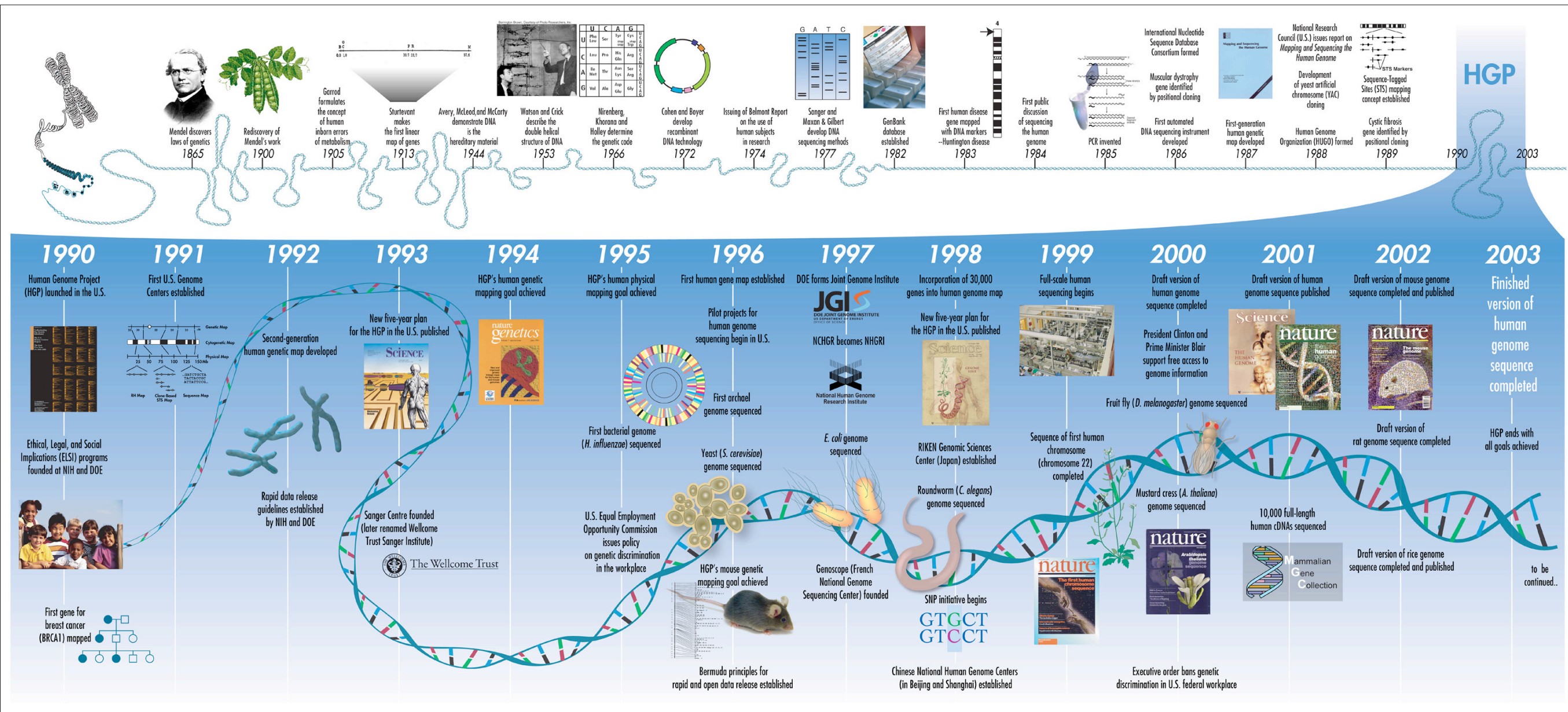


Секвенирование.
Сборка геномов.

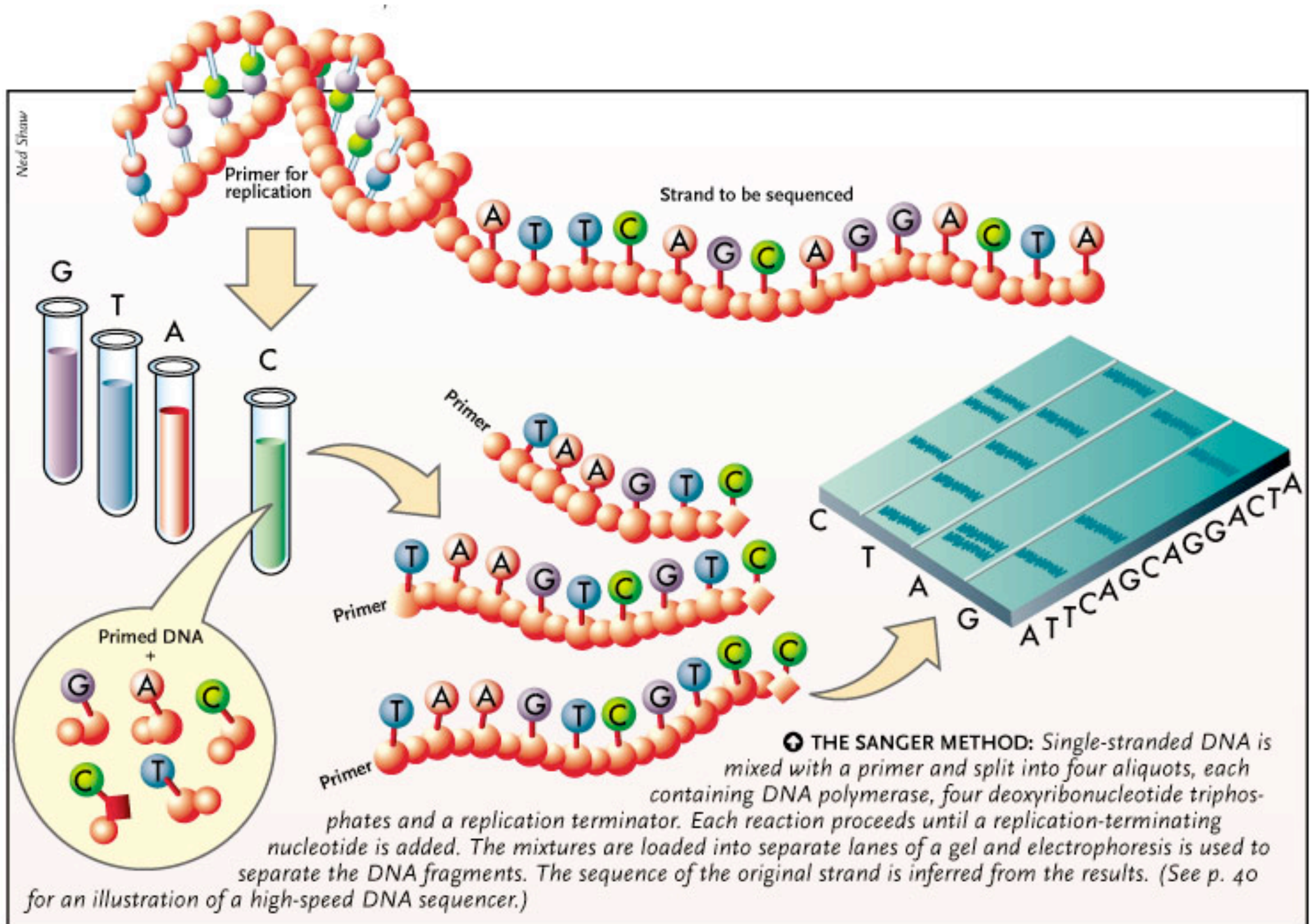
Проект “Геном человека”



