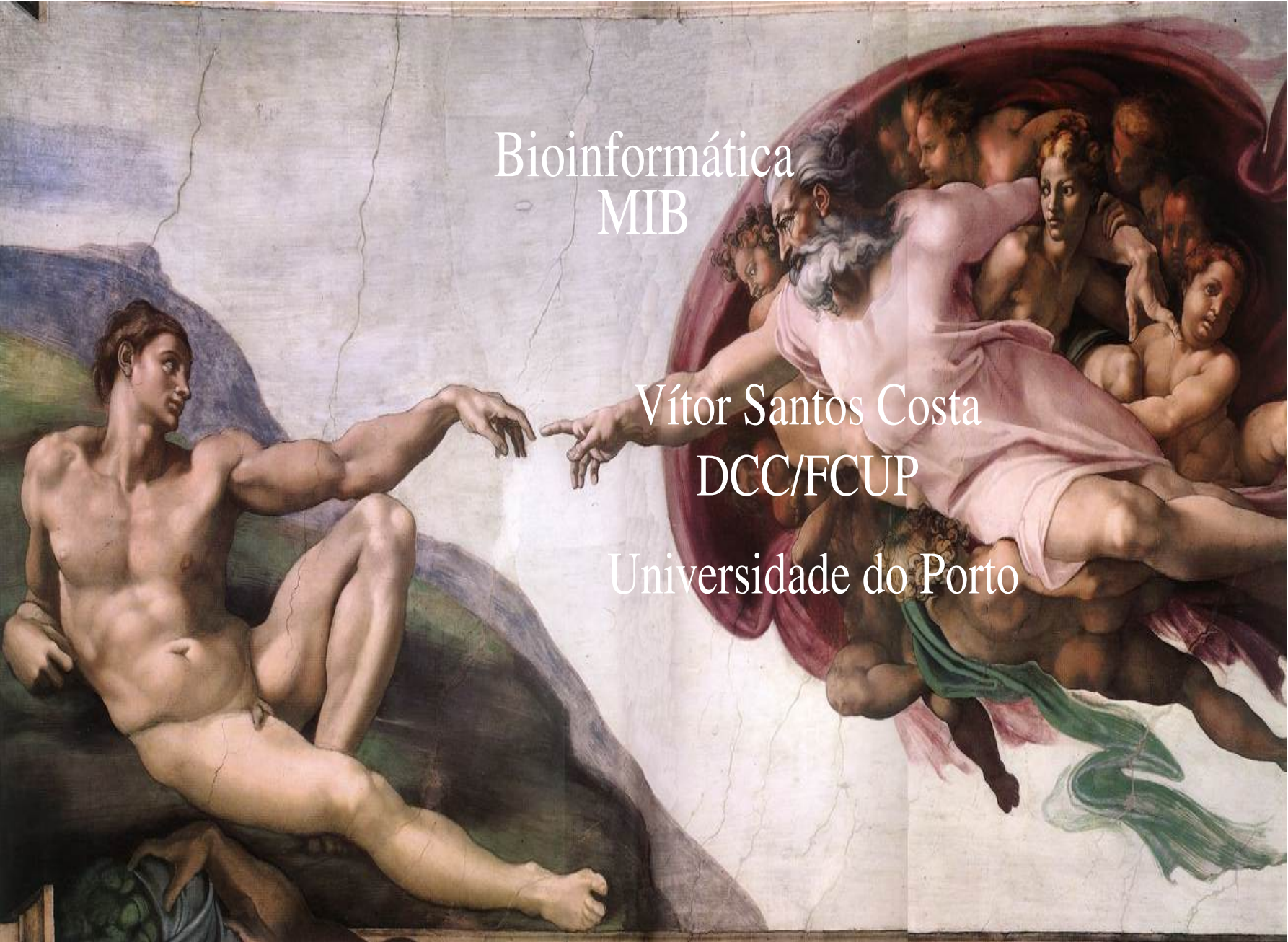


Bioinformática
MIB

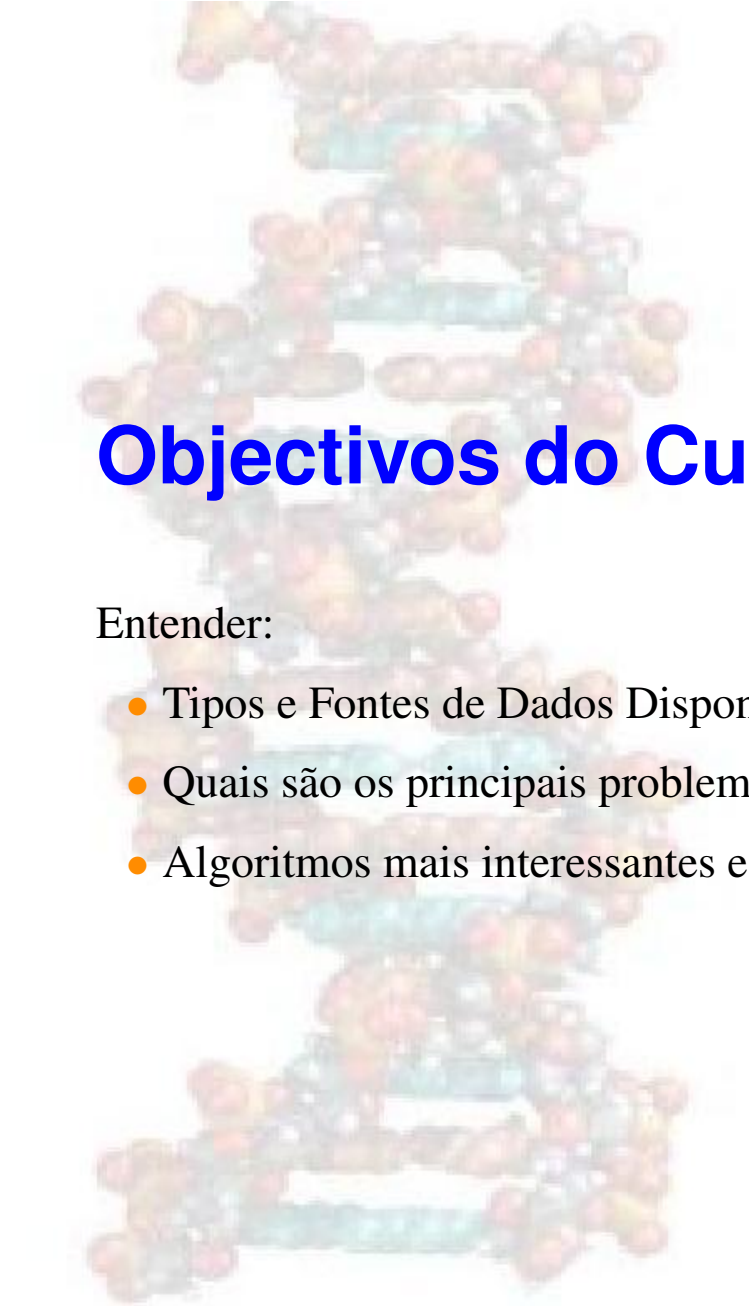
Vítor Santos Costa
DCC/FCUP
Universidade do Porto

BioInformática

Processamento/armazenamento/apresentação/pesquisa de dados biológicos:

1. *sequências*;
2. *estruturas*;
3. *funções*;
4. *níveis de actividade*;
5. *redes de interacção*;

de/entre biomoléculas.
Também conhecida como *Biologia Computacional* ou *Biologia Molecular Computacional*



Objectivos do Curso

Entender:

- Tipos e Fontes de Dados Disponíveis em Biologia Molecular;
- Quais são os principais problemas computacionais;
- Algoritmos mais interessantes e relevantes.

