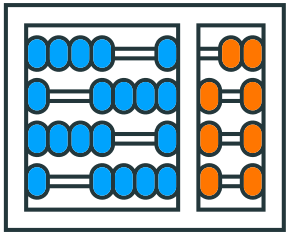


Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

# BLAST

## MO640 - Biologia Computacional / MC668 - Bioinformática

Zanoni Dias

2021

Instituto de Computação

# Roteiro

Banco de Dados de Sequências

BLAST

BLAST 2.0

# Banco de Dados de Sequências

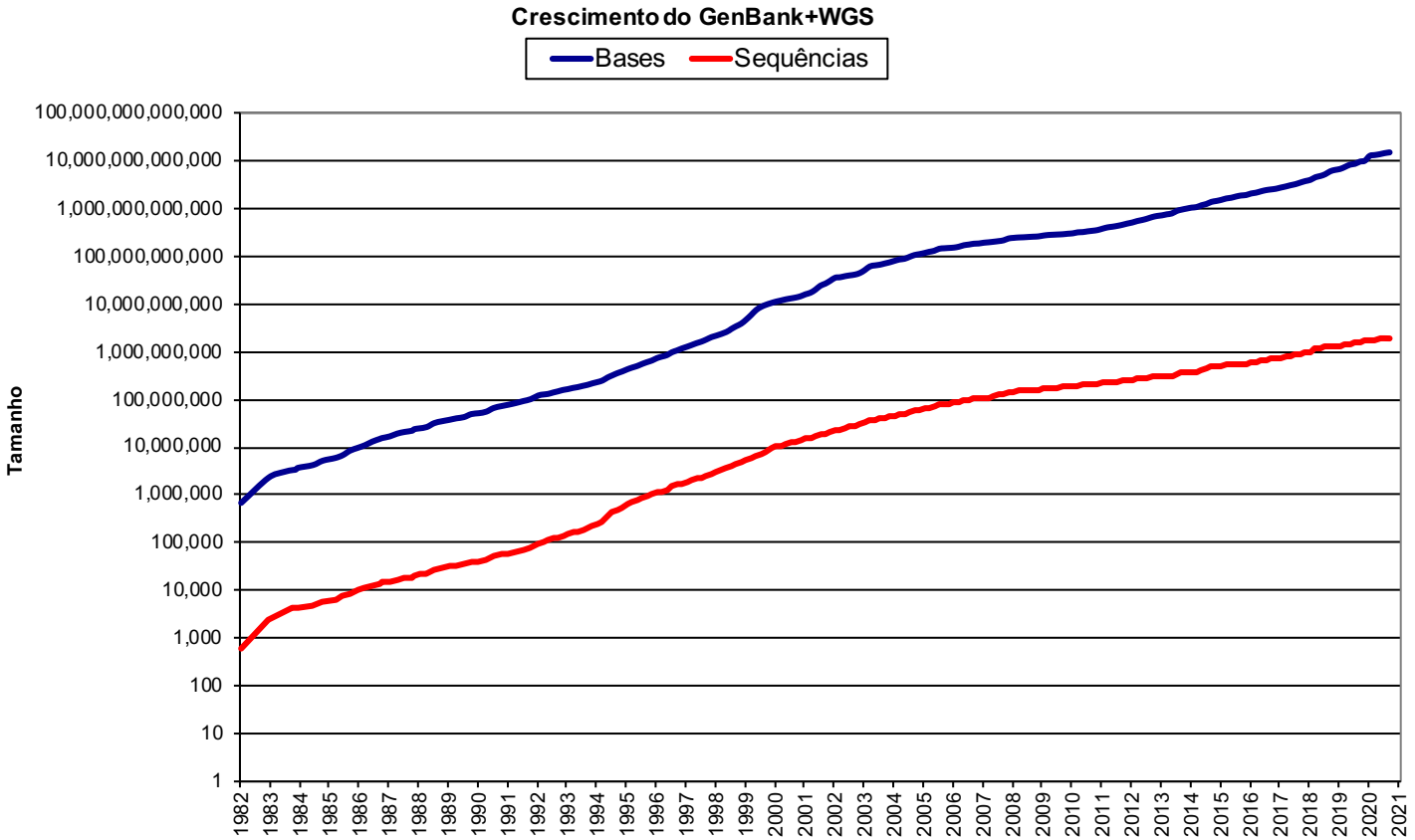
# GenBank

- Maior banco público de sequências de nucleotídeos e de proteínas do mundo.
- Fundado por Walter Goad em 1982.
- Desenvolvido pelo National Center for Biotechnology Information (NCBI) e financiado pelo National Institutes of Health (NIH).
- Crescimento do GenBank:
  - *From 1982 to the present, the number of bases in GenBank has doubled approximately every 18 months.*

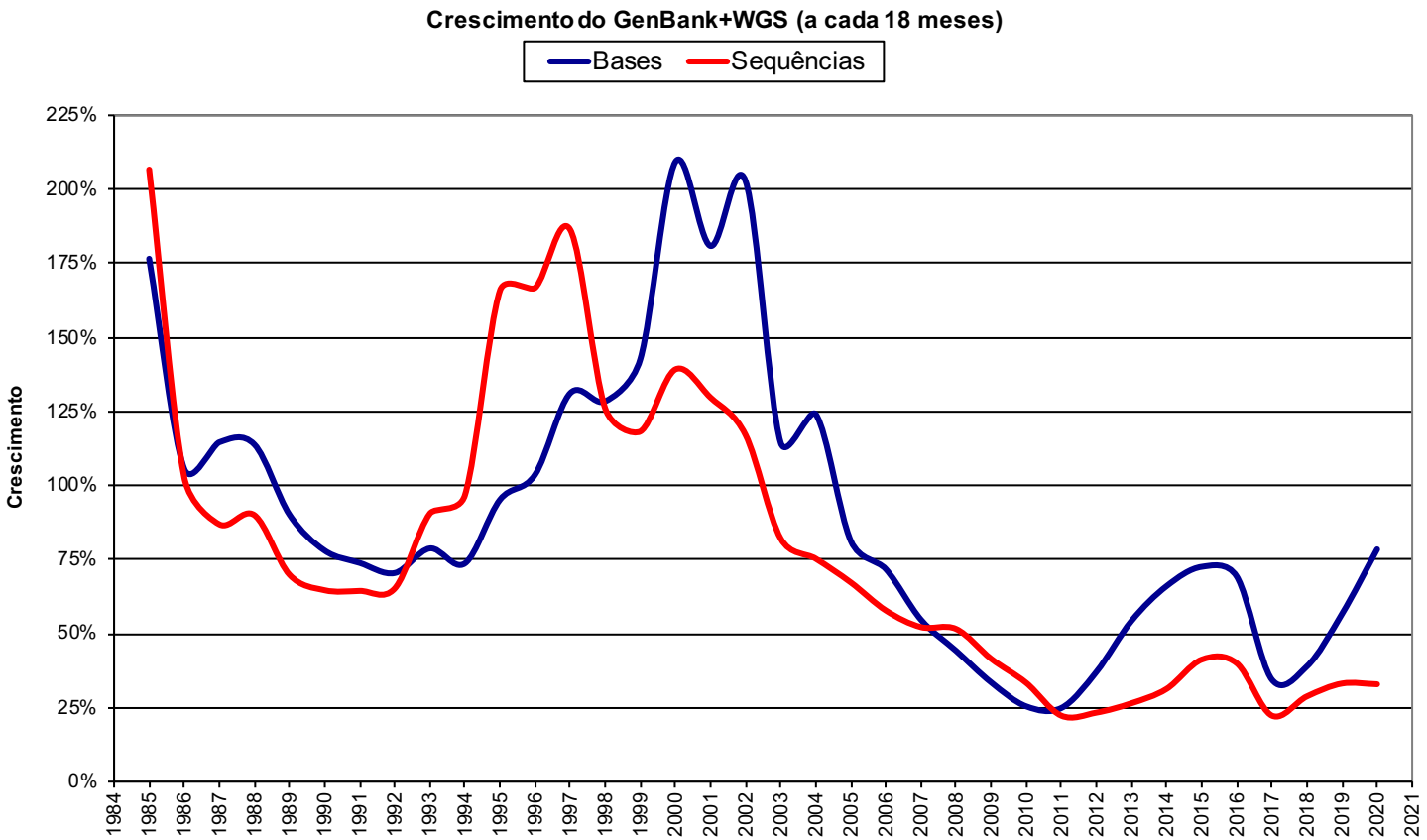
# WGS (Whole Genome Shotgun)

- Com a simplificação, automatização e barateamento do processo de sequenciamento, o número de genomas completamente ou parcialmente sequenciados cresceu muito nas últimas duas décadas.
- Dois importantes marcos da genômica:
  - Nacional: em 2000, sequenciamento da bactéria *Xylella fastidiosa* (2.7Mbp, 3 mil genes), causadora da doença “amarelinho” que afeta laranjeiras.
  - Internacional: em 2001, sequenciamento do genoma humano (3Gbp, 30 mil genes).
- Desde 2002 o GenBank possui uma divisão (WGS - Whole Genome Shotgun) dedicado a montagens completas ou parciais de genomas.
- A proporção de bases e sequências do WGS em relação ao GenBank vem crecendo muito nos últimos anos.

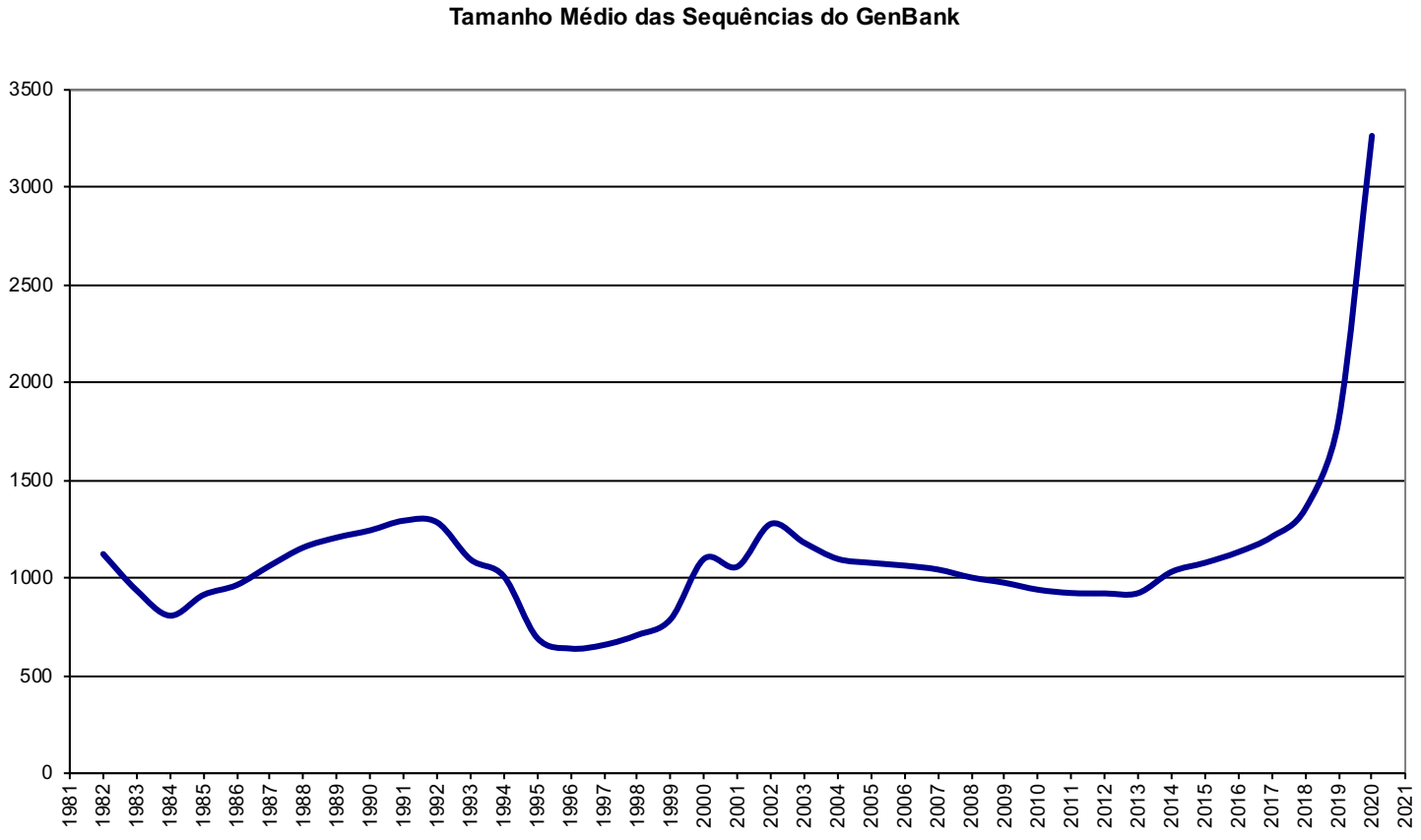
# GenBank+WGS


Crescimento do GenBank+WGS
Bases
Sequências
Tamanho
100,000,000,000,000
10,000,000,000,000
1,000,000,000,000
100,000,000,000
10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

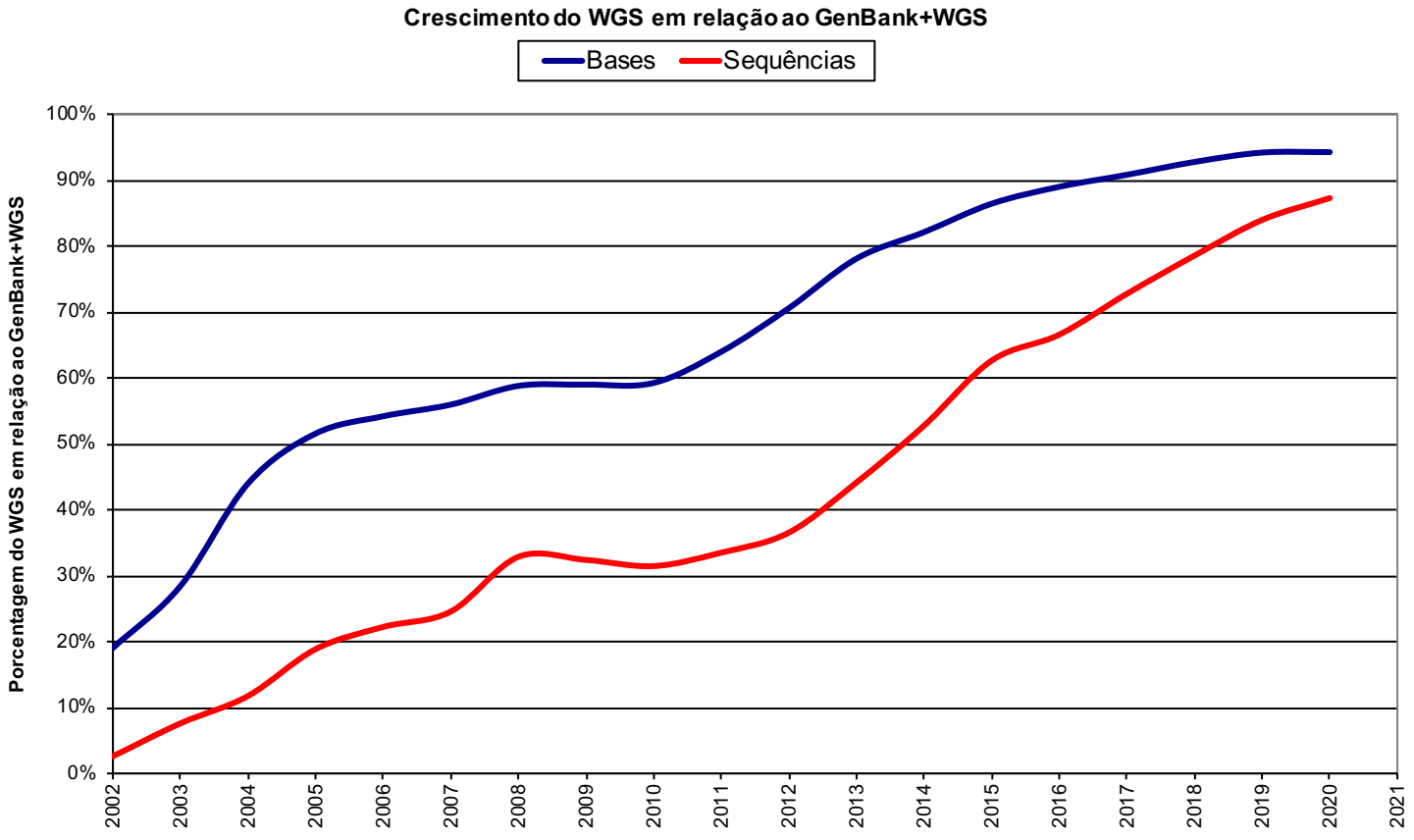
# GenBank+WGS


Crescimento do GenBank+WGS (a cada 18 meses)
Bases
Sequências
Crescimento
225%
200%
175%
150%
125%
100%
75%
50%
25%
0%
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

