

Сборка. Часть вторая

Алгоритмы в биоинформатике

Антон Елисеев

eliseevantoncoo@gmail.com

В прошлой лекции

- Подсчет k-меров, фильтр Блума
- Задача сборки генома
- Задача SCS
- Жадное решение и overlap graph

В этой лекции

- Задача сборки генома
- Граф Де Брюина
- Сборка идеальных ридов при помощи графа Де Брюина
- Граф Де Брюина на реальных данных

Сборка геномов

Дано:

Множество ридов

Цель:

Найти геном, из которого эти риды получены

CTAGGCCCTCAATT
CTCTAGGCCCTCAATT
GGCTCTAGGCCCTCATT
CTCGGCTCTAGCCCCTCATT
TATCTCGACTCTAGGCCCTCA
TATCTCGACTCTAGGCC
TCTATATCTGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT
GGCGTCTATATCTGGCTCTAGGCCCTCATT

Сборка геномов

Дано:

Множество ридов

Цель:

Найти геном, из которого эти риды получены

CTAGGCCCTCAATT
GGCGTCTATATCT
CTCTAGGCCCTCAATT
TCTATATCTGGCTCTAGG
GGCTCTAGGCCCTCATT
CTCGGCTCTAGCCCCTCATT
TATCTCGACTCTAGGCCCTCA
GGCGTCGATATCT
TATCTCGACTCTAGGCC
GGCGTCTATATCTCG

?????????????????????????????????

Граф Де Брюина

Задача:

Задан некоторый алфавит \mathcal{A} , $|\mathcal{A}| = n$, нужно найти такую кратчайшую строку, которая содержит все возможные слова длины l над этим алфавитом.

Граф Де Брюина

Пусть $n = 2, l = 4$

Граф Де Брюина

Пусть $n = 2, l = 4$

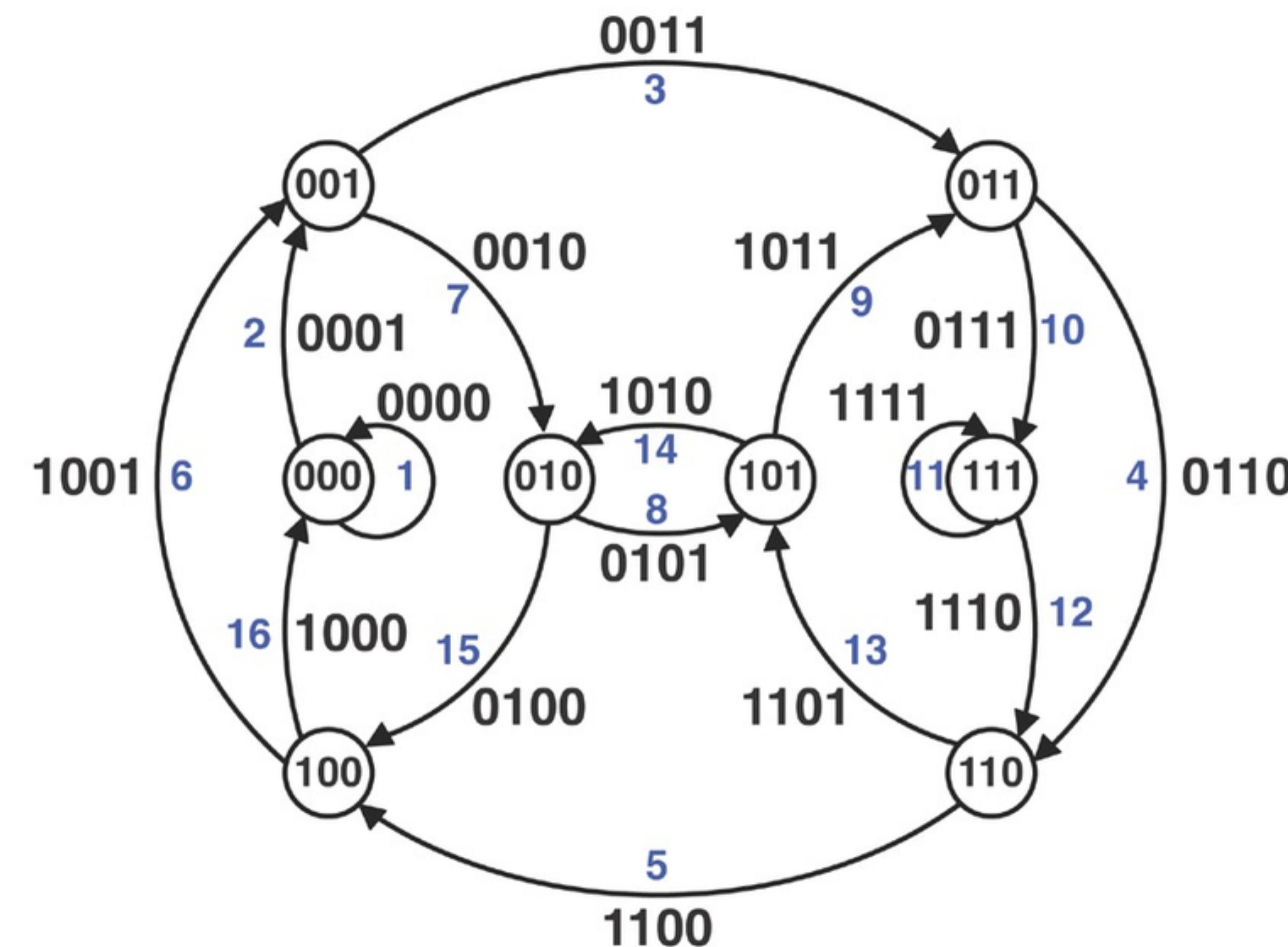
0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111

1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111

Граф Де Брюина

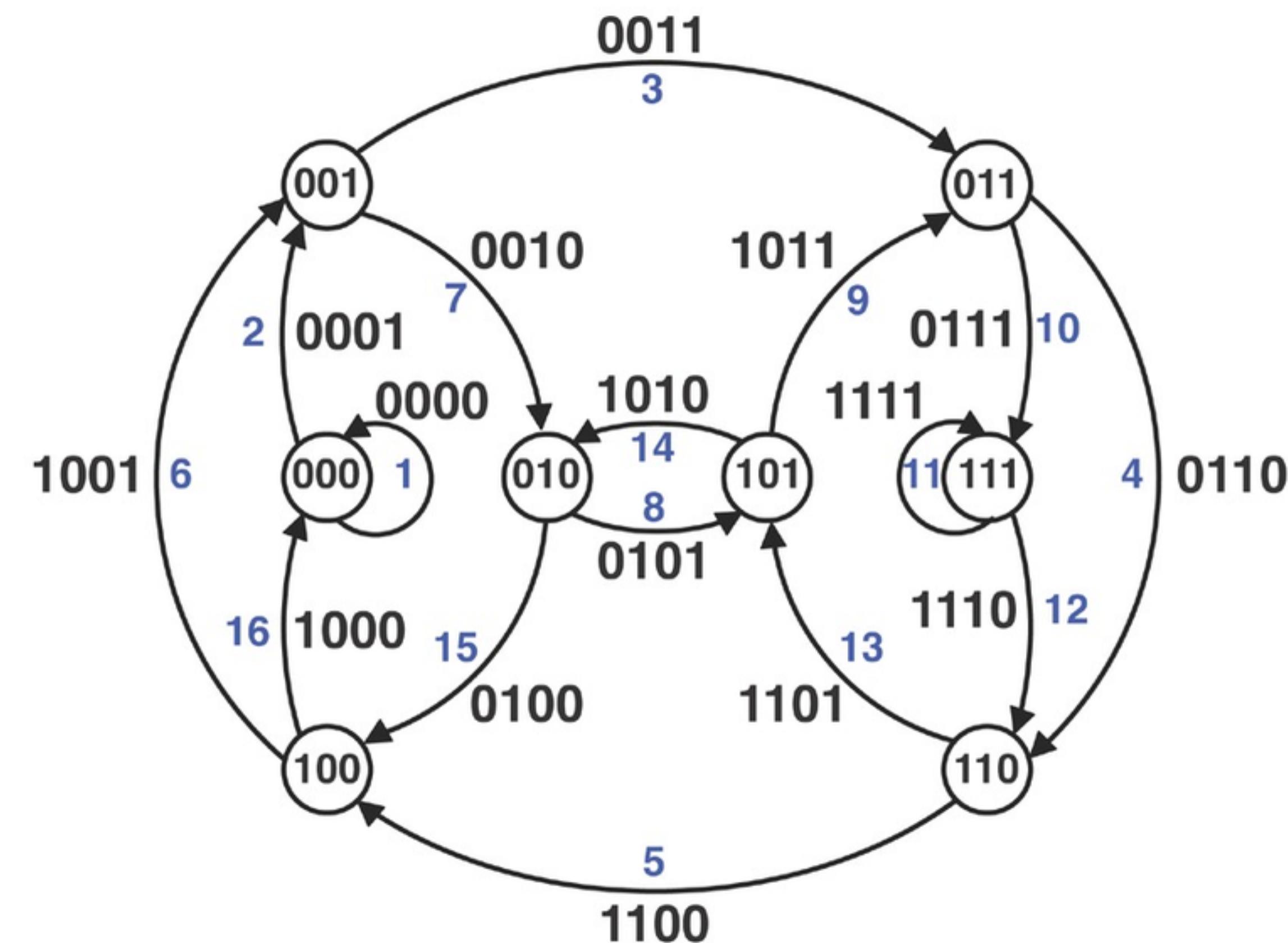
0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111

1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111



Граф Де Брюина

0000110010111101



Граф Де Брюина

- $B(n, l)$ – эйлеров
- Последовательность меньшей длины составить нельзя: в полученной последовательности ровно n^l подстрок длины l , и именно столько чисел можно составить из цифр от 1 до n .

Идеальные риды.

- Все риды из одного генома
- Однаковой длины l
- Все подстроки генома длины l встречаются в ридах
- Риды без ошибок

Идеальные риды.

TAATGCCATGGGATGTT

Идеальные риды.

TAATGCCATGGGATGTT

TAATG
AATGC
ATGCC
TGCCA
GCCAT
CCATG
CATGG
ATGGG
TGGGA
GGGAT
GGATG
GATGT
ATGTT

Идеальные риды.

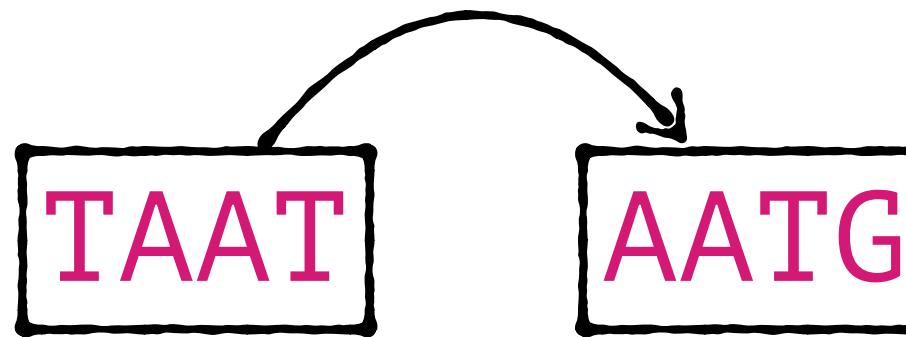
TAATGCCATGGGATGTT

TAATG, AATGC, ATGCC, TGCCA, GCCAT, CCATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT

Идеальные риды.

TAATGCCATGGGATGTT

TAATG, AATGC, ATGCC, TGCCA, GCCAT, CCATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Идеальные риды.

TAATGCCATGGGATGTT

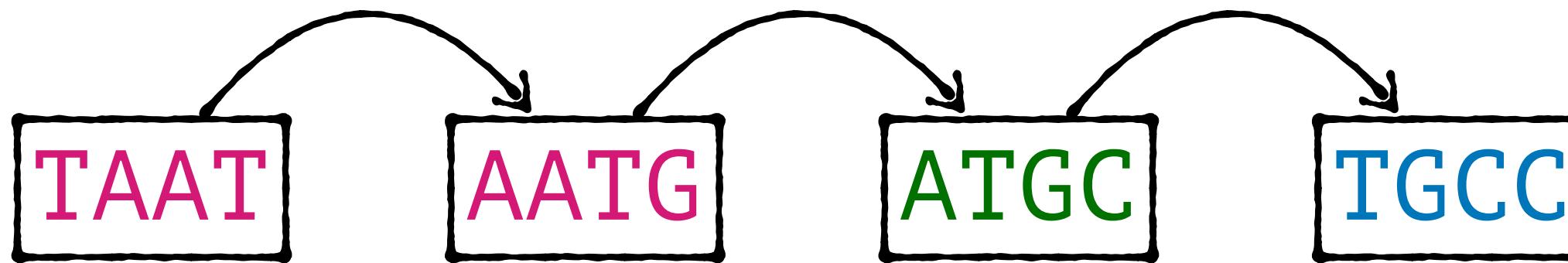
TAATG, AATGC, ATGCC, TGCCA, GCCAT, CCATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Идеальные риды.

