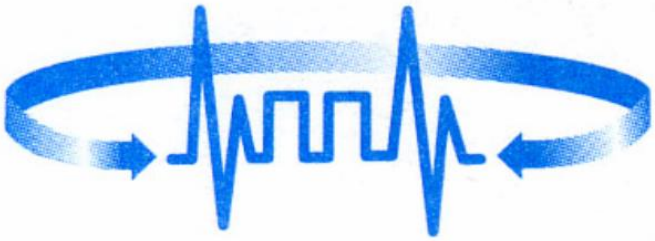
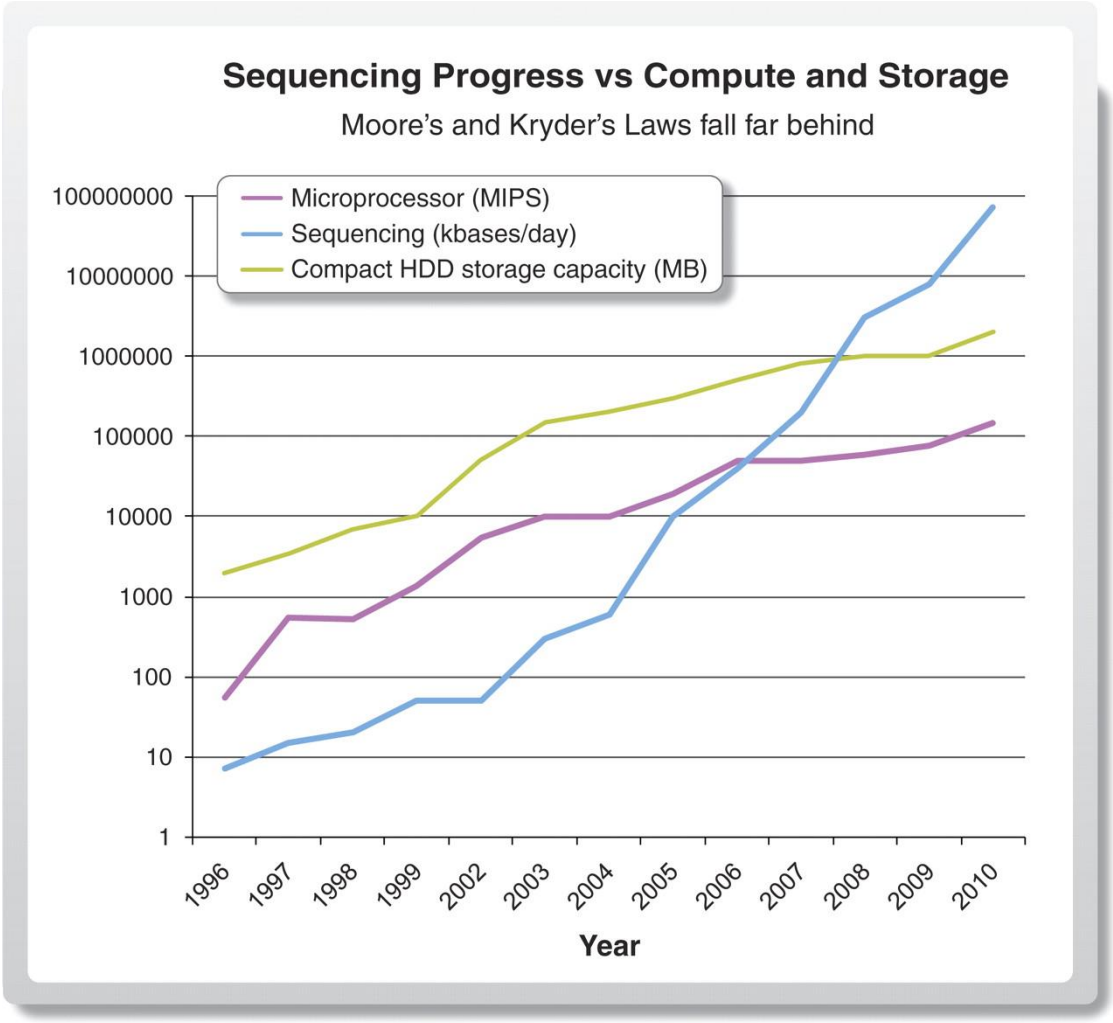


Биоинформатика. Молекулярная биология между пробиркой и компьютером

Михаил Гельфанд

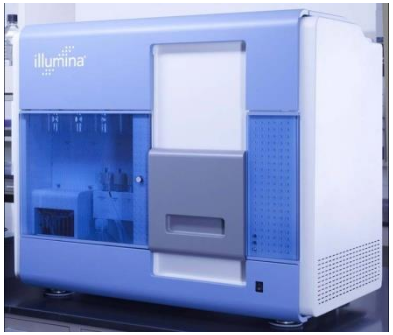
Малый мехмат
11.IV.2015

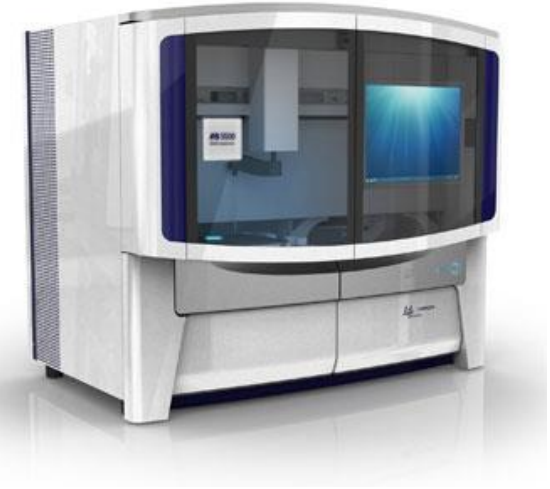
Fig. 1 A doubling of sequencing output every 9 months has outpaced and overtaken performance improvements within the disk storage and high-performance computation fields.



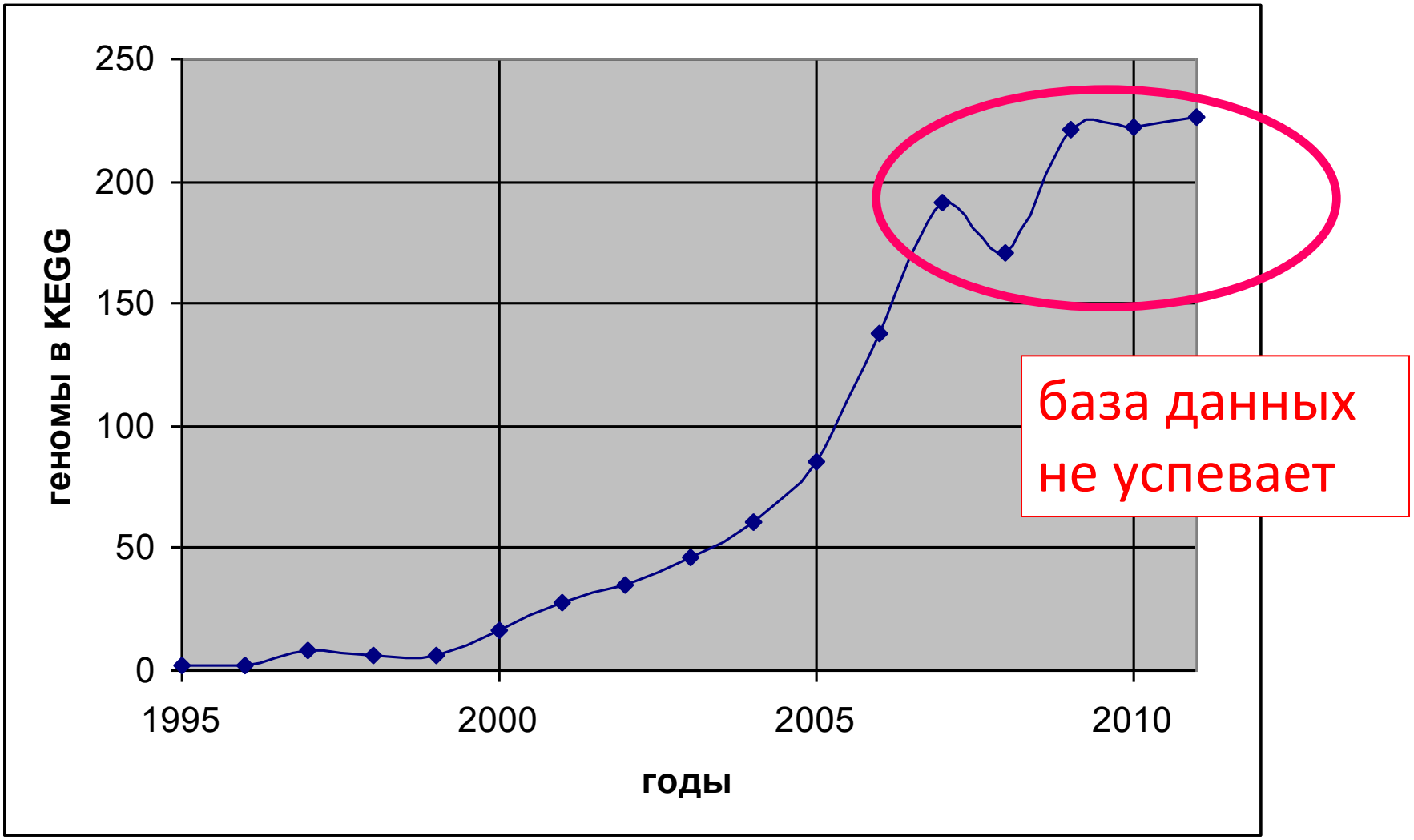
S D Kahn Science 2011;331:728-729

Вот они, эти чудовища





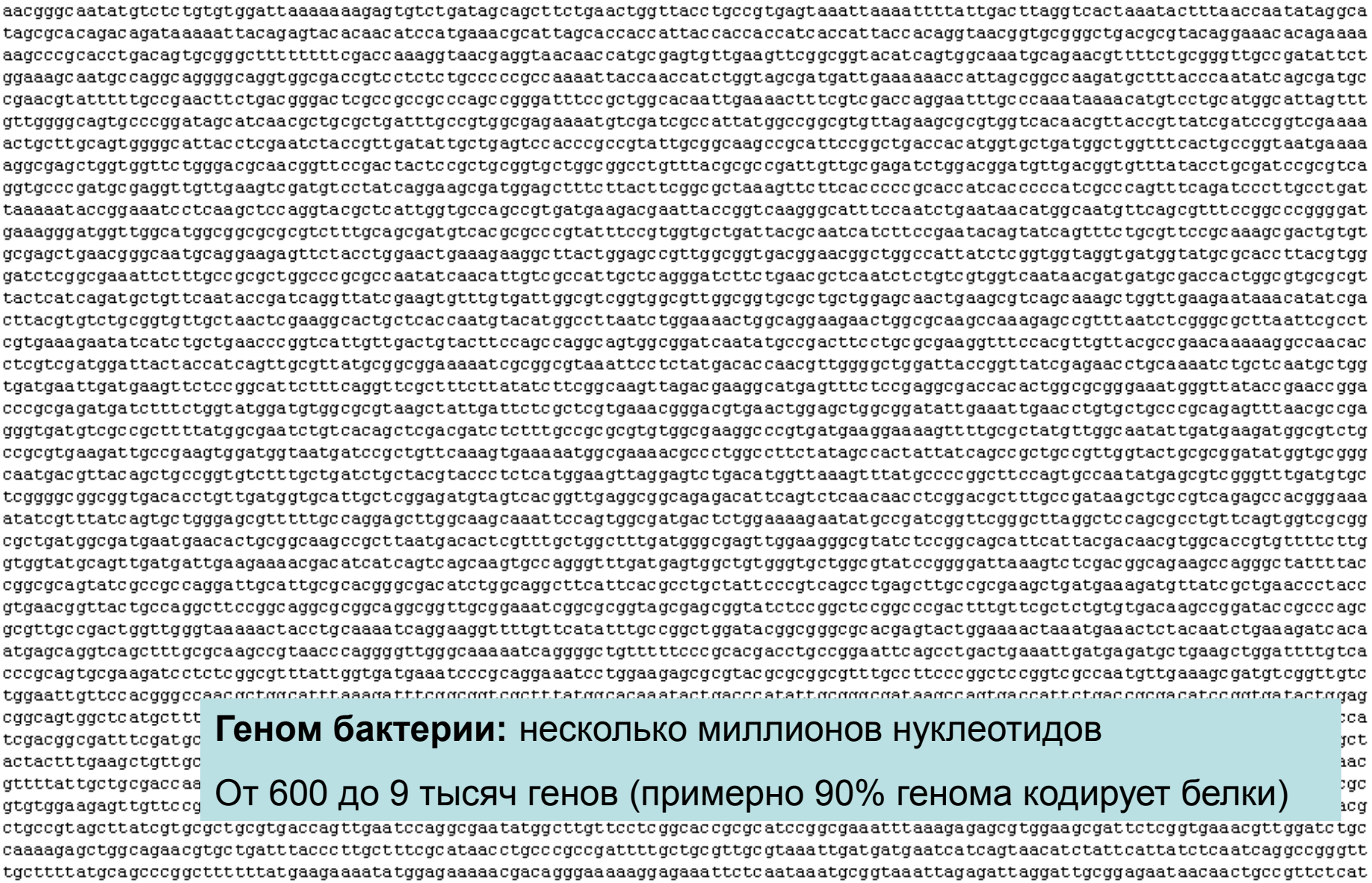
1464 расшифрованных геномов прокариот
(на самом деле, уже много больше)



Расшифрован ли геном?

Перехватить зашифрованное сообщение – ещё не значит его понять

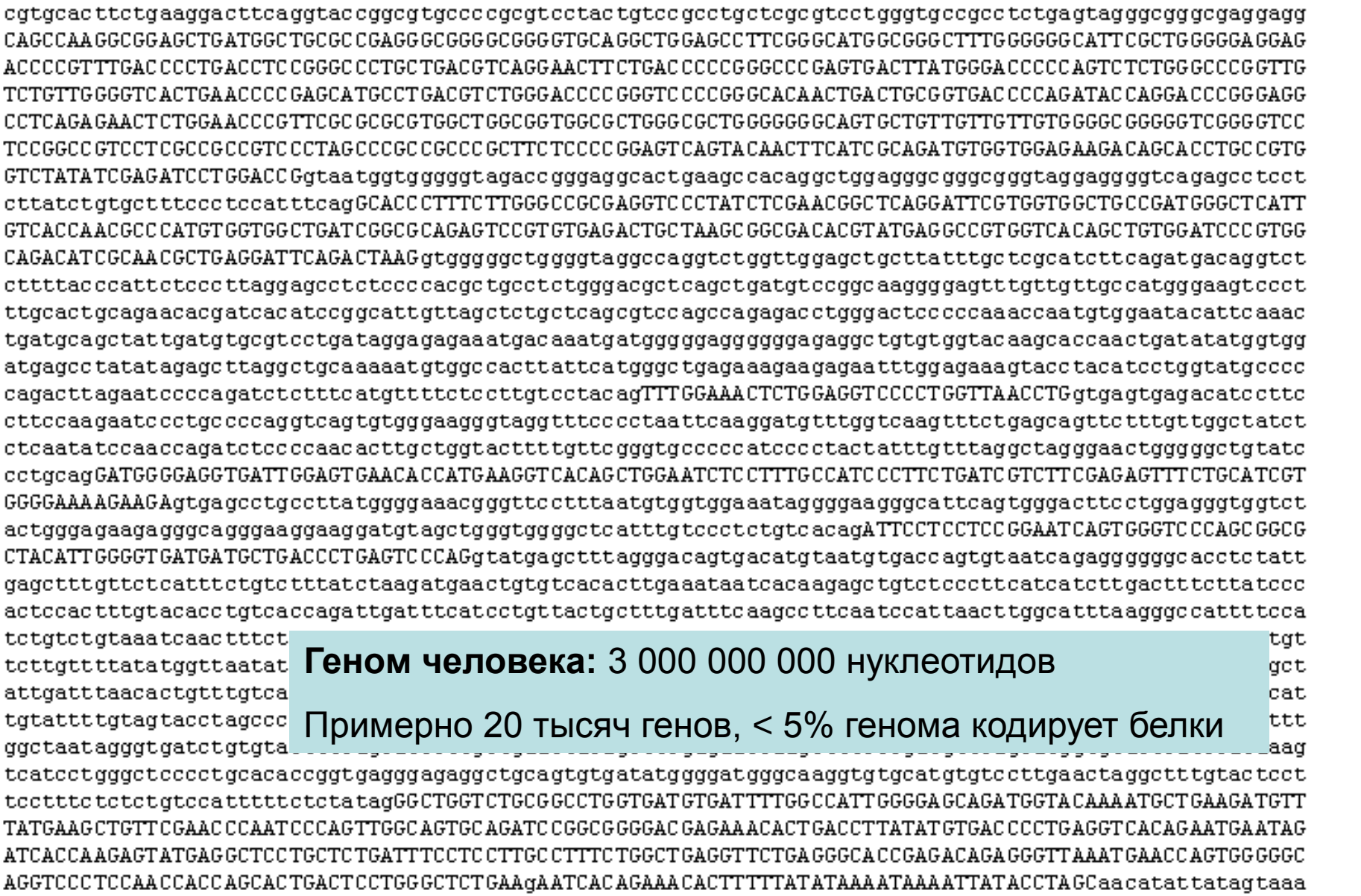
0.1% генома *E. coli*



Геном бактерии: несколько миллионов нуклеотидов

От 600 до 9 тысяч генов (примерно 90% генома кодирует белки)

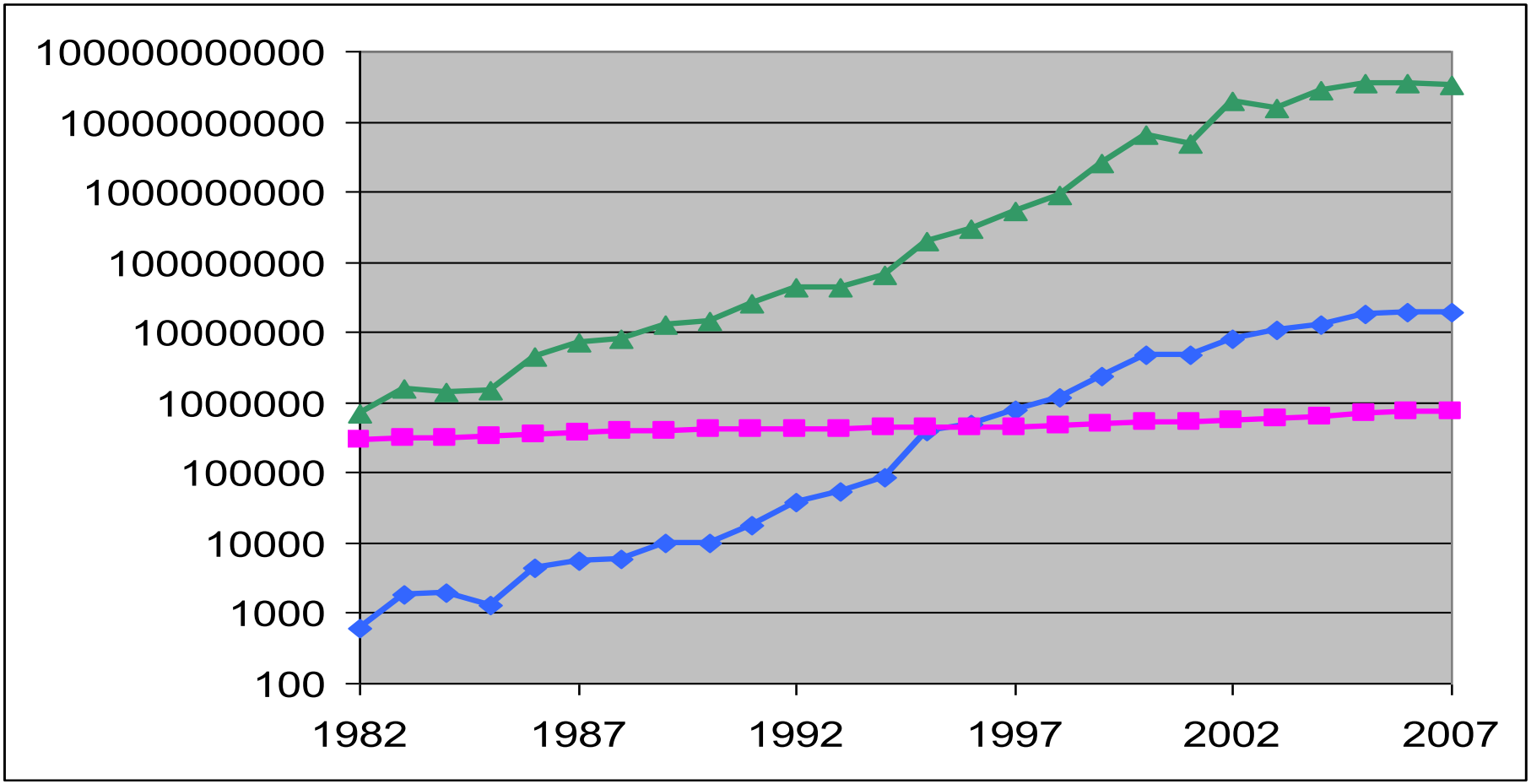
0.0001% генома человека



Геном человека: 3 000 000 000 нуклеотидов

Примерно 20 тысяч генов, < 5% генома кодирует белки

Рост объема данных



красный – статьи (PubMed)

голубой – фрагменты ДНК (GenBank)

зеленый – нуклеотиды (GenBank)

из 18 млн. статей в PubMed, ~675 тыс. имеют ключевое слово “bioinformat* OR comput*”

Задачи

- Картирование генов и составление списка белков, структурных и функциональных РНК и т.п.