# GenBank+WGS


Tamanho Médio das Sequências do GenBank
3500
3000
2500
2000
1500
1000
500
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

# WGS × (GenBank+WGS)



# UniProt

- UniProt: Universal Protein Resource.
- Consórcio criado em 2002 envolvendo:
  - Swiss-Prot: Swiss Institute of Bioinformatics (SIB) e European Bioinformatics Institute (EBI). Maior banco manualmente curado de proteínas do mundo.
  - TrEMBL: Swiss Institute of Bioinformatics (SIB) e European Molecular Biology Laboratory (EMBL-EBI). Banco de proteínas gerado computacionalmente pela tradução dos dados do EMBL Nucleotide Sequence Database.
  - PIR: Georgetown University Medical Center (GUMC). Conjunto de banco de dados de proteínas criados para auxiliar a análise genômica e proteômica.

# Alinhamento de uma sequência contra todas do GenBank

- Exemplo: computador de 3GHz (1 instrução por ciclo)
  - n = 10:
    - Tempo: $(10 \times 10^{13})/(3 \times 2^{30})$ = 9 minutos
  - n = 100:
    - Tempo: $(100 \times 10^{13})/(3 \times 2^{30})$ = 4 dias
  - n = 1000:
    - Tempo: $(1000 \times 10^{13})/(3 \times 2^{30})$ = 36 dias
  - n = 10000:
    - Tempo: $(10000 \times 10^{13})/(3 \times 2^{30})$ = 1 ano
  - n = 100000:
    - Tempo: $(100000 \times 10^{13})/(3 \times 2^{30})$ = 1 década
  - n = 1000000:
    - Tempo: $(1000000 \times 10^{13})/(3 \times 2^{30})$ = 1 século
  - n = 10000000:
    - Tempo: $(10000000 \times 10^{13})/(3 \times 2^{30})$ = 1 milênio

# BLAST

# BLAST

- BLAST: Basic Local Alignment Search Tool.
- Ferramenta proposta por Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers e David Lipman em 1990.
- Desenvolvido pelo National Center for Biotechnology Information (NCBI) e financiado pelo National Institutes of Health (NIH).
- Heurística para alinhamento local: não garante a obtenção do alinhamento local ótimo.
- Possui uma forte base estatística.
- Site oficial:
  - `https://blast.ncbi.nlm.nih.gov`
- O artigo original do BLAST foi o artigo mais citado da década de 1990 e hoje em dia possui mais de 95000 citações.

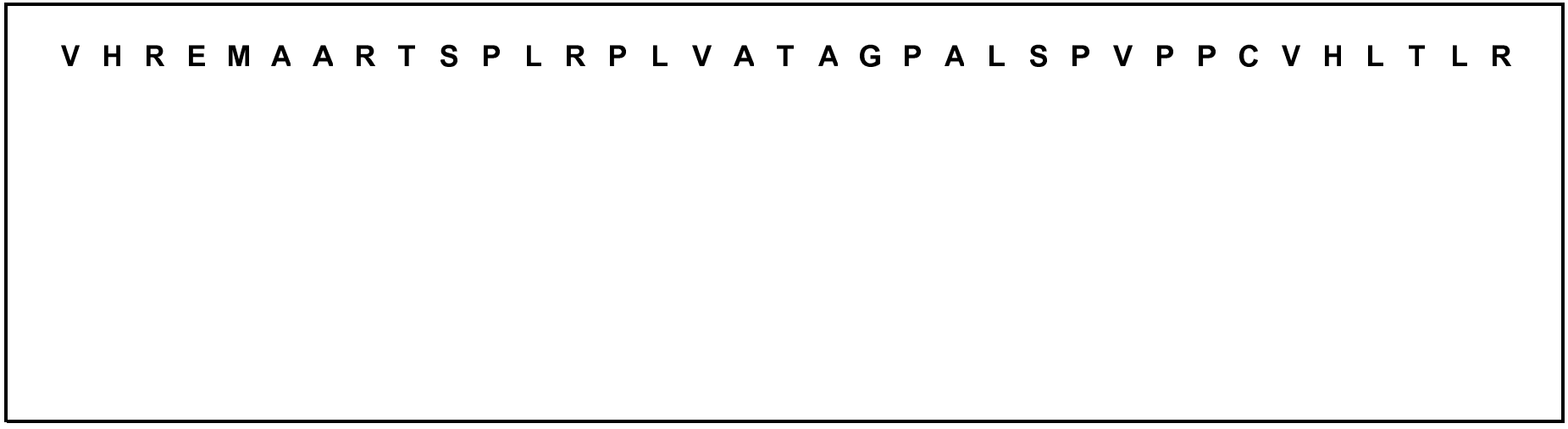
# BLAST

- Nomenclatura:
  - *query*: sequência que será comparada.
  - *database*: banco de sequências.
  - *HSP*: high-scoring sequence pair, par de subsequências com alta similariedade.
  - *seed*: sequência curta utilizada para iniciar um alinhamento.
  - *hit*: alinhamento com similariedade maior que a mínima.
- Passos básicos:
  - Obter uma lista de *seeds* da *query*.
  - Procurar *hits* de *seeds* com as sequências do banco de dados.
  - Estender os *hits* para obter os alinhamentos.

# BLAST

- Remover regiões de baixa complexidade da *query* (regiões com poucos tipos de elementos). Estas regiões são marcadas como subsequências de Xs (para sequências protéicas) e de Ns (para sequências de DNA).
- Construir uma lista de sementes (*seeds*) com todas as sequências de tamanho $w$ que possuam pontuação pelo menos $T$ quando alinhadas com a *query*.
- Geralmente os parâmetros $w$ e $T$, sob algum esquema de pontuação específico, são ajustado para se obter uma lista de sementes (*seeds*) cerca de 50x maior que o tamanho da *query*.
- Em geral, $w \geq 3$ para proteínas e $w \geq 11$ para sequências de DNA.
- A escolha de uma matriz de pontuação adequada ($PAM_{120}$, $BLOSUM_{62}$, etc) é fundamental nesta fase.