TAATGCCATGGGATGTT

TAATG, AATGC, ATGCC, TGCCA, GCCAT, CCATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Идеальные риды.

TAATGCCATGGGATGTT

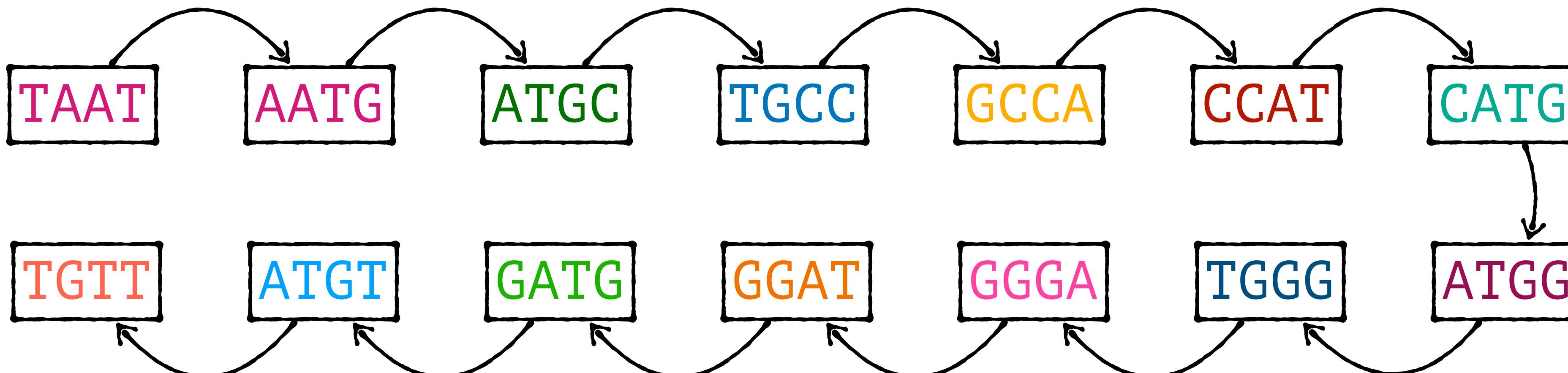
TAATG, AATGC, ATGCC, TGCCA, GCCAT, CCATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Идеальные риды.

TAATGCCATGGGATGTT

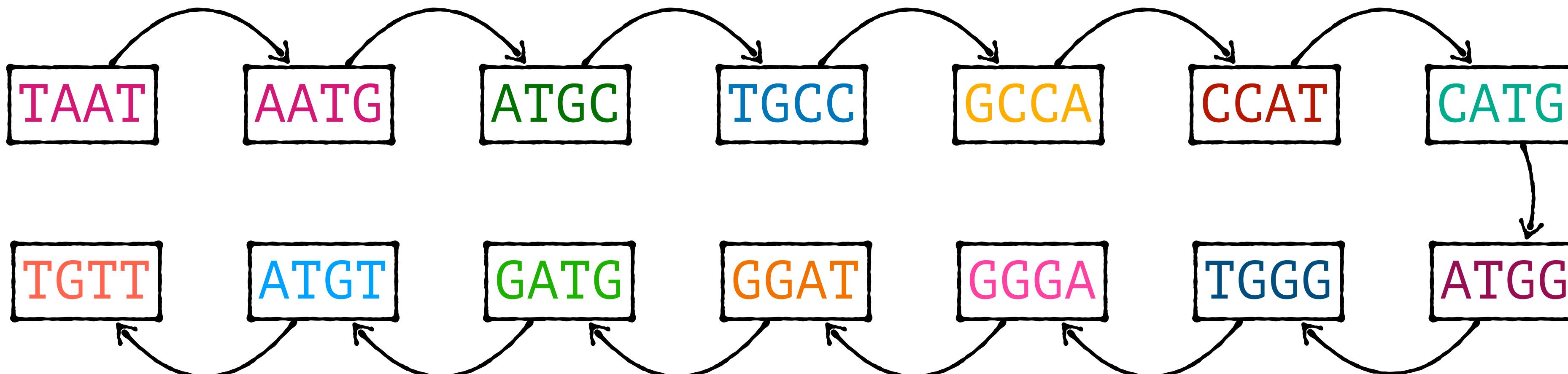
TAATG, AATGC, ATGCC, TGCCA, GCCAT, CCATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Идеальные риды.

TAATGCCATGGGATGTT

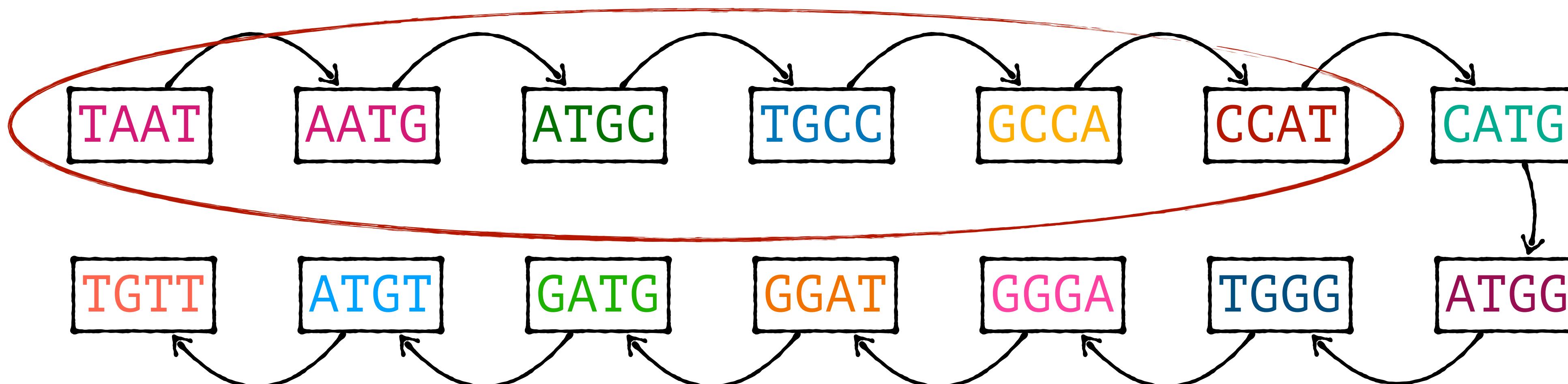
TAATGCCATGGGATGTT



Повторы.

AAAAAAACATGGGATGTT

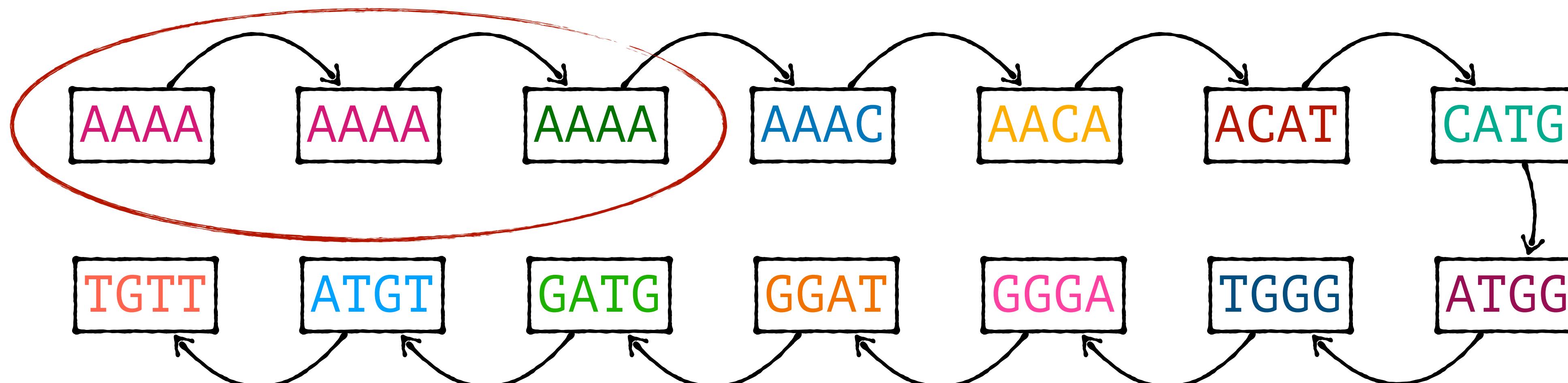
AAAAAA, AAAAAA, AAAAC, AAACA, AACAT, ACATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Повторы.

AAAAAAA CATGGGATGTT

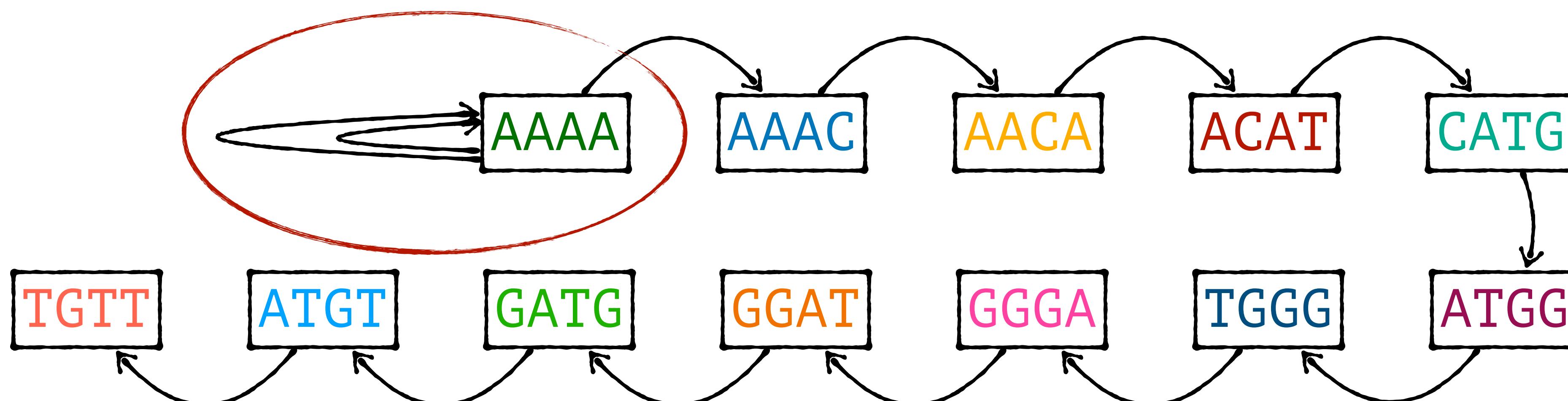
AAAAAA, AAAAA, AAAAC, AAACA, AACAT, ACATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Повторы.

AAAAAAA CATGGGATGTT

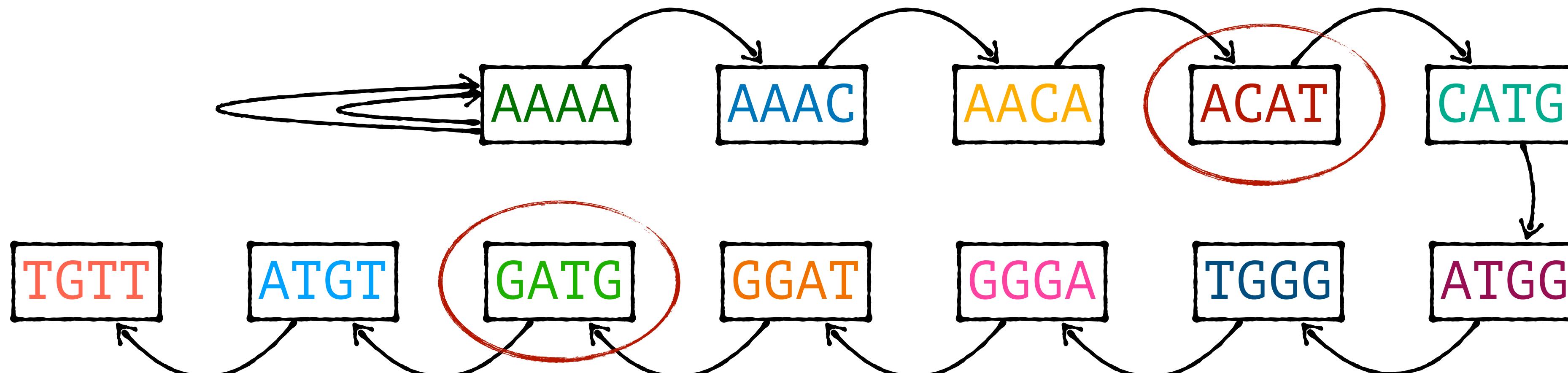
AAAAAA, AAAAA, AAAAC, AAACA, AACAT, ACATG, CATGG,
ATGGG, TGGGA, GGGAT, GGATG, GATGT, ATGTT



Неидеальное покрытие.

