Межфакультетский курс «Биоинформатика»
Факультет биоинженерии и биоинформатики МГУ
весна 2017

Лекция 4

Молекулярная эволюция

Мутации

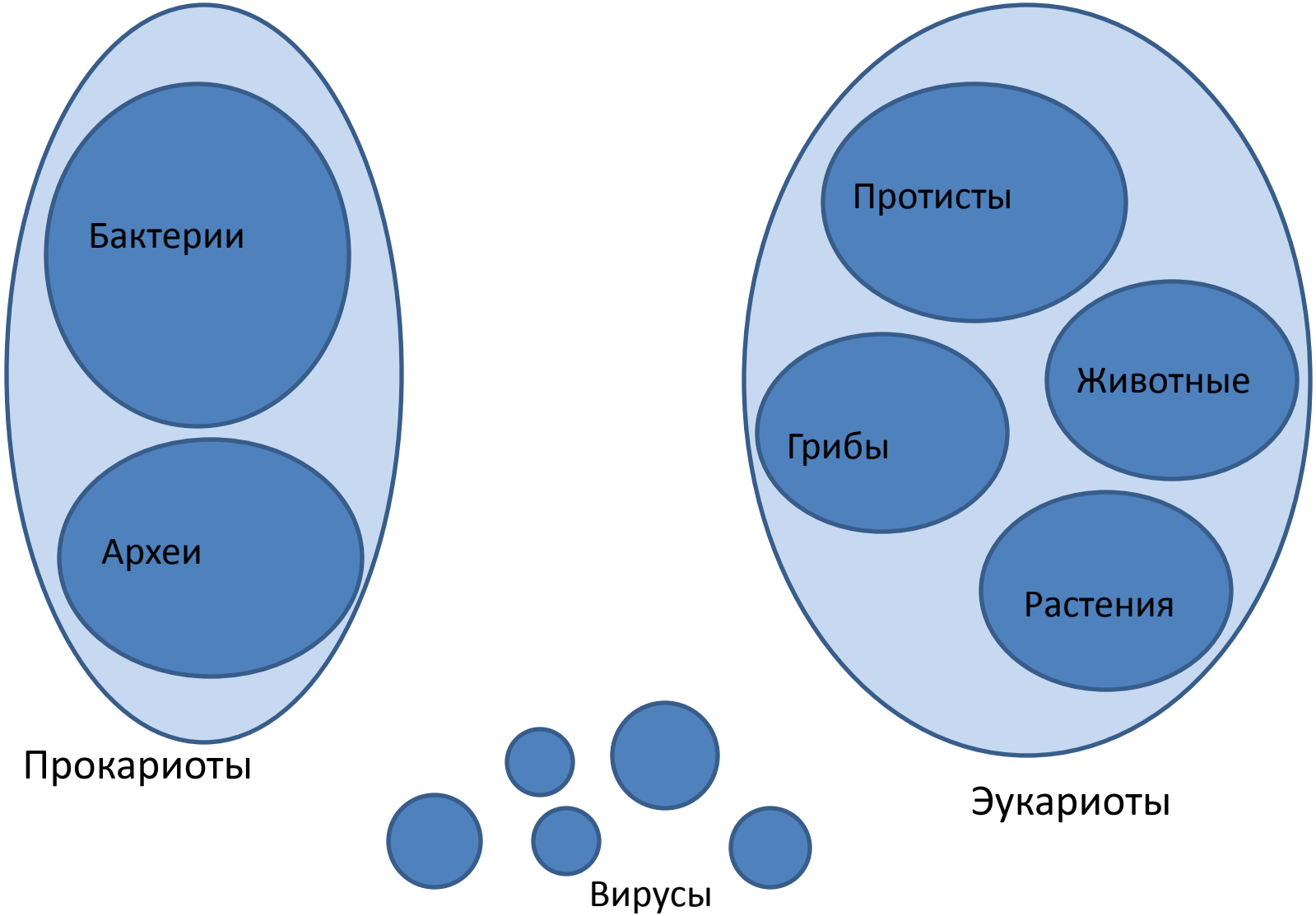
Дрейф и отбор

Выравнивание

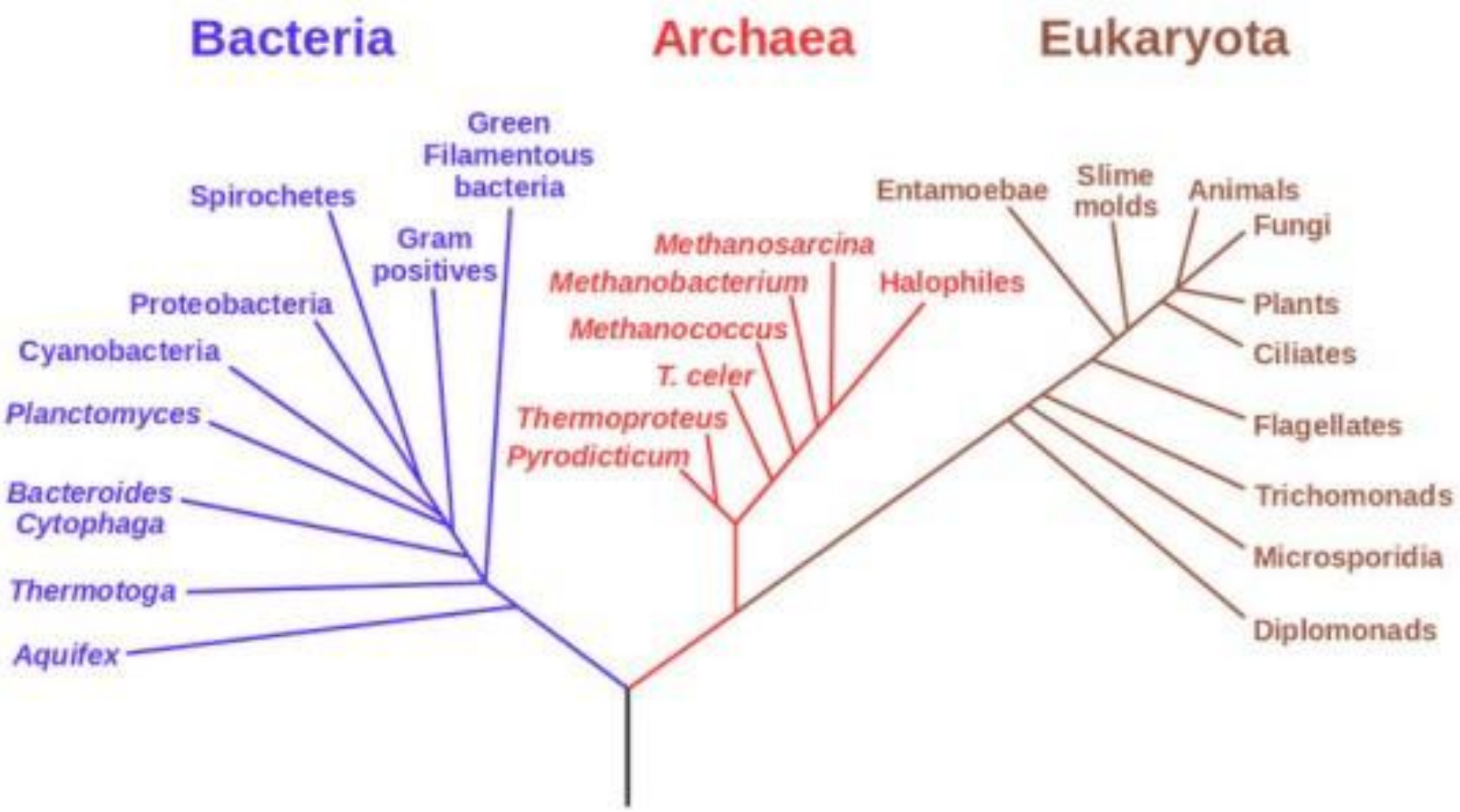
С.А. Спирин

15 марта 2017

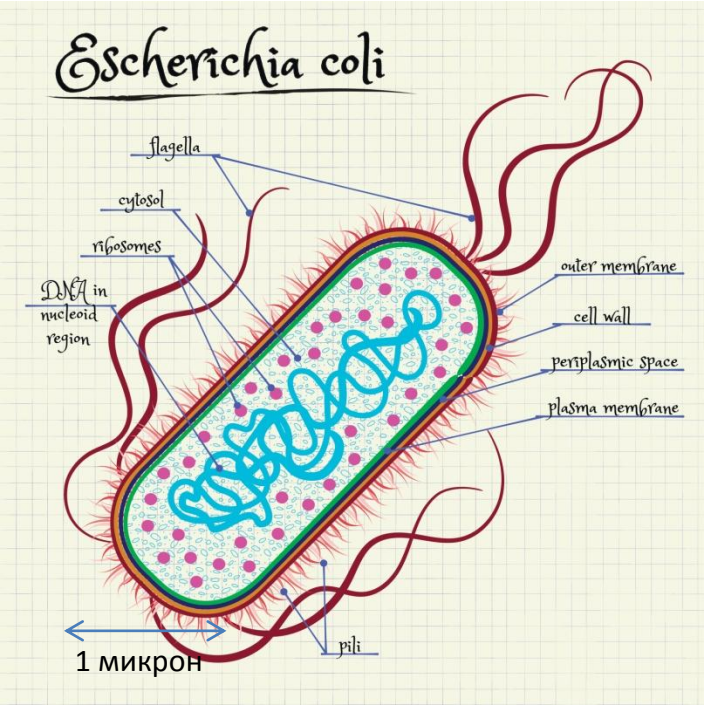
Живые существа



Phylogenetic Tree of Life

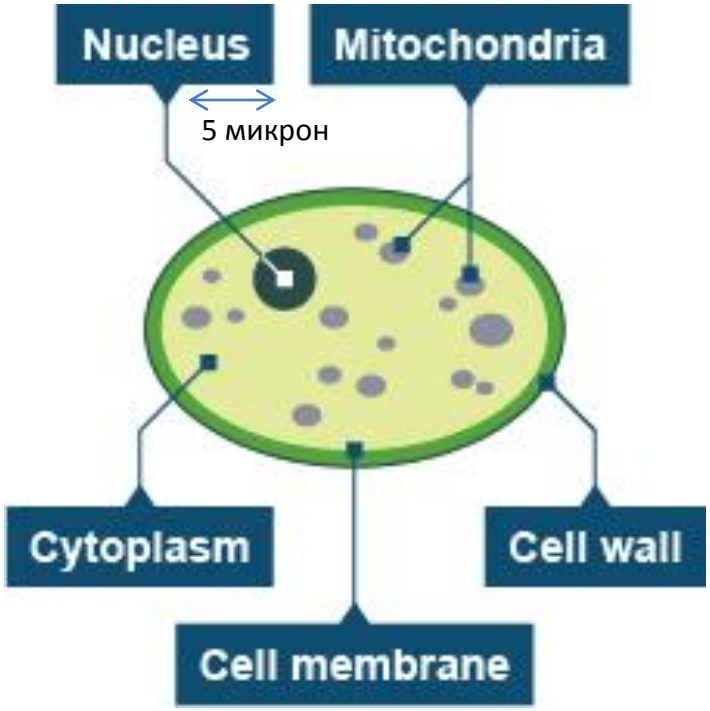


Клетки



Прокариотическая клетка

ДНК, как правило, кольцевая (одна, реже две-три хромосомы и несколько маленьких плазмид)



Эукариотическая клетка

ДНК в ядре – большие линейные хромосомы, в митохондриях – одна маленькая кольцевая

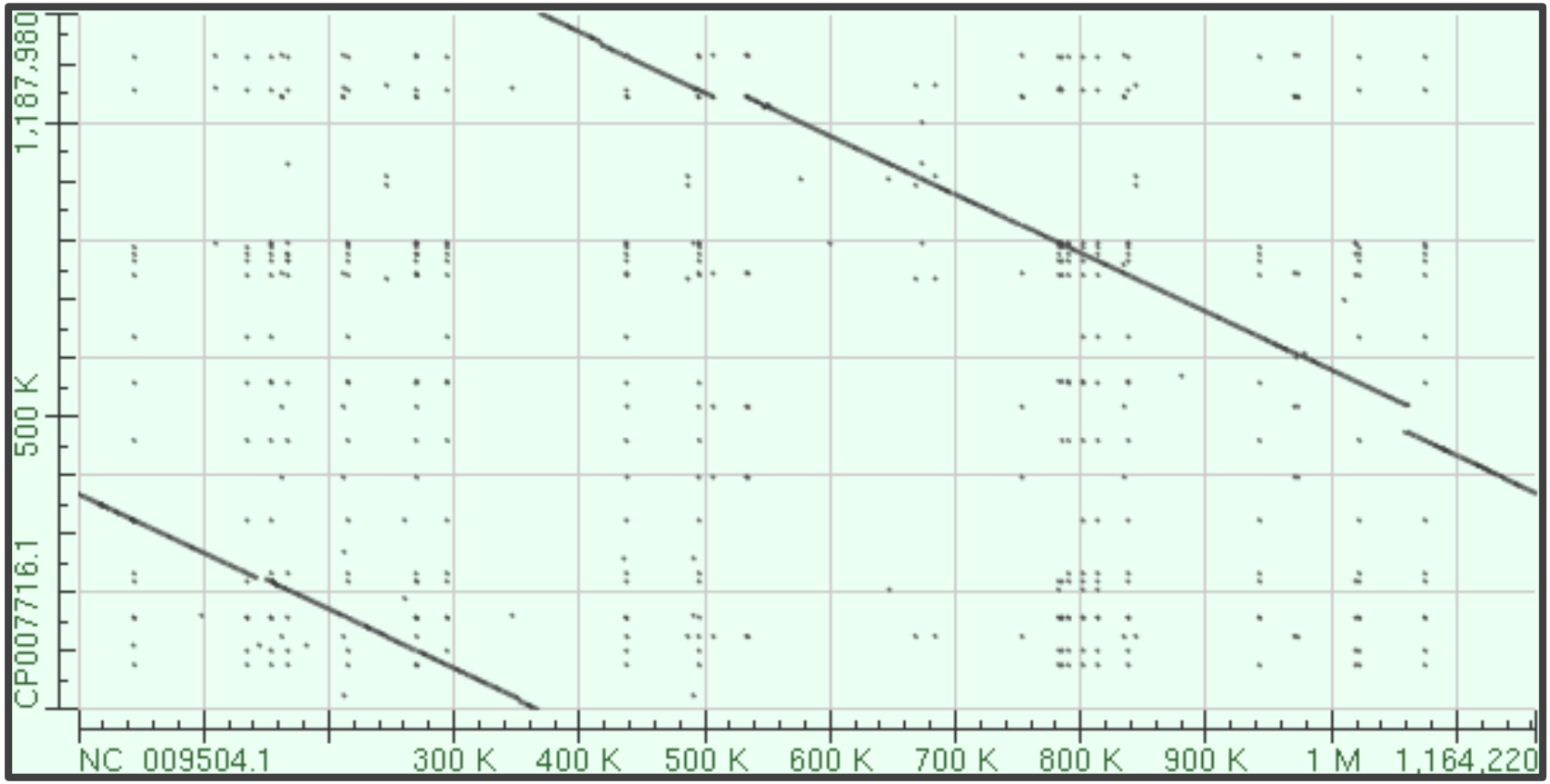
Последовательность ДНК

ctaaccctaaccctaaccctaaccctaaccctaaccctctgaaagtggacctatcagc
aggatgtgggtgggagcagattagagaataaaagcagactgcctgagccagcagtggc
aacccaatggggtccctttccatactgtggaagcttcgttctttcactctttgcaata
aatcttgctattgctcactctttgggtccacactgcctttatgagctgtgacactcac
cgcaaaggtctgcagcttcactcctgagccagtgagaccacaaccccaccagaaagaa
gaaactcagaacacatctgaacatcagaagaaacaaactccggacgcgccacctttaa
gaactgtaacactcaccgcgaggttccgcgtcttcattcttgaagtcagtgagaccaa
gaacccaccaattccagacacactaggaccctgagacaaccccctagaagagcacctgg
ttgataacccagttcccatctgggatttaggggacctggacagcccggaaaatgagct
cctcatctctaacccagttcccctgtggggatttaggggaccagggacagcccgttgc
atgagcccctggactctaacccagttcccttctggaatttaggggccctgggacagcc
ctgtacatgagctcctggtctgtaacacagttccccctgtggggatttagggacttggg
ccttctgtctttgggatctactctctatgggccacacagaccagttccccctgtgggga

В банках последовательностей лежит всегда последовательность одной из цепей, выбранной произвольно. Вторая цепь может быть восстановлена по комплементарности.

Если ДНК кольцевая, то начало тоже выбирается произвольно.