# BLAST - Criação da Lista de *Seeds*



# BLAST - Criação da Lista de *Seeds*

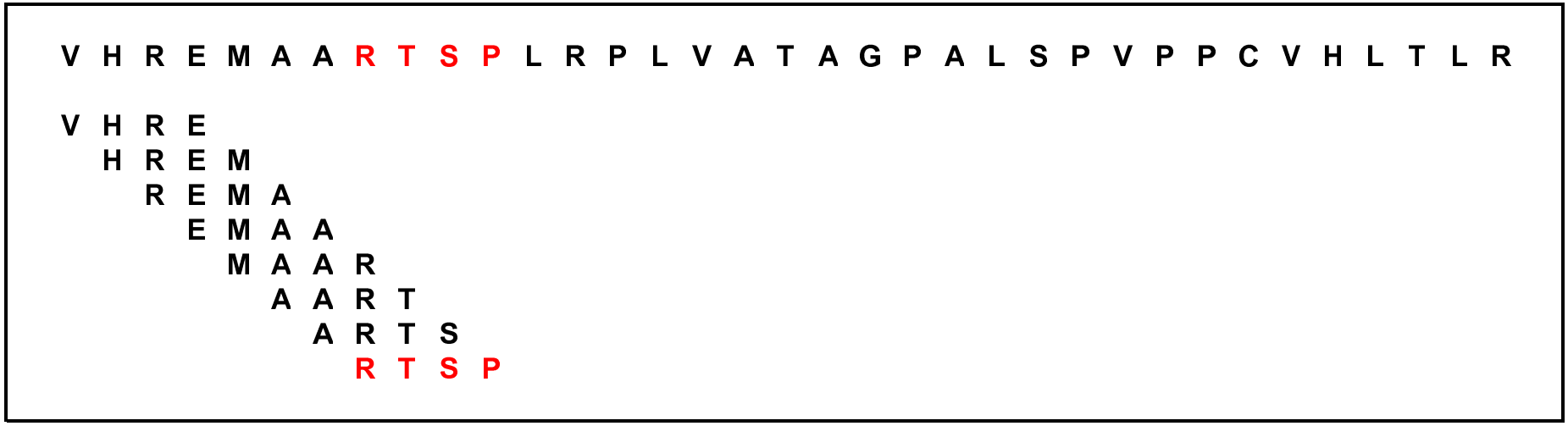
V H R E M A A R T S P L R P L V A T A G P A L S P V P P C V H L T L R

V H R E

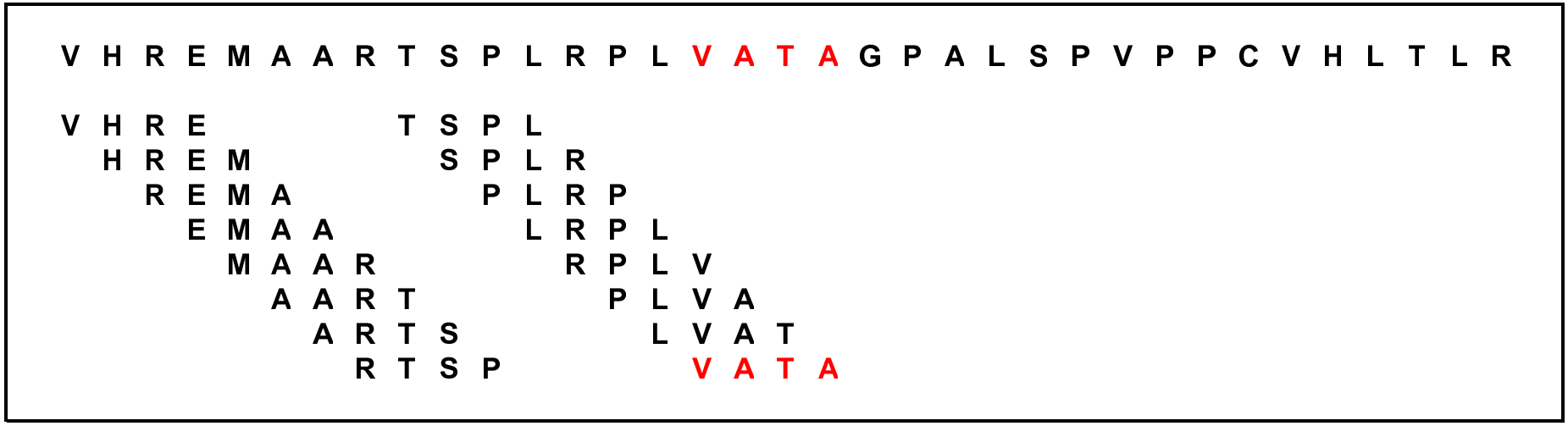
# BLAST - Criação da Lista de *Seeds*

# BLAST - Criação da Lista de *Seeds*

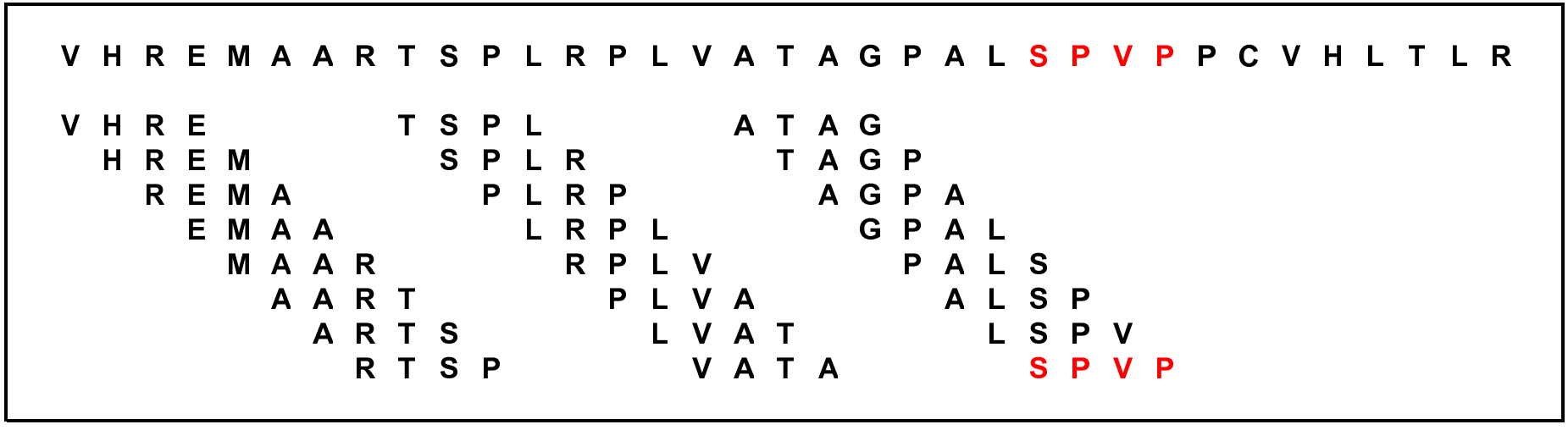
# BLAST - Criação da Lista de *Seeds*



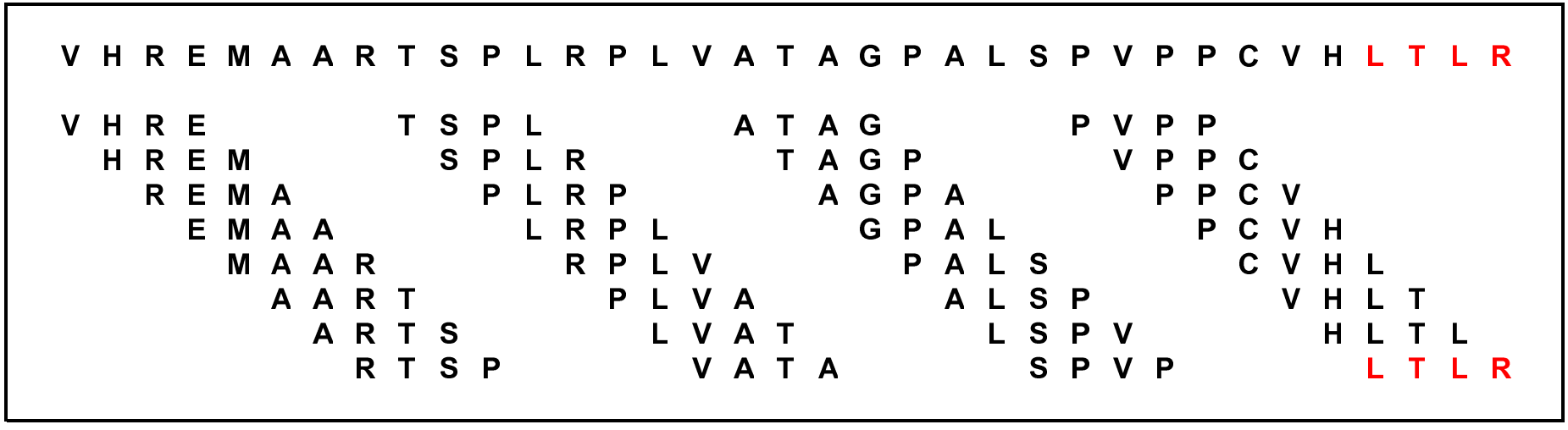
# BLAST - Criação da Lista de *Seeds*



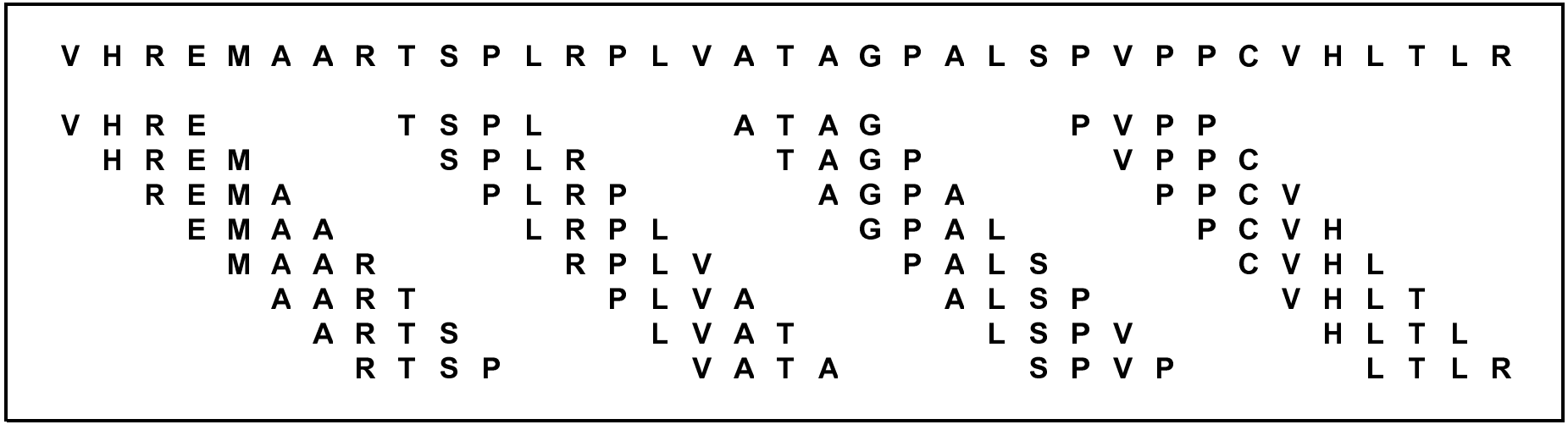
# BLAST - Criação da Lista de *Seeds*



# BLAST - Criação da Lista de *Seeds*



# BLAST - Criação da Lista de *Seeds*



# BLAST - Criação da Lista de *Seeds*

# BLAST - Criação da Lista de *Seeds*

V H R E M A A R T S P L R P L V A T A G P A L S P V P P C V H L T L R

E M A A = 18

# BLAST - Criação da Lista de *Seeds*

V H R E M A A R T S P L R P L V A T A G P A L S P V P P C V H L T L R

E M A A = 18
A A A C = 6

# BLAST - Criação da Lista de *Seeds*

V H R E M A A R T S P L R P L V A T A G P A L S P V P P C V H L T L R

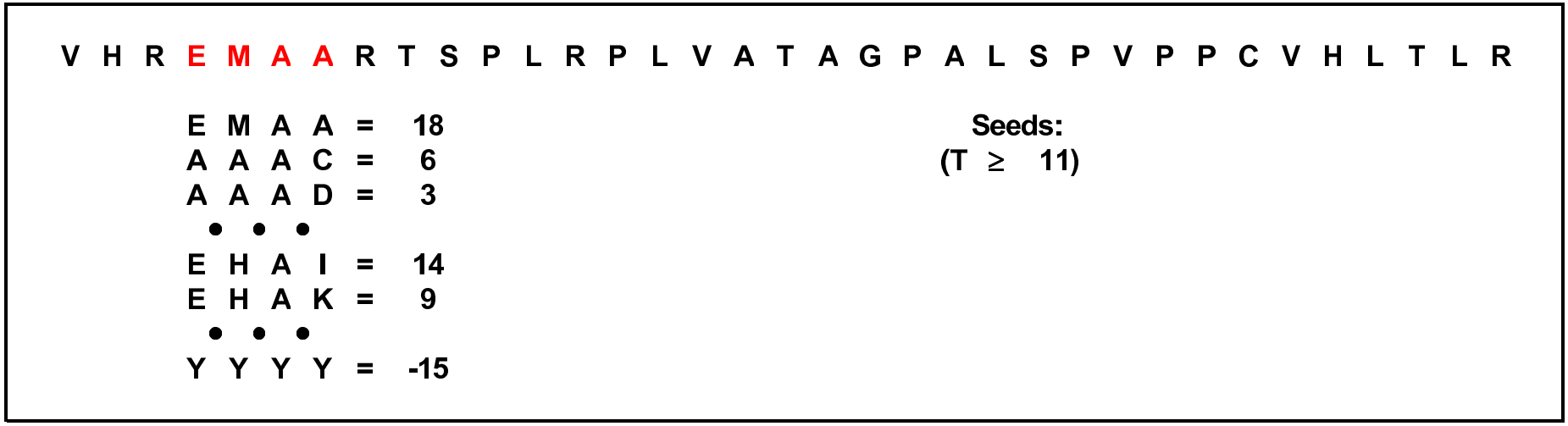
E M A A = 18
A A A C = 6
A A A D = 3

# BLAST - Criação da Lista de *Seeds*

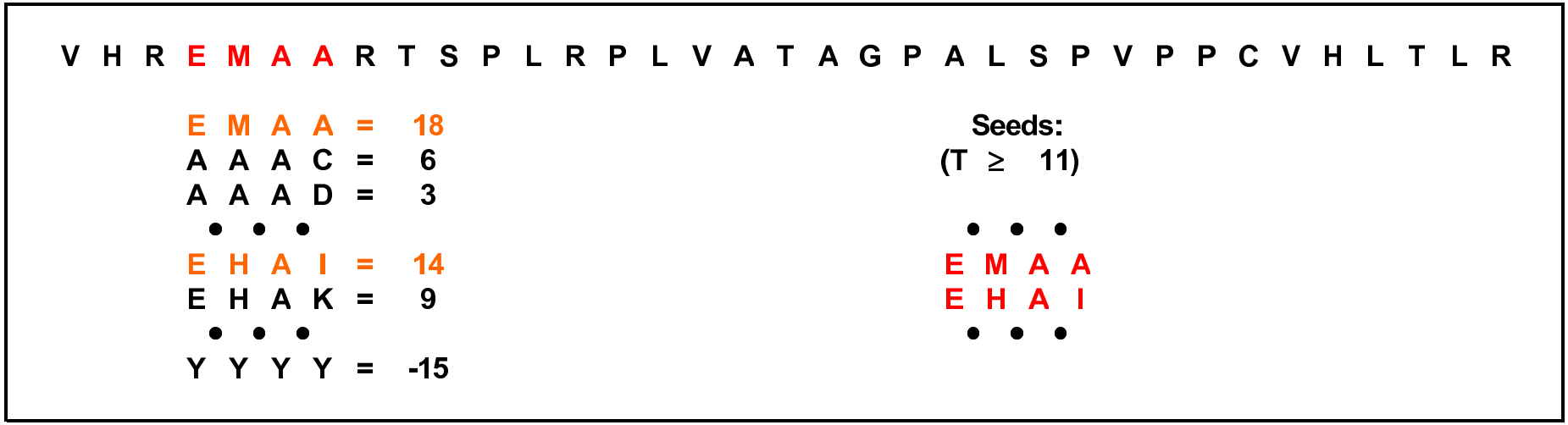
V H R E M A A R T S P L R P L V A T A G P A L S P V P P C V H L T L R

E M A A = 18
A A A C = 6
A A A D = 3
• • •
E H A I = 14
E H A K = 9
• • •
Y Y Y Y = -15

# BLAST - Criação da Lista de *Seeds*



# BLAST - Criação da Lista de *Seeds*



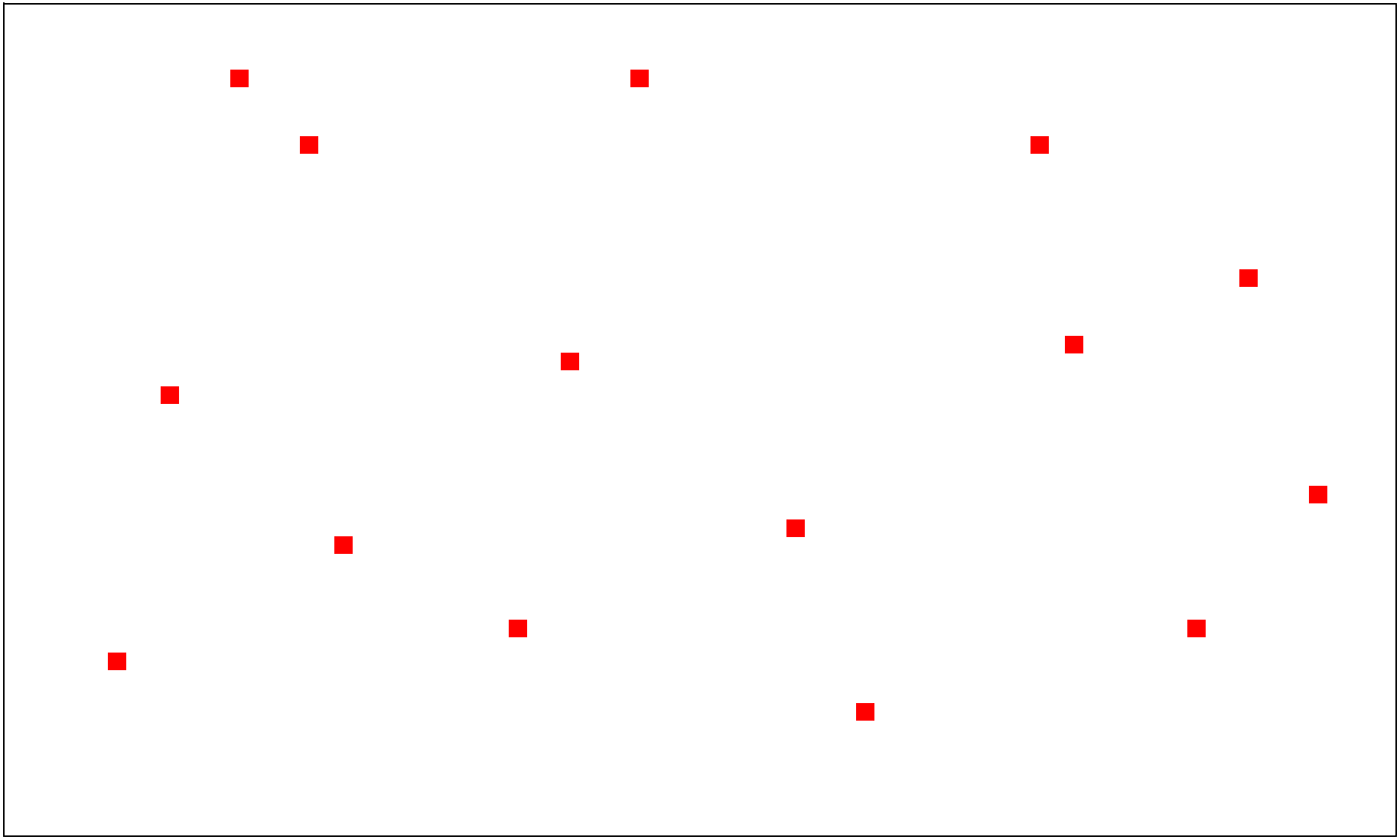
# BLAST - Busca por *Hits*

- Duas opções de busca:
  - Para cada *seed*, buscar *hits* em cada uma das sequências do banco de dados.
  - Para cada sequência do banco de dados, buscar *hits* com cada uma das *seeds*.
- Duas opções de estrutura de dados para auxiliar a busca:
  - Construção de um vetor, onde cada posição representa uma sequência protéica de tamanho *w*. A *i*-ésima posição do vetor armazena uma lista de todas ocorrências de *hits* da *i*-ésima sequência na *query*.
    - Poucas posições deste vetor armazenam informações úteis.
    - Alternativa: armazenar as informações num *hash*.
  - Construção de uma máquina de estados, usando autômatos finitos, onde cada estado representa a última palavra lida, e as transições de estados ocorrem a cada leitura de uma nova base da sequência do banco onde se está buscado por *hits*.
- Geralmente usa-se autômatos finitos para buscar todos as *seeds* em cada uma das sequência do banco (uma por uma).