Fundamentos

1. Algoritmos e Estruturas de Dados
2. Estatística.
3. Biologia Molecular.

Bibliografia

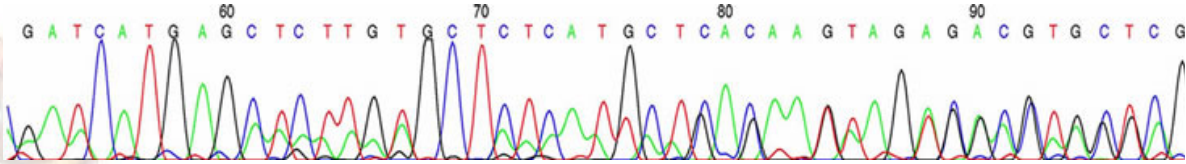
1. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. Cambridge University Press, 1998.
2. Computational Genome Analysis An Introduction, Richard Deonier, S Tavaré, and Michael S. Waterman, Springer Verlag, 2005.
3. Bioinformatics and Functional Genomics, Jonathan Pevsner, 2nd edition, Wiley-Blackwell, 2009.
4. Papers, etc.

Exemplo de Problemas Computacionais em Biologia Molecular

- Alinhamento de pares de sequências;
- Procura em bancos de dados de sequências;
- Alinhamento múltiplo de sequências;
- Modelagem e reconhecimento de genes;
- Modelagem e reconhecimento de “sinais”;
- Estrutura e funções de proteínas;
- Análise da Expressão de genes;
- Construção de árvores filogenéticas.