AAAAAAA CATGGGATGTT

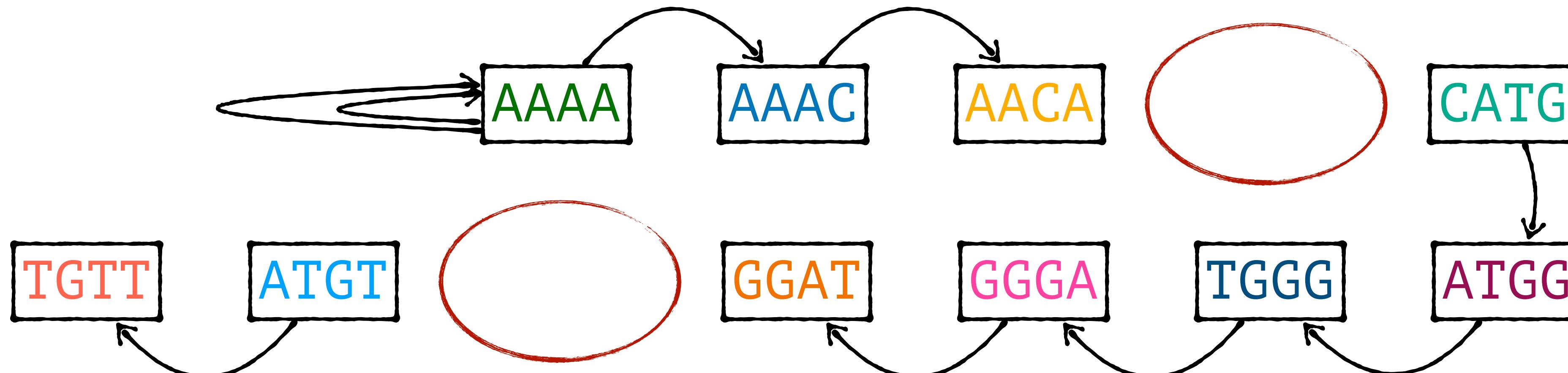
AAAAAA, AAAAA, AAAAC, AAACA,
ATGGG, TGGGA, GGGAT, , GATGT, ACATG, CATGG,
ATGTT



Неидеальное покрытие.

AAAAAAACATGGGATGT

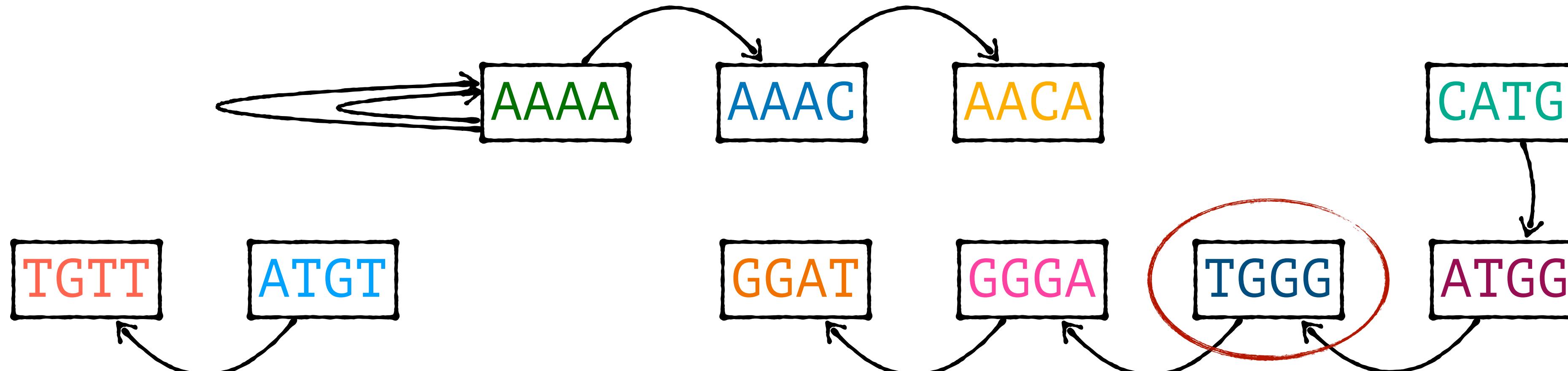
AAAAAA , AAAAAA , AAAAAC , AAACCA , ACATG , CATGG ,
ATGGGG , TGGGGA , GGGAT , GATGT , ATGTT



Ошибки.

AAAAAAA CATGGGATGTT

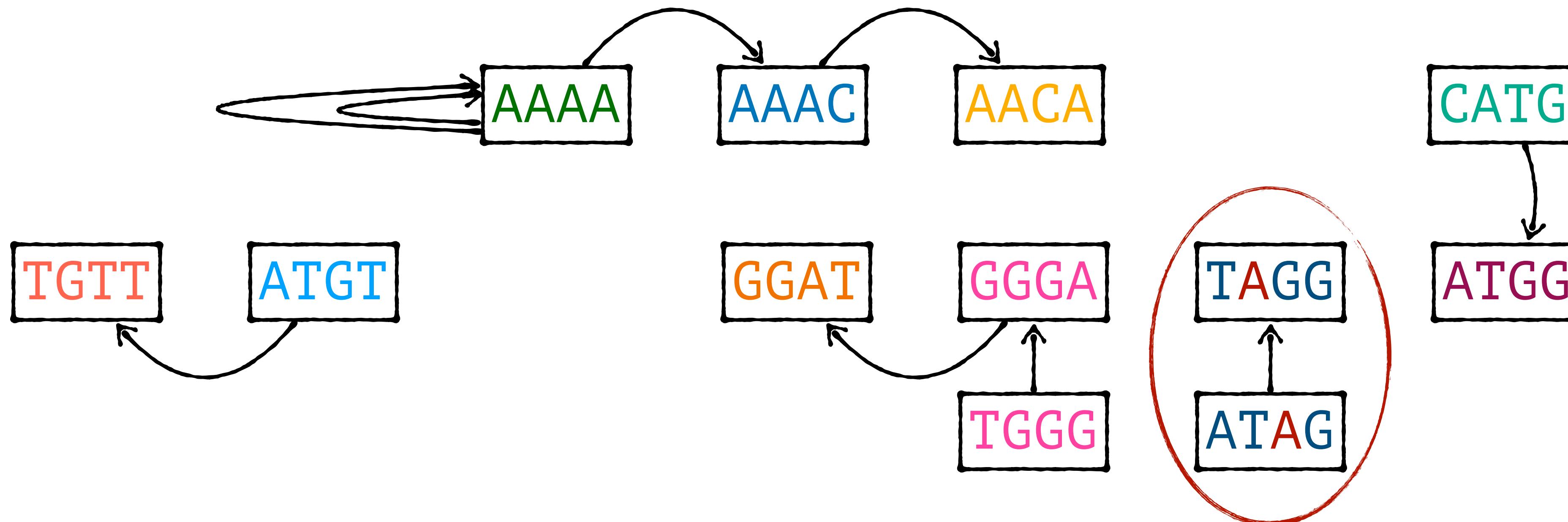
AAAAAA, AAAAA, AAAAC, AAACA,
ATAGG, TGGGA, GGGAT, , GATGT, ACATG, CATGG,
ATGTT



Ошибки.

AAAAAAA CATGGGATGTT

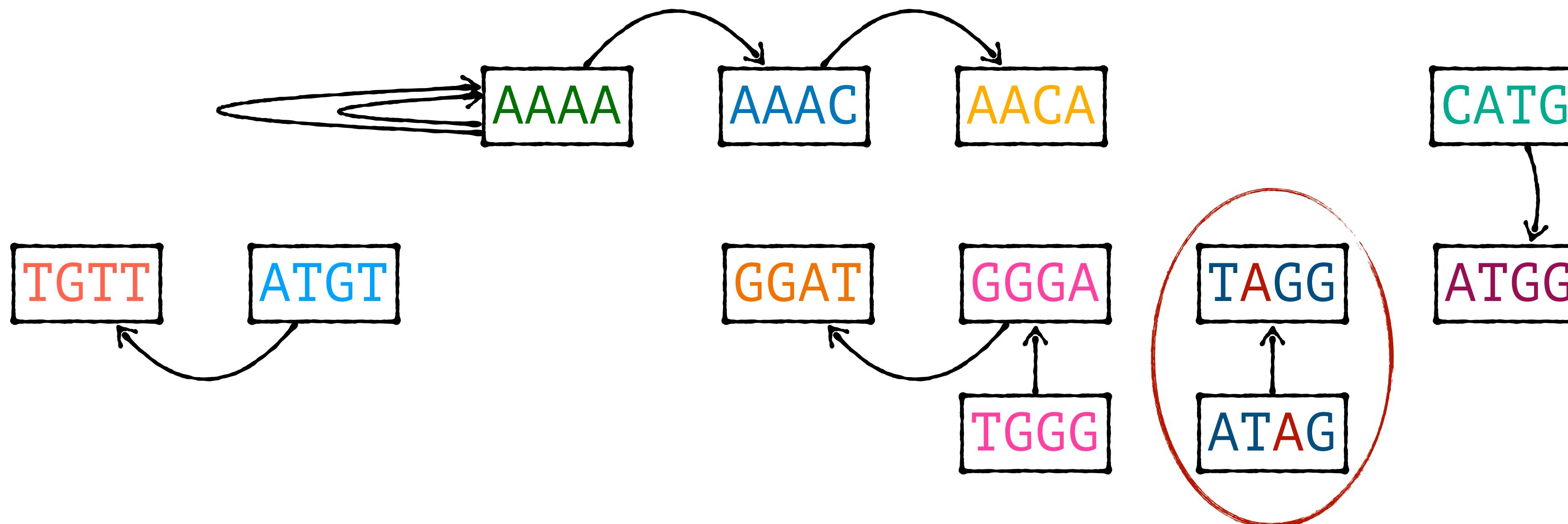
AAAAAA, AAAAA, AAAAC, AAACA,
ATAGG, TGGGA, GGGAT, , GATGT, ACATG, CATGG,
ATGTT



Сборка геномов.

AAAAAAA**CATGGGATGTT**

AAAAAA**ACA**, **CATGG**, **ATAGG**, **GGGAT**, **ATGTT**



Сборка геномов.

- Граф будем строить так:

Пусть даны риды $S = s_1, \dots, s_n$, по этому множеству построим S_k и S_{k-1} – множества k -меров встречающихся в качестве подстрок по крайней мере в одном риде.

Граф Де Брюина $B(S, k)$ в качестве вершин содержит все $k - 1$ меры, причем вершина $v \in S_{k-1}$ соединяется с $u \in S_{k-1}$ направленным ребром, если в S_k присутствует строка $v[: -1] + u[-1]$.

- Уменьшим k

Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

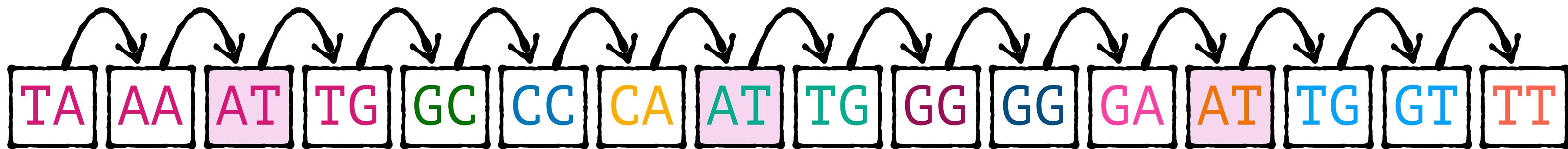
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

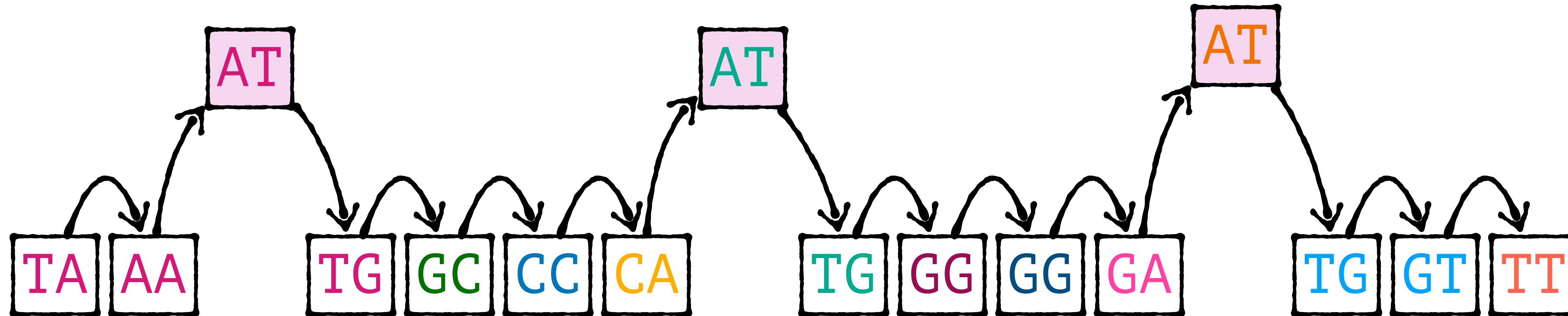
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