- Функциональная аннотация генов и белков
  - биологическая функция (что делает)
  - регуляция (в каких условиях работает)

- Функциональная аннотация геномов
  - метаболическая реконструкция и моделирование
  - регуляторные сети, моделирование развития
  - предсказание свойств организма по геному

Идентификация генов

- Основные идеи придуманы (в 80-90х гг.) и реализованы (в 90-2000х).
- Постепенное улучшение программ

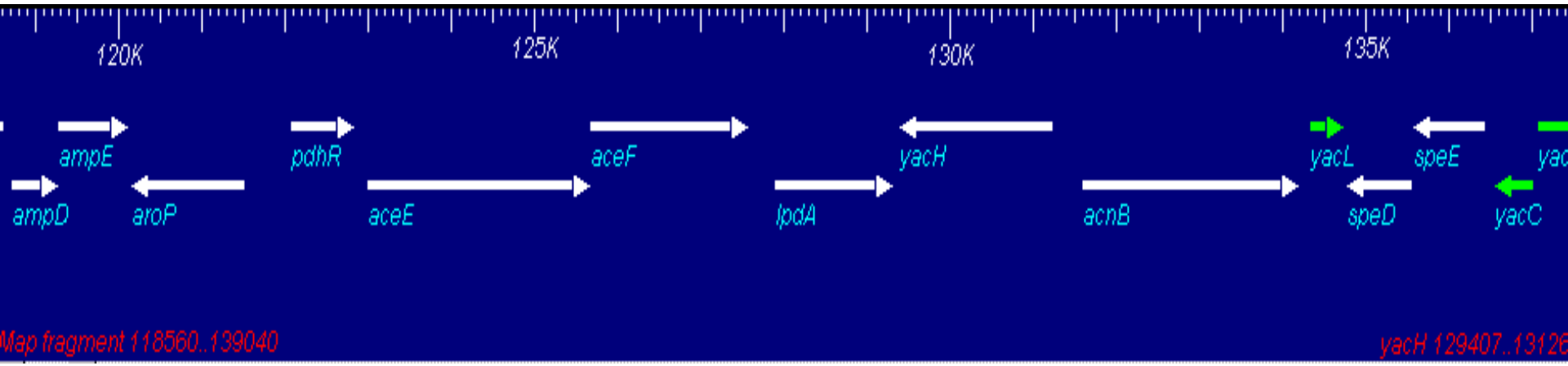


Таблица генетического кода

TTT	F	TCT	S	TAT	Y	TGT	C
TTC	F	TCC	S	TAC	Y	TGC	C
TTA	L	TCA	S	TAA	*stop*	TGA	*stop*
TTG	L	TCG	S	TAG	*stop*	TGG	W
CTT	L	CCT	P	CAT	H	CGT	R
CTC	L	CCC	P	CAC	H	CGC	R
CTA	L	CCA	P	CAA	Q	CGA	R
CTG	L	CCG	P	CAG	Q	CGG	R
ATT	I	ACT	T	AAT	N	AGT	S
ATC	I	ACC	T	AAC	N	AGC	S
ATA	I	ACA	T	AAA	K	AGA	R
ATG	M/ *start*	ACG	T	AAG	K	AGG	R
GTT	V	GCT	A	GAT	D	GGT	G
GTC	V	GCC	A	GAC	D	GGC	G
GTA	V	GCA	A	GAA	E	GGA	G
GTG	V	GCG	A	GAG	E	GGG	G

Поиск генов, если известен белок: просто

```
aaacgccctggccttctatagccactattatcagccgctgccgttggtactgcgcggata

tggtgcgggcaatgacgttacagctgccggtgtctttgctgatctgctacgtaccctctc

                          M  V  K  V  Y  A  P  A  S  S  A  N  M
atggaagttaggagtctgacatggttaaagtttatgccccggcttccagtgccaatatga

S  V  G  F  D  V  L  G  A  A  V  T  P  V  D  G  A  L  L  G
gcgtcgggtttgatgtgctcggggcggcggtgacacctgttgatggtgcattgctcggag

D  V  V  T  V  E  A  A  E  T  F  S  L  N  N  L  G  R  F  G
atgtagtcacggttgaggcggcagagacattcagtctcaacaacctcggacgctttgggt

A  D  K  L  P  S  E  P  R  E  N  V  Y  Q  C  W  E  R  F  C
ccgataagctgccgtcagagccacgggaaaatgtttatcagtgctgggagcgttttgcc

Q  E  L  G  K  Q  I  P  V  A  M  T  L  E  K  N  M  P  I  G
aggagcttggcaagcaaattccagtggcgatgactctggaaaagaatatgccgatcggtt

F  V  H  I  C  R  L  D  T  A  G  A  R  V  L  E  N
ttgttcatatttgccggctggatacggcgggcgcacgagtactggaaaactaaatgaaac

tctacaatctgaaagatcacaatgagcaggtcagctttgcgcaagccgtaacccaggggt

tgggcaaaaatcaggggctgtttttcccgcacgacctgccggaattcagcctgactgaaa
```

TTT	F	TCT	S	TAT	Y	TGT	C
TTC	F	TCC	S	TAC	Y	TGC	C
TTA	L	TCA	S	TAA	*stop*	TGA	*stop*
TTG	L	TCG	S	TAG	*stop*	TGG	W
CTT	L	CCT	P	CAT	H	CGT	R
CTC	L	CCC	P	CAC	H	CGC	R
CTA	L	CCA	P	CAA	Q	CGA	R
CTG	L	CCG	P	CAG	Q	CGG	R
ATT	I	ACT	T	AAT	N	AGT	S
ATC	I	ACC	T	AAC	N	AGC	S
ATA	I	ACA	T	AAA	K	AGA	R
ATG	M/*start*	ACG	T	AAG	K	AGG	R
GTT	V	GCT	A	GAT	D	GGT	G
GTC	V	GCC	A	GAC	D	GGC	G
GTA	V	GCA	A	GAA	E	GGA	G
GTG	V	GCG	A	GAG	E	GGG	G

… или родственный белок: тоже просто

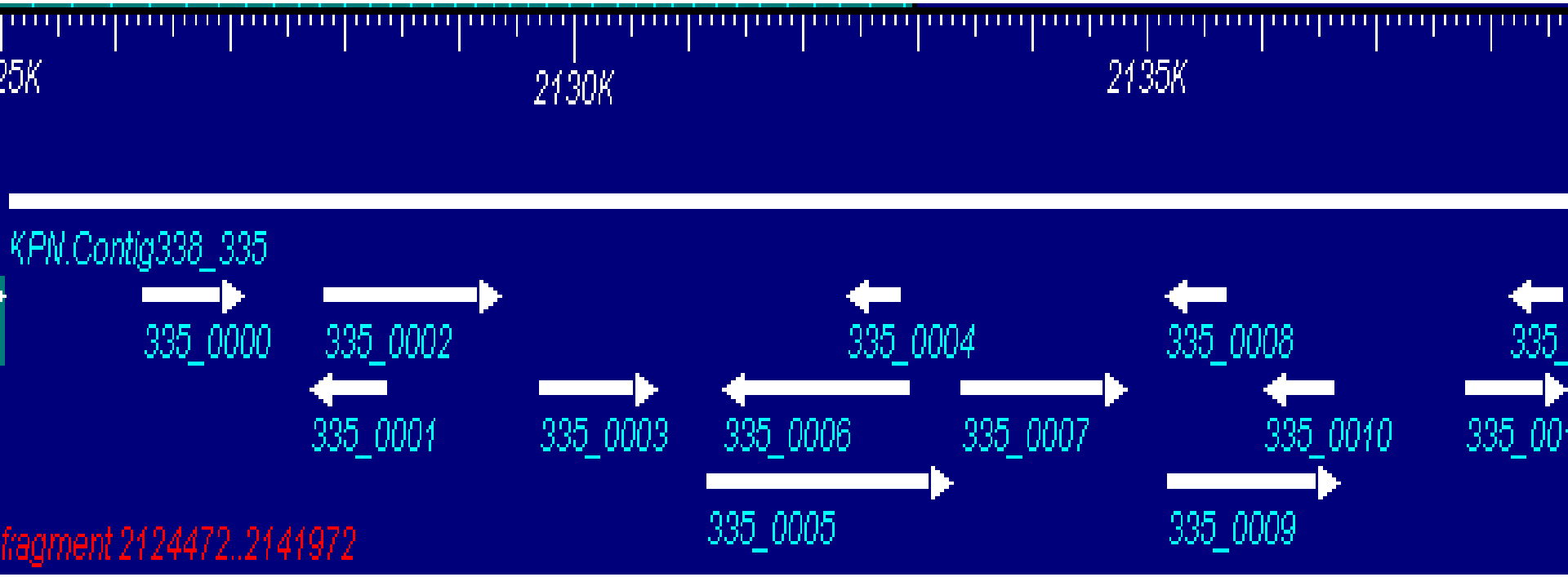
```
aaacgccctggccttctatagccactattatcagccgctgccgttggtactgcgcggata

tggtgcgggcaatgacgttacagctgccggtgtctttgctgatctgctacgtaccctctc

                          M  V  V  V  Y  A  P  A  S  I  G  N  V
atggaagttaggagtctgacatggttaaagtttatgccccggcttccagtgccaatatga

S  V  G  F  D  V  L  G  A  A  V  S  P  I  D  G  S  L  L  G
gcgtcgggtttgatgtgctcggggcggcggtgacacctgttgatggtgcattgctcggag

D  R  V  E  V  A  A  G  E  Q  P  F  T  L  K  C  V  G  D  F
atgtagtcacggttgaggcggcagagaca---ttcagtctcaacaacctcggacgctttg

V  A  K  L  P  V  E  Q  E  E  N  V  Y  H  C  W  L  V  F  A
ccgataagctgccgtcagagccacgggaaaatgtttatcagtgctgggagcgttttgcc

R  E  L  D  K  K  V  P  V  S  M  T  L  E  K  N  M  P  I  G
aggagcttggcaagcaaattccagtggcgatgactctggaaaagaatatgccgatcggtt

F  V  H  V  C  R  L  D  S  T  G  S  K  V  L  E  N
ttgttcatatttgccggctggatacggcgggcgcacgagtactggaaaactaaatgaaac

tctacaatctgaaagatcacaatgagcaggtcagctttgcgcaagccgtaacccaggggt

tgggcaaaaatcaggggctgtttttcccgcacgacctgccggaattcagcctgactgaaa
```

Генетический код: стоп-кодоны

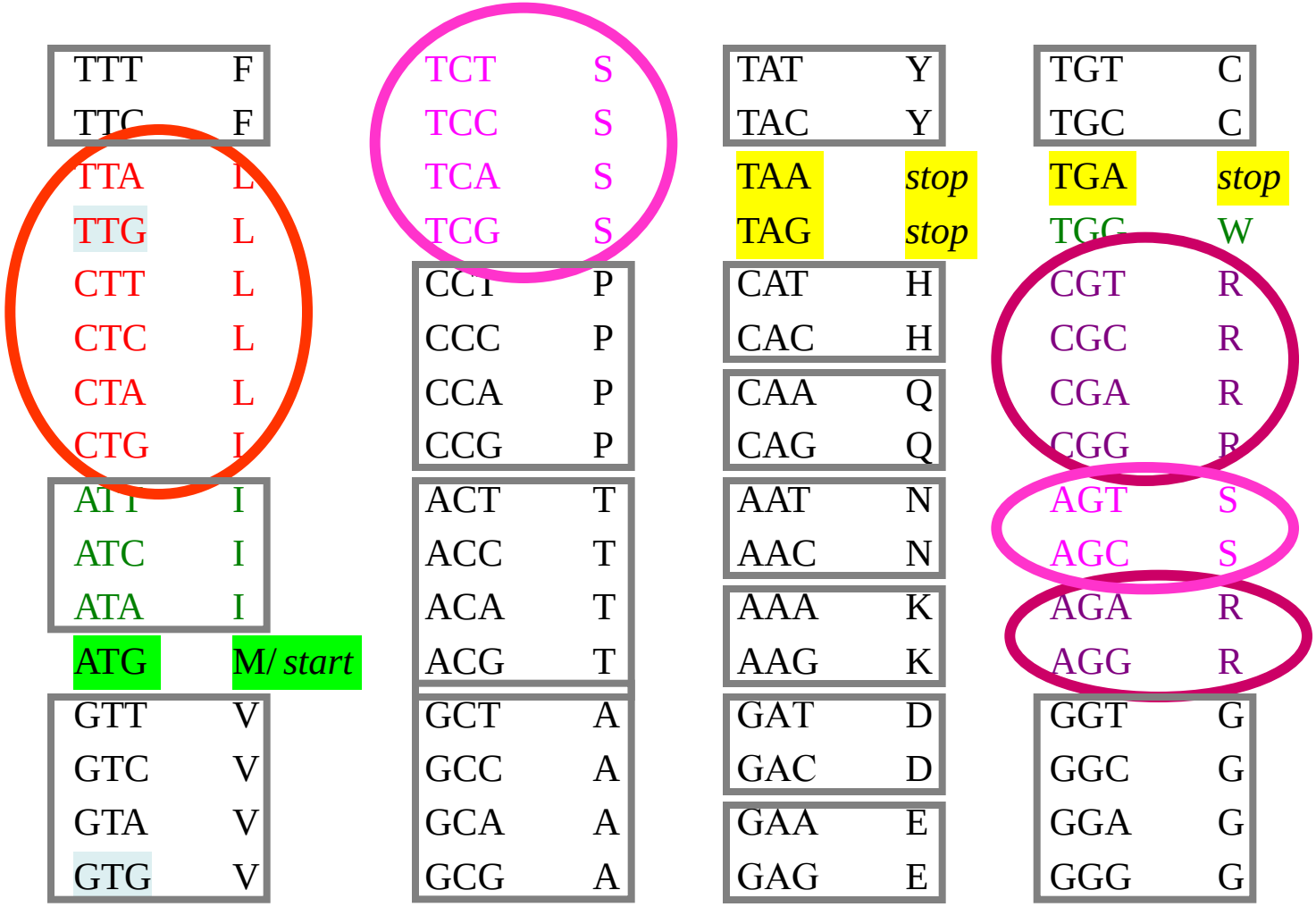
TTT	F	TCT	S	TAT	Y	TGT	C
TTC	F	TCC	S	TAC	Y	TGC	C
TTA	L	TCA	S	TAA	*stop*	TGA	*stop*
TTG	L	TCG	S	TAG	*stop*	TGG	W
CTT	L	CCT	P	CAT	H	CGT	R
CTC	L	CCC	P	CAC	H	CGC	R
CTA	L	CCA	P	CAA	Q	CGA	R
CTG	L	CCG	P	CAG	Q	CGG	R
ATT	I	ACT	T	AAT	N	AGT	S
ATC	I	ACC	T	AAC	N	AGC	S
ATA	I	ACA	T	AAA	K	AGA	R
ATG	M/ *start*	ACG	T	AAG	K	AGG	R
GTT	V	GCT	A	GAT	D	GGT	G
GTC	V	GCC	A	GAC	D	GGC	G
GTA	V	GCA	A	GAA	E	GGA	G
GTG	V	GCG	A	GAG	E	GGG	G

Открытые рамки считывания

Ген должен располагаться внутри области от стоп-кодона до следующего стоп-кодона (в той же фазе)



Генетический код: синонимы

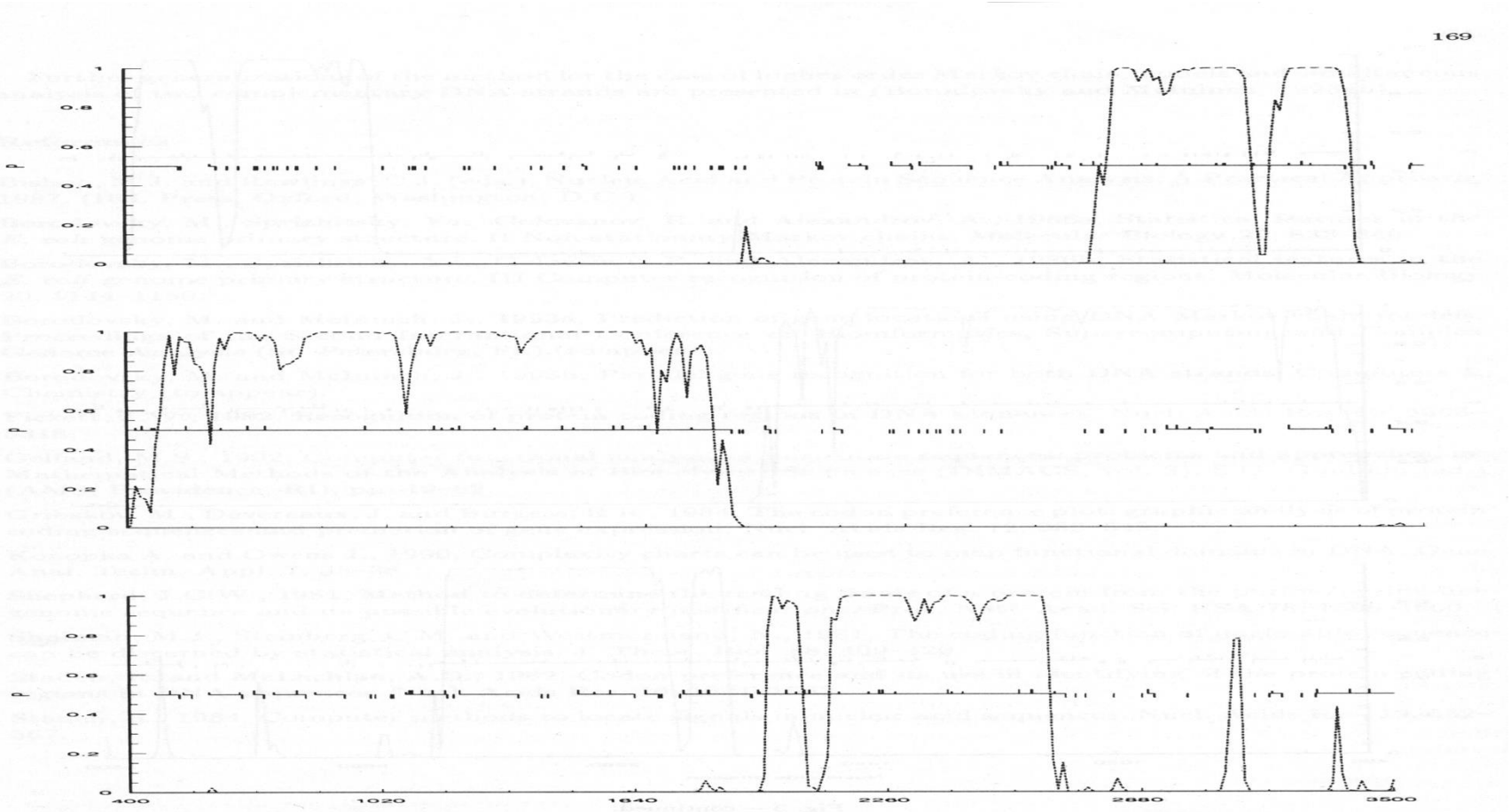


Codon usage
(статистика употребления кодонов)

- частоты кодонов отличаются от частот триплетов в некодирующих областях
  - различия в частотах аминокислот в белках
  - различия в частотах синонимичных кодонов
- частоты синонимичных кодонов
  - специфичны для генома
  - коррелируют с концентрациями тРНК

Статистические особенности

Можно ввести функцию, которая измеряет частоты кодонов в кодирующих и некодирующих областях (скользящее окно, три рамки считывания)



GenMark

Генетический код: старт-кодоны

TTT	F	TCT	S	TAT	Y	TGT	C
TTC	F	TCC	S	TAC	Y	TGC	C
TTA	L	TCA	S	TAA	*stop*	TGA	*stop*
TTG	L	TCG	S	TAG	*stop*	TGG	W
CTT	L	CCT	P	CAT	H	CGT	R
CTC	L	CCC	P	CAC	H	CGC	R
CTA	L	CCA	P	CAA	Q	CGA	R
CTG	L	CCG	P	CAG	Q	CGG	R
ATT	I	ACT	T	AAT	N	AGT	S
ATC	I	ACC	T	AAC	N	AGC	S
ATA	I	ACA	T	AAA	K	AGA	R
ATG	M/ *start*	ACG	T	AAG	K	AGG	R
GTT	V	GCT	A	GAT	D	GGT	G
GTC	V	GCC	A	GAC	D	GGC	G
GTA	V	GCA	A	GAA	E	GGA	G
GTG	V	GCG	A	GAG	E	GGG	G

Начала генов *Bacillus subtilis*

dnaN	ACATTATCCGTTAGGAGGATAAAA**ATG**
gyrA	GTGATACTTCAGGGAGGTTTTTTA**ATG**
serS	TCAATAAAAAAAGGAGTGTTTCGC**ATG**
bofA	CAAGCGAAGGAGATGAGAAGATTC**ATG**
csfB	GCTAACTGTACGGAGGTGGAGAAG**ATG**
xpaC	ATAGACACAGGAGTCGATTATCTC**ATG**
metS	ACATTCTGATTAGGAGGTTTCAAG**ATG**
gcaD	AAAAGGGATATTGGAGGCCAATAA**ATG**
spoVC	TATGTGACTAAGGGAGGATTCGCC**ATG**
ftsH	GCTTACTGTGGGAGGAGGTAAGGA**ATG**
pabB	AAAGAAAATAGAGGAATGATACAA**ATG**
rplJ	CAAGAATCTACAGGAGGTGTAACC**ATG**
tufA	AAAGCTCTTAAGGAGGATTTTAGA**ATG**
rpsJ	TGTAGGCGAAAAGGAGGGAAAATA**ATG**
rpoA	CGTTTTGAAGGAGGGTTTTAAGTA**ATG**
rplM	AGATCATTTAGGAGGGGAAATTCA**ATG**

Участок связывания рибосом

dnaN	ACATTATCCGTTAGGAGGATAAAAATG
gyrA	GTGATACTTCAGGGAGGTTTTTTAATG
serS	TCAATAAAAAAAGGAGTGTTTCGCATG
bofA	CAAGCGAAGGAGATGAGAAGATTCATG
csfB	GCTAACTGTACGGAGGTGGAGAAGATG
xpaC	ATAGACACAGGAGTCGATTATCTCATG
metS	ACATTCTGATTAGGAGGTTTCAAGATG
gcaD	AAAAGGGATATTGGAGGCCAATAAATG
spoVC	TATGTGACTAAGGGAGGATTCGCCATG
ftsH	GCTTACTGTGGGAGGAGGTAAGGAATG
pabB	AAAGAAAATAGAGGAATGATACAAATG
rplJ	CAAGAATCTACAGGAGGTGTAACCATG
tufA	AAAGCTCTTAAGGAGGATTTTAGAATG
rpsJ	TGTAGGCGAAAAGGAGGGAAAATAATG
rpoA	CGTTTTGAAGGAGGGTTTTAAGTAATG
rplM	AGATCATTTAGGAGGGGAAATTCAATG

Сравнение генов в родственных геномах

Гены консервативнее, чем межгенные области (точнее, особенности эволюции другие)

pdxB

Периодичность нуклеотидных замен в белок-кодирующих областях

```
Sty   GTACGACAGTC--GTGAAAATTCTCGTTGATGAAAATATGCCTTACGCCCGC
Stm   GTACGACAGTC--GTGAAAATTCTCGTTGATGAAAATATGCCTTACGCCCGC
Sen   GTACGACAGTC--GTGAAAATTCTCGTTGATGAAAATATGCCTTACGCCCGC
Eco   GTAACACAAAC--GTGAAAATCCTTGTTGATGAAAATATGCCTTATGCCCGC
Kpn   GTACTGAACT---GTGAAAATCCTCGTTGATGAAAATATGCCCTATGCCCGT
Ype   GTCGGAGAGTAAAGTGAAAATTCTGGTTGATGAAAATATGCCGTACGCTGAG
        *     *      ******** ** ********************* ** **
                     123123123123123123123123123123123123
```

5 синонимичных замен, 1 замена аминокислоты

Размер вставок кратен 3 (иначе случится сдвиг рамки)

rbsD в энтеробактериях

```
Sty AGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Sen AGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Stm GGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Eco AGGATTAAACTGTGGGTCAGCGAAACGTTTCGCTGATGGAGAA-AAAAATGAAAAAAGGC
Ype TTTTCTAAACTCCTTGTTAGCGAAACGTTTCGCTCTTGGAGTA-GATCATGAAAAAAGGT
          ** ***       ****************  ***** *  *  ***** *****

Sty ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Sen ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Stm ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Eco ACCGTTCTTAATTCTGATATTTCATCGGTGATCTCCCGTCTGGGACATACCGATACGCTG
Ype GTATTACTGAACGCTGATATTTCCGCGGTTATCTCCCGTCTGGGCCATACCGATCAGATT
          * ** **  **** ** **  **** ** ********** ***** ***    *
```

rbsD в энтеробактериях: ответ

```
Sty AGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Sen AGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Stm GGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Eco AGGATTAAACTGTGGGTCAGCGAAACGTTTCGCTGATGGAGAA-AAAAATGAAAAAAGGC
Ype TTTTCTAAACTCCTTGTTAGCGAAACGTTTCGCTCTTGGAGTA-GATCATGAAAAAAGGT
          ** ***        ****************  ***** *  *  ***** *****

Sty ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Sen ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Stm ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Eco ACCGTTCTTAATTCTGATATTTCATCGGTGATCTCCCGTCTGGGACATACCGATACGCTG
Ype GTATTACTGAACGCTGATATTTCCGCGGTTATCTCCCGTCTGGGCCATACCGATCAGATT
          * ** **  **** ** **  **** ** ********** ***** ***    *
```

Существующая аннотация (была) неправильна

```
Sty AGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Sen AGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Stm GGGGTTACACTGCGGC-CAGCGAAACGTTTCGCTAGTGGAGCAGAAAAATGAAGAAAGGC
Eco AGGATTAAACTGTGGGTCAGCGAAACGTTTCGCTGATGGAGAA-AAAAATGAAAAAAGGC
Ype TTTTCTAAACTCCTTGTTAGCGAAACGTTTCGCTCTTGGAGTA-GATCATGAAAAAAGGT
          ** ***        ***************  ***** *   *  ***** *****
```

```
Sty ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Sen ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Stm ACCGTACTCAACTCTGAAATCTCGTCGGTCATTTCCCGTCTGGGGCATACTGATACTCTG
Eco ACCGTTCTTAATTCTGATATTTCATCGGTGATCTCCCGTCTGGGACATACCGATACGCTG
Ype GTATTACTGAACGCTGATATTTCCGCGGTTATCTCCCGTCTGGGCCATACCGATCAGATT
          * ** **  **** ** **  **** ** ********** ***** ***    *
```

Мораль

- **Комплексный подход:** использование многих разнородных соображений, каждое из которых по отдельности – слабое