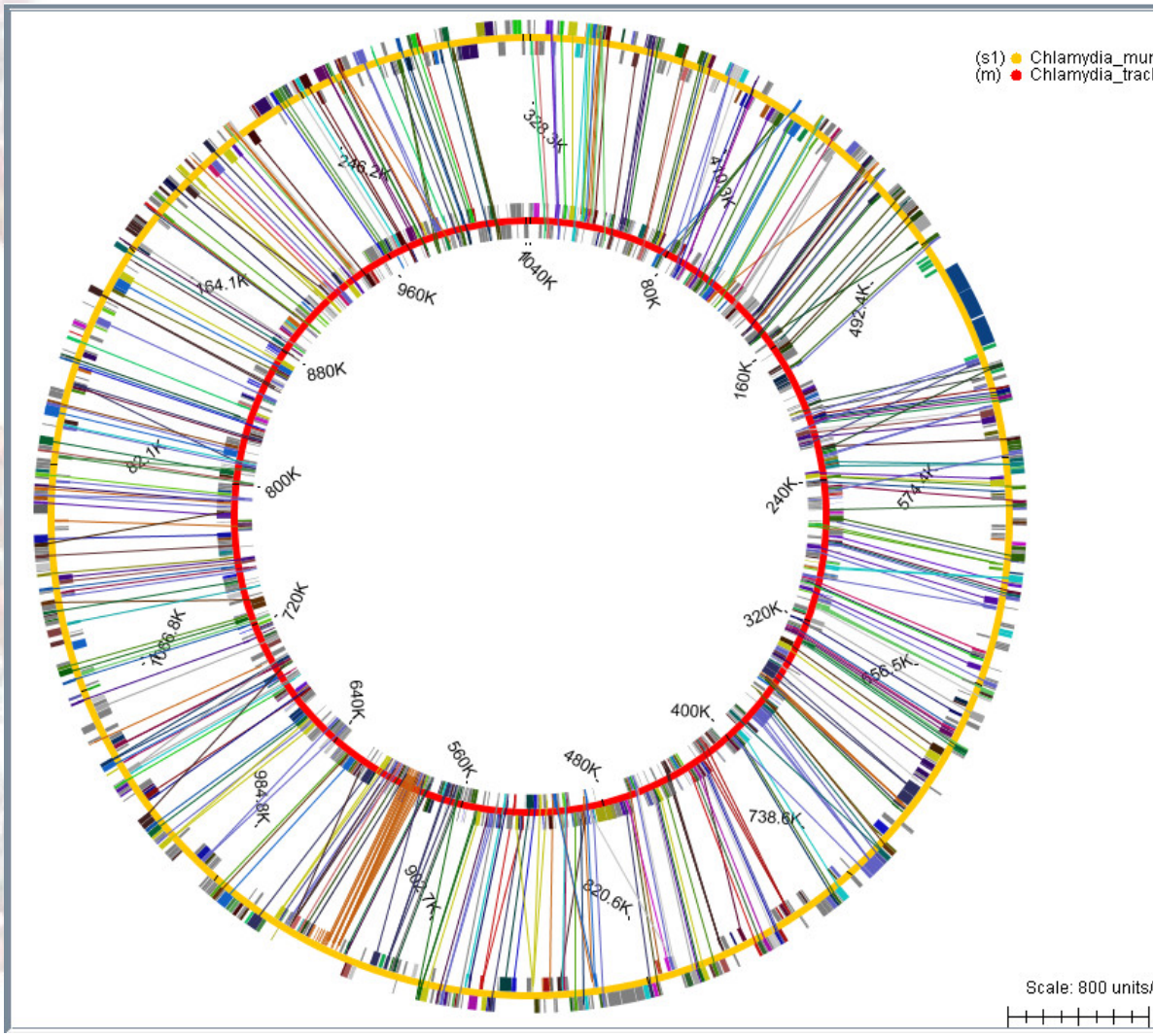
Montar Genoma

- Pipelines
- Phred/Phrap
- ABySS
- Estatística



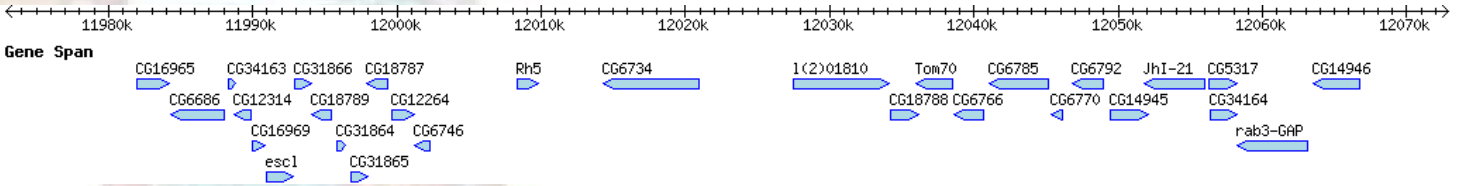
Comparar duas sequências de Genes

- Sequências
- Programação vDinâmica



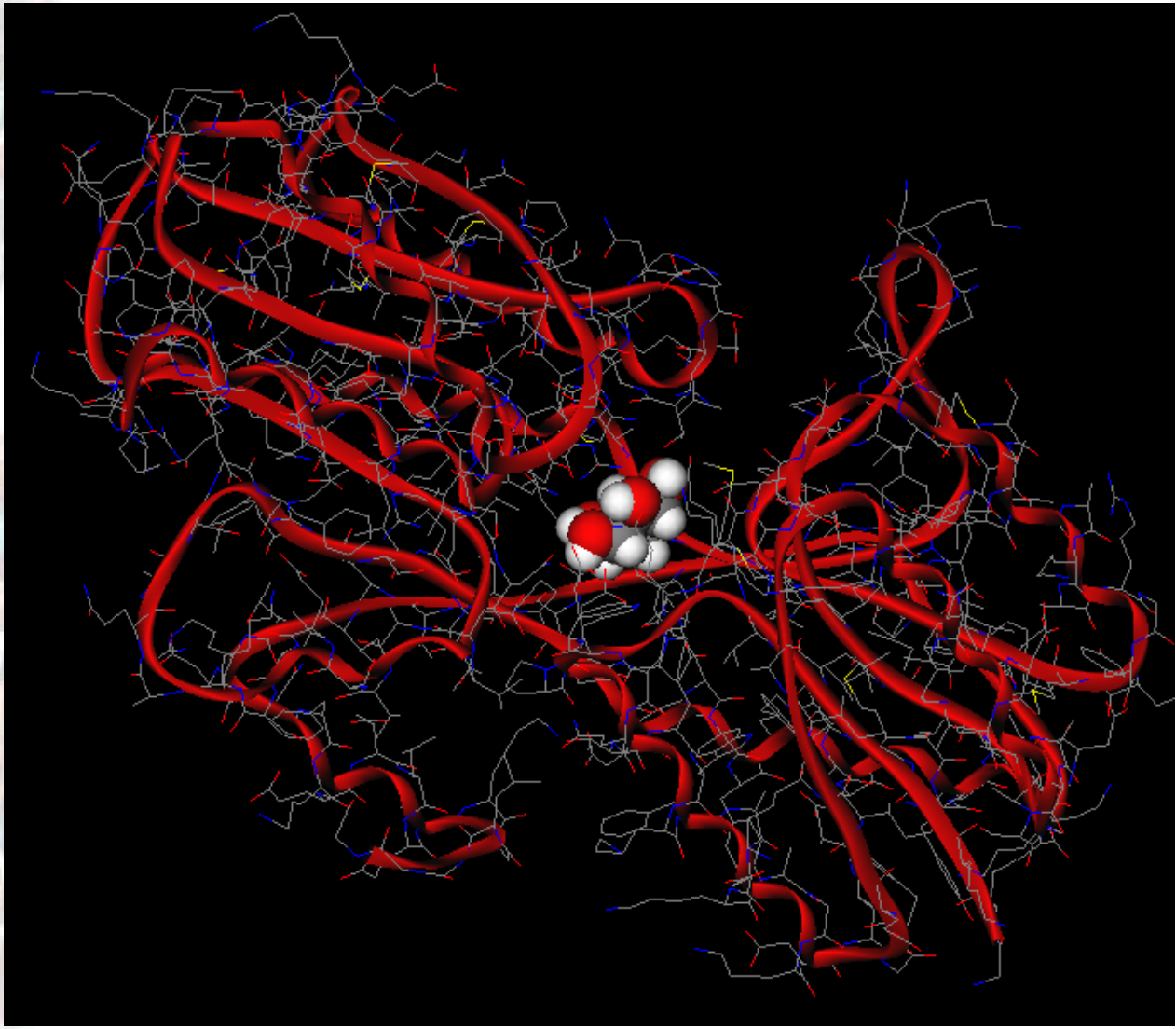
Encontrar Genes no Genoma

- Markov Models
- Hidden Markov Models



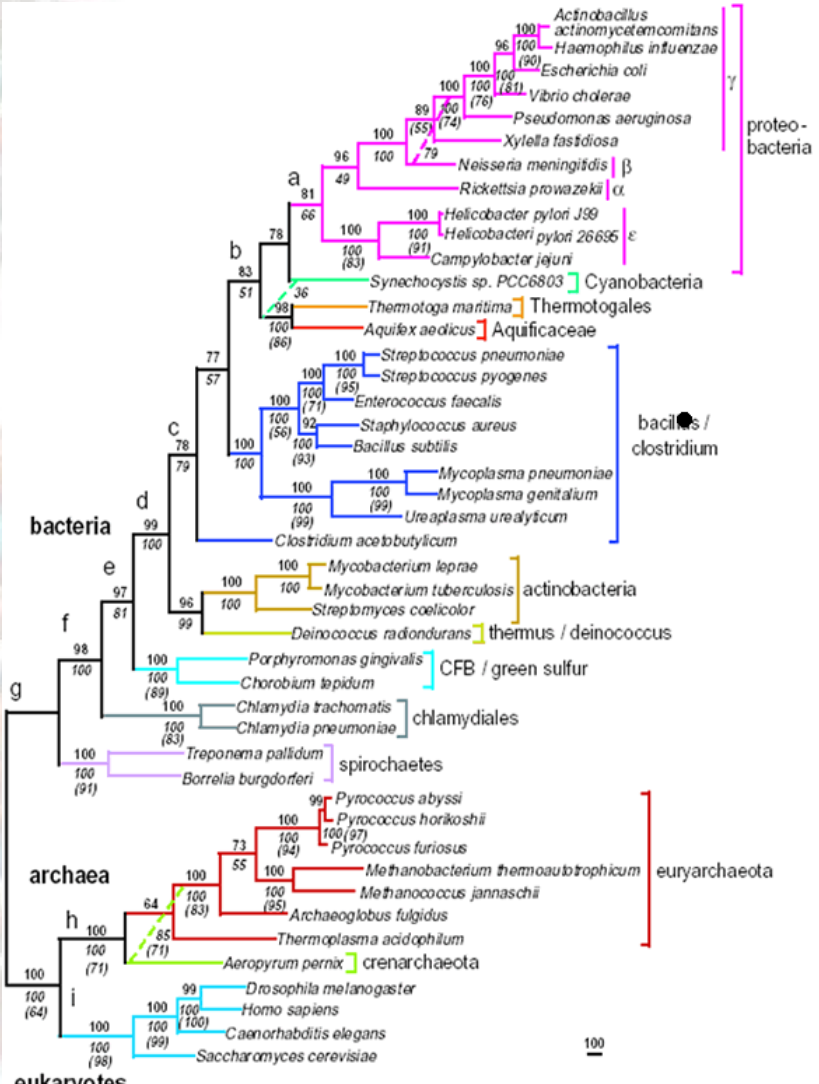
Estrutura de Proteínas

- Programação Dinâmica