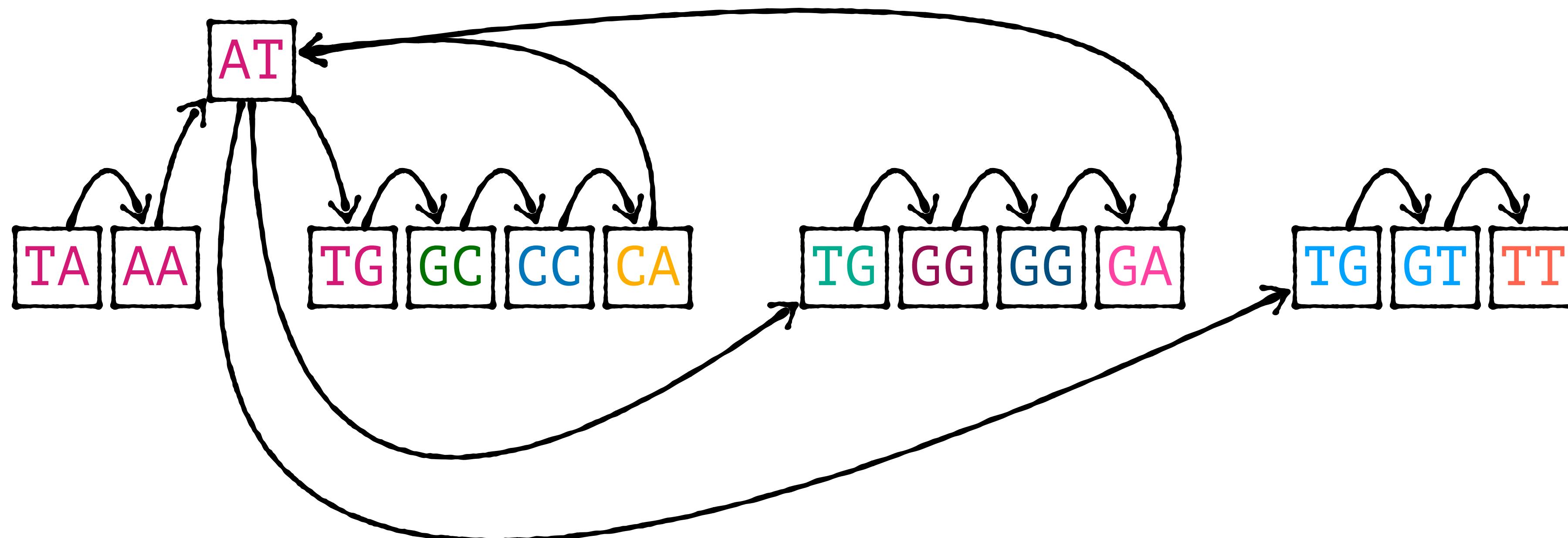
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

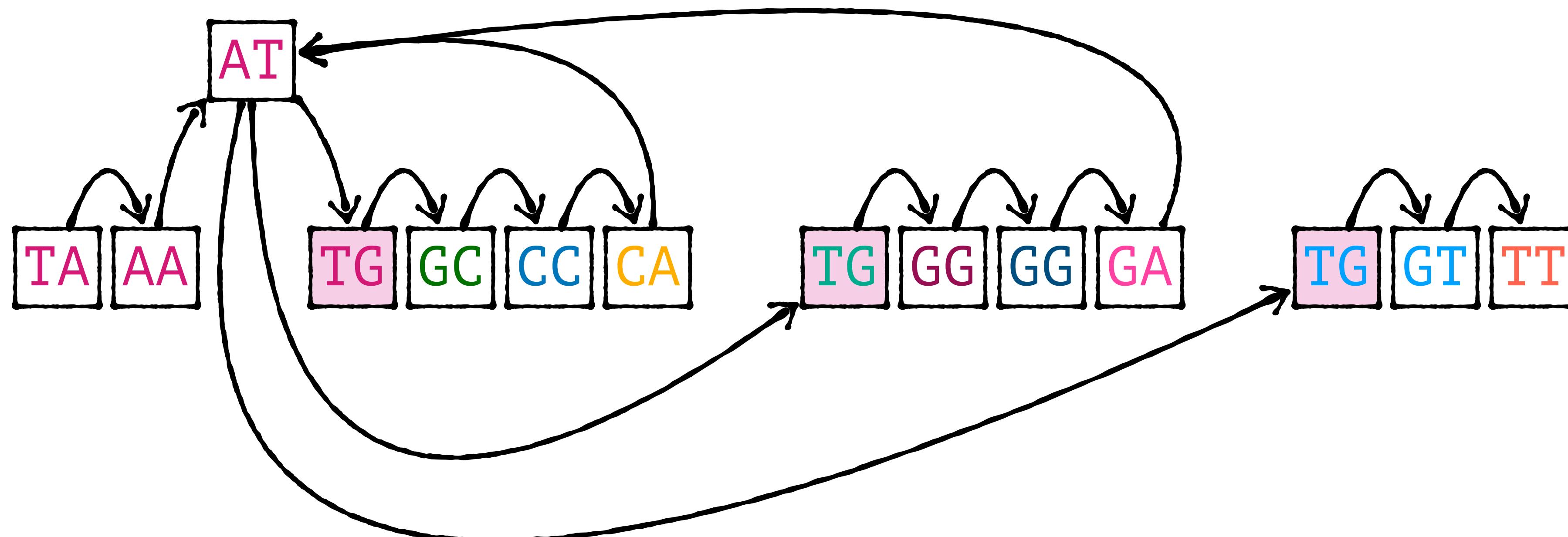
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

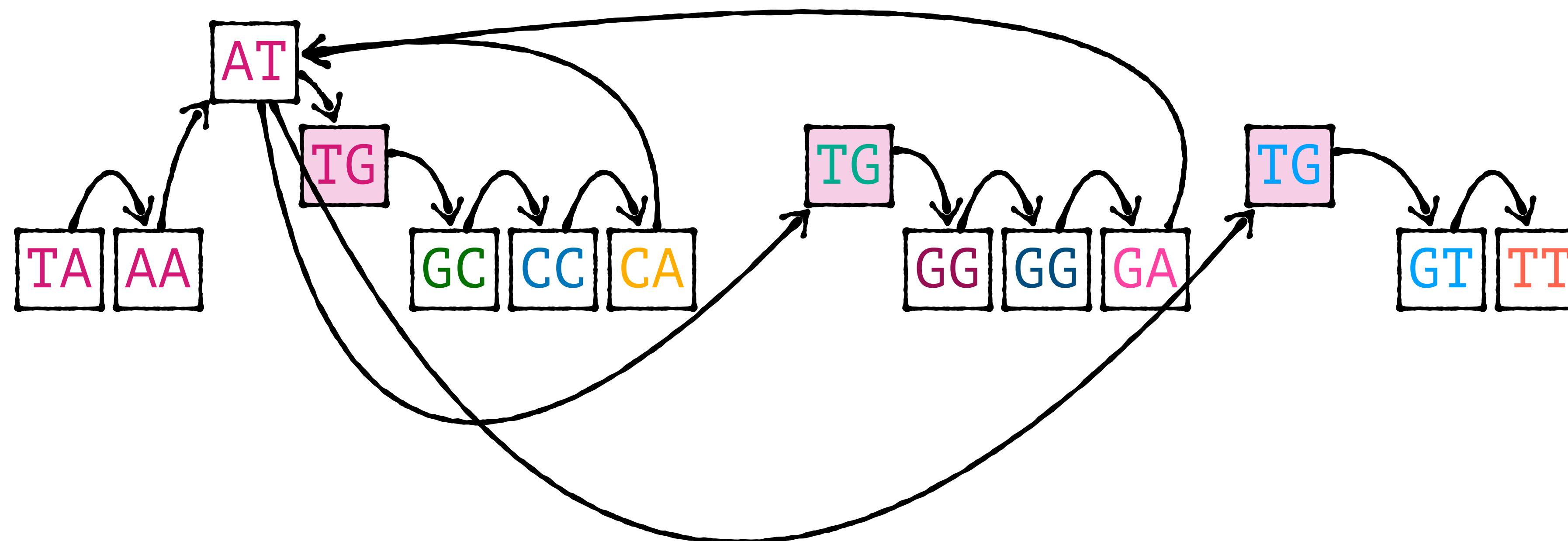
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

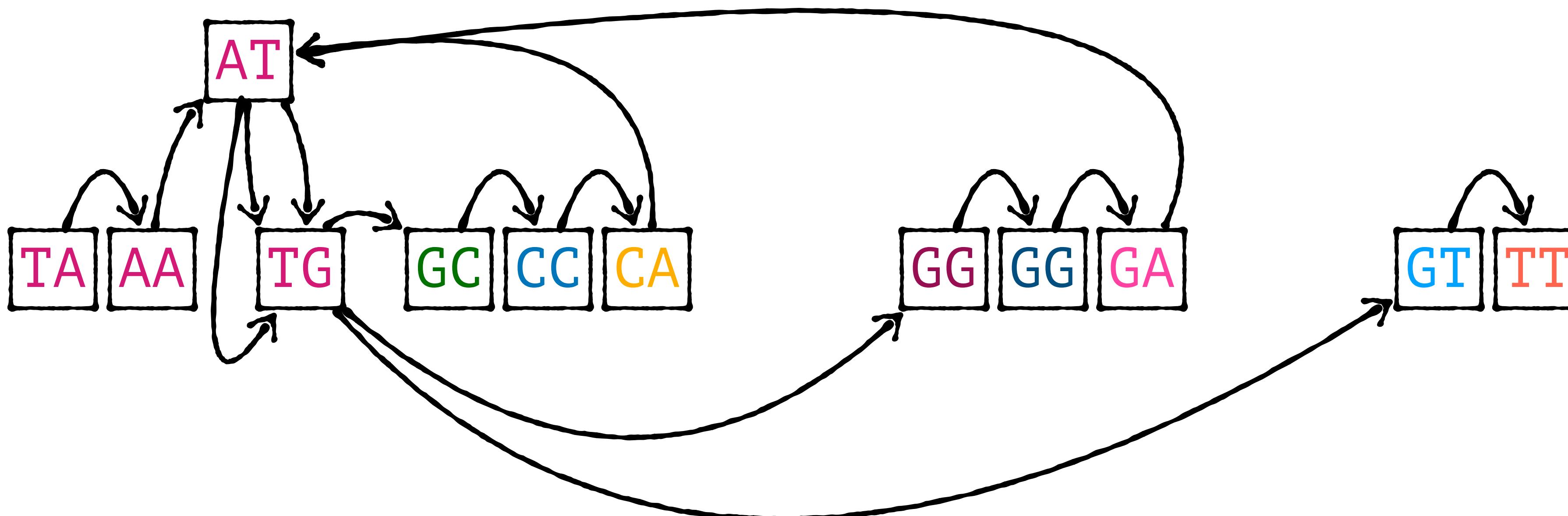
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

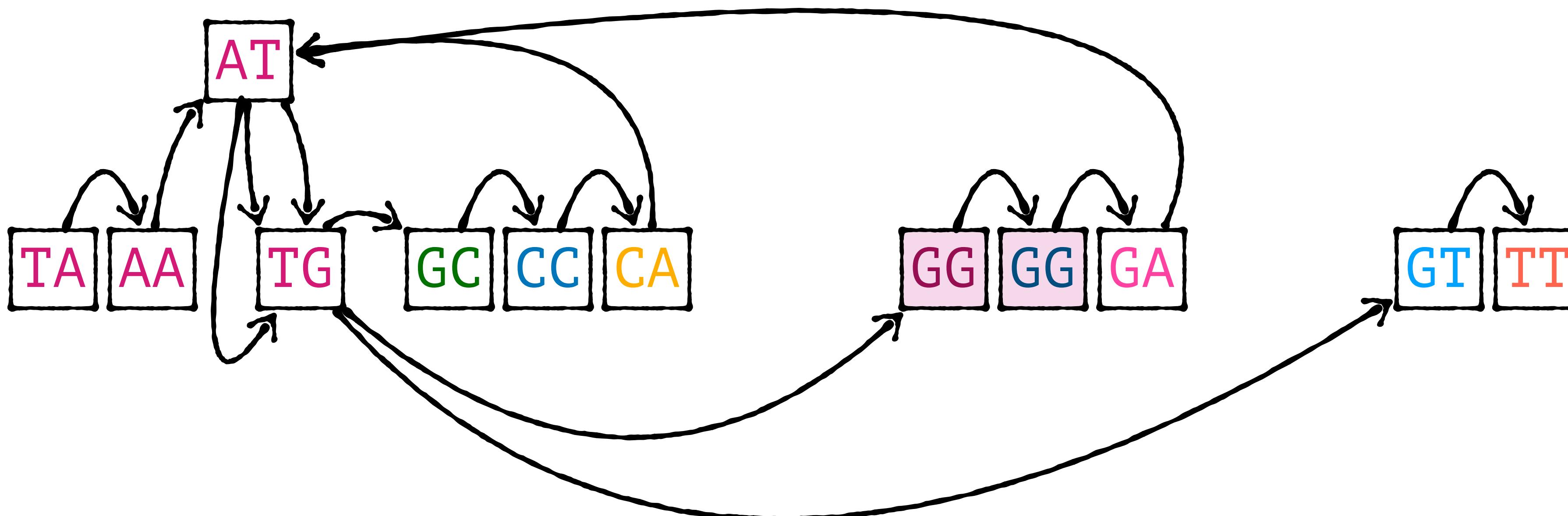
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