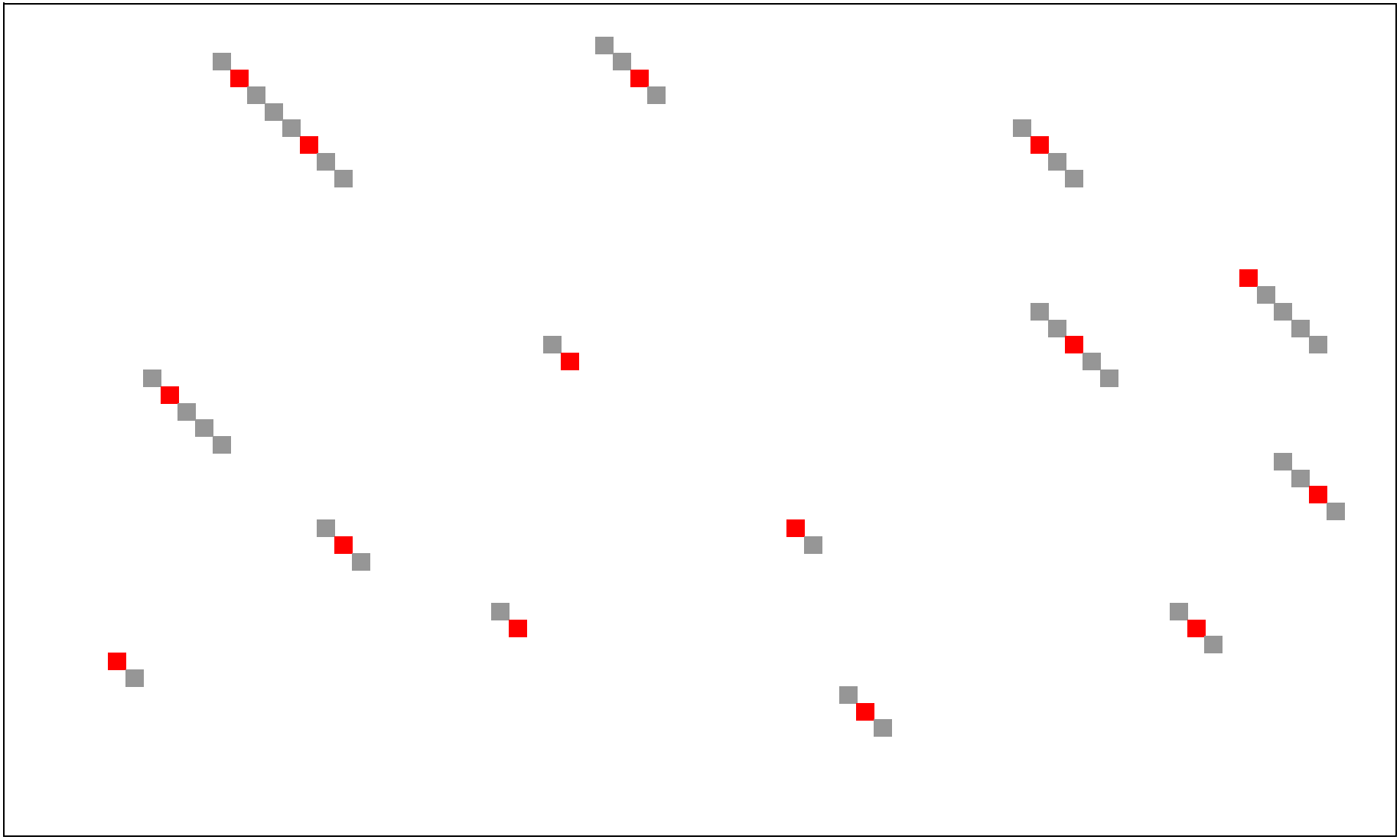
# BLAST - Obtenção dos *HSPs*

- Estende-se o *hit* em ambas as direções, apenas considerando alinhamento sem buracos.
- A extensão é interrompida após se distanciar muito do melhor alinhamento obtido até então.
- Por exemplo, para proteínas, o valor da distância máxima é 20. Este valor garante que a probabilidade deste método perder um alinhamento de maior pontuação é de cerca de 0,1%.
- Apenas *HSPs* com pontuação maior ou igual a um limiar *S* são apresentados como respostas.
- Estima-se que 90% do tempo de processamento é gasto nesta etapa.
- A performance do algoritmo nesta fase está intimamente relacionada a escolha dos parâmetros *w* e *T*.
    - Quanto maior for o valor de *w*, maior o número de *seeds* a se considerar.
    - Quanto maior for o valor de *T*, mais restrita será a busca por *HSPs*.