Технологии секвенирования

Технологии секвенирования первого поколения	Технологии секвенирования второго поколения	Технологии секвенирования третьего поколения
Секвенирование по Сэнгеру	Roche 454	Pac Bio SMRT
	Illumina Genome Analyser (Solexa)	Oxford Nanopore
	Applied Biosystems SOLiD	
	HeliScope	

Секвенирование по Сэнгеру



Секвенирование по Сэнгеру

www.sector3.iitp.ru/video/Sanger_Sequencing.mov

Технологии секвенирования второго поколения - Roche 454: пиросеквенирование

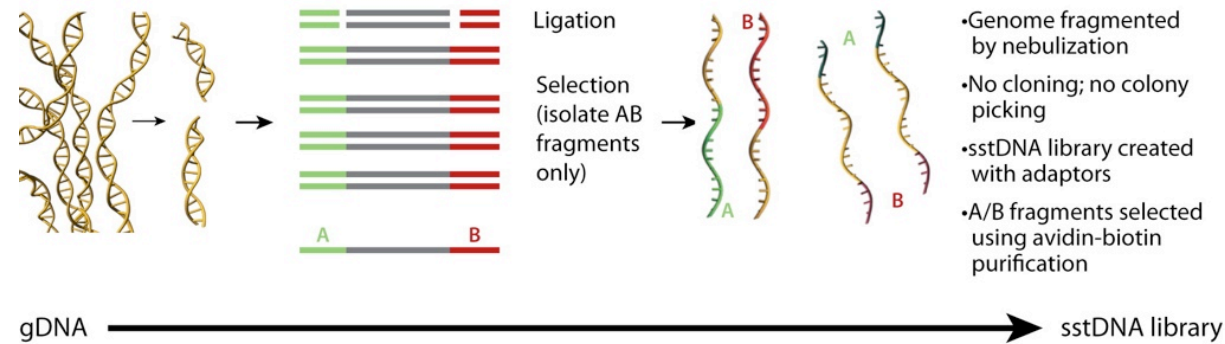
www.sector3.iitp.ru/video/pyrosequencing.mov