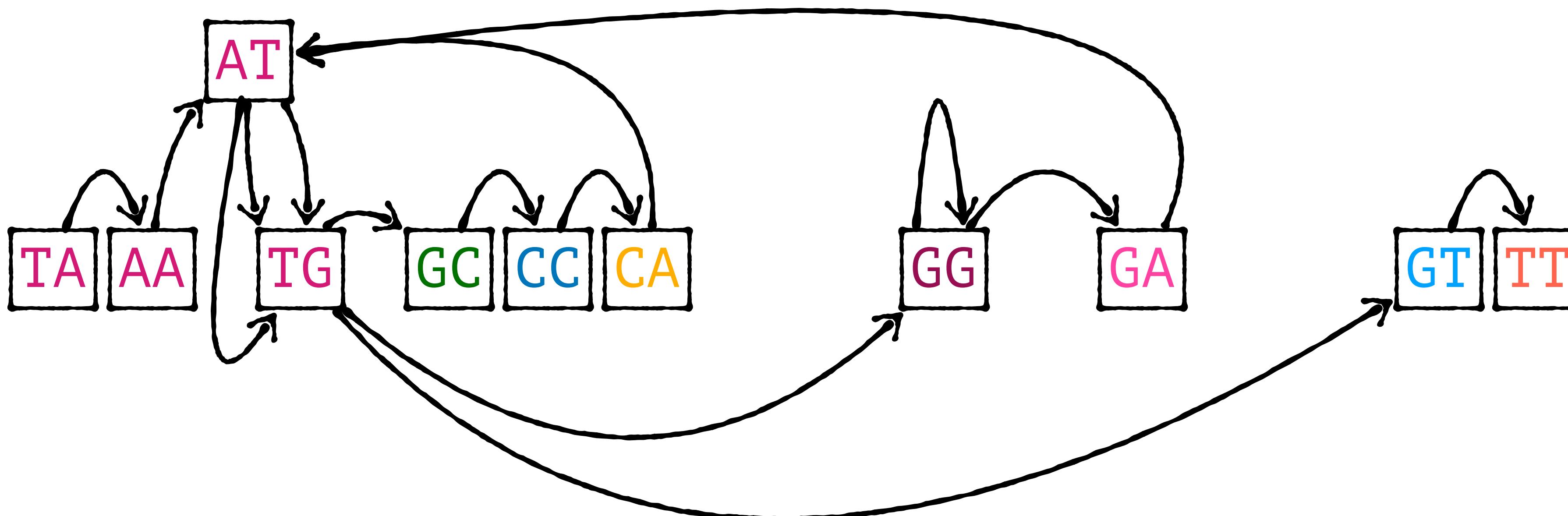
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

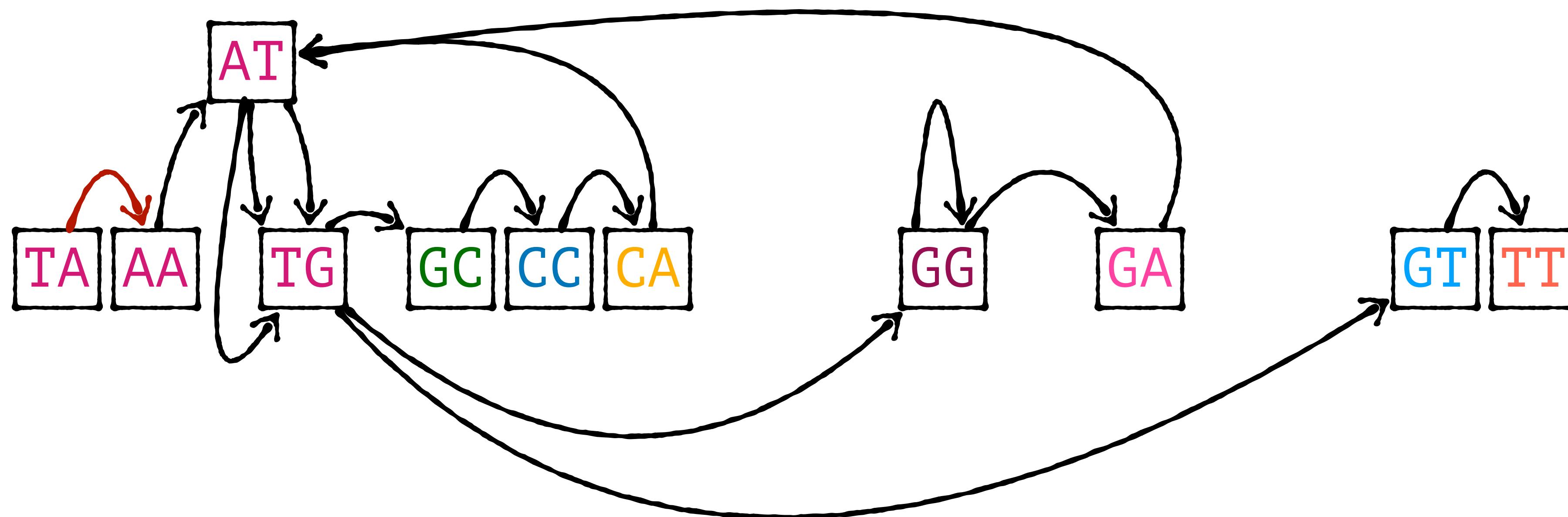
TAATG, AATGC, ATGCC, TGCCA,
ATGGG, TGGGA, GGGAT, , GATGT, CCATG, CATGG,
ATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

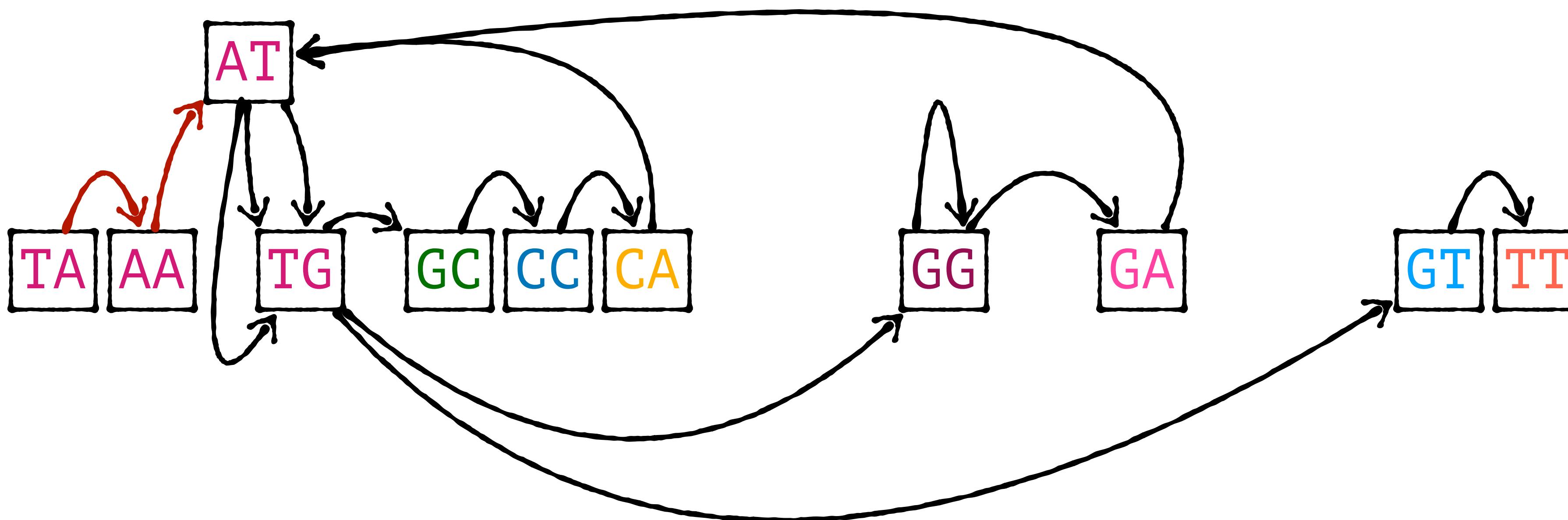
TAA



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

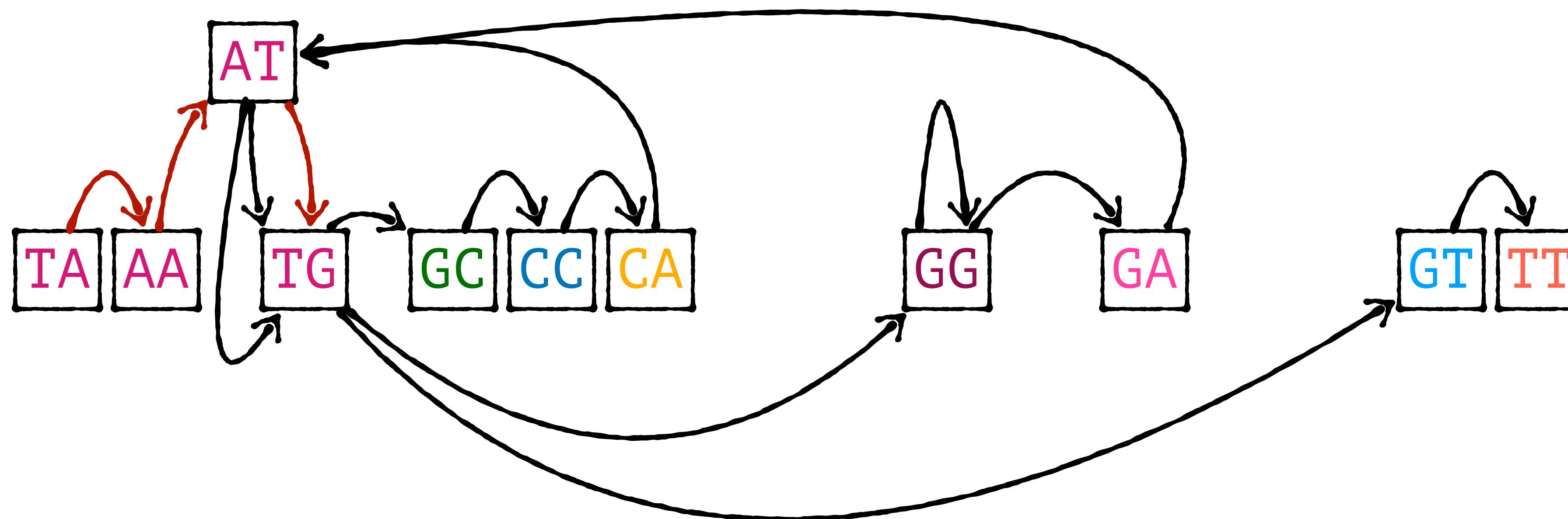
TAAT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

