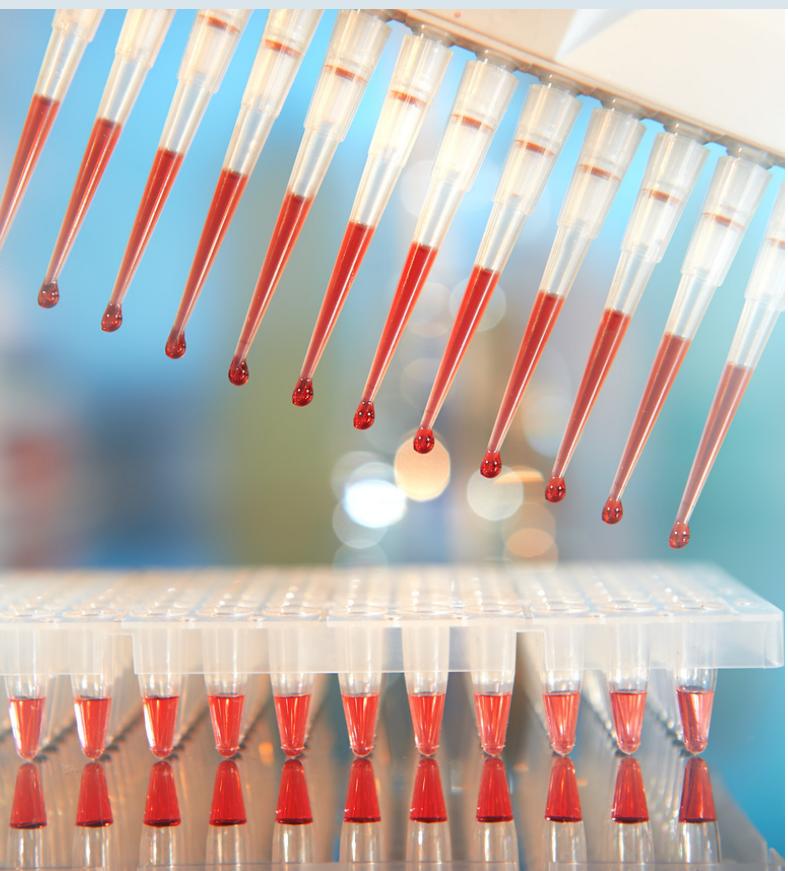
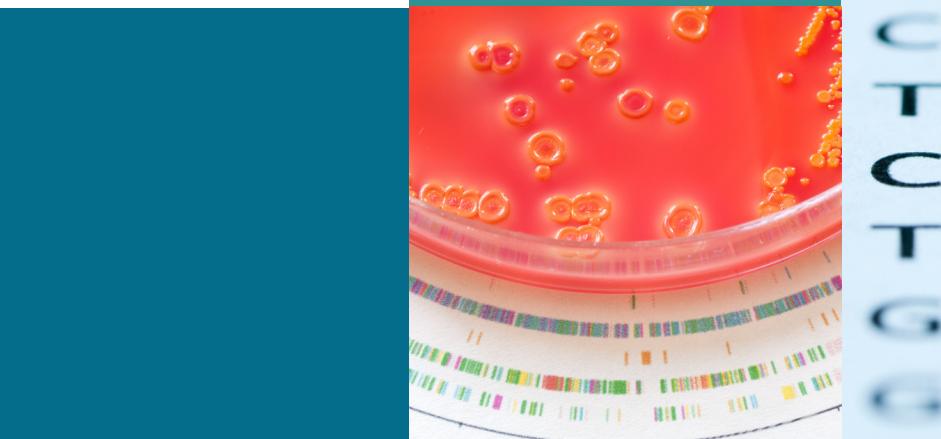


# Modelos e algoritmos para montagem de genomas



# Sequenciamento de genomas

**Consiste em decifrar a sequência de nucleotídeos do genoma dos organismos**

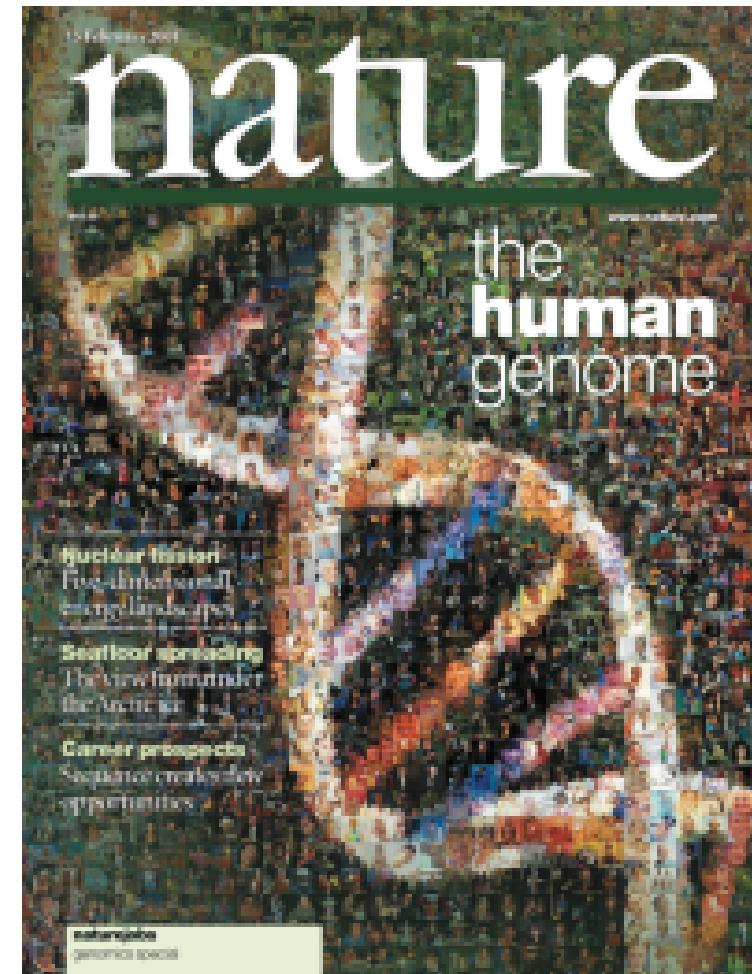
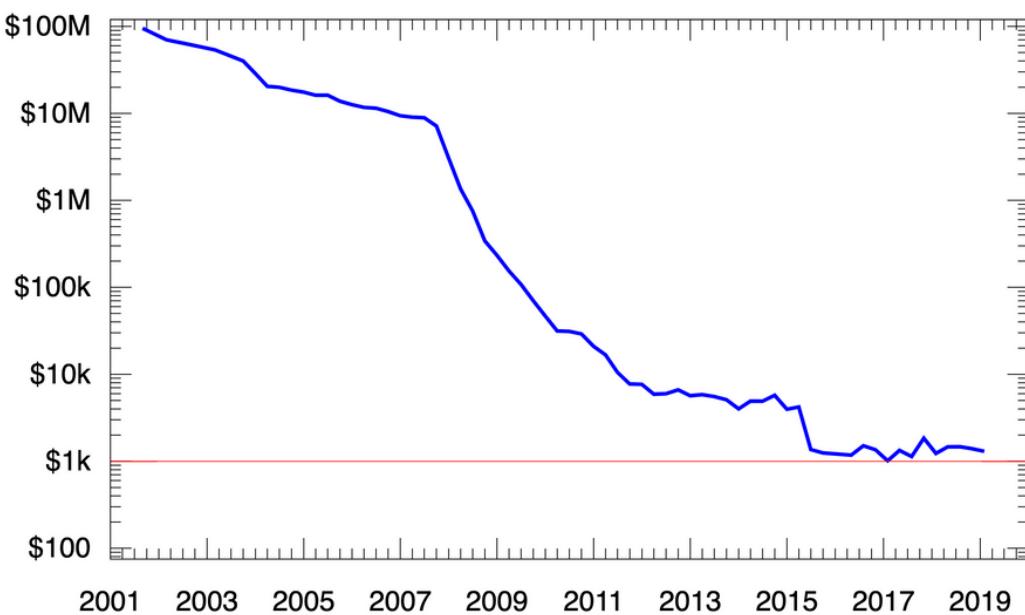
Tem contribuído enormemente para inúmeras áreas de pesquisa:

- Biologia molecular
- Biologia evolucionária
- Metagenômica
- Virologia
- Medicina
- Ciência forense

