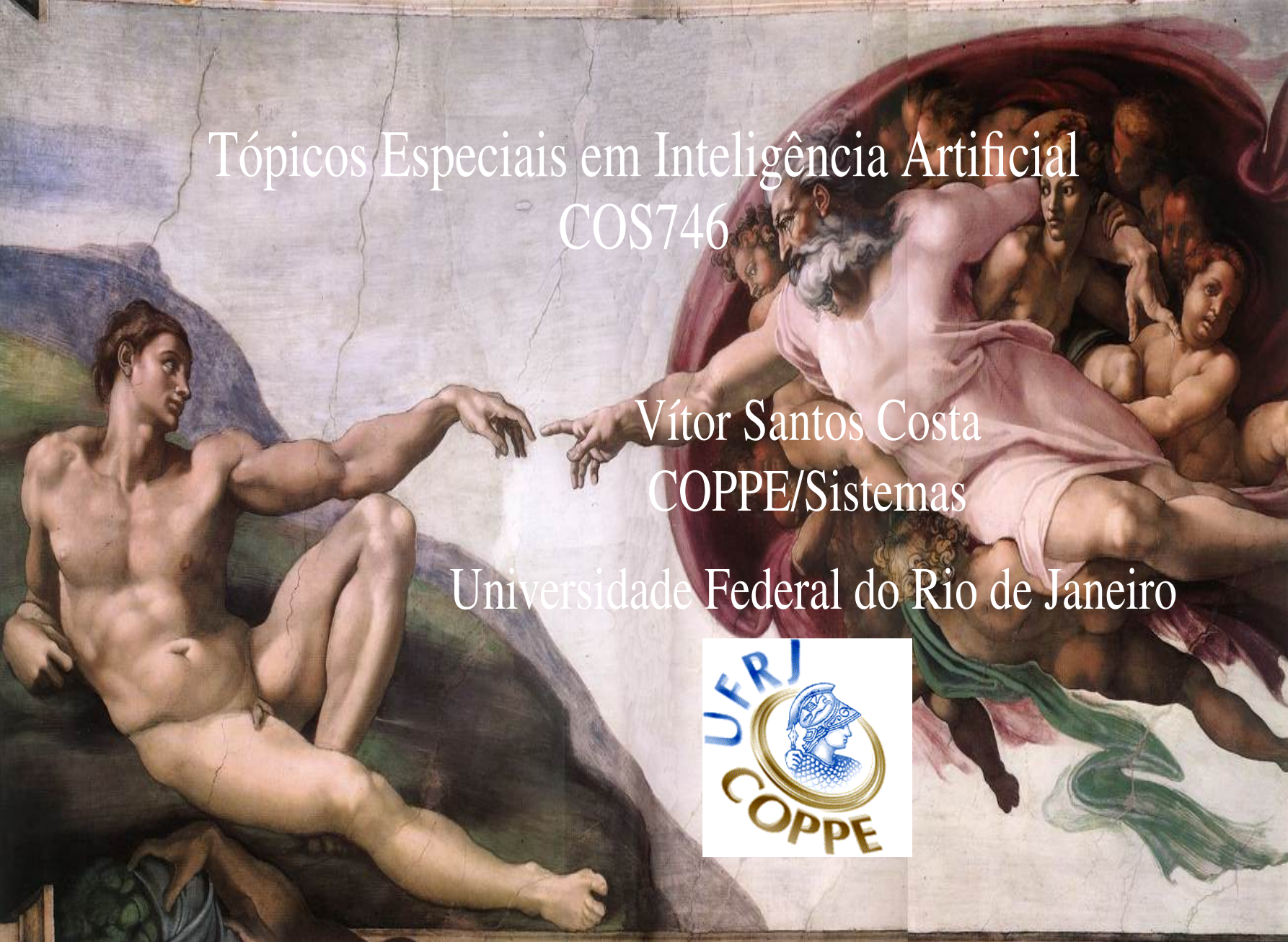


Tópicos Especiais em Inteligência Artificial
COS746

Vítor Santos Costa
COPPE/Sistemas
Universidade Federal do Rio de Janeiro

Agradecimento

- Copiado dos slides de Mark Craven/C. David Page para BMI/CS 576, UW-Madison

Funções de Penalização de buracos

- linear

$$w(k) = gk$$

- afim

$$w(k) = \begin{cases} h + gk, & k \geq 1 \\ 0, & k = 0 \end{cases}$$

- Côncava:

$$w(k+m+l) - w(k+m) \leq w(k+m) - w(k)$$

  - ⋆ Ex: $w(k) = h + g \times log(k)$

Programação Dinâmica para o caso afim

- Para conseguir em tempo $O(n^2)$ precisamos de 3 matrizes em vez de 1:
  - $\star$ $M(i,j)$ melhor valor se $x[i]$ estiver alinhado com $y[j]$
  - $\star$ $I_x(i,j)$ melhor valor se $x[i]$ estiver alinhado com um buraco
  - $\star$ $I_y(i,j)$ melhor valor se $y[i]$ estiver alinhado com um buraco

DP para o caso afim, global

- $M(i,j) = max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ I_x(i-1,j-1) + s(x_i, y_j) \\ I_y(i-1,j-1) + s(x_i, y_j) \end{cases}$
- $I_x(i,j) = max \begin{cases} M(i-1,j) + h + g \\ I_x(i-1,j) + g \end{cases}$
- $I_y(i,j) = max \begin{cases} M(i,j-1) + h + g \\ I_y(i,j-1) + g \end{cases}$
- Assumimos que é sempre melhor um match do que 2 buracos

DP para o caso afim global

- Inicialização
  - $M(0,0) = 0$
  - $I_x(i,0) = h + g \times i$
  - $I_y(0,j) = h + g \times j$
  - outras células no topo e coluna da esquerda $= -\infty$
- Voltar para trás:
  - começar no maior de $M(m,n), I_x(m,n), I_y(m,n)$
  - parar num de $M(0,0), I_x(0,0), I_y(0,0)$

DP para o caso afim local

- $$M(i,j) = max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ I_x(i-1,j-1) + s(x_i, y_j) \\ I_y(i-1,j-1) + s(x_i, y_j) \\ 0 \end{cases}$$
- $$I_x(i,j) = max \begin{cases} M(i-1,j) + h + g \\ I_x(i-1,j) + g \end{cases}$$
- $$I_y(i,j) = max \begin{cases} M(i,j-1) + h + g \\ I_y(i,j-1) + g \end{cases}$$

DP para o caso afim local

- Inicialização
  - $M(0,0) = 0$
  - $I_x(i,0) = 0$
  - $I_y(0,j) = 0$
  - outras células no topo e coluna da esquerda $= -\infty$
- Voltar para trás:
  - começar no maior de $M(i,j)$
  - parar num $M(i,j) = 0$

DP Para o Caso Geral

Alinhamento Global:

- $$F(i,j) = max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(k,j) + \gamma(i-k) \\ F(i,k) + \gamma(j-k) \end{cases}$$
  - Considerar todos os elementos anteriores na linha!
  - Considerar todos os elementos anteriores na coluna!

Complexidade Computacional

Dependendo da penalização de buracos:

- linear:

$$O(n^2)$$

- afim:

$$O(n^2)$$

- geral:

$$O(n^3)$$

Motivação para Uso de Heurísticas

- $O(mn)$ demasiado lento para grandes bancos de dados com muitas interrogações
- métodos heurísticos permitem aproximação rápida à programação dinâmica:
  - ⋆ FASTA, de Pearson & Lipman, 1988
  - ⋆ BLAST, de Altschul et al., 1990

Motivação para Alinhamento Por Heurísticas

- Imaginem procurar SWISS-PROT contra uma sequência de interrogação:
  - ⋆ imaginem que a nossa pergunta tem 362 amino-ácidos
  - ⋆ SWISS-PROT versão 38 contém 29.085.265 amino-ácidos
  - ⋆ procurar alinhamentos locais através da programação dinâmica obrigaria a $O(10^{10})$ operações em matrizes
- muitos servidores têm que resolver milhares de tais perguntas por dia
  - ⋆ NCBI > 100.000