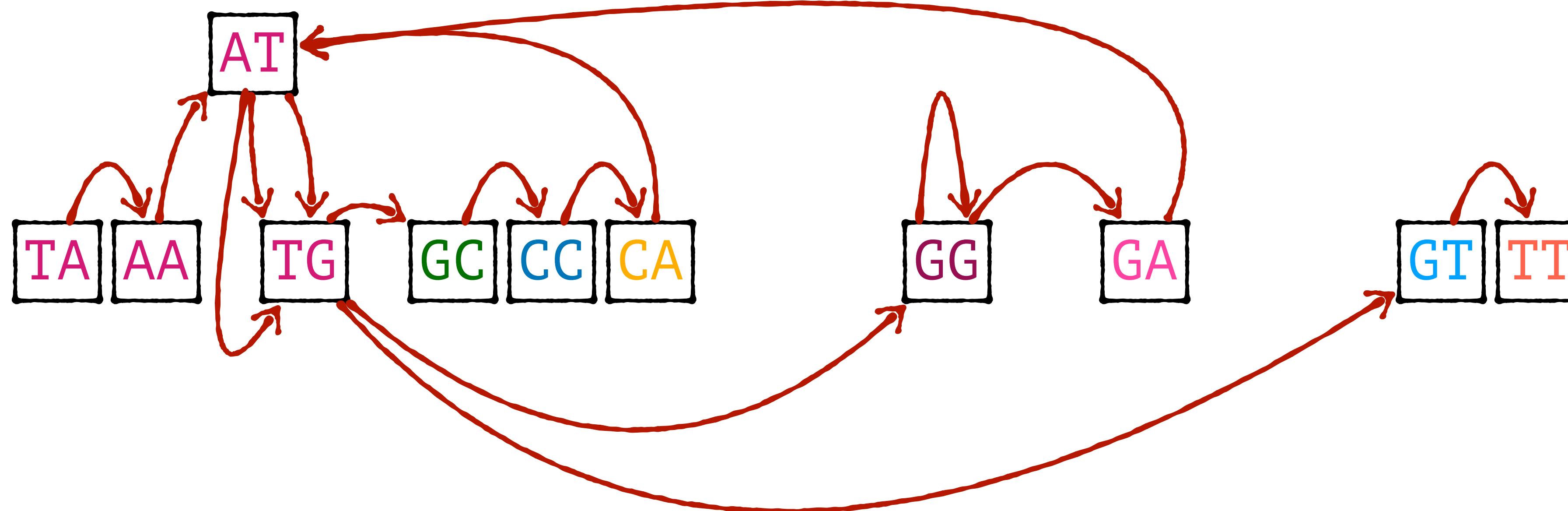
TAATG



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

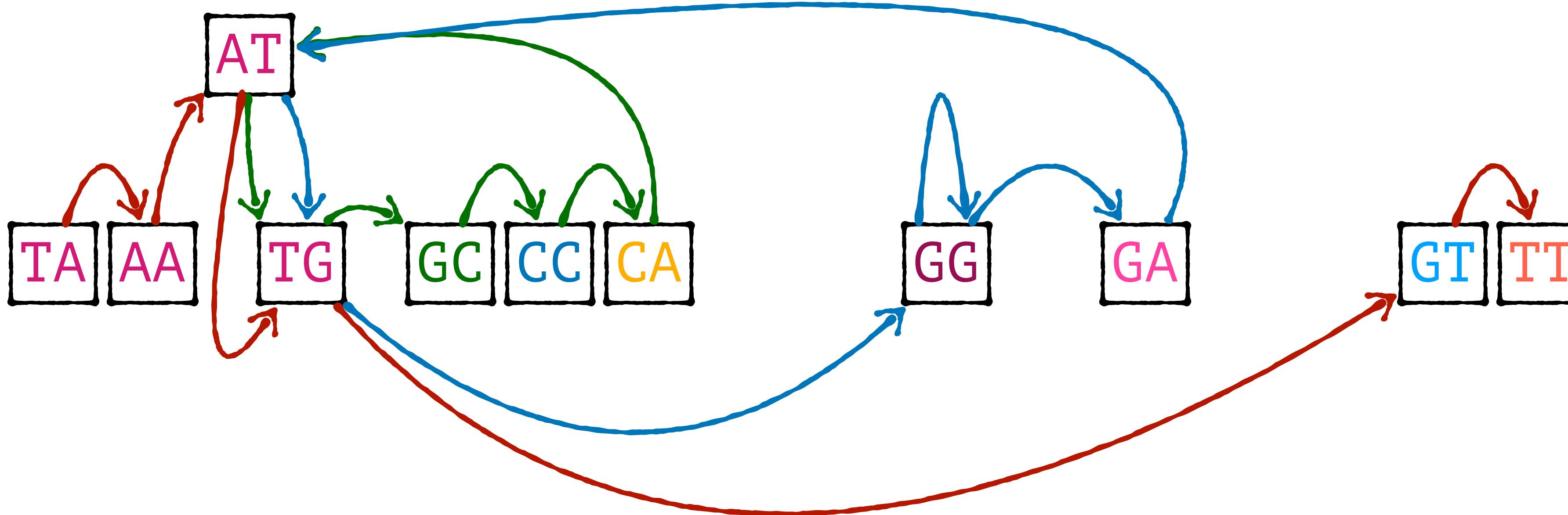
TAATGCCATGGGATGTT



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

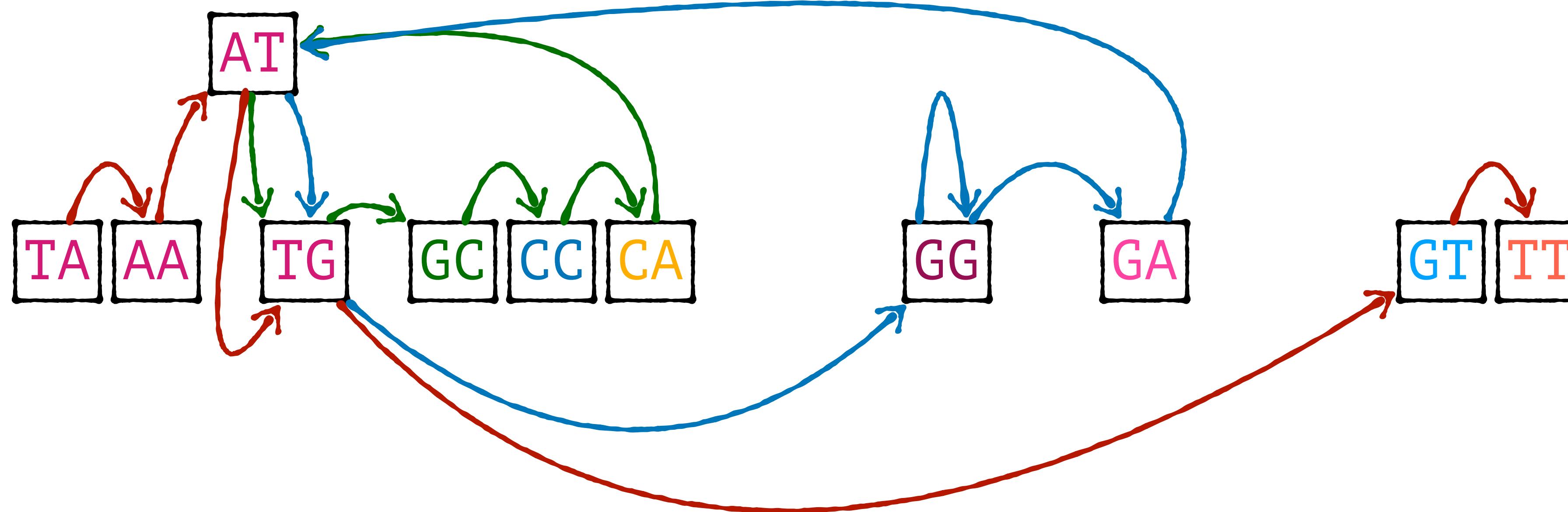
TAAT**GCCATGGGATGTT**



Сборка геномов. Неидеальное покрытие.

TAATGCCATGGGATGTT

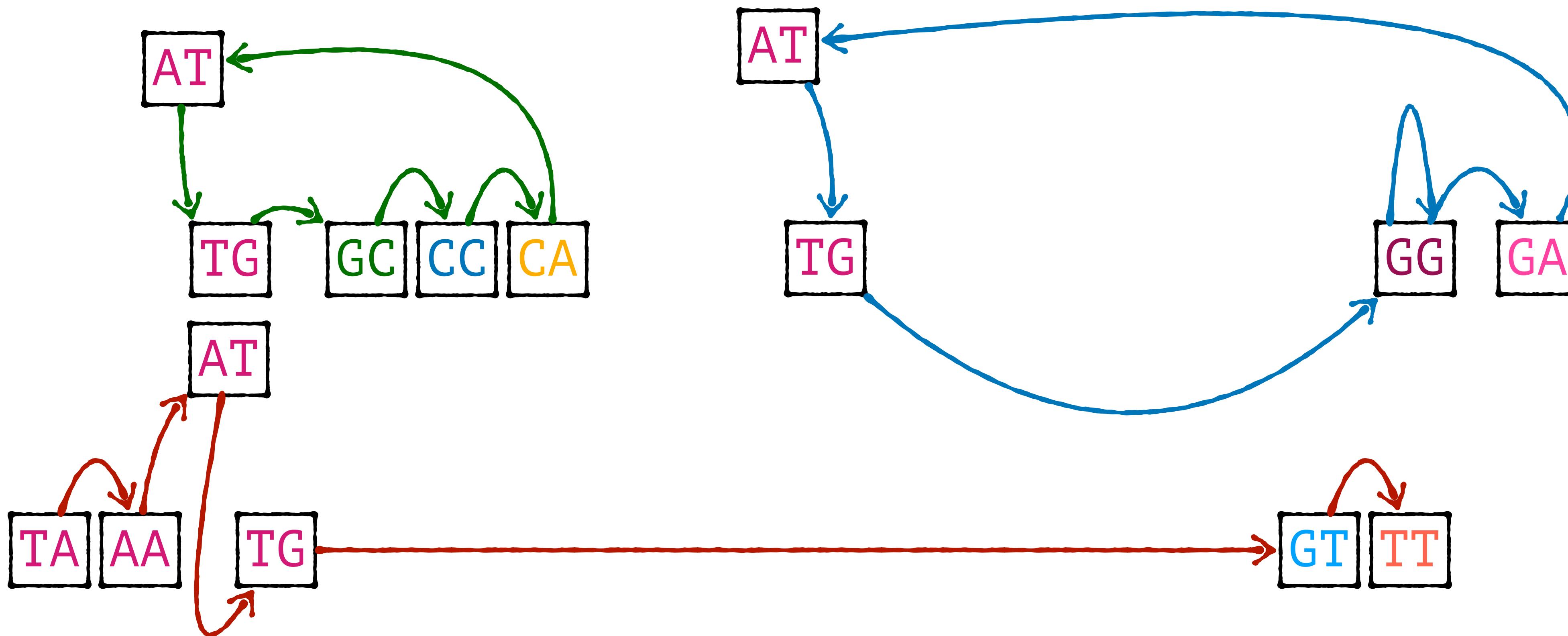
TAATGCCATGGGATGTT
TAATGGGATGCCATGTT



Сборка геномов. Неидеальное покрытие.

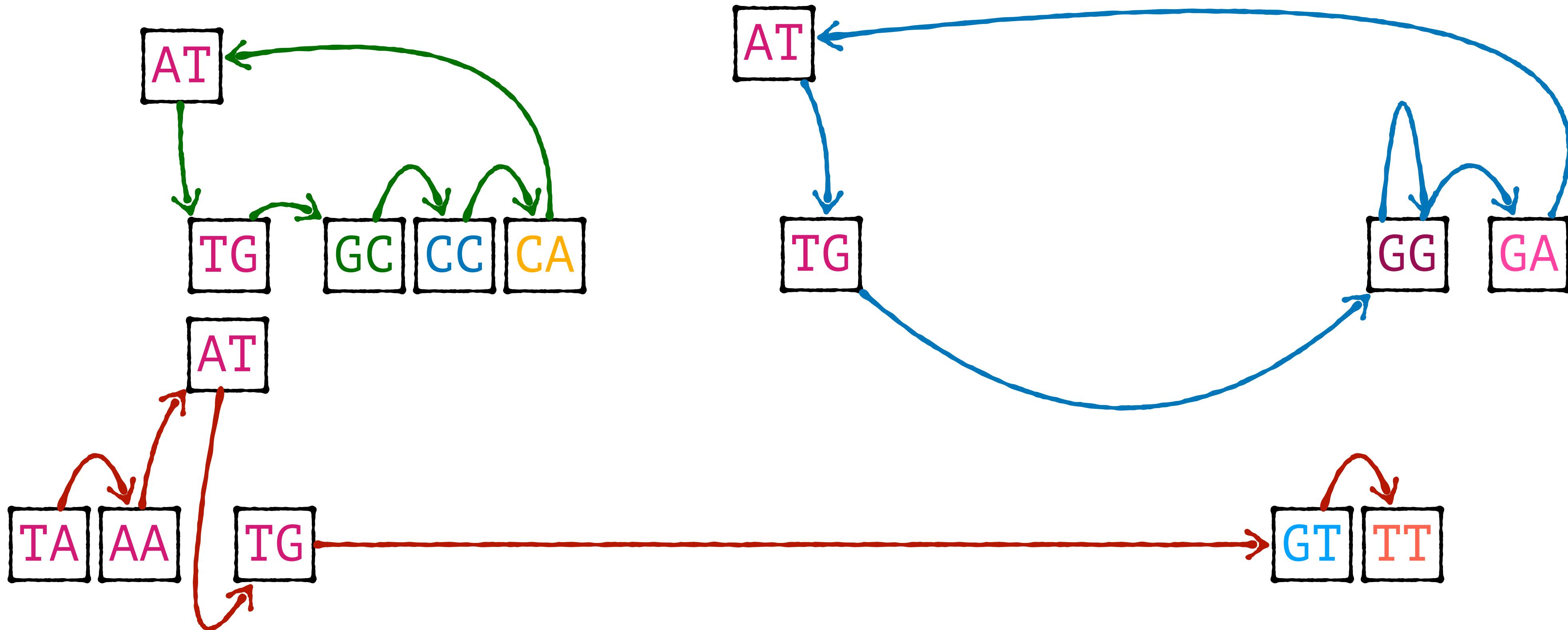
TAATGCCATGGGATGTT

TAAT, GGGAT, GCCAT, GTT



Сборка геномов. Неидеальное покрытие.