- **Сравнительный подход:** одновременный анализ множества геномов (находящихся на различных эволюционных расстояниях друг от друга)

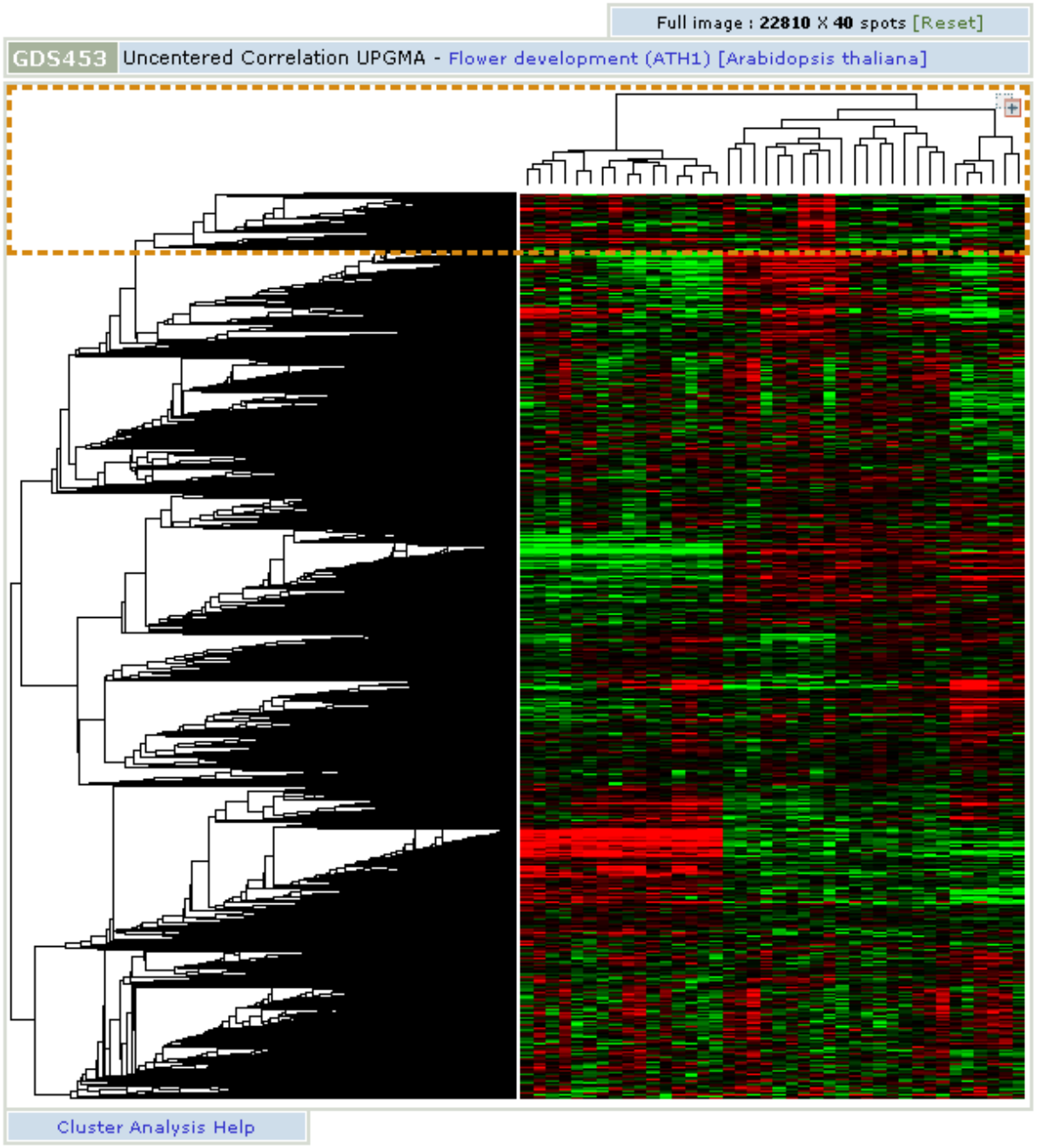
Не только тексты

Можно использовать данные, которые порождаются другими типами массовых экспериментов

- Уровень экспрессии
  - Концентрации мРНК
  - Концентрации белков
  - Время жизни мРНК и белков
- Взаимодействия
  - Белок-ДНКовые
  - Белок-белковые
- Структура генома
  - Метилирование ДНК
  - Положение и модификация нуклеосом
  - Пространственная структура
- Функционально-генетические
  - Летальность и фенотип мутаций
  - Синтетические летали

Экспрессия генов – 1. Развитие цветка резуховидки Таля

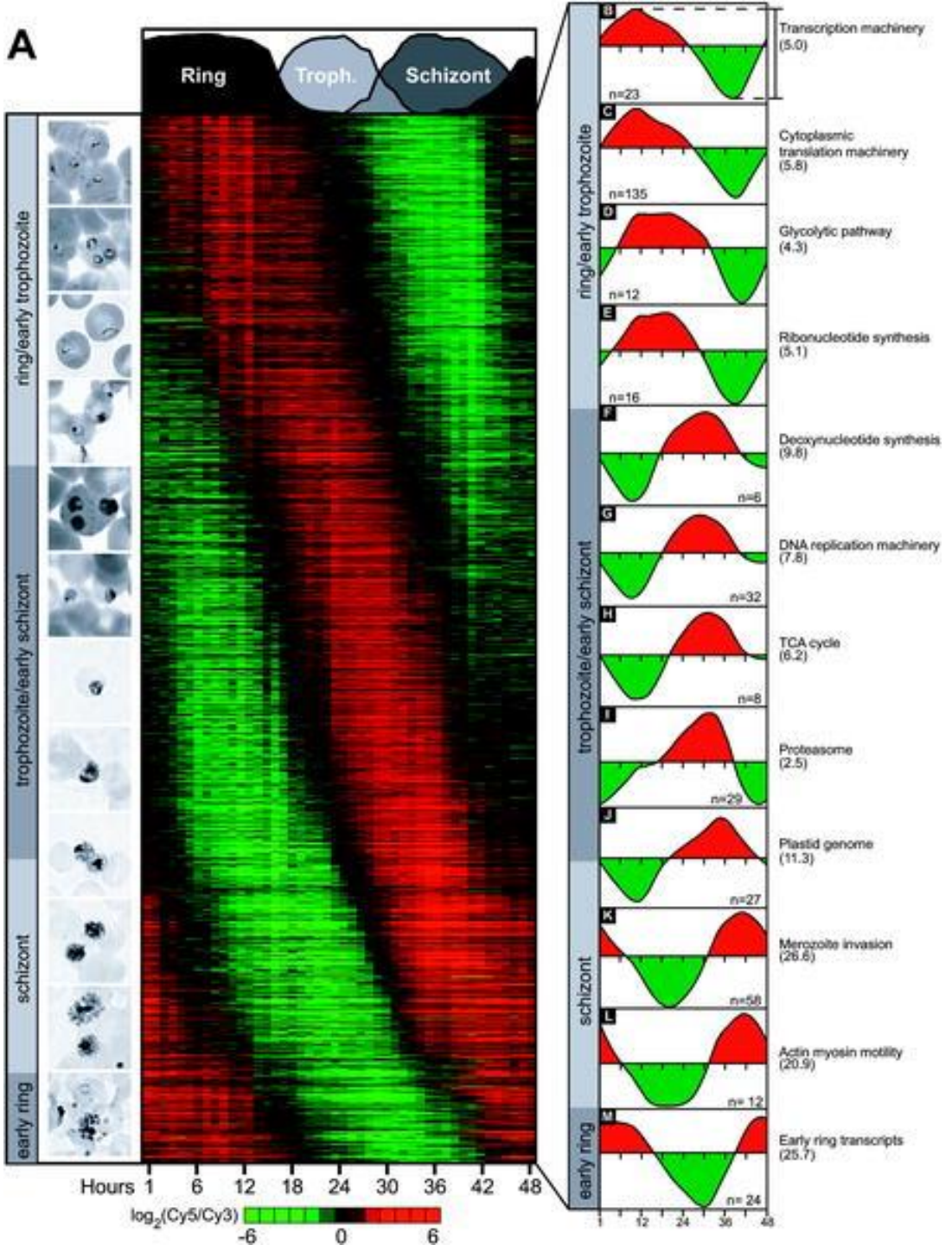
двойная кластеризация – на генах и на условиях



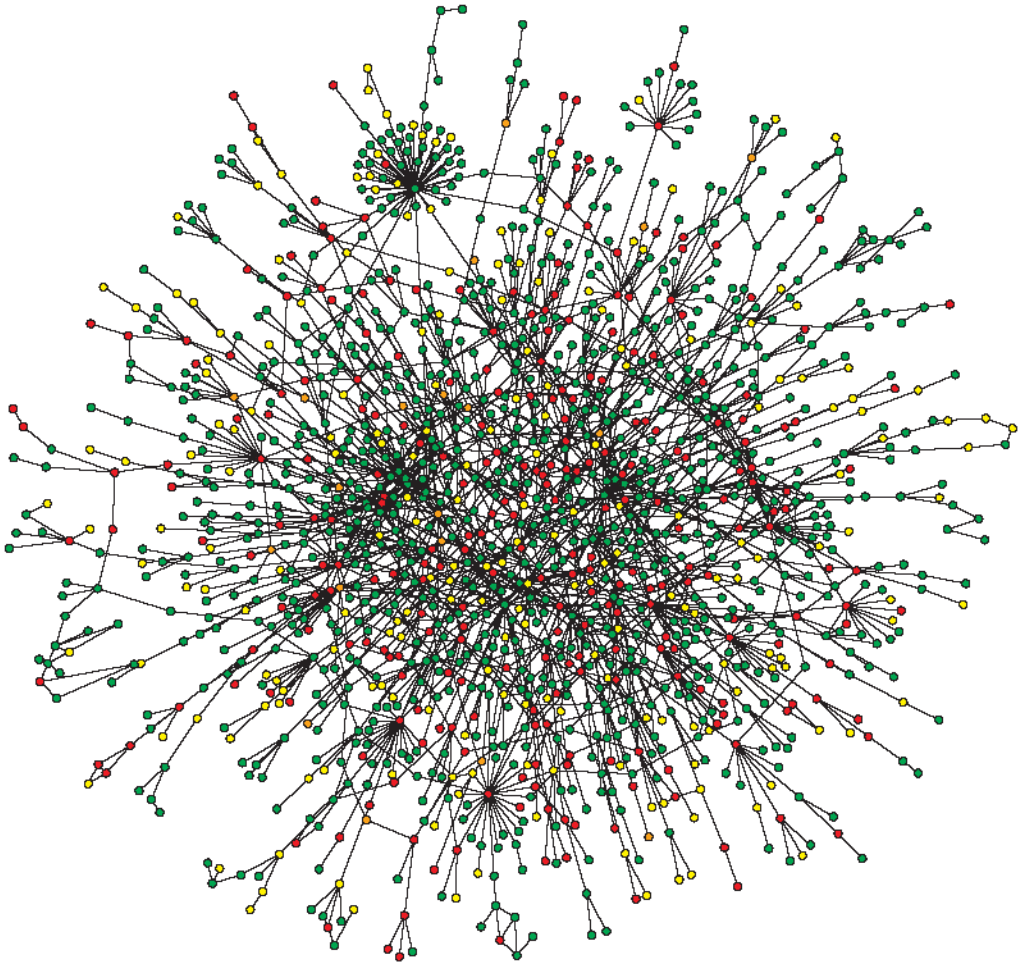
Экспрессия генов – 2. Цикл развития малярийного плазмодия

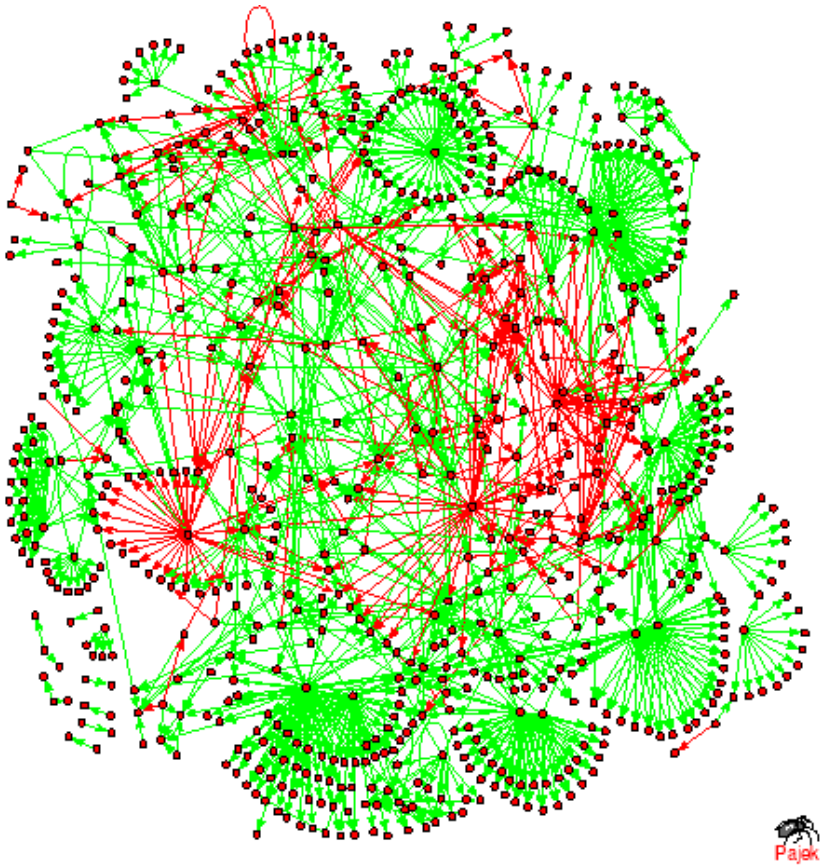
The Transcriptome
of the Intraerythrocytic Developmental Cycle
of *Plasmodium falciparum*

Zbynek Bozdech[1☯], Manuel Llinás[1☯], Brian Lee Pulliam[1], Edith D. Wong[1], Jingchun Zhu[2], Joseph L. DeRisi[1*]

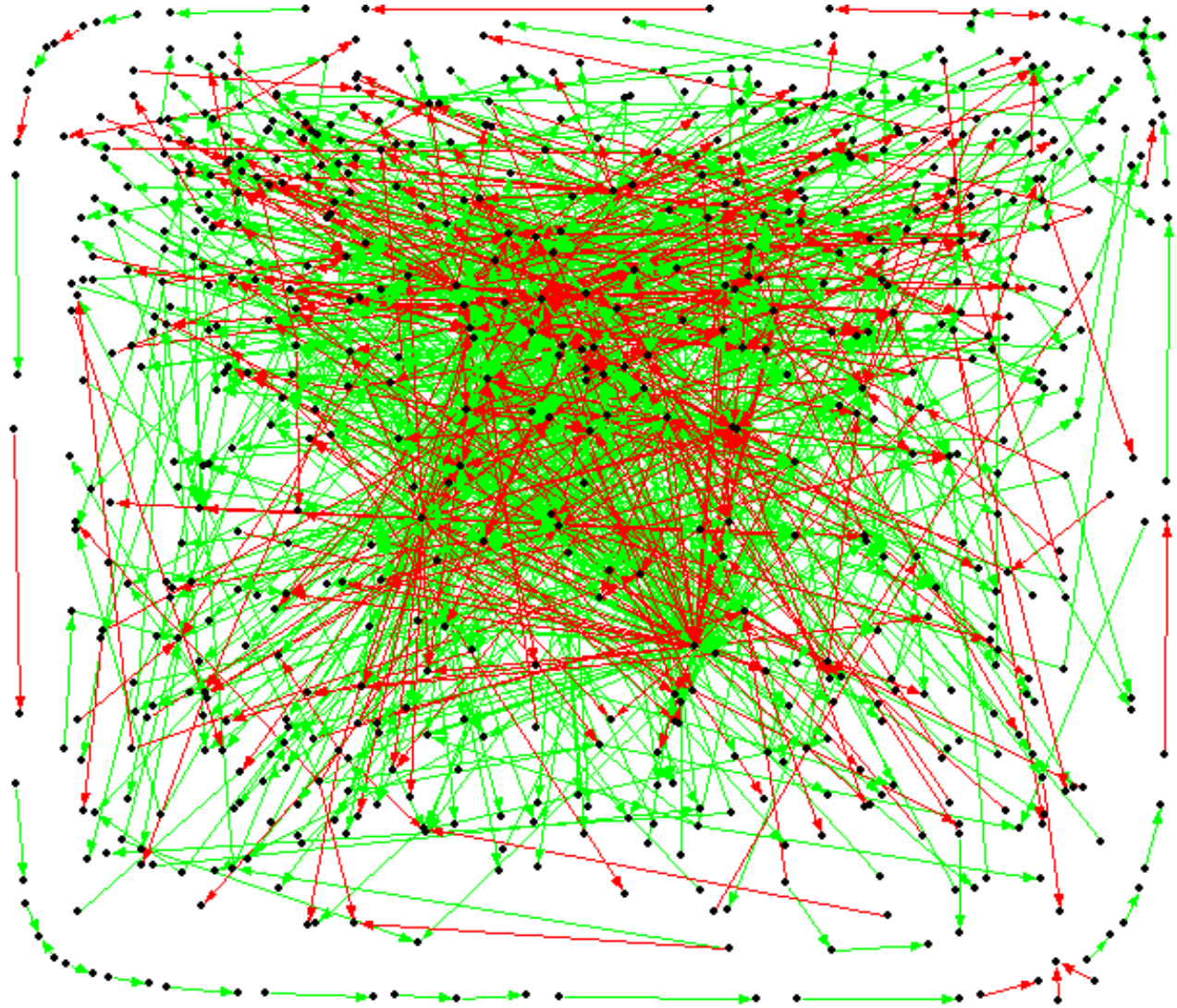


Белок-белковые (структурные, сигнальные и др.) **и белок-ДНКовые** (регуляторные) **взаимодействия в дрожжах**





Регуляция транскрипции у человека

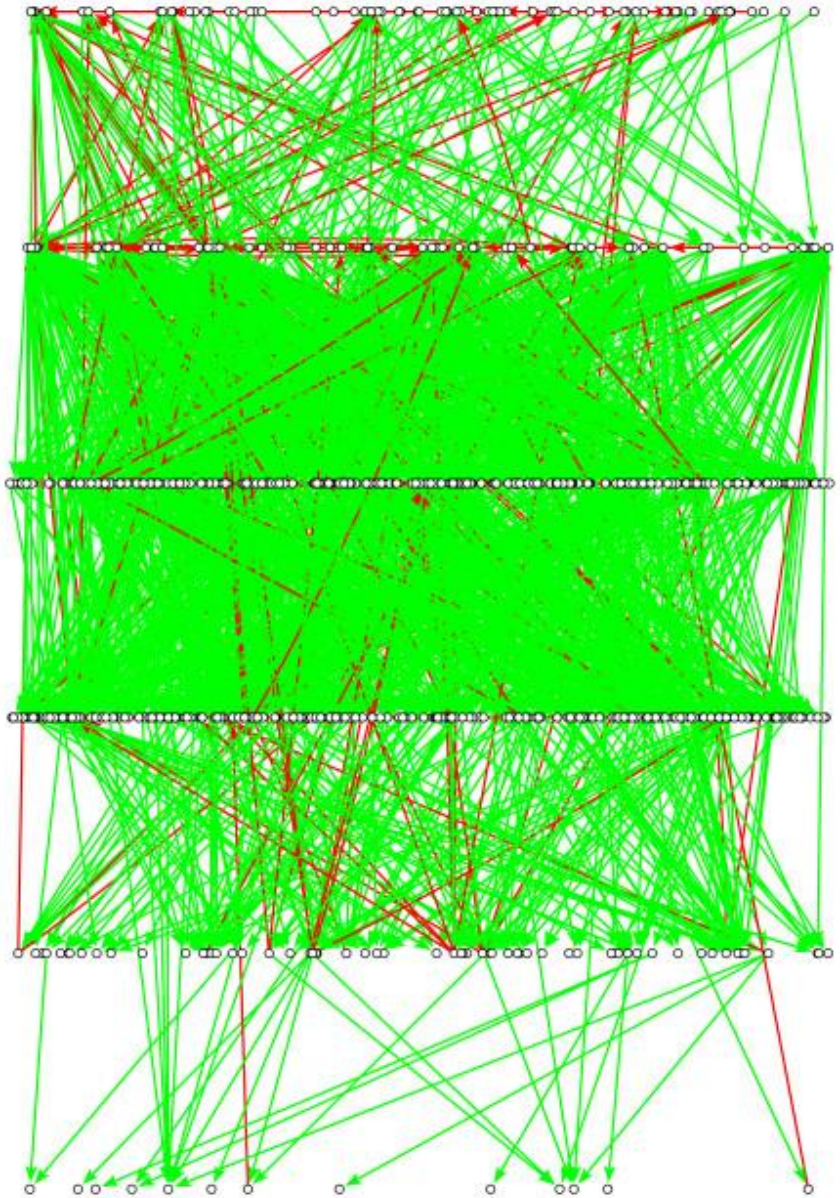


- 1449 взаимодействий между 689 генами
- Отношение «активаторы : репрессоры» = 3:1
- До 95 регулируемых генов, до 45 регуляторов.