Технологии секвенирования второго поколения: Roche 454

a

DNA library preparation

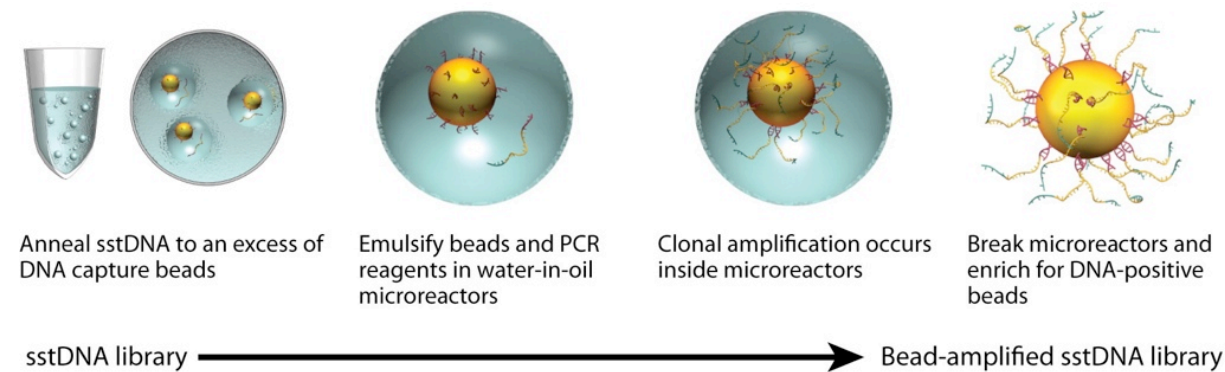
4.5 hours



b

Emulsion PCR

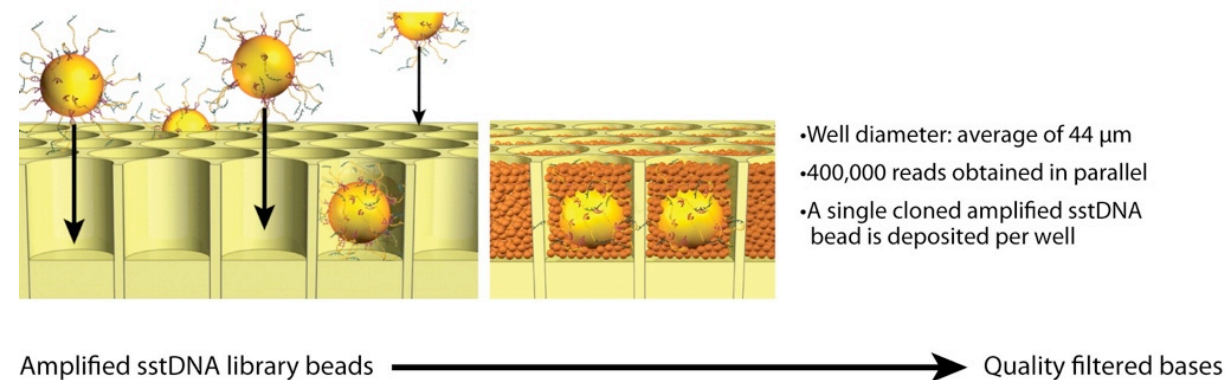
8 hours



c

Sequencing

7.5 hours

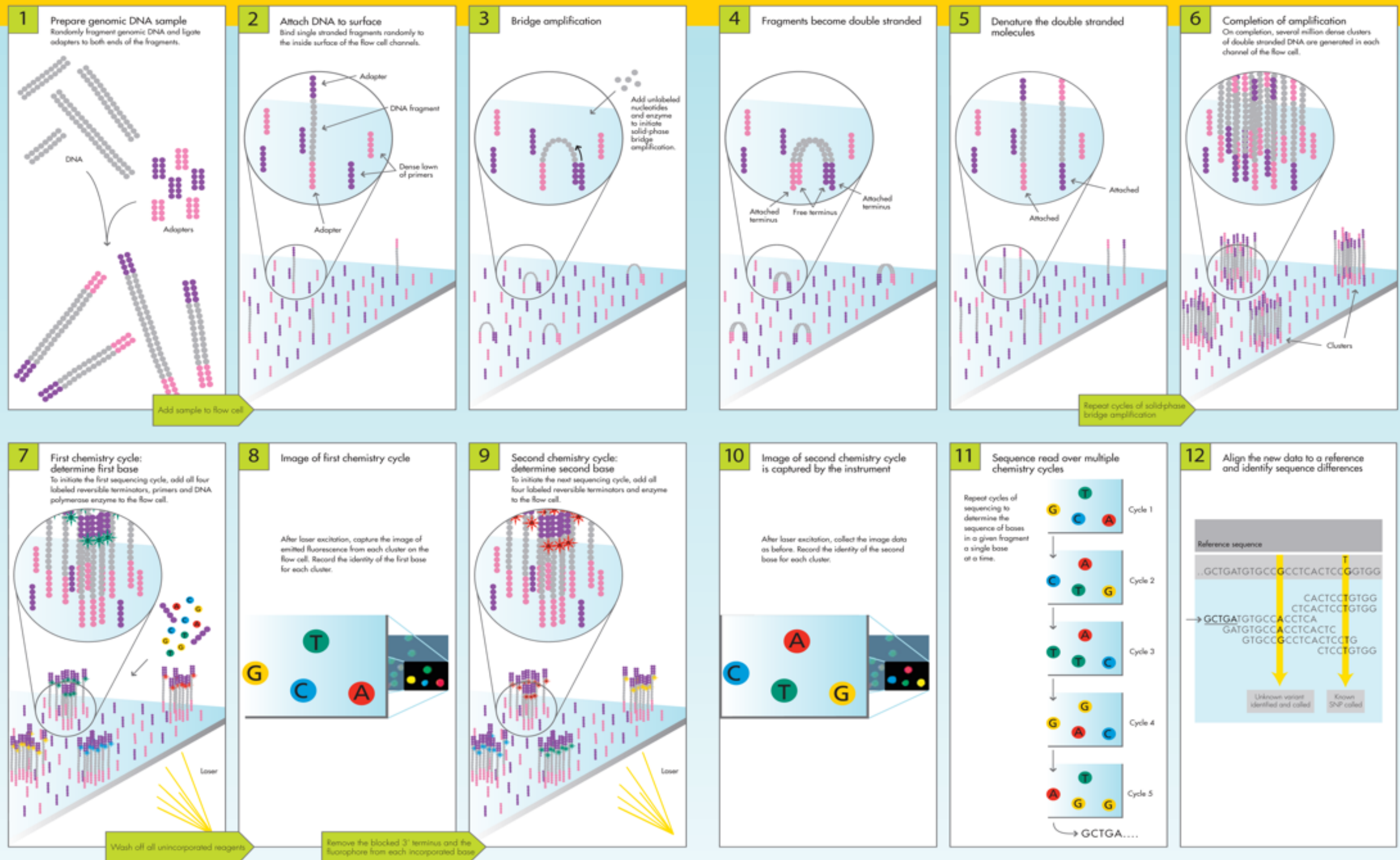


Технологии секвенирования второго поколения: Roche 454

www.sector3.iitp.ru/video/454.mov

Технологии секвенирования второго поколения: Illumina (Solexa)

Sequencing Technology Overview



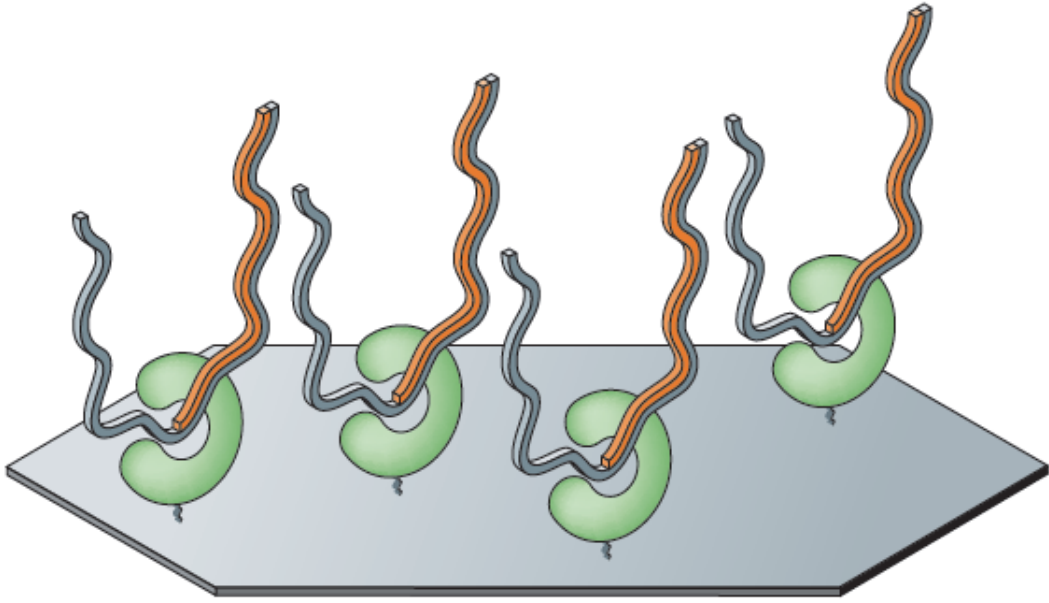
Технологии секвенирования второго поколения: Illumina (Solexa)

