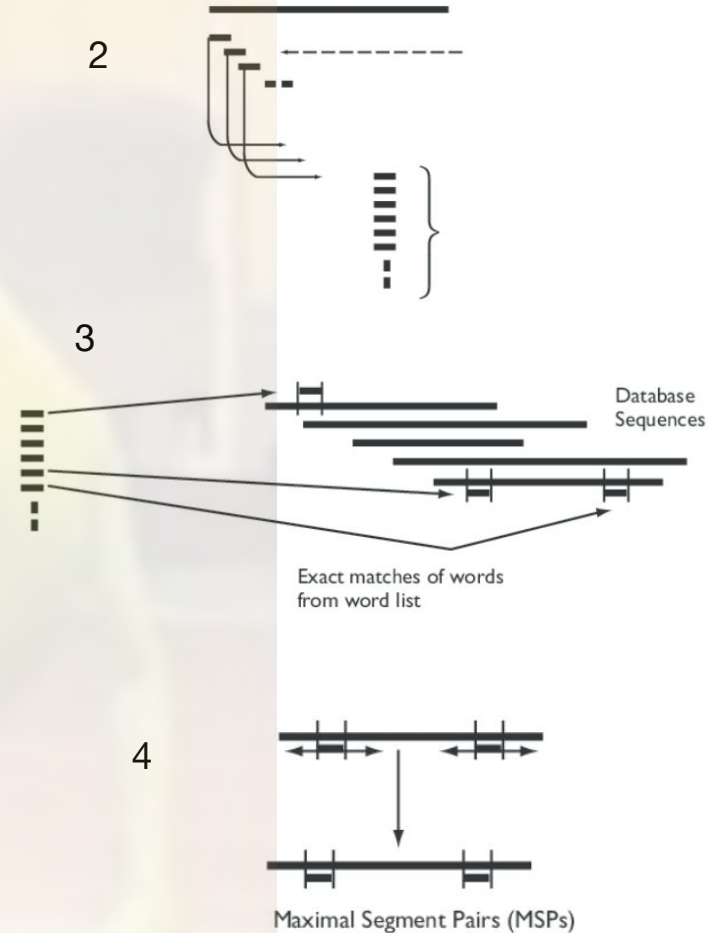
CGACTAGATC  
| . | . | . | .  
C C A G T T G T A  
5/10

This image shows a handwritten DNA sequence alignment on a black background. The top sequence is 'CGACTAGATC' in purple. The bottom sequence is 'C C A G T T G T A' in blue. There are vertical lines connecting the two sequences at positions 1, 3, 5, 6, 7, and 8, indicating matches. The score '5/10' is written in blue at the bottom. This alignment represents a better contiguous match than the one on the left.



## Суть бласта:

1. Уберем низкоинформативные участки последовательности
2. Разобьем последовательность на кусочки
3. Будем искать совпадения маленьких кусочков последовательности A и геномов из базы
4. Будем продливать эти совпадения методом Смита-Вотермана пока суммарный счет выравниваний будет больше заданного порога
5. Выберем какое-то количество лучших из получившихся выравниваний



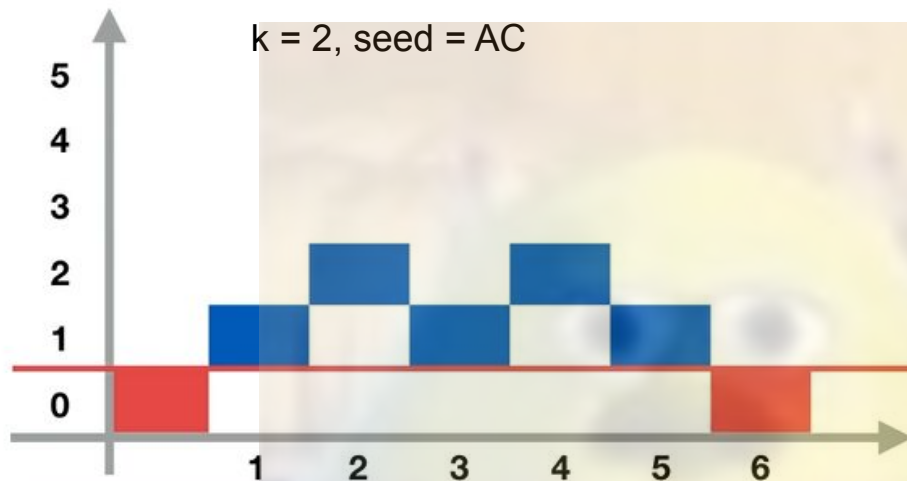
Составим из геномов базы данных все возможные куски длины  $K$ , это долго, но делаем только 1 раз

$AACTCACT \longrightarrow \{AAC: [0], ACT: [1, 5], CTC: [2], TCA: [3], CAC: [4]\}$