- Сложность обхода всех подграфов $O(E)$



Сборка геномов. Парные риды.

TAATGCCATGGGATGTT

TAATG ATGGG
AATGC TGGGA
ATGCC GGGAT
TGCCA GGATG
GCCAT GATGT
CCATG ATGTT

Сборка геномов. Парные риды.

TAATGCCATGGGATGTT

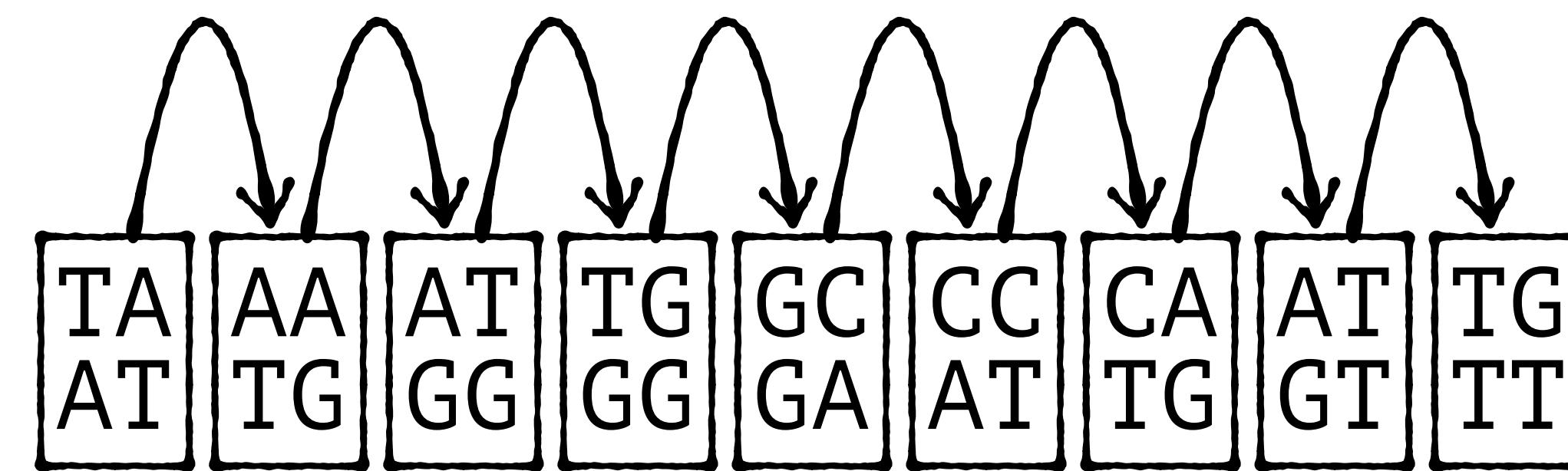
TAATG ATGGG
AATGC TGGGA
ATGCC GGGAT
TGCCA GGATG
GCCAT GATGT
CCATG ATGTT

Будем работать не просто с k-мерами,
а с парными k-мерами: TAA, ATG -> (TAA, ATG)

Сборка геномов. Парные риды.

TAATGCCATGGGATGTT

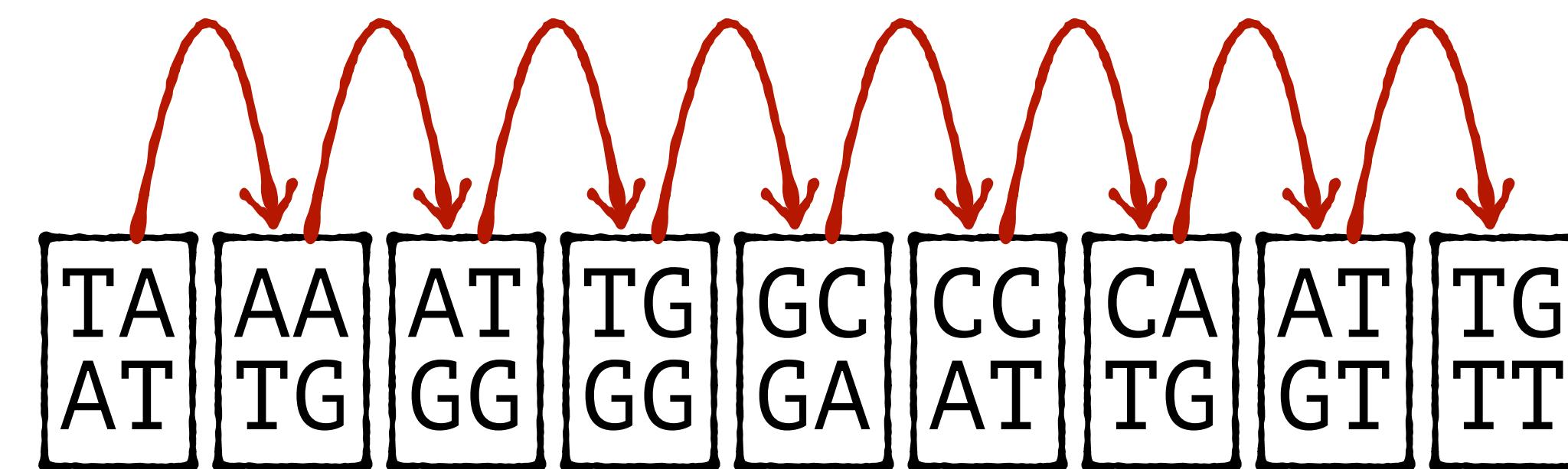
TAATG ATGGG
AATGC TGGGA
ATGCC GGGAT
TGCCA GGATG
GCCAT GATGT
CCATG ATGTT



Сборка геномов. Парные риды.

TAATGCCATGGGATGTT

TAATG ATGGG
AATGC TGGGA
ATGCC GGGAT
TGCCA GGATG
GCCAT GATGT
CCATG ATGTT

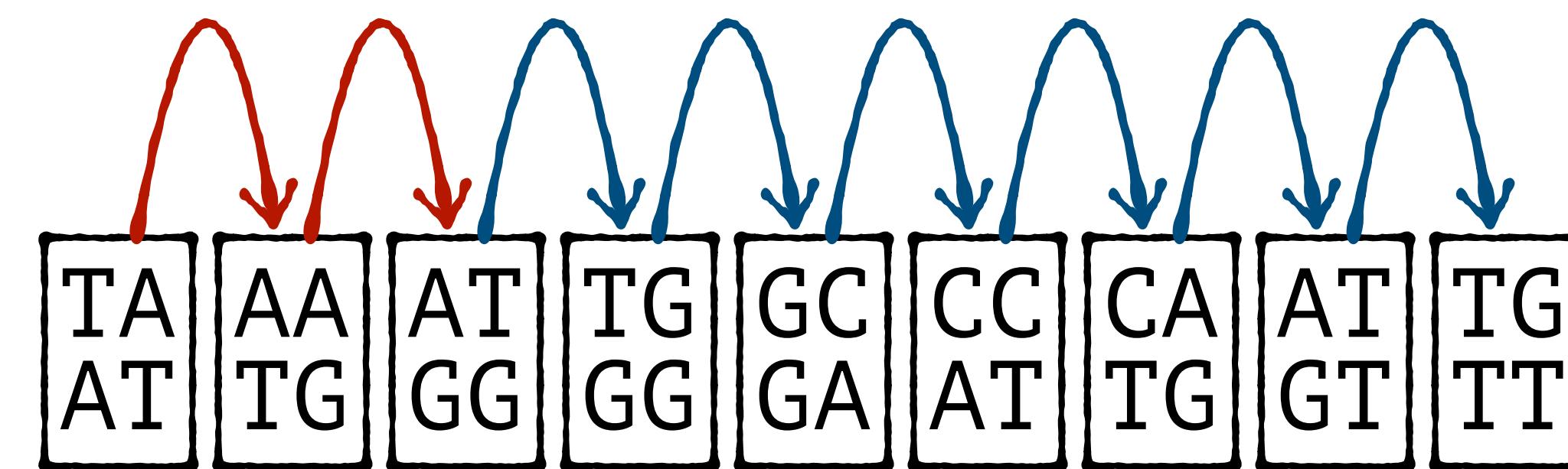


TAATGCCATG

Сборка геномов. Парные риды.