- Branch & Bound
- HMMs



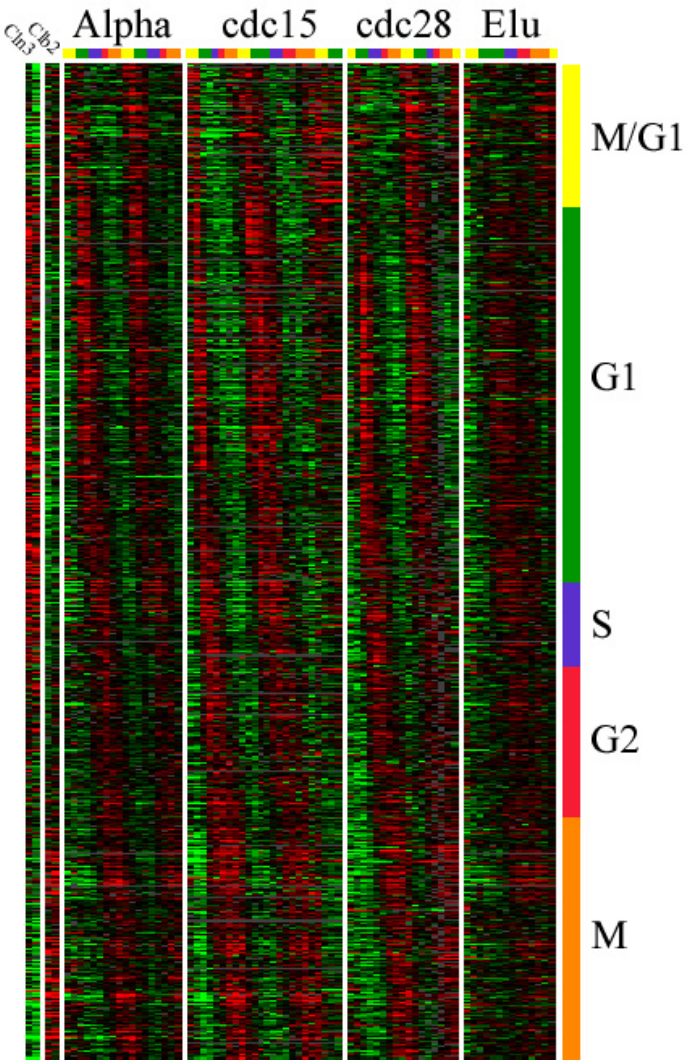
Arvores Filogenéticas

- Inferência em Árvores
- Métodos de Procura?

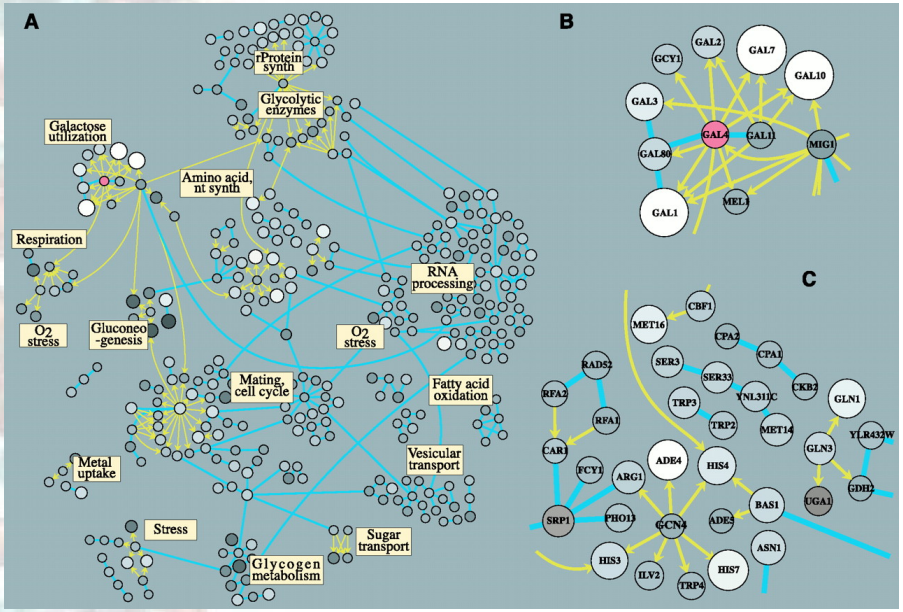


Expressão de Genes

- a figure mostra a expressão de um gene de fermento:
  - ⋆ cada linha é um gene
  - ⋆ coluna representa medida da expressão de genes em certa altura
- vermelho: acima de um certo nível
- azul: abaixo de um certo nível