# Histórico

## Linha do tempo

Custo do sequenciamento de um genoma humano



**1970**

**Primeiras sequências de DNA** foram obtidas de forma bem artesanal e laboriosa

**1977**

**Primeiro genoma completo** sequenciado (bacteriófago)

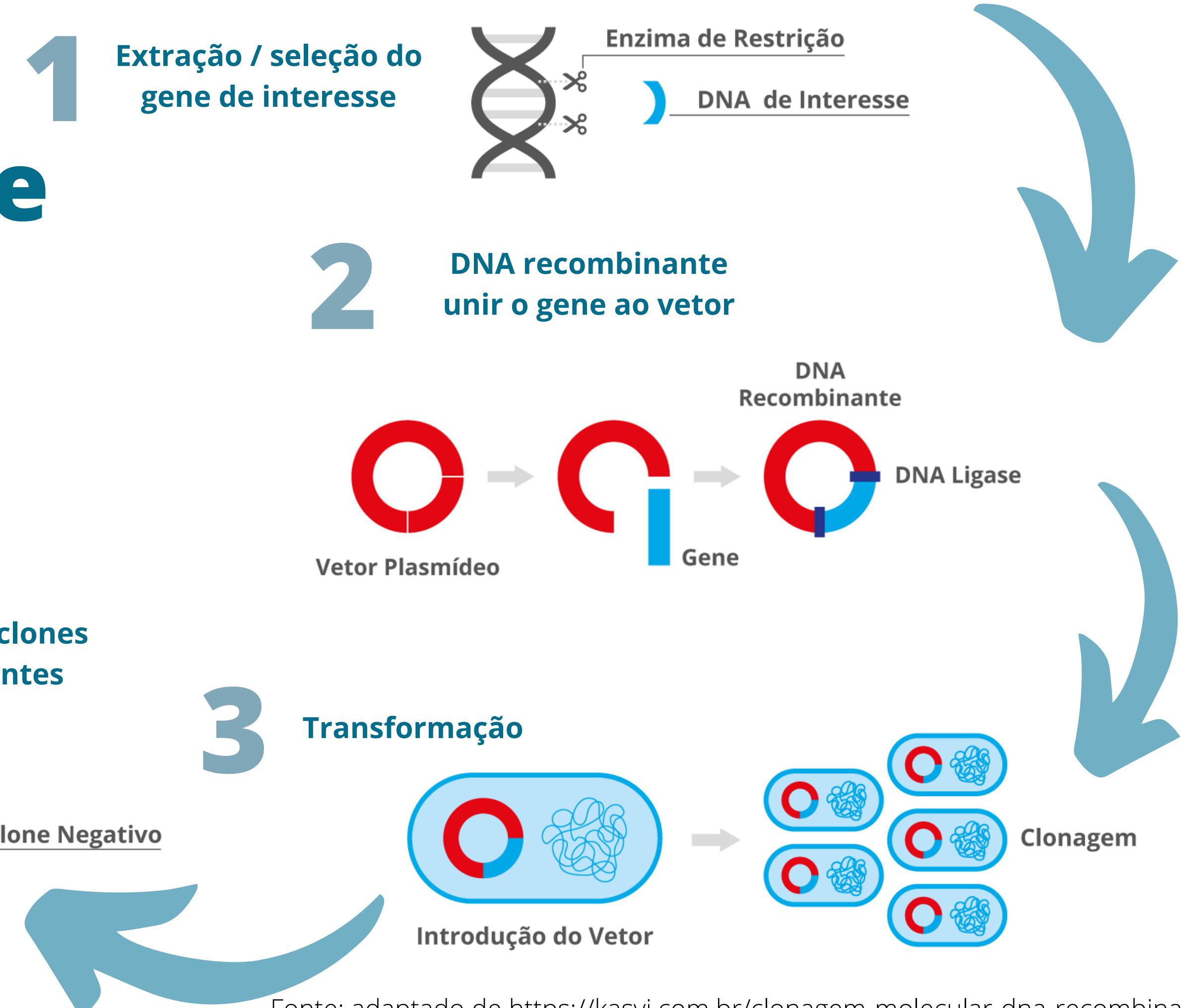
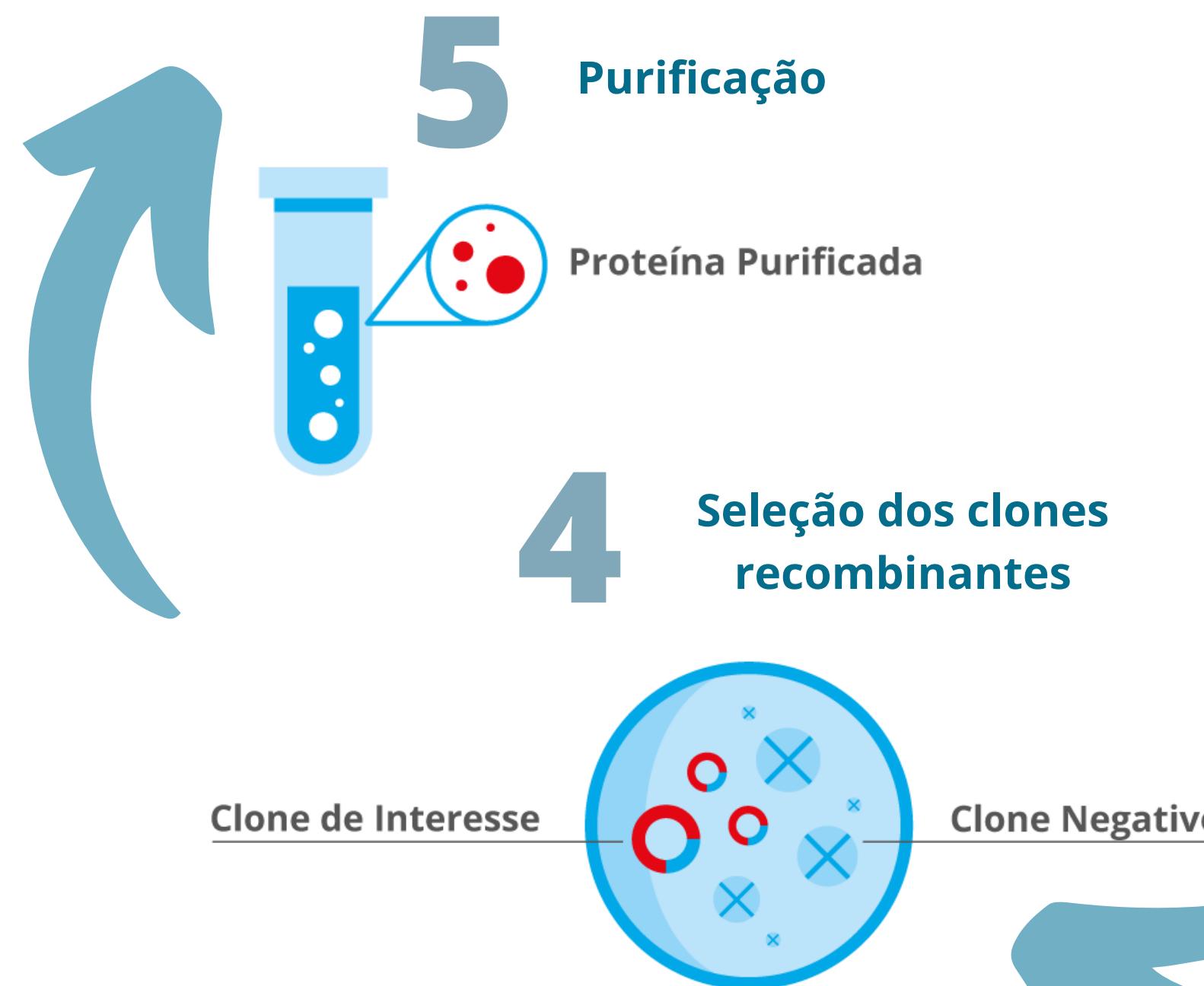
**2000**

**Sequenciadores high-throughput** (HTS) ou sequenciadores de próxima/segunda geração (NGS)

**2001**

Publicação do sequenciamento do **genoma humano**

# DNA recombinante



# Sequenciamento

*Reads* são os fragmentos de DNA lidos no sequenciamento

**3-5**

Transformação,  
crescimento e  
isolamento

**6**

Sequenciamento  
da biblioteca

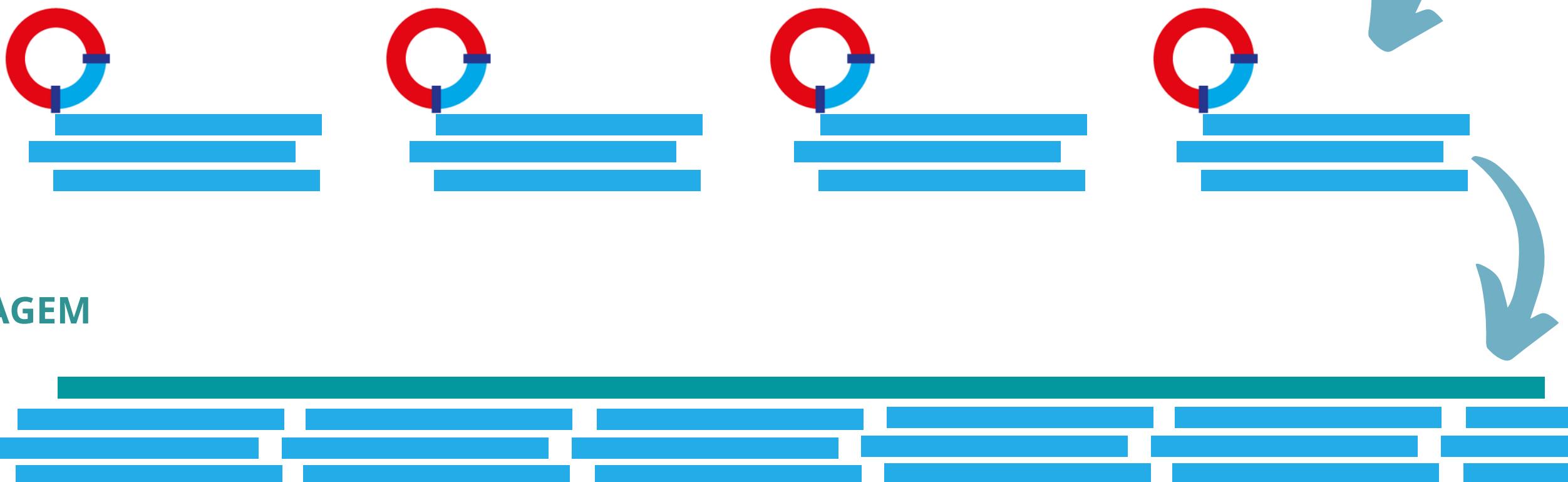
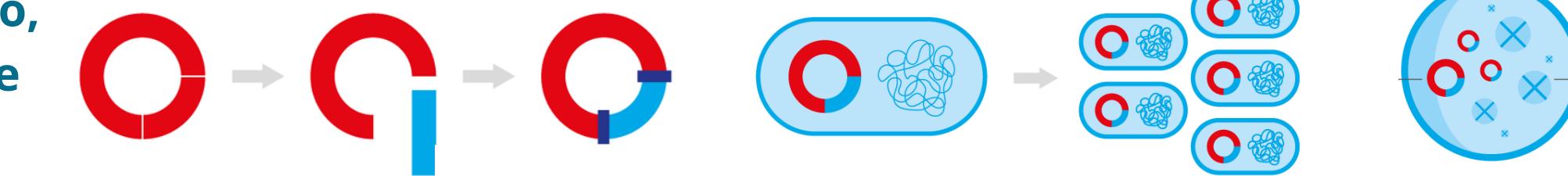
**7** MONTAGEM

**1**

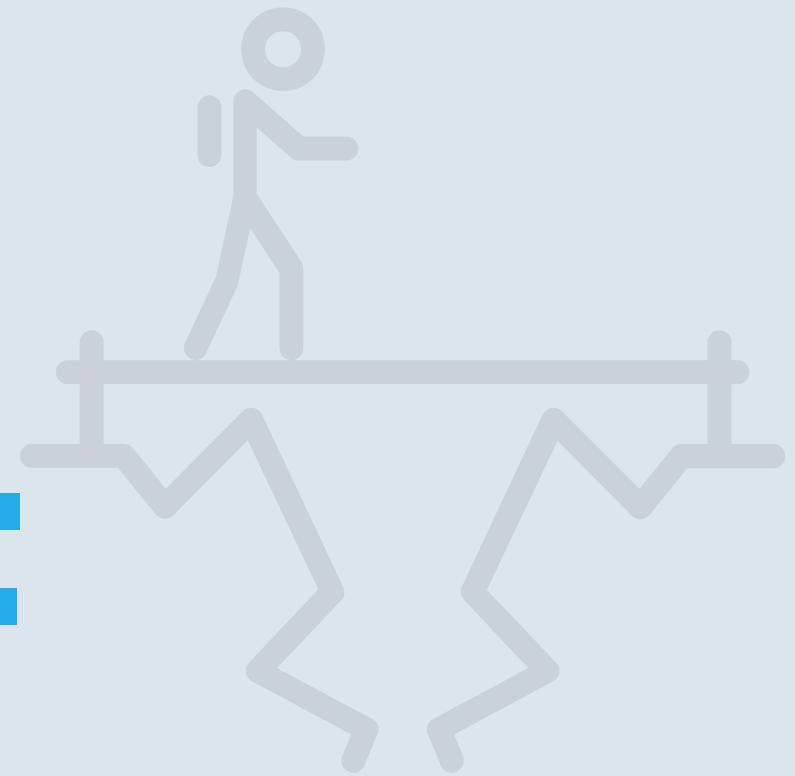
Extração

**2**

Fragmentação



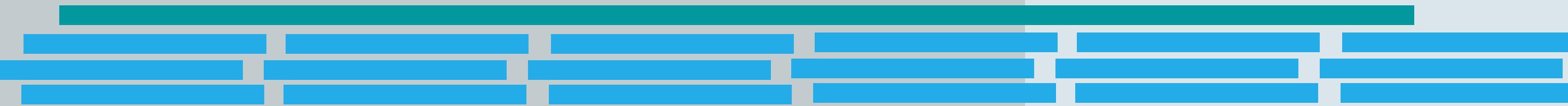
# Montagem



## Desafios computacionais...

- Montar a sequência completa através dos **fragmentos (reads)** que se sobrepõem
- Há **muitas cópias do genoma**
- Há **muitas regiões repetitivas**