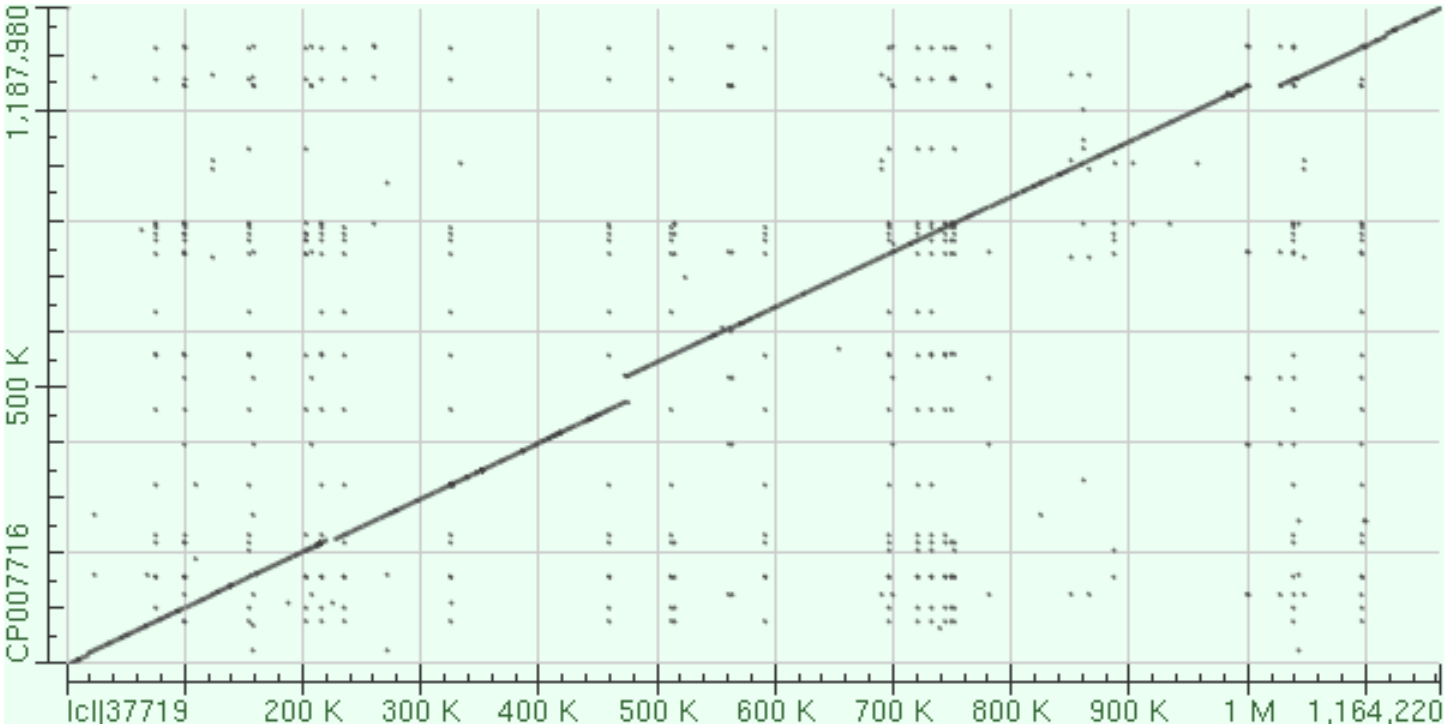
Карта сходства: два генома



Brucella suis

Brucella ovis

Карта сходства:
те же два генома



Brucella suis

Brucella ovis

Гены и белки

Геном
$3 \cdot 10^9$ букв у человека,
~ 10^6 букв у бактерий

содержит →

Кодирующие участки
<2% генома у человека,
~ 90% у бактерий

кодируют →

Белки
~ 25 000 у человека,
600 – 6000 у бактерий

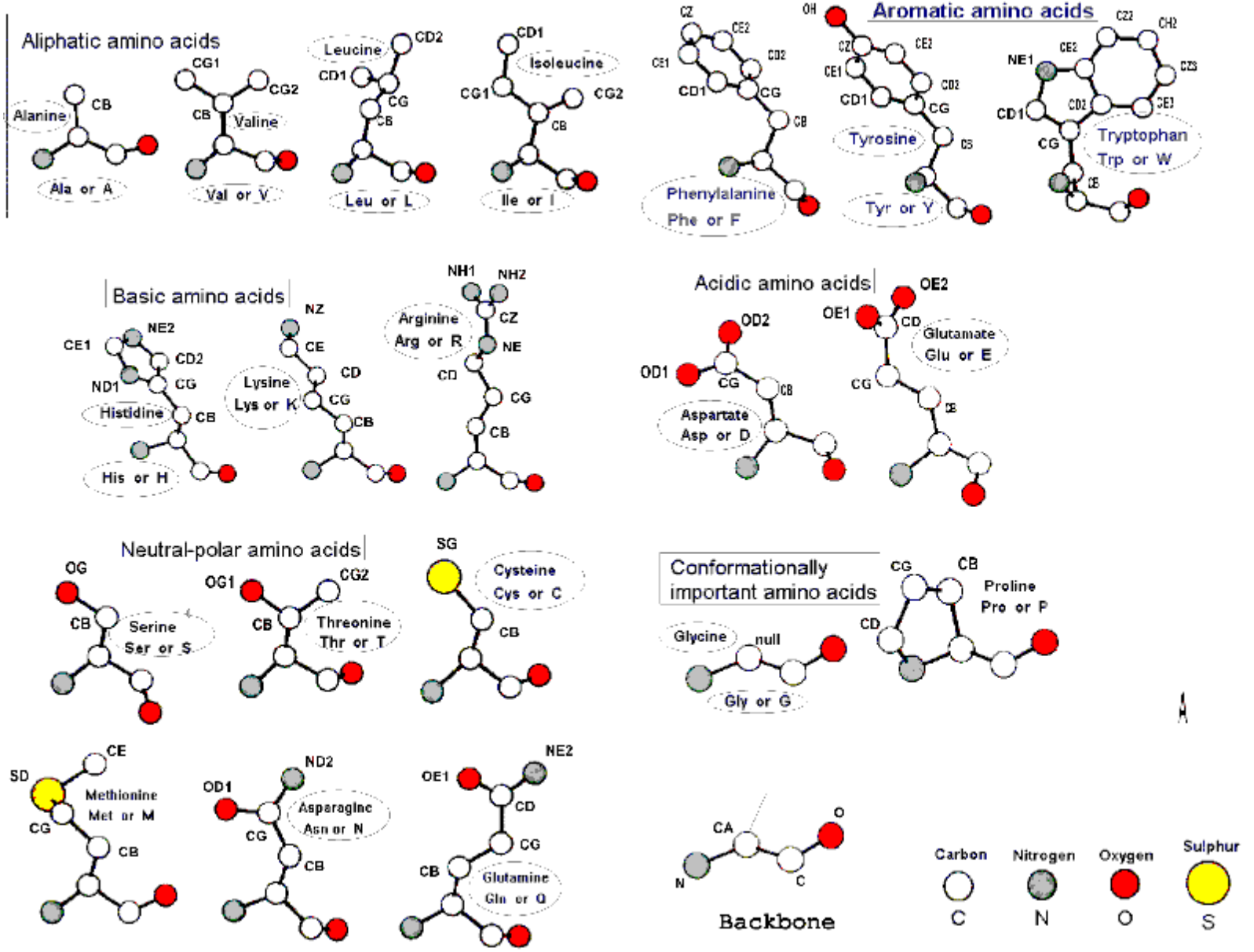
Генетический код

	T(U)	C	A	G
T(U)	TTT Phe	TCT Ser	TAT Tyr	TGT Cys
	TTC Phe	TCC Ser	TAC Tyr	TGC Cys
	TTA Leu	TCA Ser	TAA Stop	TGA Stop
	TTG Leu	TCG Ser	TAG Stop	TGG Trp
C	CTT Leu	CCT Pro	CAT His	CGT Arg
	CTC Leu	CCC Pro	CAC His	CGC Arg
	CTA Leu	CCA Pro	CAA Gln	CGA Arg
	CTG Leu	CCG Pro	CAG Gln	CGG Arg
A	ATT Ile	ACT Thr	AAT Asn	AGT Ser
	ATC Ile	ACC Thr	AAC Asn	AGC Ser
	ATA Ile	ACA Thr	AAA Lys	AGA Arg
	ATG Met	ACG Thr	AAG Lys	AGG Arg
G	GTT Val	GCT Ala	GAT Asp	GGT Gly
	GTC Val	GCC Ala	GAC Asp	GGC Gly
	GTA Val	GCA Ala	GAA Glu	GGA Gly
	GTG Val	GCG Ala	GAG Glu	GGG Gly

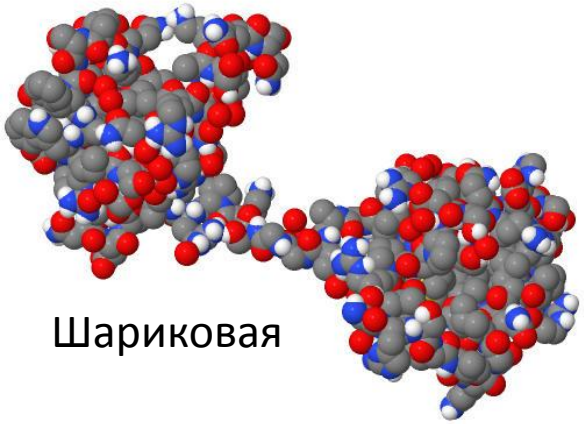
Аминокислоты

A Ala Alanine Аланин
R Arg Arginine Аргинин
N Asn Asparagine Аспарагин
D Asp Aspartic Acid Аспарагиновая кислота
C Cys Cysteine Цистеин
Q Gln Glutamine Глютамин
E Glu Glutamic Acid Глутаминовая кислота
G Gly Glycine Глицин
H His Histidine Гистидин
I Ile Isoleucine Изолейцин
L Leu Leucine Лейцин
K Lys Lysine Лизин
M Met Methionine Метионин
F Phe Phenylalanine Фенилаланин
P Pro Proline Пролин
S Ser Serine Серин
T Thr Threonine Треонин
W Trp Thryptophan Триптофан
Y Tyr Tyrosine Тирозин
V Val Valine Валин
"Stop" в таблице кода означает стоп-кодон – сигнал окончания трансляции.