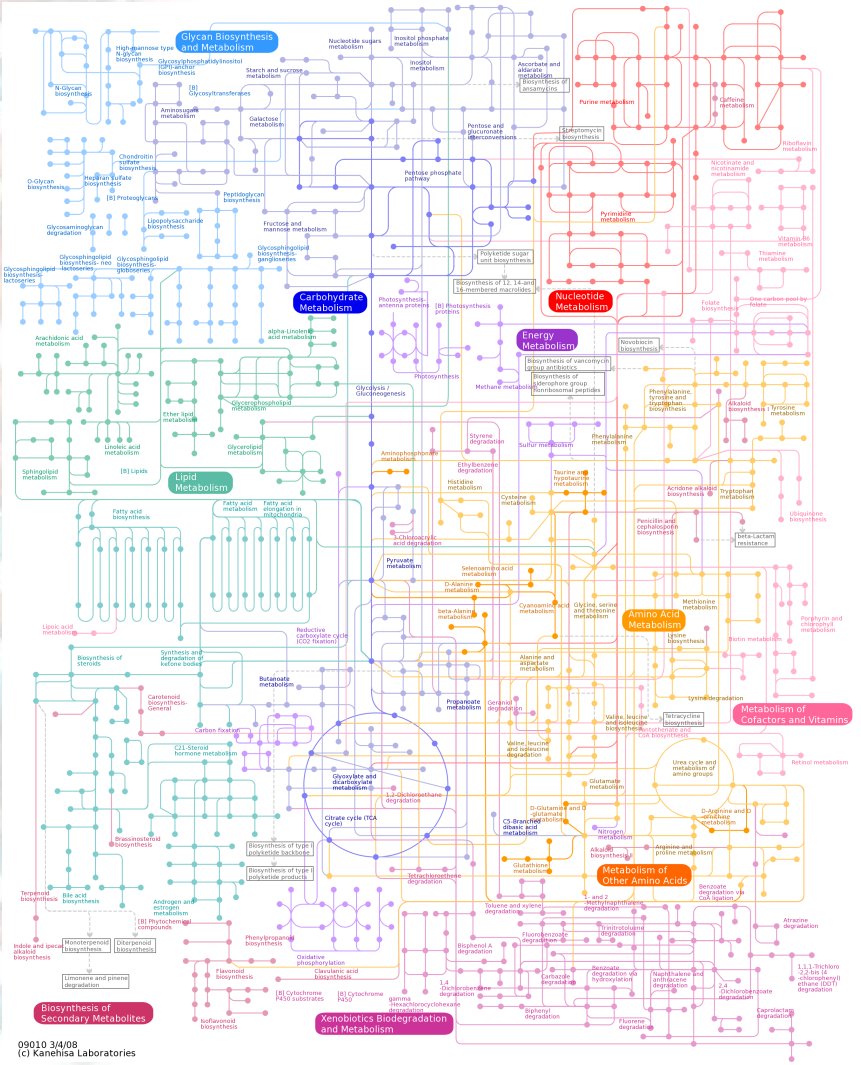


Interacções

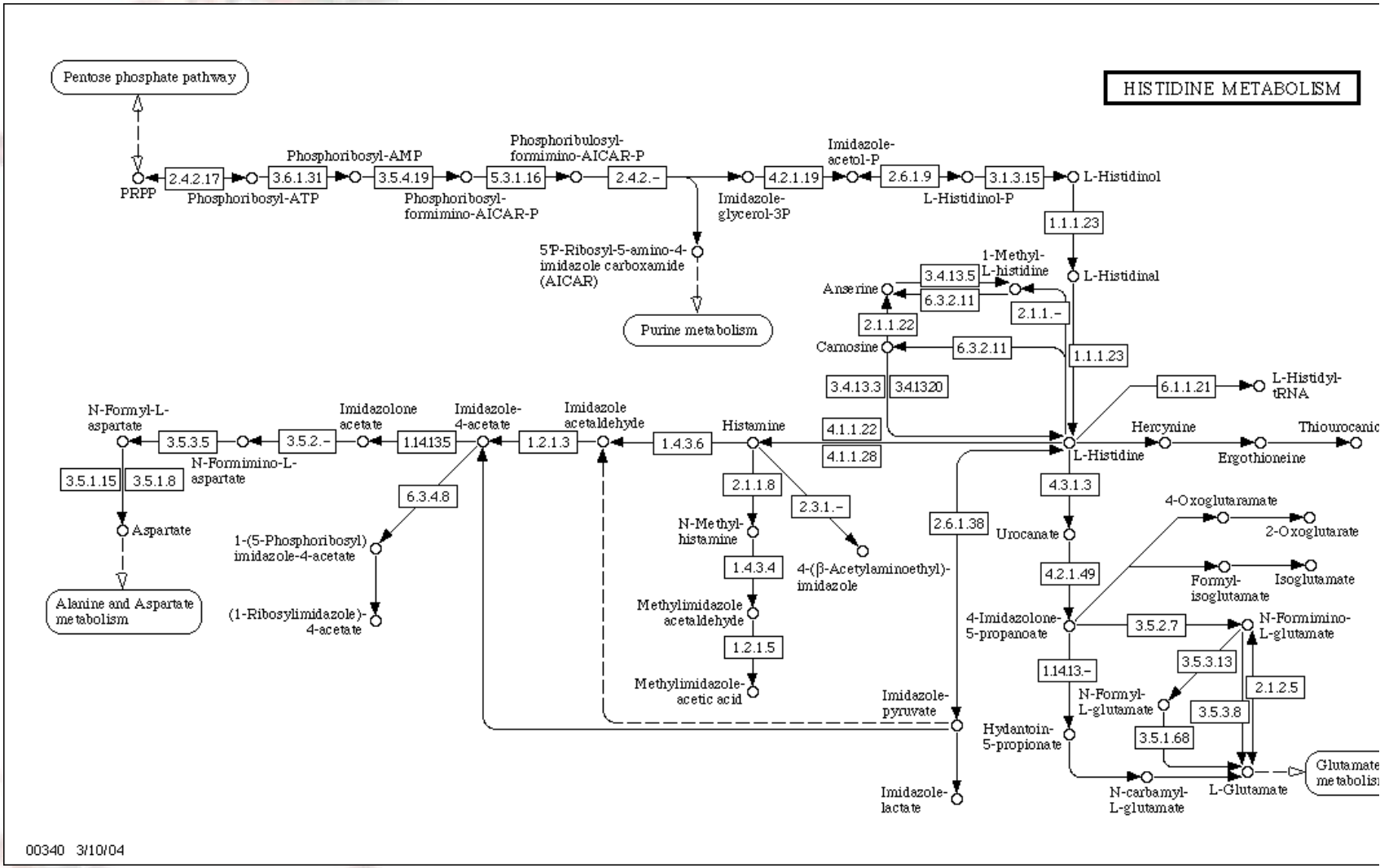


- cada nó representa o produto de um gene (proteína)
- linhas azuis representam interacções directas entre proteínas
- linhas amarelas mostram interacções em que uma proteína associa a DNA e altera a expressão de outra.

Mapa do Metabolismo



Um Betaine: Histidine



A Corrida do Genoma

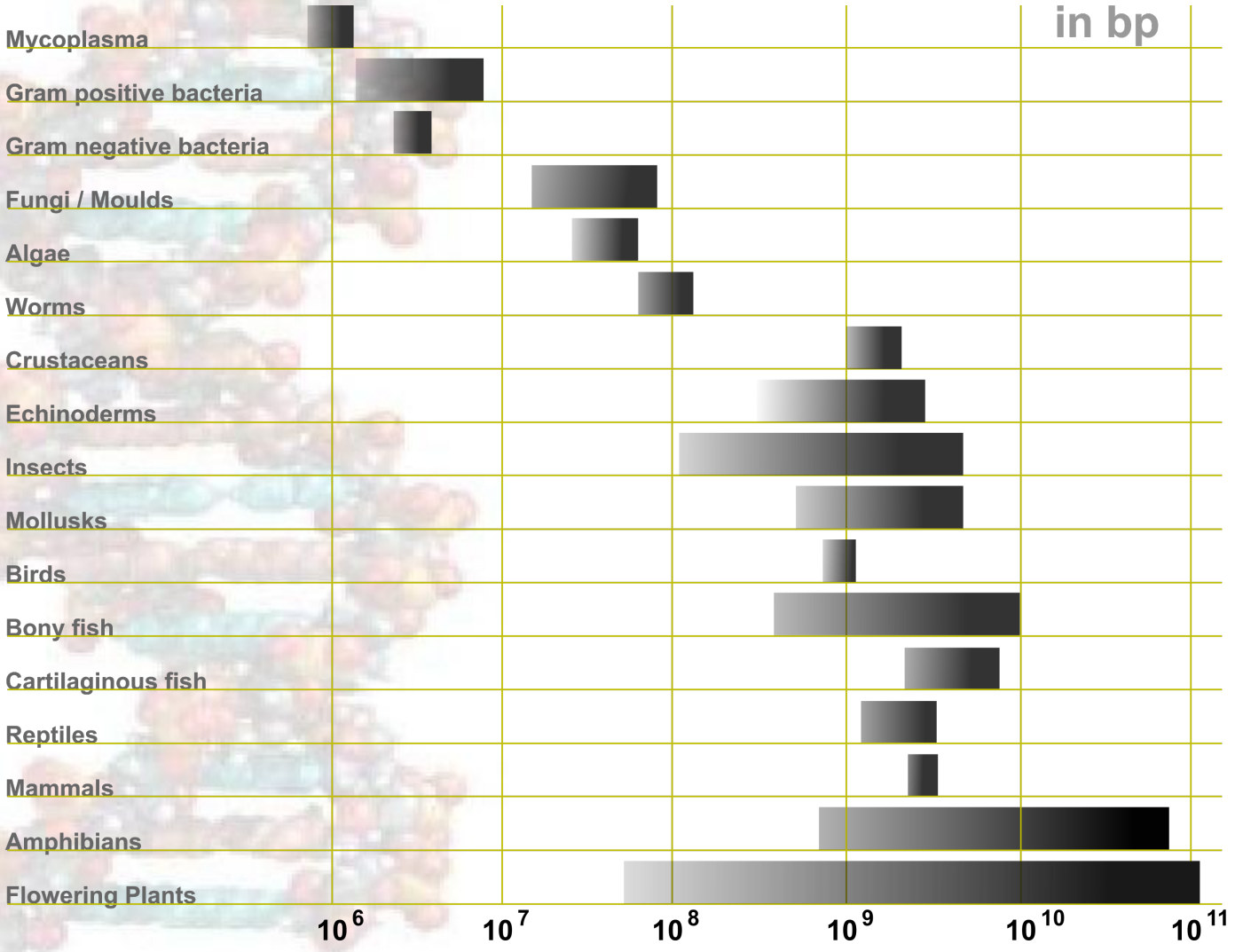
Tipo	Genoma	One	Ano
Procariote	H. Influenza	TIGR	1995
Eucariote	S. Cerevisiae (fermento)	colab. interna.	1997
Animal	C. Elegans (verme)	Washington U./Sanger	1998
Planta	A. thaliana	multiple groups	2000
Mosca:	Drosophila M.	multiple groups	2000
Primata:	H. Sapiens	colaboração internacional/Celera	2001