www.sector3.iitp.ru/video/solexa.mov

Сравнение методов секвенирования второго поколения

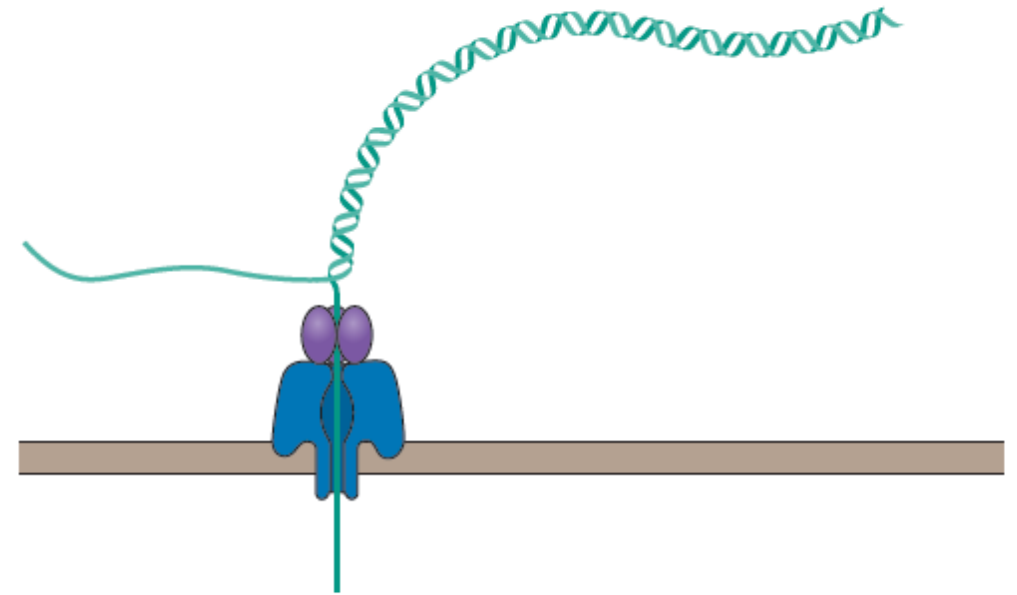
Метод	Метод амплификации	Тип секвенирования	Стоимость 1K нуклеотидов	Стоимость оборудования	Преобладающие ошибки	Длина фрагмента
454	Эмульсионный ПЦР	Синтез (пиросеквенирование)	~\$60	\$500,000	Вставки/ делеции	250 bp
Solexa	Мостовой ПЦР	Синтез (обратимые терминаторы)	~\$2	\$430,000	Замены	36 bp
SOLiD	Эмульсионный ПЦР	Лигирование (октамеры с код. парами)	~\$2	\$591,000	Замены	35 bp
HeliScope	без амплификации	Синтез (асинхронное)	~\$1	\$1,350,000	Делеции	30 bp

Методы третьего поколения : single-molecule sequencing



Pac Bio SMRT sequencing

- Sequencing by synthesis
- Single-molecule templates
- Low accuracy but long read lengths