Маслов

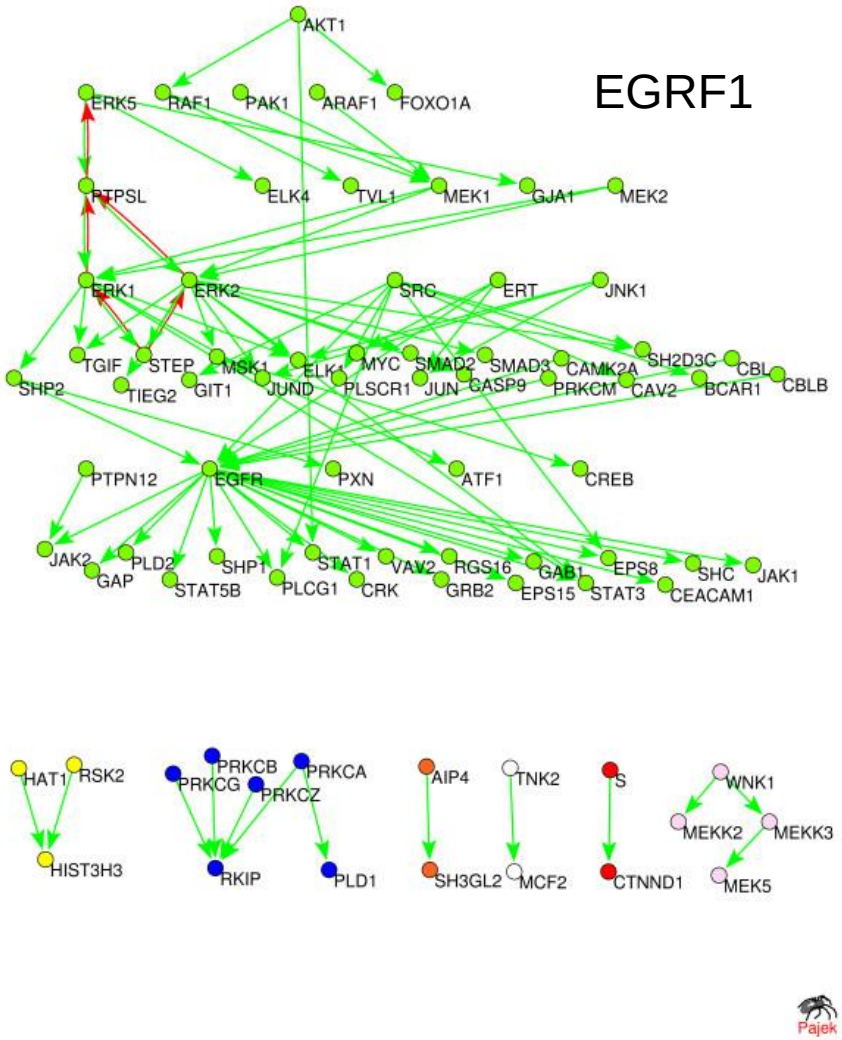
Иерархия:
732 белков
(71 рецепторов),
1671 взаимодействий
(фосфорилирование,
дефосфорилировние,
гидролиз etc)

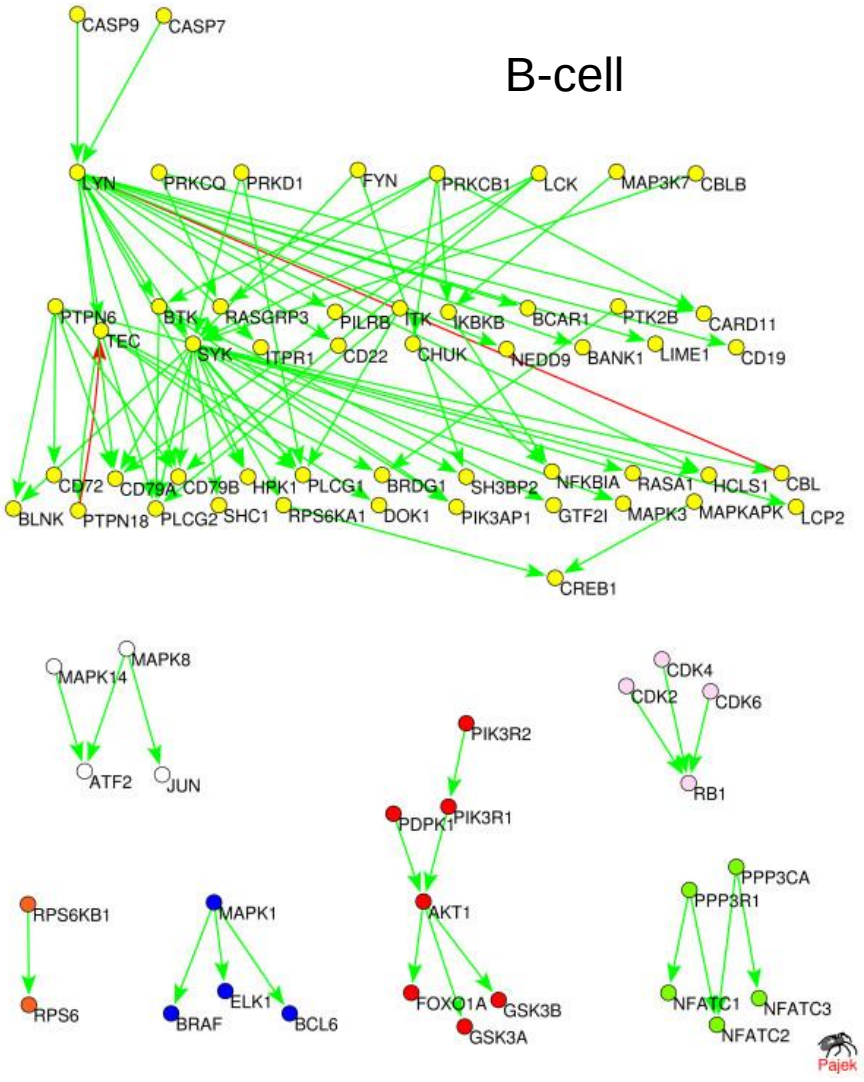
208 анти-
иерархических ребер



Маслов-Исполатов

Подсети

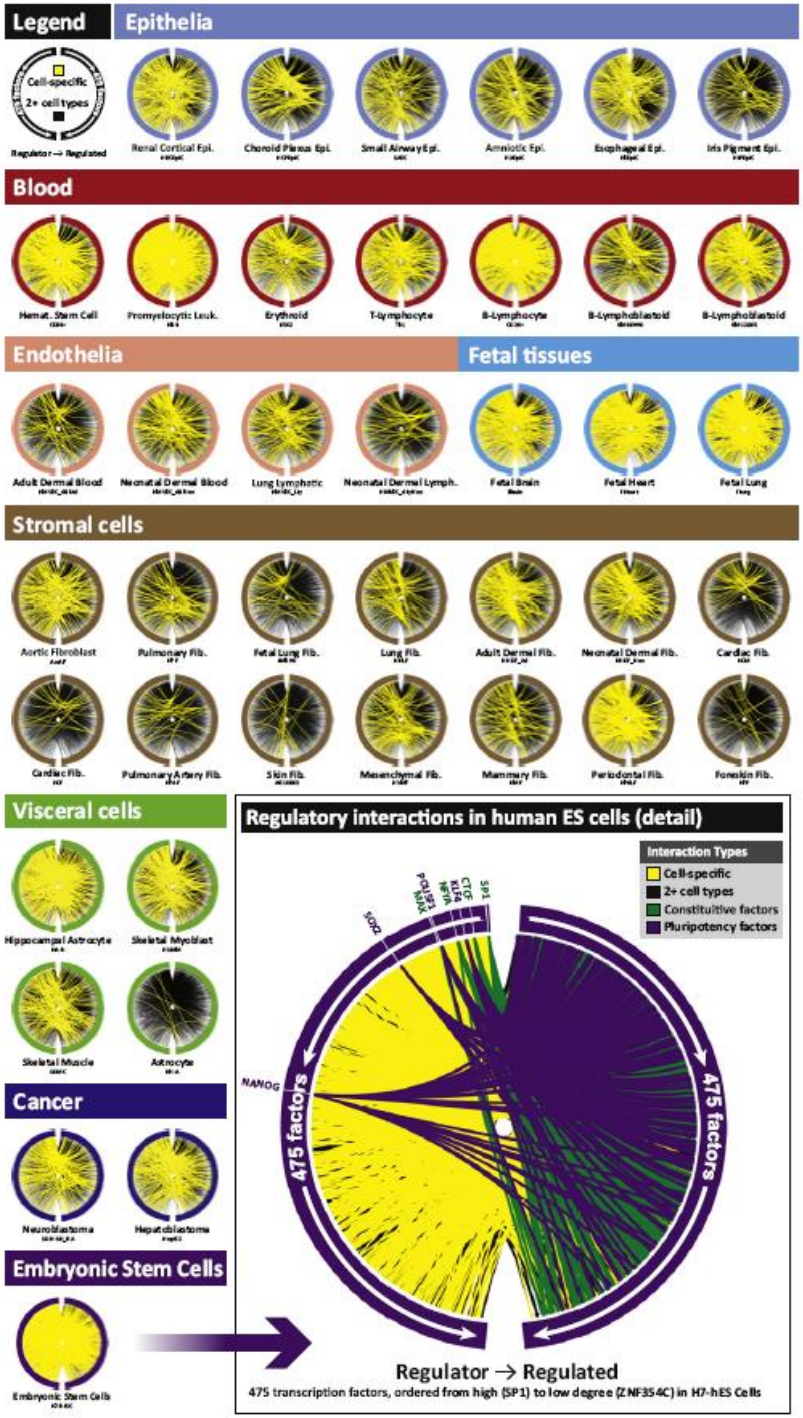




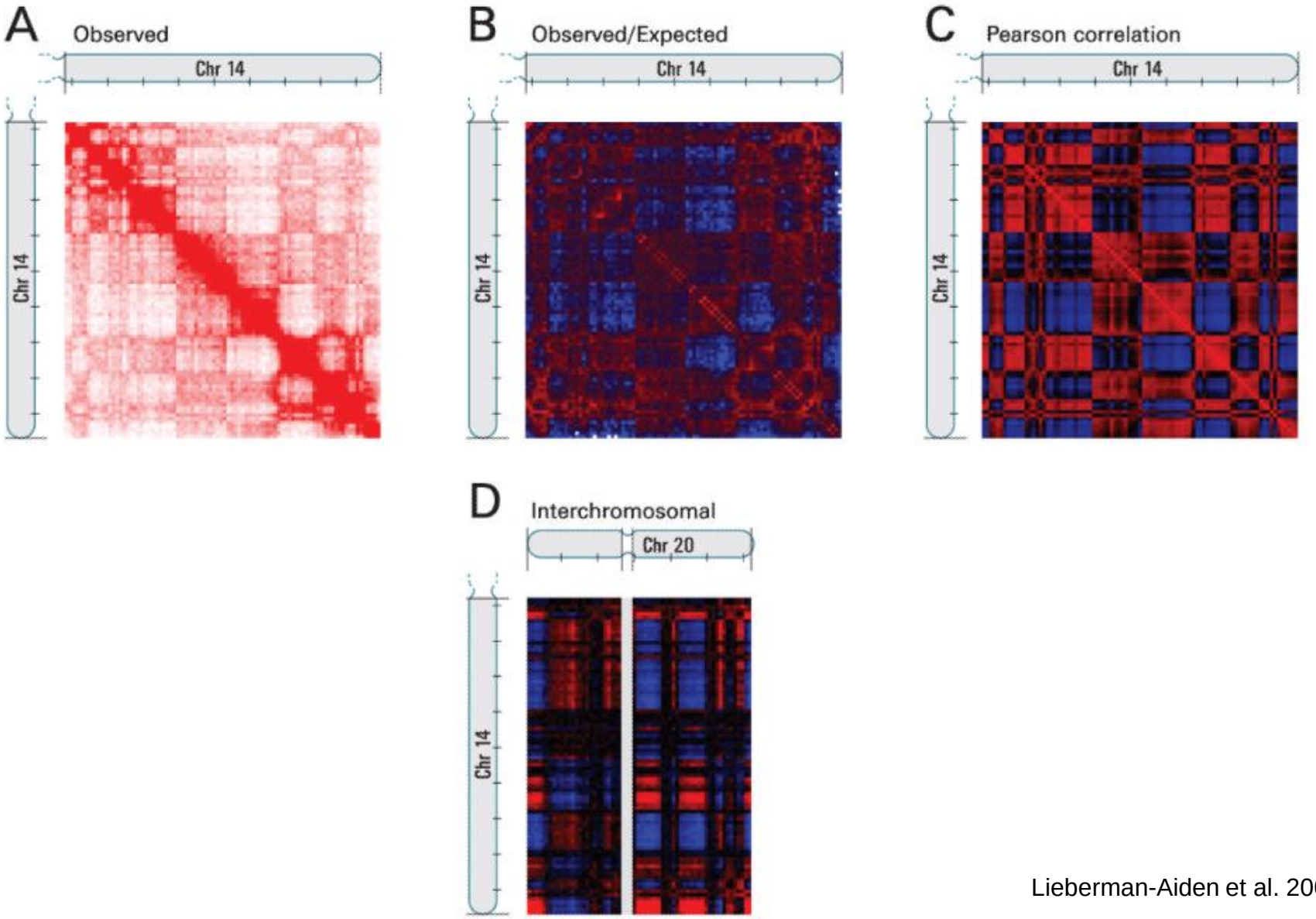
Анти-иерархические связи (красные) – дефосфорилирование, ибиквитинилирование – перезарядка сети

Маслов-Исполатов

Динамика: активность транскрипционных взаимодействий в клеточных линиях

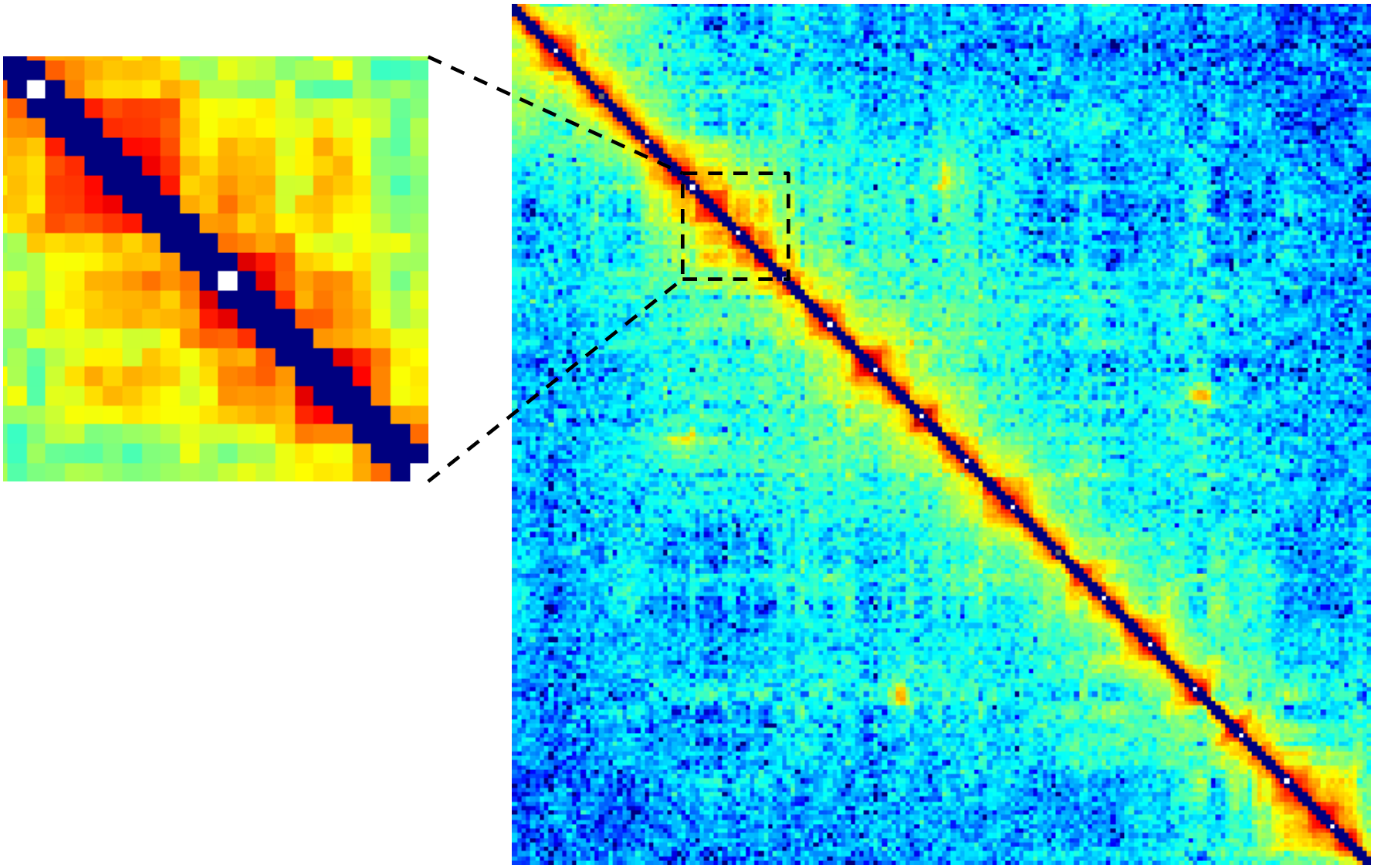


Пространственная структура ДНК

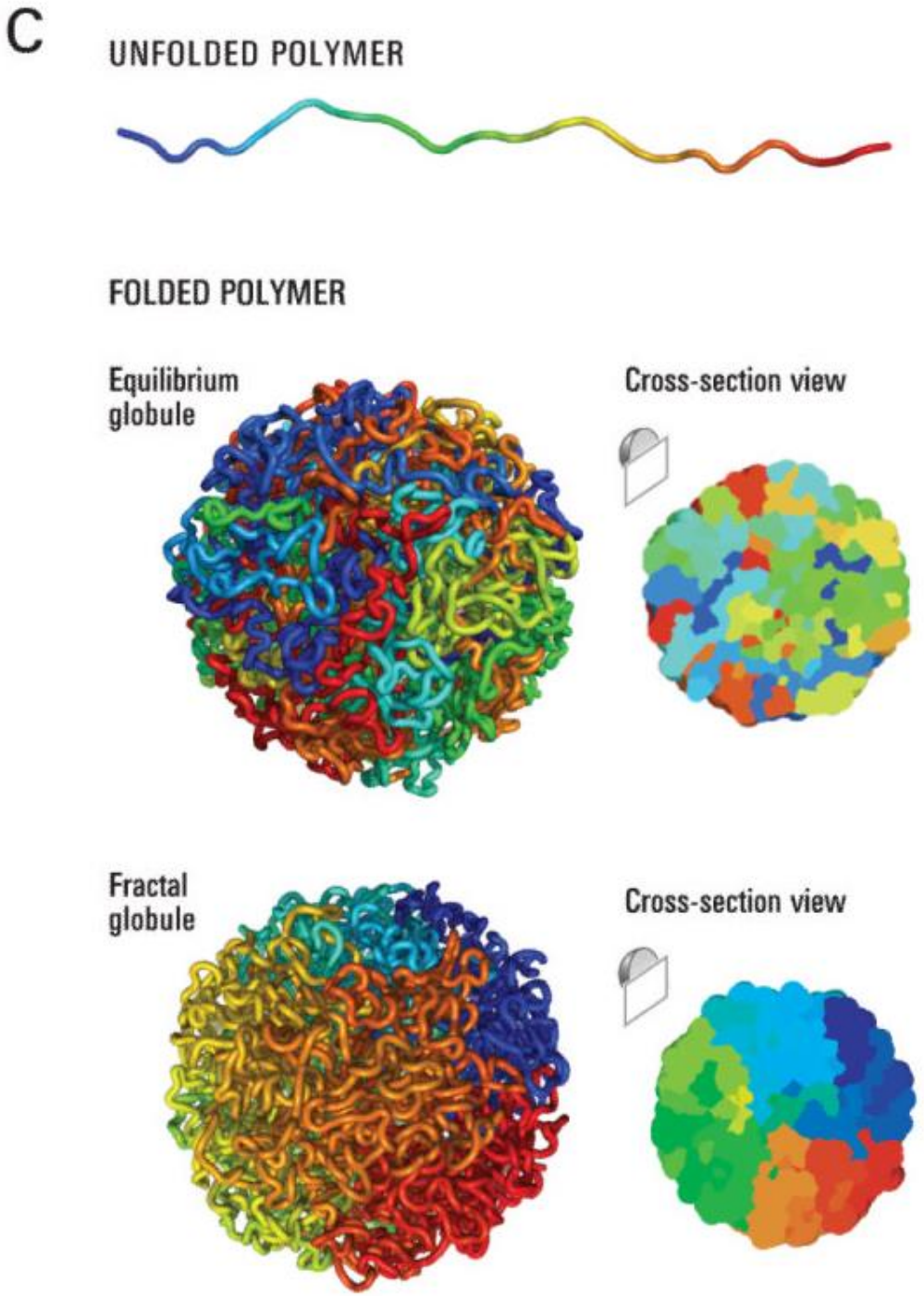


Lieberman-Aiden et al. 2009

Топологические домены

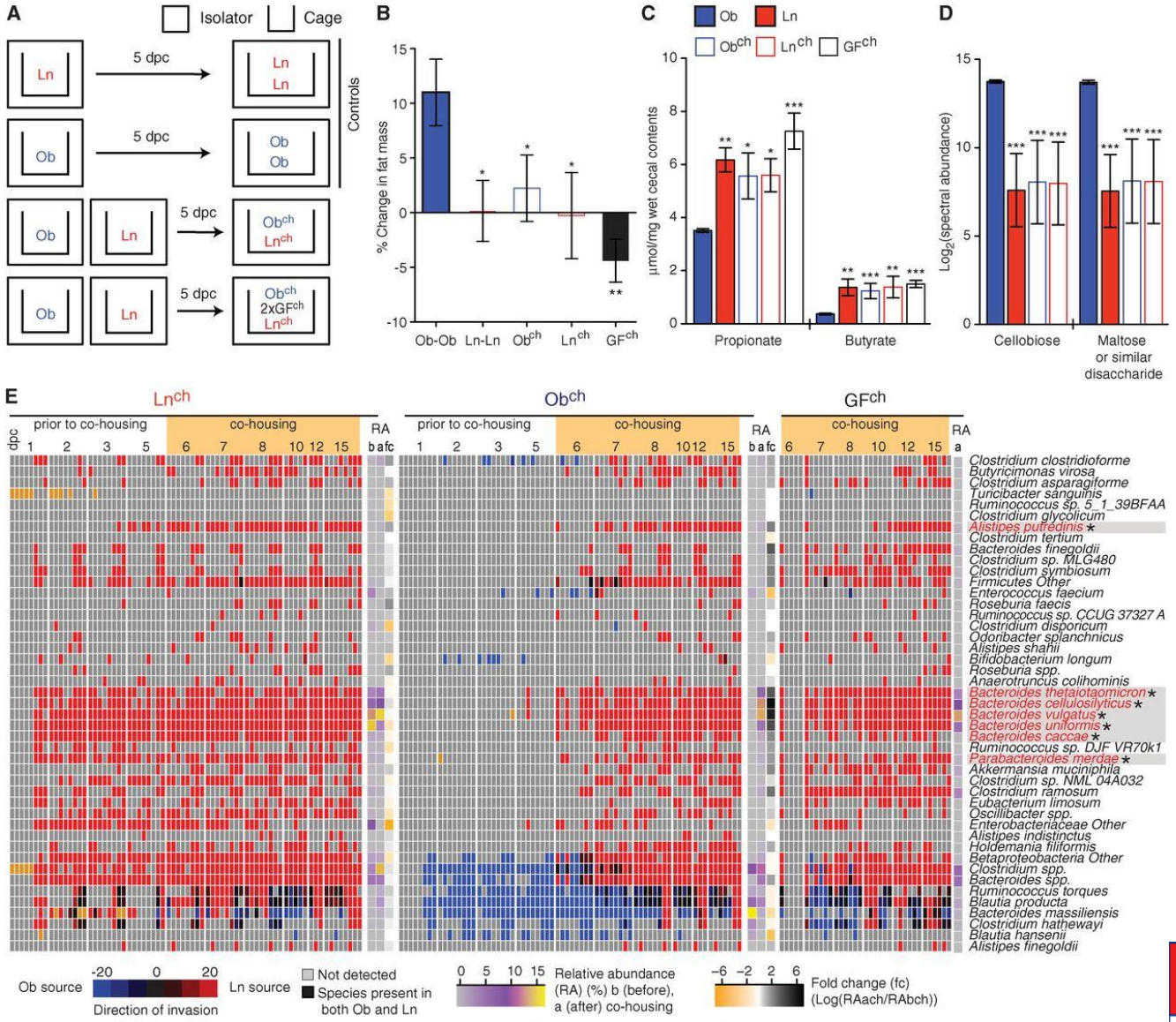


Глобальная модель: фрактальная глобула



Lieberman-Aiden et al. 2009

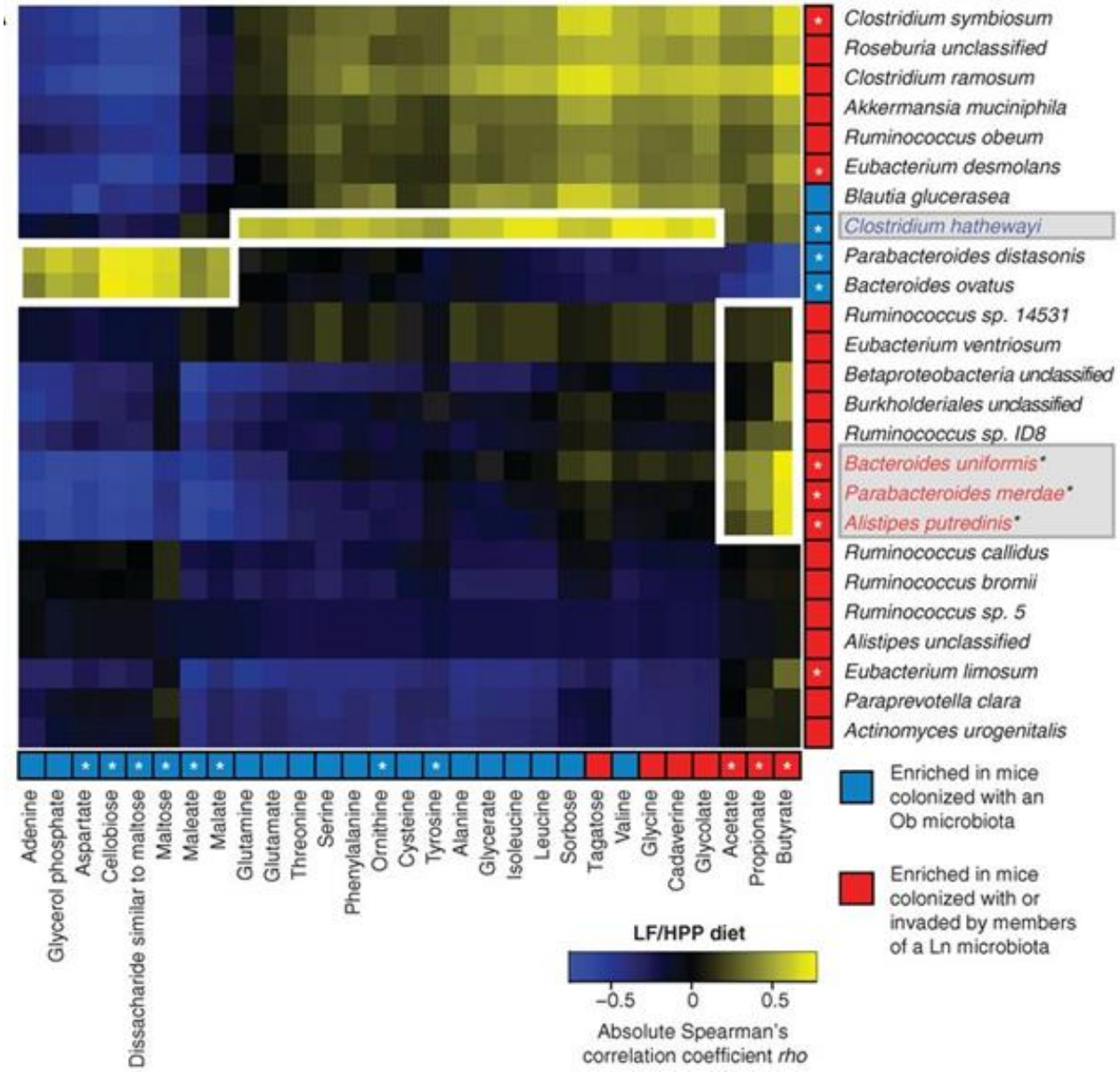
Метагеномика: кто в ком живёт



Science
AAAS

V K Ridaura et al. Science 2013;341:1241214

Метагеномика и метаболомика: кто что делает



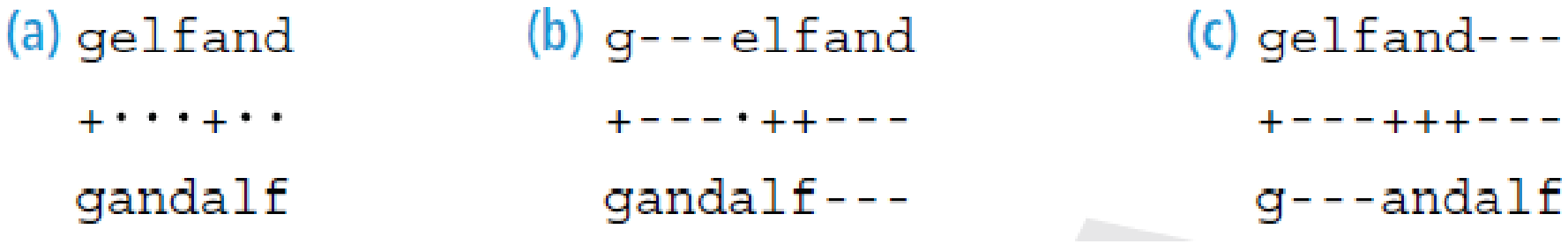
V K Ridaura et al.
Science
2013;341:1241214

Science
AAAS

Динамическое программирование

(одна алгоритмическая затычка во много биологических бочек)

Выравнивание



Три из многих выравниваний двух последовательностей.