# Montagem

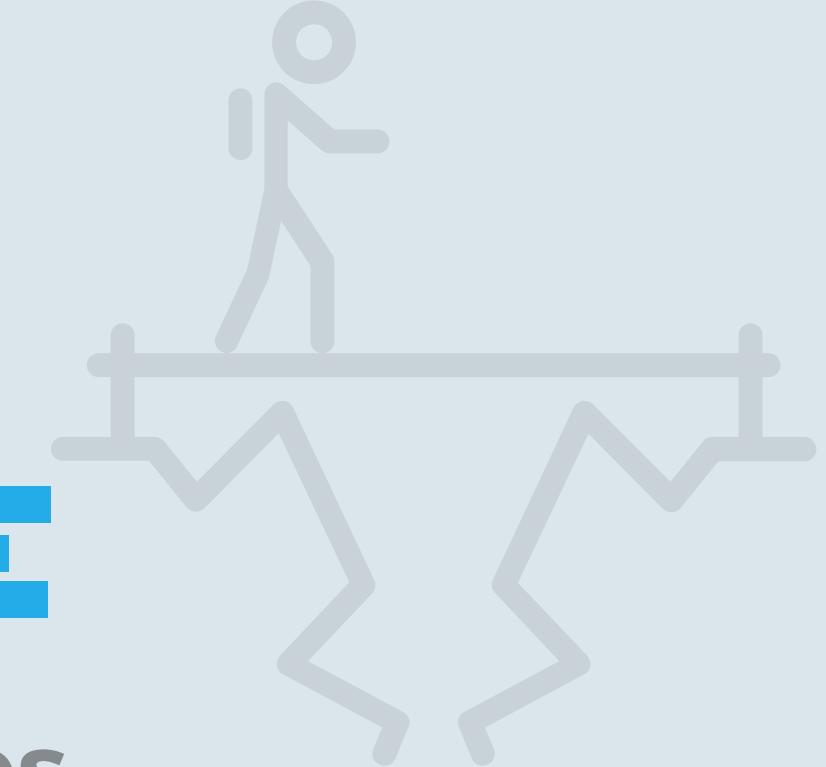


Desafios computacionais...

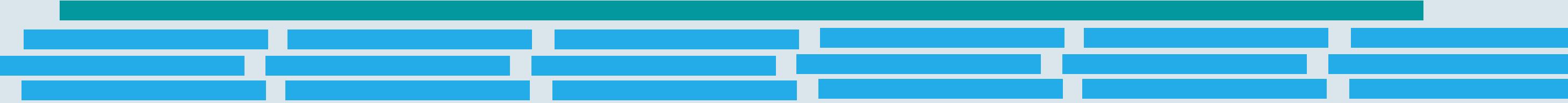
- Montar a sequência completa através dos **fragmentos (reads)** que se sobrepõem
- Há **muitas cópias do genoma**
- Há muitas **regiões repetitivas**

...há mais desafios

- Não sabemos a direção do *read* (se é o complemento reverso)
- Erros de sequenciamento (troca de nucleotídeos)
- Nem todas as regiões de um genoma podem estar cobertas pelos *reads*



# Montagem



regiões de repetições

**GCTGG**

sequência completa

**ATTGGCTGGCTAGGGCTGGGAGGCTGGCTGGA**

fragmentos

**ATTGGCTGGCTAG**

**TAGGGCTGGGAGGC**

**GAGGCTGGCTGGA**

# Problema da reconstrução de *strings*

## Problema da composição de *strings*

Gerar os *k-mers* que compõem uma *string*

- Entrada: uma *string*
- Saída: A composição da *string* em termos dos *k-mers*

TATGGGGTGC  
{ATG, GGG, GGG, GGT, GTG,  
TAT, TGC, TGG}

# Problema da reconstrução de *strings*

## Problema da reconstrução de *strings*

{AAT, ATG, GTT, TAA, TGT}

Reconstruir uma *string* a partir de sua  
composição de *k-mers*

- Entrada: um inteiro  $k$  e uma coleção  
de *k-mers*
- Saída: Uma *string* que foi  
recomposta do conjunto de *k-mers*

**TAA**

AAT

ATG

TGT

GTT

**TAATGTT**

# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

TAA

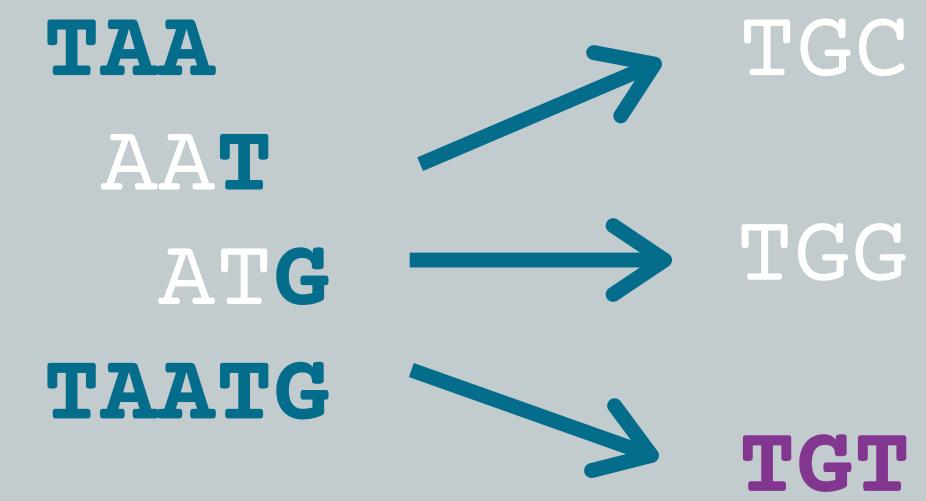
AA T

ATG

TAATG

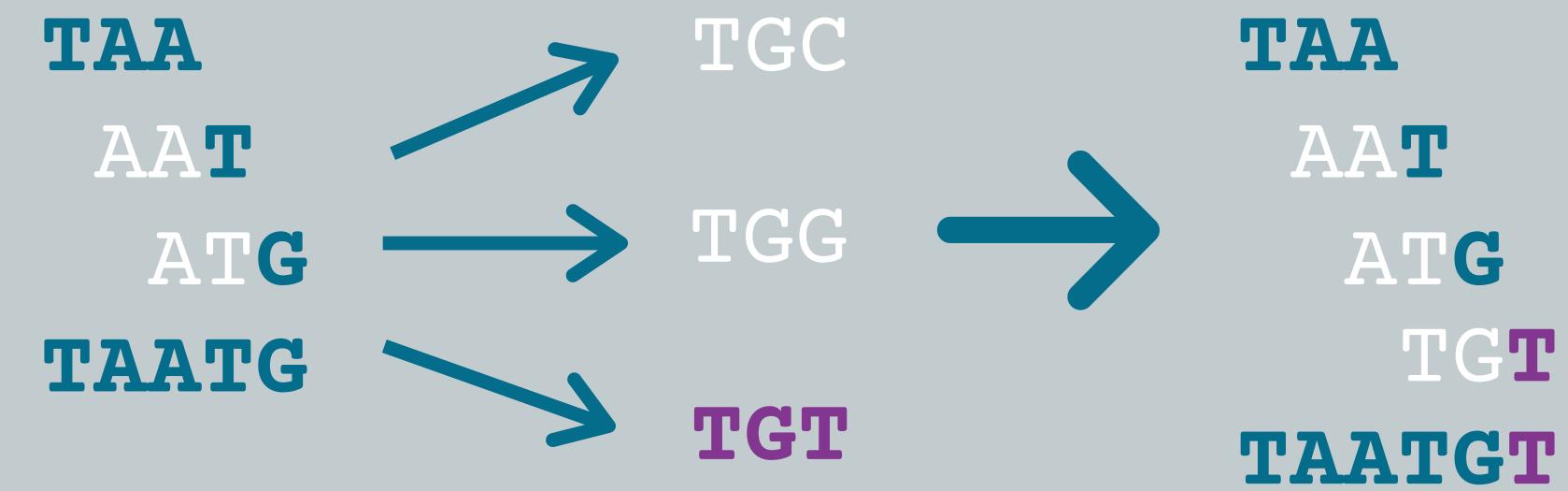
# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT



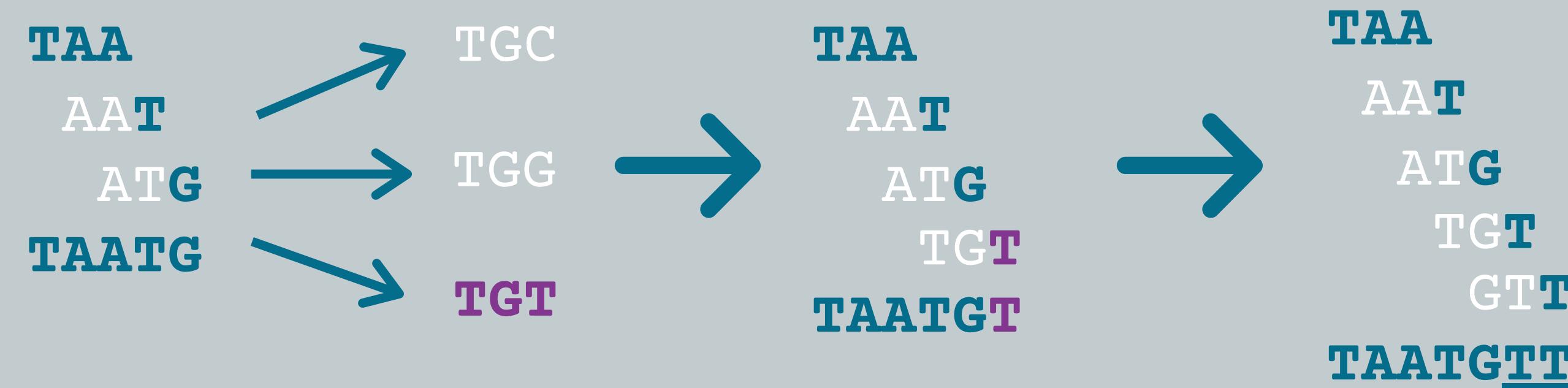
# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT



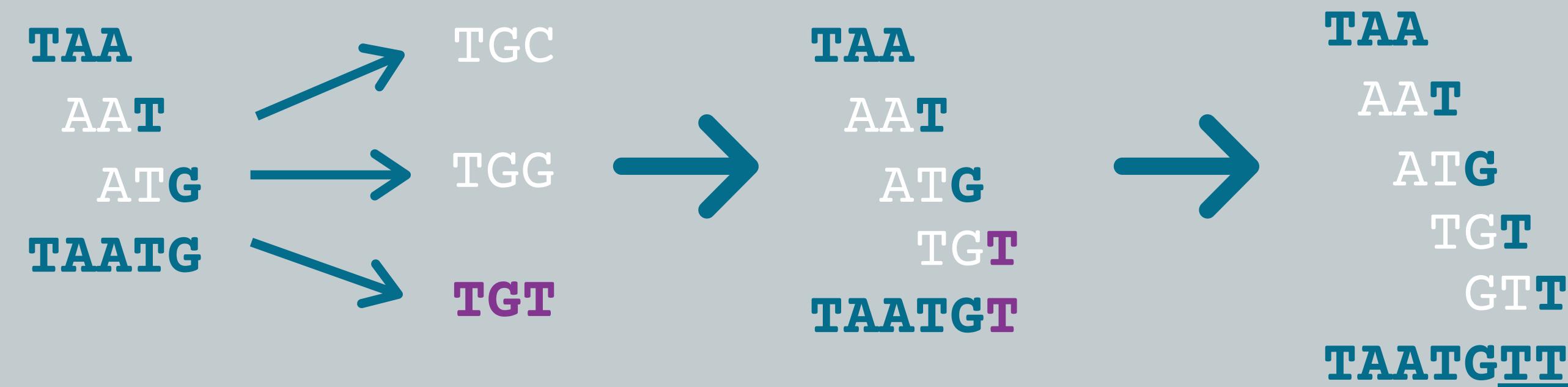
# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT



# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT



# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

**TAA**

AAT

ATG

TGC

GCC

CCA

CAT

ATG

TGG

GGA

GAT

ATG

TGT

GTT

**TAATGCCATGGATGTT**

**TAA**

AAT

ATG

TGC

**TAATGC**



# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

TAA  
AAT  
ATG  
TGC  
TAATGC



TAA  
AAT  
ATG  
TGC  
GCC  
CCA  
CAT  
ATG  
TGG  
GGA  
GAT  
ATG  
TGT  
GTT  
TAATGCCATGGATGTT

Ainda não conseguimos  
montar todos os *k-mers*

# Vamos dificultar um pouco...

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

Quantas possibilidades temos para montar esse "genoma"?

- Podemos ter todas as possíveis permutações de  $n$  ou seja  $n!$ 
  - Nesse exemplo,  $15! = 1.307.674.368.000$  possibilidades

# Repetições dificultam...

AAT **ATG ATG ATG** CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

Regiões repetitivas dificultam a montagem heurística

Por que o desafio *Triazzle* é  
tão difícil mesmo tendo apenas  
16 peças?



# Modelando o problema através de grafos de sobreposição

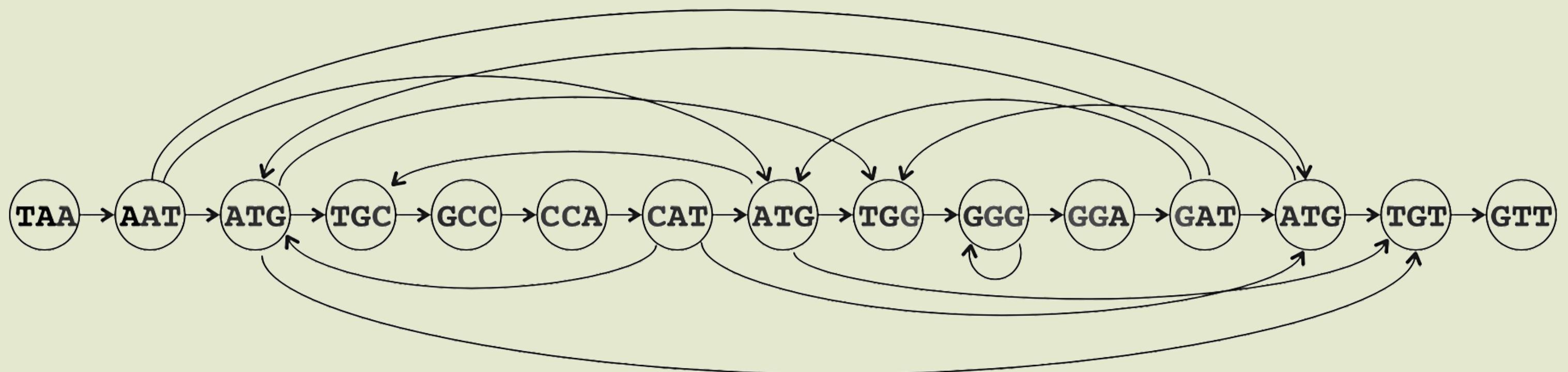
As repetições em um genoma requerem alguma maneira de "**olhar adiante**" para ver a montagem correta com **antecedência**

- Usaremos os termos:
  - **prefixo:** primeiros  $k-1$  nucleotídeos de um  $k\text{-mer}$ 
    - PREFIXO(**TAA**) = **TA**
  - **sufixo:** últimos  $k-1$  nucleotídeos de um  $k\text{-mer}$ 
    - SUFIXO(**TAA**) = **AA**
- O sufixo de um  $3\text{-mer}$  no caminho do genoma é igual ao prefixo do  $3\text{-mer}$  seguinte
  - SUFIXO(**TAA**) = PREFIXO(**AAT**) = **AA** em **TAATGCCATGGGATGTT**.

# Modelando o problema através de grafos de sobreposição

Um grafo dirigido  $G = (V,E)$  no qual:

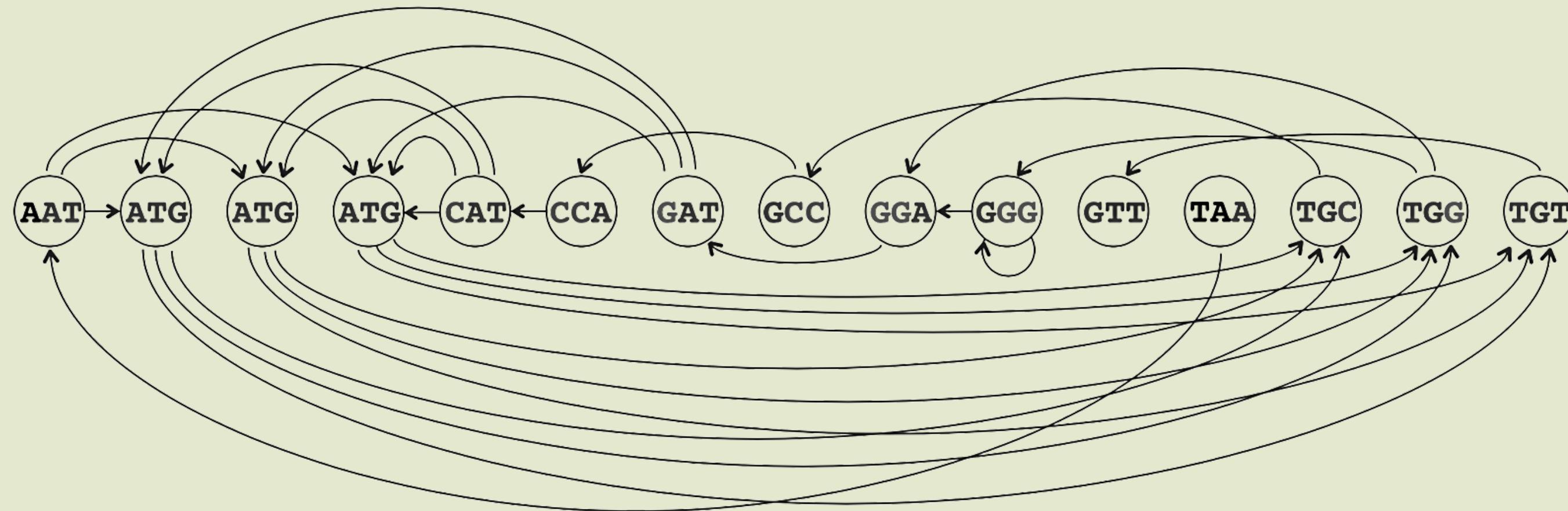
- $V$  é o conjunto de **nós** do grafo que representam os 15  **$k$ -mers**
- $E$  é o conjunto de **arestas** que indicam que  $SUFIXO(v1) = PREFIXO(v2)$ , onde  $v1$  e  $v2$  são nós do grafo (elementos de  $V$ )



Note que o genoma pode ser montado caminhando horizontalmente pelas arestas

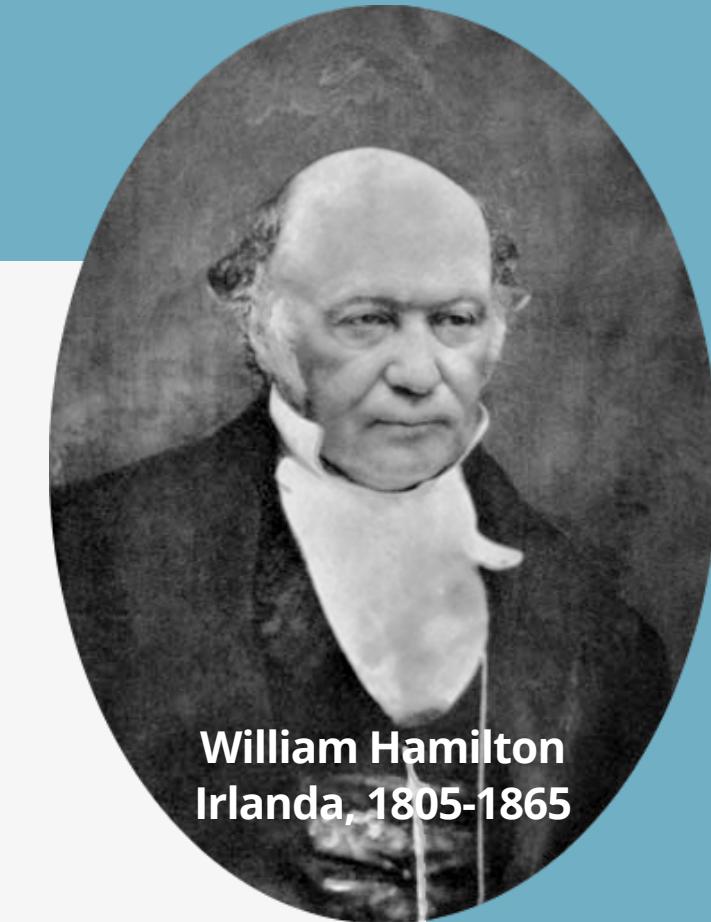
# Modelando o problema através de grafos de sobreposição

Na realidade, não conhecemos a orientação dos k-mers e eles são ordenados em **ordem lexicográfica**



Como montar o genoma então?