BLAST

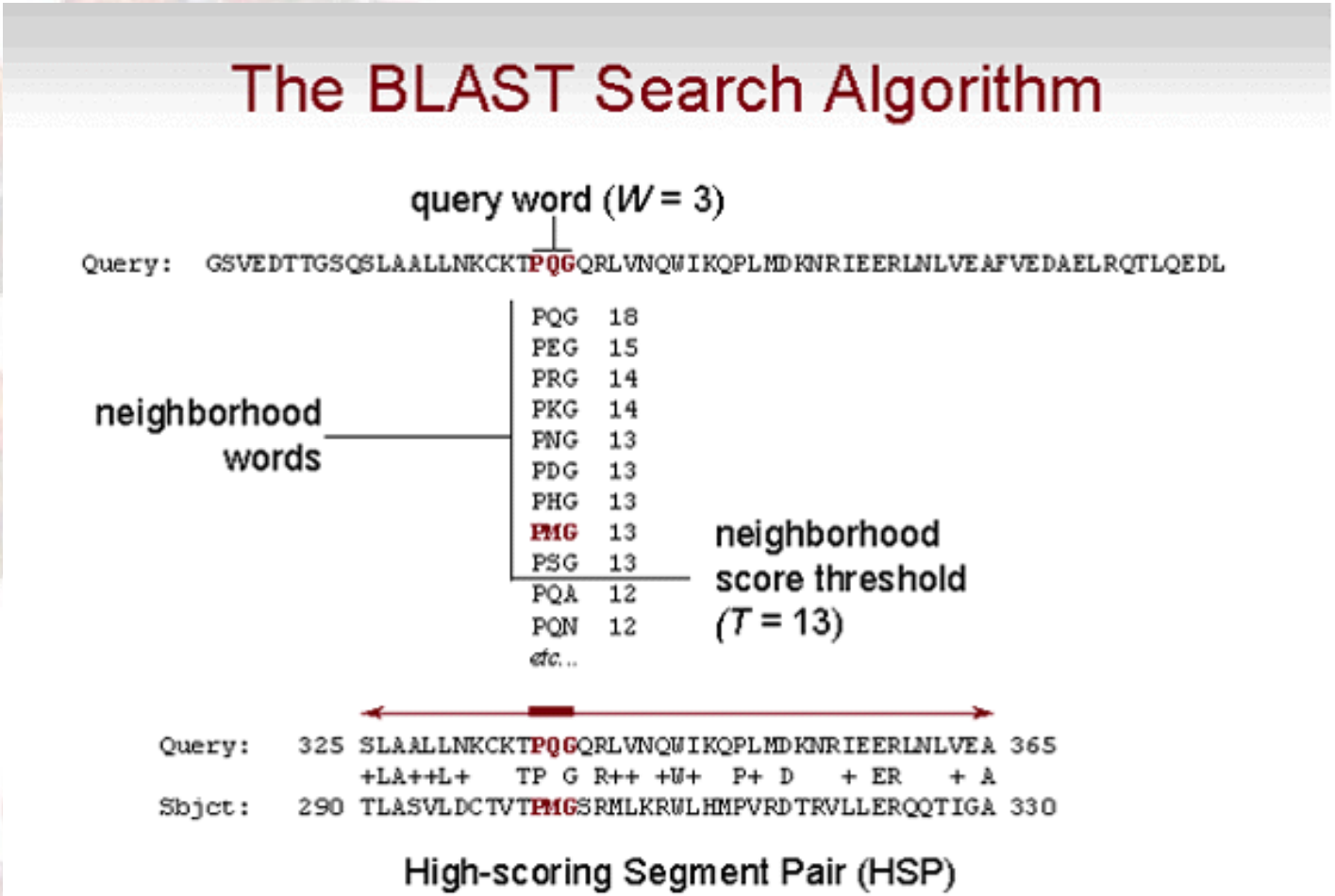
- **B**asic **L**ocal **A**lignment **S**earch **T**ool
- BLAST usa heurísticas para encontrar *pares com pontuação alta* (HSPs):
  - ⋆ Segmentos do mesmo tamanho de 2 sequências com pontuação de alinhamento estatisticamente significantes
  - ⋆ ie, alinhamentos locais sem buracos
- Escolha entre precisão e velocidade

$$precisao = \frac{\#Emparelhamentos\ Significantes}{\#EmparelhamentosnaDB}$$

Ideia do BLAST

- Dada uma sequência de interrogação q tamanho de palavra w, um limite de pontuação T, e um limite de segmento S:
  - ⋆ compilar uma lista de palavras que têm resultado $\geq T$ quando comparadas com palavras de q
  - ⋆ percorre a BD por alinhamentos com palavras na lista
  - ⋆ extender todos alinhamentos para procurar os pares de sequência com pontuação mais alta.
- resultado: pares de segmentos com resultado $\geq S$