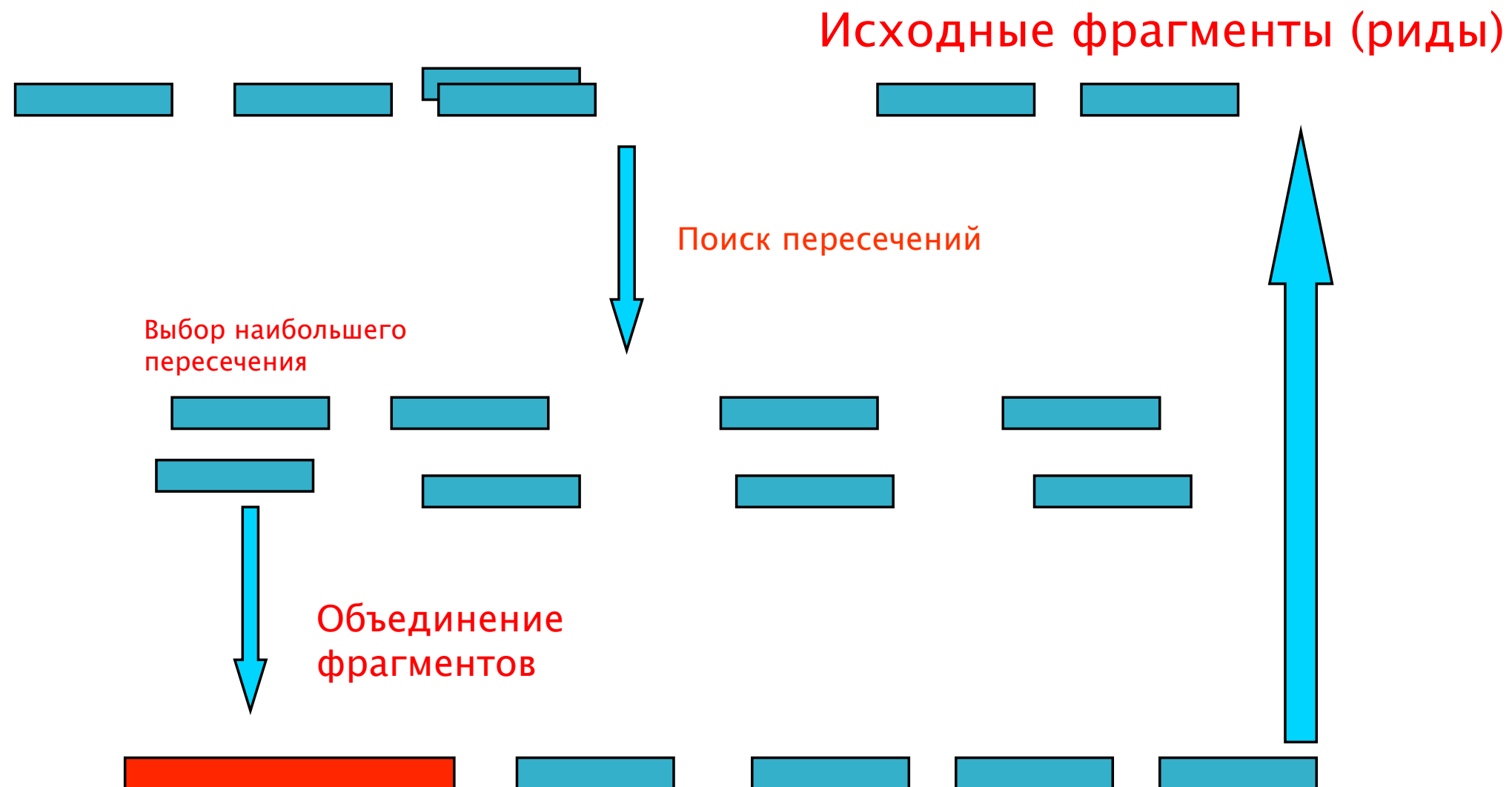


Oxford Nanopore sequencing

- Nanopore sequencing
- Single-molecule templates
- Low accuracy but long read lengths

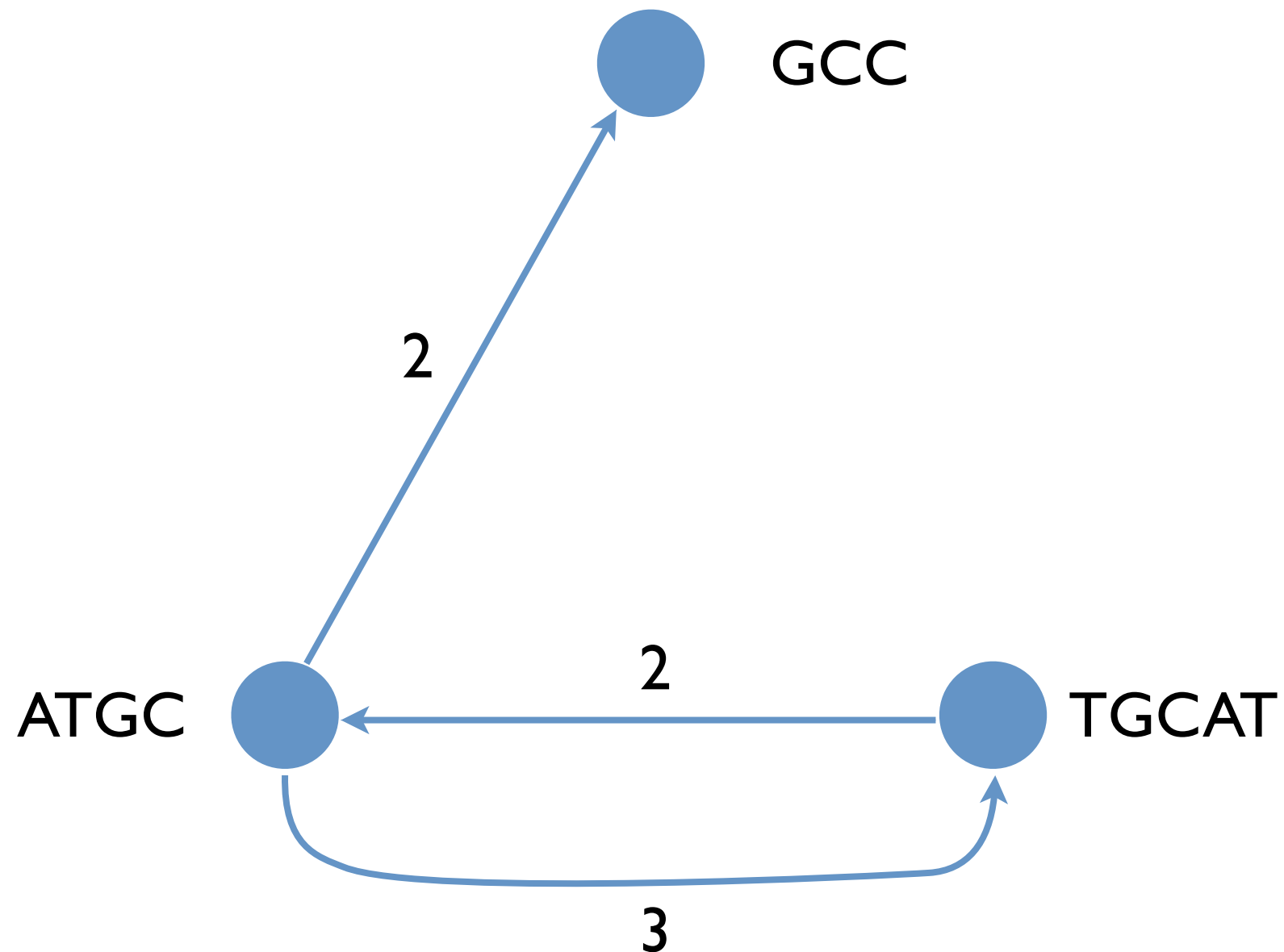
Сборка геномов: “жадные” (greedy) алгоритмы

1. Найти пересечение фрагментов.
2. Выбрать два фрагмента с наибольшим пересечением.
3. Объединить выбранные фрагменты.
4. Повторить пункты 2 и 3 до объединения всех фрагментов.