Genomas Completamente Publicados

Tipo	Número Aproximado
Archaea	117
Bacteria	1644
Eucariotes	153

- dados de GOLD
- Não conta vírus, fagos, etc.
- Organismos multicelulares: rato, fungos, e homo sapiens.
- Em progresso:
  - ⋆ 212 archaea
  - ⋆ 6980 bacterias
  - ⋆ 2013 eucariotes
- 10K genome project

Tamanhos de Alguns Genomas



Há Mais

- $>$ 300 outros bancos de dados sobre biologia nuclear.
- GenBank (Feb 2008):
  - 126,551,501,141 bases
  - 135,440,924 sequências
- UniProt com SWISS-Prot (2011_08):
  - 200346 entradas com sequências de proteínas
  - 188463640 amino-ácidos
- Protein Data Bank (Abril 06):
  - 70209 proteinas e estruturas relacionadas.

Técnicas

- Algoritmos sobre Sequências
- Programação Dinâmica
- Aprendizagem Automática
- Modelos baseados em cadeias de Markov
- Cadeias de Markov escondidas (HMM)
- Algoritmos EM
- Paricionamento de Dados
- Algoritmos sobre Árvores
- . . .

v

Significado da Revolução Genómica

- Biologia baseada em dados:
  - ⋆ genómica funcional
  - ⋆ genómica comparativa
  - ⋆ biologia de sistemas
- Medicina Molecular:
  - ⋆ Identificação de componentes genéticos de várias doenças
  - ⋆ diagnose/prognose a partir de sequências/expressões
  - ⋆ terapia com genes
- Farmacogenómicas:
  - ⋆ Desenvolver drogas altamente especializada
- Toxicogenómicas:
  - ⋆ Que genes são afectadas por que agentes químicos.

Bioinformática Revisitada

Representação/Armazenamento/Recuperação/Análise de dados biológicos sobre sequências (DNA, proteínas)

- estruturas (proteínas)
- funções (proteínas, sinais de sequências)
- níveis de actividade (mRNA, proteínas)
- redes de interacções (caminhos metabólicos, caminhos regulatórios, caminhos de sinalização)

de/entre biomoléculas

Alinhamento de Pares de Sequências

Dada:

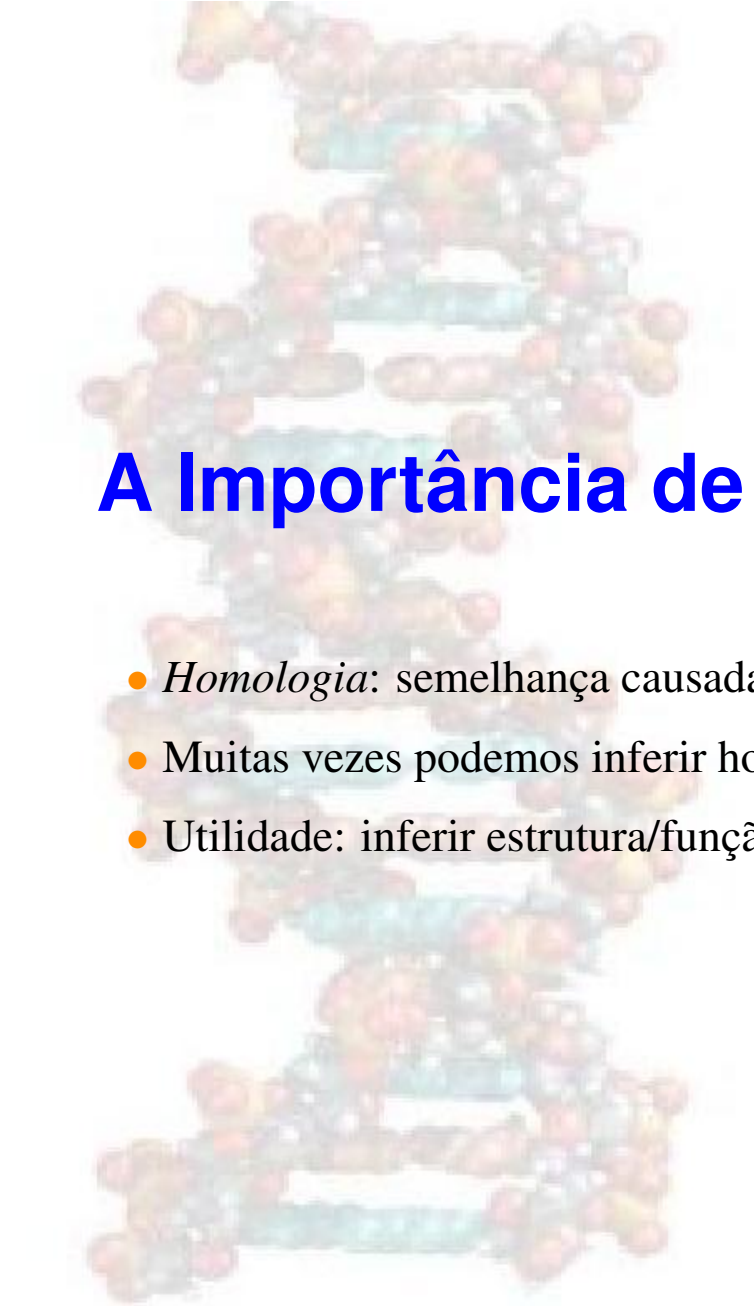
- Um par de sequências (DNA ou proteína)
- um método para calcular a similaridade de um par de caracteres na sequência

Faça

- Encontre as correspondências entre subsequências nas sequências que maximizam uma *função de semelhança*.