Intuição


The BLAST Search Algorithm
query word (W = 3)
Query: GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
neighborhood words
PQG 18
PEG 15
PRG 14
PKG 14
PNG 13
PDG 13
PHG 13
PMG 13
PSG 13
PQA 12
PQN 12
etc...
neighborhood score threshold (T = 13)
Query: 325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
+LA++L+ TP G R++ +W+ P+ D + ER + A
Sbjct: 290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
High-scoring Segment Pair (HSP)

Determinação de Palavras da Interrogação

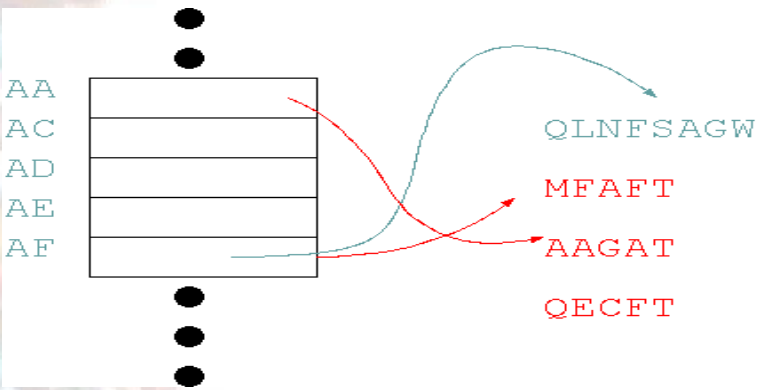
- Dada:
  - ⋆ sequência de interrogação: QLNFSAGW
  - ⋆ tamanho de palavra $w = 2$ (para proteína usualmente $w = 3$)
  - ⋆ limite para pontuação de palavra; $T = 8$
- Passo 1: determinar todas as palavras de tamanho w na sequência de interrogação:
  - ⋆ QL LN NF FS SA AG GW

Palavras Similares Query

- Passo 2
  - ⋆ Procurar todas as palavras com resultado acima de limiar T
  - ⋆ Usando $T = 9$
    - ∗ `QL` $\Rightarrow$ `QL (10)`
    - ∗ `LN` $\Rightarrow$ `LN (11), LS (9)`
    - ∗ `FS` $\Rightarrow$ `FS (12)`

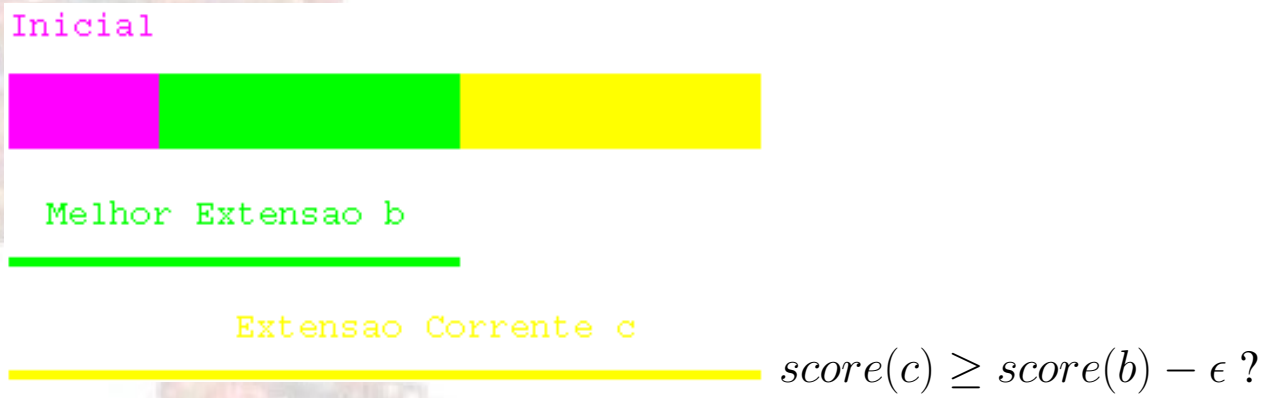
Procurando na BD

- Procurar na BD por todas as instâncias das palavras na sequência de interrogação:
- método:
  - ⋆ indexar sequências na BD com *tabela de palavras*
  - ⋆ procurar palavras da interrogação na tabela