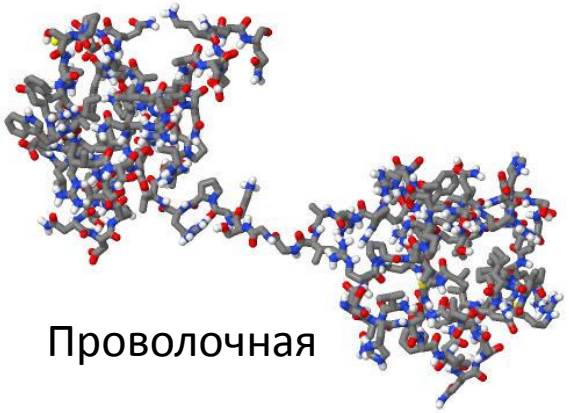
Аминокислотные остатки



Разные визуализации структуры белка

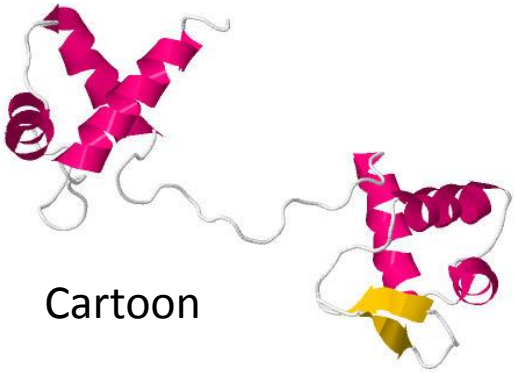


Шариковая



Проволочная

Остовная

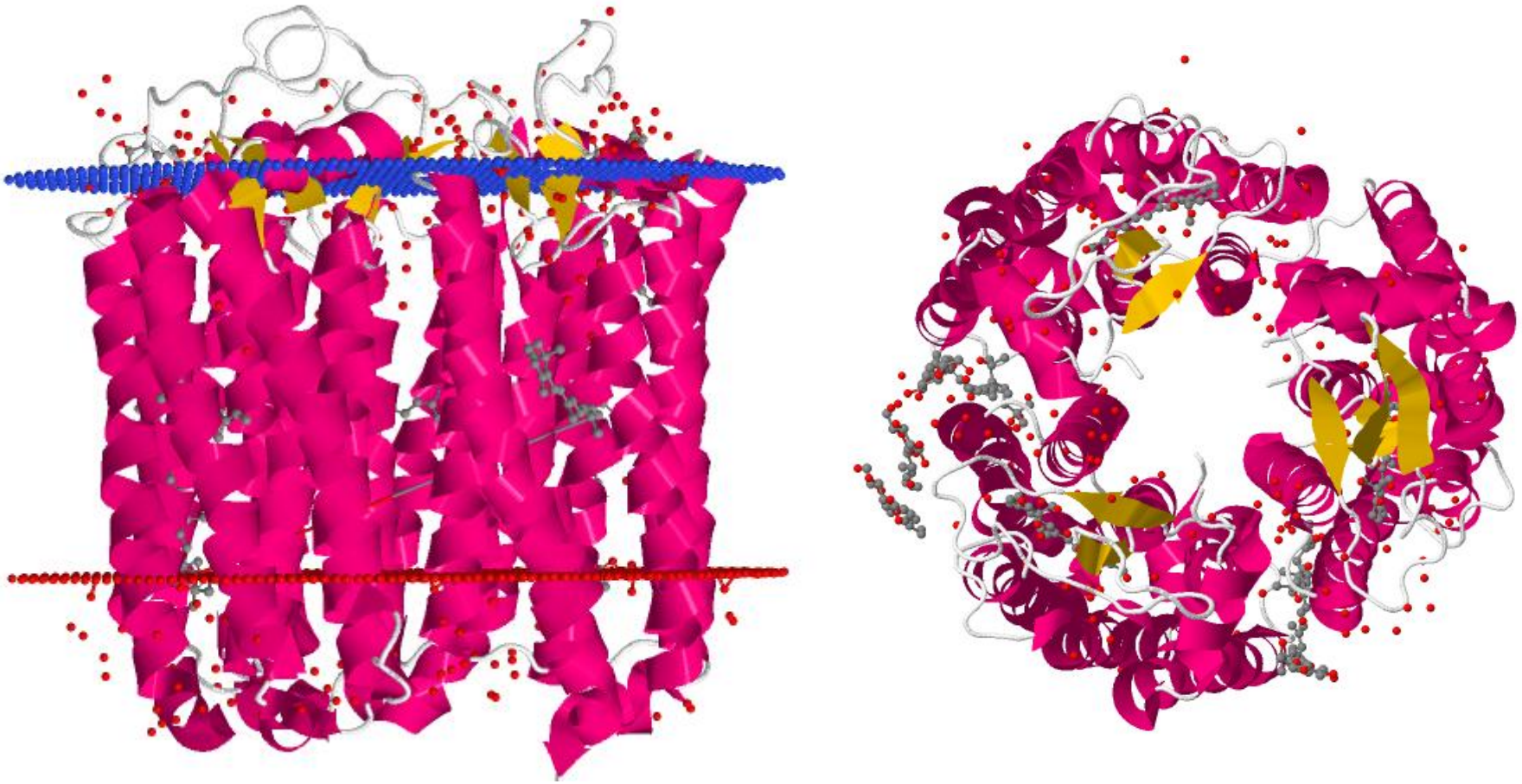


Cartoon

SHSGVNQLGGVFVNGRPLPDSTRQRIVELAHSGARPCDISRILQVSNGCVSKILGRYYAT
GSIRPRAIGGSKPRVATPEVVSKIAQYKQECPSIFAWEIRDRLLSEGVCTNDNIPSVSSI
NRVLRNLASEKQQ