+ совпадение; **•** несовпадение, **–** пробел

a) 2 совпадения, 5 несовпадений
b) 3 совпадения, 1 несовпадение, 2 вставки длины 3 (6 пробелов)
c) 4 совпадения, 2 вставки длины 3 (6 пробелов)

Выравнивание двух белковых последовательностей

BRCA1 Xenopus laevis vs Pan troglodytus

```
fr  MtcSrMdIEgIcSVISvMQKnLECPICLELMKEPVATKCDHIFCKFCMLQLLSkKKKGtv
ch  MdlSaLrVEeVqNVINaMQKiLECPICLELIKEPVSTKCDHIFCKFCMLKLLN-QKKGps

fr  pCPLCKtEVTRRSLQEShRFklLVEgqLKIIkAFEfDSGyKFfpSqehtKglDSTiEdvl
ch  qCPLCKnDITKRSLQEStRFsqLVEelLKIIcAFQlDTGlEYanSynfaKkeNNSpEh--

fr  VKEDqSIVhckGYRNRkKgVfnrKtyEetgMlsvSkAeEqfakevtRlIpcRQK-KPKKE
ch  LKDEvSIIqsmGYRNRaKrLlqsEp-EnpsLqetSlSvQlsnlgtvRtLrtKQRiQPQKK

fr  AalIf--SNcvpDS---------sDgDLLn-kenGlRNDcSplhyekeDTqipemeEmvE
ch  SvyIelgSDsseDTvnkatycsvgDqELLqitpqGtRDEiSl------DSakkaacEfsE

fr  SDLaecEfaEsAgSNLlgfD--gpEgiPEisaeTSINaagNcDfyGrkTeqfpndHhcSf
ch  TDVtntEhhQpSnNDLnttEkratErhPEkyqgSSVSnl-HvEpcGtnThasslqHenSs

fr  kqniaDaeqnKRnQhCgnvpfapMgKSnlDeketvEtdfDNQhndSnpE----NnDPLgK
ch  llltkDrmnvEKaEfCnkseqpgLaRSqhNrwagsKetcNDRrtpSteKkvdlNaDPLcE
```

Выравниваний очень много

выравниваний двух последовательностей длины N

$$\sim (1+\sqrt{2})^{2N+1}\sqrt{N}$$

при $N = 1000$ $\quad \# \approx 10^{767}$

(**#** элементарных частиц во Вселенной $\approx 10^{80}$)

при $N = 100$ $\quad \# \approx 10^{76}$

предположим, что

a) на построение выравнивания нужна 1 операция

b) мы делаем 10^{12} операций в секунду

=> понадобится 10^{57} лет

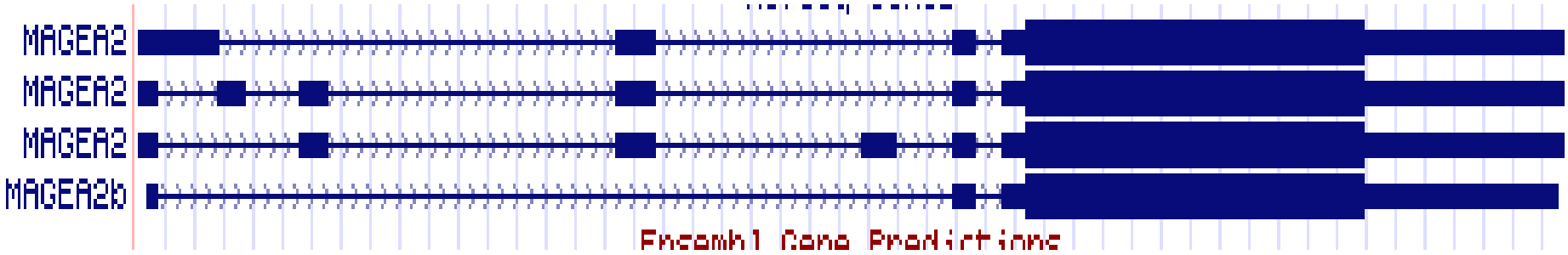
=> мы не можем строить их по одному и сравнивать

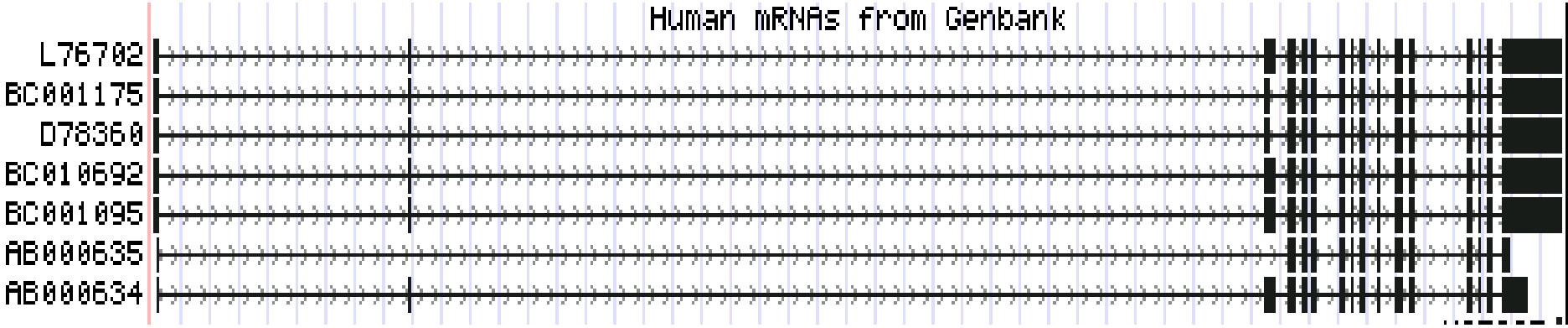
Распознавание генов

Сегментация геномной последовательности на белок-кодирующие и некодирующие области

- использует различия в статистических особенностях этих областей
- в геномах эукариот – сложно, потому что в одном гене кодирующие последовательности (*экзоны*) перемежаются некодирующими вставками (*интроны*). Ср. рекламные паузы.

Гены (схематическое изображение)





Игрушечный пример

Сколько нужно операций, чтобы вычислить

$$\sum_{i=1\ldots m,\, j=1\ldots n} x_i \cdot y_j =$$

$$= x_1 \cdot y_1 + x_1 \cdot y_2 + \ldots + x_1 \cdot y_n +$$

$$+ x_2 \cdot y_1 + x_2 \cdot y_2 + \ldots + x_2 \cdot y_n +$$

$$+ \ldots +$$

$$+ x_m \cdot y_1 + x_m \cdot y_2 + \ldots + x_m \cdot y_n$$

Игрушечный пример

Сколько нужно операций, чтобы вычислить

$$\sum_{i=1\ldots m,\, j=1\ldots n} x_i \cdot y_j =$$

$$= x_1 \cdot y_1 + x_1 \cdot y_2 + \ldots + x_1 \cdot y_n +$$

$$+ x_2 \cdot y_1 + x_2 \cdot y_2 + \ldots + x_2 \cdot y_n +$$

$$+ \ldots +$$

$$+ x_m \cdot y_1 + x_m \cdot y_2 + \ldots + x_m \cdot y_n$$

Наивный ответ:

mn умножений и ***mn*** **– 1** сложений

но перепишем это как

$$(x_1 + x_2 + \ldots + x_m) \cdot (y_1 + y_2 + \ldots + y_n) = \\ = \sum_{i=1\ldots m} x_i \cdot \sum_{j=1\ldots n} y_j$$

и нужно

***m* + *n* – 2** сложений и всего **1** умножение

Задача

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные**?**

(b) опытные**?**

Ответ

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot$$
$$\cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные? $(y_1 + y_2 + \ldots + y_n) \cdot m + m - 1$

(b) опытные?

Ответ

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные? $(y_1 + y_2 + \ldots + y_n) \cdot m + m - 1$

(b) опытные? $(y_1 + y_2 + \ldots + y_n) + m - 2$

Задача

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные? $(y_1 + y_2 + \ldots + y_n) \cdot m - 1$

(b) опытные? $(y_1 + y_2 + \ldots + y_n) + m - 2$

(c) есть ещё и операция “возведение в степень”?

Ответ

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные? $(y_1 + y_2 + \ldots + y_n) \cdot m - 1$

(b) опытные? $(y_1 + y_2 + \ldots + y_n) + m - 2$

(c) есть ещё и операция “возведение в степень”?
mn в степень и $mn - 1$ умножений или
n в степень и $m + n - 2$ умножений

Задача

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные? $(y_1 + y_2 + \ldots + y_n) \cdot m - 1$

(b) опытные? $(y_1 + y_2 + \ldots + y_n) + m - 2$

(c) есть ещё и операция “возведение в степень”?
mn в степень и $mn - 1$ умножений или
n в степень и $m + n - 2$ умножений

(d) есть ещё и сложение?

Ответ

сколько умножений нужно, чтобы вычислить

$$x_1^{y_1} \cdot x_1^{y_2} \cdot \ldots \cdot x_1^{y_n} \cdot x_2^{y_1} \cdot x_2^{y_2} \cdot \ldots \cdot x_2^{y_n} \cdot \ldots \cdot x_m^{y_1} \cdot x_m^{y_2} \cdot \ldots \cdot x_m^{y_n} = \prod_{i=1\ldots m,\, j=1\ldots n} x_i^{y_j}$$

если мы

(a) наивные? $(y_1 + y_2 + \ldots + y_n) \cdot m - 1$

(b) опытные? $(y_1 + y_2 + \ldots + y_n) + m - 2$

(c) есть ещё и операция “возведение в степень”?
mn в степень и $mn - 1$ умножений или
n в степень и $m + n - 2$ умножений

(d) есть ещё и сложение?
1 в степень, $m - 1$ умножение, $n - 1$ сложение

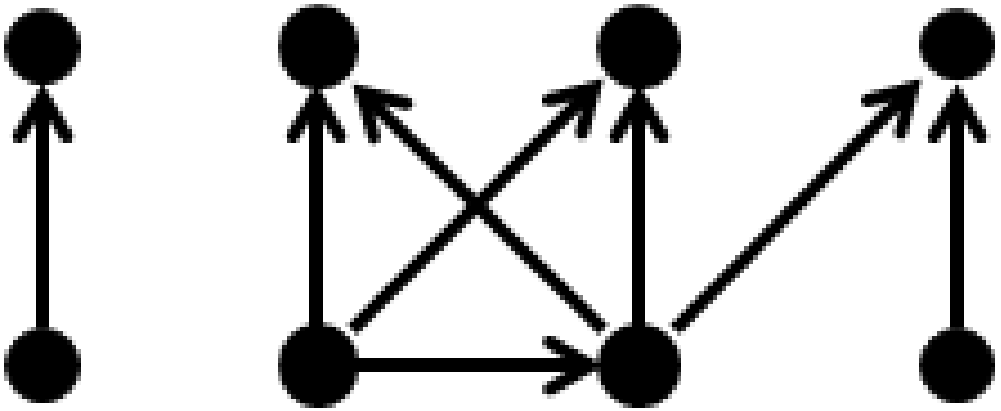
Мудрость

Изменение порядка вычислений с использованием свойств данных может сильно сократить число операций

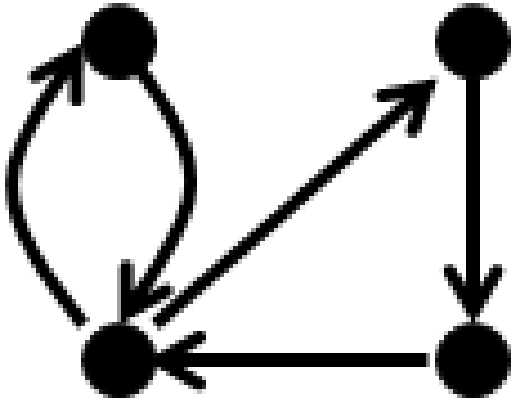
Графы

Вершины

Рёбра – упорядоченные пары вершин

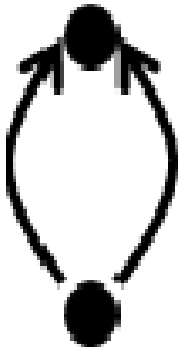


множественные источники и стоки

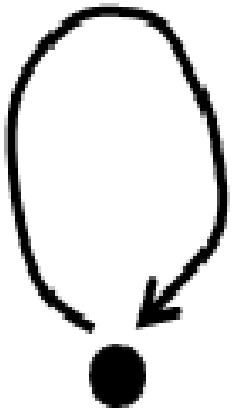


содержит циклы

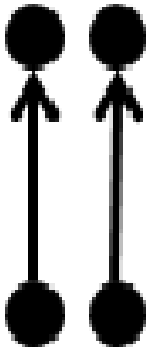
“плохие” графы и не графы



множест-
венные
ребра

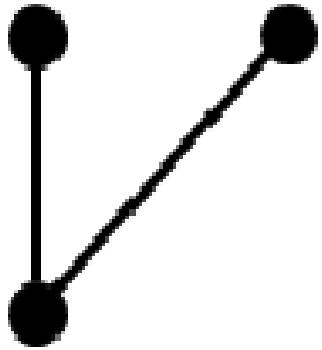


петля



много
компо-
нент

не граф
(висит
ребро)



ненаправ-
ленный
граф

Определения

Источник – вершина, в которую не входит ни одно ребро

Сток – вершина, из которой не выходит ни одного ребра

Проход p длины N – упорядоченное множество N рёбер $p = (a_1, \ldots, a_N)$ такое, что конец ребра $a_n = (b_n, e_n)$ совпадает с началом ребра a_{n+1}, то есть $e_n = b_{n+1}$ для всех $n = 1, \ldots, N-1$. В графе без петель и множественных ребер проход можно определять и как упорядоченное множество вершин $p = (v_1, \ldots, v_{N+1})$ такое, что для каждой пары соседних вершин v_n, v_{n+1} существует ребро $a_n = (v_n, v_{n+1})$, $n = 1, \ldots, N$.

Путь – проход, в котором каждое ребро используется только один раз.

Цикл – путь, в котором конец последнего ребра a_N совпадает с началом первого ребра a_1, то есть $e_N = b_1$.

Ациклический граф не содержит циклов.

Задача

(a) Нарисовать все ориентированные связные ациклические графы с тремя вершинами

(b) Сколько будет разных графов, если вершины помечены символами ***A***, ***B*** и ***C***?

(c) Докажите, что в ациклическом графе есть хотя бы один источник и один сток.

(d) Укажите источники и стоки в графах из **(a)**.

Ответ

(a) Нарисовать все ориентированные связные ациклические графы с тремя вершинами
(4)

(b) Сколько будет разных графов, если вершины помечены символами ***A***, ***B*** и ***C***?
(18)

(c) Докажите, что в ациклическом графе есть хотя бы один источник и один сток.

(d) Укажите источники и стоки в графах из **(a)**.

Проблема

Рассмотрим ациклический граф с одним источником и одним стоком. Припишем каждому ребру число (*вес*). *Вес пути* определим как сумму весов ребер.

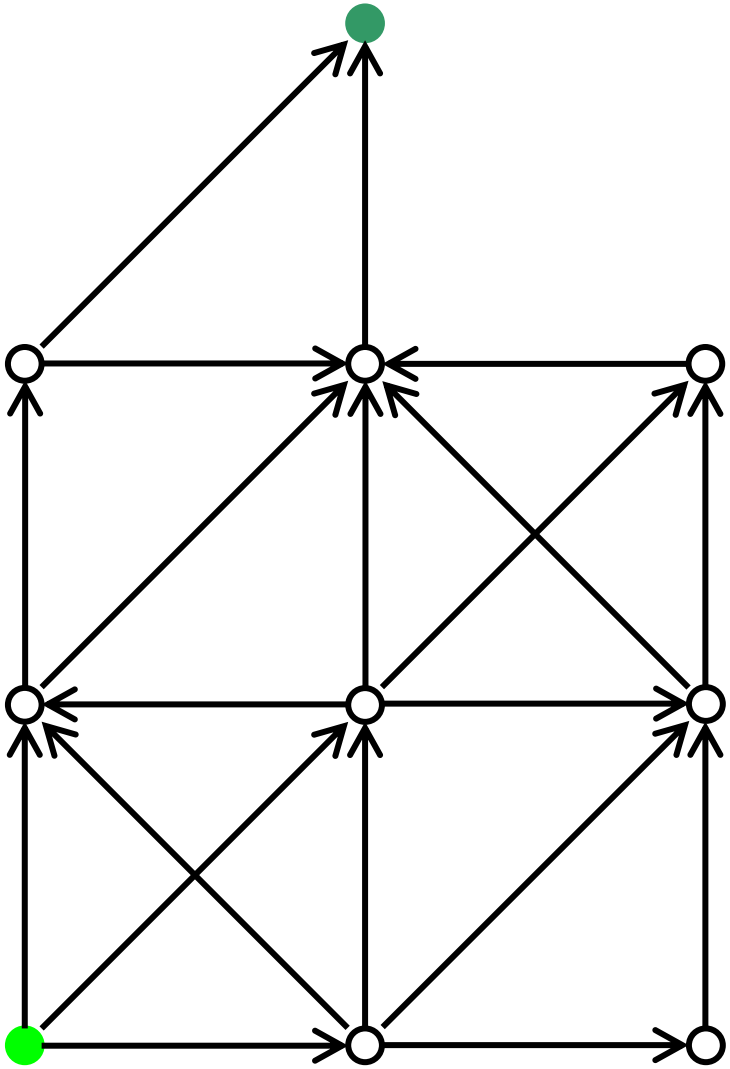
Надо найти путь наибольшего веса от источника к стоку.

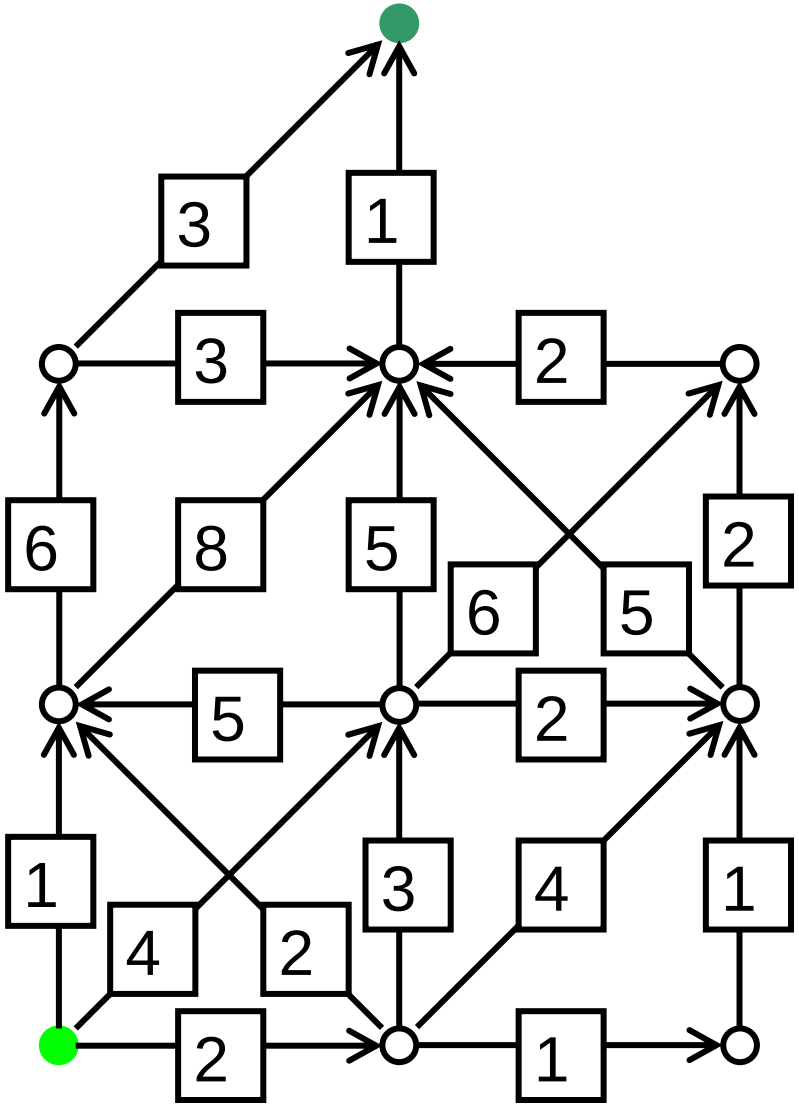
Наблюдение

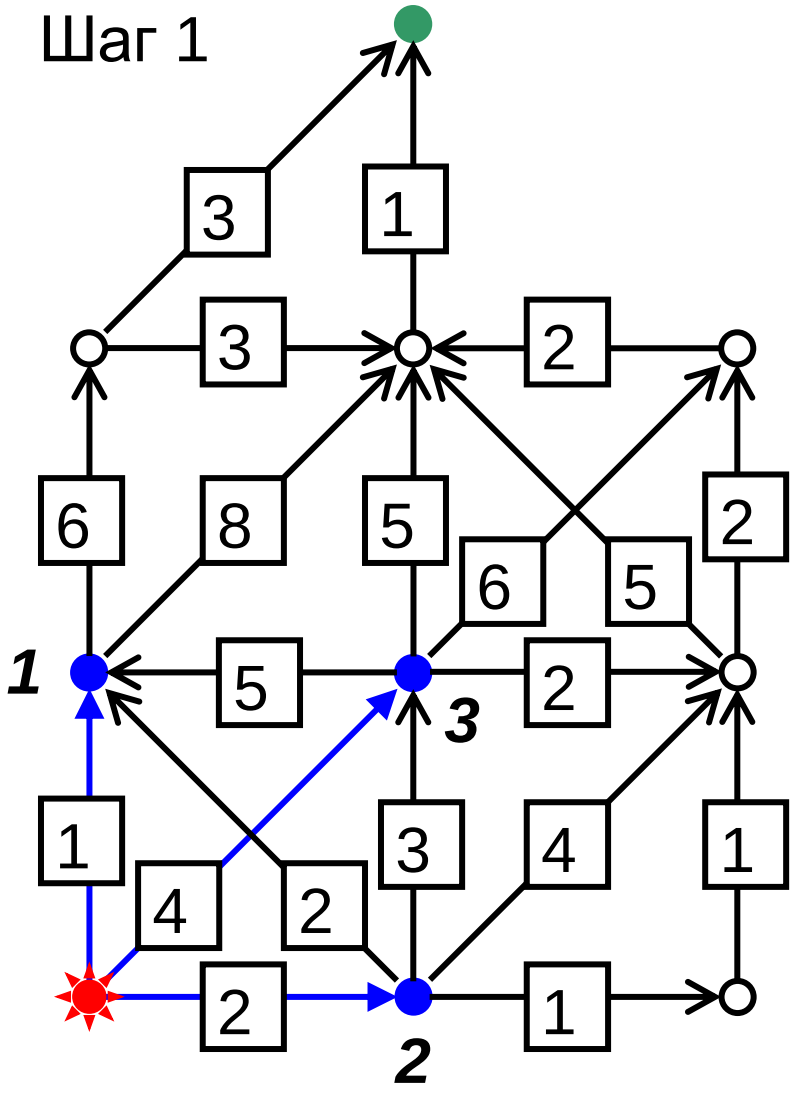
Если два подпути ***P*** и ***Q*** заканчиваются на одной и той же вершине ***v***, а вес пути ***P*** больше, чем вес пути ***Q***, то для любой пары путей ***P**** и ***Q****, которые начинаются с ***P*** и ***Q*** соответственно и совпадают после ***v***, вес ***P**** будет больше, чем вес ***Q****.