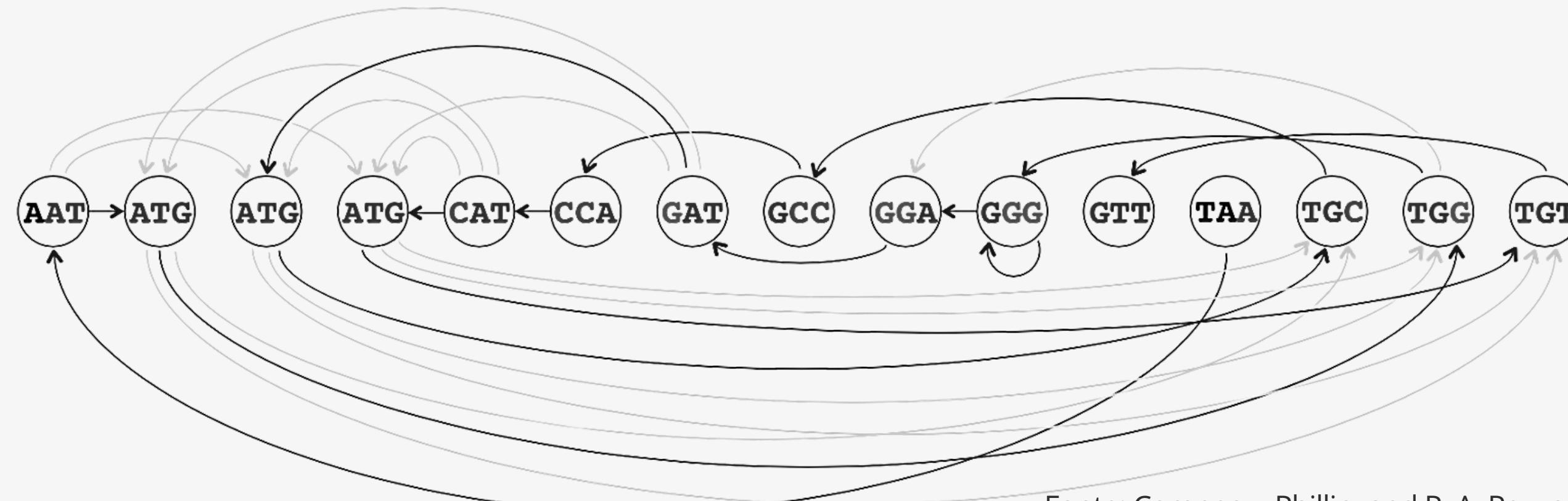
# Caminho hamiltoniano



William Hamilton  
Irlanda, 1805-1865

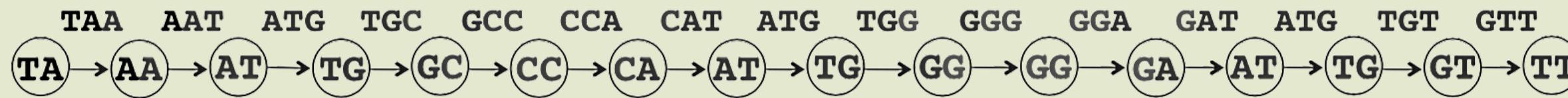
Um caminho hamiltoniano em um grafo é um caminho que passa por todos os nós, não repetindo nenhum

É um problema NP-completo



Um grafo dirigido  $G = (V, E)$  no qual:

- $E$  são **arestas** que representam os **3-mers**
- $V$  são **2-mers** que representam os nucleotídeos sobrepostos compartilhados pelas bordas de cada **3-mer**