Motivação

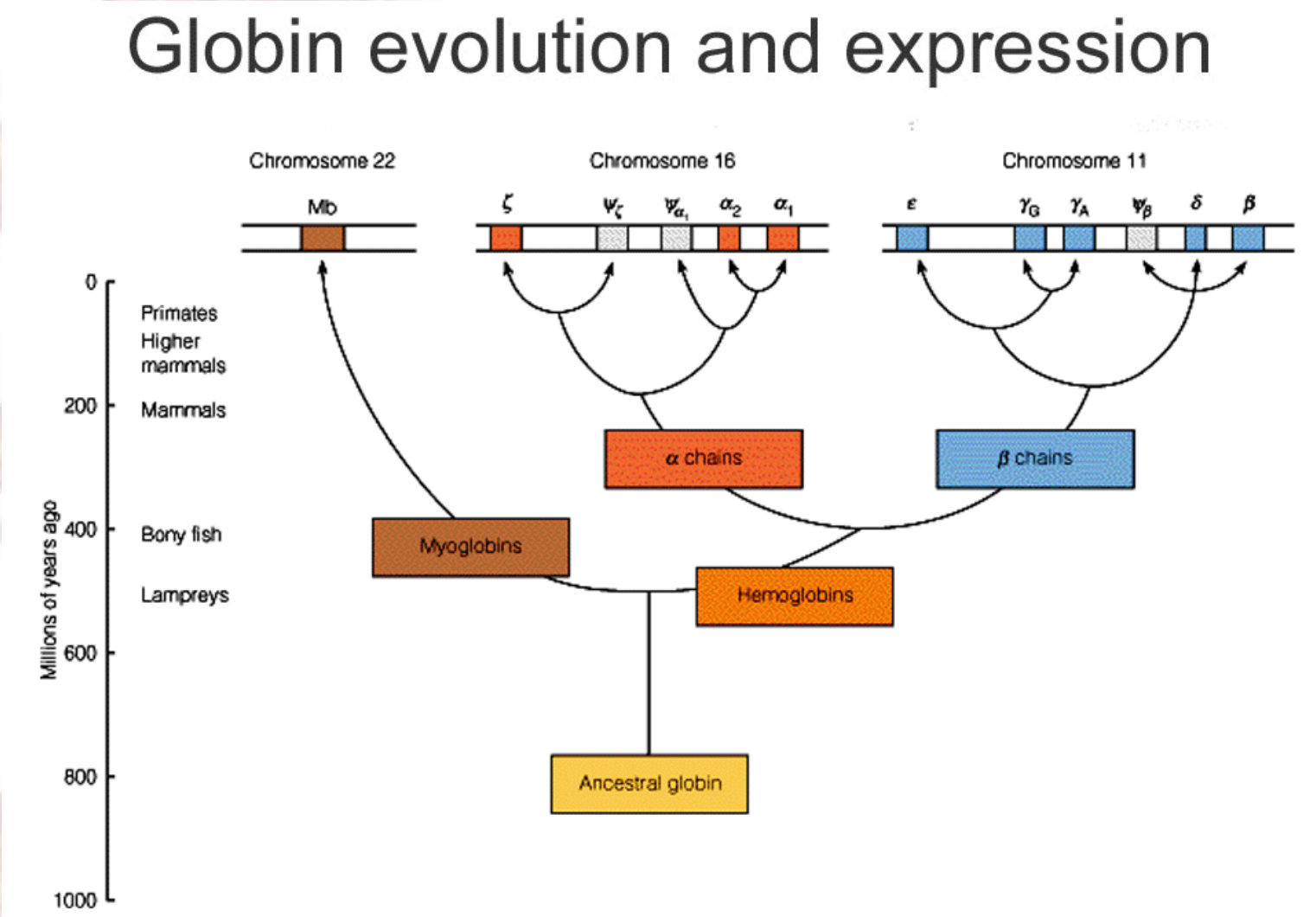
- Comparar sequências para obter informação sobre a estrutura e função de uma sequência.
- Juntar um conjunto de fragmentos de sequência
- Comparar um segmento sequenciado por diferentes laboratórios



A Importância de Homologia

- *Homologia*: semelhança causada por descendência do mesmo antepassado
- Muitas vezes podemos inferir homologia de similaridade
- Utilidade: inferir estrutura/função a partir de similaridade.

Exemplo: Globinas



Homologia

- Sequências homologas podem ser divididas em dois grupos:
  - ⋆ Ortólogas: divergiram para espécies diferentes (eg, α-globina humana e do rato)
  - ⋆ Parálogas: divergiram devido a duplicação de genes na mesma espécie (eg, as várias versões da α-globina humana e da β-globina humana).

Problemas no Alinhamento de Sequências

- As sequências que vamos comparar provavelmente diferem em tamanho
- Pode haver apenas uma pequena região nas sequências que alinha
- queremos permitir alinhamentos parciais: por ex, alguns pares de amino-ácidos são mais substituíveis do que outros
- regiões de tamanho variável podem ter sido inseridas ou removidas do antepassado comum.

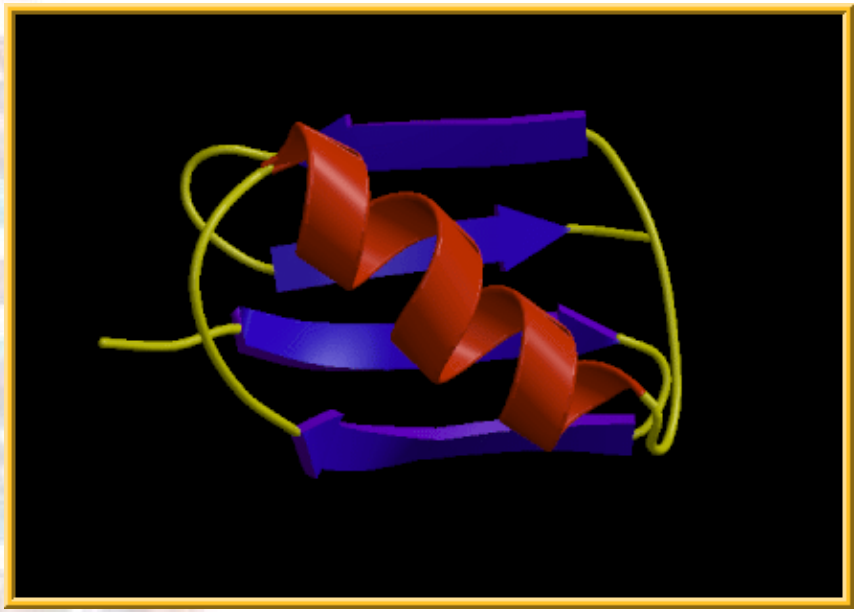
Buracos

- Sequências podem ter divergido do antepassado comum através de vários tipos de mutações:
  - ⋆ Substituições: $(ACGA \longrightarrow AGGA)$
  - ⋆ Inserções: $(ACGA \longrightarrow ACCGA)$
  - ⋆ Remoções: $(ACGA \longrightarrow AGA)$
- os últimos dois casos correspondem a buracos no alinhamento.

Inserções e Remoções vs Estrutura da Proteína

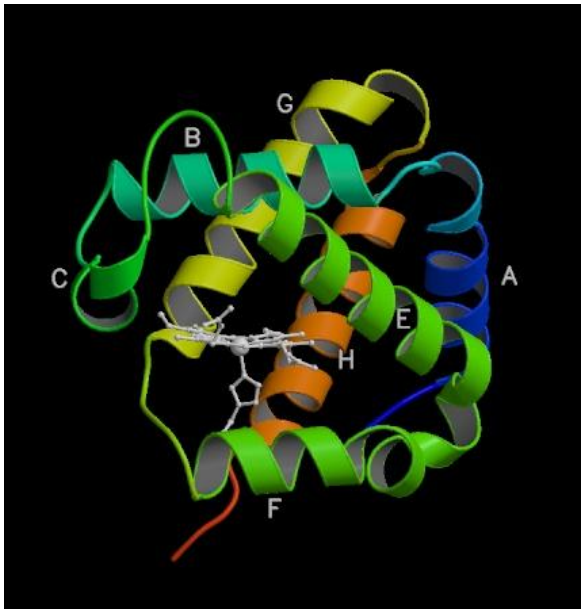
Porque é que duas sequências semelhantes podem ter muitas inserções ou remoções

- Inserções e remoções podem não afectar significativamente a estrutura da proteína.