Составим из входной последовательности  $A$  все возможные куски

$ATCCGA \longrightarrow ATC, TCC, CCG, CGA$

Находим идеальные совпадения в базе данных

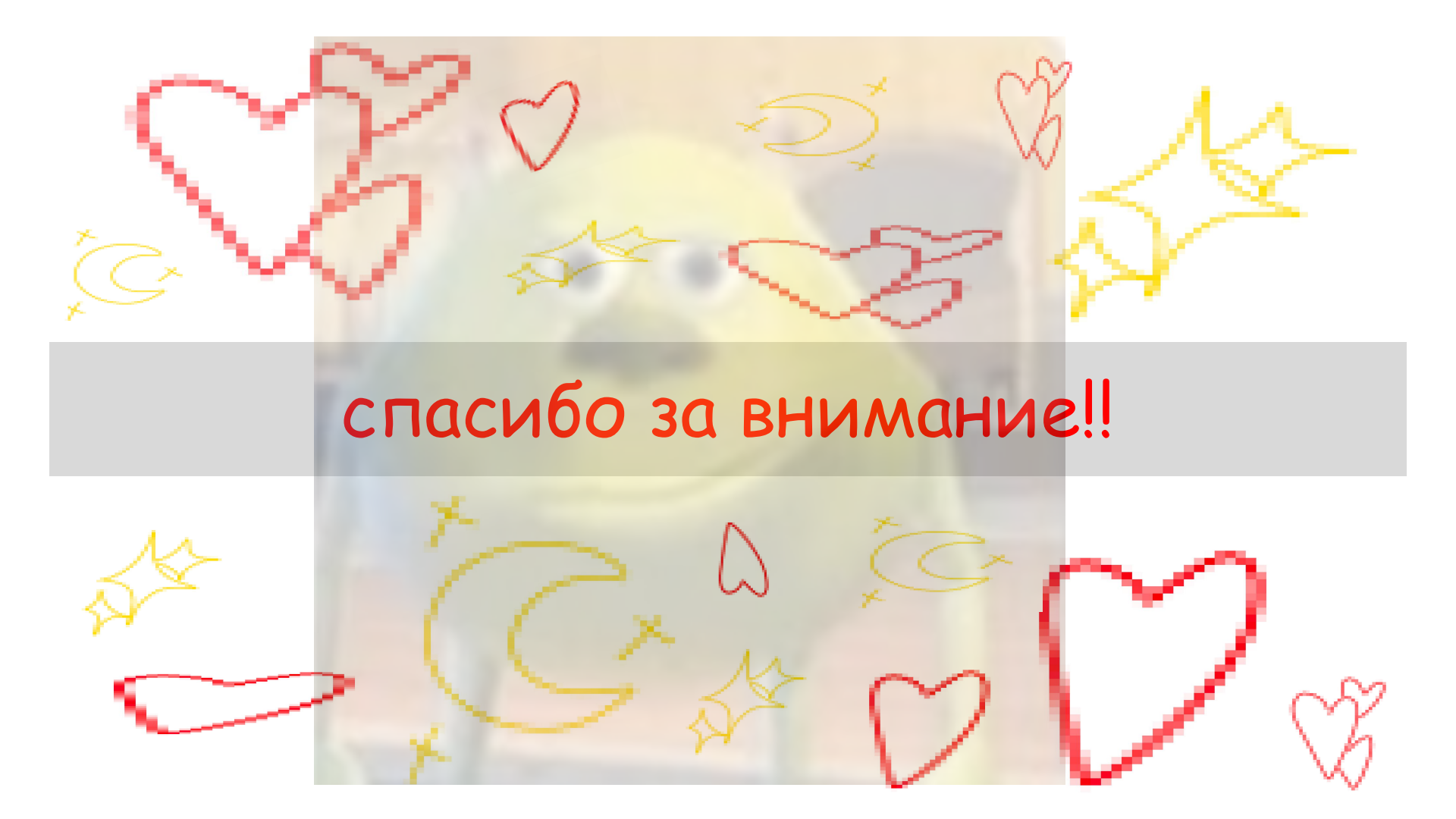


		A	C	G	G	A	T	T
	0	0	0	0	0	0	0	0
G	0	0	0	1	1	0	0	0
A	0	1	0	0	0	2	1	0
C	0	0	2	1	0	1	1	0
A	0	1	1	1	0	1	0	0
G	0	0	0	2	2	1	0	0
C	0	0	1	1	1	1	0	0

## Scoring Scheme

Match	1
Mismatch	-1
Gap Insertion	-1
<b>Score Threshold</b>	<b>1</b>

Получим локальное выравнивание:  
**ACGGA**



спасибо за внимание!!

## домашнее задание

нарисуйте таблицу для выравнивания последовательностей ATGAGTCTCT и CTGTCTCCTG, запишите score и оптимальное выравнивание

бонус: попробуйте решить ту же задачу с аффинными гэпами ( $d = 10$ ,  $e = 0,5$ )

*(бонус бонусом, но понимать принцип работы аффинного гэпа к следующему семинару хорошо бы всем)*