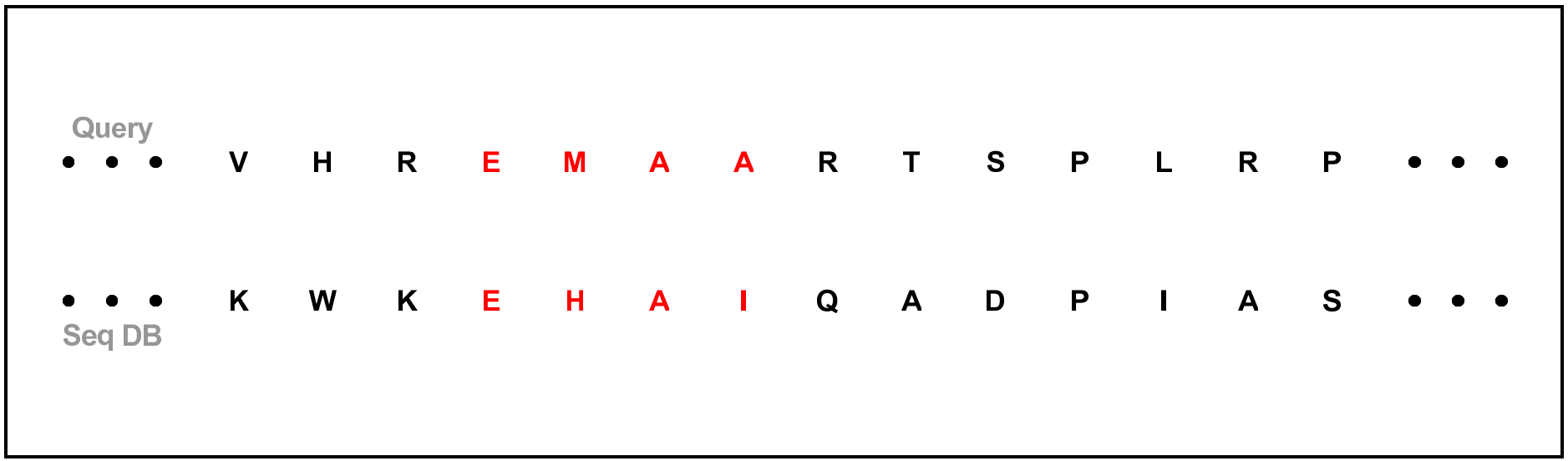
# BLAST - Obtenção dos *HSPs*



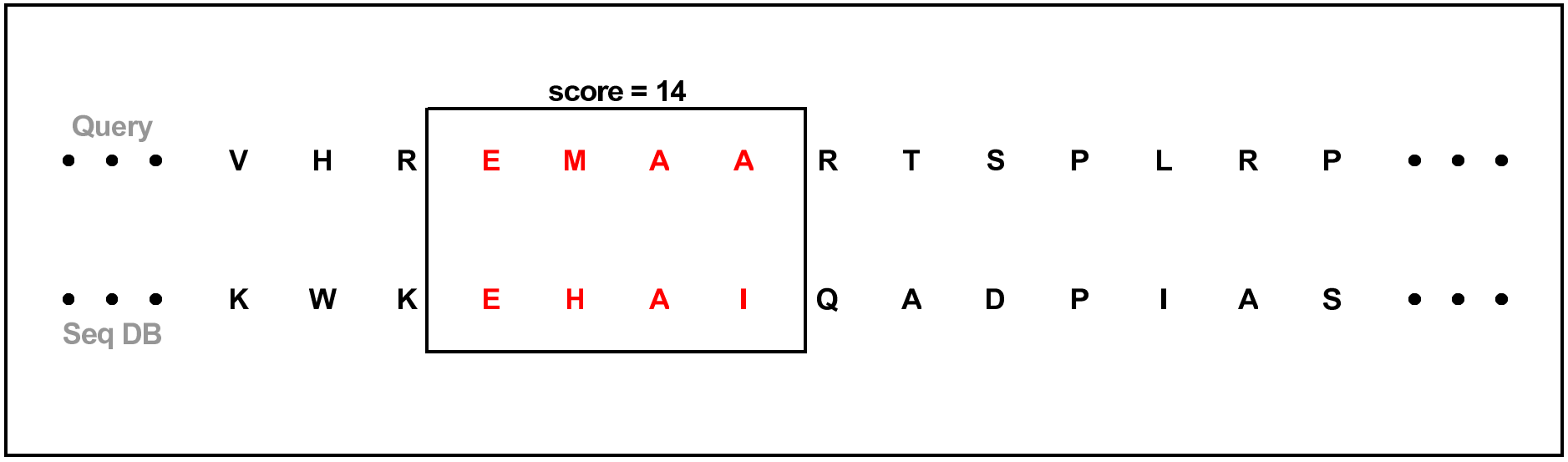
# BLAST - Obtenção dos *HSPs*



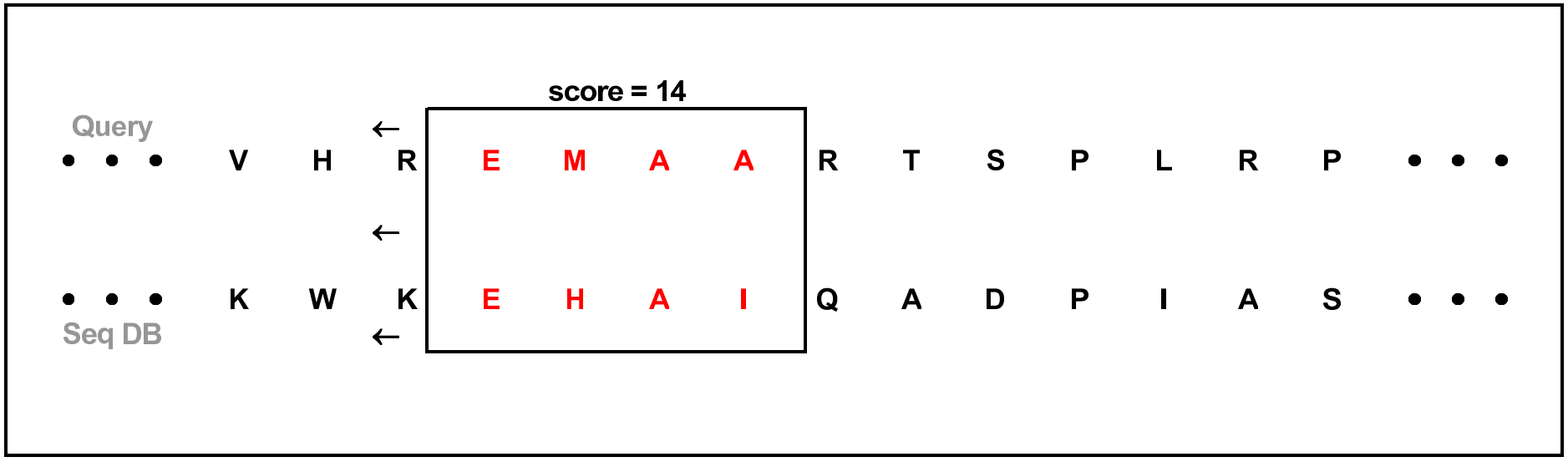
# BLAST - Obtenção dos *HSPs*



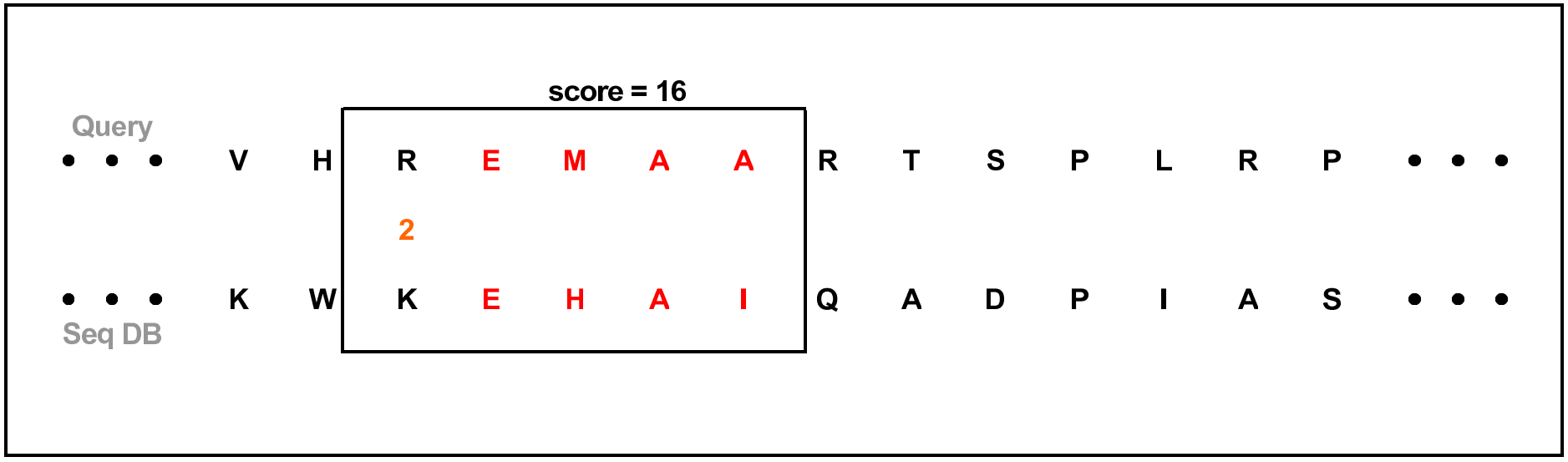
# BLAST - Obtenção dos *HSPs*



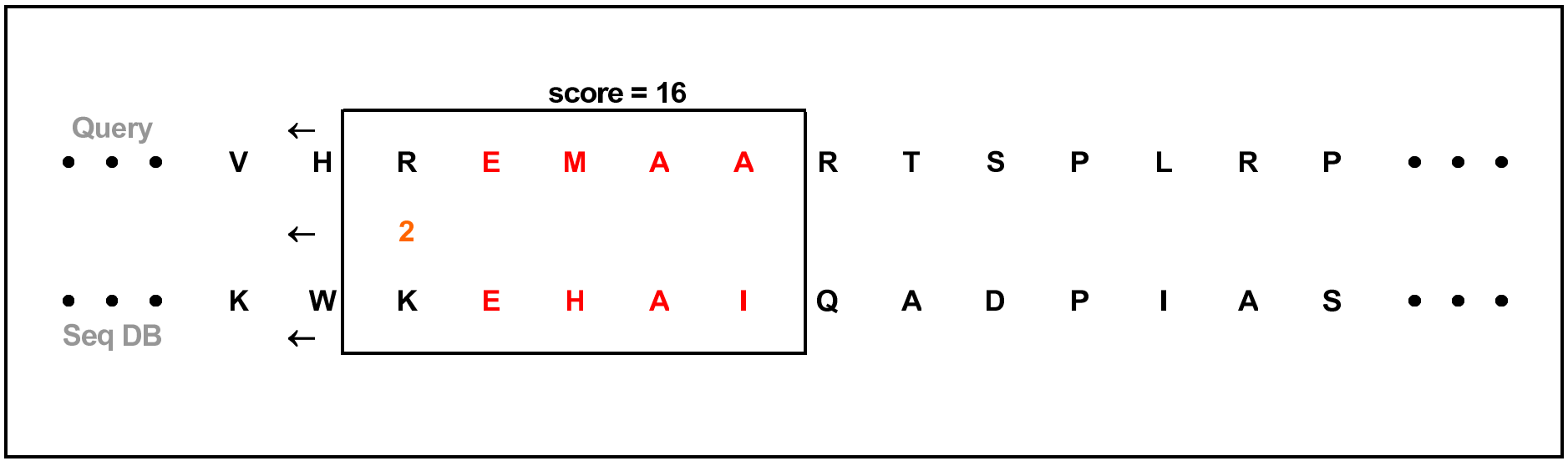
# BLAST - Obtenção dos *HSPs*



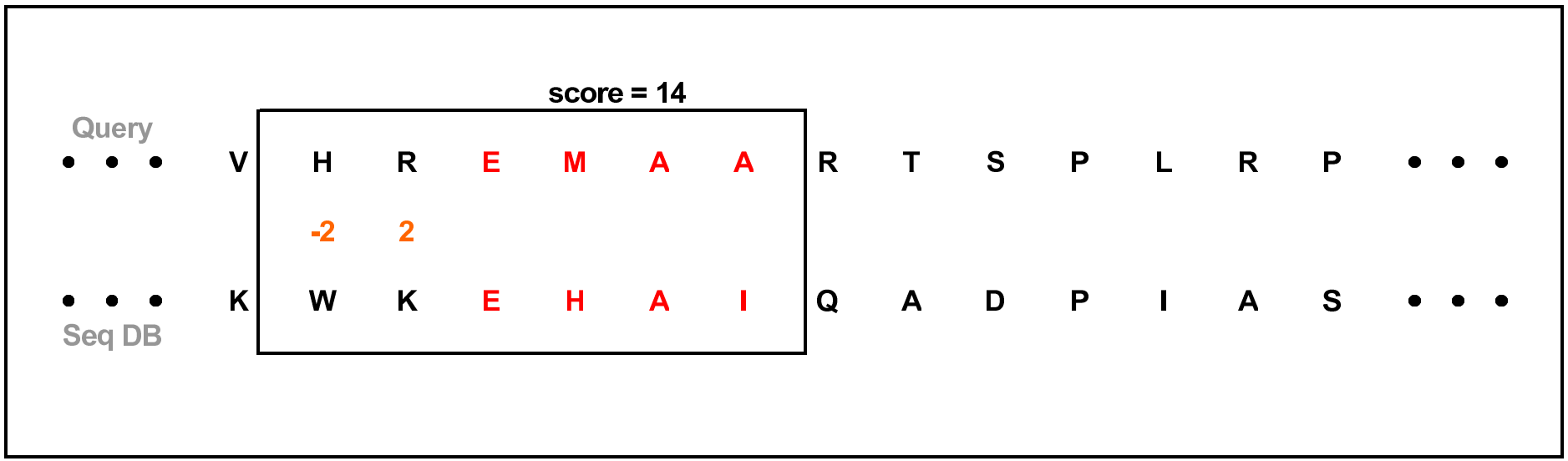
# BLAST - Obtenção dos *HSPs*



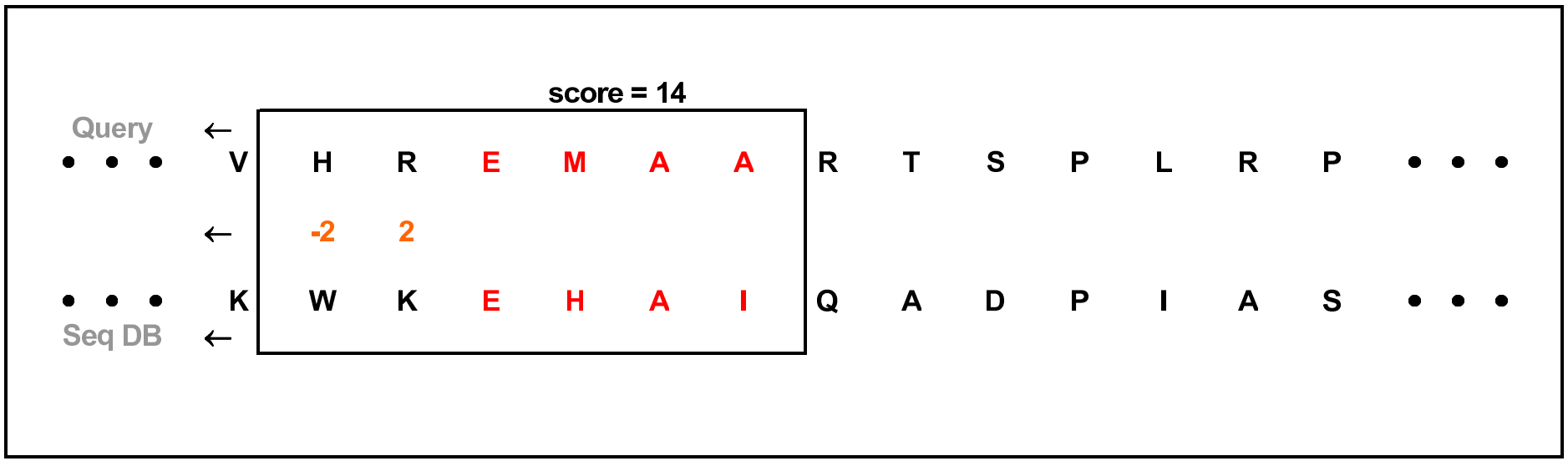
# BLAST - Obtenção dos *HSPs*



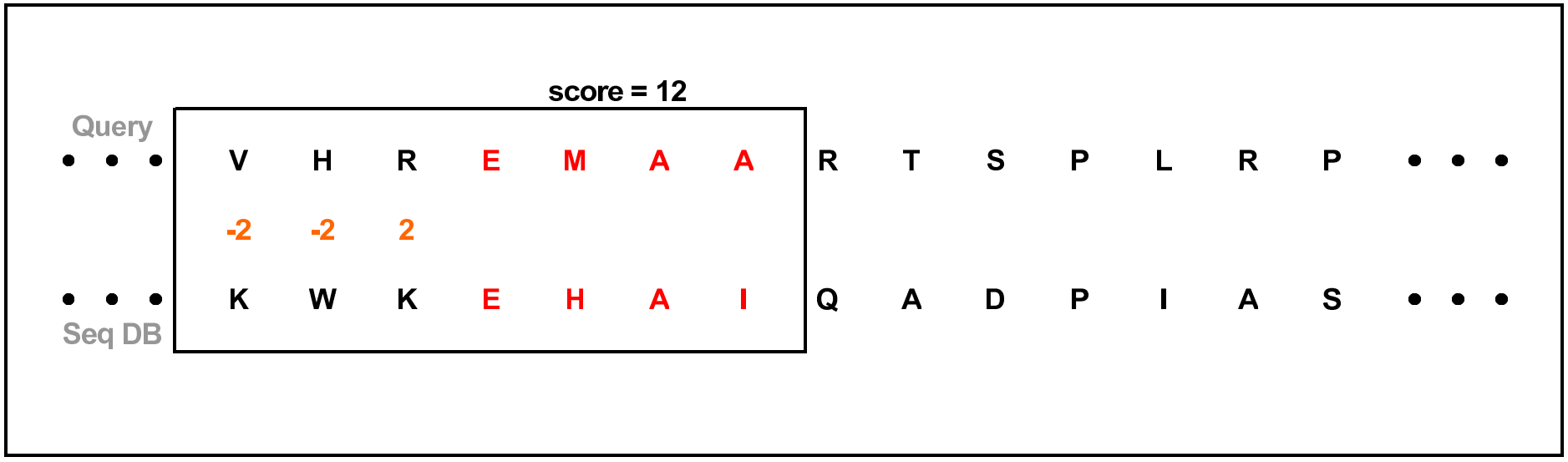
# BLAST - Obtenção dos *HSPs*



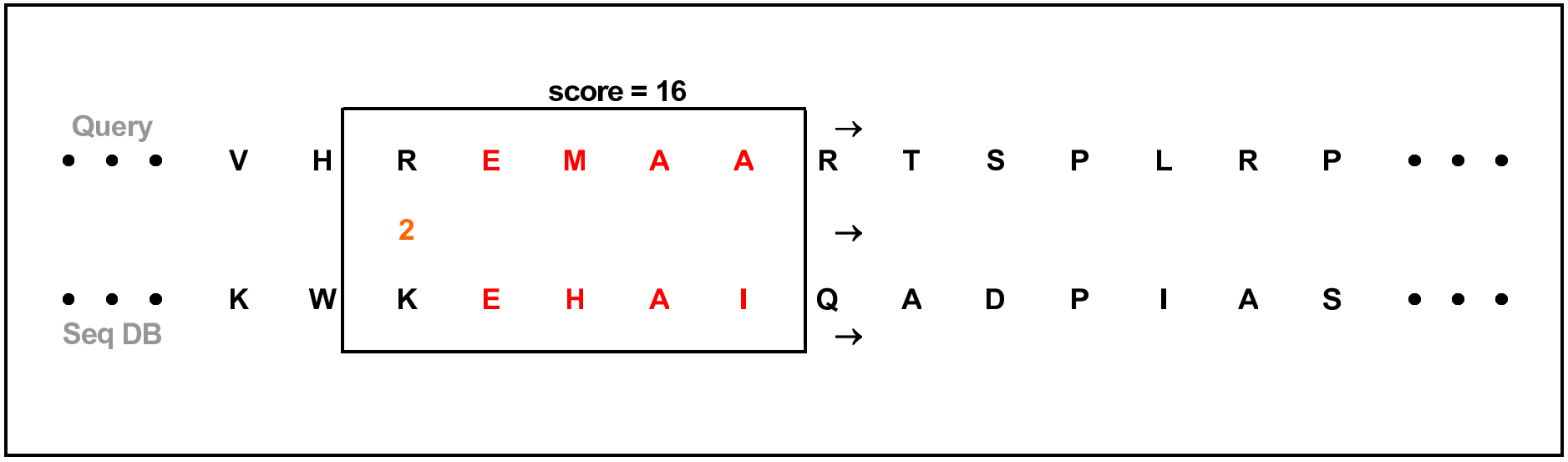
# BLAST - Obtenção dos *HSPs*



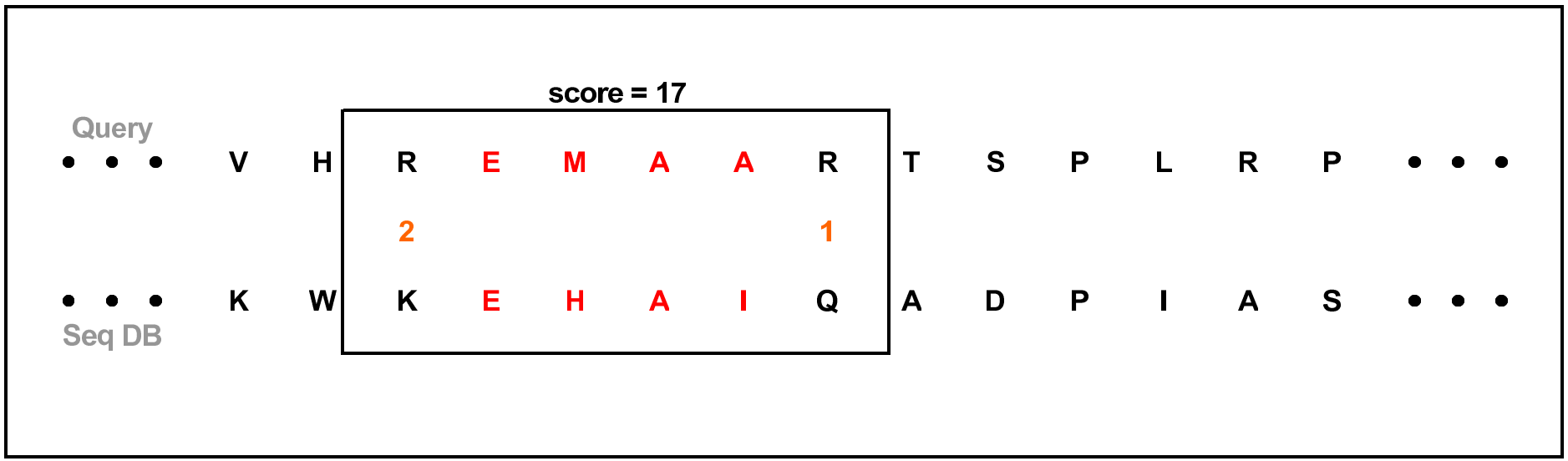