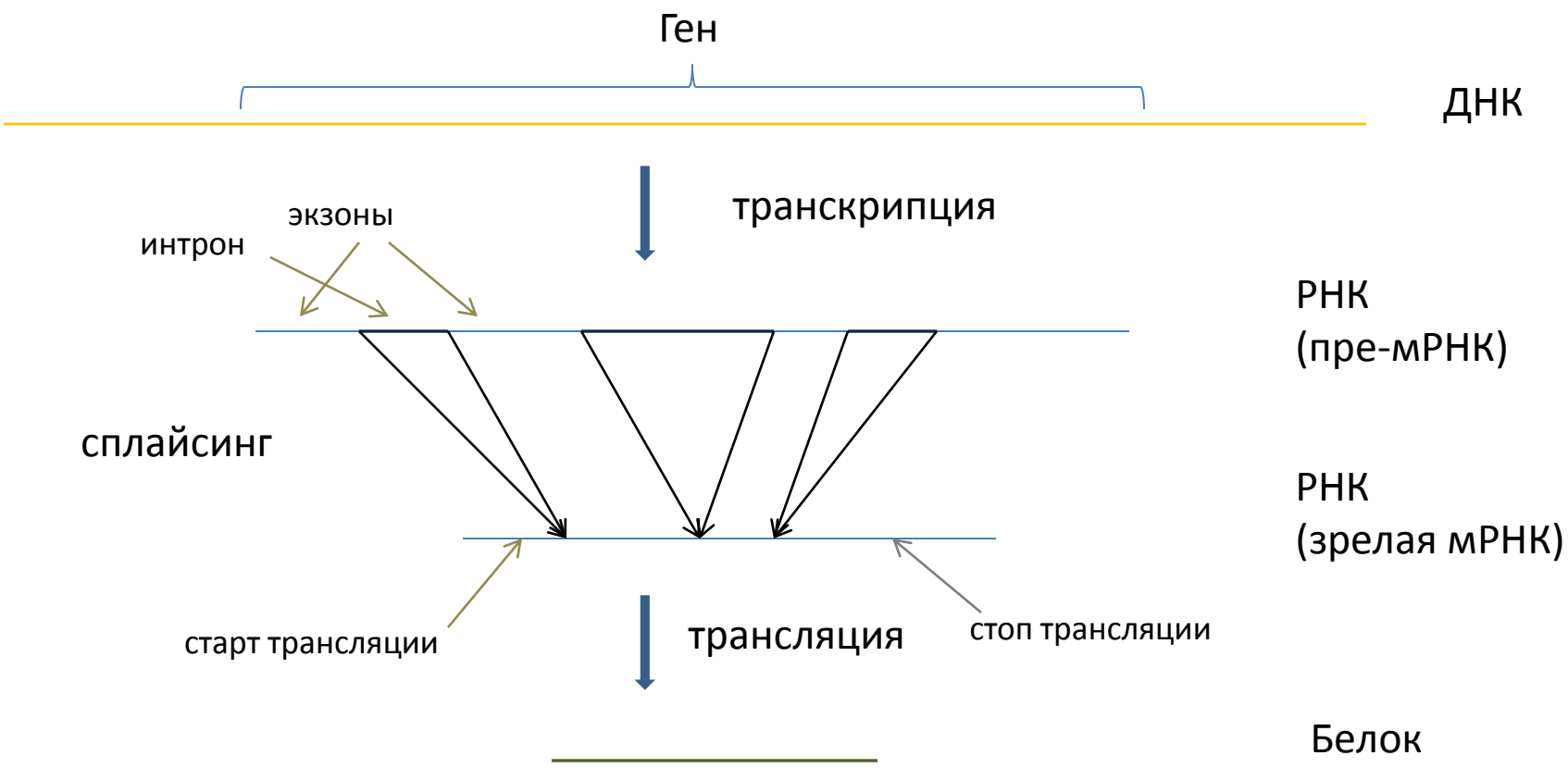
Трансмембранный белок

Галородопсин из археи *Natronomonas*

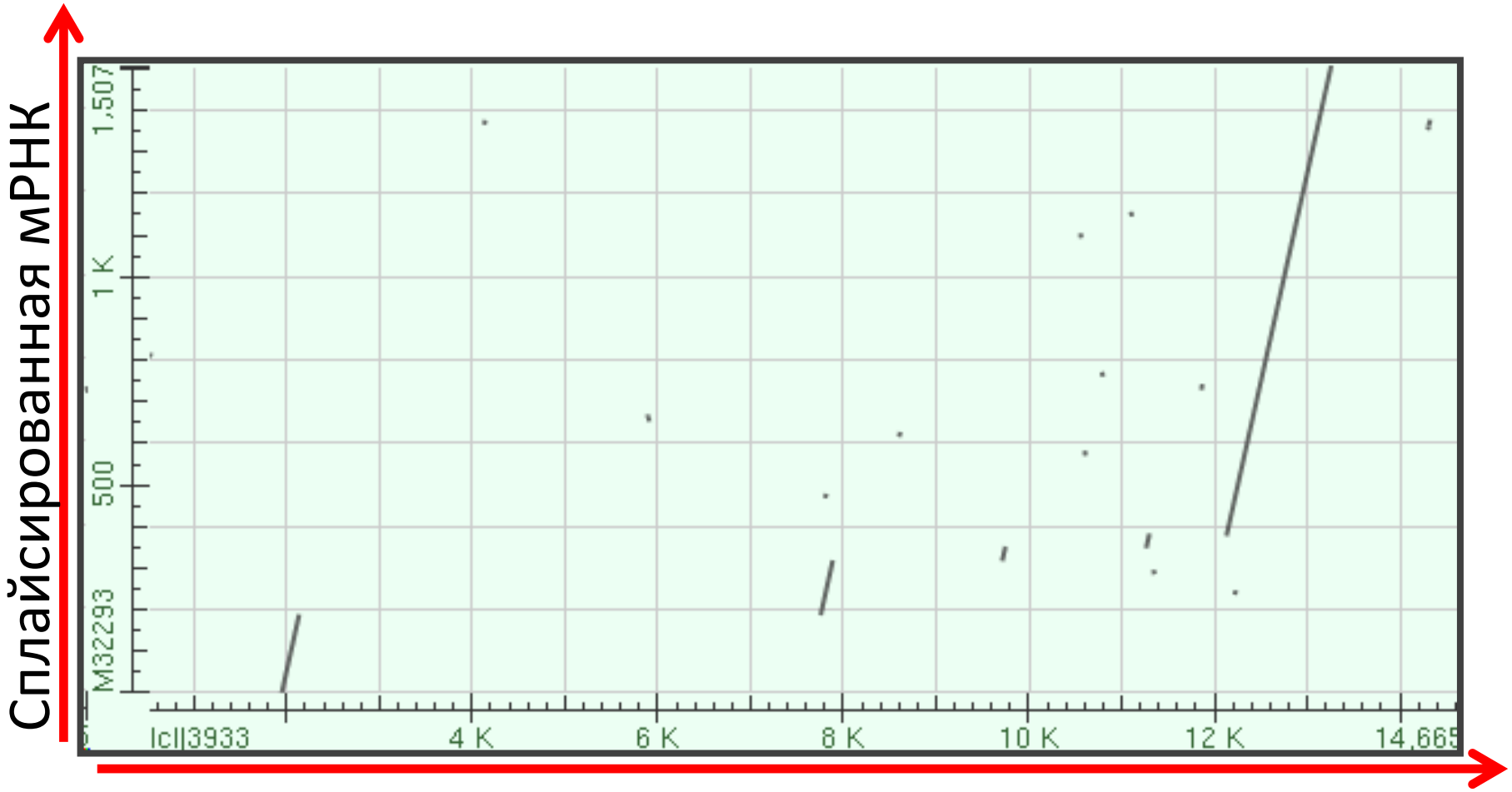


От ДНК к белку

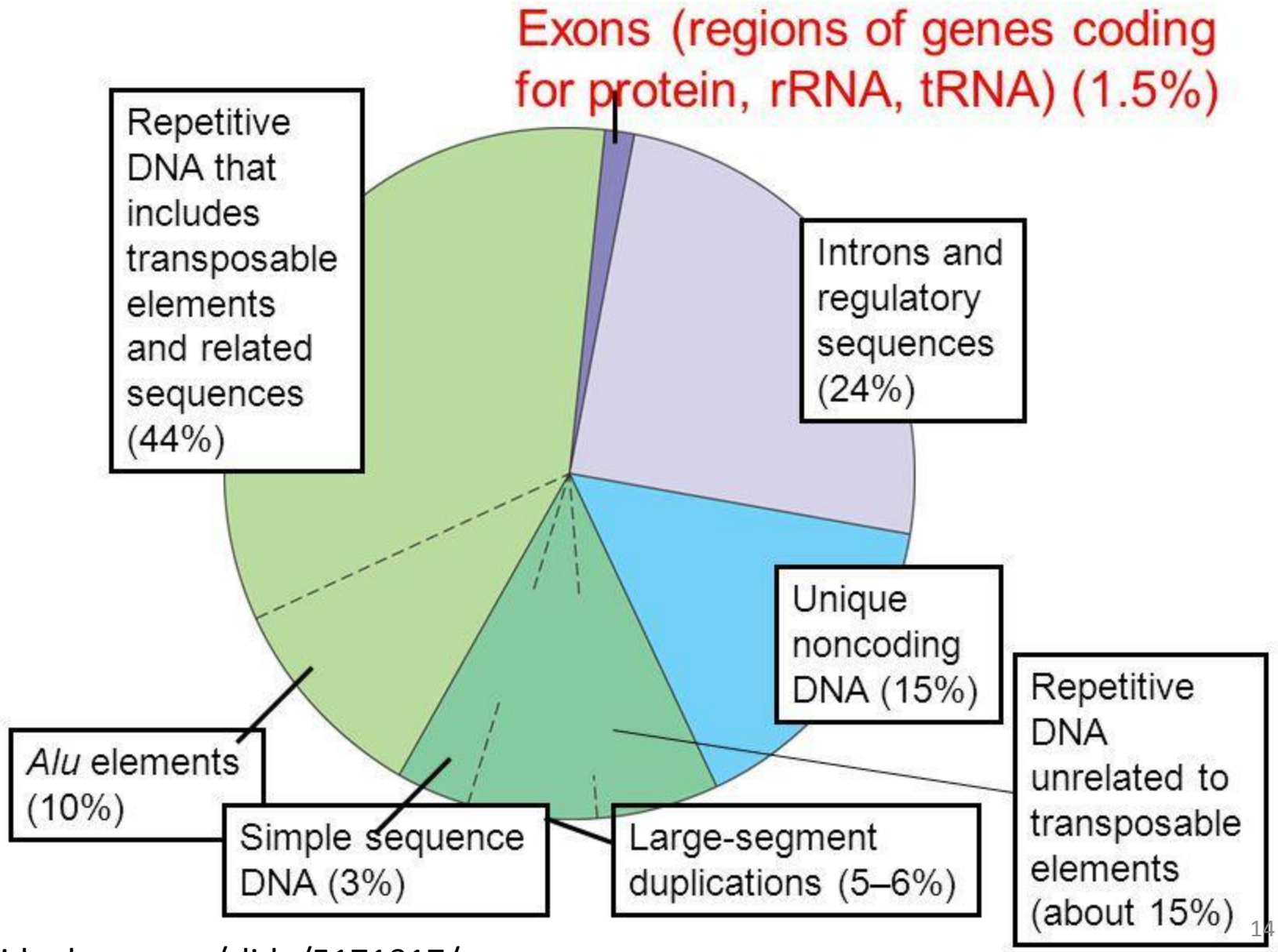
(эукариоты)