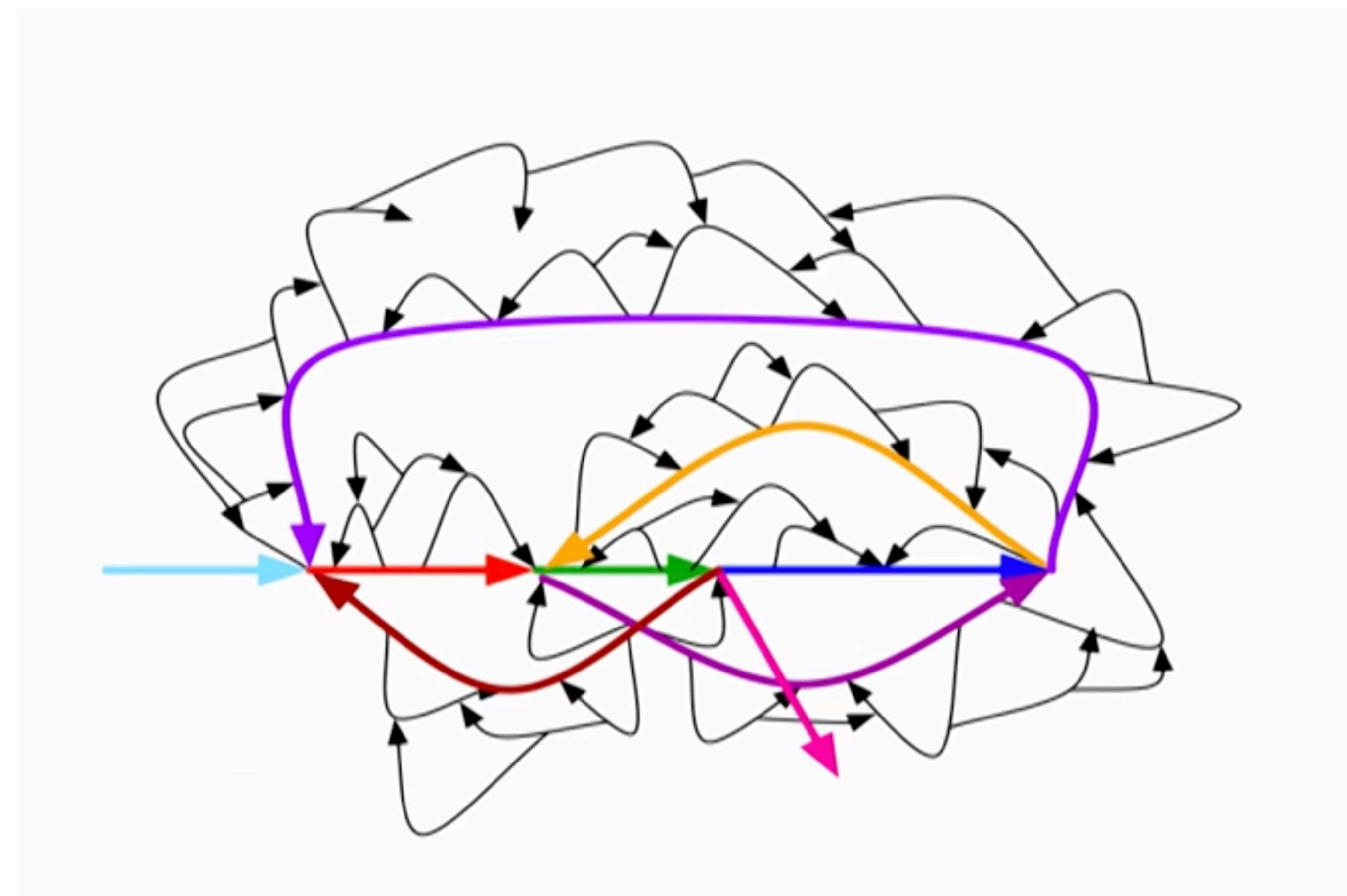
TAATGCCATGGGATGTT

TAATG ATGGG
AATGC TGGGA
ATGCC GGGAT
TGCCA GGATG
GCCAT GATGT
CCATG ATGTT



TAATGCCATGGGATGTT

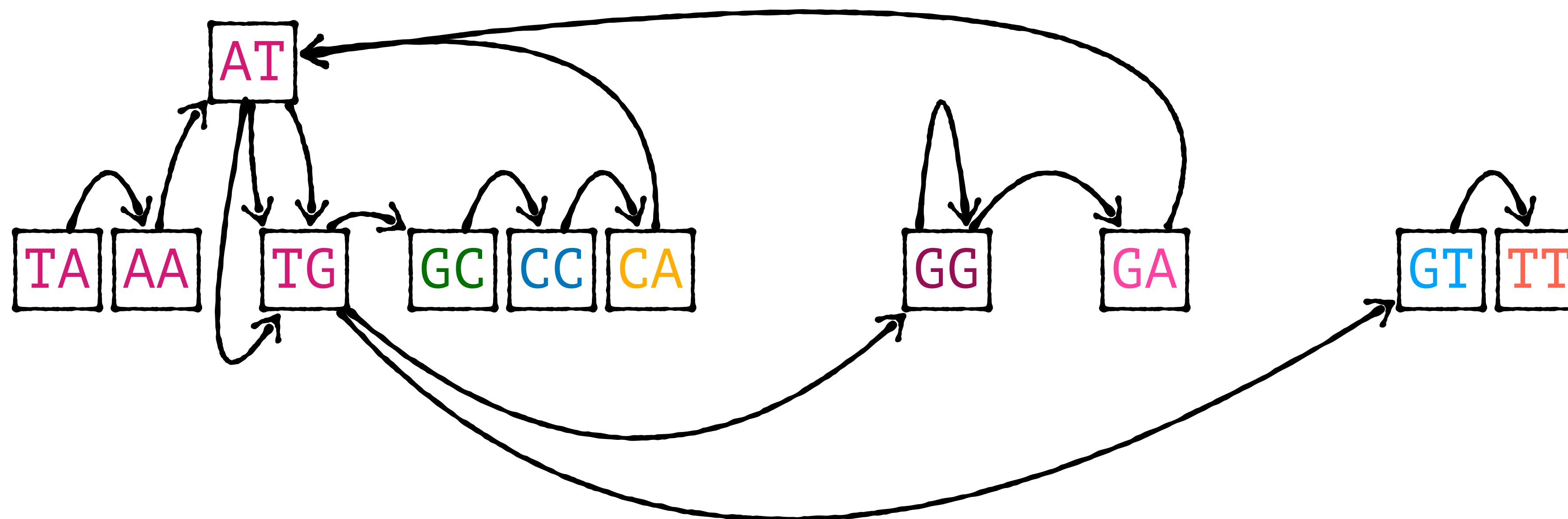
Сборка геномов. Ошибки



Сборка геномов.

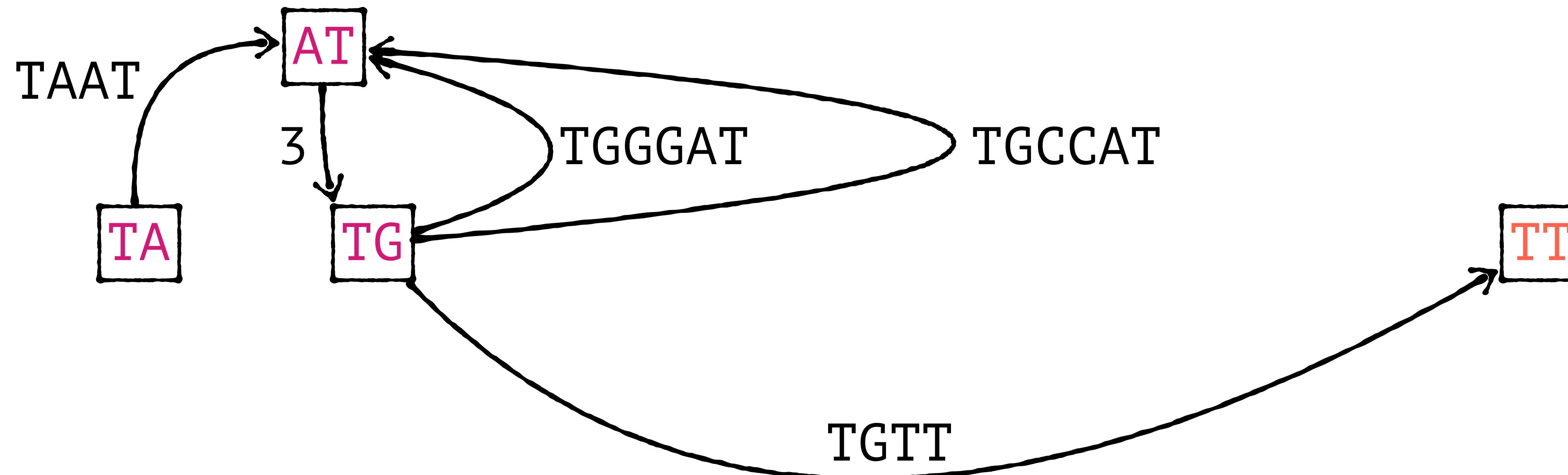
- Покрытие ребер графа – количество k -меров в S_k которые соответствуют именно такому ребру.
- Сжатие графа Де Брюина – удаление таких вершин, в которые входит одно ребро и выходит одно ребро, при этом ребра, которые соединяли смежные вершины сливаются.

Сборка геномов. Сжатие.

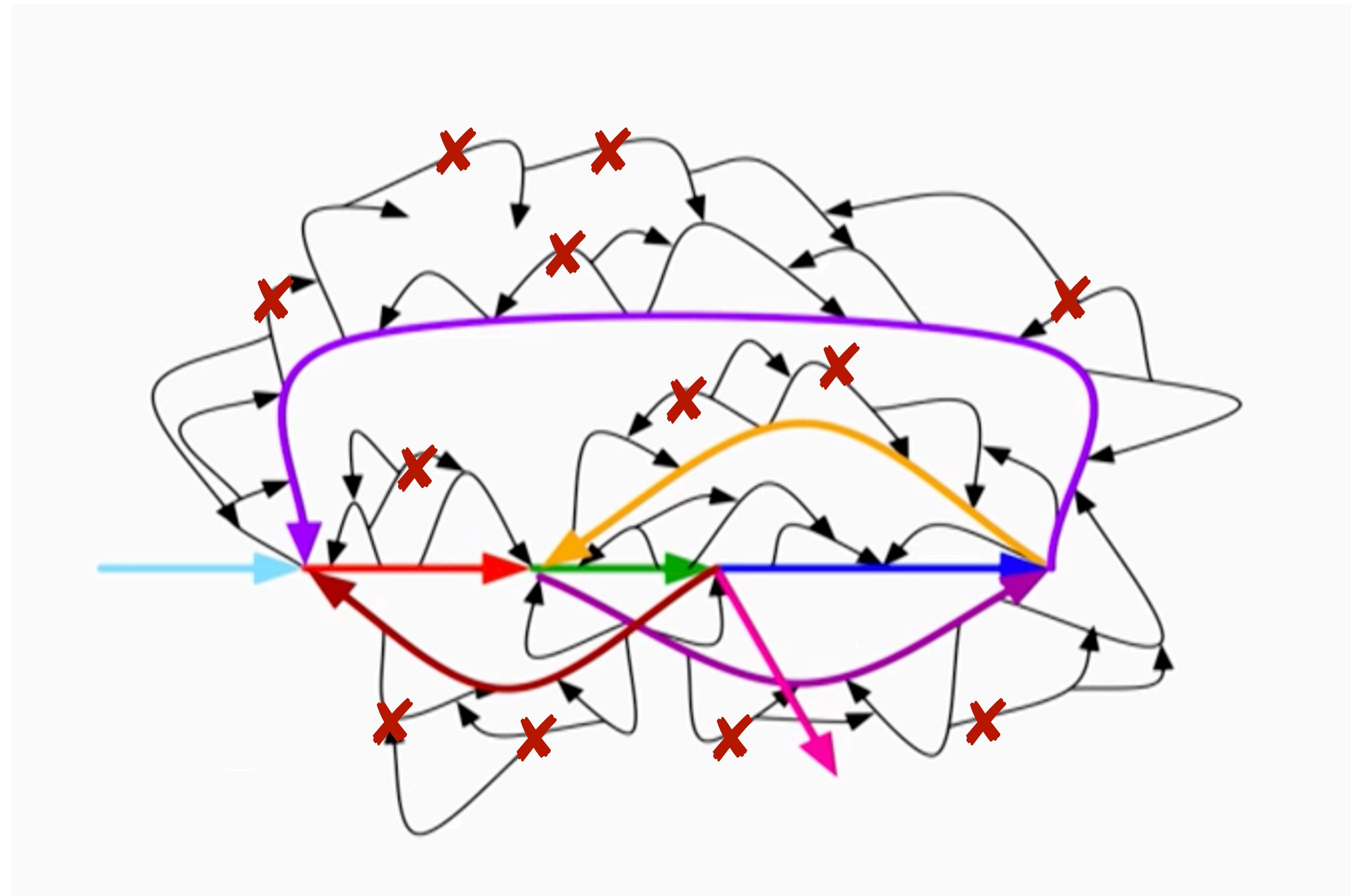


Сборка геномов. Сжатие.

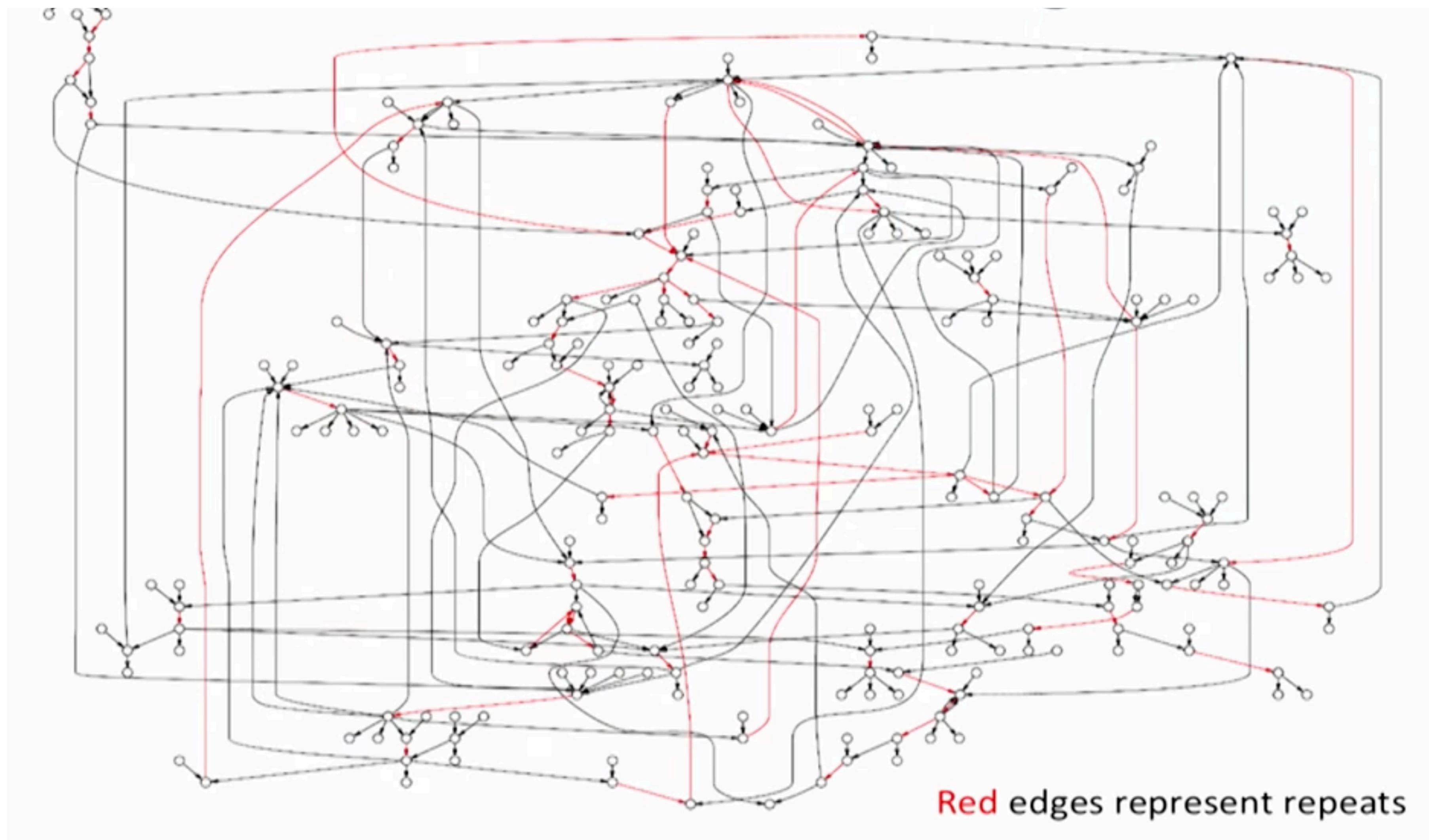
TAATGCCATGGGATGTT



Сборка геномов. Удаление ребер.



Сборка геномов.



Сборка геномов. Общий алгоритм.

- Предварительное исправление ошибок
- Построение графа де Брюина
- Коррекция графа
- Разрешение повторов
- Получение контигов и скэффолдов
- Этап консенсуса

Резюмируем

- Современные алгоритмы сборки геномов работают с такой структурой, как граф Де Брюина
- Неидеальность ридов ставит огромное количество задач при сборке
- При сборке используется как предобработка, так и постобработка данных
- Результат сборки, как правило, не весь геном, а контиги и скэффолды