# BLAST - Obtenção dos *HSPs*



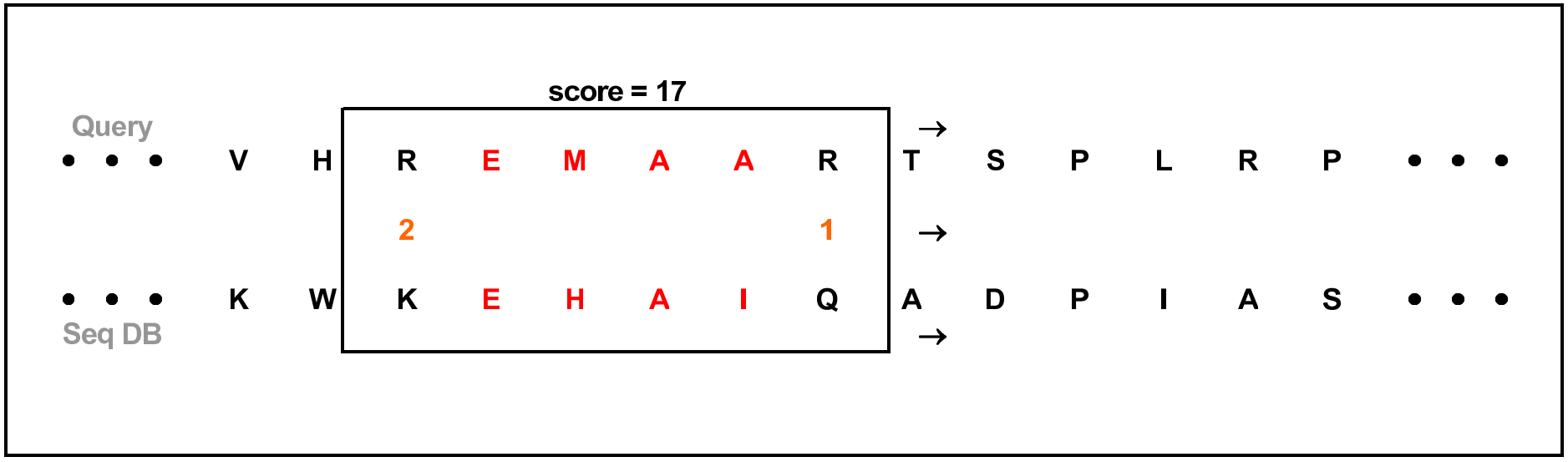
# BLAST - Obtenção dos *HSPs*



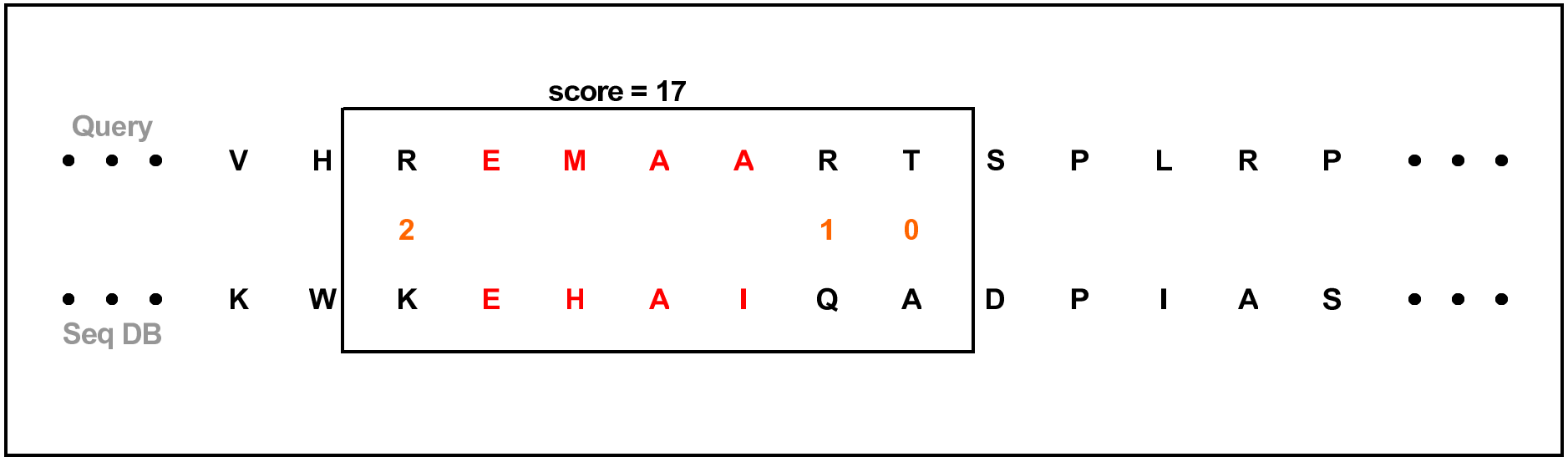
# BLAST - Obtenção dos *HSPs*



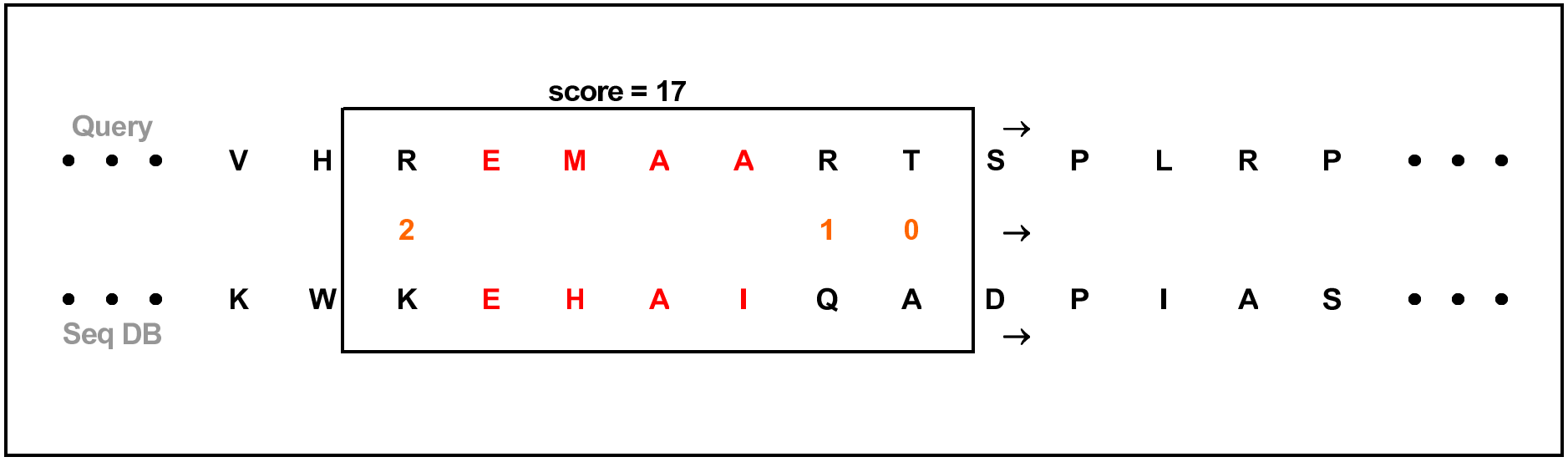
# BLAST - Obtenção dos *HSPs*



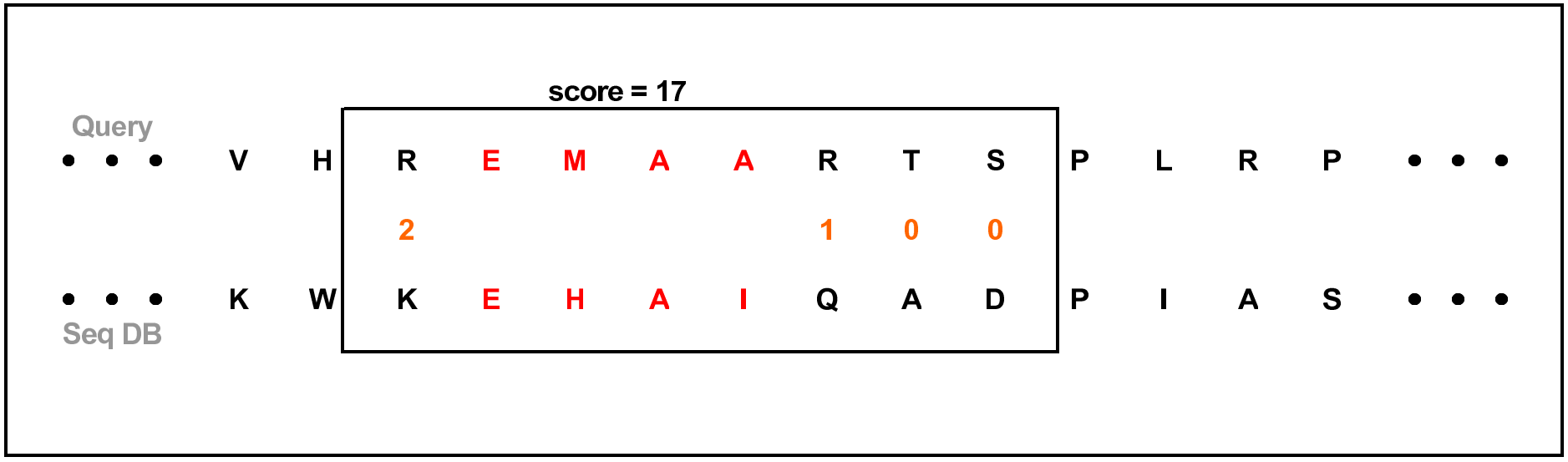
# BLAST - Obtenção dos *HSPs*



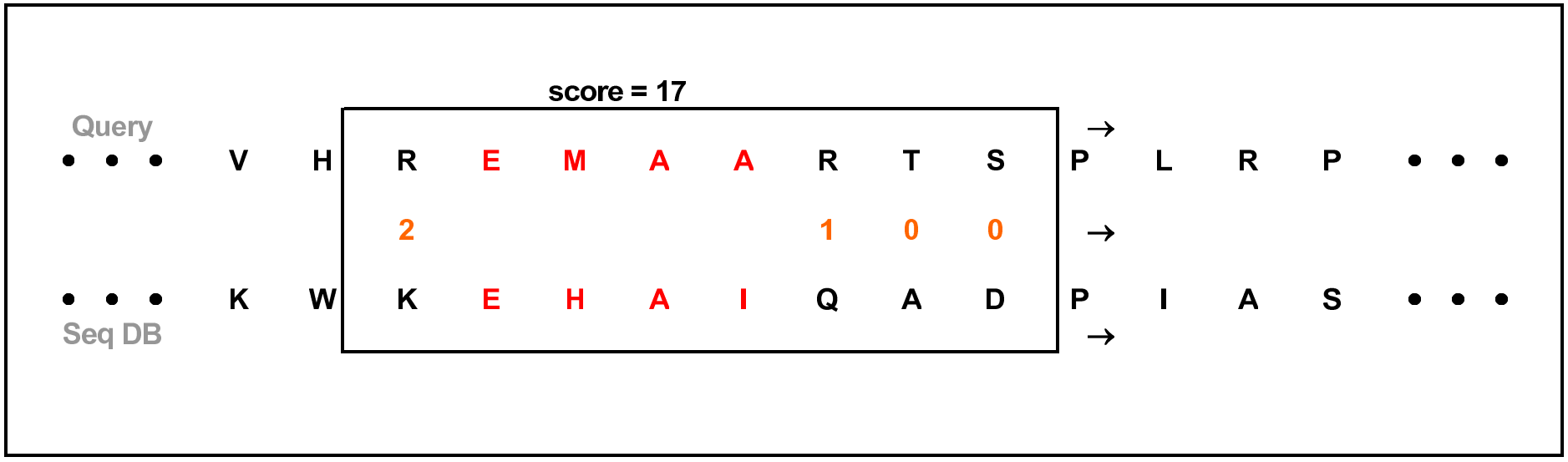
# BLAST - Obtenção dos *HSPs*



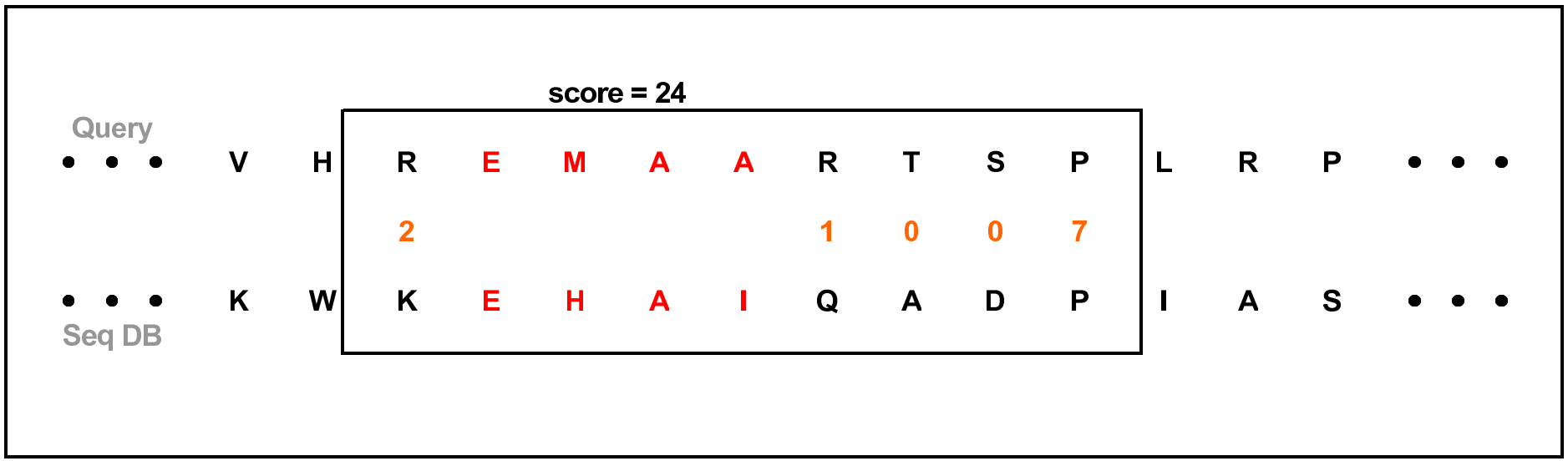
# BLAST - Obtenção dos *HSPs*



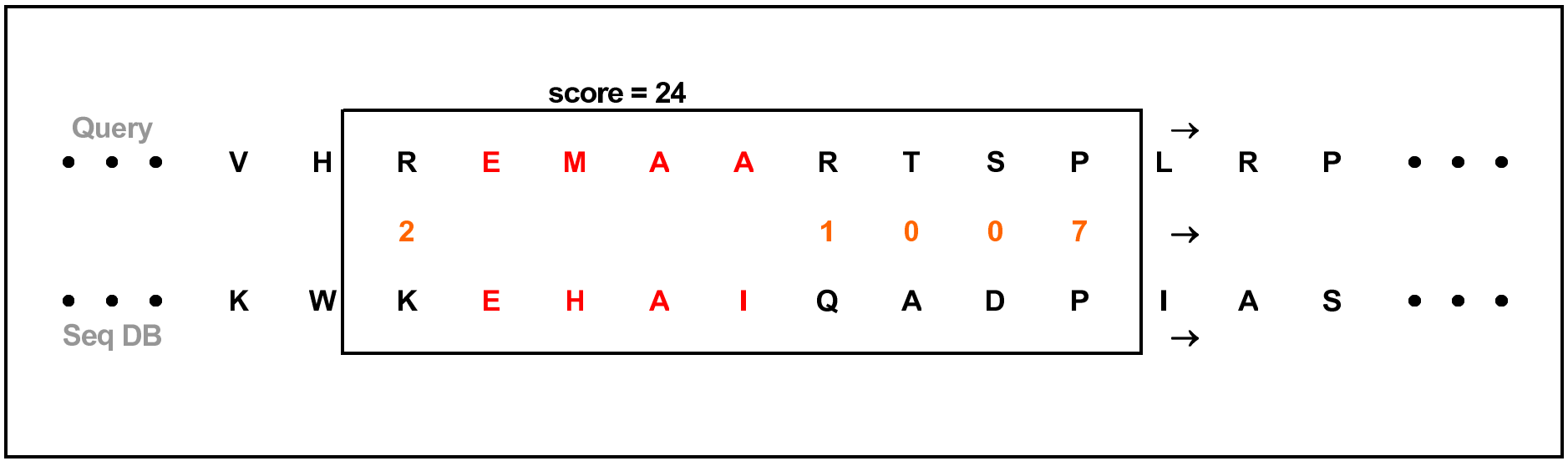
# BLAST - Obtenção dos *HSPs*



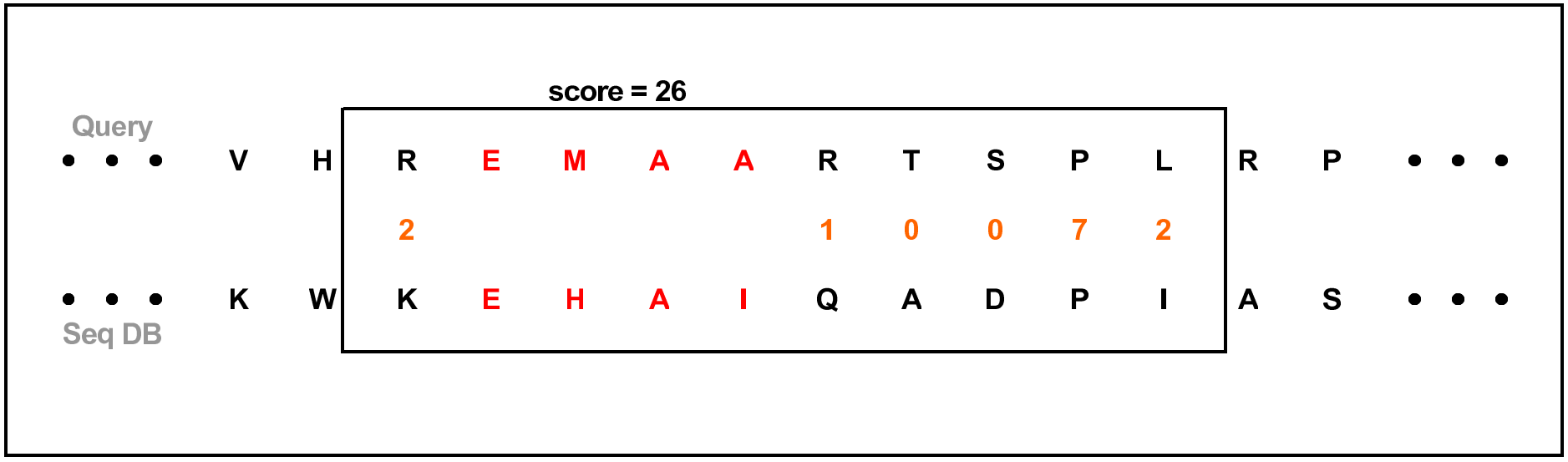
# BLAST - Obtenção dos *HSPs*



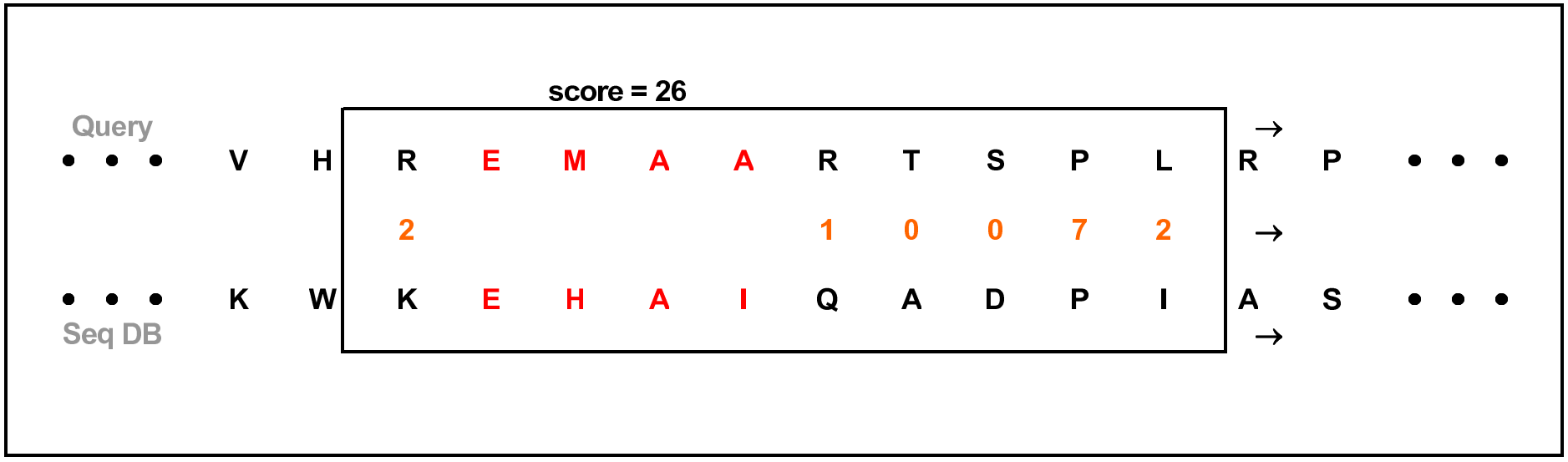
# BLAST - Obtenção dos *HSPs*



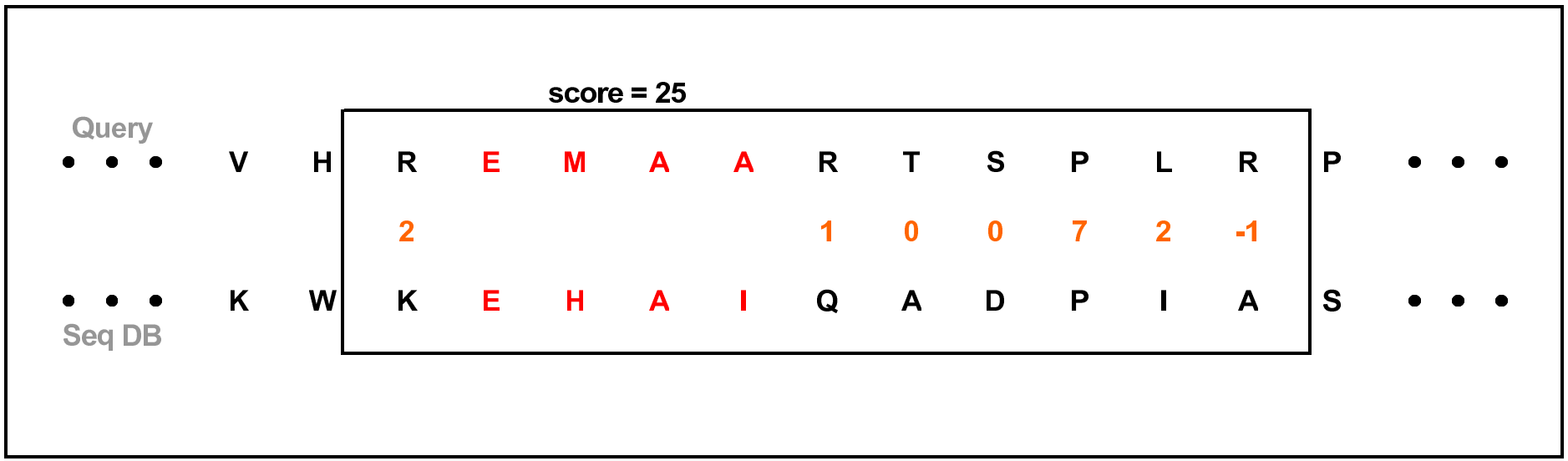