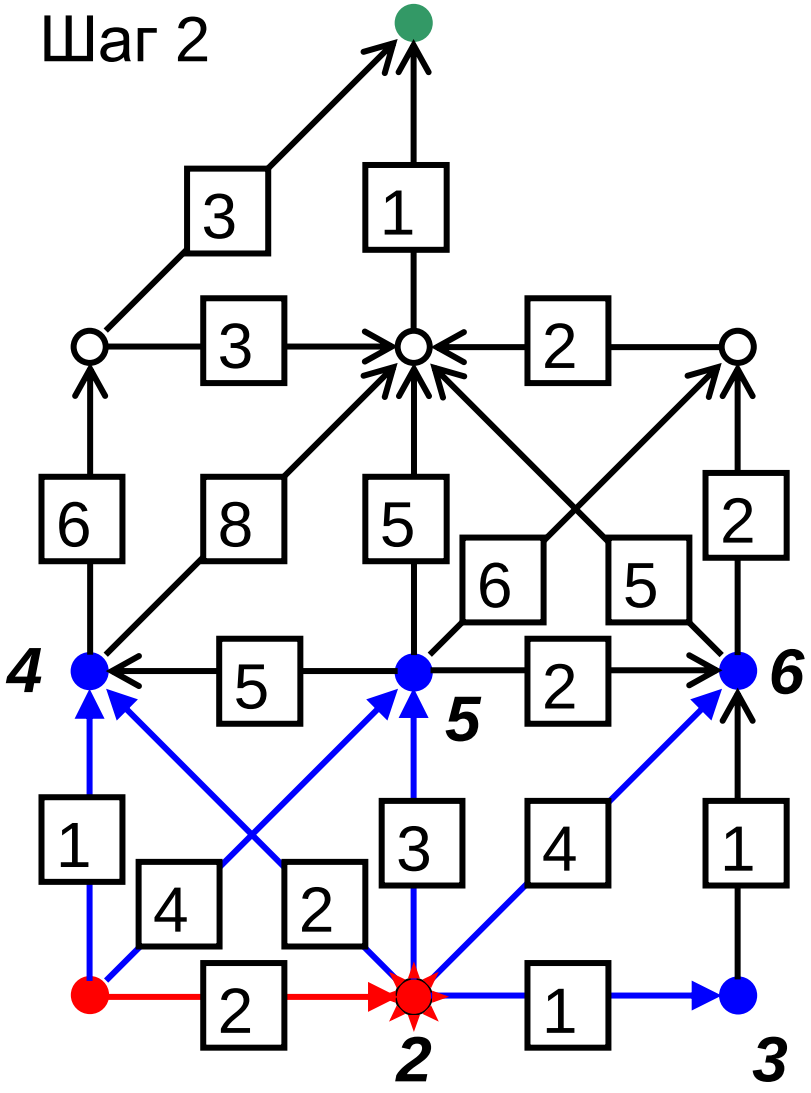
Теперь нам не надо рассматривать все пути, достаточно для каждой вершины построить наилучший путь до неё из источника, завершив построение в стоке.

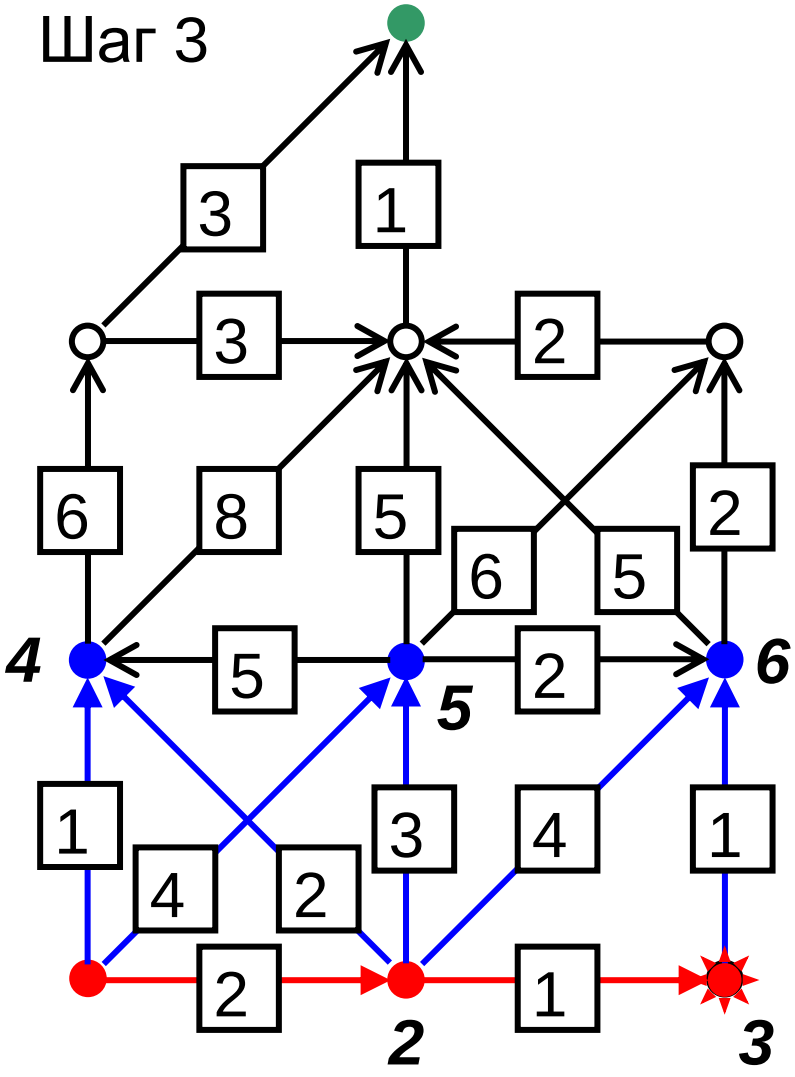
Давайте сделаем это для графа

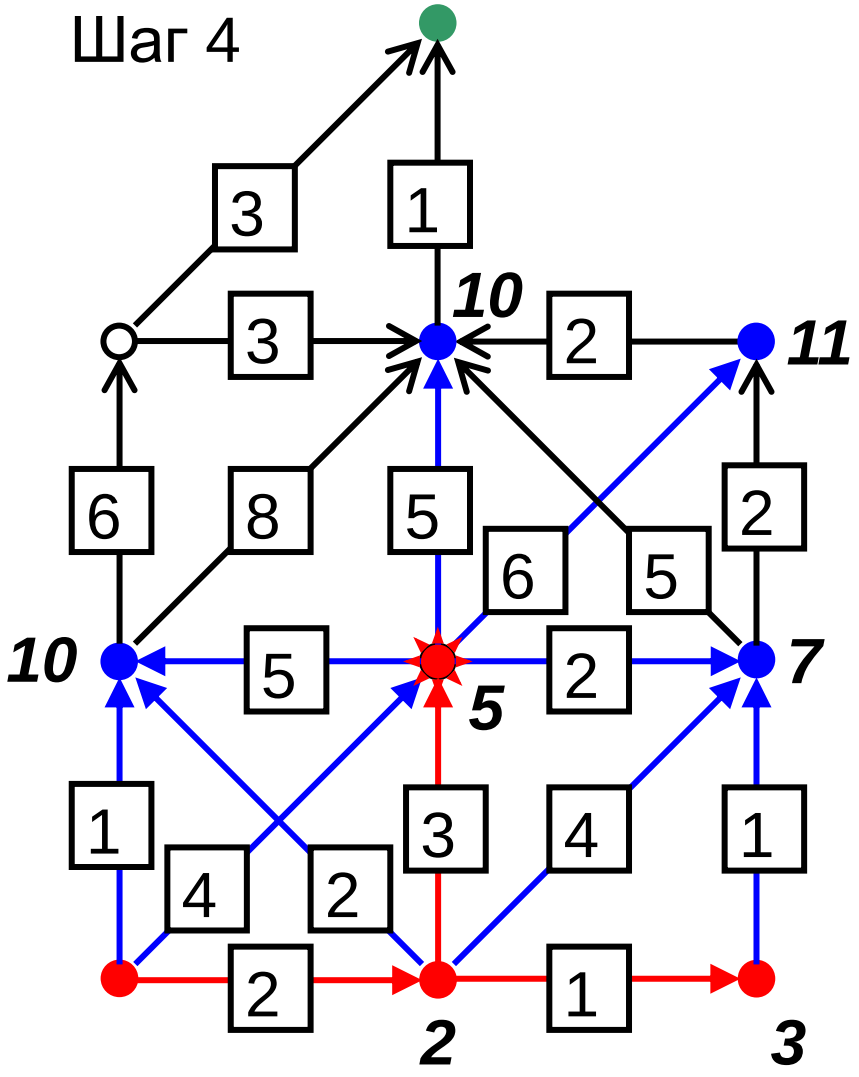


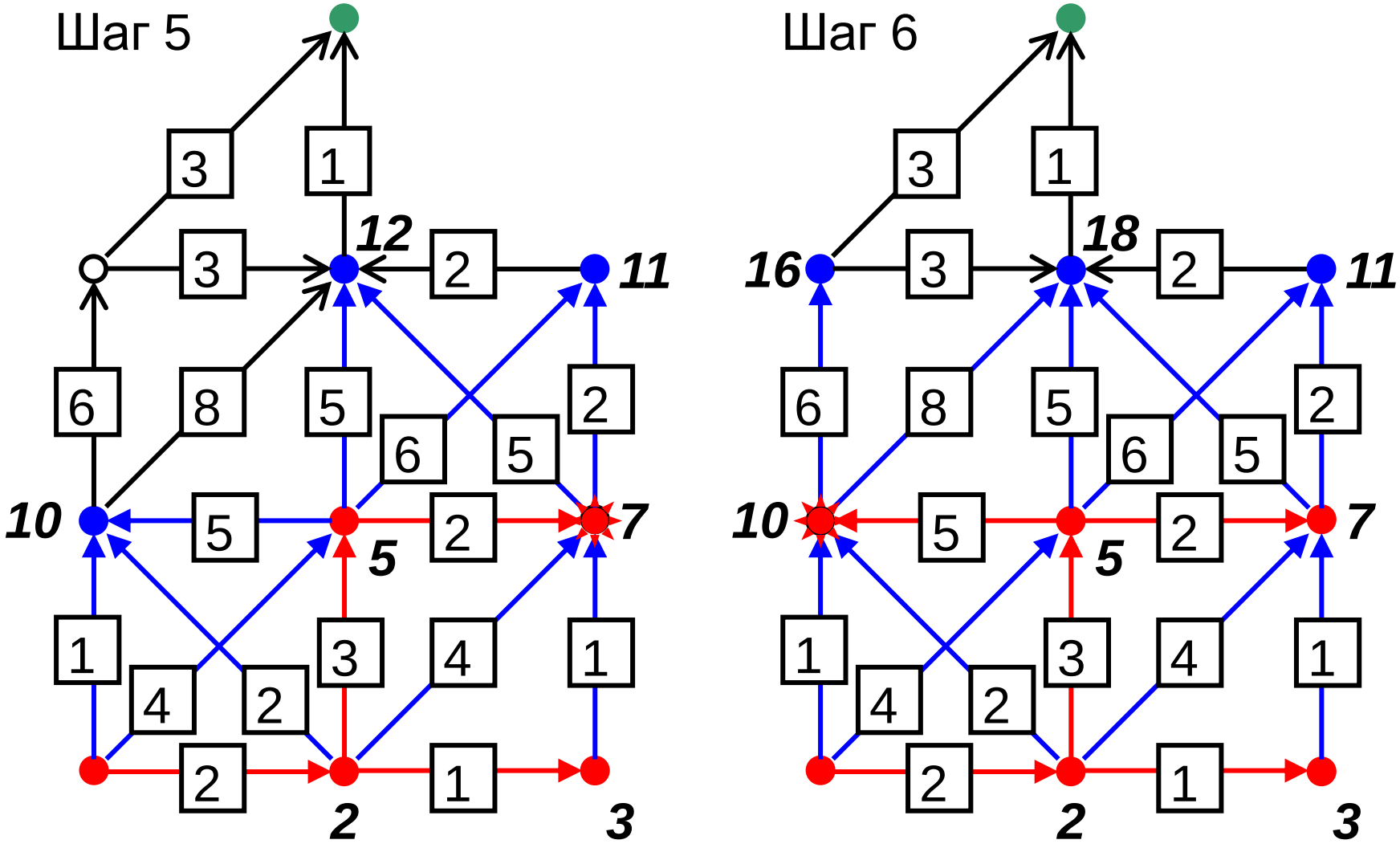


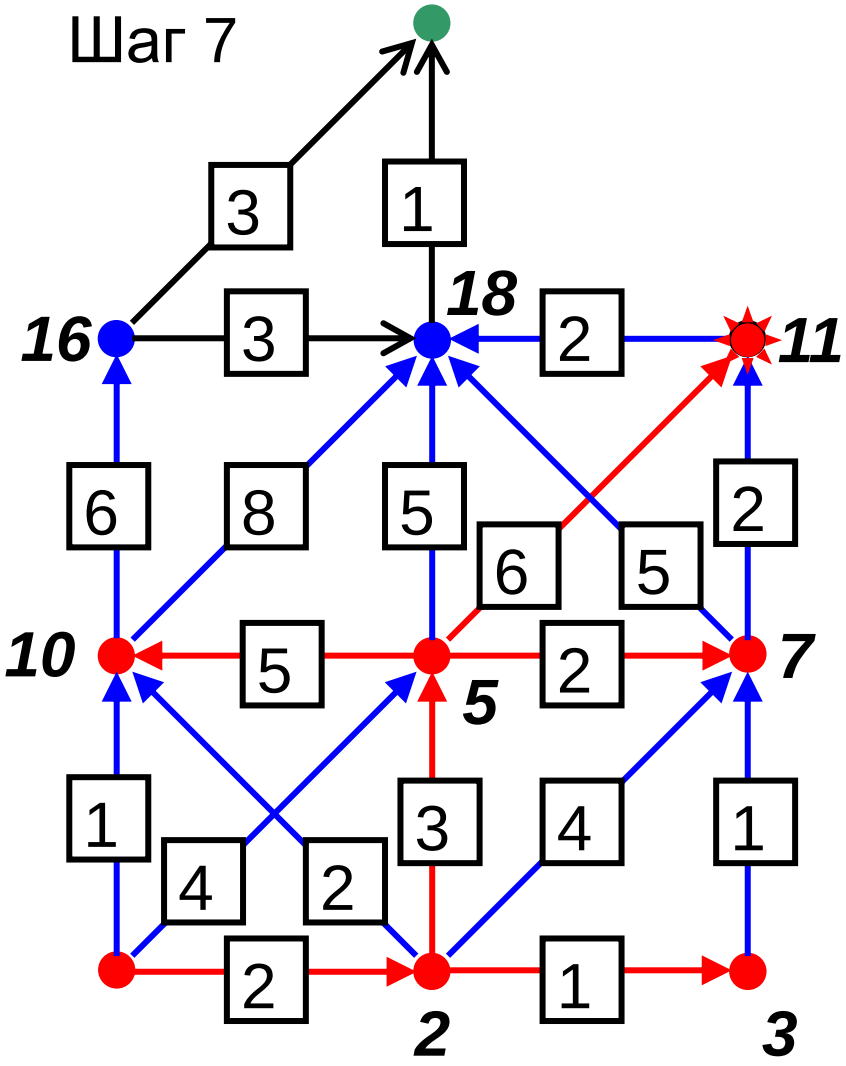

Шаг 1
3
1
3
2
6
8
5
6
5
2
1
5
2
3
1
3
4
1
4
2
2
1
2

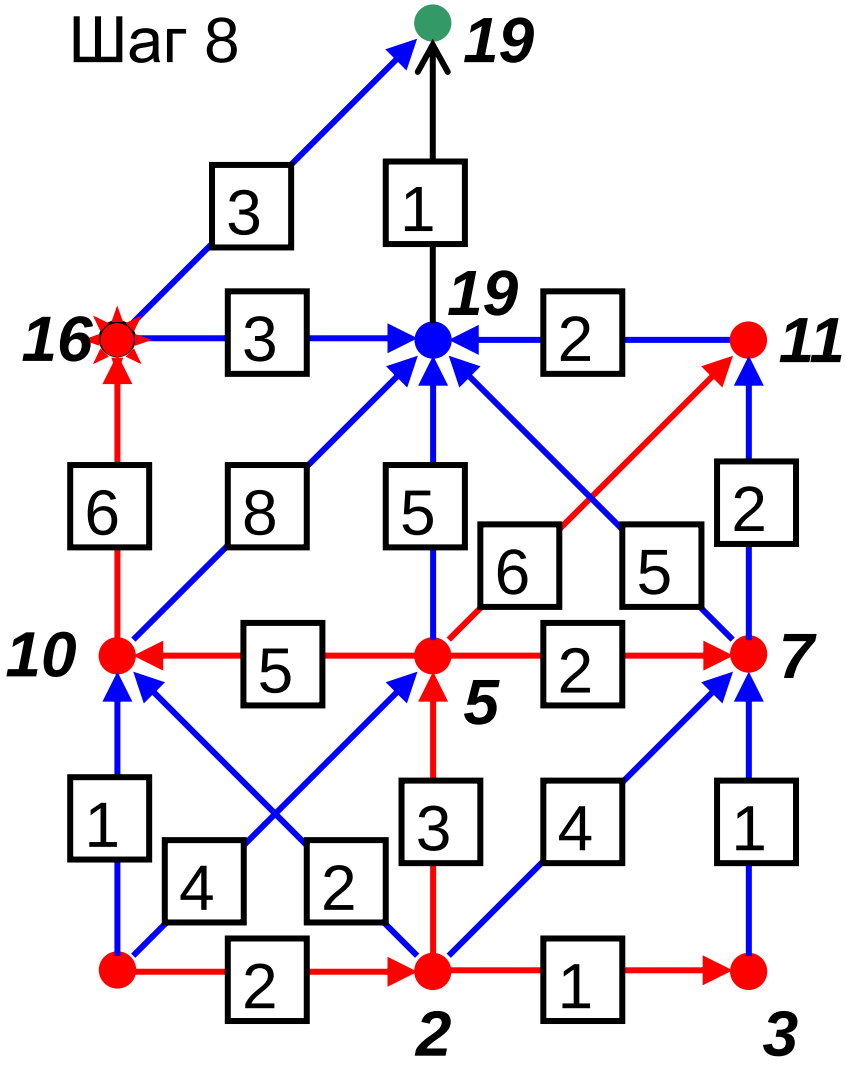

Шаг 2
3
1
3
2
6
8
5
6
5
2
4
5
2
6
5
1
3
4
1
4
2
2
1
3


Шаг 3
3
1
3
2
6
8
5
6
5
2
4
5
5
2
6
1
4
2
3
4
1
2
1
2
3

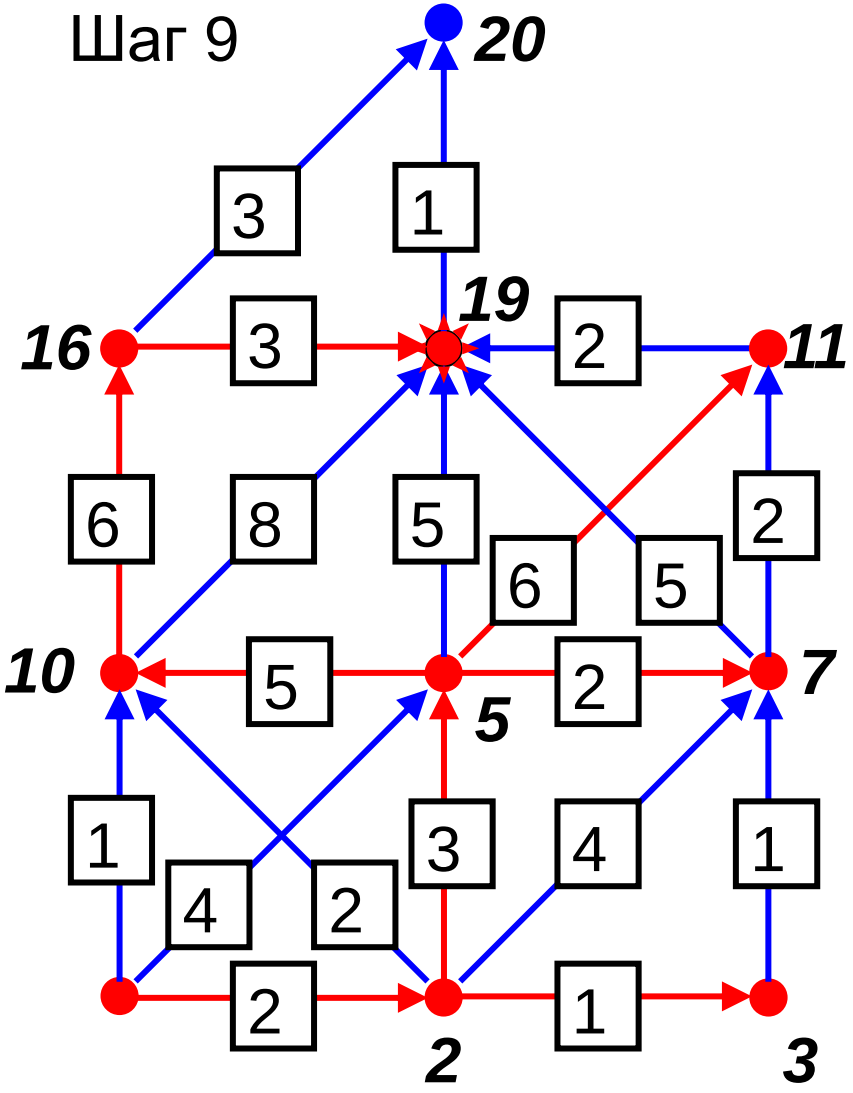

Шаг 4
3
1
10
3
2
11
6
8
5
6
5
2
10
5
5
2
7
1
4
2
3
4
1
2
1
2
3


Шаг 5
3
1
12
3
2
11
6
8
5
6
5
2
10
5
2
7
5
1
3
4
4
2
2
1
2
3
Шаг 6
3
1
18
16
3
2
11
6
8
5
6
5
2
10
5
2
7
5
1
3
4
4
2
2
1
2
3