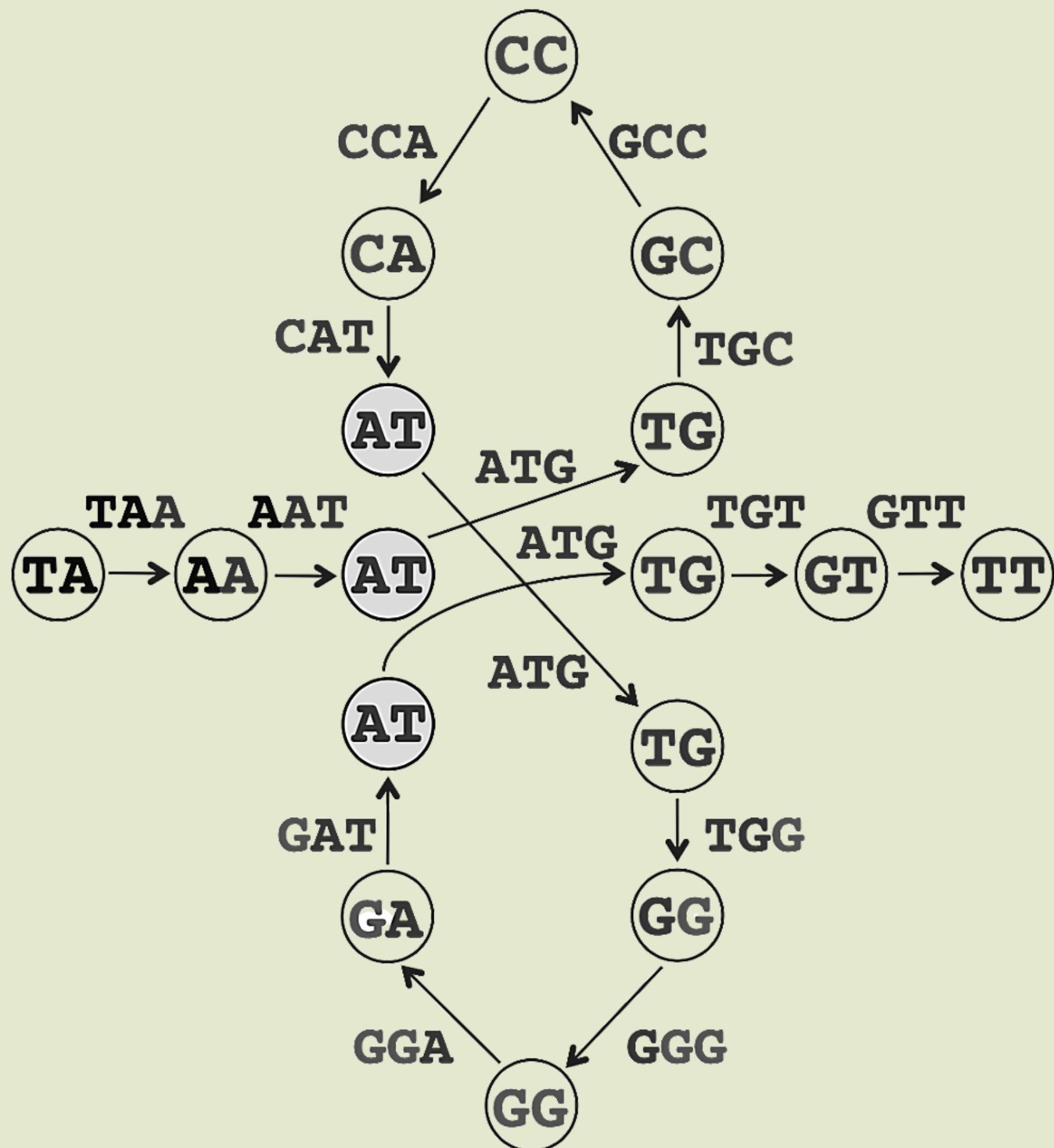


# Grafos de Bruijn



Nicolaas de Bruijn,  
Holanda, 1918-2012

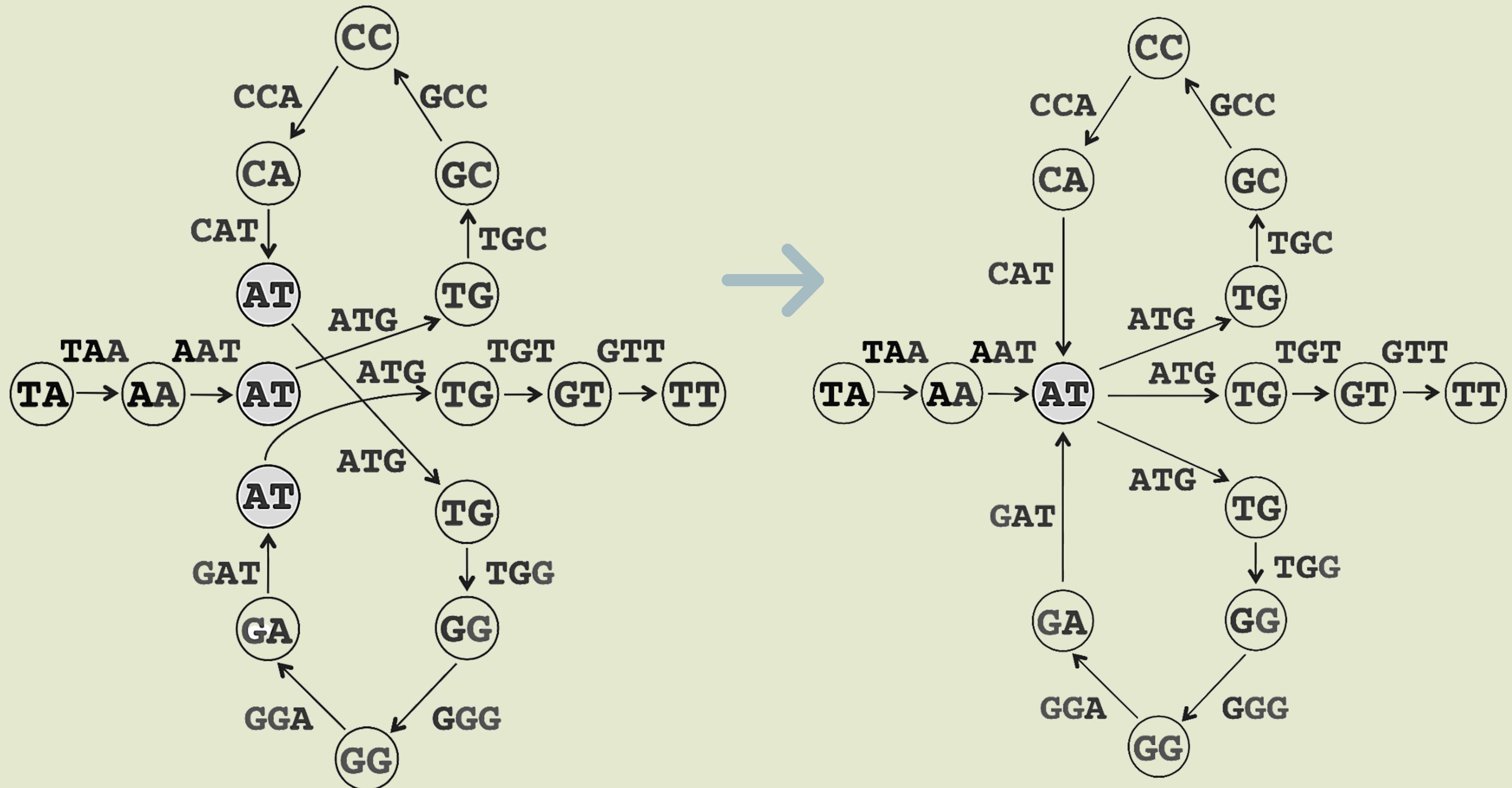
# Grafos de Bruijn



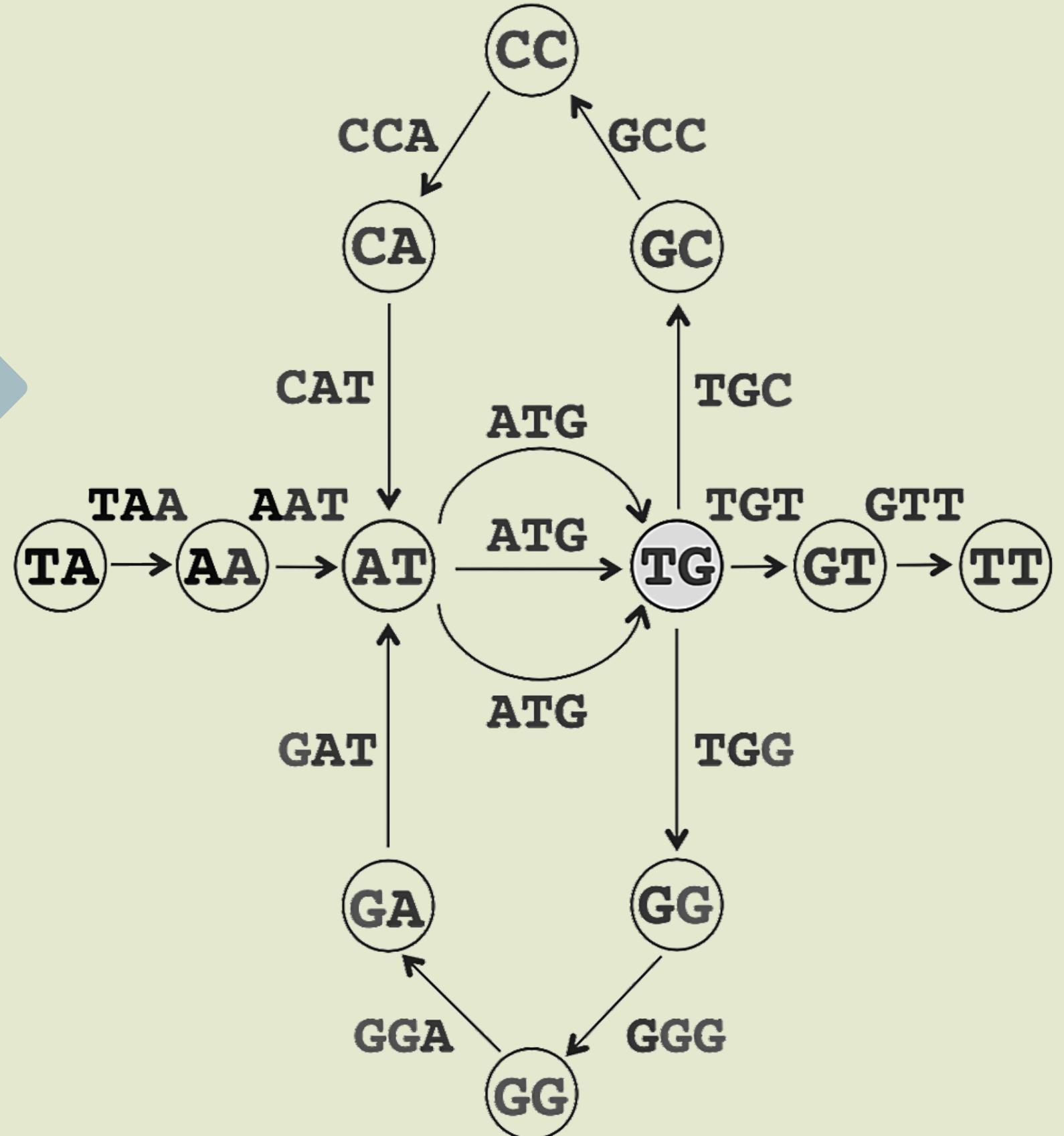
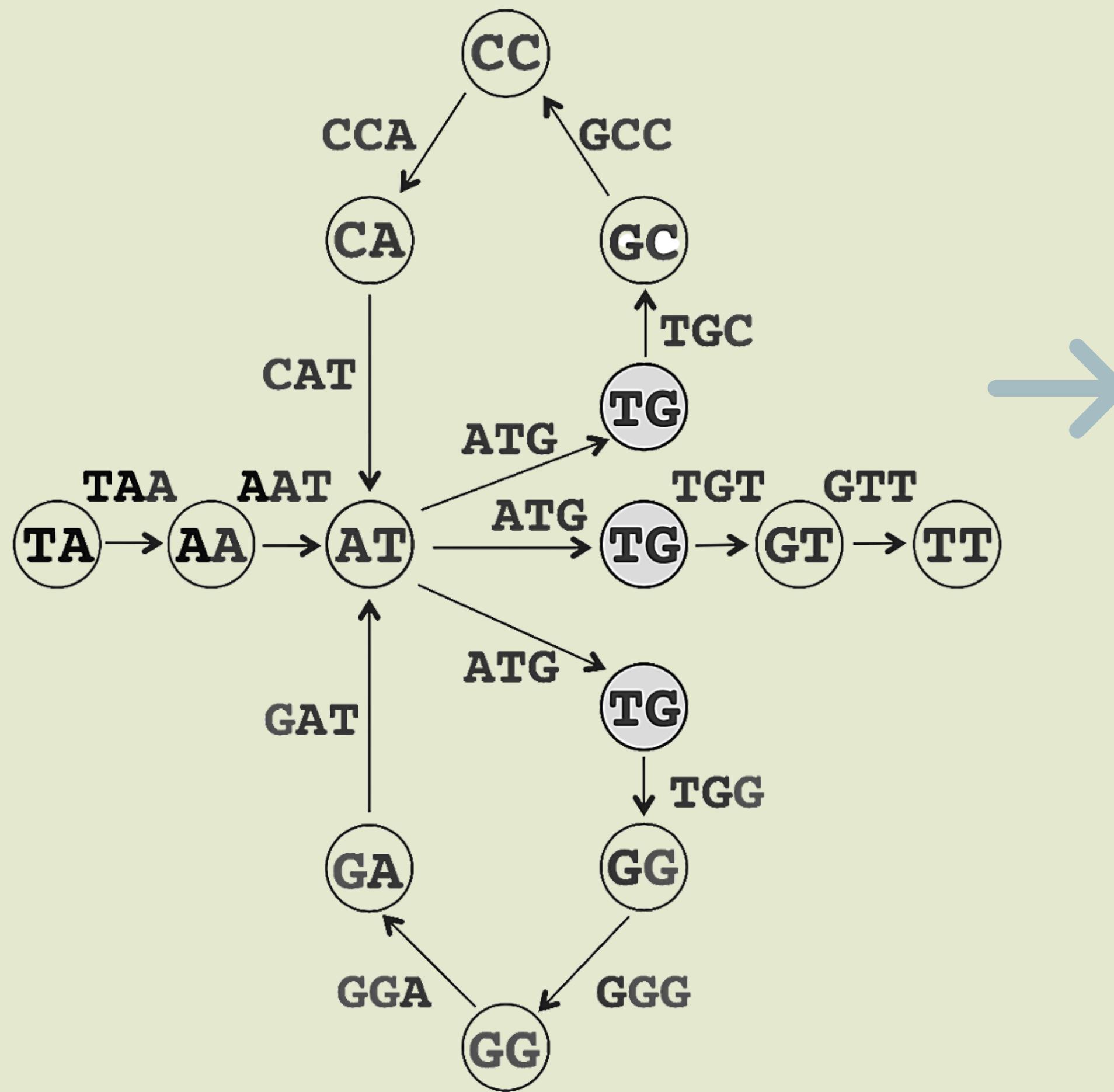
Fonte: Compeau, Phillip, and P. A. Pevzner. Bioinformatics Algorithms: An Active Learning Approach. La Jolla, CA: Active Learning Publishers, 2018.