Сборка геномов: “жадные” алгоритмы

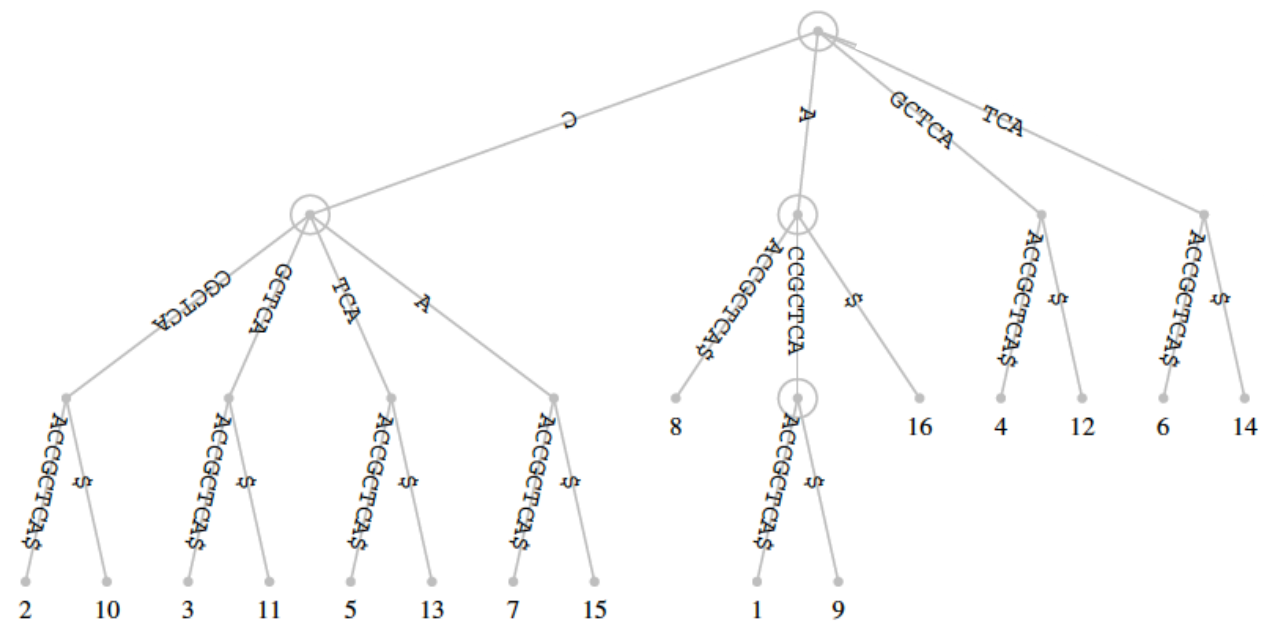
- Пример, когда жадный алгоритм дает сбой



Сборка геномов: “жадные” алгоритмы - SSAKE

- Строит хэш-таблицу, в которой ключи - риды, а значения - число встречаемости
- Хэш-таблица сортируется по значениям
- Организуется цикл обработки хэш-таблицы начиная с часто встречаемых ридов (число ридов с ошибками мало)
- Для каждого рида организуется цикл по длине суффикса n в порядке убывания длины
- Для суффикса определенной длины выполняется поиск пересечений с другими ридами в предварительно построенном префиксном дереве
- Найденный рид добавляется к текущему и удаляется из хэш-таблицы и префиксного дерева

“ATGCGTTACGAT” → 30
“GTCATGACCTGA” → 27
“CCGTATAGCTAGA” → 23
“TGACTAAGCTTAA” → 17
“CGCTGTCGTTGC” → 15
“CATTATCGATAGC” → 11



Сборка геномов: “жадные” алгоритмы - VCAKE, SHARCGS

- VCAKE

Алгоритм аналогичен SSAKE, но учитывает возможные ошибки секвенирования.

На этапе поиска перекрывающихся фрагментов по префиксному дереву выбирается не один лучший фрагмент, а все найденные. Далее текущий фрагмент продлевается по одному нуклеотиду, если доля данного нуклеотида в найденных фрагментах велика (превышает порог). Иначе, процесс увеличения текущего фрагмента прекращается.

- SHARCGS

Справляется с ошибками секвенирования с помощью двух этапов фильтрации.

На первом этапе отфильтровывает фрагменты встречающиеся меньше определенного числа раз (порог).

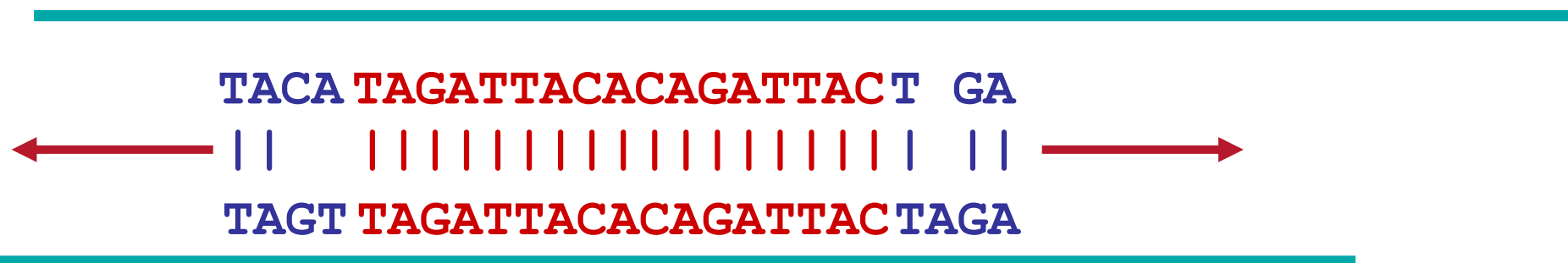
На втором этапе ищет точные пересечения с другими фрагментами обоих концов текущего фрагмента. Фрагмент удаляется если такие пересечения не найдены.

Алгоритмы пересечения-расположения-консунсуса (Overlap-layout-consensus algorithms)

- Шаг 1: поиск пересекающихся (overlapping) фрагментов

Найти пары фрагментов, имеющих общие k-меры (строки длины k, обычно $k \sim 24$).