Ampliar Sucessos

- Ampliar sucessos em ambas as direcções (sem permitir buracos)
- terminar a ampliação numa direcção quando a pontuação cair abaixo de certa distância abaixo pontuação óptima para pequenas extensões



- resultado: pares de segmentos com resultado pelo menos S

Pontuação

- Se na BD $\mathcal{D}$ a sequência X tem o pontuação $S(\mathcal{D}, \mathcal{X}) = \int$, então:
  - *p-value:* $P(S(D, Y) \leq s$, onde Y é uma sequência aleatória
  - Quanto menor, melhor, eg:
  - 0.1: 1 em 10 têm pontuação $\geq$
  - 10^{-6}: 1 em 1000000
- Como fazer isso para BLAST?
- Simplificação:
  - Se pontuação 1 para acerto
  - $-\inf$
  - pontuação óptima: maior alinhamento

Pontuação: Explicação

- Se matriz de alinhamento tem tamanho $n \times m$,
- Alinhamento pode começar $\approx n \times m$ posições
- Além disso:
  - $\star$ p probabilidade de 2 letras à sorte serem iguais.
- Probabilidade de alinhamento de tamanho t

$$p^t(1-p)$$

- Média será:

$$nmp^t(1-p)$$

- É conhecido que binomial pode ser aproximado por Poisson se tp e p pequenos

$$P(seq_t) = 1 - P(\neg seq_t) = 1 - \frac{\lambda^0 e^{-\lambda}}{0!} = 1 - e^{-nmp^t(1-p)}$$

- BLAST:

$$p - value \approx 1 - e^{-nm\gamma\xi^t}$$

Pontuação: Explicação

- Valor-E:

$$e - value = \gamma nm\xi^t = Knme^{-\lambda S}$$

  - ⋆ Mais intuitivo para matches fracos
  - ⋆ E-value de 5, P-value de 0.993
  - ⋆ E-value de 10, P-value de 0.99995
  - ⋆ Convergem para matches fortes
- Unidades?
- Bit-Score não depende dos parâmetros K e λ