Карта сходства:
кусок генома *vs* мРНК

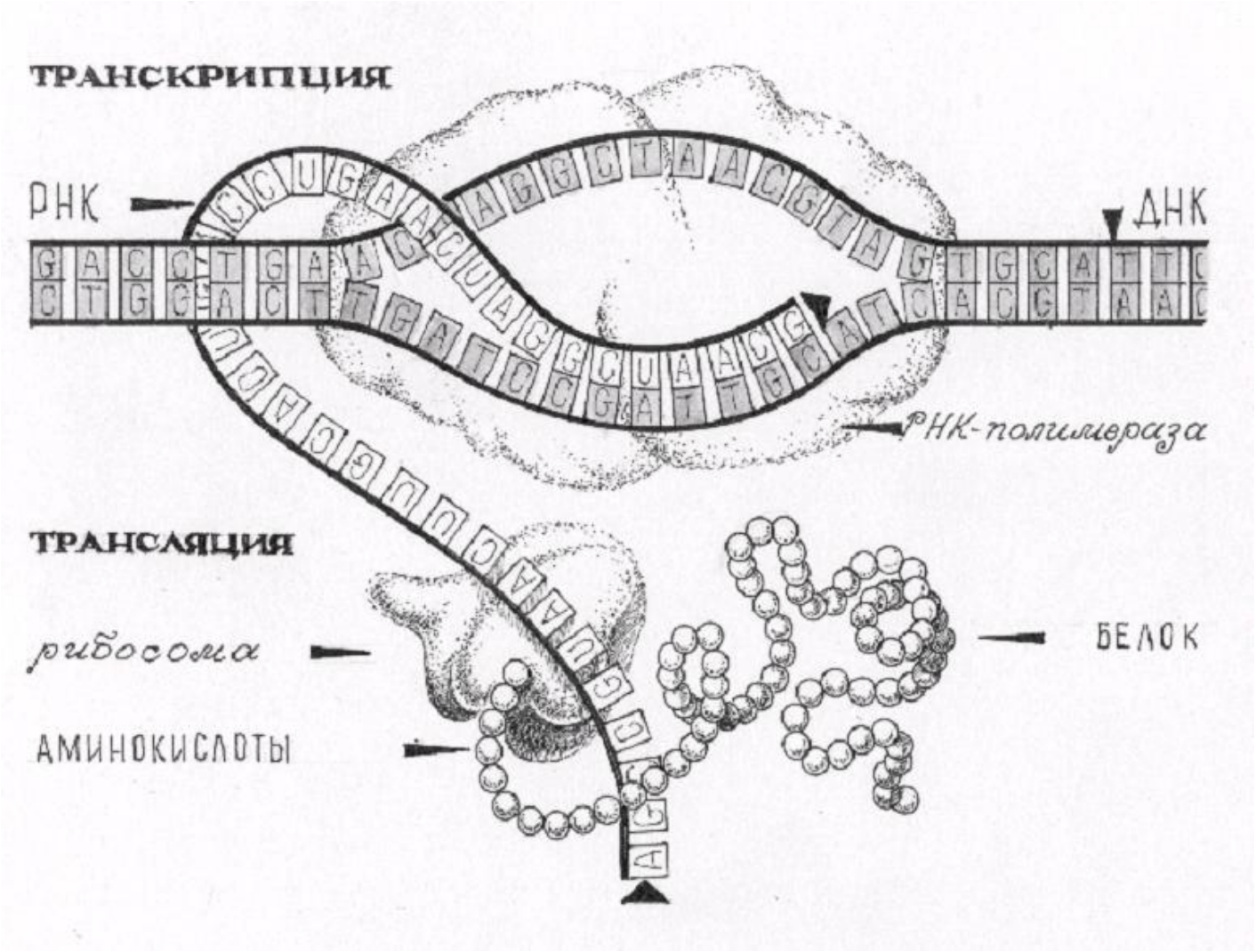


Overview of the human genome



http://slideplayer.com/slide/5171017/

Производство белка в прокариотах



Мутации

gatcaacactacttgacttcaa**g**acttaccataaagaaaac

gatcaacactacttgacttcaa**a**acttaccataaagaaaac

точечная замена

---

gatcaacactacttgacttcaa**ga**cttaccataaagaaaac

gatcaacactacttgacttcaacttaccataaagaaaac

делеция

---

gatcaacactacttgacttcaagacttaccataaagaaaac

gatcaacactacttgacttcaaga**ta**cttaccataaagaaaac

инсерция
(вставка)

Мутации

Бактерия разделилась, и у одного из потомков произошла ошибка репликации. Или произошло повреждение ДНК и ошибка репарации.

Что будет с потомством мутанта? Увидим ли мы эту мутацию, если отсеквенируем 1 000 000 бактерий этого штамма через 10 лет?

Потомство бактерии

В благоприятных условиях бактерия может делиться каждый час.

Сколько будет бактерий через 24 часа? А через год????

Потомство бактерии

В благоприятных условиях бактерия может делиться каждый час.

Сколько будет бактерий через 24 часа? А через год????

Ответ: примерно столько же, сколько сейчас.

Потомство бактерии

В благоприятных условиях бактерия может делиться каждый час.

Сколько будет бактерий через 24 часа? А через год????

Ответ: примерно столько же, сколько сейчас.

Численность подавляющего большинства популяций **постоянна** (по крайней мере на отрезках времени порядка лет) – погибает примерно столько же, сколько рождается.
Современная популяция человека – исключение!

Если члены популяции генетически идентичны, то вероятность оставить потомство для всех **одинакова** (точнее, зависит от только от внешних факторов).

Следствие: математическое ожидание числа потомков одной бактерии через достаточно большой промежуток времени равно 1.

Судьба нейтральной мутации

Предположим, что мутация **нейтральна** = никак не влияет на матожидание числа потомков (таких мутаций довольно много).

Мутация произошла и передаётся потомкам мутанта. Значит, в популяции появился новый **полиморфизм**. У данного варианта кода есть **частота** (сначала очень маленькая).

Судьба нейтральной мутации

Предположим, что мутация **нейтральна** = никак не влияет на матожидание числа потомков (таких мутаций довольно много).

Мутация произошла и передаётся потомкам мутанта. Значит, в популяции появился новый **полиморфизм**. У данного варианта генома есть **частота** (сначала очень маленькая).

Что произойдёт с частотой через пару суток?

Судьба нейтральной мутации

Предположим, что мутация **нейтральна** = никак не влияет на матожидание числа потомков (таких мутаций довольно много).

Мутация произошла и передаётся потомкам мутанта. Значит, в популяции появился новый **полиморфизм**. У данного варианта генома есть **частота** (сначала очень маленькая).

Что произойдёт с частотой через пару суток?

Ответ: частота либо немного возрастёт, либо немного упадёт. То и другое примерно равновероятно.

Случайное блуждание

Частота любого нейтрального полиморфизма постоянно колеблется случайным образом.
Математическая модель такого процесса называется «случайное блуждание».

На тротуаре стоит пьяный и каждые 10 сек. делает шаг либо направо, либо налево, случайно выбирая направление. Как далеко он уйдёт за время T?

Ответ: в среднем на расстояние, пропорциональное √T.

Случайное блуждание с поглощением

По длинной дамбе идёт пьяный и с каждым шагом отклоняется либо на полметра вправо, либо на полметра влево. Как скоро он свалится с дамбы?
Ответ: *скоро…*

Когда частота генетического варианта достигает 100% или 0%, процесс её изменения прекращается.

За исторически короткое время любой нейтральный вариант либо исчезает из популяции, либо закрепляется в ней!

Процесс закрепления новых нейтральных (или почти нейтральных) вариантов называется «генетический дрейф».

Генетический дрейф

Вероятность закрепиться для новой нейтральной мутации очень мала, но не 0.

Организмов в популяции много, мутаций в них происходит тоже много (примерно 10^{-8} на п.н. на поколение – каждая сотая новорождённая бактерия несёт новую мутацию). Значительная доля мутаций нейтральна.

Итог: геномы независимых популяций начинают различаться, чем дальше, тем больше – в них независимо накапливаются нейтральные мутации.

А если мутация не нейтральна?

Каждому варианту генома можно сопоставить его «приспособленность» f = матожидание числа потомков организма с таким геномом (через какой-то фиксированный промежуток времени).

В подавляющем большинстве случаев новая мутация порождает либо нейтральный вариант ($f = 1$) либо вредный ($f < 1$).

Вредный вариант тоже начинает «блуждать», но вероятность «шага вверх» оказывается меньше вероятности «шага вниз». Это очень сильно уменьшает вероятность закрепления – тем сильнее, чем меньше f, и тем сильнее, чем больше популяция.

Явление невозможности закрепления вредной мутации называется **стабилизирующий отбор** или же **отрицательный отбор**.

Положительный отбор

Если вдруг $f > 1$, то вероятность закрепления мутации вырастает во много раз. Процесс закрепления полезных мутаций называется **положительным отбором.**

Собственно, полезных мутаций так мало именно потому, что большинство возможных полезных мутаций уже закрепились.

Обычно полезные мутации начинают появляться в заметном количестве только при изменении условий жизни организмов – например при появлении нового источника пищи или новой опасности или попадании части популяции в другой климат…

А если родителей двое?

Пусть популяция по-прежнему стабильна.
Матожидание числа (выживших и оставивших своё потомство) потомков каждого индивида в следующем поколении равно теперь 2 (что очевидно, если подумать).

А если родителей двое?

Пусть популяция по-прежнему стабильна.
Матожидание числа (выживших и оставивших своё потомство) потомков каждого индивида в следующем поколении равно теперь 2 (что очевидно, если подумать).

А через много поколений?

На самом деле вероятны только два варианта: 0 и вся популяция (остальные крайне маловероятны).

А если родителей двое?

Пусть популяция по-прежнему стабильна.
Матожидание числа (выживших и оставивших своё потомство) потомков каждого индивида в следующем поколении равно теперь 2 (что очевидно, если подумать).

А через много поколений?

На самом деле вероятны только два варианта: 0 и вся популяция (остальные крайне маловероятны).

Что же, вся изложенная теория не работает?

Будем следить за конкретным геном

Давайте рассмотрим не популяцию организмов, а популяцию генов!

Организмы, в геномах которых они присутствуют, для них – часть окружающей среды. Размножаются они по тем же законам. что и бактерии.

Будем следить за конкретным геном

Давайте рассмотрим не популяцию организмов, а популяцию генов!

Организмы, в геномах которых они присутствуют, для них – часть окружающей среды. Размножаются они по тем же законам. что и бактерии.

Тогда вся формальная теория сохраняется. У каждого варианта гена есть приспособленность = матожидание числа потомков через фиксированное время. Нейтральные варианты исчезают или некоторой вероятностью закрепляются (дрейф), вредные исчезают (отрицательный отбор), полезные с заметной вероятностью закрепляются (положительный отбор).

А что такое «ген»? В данном контексте – что хотите, любой участок генома. Можно даже эволюцию конкретной позиции в геноме отслеживать с тех же позиций.

Закрепление мутаций

Вероятность закрепления нейтральной мутации обратно пропорциональна размеру популяции.
Число мутаций, возникающих в единицу времени в популяции, пропорционально её размеру.
Поэтому скорость генетического дрейфа не зависит от размера популяции.

Закрепление мутаций

Вероятность закрепления нейтральной мутации обратно пропорциональна размеру популяции.
Число мутаций, возникающих в единицу времени в популяции, пропорционально её размеру.
Поэтому скорость генетического дрейфа не зависит от размера популяции.

Вероятность закрепления полезной мутации почти не зависит от размера популяции.
Поэтому скорость положительного отбора тем выше, чем больше популяция.

Вероятность закрепления слабовредной мутации очень быстро падает с размером популяции (быстрее, чем $1/N$).
Поэтому в маленьких популяциях чаще закрепляются вредные мутации.
Точнее: при размере популяции менее 10 000 вероятность закрепления слабовредной мутации становится не пренебрежимой.