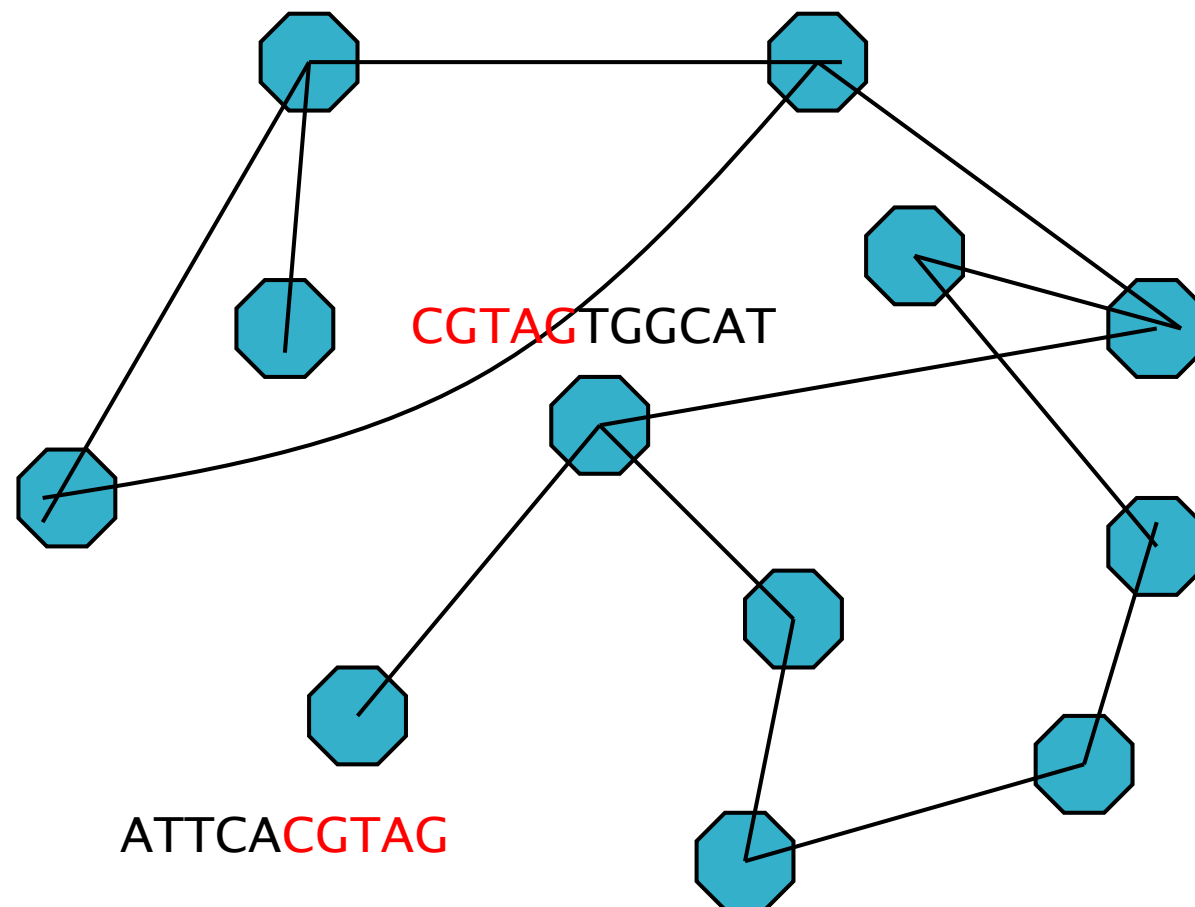
Увеличить область пересечения путем выравнивания строк в обе стороны. Прекратить процесс если сходство строк падает ниже определенного порога (<95% сходства).



Алгоритмы пересечения-расположения-консунсуса (Overlap-layout-consensus algorithms)

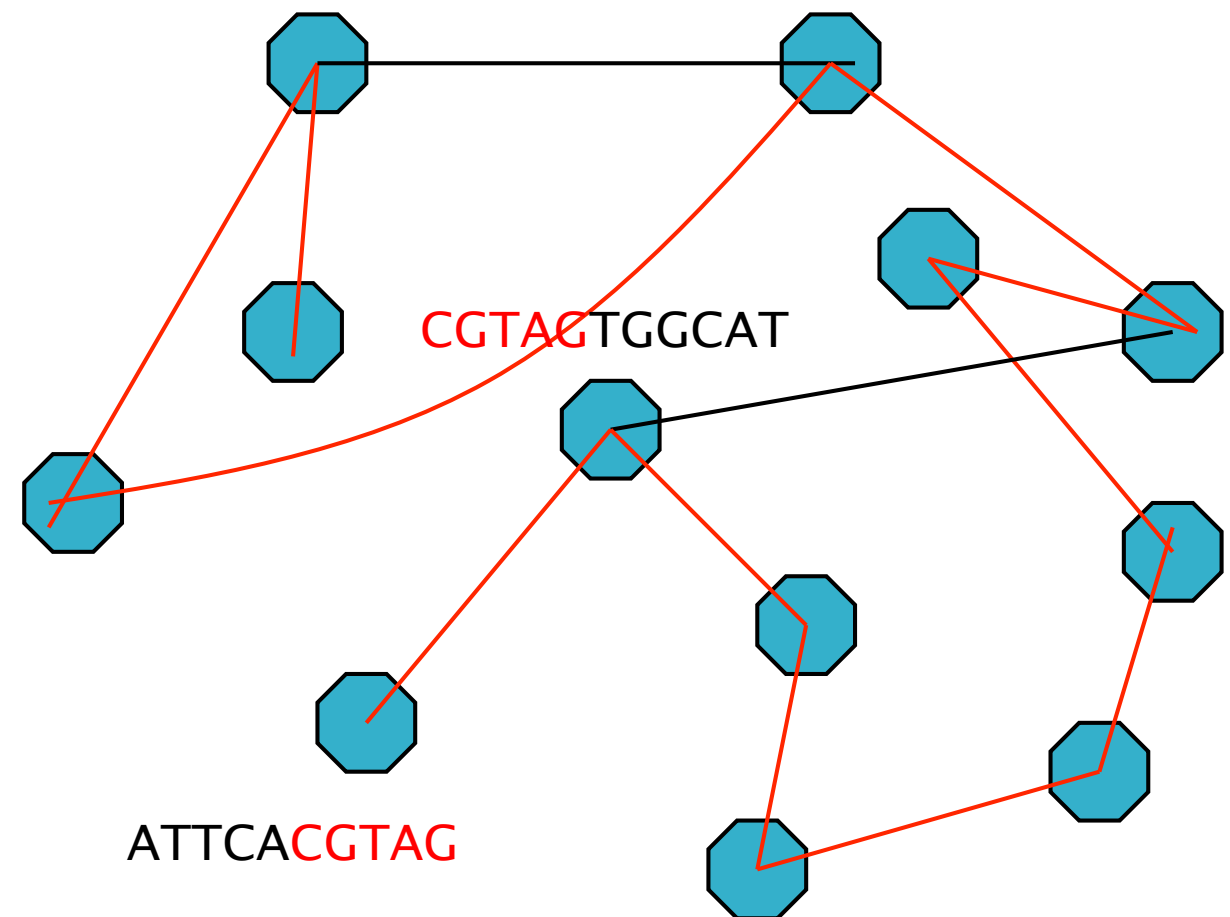
- Шаг 2: построение графа, поиск контигов - последовательных участков генома, образованных пересекающимися ридом без пропусков.