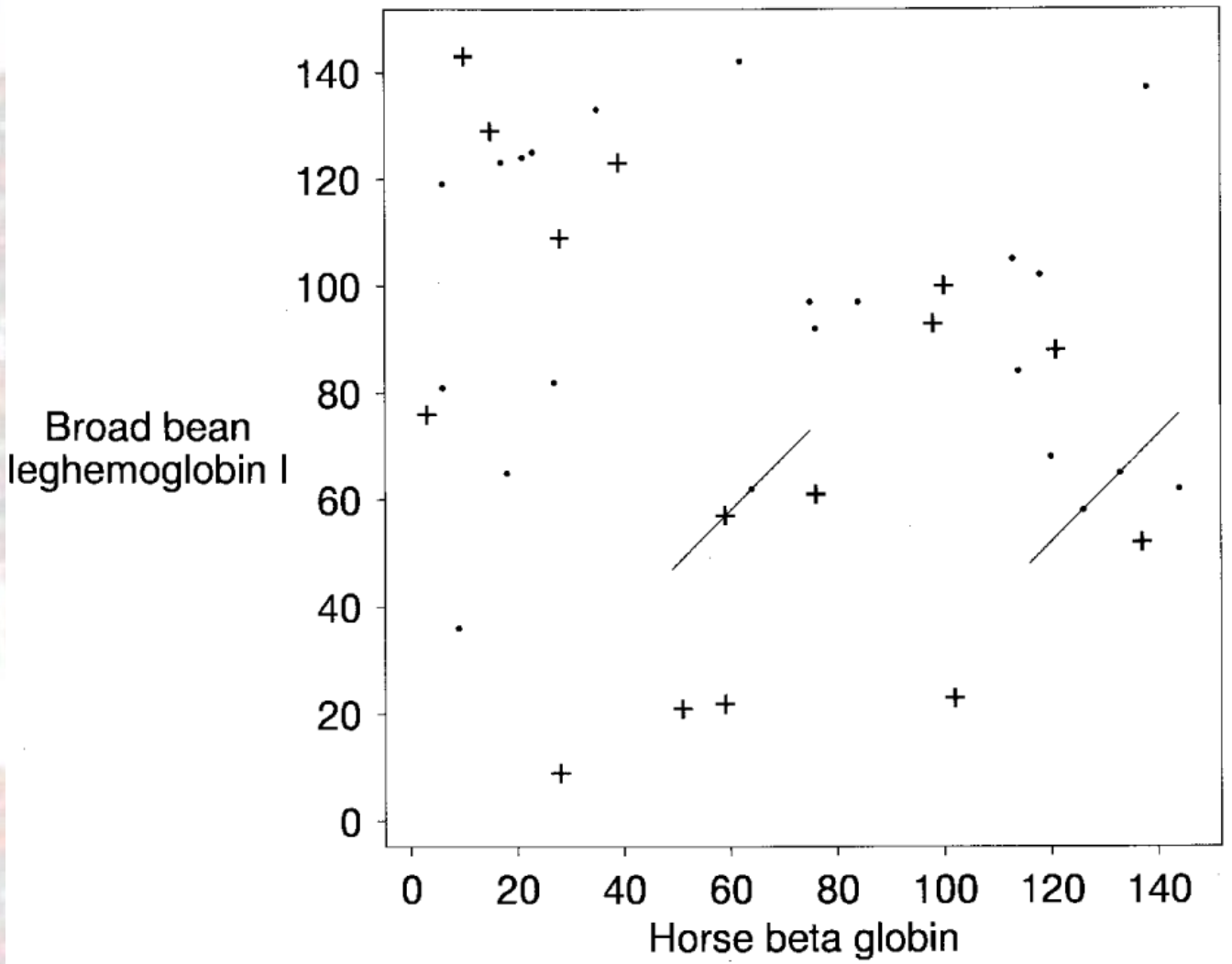
$$S' = \frac{\lambda S - lnK}{ln2}$$

- Para entender significância precisa apenas de mn

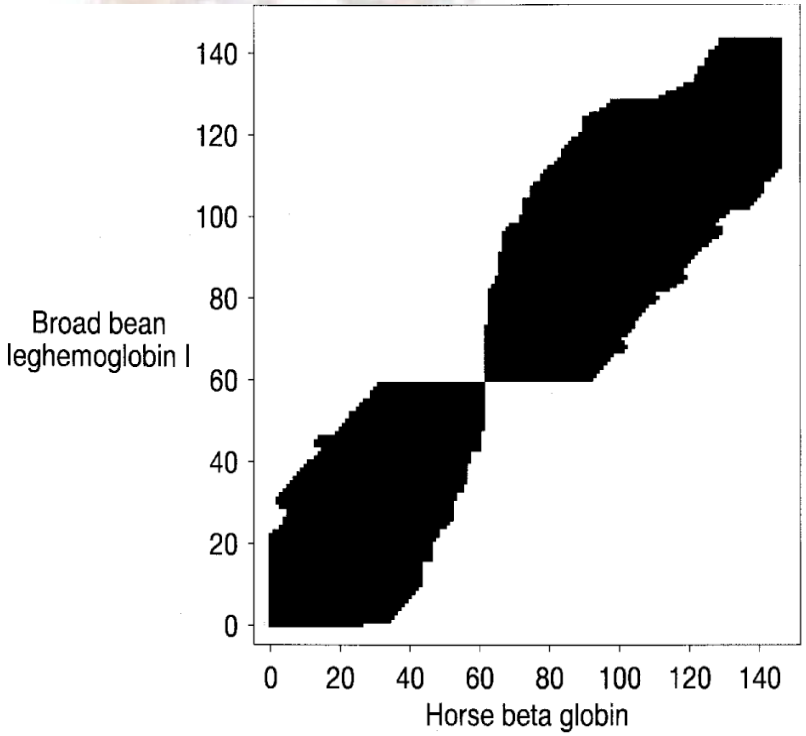
Extensões de BLAST

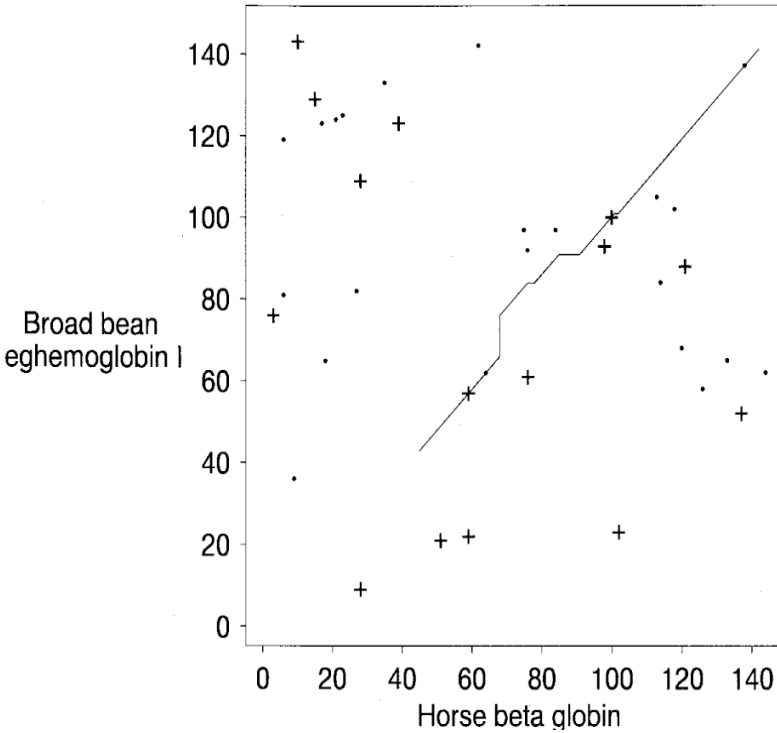
- O método *two-hit*: ampliar apenas quando há dois acertos perto e na mesma diagonal
  - ⋆ permite abaixar T
- BLAST com buracos:
  - ⋆ usar DP a partir de alinhamento com pontuação melhor
- PSI-BLAST: generalizar iterativamente a questão (fazê-la parecer mais como acertos) e voltar a procurar
- Todas tentam aumentar precisão enquanto limitam o tempo de execução