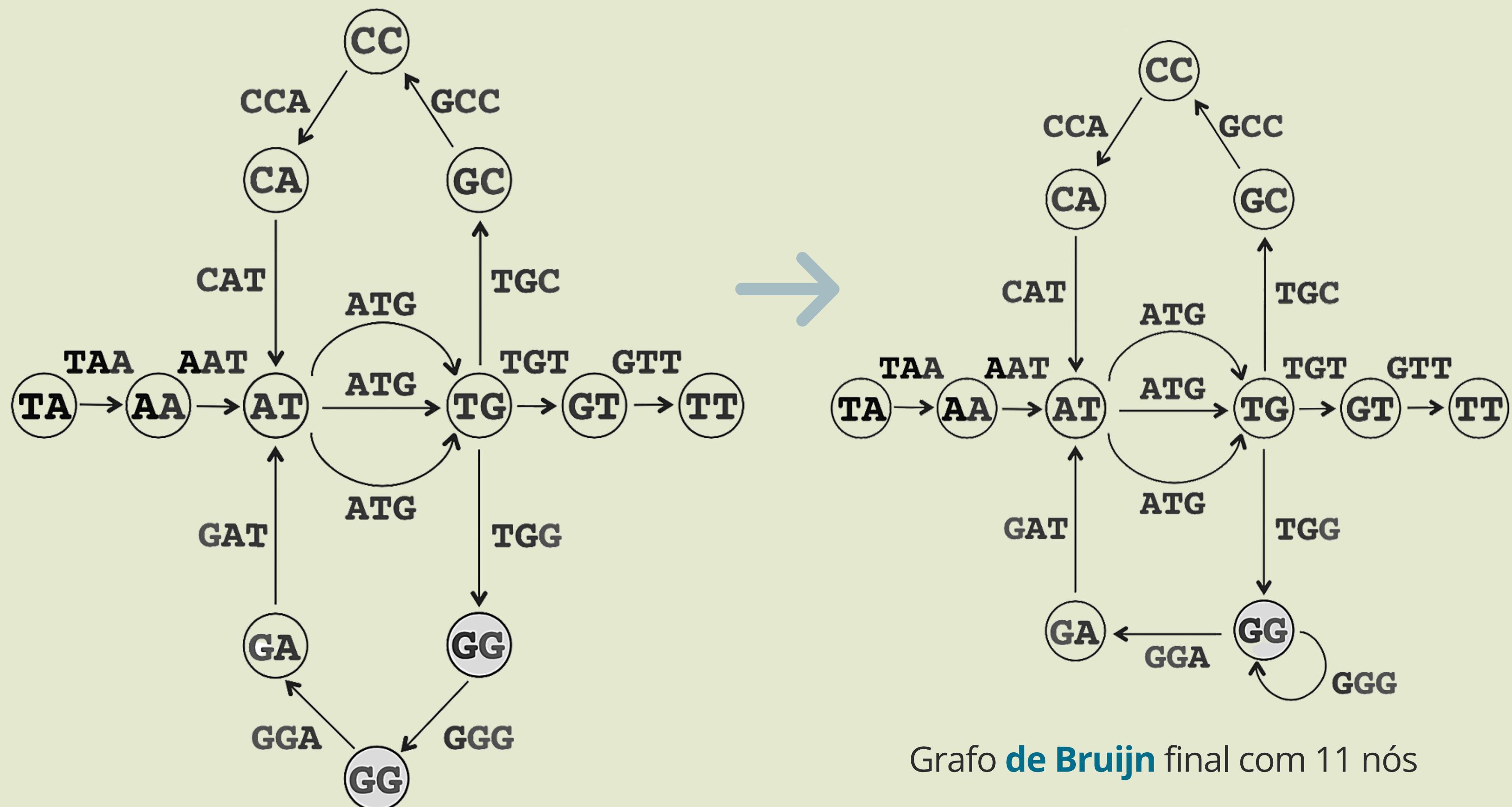


Nicolaas de Bruijn,  
Holanda, 1918-2012

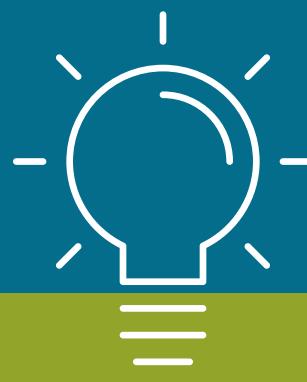


Fonte: Compeau, Phillip, and P. A. Pevzner. Bioinformatics Algorithms: An Active Learning Approach. La Jolla. CA: Active Learning Publishers, 2018.

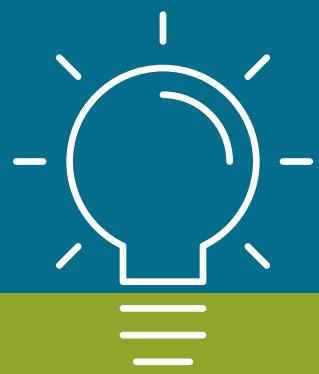




Grafo de Bruijn final com 11 nós



# Alguma idéia sobre como montar esse genoma?



# Caminho euleriano

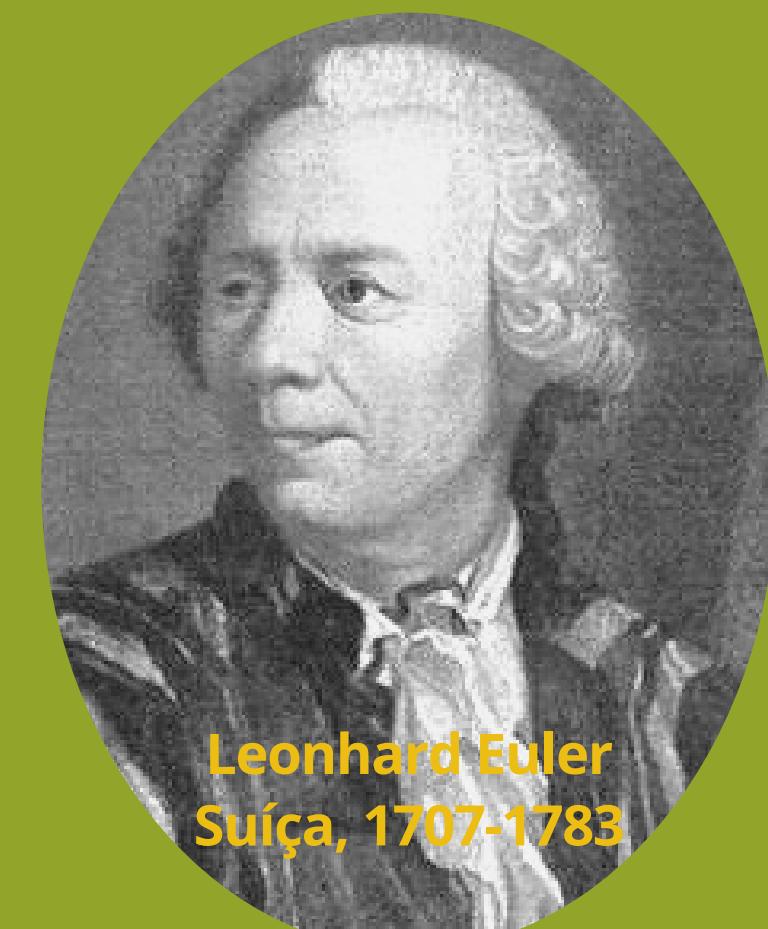
pronuncia-se "oileriano"

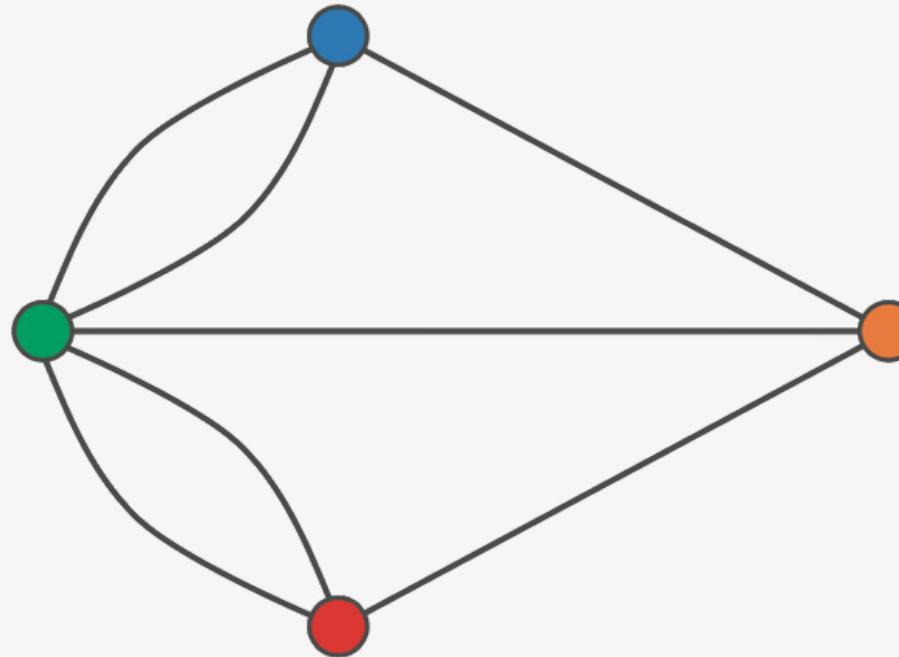
Um caminho euleriano em um grafo é um caminho que visita toda aresta exatamente uma vez

É um problema polinomial

Um grafo não direcionado tem um caminho euleriano se:

- todos os nós tem grau par
- apenas dois nós tem grau ímpar

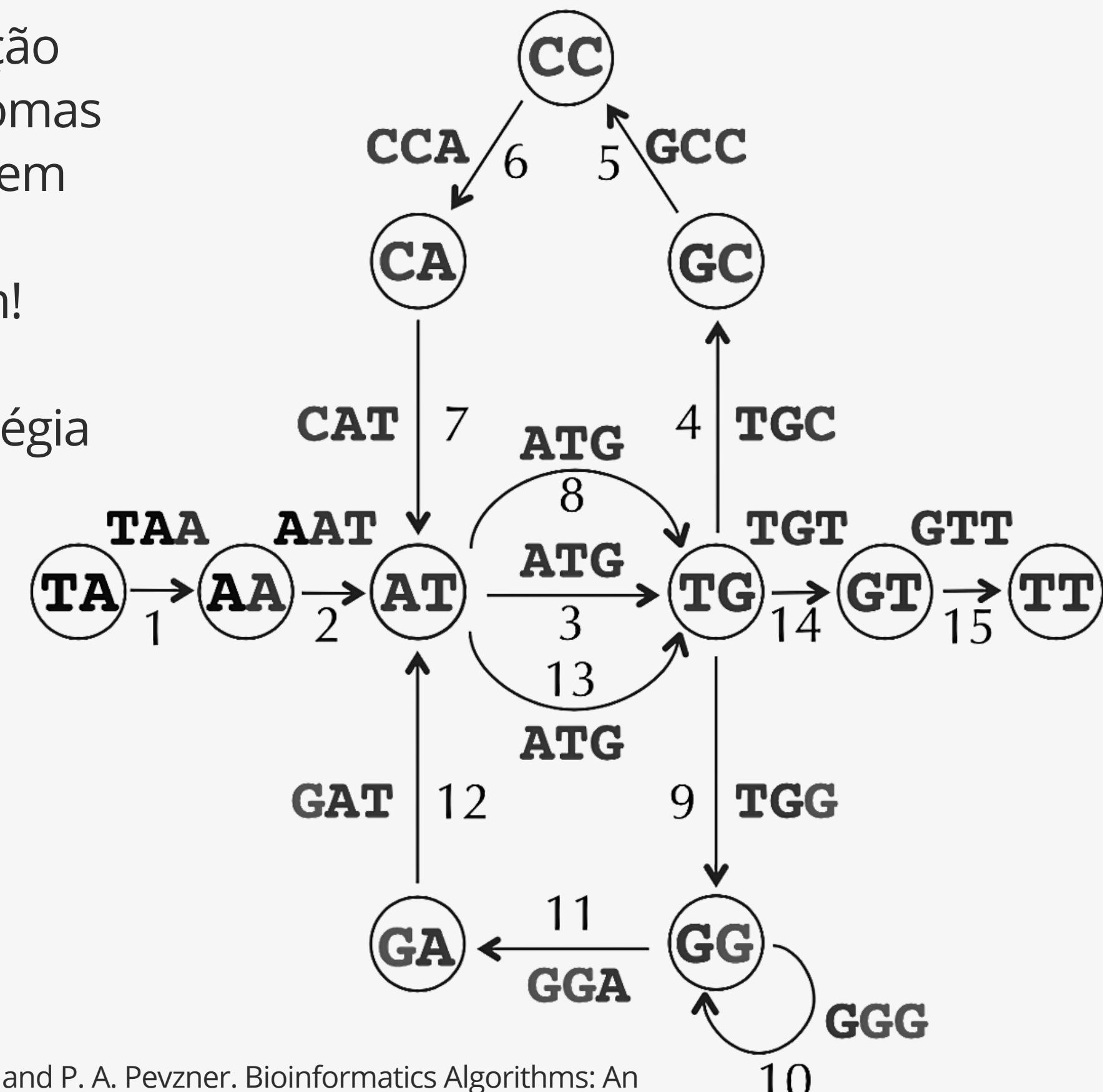




## As 7 pontes de Königsberg

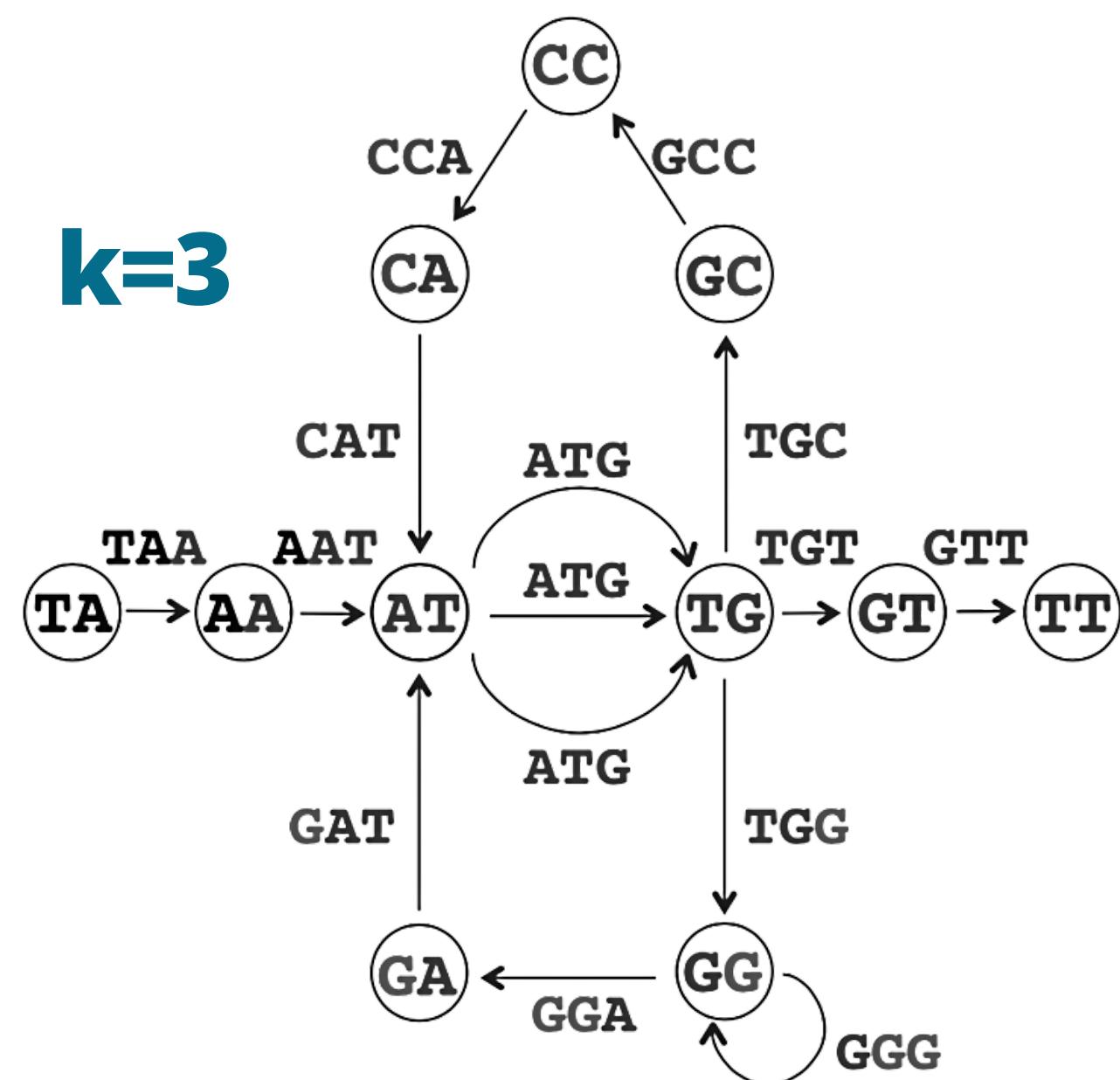
- A cidade (hoje Kaliningrado) era compreendida entre as margens do rio *Pregel* e duas ilhas fluviais tendo **sete pontes** conectando essas quatro partes da cidade
- É possível passear pela cidade e cruzar cada ponte exatamente uma vez e retornar ao ponto de partida?

- Nas duas primeiras décadas após a invenção dos métodos de sequenciamento, os genomas eram montados usando a primeira abordagem (grafos de sobreposição)
    - O genoma humano foi montado assim!
  - Atualmente, os montadores usam a estratégia do grafo de Bruijn

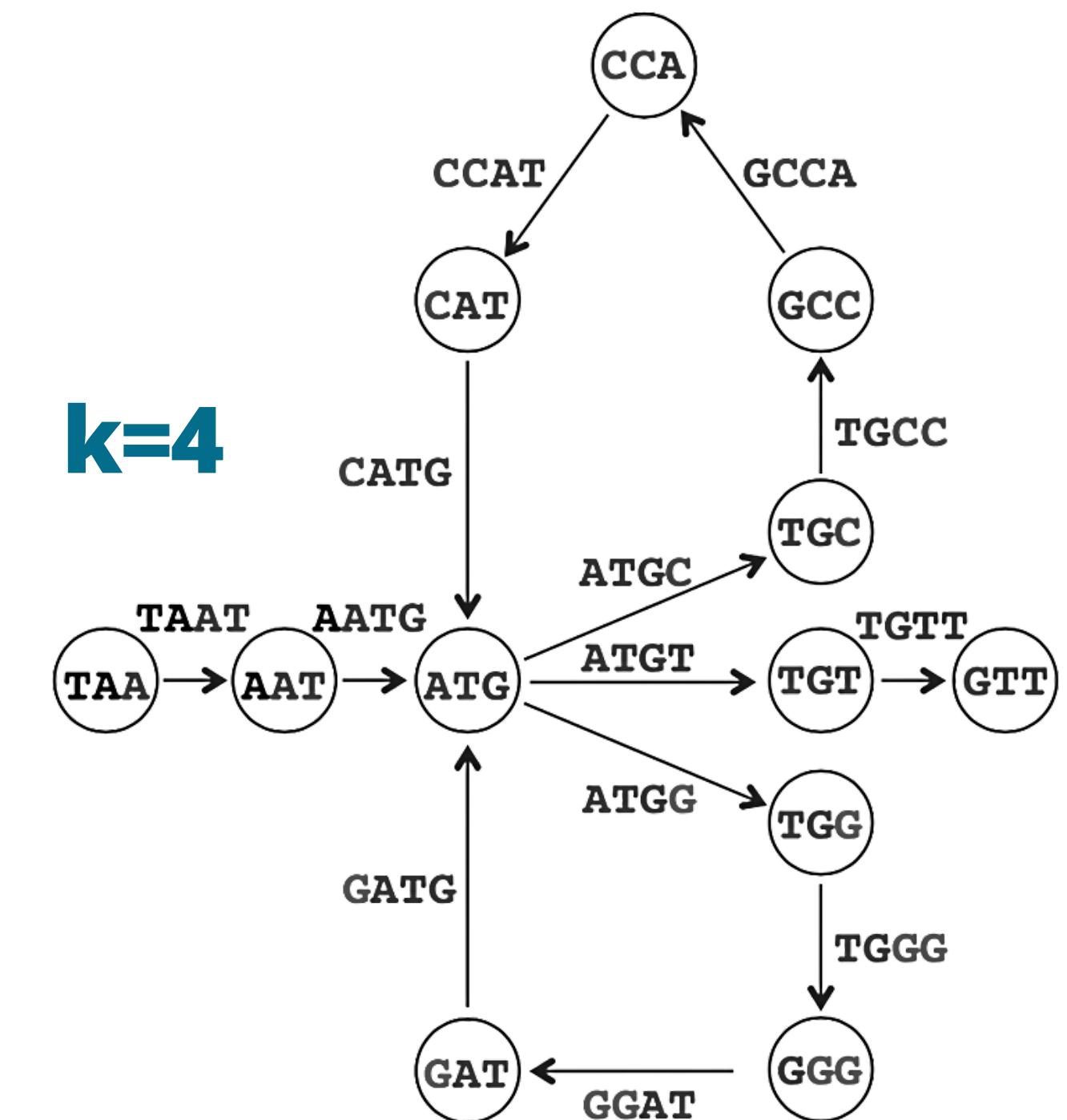
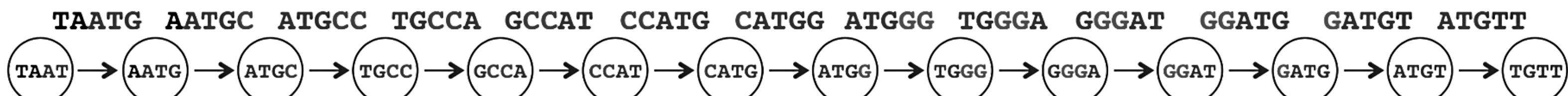


Fonte: Compeau, Phillip, and P. A. Pevzner. Bioinformatics Algorithms: An Active Learning Approach. La Jolla. CA: Active Learning Publishers, 2018.

# Aumento no k



k=5



# Na prática, há muitos 'desafios...'

- Há muitas **regiões repetidas**
- Quanto menor o *read*, mais difícil o trabalho
  - Técnicas de sequenciamento modernas tem **reads curtos** (300 nuclétideos)
  - Estratégias para **aumentar artificialmente os tamanhos dos reads**

$$k + d + k$$
$$3 + 1 + 3$$

TAATGCCATGGGATGTT

TAA-GCC  
AAT-CCA  
ATG-CAT  
TGC-ATG  
GCC-TGG  
CCA-GGG  
CAT-GGA  
ATG-GAT  
TGG-ATG  
GGG-TGT  
GGA-GTT

AAT-GAT  
GCC-TGG  
CAT-GGA  
TGC-ATG  
GGG-TGT  
ATG-CAT  
TGG-ATG

GGA-GTT