«Отбор эффективен только в больших популяциях»
(Это объясняет, почему видовое разнообразие крупных животных так невелико по сравнению с мелкими, и почему крупные виды быстро вымирают)

Точечные замены в кодирующей последовательности

... AATCCGTCAAGTCTA...

... Asn Pro Ser Ser Leu ...

1) “молчащая”(синонимическая)мутация

... AATCCGTCGAGTCTA...

... Asn Pro Ser Ser Leu ...

2) замена остатка на близкий по свойствам

... AATCCGACAAGTCTA...

... Asn Pro Thr Ser Leu ...

3) замена остатка на остаток с иными свойствами

... AATCCGTCAAGACTA...

... Asn Pro Ser Arg Leu ...

Эволюция белков

Мутации возникают случайно.

Конкретная мутация может быть:

- летальной;
- вредной;
- слабовредной;
- нейтральной;
- полезной.

Мутация порождает **полиморфизм данного белка в популяции.**
Доля каждого варианта подвержена случайным изменением (модель: «случайное блуждание с поглощением»).
За исторически короткое время один из вариантов (старый или новый) исчезает.
Во втором случае говорят, что мутация **закрепилась.**

Мы видим лишь закрепившиеся мутации

… а шанс закрепиться есть лишь у безвредных мутаций.

Выравнивание двух белков:

```
CYB5_CHICK          1 MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHP     50
                        .:|...||  .:||.|||:||||:|:|||:|:||::||:||||:|||
CYB5_HUMAN          1 ---MAEQSDEA--VKYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHP     45

CYB5_CHICK         51 GGEEVLREQAGGDATENFEDVGHSTDARALSETFIIGELHPDDRPKLQKP    100
                      |||||||||||||||||||||||||.:|:|||||||||||||||.||
CYB5_HUMAN         46 GGEEVLREQAGGDATENFEDVGHSTDAREMSKTFIIGELHPDDRPKLNKP     95

CYB5_CHICK        101 AETLITTVQSNSSSWSNWVIPAIAAIIVALMYRSYMSE-    138
                      .||||||:.|:||.|:||||||:|:.||||||.||:|
CYB5_HUMAN         96 PETLITTIDSSSSWWTNWVIPAISAVAVALMYRLYMAED    134
```

Эти белки начали эволюционировать независимо около 250 млн. лет назад
За это время во всех позициях, где были возможны нейтральные мутации, они произошли и закрепились, во многих не по одному разу.

Бывают и более консервативные белки...

```
ACTS_HUMAN         1 MCDEDETTALVCDNGSGLVKAGFAGDDAPRAVFPSIVGRPRHQGVMVGMG     50
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK         1 MCDEDETTALVCDNGSGLVKAGFAGDDAPRAVFPSIVGRPRHQGVMVGMG     50

ACTS_HUMAN        51 QKDSYVGDEAQSKRGILTLKYPIEHGIITNWDDMEKIWHHTFYNELRVAP    100
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK        51 QKDSYVGDEAQSKRGILTLKYPIEHGIITNWDDMEKIWHHTFYNELRVAP    100

ACTS_HUMAN       101 EEHPTLLTEAPLNPKANREKMTQIMFETFNVPAMYVAIQAVLSLYASGRT    150
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK       101 EEHPTLLTEAPLNPKANREKMTQIMFETFNVPAMYVAIQAVLSLYASGRT    150

ACTS_HUMAN       151 TGIVLDSGDGVTHNVPIYEGYALPHAIMRLDLAGRDLTDYLMKILTERGY    200
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK       151 TGIVLDSGDGVTHNVPIYEGYALPHAIMRLDLAGRDLTDYLMKILTERGY    200

ACTS_HUMAN       201 SFVTTAEREIVRDIKEKLCYVALDFENEMATAASSSSLEKSYELPDGQVI    250
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK       201 SFVTTAEREIVRDIKEKLCYVALDFENEMATAASSSSLEKSYELPDGQVI    250

ACTS_HUMAN       251 TIGNERFRCPETLFQPSFIGMESAGIHETTYNSIMKCDIDIRKDLYANNV    300
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK       251 TIGNERFRCPETLFQPSFIGMESAGIHETTYNSIMKCDIDIRKDLYANNV    300

ACTS_HUMAN       301 MSGGTTMYPGIADRMQKEITALAPSTMKIKIIAPPERKYSVWIGGSILAS    350
                     ||||||||||||||||||||||||||||||||||||||||||||||||||
ACTS_CHICK       301 MSGGTTMYPGIADRMQKEITALAPSTMKIKIIAPPERKYSVWIGGSILAS    350

ACTS_HUMAN       351 LSTFQQMWITKQEYDEAGPSIVHRKCF    377
                     ||||||||||||||||||||||||||||
ACTS_CHICK       351 LSTFQQMWITKQEYDEAGPSIVHRKCF    377
```

Матрица замен аминокислот

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	0	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	0	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

Задача выравнивания

>CYB5_CHICK
MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHPGGEEVLREQA
GGDATENFEDVGHSTDARALSETFIIGELHPDDRPKLQKPAETLITTVQSNSSSWSNWVI
PAIAAIIVALMYRSYMSE
>CYB5_HUMAN
MAEQSDEAVKYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHPGGEEVLREQAGGDAT
ENFEDVGHSTDAREMSKTFIIGELHPDDRPKLNKPPETLITTIDSSSSWWTNWVIPAISA
VAVALMYRLYMAED

Как сопоставить буквы одной последовательности буквам другой?

Хочется не возиться с каждой парой последовательностей, а поручить это дело компьютеру…

Задача выравнивания

Биологическая задача: сопоставить буквы одной последовательности буквам другой, чтобы соответствующие буквы имели общее происхождение (построить правильное выравнивание)

Формализация: каждому возможному выравниванию сопоставить число – его качество (обычно называется “вес”, по-английски Score), так, чтобы правильное выравнивание по возможности отличалось от неправильных льшим весом.

Алгоритмическая задача: придумать алгоритм, находящий выравнивание с самым м весом за разумное время.

Змей-горыныч биоинформатики

Формализация 1: вес глобального выравнивания

```
CYB5_CHICK         1 MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHP     50
                     .:|...||  .:||.|||:||||:|:|||:|:||::||:||||:|||
CYB5_HUMAN         1 ---MAEQSDEA--VKYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHP     45

CYB5_CHICK        51 GGEEVLREQAGGDATENFEDVGHSTDARALSETFIIGELHPDDRPKLQKP    100
                     ||||||||||||||||||||||||||.:|:|||||||||||||||.||
CYB5_HUMAN        46 GGEEVLREQAGGDATENFEDVGHSTDAREMSKTFIIGELHPDDRPKLNKP     95

CYB5_CHICK       101 AETLITTVQSNSSSWSNWVIPAIAAIIVALMYRSYMSE-    138
                     .||||||:.|:||.|:||||||:|:.||||||.||:|
CYB5_HUMAN        96 PETLITTIDSSSSWWTNWVIPAISAVAVALMYRLYMAED    134
```

Вес выравнивания = сумма весов **позиций** выравнивания

Вес позиции, если в ней сопоставлены две буквы равен соответствующему элементу матрицы замен (например, для позиции 4 он равен −1, потому что в ней сопоставлены M и S).

За каждый символ несоответствия («гэп», в данном случае это минус) из веса вычитается некоторое положительное число («штраф за гэп»)

Проблема 1: адекватность формализации

- Оптимальное выравнивание = максимальное по весу
- Эволюционно правильное выравнивание: в каждой колонке стоят гомологичные (имеющие общее происхождение) буквы.

Это не всегда одно и то же!

Проблема 2: реализация

Нужен алгоритм, находящий оптимальное выравнивание.

Решение 1: перепробуем все варианты выравнивания и для каждого посчитаем вес, потом выберем лучшее.

Можно посчитать, что для двух последовательностей длины 100 имеется 10^{59} различных вариантов их выравнивания.

Ни один компьютер не перепробует все варианты даже за тысячу лет...

Проблема 2: реализация

Нужен алгоритм, находящий оптимальное выравнивание.

Решение 2: алгоритм динамического программирования – от части к целому.

Идея: посчитаем оптимальное выравнивание сначала для начальных отрезков – «префиксов» обеих последовательностей.

Формула для веса выравнивания подобрана так, что если мы знаем оптимальное выравнивание для:

1. первых *k* букв (*k*-префикса) первой последовательности и *m*-префикса второй
2. (*k*+1)-префикса первой и *m*-префикса второй
3. *k*-префикса первой и (*m*+1)-префикса второй

то мы можем найти оптимальное выравнивание (*k*+1)-префикса первой и (*m*+1)-префикса второй, перебрав всего три варианта!