Узлы - фрагменты генома (риды). Ребра - пересечения фрагментов.



Алгоритмы пересечения-расположения-консунсуса (Overlap-layout-consensus algorithms)

- Шаг 2: построение графа поиск контигов - последовательных участков генома, образованных пересекающимися ридом без пропусков.
 - Простой путь - путь в графе, проходящий через узлы только один раз.
 - Наибольший простой путь - простой путь, который невозможно продлить.
 - Гамильтонов путь - простой путь проходящий через все вершины графа.
 - Поиск гамильтонова пути - NP-полная задача.



Алгоритмы пересечения-расположения-консунсуса (Overlap-layout-consensus algorithms)

- Шаг 3: определение консенсусной последовательности.
 - Построение множественного выравнивания по попарным выравниваниям фрагментов.
 - Корректировка ошибок.

TAGATTACACAGATTACTGA TTGATGGCGTAA СТА
TAGATTACACAGATTACTGACTTGATGGCGTAAАСТА
TAG TTACACAGATTA**T**GACTT**C**ATGGCGTAA СТА
TAGATTACACAGATTACTGACTTGATGGCGTAA СТА
TAGATTACACAGATTACTGACTTGATGGGGTAA СТА



TAGATTACACAGATTACTGACTTGATGGCGTAA СТА

Методы, использующие граф де Брейна (de Bruijn)

- Путь Эйлера - путь, проходящий один раз по каждому ребру графа.
- Существует линейный по времени алгоритм нахождения Эйлерава пути для графа.

Методы, использующие граф де Брейна (de Bruijn)

- Фрагменты (риды) разбиваются на пересекающиеся k-меры.

Пример:

фрагмент 10bp: ATTCTGACTCC

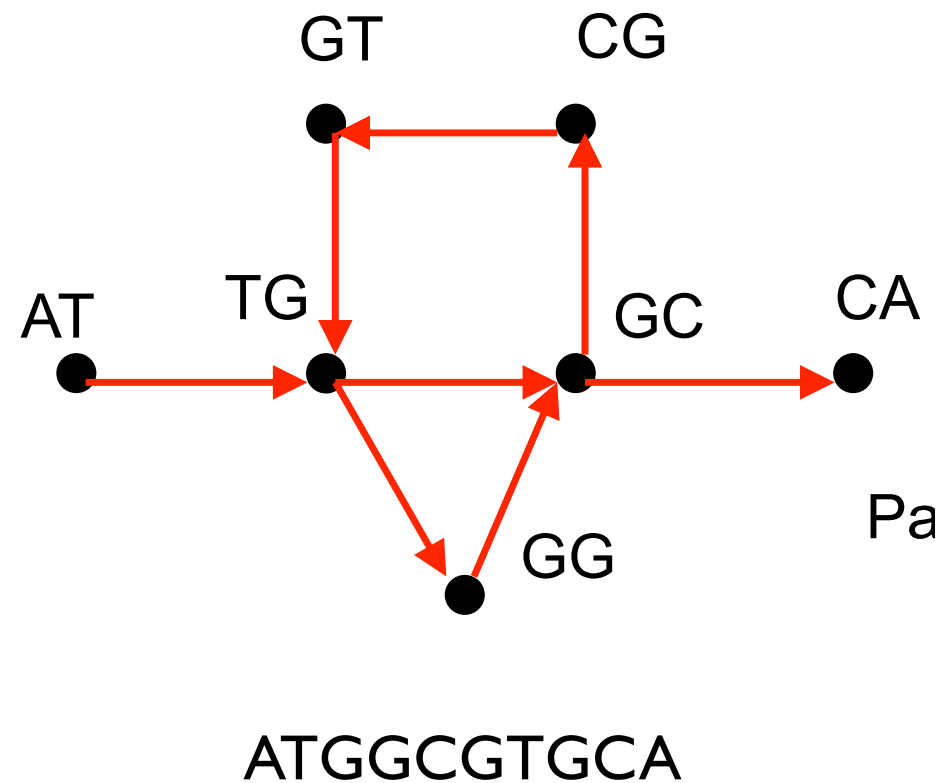
разбиение на k=5-меры: ATTCT
TCTGA
CTGAC
TGACT
GACTC
ACTCC

Методы, использующие граф де Брейна (de Bruijn)

- Построение графа.

- Узлы - (k-1)- меры
- Ребра - связывающие их k-меры.

$S = \{ \text{ATG, TGC, GTG, GGC, GCA, GCG, CGT} \}$



Инструменты сборки геномов

Тип методов сборки геномов	Программное обеспечение
“Жадные” (greedy) алгоритмы	SSAKE, VCAKE, SHARCGS
Overlap-layout-consensus	Celera, Arachne, CAP, PCAP, Newbler
Граф де Брейна	Euler, Velvet, Edena, Abyss, Allpaths, SOAPdenovo

- При подготовке слайдов использовались материалы лекций:
 - Михаила Гельфанда (ИППИ)
 - Андрея Миронова (МГУ)
 - Serafim Batzoglou (Stanford)
 - Manolis Kellis (MIT)
 - Pavel Pevzner (UCSD)