# BLAST - Obtenção dos *HSPs*



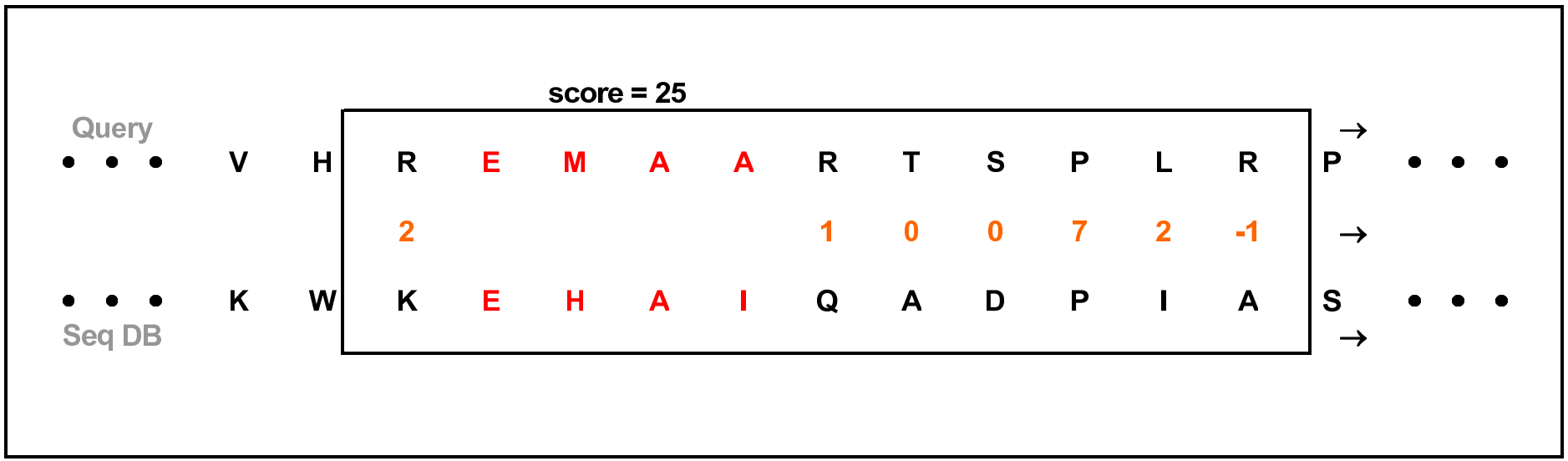
# BLAST - Obtenção dos *HSPs*



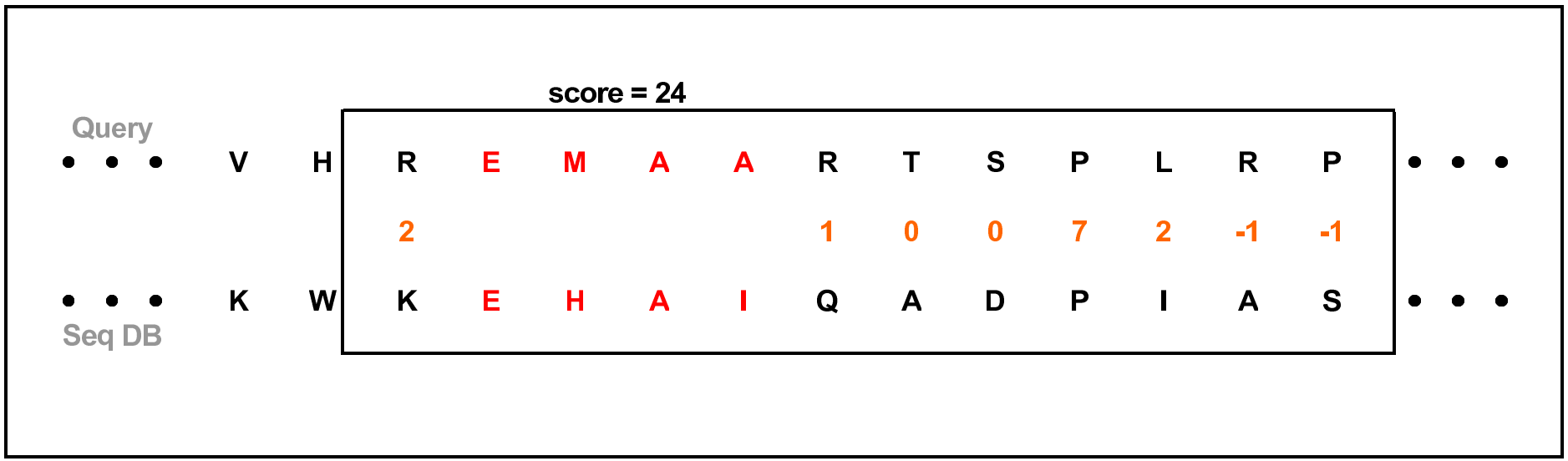
# BLAST - Obtenção dos *HSPs*



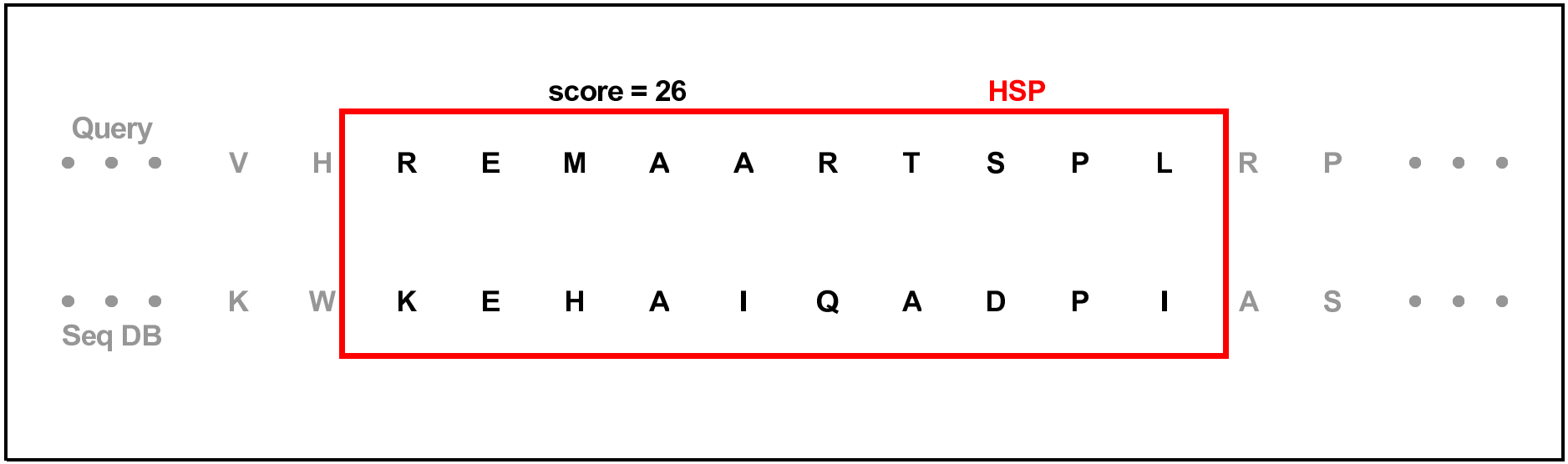
# BLAST - Obtenção dos *HSPs*



# BLAST - Obtenção dos *HSPs*



# BLAST - Obtenção dos *HSPs*



# BLAST 2.0

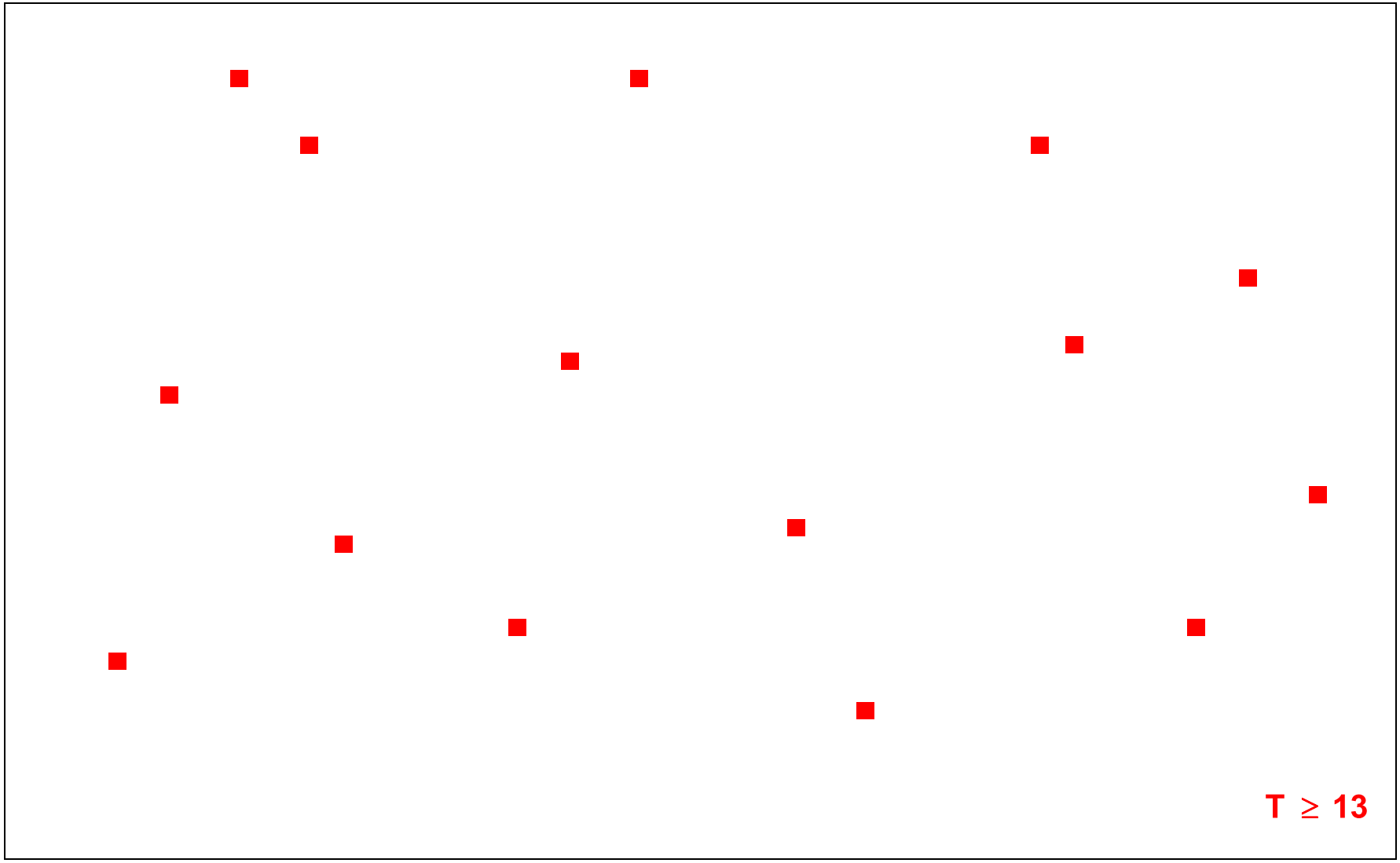
# BLAST 2.0

- Extensão apresentada por Stephen Altschul, Thomas Madden, Alejandro Schaeffer, Jinghui Zhang, Zheng Zhang, Webb Miller e David Lipman em 1997.
- Duas principais inovações:
  - The Two-Hit Method
  - Gapped BLAST
- O artigo que introduziu a segunda versão do BLAST é um dos artigos mais citados do mundo, com mais de 80000 citações.