Método Two-Hits



Gapped Blast



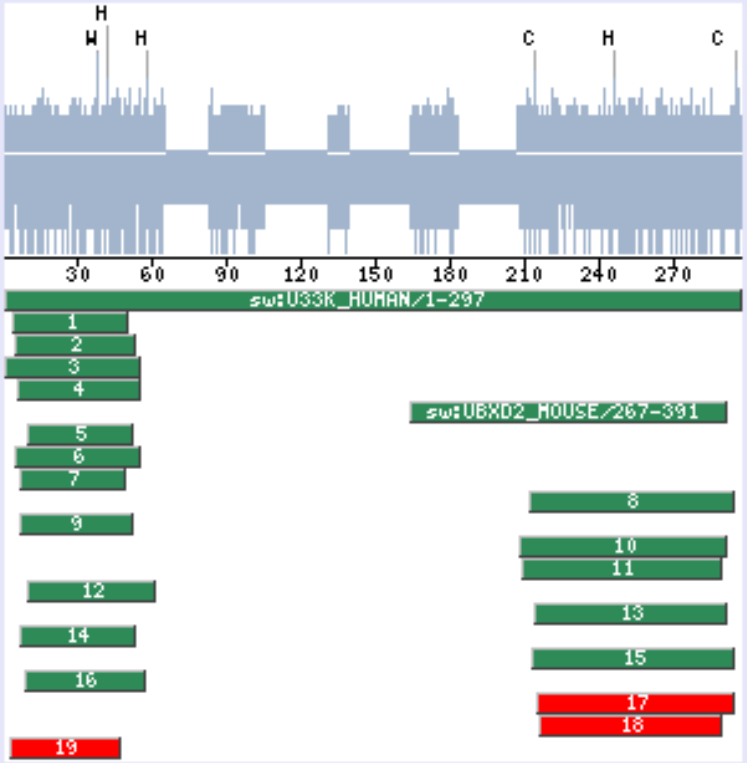



```
Leghemoglobin  43 FSFLKDSAGVVDSPKLGAHAEKVFGMVRDSAVQLRATGEVV--LDGKDGS------  90
                  F  L +   V+ +PK+ AH +KV          L + GE V  LD   G+
Beta globin    45 FGDLSNPGAVMGNPKVKAHGKKV----------LHSFGEGVHHLDNLKGTFAALSE  90


Leghemoglobin  91 IHIQKGVLDP-HFVVVKEALLKTIKEASGDKWSEELSAAWEVAYDGLATAI 140
                  +H  K  +DP +F ++   L+  +    G  ++ EL A+++    G+A A+
Beta globin    91 LHCDKLHVDPENFRLLGNVLVVVLARHFGKDFTPELQASYQKVVAGVANAL 141
```