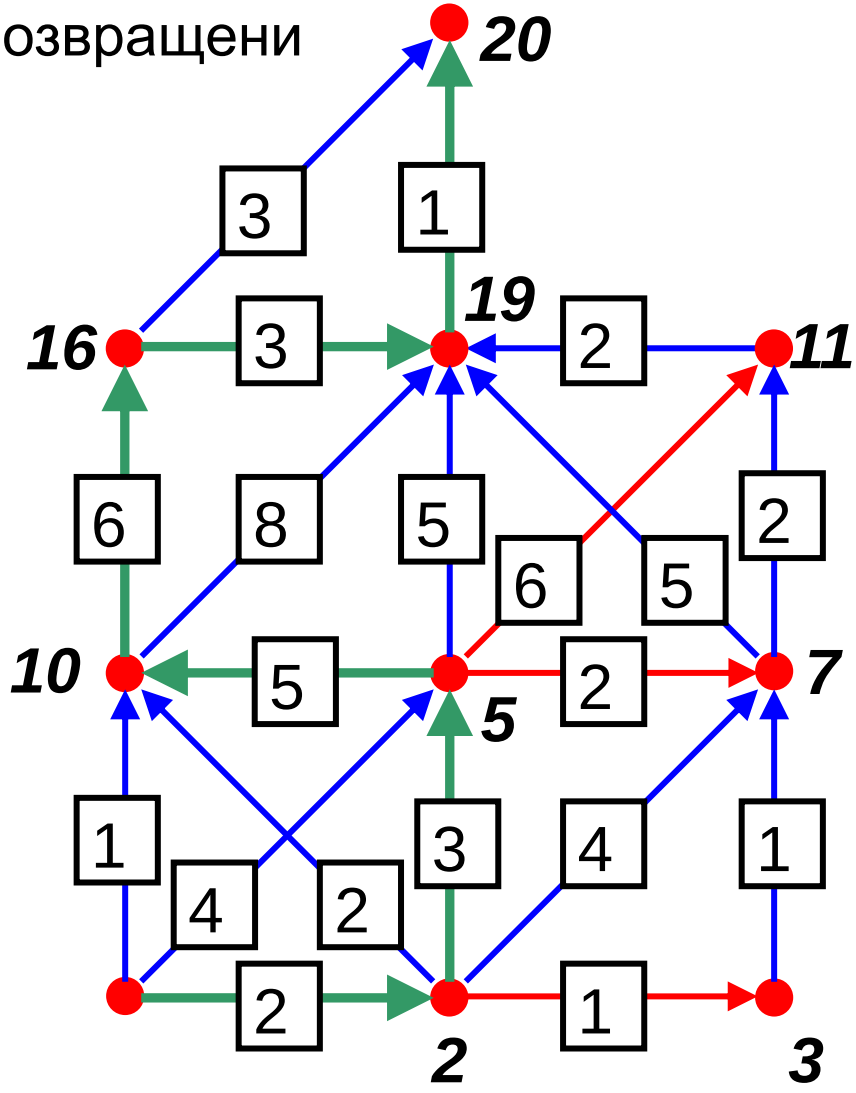

Шаг 7
3
1
16
3
18
2
11
6
8
5
6
5
2
10
5
5
2
7
1
3
4
2
4
1
2
1
2
3


Шаг 8
19
3
1
16
3
19
2
11
6
8
5
6
5
2
10
5
5
2
7
1
3
4
2
4
1
2
1
2
3

Шаг 9



Возвращение



Алгоритм

Data types and definitions:

vertices: *v*, *u*, *Source*, *Sink*;
arcs: (*v*,*u*), *a*;
start vertex of arc *a*: *B*(*a*);
weight of arc (*v*,*u*): *W*(*v*,*u*);
path: *BestPath*; // defined as a set of arcs
the highest score of subpath ending at *v*: *S*(*v*);
the highest score of subpath ending at *u* and coming through (*v*,*u*): *T*(*v*,*u*);
the last arc of the highest scoring subpath ending at *u*: *L*(*u*);

Initialize: **for** each vertex *v*: *S*(*v*) := minus_infinity.
Forward process**: while** There are unprocessed vertices:
  v := arbitrary unprocessed vertex with all incoming arcs processed;
  for each arc (*v*,*u*): // consider all arcs starting at *v*
    T(*v*,*u*) := *S*(*v*)+*W*(*v*,*u*);
    if *T*(*v*,*u*)>*S*(*u*) // subpath coming through *v* is better than the current best subpath ending at *u*
    then: // update the data for *u*
      S(*u*) := *T*(*v*,*u*);
      L(*u*) := (*v*,*u*);
    endif;
    (*v*,*u*) := processed_arc;
  endfor;
  v := processed_vertex;
endwhile.
Backtracing:
  BestPath = *empty_set*; // initialize
  v := *Sink*; // go from the sink backwards by marked arcs
  until *v=Source*
    Add *L*(*v*) to *BestPath*; // add the last arc of the best path ending at the current vertex
    v := *B*(*L*(*v*)); // go to the start vertex of this arc
  enduntil.
Output *BestPath*.

Количество операций

Лимитирующая операция – обработка вершин и добавление рёбер к (под)путям, но мы рассматриваем каждое ребро только один раз.

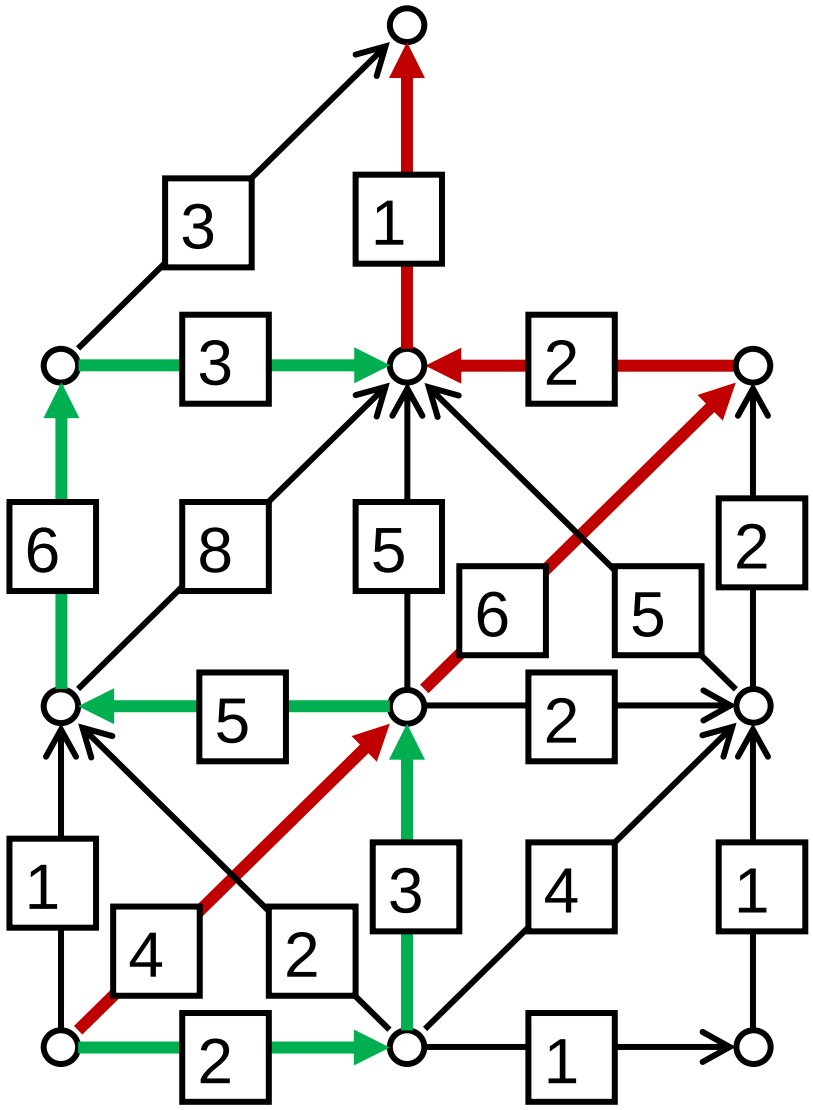
Стало быть, количество операций линейно зависит от количества рёбер A: время работы алгоритма $O(A)$

Жадный алгоритм

Стартовать у источника и всякий раз выбирать ребро наибольшего веса.

13 < 20

Не работает.



Задача

(а) Построить самый простой граф, для которого жадный алгоритм будет давать правильный ответ

Задача

(a) Построить самый простой граф, для которого жадный алгоритм будет давать правильный ответ

(b) Построить граф с тремя вершинами, для которого жадный алгоритм будет давать правильный ответ

(c) Построить граф с тремя вершинами, для которого жадный алгоритм **не** будет давать правильный ответ

Ещё задача

(а) Написать алгоритм для построения пути с наибольшим количеством рёбер

Еще задача

(а) Написать алгоритм для построения пути с наибольшим количеством рёбер

Намёк: не надо менять алгоритм, надо правильно определить веса на рёбрах

Еще задача

(a) Написать алгоритм для построения пути с наибольшим количеством рёбер

Намек: не надо менять алгоритм, надо правильно определить веса на рёбрах

(b) Модифицировать алгоритм, чтобы он строил путь наименьшего веса

Мудрость

Алгоритм динамического программирования можно применять для решения различных задач. Их общее свойство – то, что они разлагаются в упорядоченное множество вложенных более простых подзадач, и для решения более сложной задачи достаточно знать решения подподзадач, а не рассматривать все возможные решения.

Примечание

Не все задачи оптимизации путей на графах таковы.

Задача коммивояжера. По заданному неориентированному графу в весами на ребрах найти путь наименьшего веса, проходящий через все вершины.

Это ***NP-полная*** задача (из-за условия про посещение всех вершин).

Эффективные алгоритмы не известны. Большинство ученых верят, что для таких задач число необходимых операций экспоненциально относительно объема данных (т.е. фактически решение сводится к перебору).

Выравнивание двух белковых последовательностей

BRCA1 Xenopus laevis vs Pan troglodytus

```
fr  MtcSrMdIEgIcSVISvMQKnLECPICLELMKEPVATKCDHIFCKFCMLQLLSkKKKGtv
ch  MdlSaLrVEeVqNVINaMQKiLECPICLELIKEPVSTKCDHIFCKFCMLKLLN-QKKGps

fr  pCPLCKtEVTRRSLQEShRFklLVEgqLKIIkAFEfDSGyKFfpSqehtKglDSTiEdvl
ch  qCPLCKnDITKRSLQEStRFsqLVEelLKIIcAFQlDTGlEYanSynfaKkeNNSpEh--

fr  VKEDqSIVhckGYRNRkKgVfnrKtyEetgMlsvSkAeEqfakevtRlIpcRQK-KPKKE
ch  LKDEvSIIqsmGYRNRaKrLlqsEp-EnpsLqetSlSvQlsnlgtvRtLrtKQRiQPQKK

fr  AalIf--SNcvpDS---------sDgDLLn-kenGlRNDcSplhyekeDTqipemeEmvE
ch  SvyIelgSDsseDTvnkatycsvgDqELLqitpqGtRDEiSl------DSakkaacEfsE

fr  SDLaecEfaEsAgSNLlgfD--gpEgiPEisaeTSINaagNcDfyGrkTeqfpndHhcSf
ch  TDVtntEhhQpSnNDLnttEkratErhPEkyqgSSVSnl-HvEpcGtnThasslqHenSs

fr  kqniaDaeqnKRnQhCgnvpfapMgKSnlDeketvEtdfDNQhndSnpE----NnDPLgK
ch  llltkDrmnvEKaEfCnkseqpgLaRSqhNrwagsKetcNDRrtpSteKkvdlNaDPLcE
```

Выравнивания

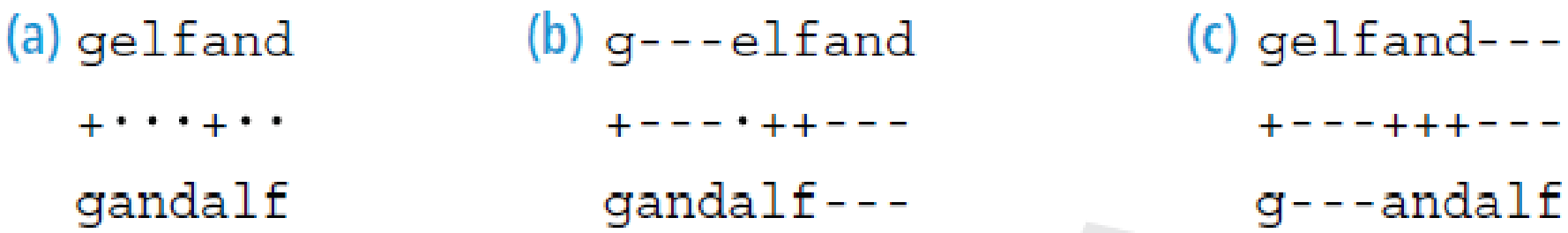
Для двух заданных символьных последовательностей (нуклеотидных или аминокислотных) длин ***M*** и ***N,*** установить соответствие так, что некоторые символы будут образовывать пары (совпадающие или несовпадающие), а некоторые символы будут игнорироваться. Порядок совпадающих символов должен совпадать.

Вес выравнивания – это сумма премий за совпадения (***r*** за пару) минус сумма штрафов за несовпадения (***p*** за пару) и делеции (***q*** за символ).

Цель – построить выравнивание наибольшего веса.

Задача

Чему равны веса выравниваний



Сведем к задаче построения оптимального пути

Построим граф.

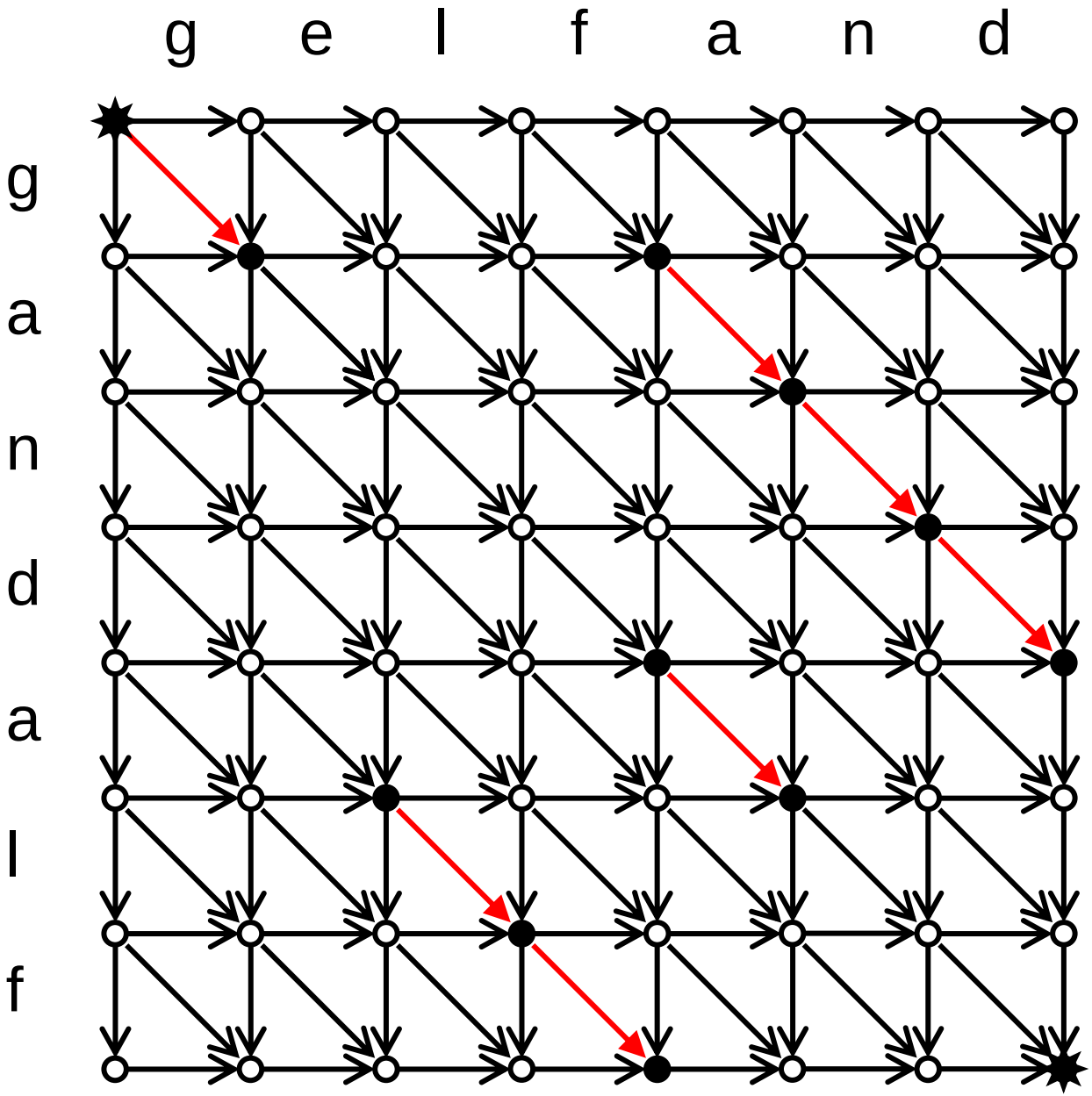
Вершины – пары позиций (концы частичных выравниваний)

Из (почти) каждой вершины выходят (и в нее входят) три ребра, описывающие продолжение выравнивания:

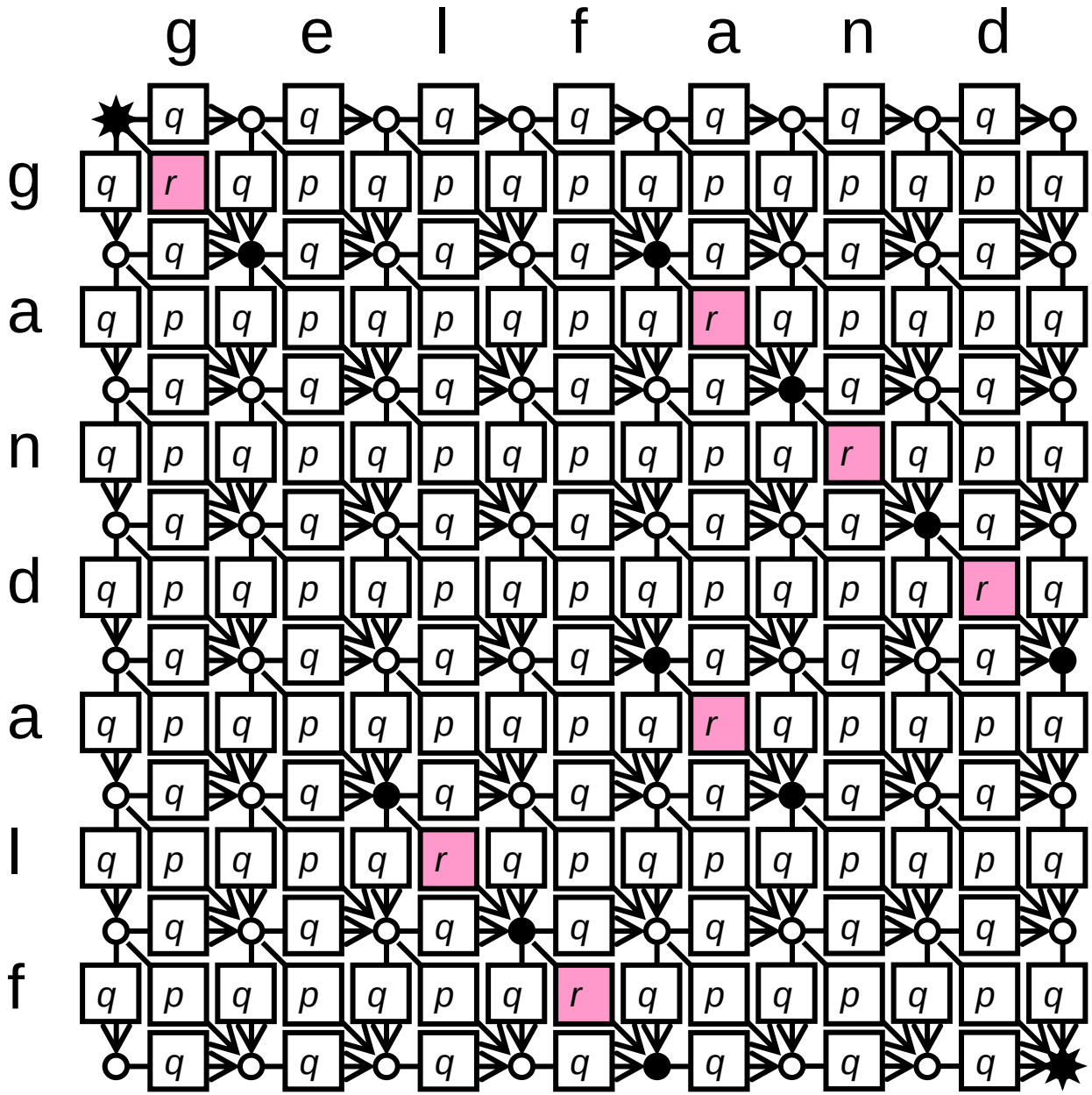
- сопоставление: совпадение (вес r)
  или несовпадение (вес $(-p)$): $M \cdot N$ ребер
- удаление в 1-й последовательности (вес $(-q)$): $M \cdot (N + 1)$ ребер
- удаление во 2-й последовательности (вес $(-q)$): $(M + 1) \cdot N$ ребер