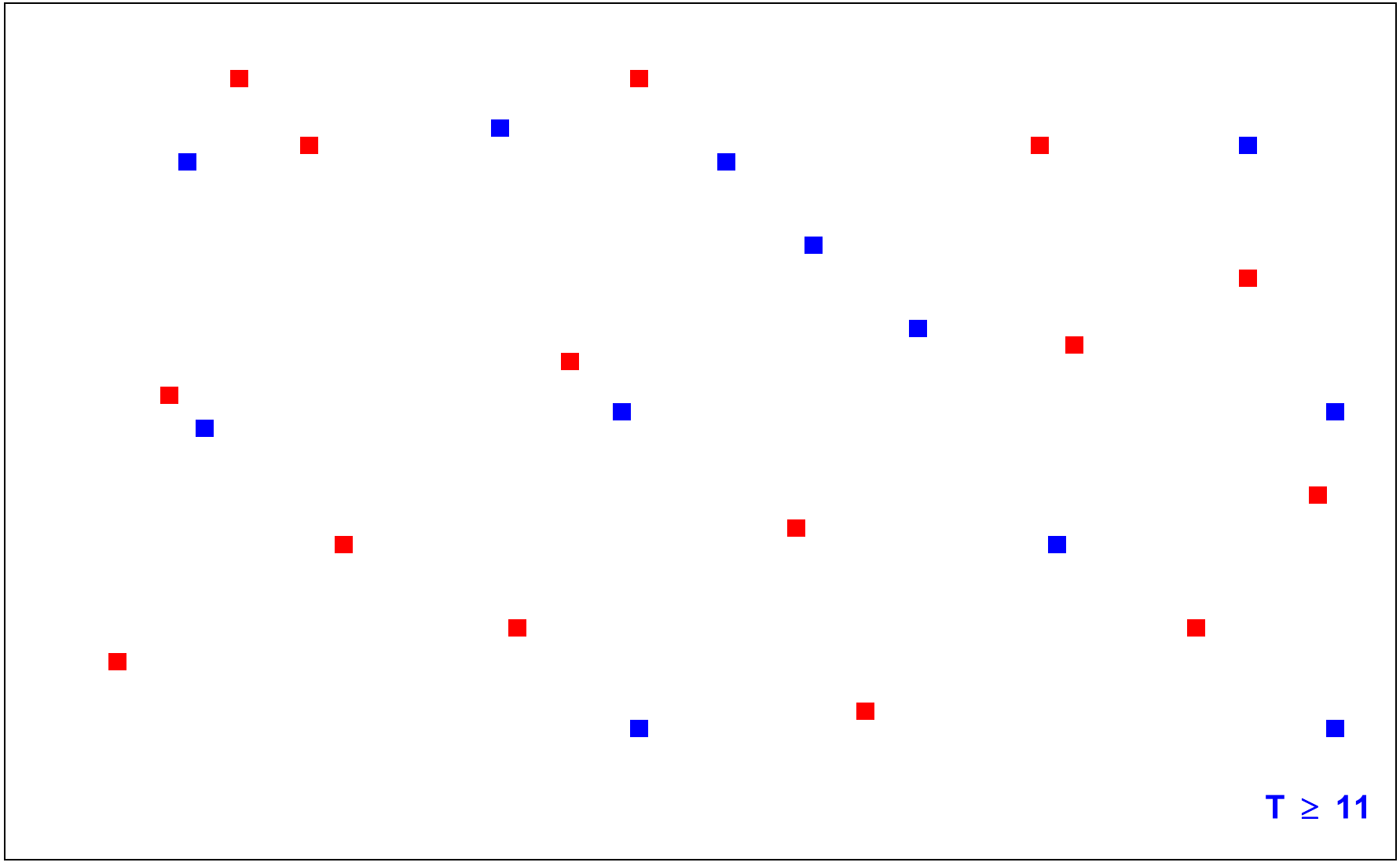
# BLAST 2.0 - The Two-Hit Method

- Objetivo: acelerar o tempo de processamento do algoritmo original.
- Reduz o número de extensões.
- Observação: *HSPs* são muito maiores que *w*.
- Um *HSP* frequentemente contém dois ou mais *hits*.
- Apenas procurar um *HSP* se existirem dois *hits* na mesma diagonal.
- Como implementar:
  - Se os *hits* se sobrepõe, ignorar.
  - Se os *hits* estiverem a uma distância menor do que um certo valor *A*, estender.
- O valor de *T* deve ser reduzido para se manter a mesma sensibilidade.

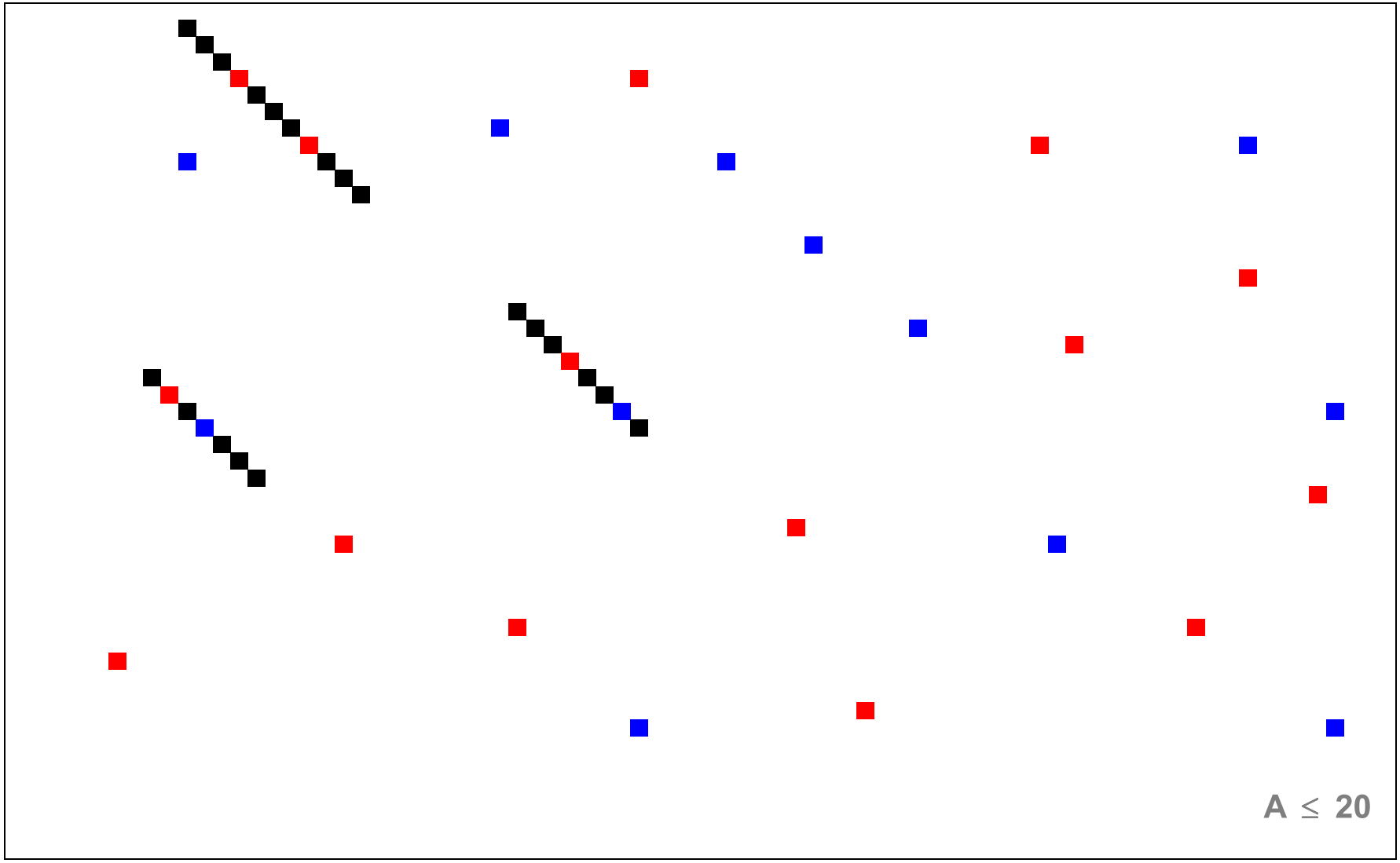
# BLAST 2.0 - The Two-Hit Method



# BLAST 2.0 - The Two-Hit Method



# BLAST 2.0 - The Two-Hit Method



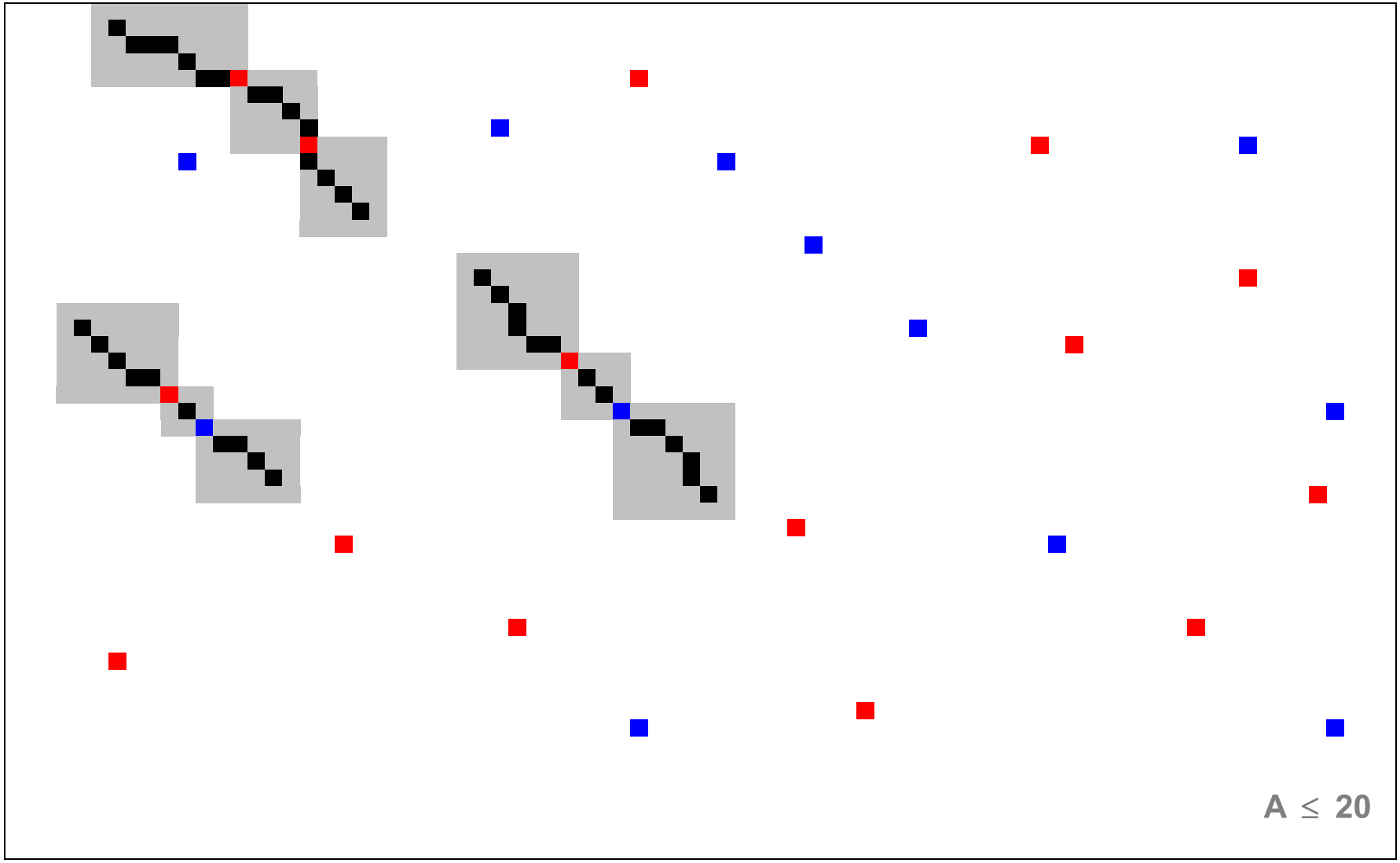
# BLAST 2.0 - The Two-Hit Method

- Métodos para extensão de hits (valores padrão para proteínas):
  - *One-Hit*: $w = 3$ e $T = 13$.
  - *Two-Hits*: $w = 3$ e $T = 11$.
- Comparação entre os métodos:
  - *Two-Hits* gera aproximadamente 3.2x mais *hits*.
  - *Two-Hits* faz aproximadamente 86% menos extensões.

# BLAST 2.0 - Gapped BLAST

- Extensões de *hits* não são mais limitados a diagonais da matriz de Programação Dinâmica, permitindo alinhamento com buracos.
- A extensão é interrompida quando o alinhamento cai abaixo de um limiar pré-estabelecido ($X_G$).
- Se a pontuação do *HSP* for maior que um parâmetro $S$, então o HSP é apresentado na lista de respostas.
- O BLAST 2.0 é cerca 3x mais rápido do que a versão original.

# BLAST 2.0 - Gapped BLAST



# Complexidade

- Seja $w$ o tamanho dos *seeds*, $\mathcal{A}$ o alfabeto ($|\mathcal{A}| = 4$ para DNA, $|\mathcal{A}| = 20$ para proteínas), $M$ o tamanho do banco de dados (número total de bases), $h$ o número de *hits* encontrados e $n$ o tamanho da *query*.
- Fase 1: Obter uma lista de *seeds*:
  - $O(nw|\mathcal{A}|^w)$
- Fase 2: Procurar *hits* de *seeds* nas sequências do banco de dados:
  - $O(M)$
- Fase 3: Estender os *hits* para obter os alinhamentos (*HSPs*):
  - $O(hn^2)$
- Total:
  - $O(nw|\mathcal{A}|^w + M + hn^2)$

# Bit Score e E-Value

- Com base no valor do alinhamento ($S$), BLAST indica para cada HSP retornado um *bit score* e um *E-value*.
- O *bit score* ($S'$) representa a pontuação normalizada do alinhamento, e é dada pela fórmula:

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

- O *E-value* representa o número esperado de HSPs com score maior ou igual a $S$, e é dado pela fórmula:

$$E = \frac{KMn}{e^{\lambda S}}$$

- As constantes $\lambda$ e $K$ são calculadas considerando a matriz de pontuação e a distribuição das bases no banco de dados.

# E-Value x P-Value

- Não confundir E-value com P-value:
  - E-Value: número esperado de alinhamentos aleatórios com pontuação maior ou igual a *S*.
  - P-Value: probabilidade de se obter aleatoriamente um alinhamento com uma pontuação maior ou igual a *S*.
- E-value pode ser um número maior que 1, enquanto o P-value é sempre um valor entre 0 e 1.
- Dado um E-value é possível obter o P-value (e vice-versa) através da seguinte fórmula:

$$P = 1 - e^{-E}$$

# Família BLAST

- O BLAST é composto por uma família de programas, todos acessíveis através do executável *blastall*:
  - blastn: [*query*: DNA] x [*database*: DNA].
  - blastp: [*query*: proteína] x [*database*: proteína].
  - blastx: [*query*: DNA] x [*database*: proteína] (nos 6 frames da *query*).
  - tblastx: [*query*: DNA] x [*database*: DNA] (nos 6 frames da *query* e de cada sequência do banco de dados).
  - tblastn: [*query*: proteína] x [*database*: DNA] (nos 6 frames de cada sequência do banco de dados).
  - megablast: ideal para comparar várias sequências contra um banco de sequências. Concatena todas as sequências de entrada em uma única, e depois faz um pós-processamento para obter os alinhamentos corretos.