Gapped BLAST

Matches map



Legends: **1**, sw:UAS3_DROME/22-68; **2**, sw:UBP14_SCHPO/580-628; **3**, sw:UBP5_MOUSE/654-708; **4**, sw:UBP14_YEAST/612-661; **5**, sw:UBP14_ARATH/622-664; **6**, sw:UBP13_HUMAN/656-706; **7**, sw:UBPA_DICDI/634-676; **8**, sw:YOJ8_CAEEL/281-359; **9**, sw:UAS3_HUMAN/25-70; **10**, sw:UBX7_YEAST/211-290; **11**, sw:UBXD1_HUMAN/332-406; **12**, sw:UPL2_ARATH/1280-1332; **13**, sw:UBX6_YEAST/191-264; **14**, sw:UBP14_SCHPO/645-690; **15**, sw:UBXD7_HUMAN/412-488; **16**, sw:UBP5_MOUSE/730-777; **17**, sw:FAF1_MOUSE/574-648; **18**, sw:UBAX1_ARATH/350-418; **19**, sw:UBP13_HUMAN/731-775.

Papers sobre extensões do BLAST

- Altschul, S.F., Madden, T.L., et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25 (17), 3389-3402. http://www.ncbi.nlm.nih.gov/CBBresearch/Altschul/
- Zhang, Z., Schwartz, S., Wagner, L. Miller, W. ”A greedy algorithm for aligning DNA sequences.”J. Computational Biology (2000) 7:203-214. http://bio.cse.psu.edu/‘¡
- Schaffer, A.A., Aravind, L., Madden, T.L., Shavirin, S., Spouge, J.L., Wolf, Y.I., Koonin, E.V., Altschul, S.F. (2001) Nucleic Acids Research, July 15;29(14):2994-3005 Improving the accuracy of

  PSI-BLAST protein database searches with composition-based statistics and other refinements. http://www.ncbi.nlm.nih.gov/CBBresearch/Altschul/

Comentários sobre BLAST

- é uma heurística: pode não encontrar alguns bons resultados
- rápido: empiricamente é de 10 a 50 vezes mais rápido do que Smith-Waterman
- Tem grande impacto:
  - ⋆ o servidor do NCBI recebe mais de 100.000 interrogações por dia
  - ⋆ é o programa mais usado em bioinformática

Parâmetros Default do BLAST

- M: ganho por match de nucleotídeos (5)
- N: perca por mismatch de DNA (-4)
- Buracos: usa afim com 11/1
- Matriz BLOSUM62
- Tamanho de palavra:
  - ⋆ 11 para nucleotídeos
  - ⋆ 3 para AAs

Conclusões

- apresentamos alinhamentos: *locais* e *globais*
- o algoritmo exacto com DP depende de ser local/global e da função da penalização de buracos
- ao permitir buracos permitimos um número exponencial de alinhamentos
- com programação dinâmica a complexidade é $O(mn)$
- algoritmos funcionam tanto para proteínas como DNA
- heurísticas como BLAST são mais rápidas mas não tão precisas.