Пути соответствуют выравниваниям

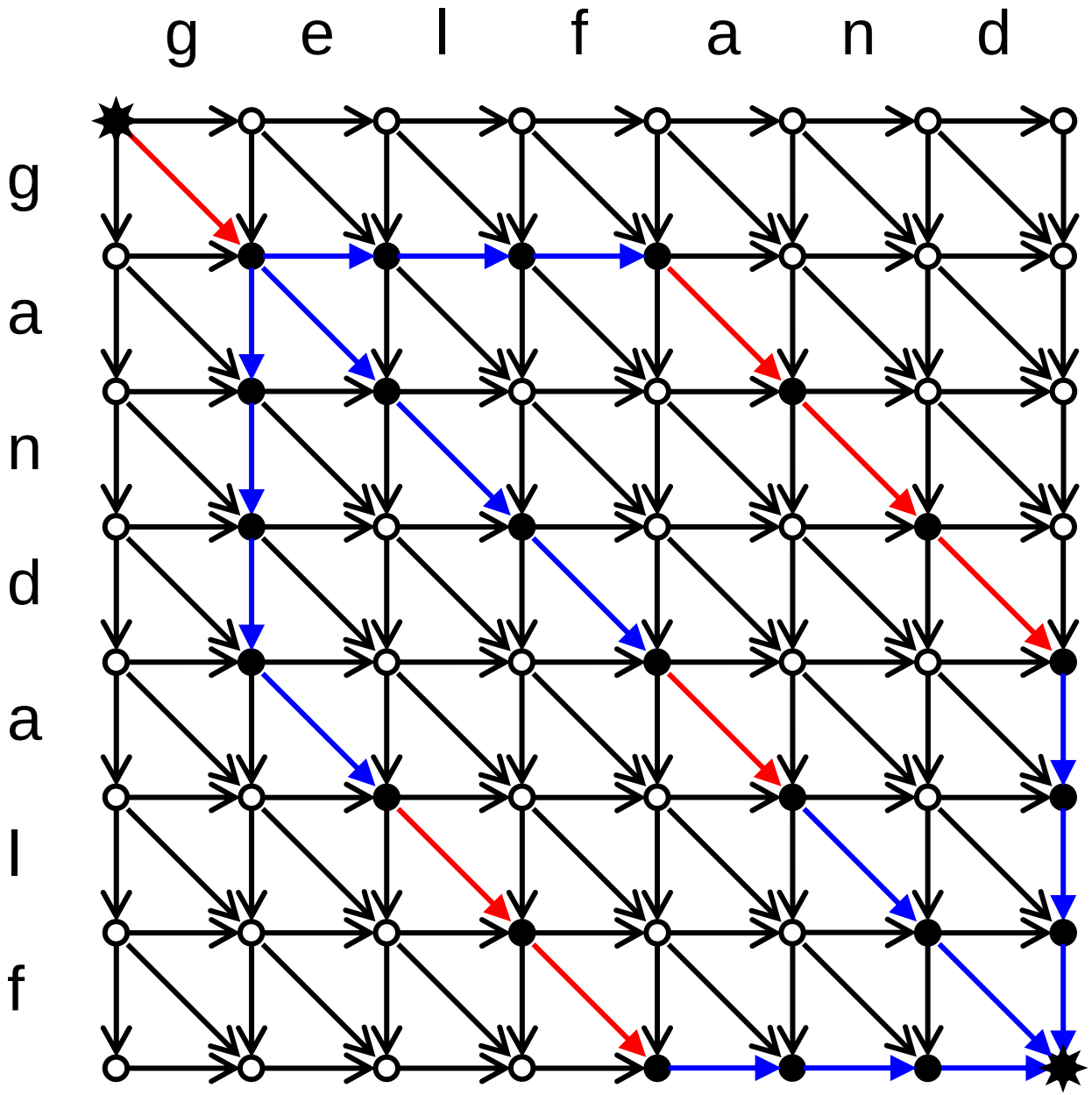
Граф выравнивания



Граф выравнивания с весами



Пути для трёх выравниваний

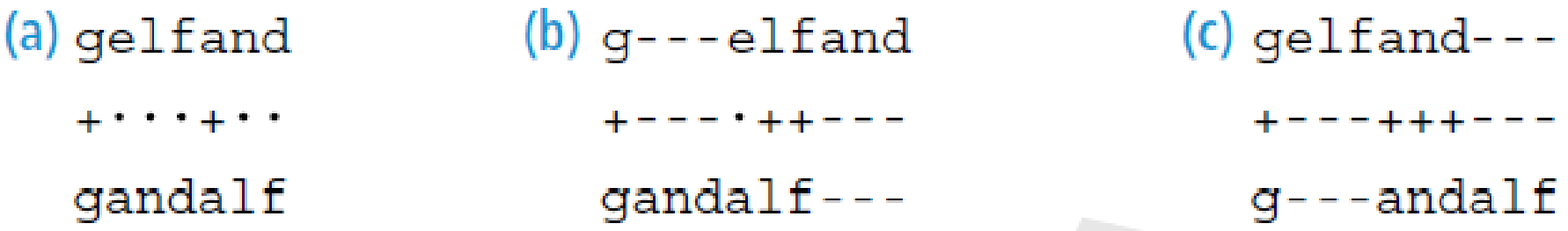


Варианты

- Выравнивание со свободными концами (сборка геномов)
  - нулевые веса рёбер от источника к верхнему и левому «периметру» и от правого и нижнего «периметра» к стоку
- Локальное выравнивание
  - нулевые веса рёбер от источника ко всем внутренним вершинам и от всех внутренних вершин к стоку

Задача

Для выравниваний

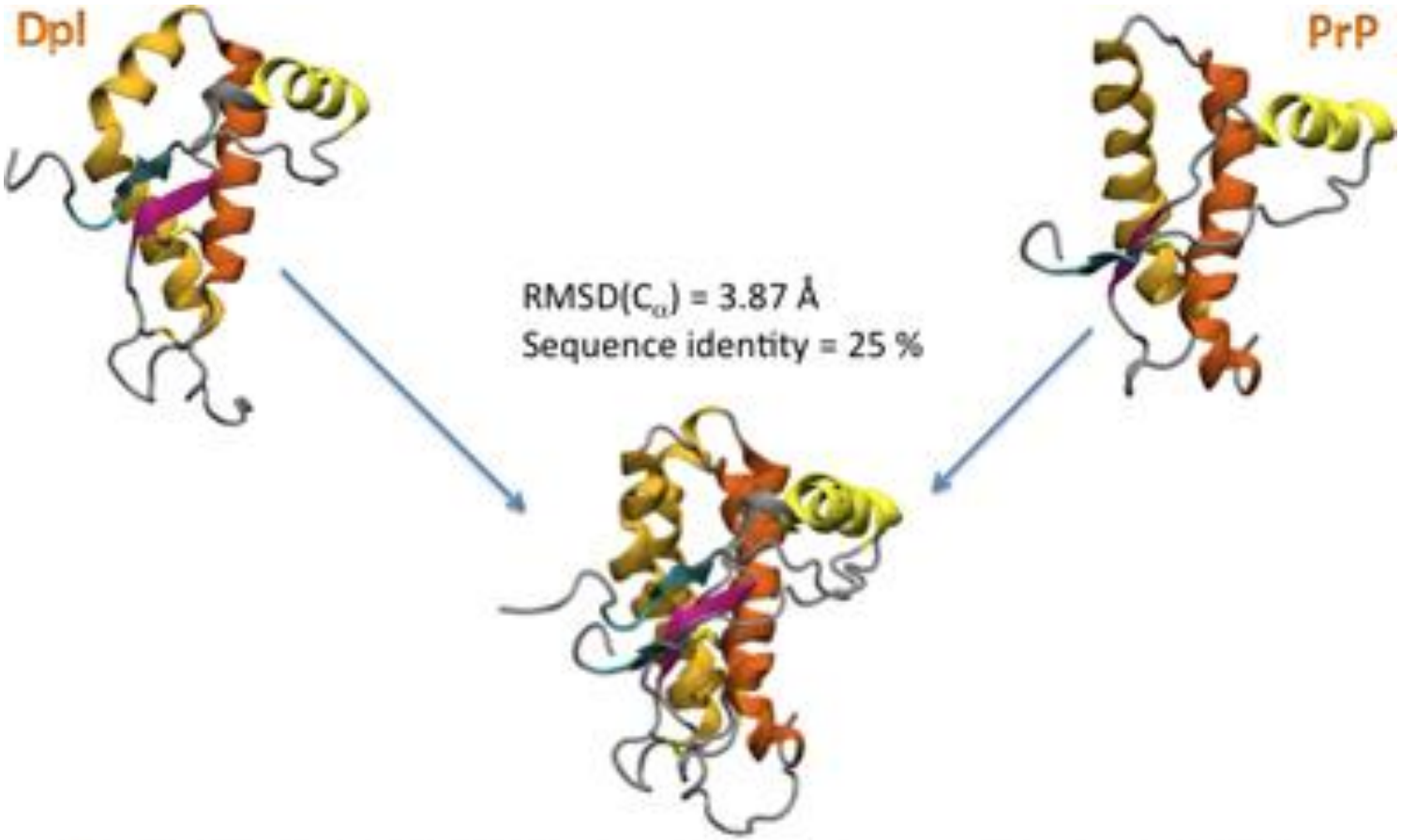


установим премию за совпадение ***r* = 10**. Какая комбинация штрафов за несовпадение и делеции сделает оптимальным каждое из выравниваний **(a), (b)** и **(c)?**

Веса

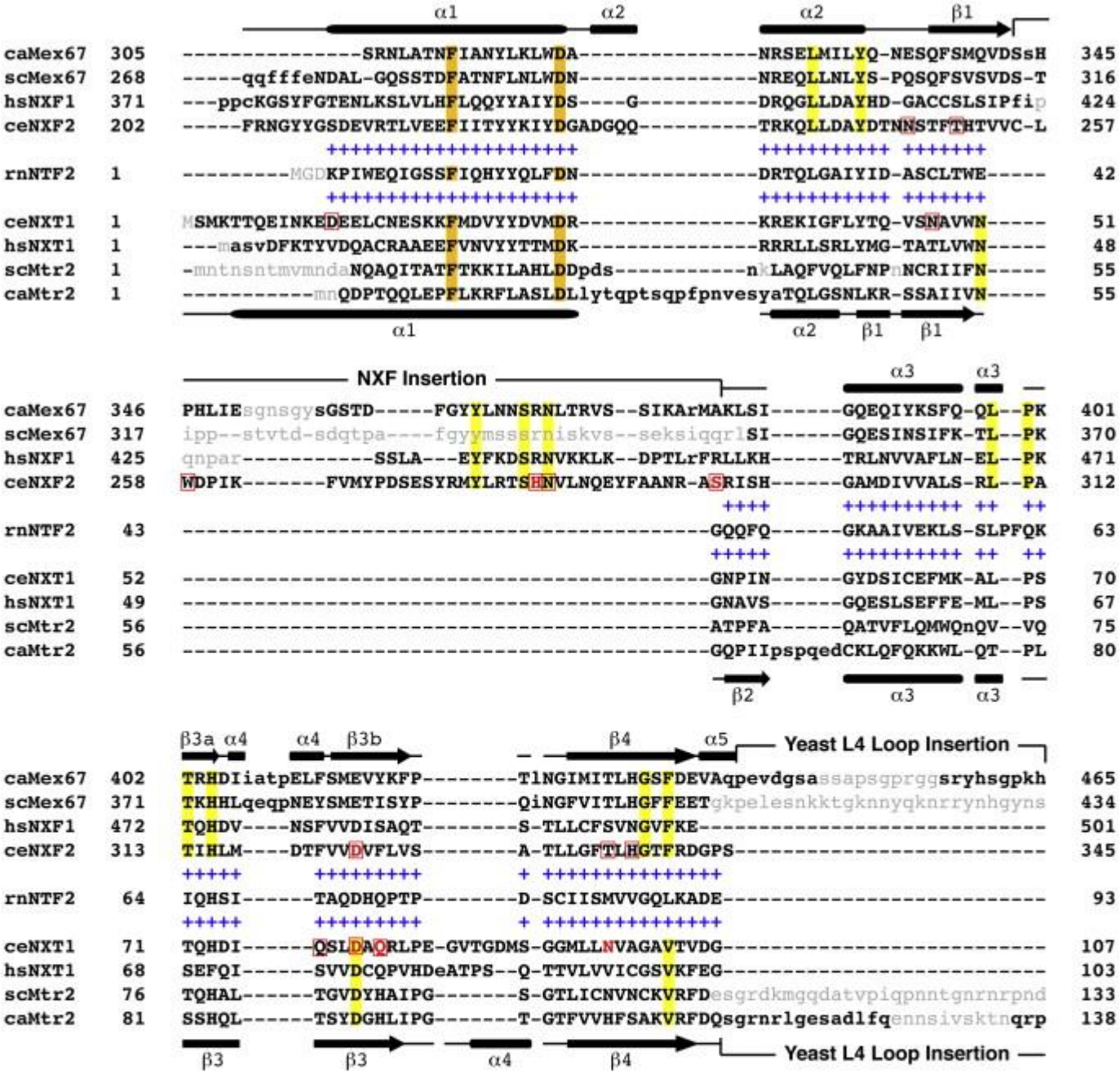
- Матрица весов замен аминокислот
  - физико-химические
  - эволюционные
- Штрафы за удаления
  - аффинные (открытие удаления и продолжение удаления)
- Золотой стандарт – структурные выравнивания

Структурные выравнивания




```
Dpl (1I17): RVAEN-RPG-AFIKQGRKLDIDFG-AEGNRYYAANYWQFPDGIYYE-GCSEANVT- …
PrP (1AG2): ----GLGGYMLGSAM-SRPMIHFGNDWEDRYYRENMYRYPNQVYYRPVD--Q-YSN …

KEMLVTSCVNATQAAN-QAEFS----REKQDSKLHQRVLWRLIKEICSAK-HCDF-----WLERGAA
QNNFVHDCVNITIKQHTVTT-TTKGENFT---ETDVKMMERVVEQMCVTQYQ-KESQAYY-------
```

© Julian Garrec website



D.E.Kerkov et al., 2012

Множественное выравнивание

- тройное → кубический граф
  - и т.д.
- для K последовательностей длины N нужно $O(N^K)$ операций
- скоро перестает работать
- последовательное выравнивание
  - все попарные => матрица расстояний
  - дерево
  - выравнивание частичных выравниваний

Мудрость

Веса существенны. Один и тот же граф с различными весами на рёбрах породит разные оптимальные выравнивания.

Распознавание генов

Ген – последовательность, разбитая на экзоны и интроны.

Границы между ними – это ***донорные сайты*** *сплайсинга* (экзон-интрон) и ***акцепторные сайты*** *сплайсинга* (интрон-экзон).

Каждому потенциальному экзону (А-Д) или интрону (Д-А) присвоен ***вес***, отражающий его статистическую похожесть на типичные кодирующие и некодирующие последовательности.

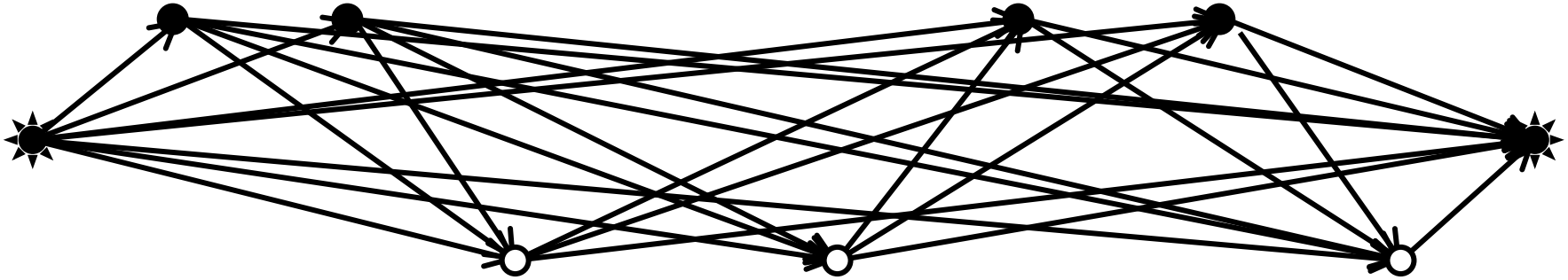
Вес гена – сумма весов составляющих его экзонов и интронов.

Цель: по последовательности с отмеченными потенциальными донорными и акцепторными сайтами построить экзон-интронную структуру наибольшего веса.

Построим граф

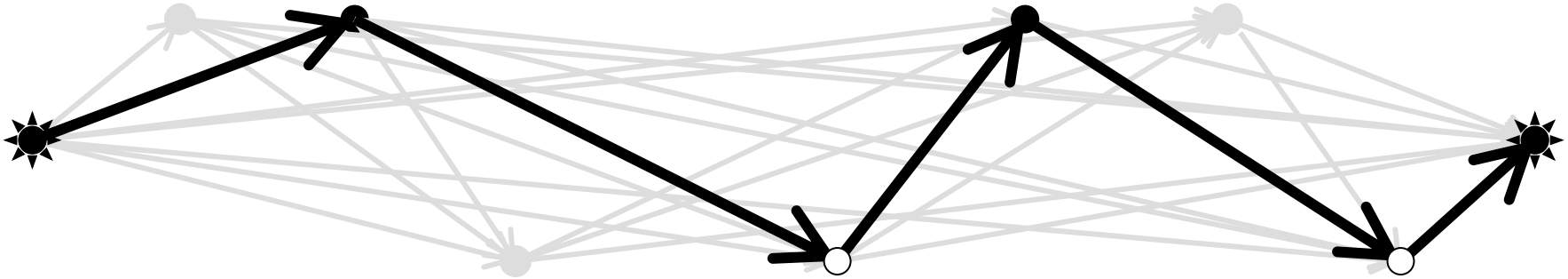
(a)

actg**ag**actgc**ag**acggac**gt**acggcactgac**gt**ata**ag**ccccac**ag**tccttac**gt**ctga



(b)

actgagactgc**ag**ACGGACGTACGGCACTGAC**gt**ata**ag**CCCCACAGTCCTTAC**gt**ctga



Сложность

Предполагая равномерное распределение сайтов (опуская детали)

=> O(L) вершин, **O(L^2)** ребер

Можно ли лучше**?**

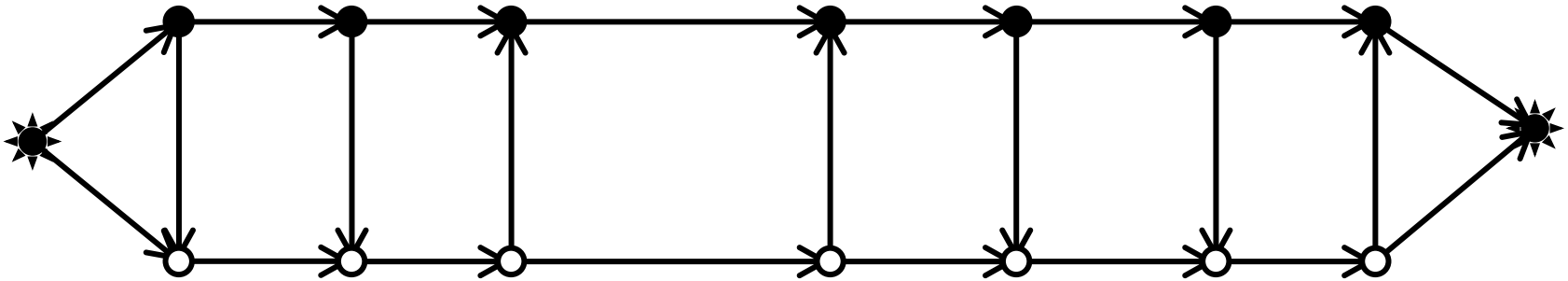
Разумно полагать, что веса сегментов аддитивны

(для экзонов мы это уже предположили).

Тогда достаточно **O(*L*)** ребер

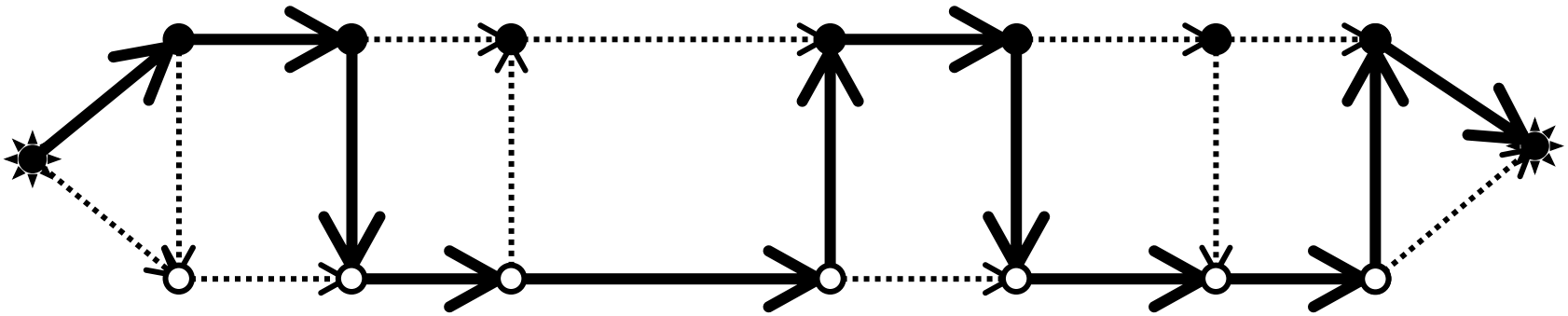
(a)

actg***ag***actgc***ag***acggac***gt***acggcactgac***gt***ata***ag***ccccac***ag***tccttac***gt***ctga



(b)

actgagactgc***ag***ACGGACGTACGGCACTGAC***gt***ata***ag***CCCCACAGTCCTTAC*gt*ctga



Мудрость

Структура важна. Одну и ту же задачу можно представить разными графами, и концептуально более простое описание не всегда самое эффективное.

Примечание

Не все проблемы, которые можно решить динамическим программированием, допускают графовое представление. Скажем, предсказание вторичной структуры РНК требует более сложных объектов, которые называются *гиперграфы*.

Return to the toy problem

calculate $\prod_{i=1\ldots m,\, j=1\ldots n} (x_i + y_j)$

the standard trick would not work because

$x \cdot z + y \cdot z = (x + y) \cdot z$ **(before) holds, but**

$(x+z) \cdot (y+z) = x \cdot y + z$ **generally does not.**

Quiz. When $(x+z) \cdot (y+z) = x \cdot y + z$ **?**

DP, generic statement. 1. Path weights

Let $\otimes$ be the operation of calculating the path score S given arc weights W. We require $(x \otimes y) \otimes z = x \otimes (y \otimes z)$.

Hence we can simply write $a \otimes b \otimes c$.

The path weight (former $S(P) = \sum_{a \in P} W(a)$) becomes $\otimes_{a \in P} W(a)$.

DP, generic statement.
2. Graph score

Let Ψ be the set of all paths. Define associative, commutative operation of combining paths:

$$(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$$

and $x \oplus y = y \oplus x$.

The graph score is defined as

$$\Omega = \oplus_{P \in \Psi} S(P) = \oplus_{P \in \Psi} \otimes_{a \in P} W(a)$$

(for the optimal path problem

$\Omega = \max_{P \in \Psi} S(P)$).

DP, generic statement. 3. Transitivity

To use dynamic programming, we need the distribution law

$$(x \otimes z) \oplus (y \otimes z) = (x \oplus y) \otimes z$$

and $(x \otimes y) \oplus (x \otimes z) = x \otimes (y \oplus z)$ **.**

This is a generalization of the property used for calculating the optimal path:

$\max(x + z, y + z) = \max(x, y) + z$.

DP, algorithm

Data types:

vertices: v, u, $Source$, $Sink$;

arcs: (v,u);

weight of arc (v,u): $W(v,u)$;

the current score of vertex v: $S(v)$;

Initialize: **for** each vertex v: $S(v)$:= undefined;

Forward process: **while** There are unprocessed vertices:

v := arbitrary unprocessed vertex with all incoming arcs processed;

for each arc (v,u): // consider all arcs starting at v

$S(u) := S(u) \oplus (S(v) \otimes W(v,u))$; // update the score of v

(v,u) := processed_arc;

endfor;

v := processed_vertex;

endwhile.

Output $S(Sink)$.

Problem (physics of polymers)

Linear polymer chain of $L+1$ monomers $k = 0, \ldots, L$.

Each monomer assumes N states $\sigma(k) \in \{\sigma_i \mid i = 1, \ldots, N\}$.

Energy of interactions between adjacent monomers is defined by an $N \times N$ matrix $\xi(\sigma_i,\sigma_j)$ (measured in the KT units).

Chain conformation P is defined by the states of the monomers $\{\sigma(0), \sigma(1), \ldots, \sigma(L)\}$.

Exponent of energy: $S(P) = \exp(-E(P)) =$
$= \prod_{k=1\ldots L} \exp(-\xi(\sigma(k-1),\sigma(k)))$.

Ψ is the set of all conformations.

Calculate *the partition function* of the set of all conformations $\Omega = \sum_{P \in \Psi} S(P)$.

Graph construction and reduction to DP

Vertices correspond to monomer states, so that their number is $(L+1)\cdot N+2$ (two additional vertices are the source and the sink, corresponding to the virtual start and end of the chain).

Arcs link vertices corresponding to adjacent monomers.

Arc weights are the interaction energies.

Paths through this graph exactly correspond to the chain conformations.

$\otimes$ is ordinary multiplication, and  is addition

The path score is the product of arc weights.

The total graph score is the sum of these products.

Standard DP solves the problem.

Quiz

(a) How many operations shall we need?

(b) How many operations shall we need if we calculate the partition function directly?

(c) Provide an algorithm for calculating the number of paths in a graph. Hint: invent suitable arc weights and reduce to the previous problem.

(d) What will Ω be if both $\otimes$ and $\oplus$ are the operation of taking the maximum?

Problem

Calculate the minimum energy and the number of conformations with the minimum energy.

Arc weights are pairs [1, ξ], with ξ as defined previously.

Path scores are pars [*n*, ε], where ε is the energy, and *n* is the number of conformations having this energy.

When two systems are combined, the resulting energy is the sum of the systems' energies, whereas the number of states is the product of the numbers of states. Hence

$$[n_1, \varepsilon_1] \otimes [n_2, \varepsilon_2] = [n_1 \cdot n_2, \varepsilon_1 + \varepsilon_2].$$

$$[n_1, \varepsilon_1] \oplus [n_2, \varepsilon_2] = \begin{cases} [n_1, \varepsilon_1] & \text{if } \varepsilon_1 < \varepsilon_2, \\ [n_1 + n_2, \varepsilon], & \text{if } \varepsilon_1 = \varepsilon_2 = \varepsilon, \\ [n_2, \varepsilon_2], & \text{if } \varepsilon_1 > \varepsilon_2, \end{cases}$$

solves the problem.

Lesson

Generalizations are useful