Идея динамического программирования

Пусть известны:

1. Оптимальное выравнивание первых *k* букв первой и *m* букв второй

```
Seq1 MVGSSEAGGEAWRGRYYRLEEV     k
          |...||  .:||.|||:
Seq1 ---MAEQSDEA--VKYYTLEEI     m
```

2. Оптимальное выравнивание первых *k* букв первой и *m* + 1 букв второй

```
Seq1 MVGSSEAGGEAWRGRYYRLEEV-     k
          |...||  .:||.|||:
Seq1 ---MAEQSDEA--VKYYTLEEIQ     m+1
```

3. Оптимальное выравнивание первых *k* +1 букв первой и *m* букв второй

```
Seq1 MVGSSEAGGEAWRGRYYRLEEVQ     k+1
          |...||  .:||.|||:
Seq1 ---MAEQSDEA--VKYYTLEEI-     m
```

Идея динамического программирования

Теперь, чтобы найти оптимальное выравнивание первых *k*+1 букв первой и *m*+1 букв второй, надо выбрать всего между тремя вариантами:

```
Seq1 MVGSSEAGGEAWRGRYYRLEEVQ     k+1
          |...||  .:||.|||:|
Seq1 ---MAEQSDEA--VKYYTLEEIQ     m+1
```

```
Seq1 MVGSSEAGGEAWRGRYYRLEEV-Q     k+1
          |...||  .:||.|||:
Seq1 ---MAEQSDEA--VKYYTLEEIQ-     m+1
```

```
Seq1 MVGSSEAGGEAWRGRYYRLEEVQ-     k+1
          |...||  .:||.|||:
Seq1 ---MAEQSDEA--VKYYTLEEI-Q     m+1
```

Алгоритм Нидлмана – Вунша (Needleman&Wunsch)

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Рис. из Википедии

Матрица $(n_1+1)\times(n_2+1)$ заполняется слева направо и сверху вниз весами лучших выравниваний двух **префиксов** исходных последовательностей и стрелками.
Стрелка показывает последний шаг **лучшего** пути в данную клетку.
После заполнения матрицы выравнивание восстанавливается движением по стрелкам, начиная с правого нижнего угла.

Алгоритм Нидлмана – Вунша (Needleman&Wunsch)

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Рис. из Википедии

Время работы пропорционально произведению длин последовательностей
(что намного меньше числа различных выравниваний,
равного числу сочетаний из n_1+n_2 по n_1)

Формализация 1а: аффинные штрафы за гэпы

В реальности одна большая делеция более вероятна, чем две малых той же суммарной длины. Поэтому хочется вычитать из веса выравнивания штраф за делецию/вставку не в виде суммы по всем гэпам, а более умным способом, зависящим от длины инделя.

Но: не удаётся создать аналог алгоритма Нидлмана – Вунша для произвольной функции зависимости величины штрафа от длины инделя ☹

Компромисс: штраф имеет вид $G + E \cdot L$, где L – длина инделя, G и E – положительные числа, причём $G > E$.
Для такой (аффинной) зависимости штрафа от длины существует аналог алгоритма Нидлмана – Вунша, работающий всего в три раза медленнее исходного.

Формализация 2: вес локального выравнивания

Часто бывает, что нужно выровнять короткую и длинную последовательность, причём в длинной есть небольшой кусок, сходный с короткой. Алгоритм Нидлмана – Вунша (точнее, соответствующая формализация) в этом случае работает плохо из за большого груза штрафов за концевые гэпы.

Вес локального выравнивания вычисляется только по позициям, заключённым между первым и последним сопоставлением букв (концевые гэпы игнорируются).

Разработан алгоритм Смита – Уотермэна (Smith&Waterman), похожий на алгоритм Нидлмана – Вунша, но выдающий оптимальное по такому весу (так называемое **оптимальное локальное**) выравнивание.

Локальное выравнивание

Задачу локального выравнивания можно сформулировать так:

1. выбрать фрагмент первой последовательности и
2. фрагмент второй последовательности и
3. выравнивание этих фрагментов

так чтобы вес был максимальным (среди всех возможных в пунктах 1, 2, 3 выборов).

Пример локального выравнивания

```
>ANTP_DROME P02833 Homeotic protein antennapedia
MTMSTNNCESMTSYFTNSYMGADMHHGHYPGNGVTDLDAQQMHHYSQNANHQGNMPYPRF
PPYDRMPYYNGQGMDQQQQHQVYSRPDSPSSQVGGVMPQAQTNGQLGVPQQQQQQQQQPS
QNQQQQQAQQAPQQLQQQLPQVTQQVTHPQQQQQQPVVYASCKLQAAVGGLGMVPEGGSP
PLVDQMSGHHMNAQMTLPHHMGHPQAQLGYTDVGVPDVTEVHQNHHNMGMYQQQSGVPPV
GAPPQGMMHQGQGPPQMHQGHPGQHTPPSQNPNSQSSGMPSPLYPWMRSQFGKCQERKRG
RQTYTRYQTLELEKEFHFNRYLTRRRRIEIAHALCLTERQIKIWFQNRRMKWKKENKTKG
EPGSGGEGDEITPPNSPQ

>MTAL2_YEAST P0CY08 Mating-type protein ALPHA2 (MATalpha2 protein) (Alpha-2 repressor)
MNKIPIKDLLNPQITDEFKSSILDINKKLFSICCNLPKLPESVTTEEEVELRDILGFLSR
ANKNRKISDEEKKLLQTTSQLTTTITVLLKEMRSIENDRSNYQLTQKNKSADGLVFNVVT
QDMINKSTKPYRGHRFTKENVRILESWFAKNIENPYLDTKGLENLMKNTSLSRIQIKNWV
SNRRRKEKTITIAPELADLLSGEPLAKKKE
```

```
ANTP_DROME       288 RSQFGKCQERKRGRQTYTRYQTLELEKEFH---FNRYLTRRRRIEIAHAL    334
                     :....|..:..||.: :|:.....||..|.   .|.||..:...:....
MTAL2_YEAST      121 QDMINKSTKPYRGHR-FTKENVRILESWFAKNIENPYLDTKGLENLMKNT    169

ANTP_DROME       335 CLTERQIKIWFQNRRMKWKK-------ENKTKGEP    362
                     .|:..|||.|..|||.|.|.       .:...|||
MTAL2_YEAST      170 SLSRIQIKNWVSNRRRKEKTITIAPELADLLSGEP    204
```

Алгоритм даёт ответ всегда, даже если осмысленного ответа не существует!

Оптимальное выравнивание двух неродственных последовательностей:

- глобальное:

```
CYB5_CHICK         1 MVGSSEAGGEAWRGRYYRLEEV--------QKHNNSQSTWIIVHHRIYDI     42
                                     :.::        |..:..:|:.:.::.:::.|
MTAL2_YEAST        1 ------------------MNKIPIKDLLNPQITDEFKSSILDINKKLFSI     32

CYB5_CHICK        43 TKFLDEHPGG----EEV-LREQAG--GDATENFEDVGHSTDARALSETFI     85
                     ...|.:.|..    ||| ||:..|  ..|.:|          |.:|
MTAL2_YEAST       33 CCNLPKLPESVTTEEEVELRDILGFLSRANKN----------RKIS----     68

CYB5_CHICK        86 IGELHPDDRPKLQKPAETLITTV-------QSNSSSWSNWVIPAIAAIIV    128
                           |:..||.:....|.||:       :|..:..||:.:........
MTAL2_YEAST       69 ------DEEKKLLQTTSQLTTTITVLLKEMRSIENDRSNYQLTQKNKSAD    112

CYB5_CHICK       129 ALMYRSYMSE-------------------------------------     138
                     .|::.....:
MTAL2_YEAST      113 GLVFNVVTQDMINKSTKPYRGHRFTKENVRILESWFAKNIENPYLDTKGL    162

CYB5_CHICK       138 ---------------------------------------------    138

MTAL2_YEAST      163 ENLMKNTSLSRIQIKNWVSNRRRKEKTITIAPELADLLSGEPLAKKKE    210
```

- локальное:

```
CYB5_CHICK         92 DDRPKLQKPAETLITTV    108
                      |:..||.:....|.||:
MTAL2_YEAST        69 DEEKKLLQTTSQLTTTI     85
```

А ещё бывает множественное выравнивание

```
CLUSTAL W (1.83) multiple sequence alignment


CYB5_CHICK      MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHPGGEEVLREQA
CYB5_HUMAN      ---MAEQSDEAV--KYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHPGGEEVLREQA
CYB5_HORSE      ---MAEQSDKAV--KYYTLEEIKKHNHSKSTWLILHHKVYDLTKFLEDHPGGEEVLREQA
CYB5_MUSDO      ------MSSEDV--KYFTRAEVAKNNTKDKNWFIIHNNVYDVTAFLNEHPGGEEVLIEQA
CYB5_DROME      ------MSSEET--KTFTRAEVAKHNTNKDTWLLIHNNIYDVTAFLNEHPGGEEVLIEQA


CYB5_CHICK      GGDATENFEDVGHSTDARALSETFIIGELHPDDRPKLQKPAE-TLITTVQSNSSSWSNWV
CYB5_HUMAN      GGDATENFEDVGHSTDAREMSKTFIIGELHPDDRPKLNKPPE-TLITTIDSSSSWWTNWV
CYB5_HORSE      GGDATENFEDIGHSTDARELSKTFIIGELHPDDRSKIAKPVE-TLITTVDSNSSWWTNWV
CYB5_MUSDO      GKDATEHFEDVGHSSDAREMMKQYKVGELVAEERSNVPEKSEPTWNTEQKTEESSMKSWL
CYB5_DROME      GKDATENFEDVGHSNDARDMMKKYKIGELVESERTSVAQKSEPTWSTEQQTEESSVKSWL


CYB5_CHICK      IPAIAAIIVALMYRSYMSE---
CYB5_HUMAN      IPAISAVAVALMYRLYMAED--
CYB5_HORSE      IPAISAVVVALMYRIYTAED--
CYB5_MUSDO      MPFVLGLVATLIYKFFFGTKSQ
CYB5_DROME      VPLVLCLVATLFYKFFFGGAKQ
```

Визуализация множественного выравнивания