Exemplo de Alinhamento: Globinas

- À direita, estrutura tipo de Globinas
- Em baixo, parte do Alinhamento para oito globinas




```
                A0   A4   A8   A12            B1        B6        B14  C2        CD1  CD4
                ↓    ↓    ↓    ↓              ↓         ↓         ↓    ↓         ↓    ↓
Hb_a  ----------VLSPADKTNVKAAWGKVGA----HAGEYGAEALERMFLSFPTTKTYFPHF
Hb_b  ----------VHLTPEEKSAVTALWGKV------NVDEVGGEALGRLLVVYPWTQRFFESF
Mb_SW ----------VLSEGEWQLVLHVWAKVEA----DVAGHGQDILIRLFKSHPETLEKFDRF
LegHb ---------GALTESQAALVKSSWEEFNA----NIPKHTHRFFILVLEIAPAAKDLFSFL
BacHb --------LDQQTINIIKATVPVLKEHG------V-TITTTFYKNLFAKHPEVRPLF---
SeaHb GGTLAIQAQGDLTLAQKKIVRKTWHQLMR----NKTSFVTDVFIRIFAYDPSAQNKFPQM
AscHb ------------ANKTRELCMKSLEHAKVDTSNEARQDGIDLYKHMFENYPPLRKYFKS-
Eryt. -----------LSADQISTVQASFDKVKG---------DPVGILYAVFKADPSIMAKFTQF
```

Tipos de Alinhamento

- *Global*: encontrar o melhor alinhamento entre sequências completas
- *Local*: encontrar o melhor alinhamento entre subsequências completas
- *Semi-Global*: encontrar o melhor alinhamento sem penalizar espaços brancos nas bordas do alinhamento

Como Avaliar um Alinhamento

- Matriz de substituição:
  - $\star$ $s(a, b)$ indica o preço de alinhar o caracter a com o caracter b.
- Função de penalização de intervalos:
  - $\star$ $w(k)$ indica o custo de um intervalo de tamanho k.

Função de Penalização Linear

- Diferentes funções de penalização podem requerer algoritmos de programação dinâmica diferente
  - ⋆ O caso mais simples é quando usamos uma função linear:

$$w(k) = gk$$

onde g é uma constante

- Vamos começar por aqui.

Pontuação de Alinhamentos

- A pontuação de um alinhamento é:
  1. somatório dos pares de caracteres alinhados,
  2. mais pontuação para buracos
- Exemplo: dado o alinhamento

```
VAHV---D--DMPNALSALSDLHAHKL

AIQLQVTGVVVTDATLKNLGSVHVSKG
```

- a pontuação será:

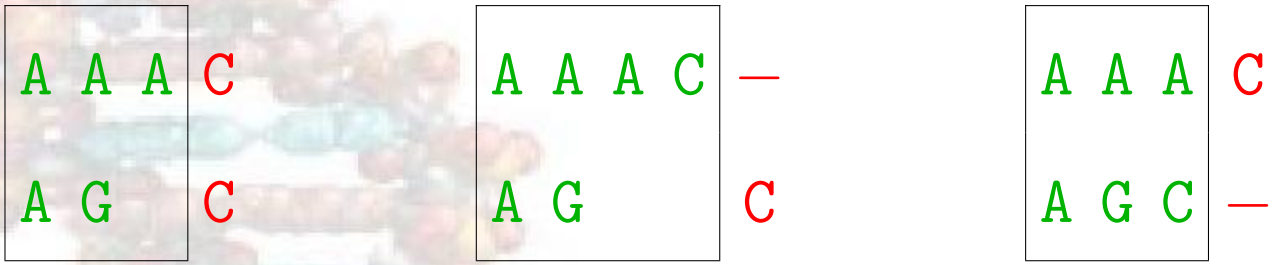
$$s(\mathtt{V},\mathtt{A}) + s(\mathtt{A},\mathtt{I}) + s(\mathtt{H},\mathtt{Q}) + 3g + s(\mathtt{D},\mathtt{G}) + \ldots$$

Alinhamento de Pares por Programação Dinâmica

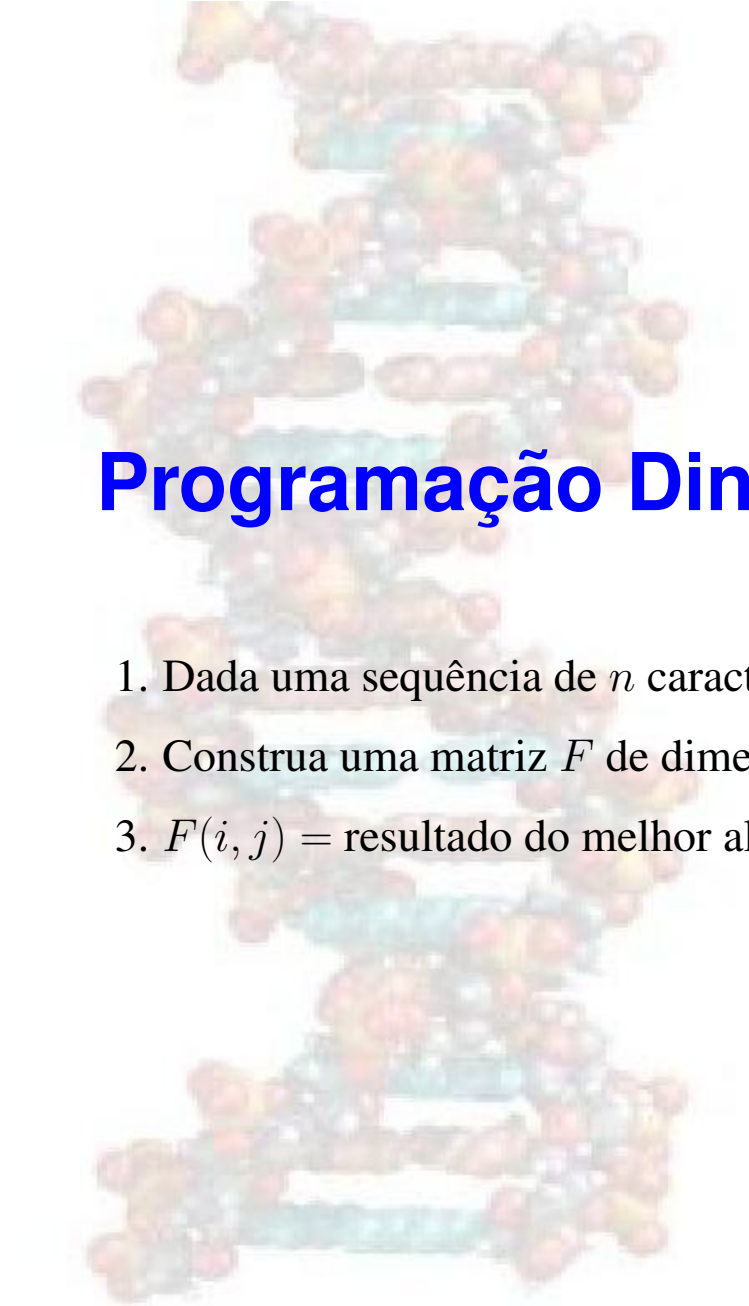
- Needleman & Wunsch, Journal of Molecular Biology, 1970
- *Programação Dinâmica*: resolver uma instância de um problema usando soluções computadas para pequenas partes do problema.
- Ideia: determinar alinhamento óptimo de duas sequências determinando o melhor alinhamento para todos os prefixos.

Alinhamento de Pares por Programação Dinâmica

- Considere o último passo na computação do alinhamento de `AAAC` com `AGC`
- Três opções possíveis:



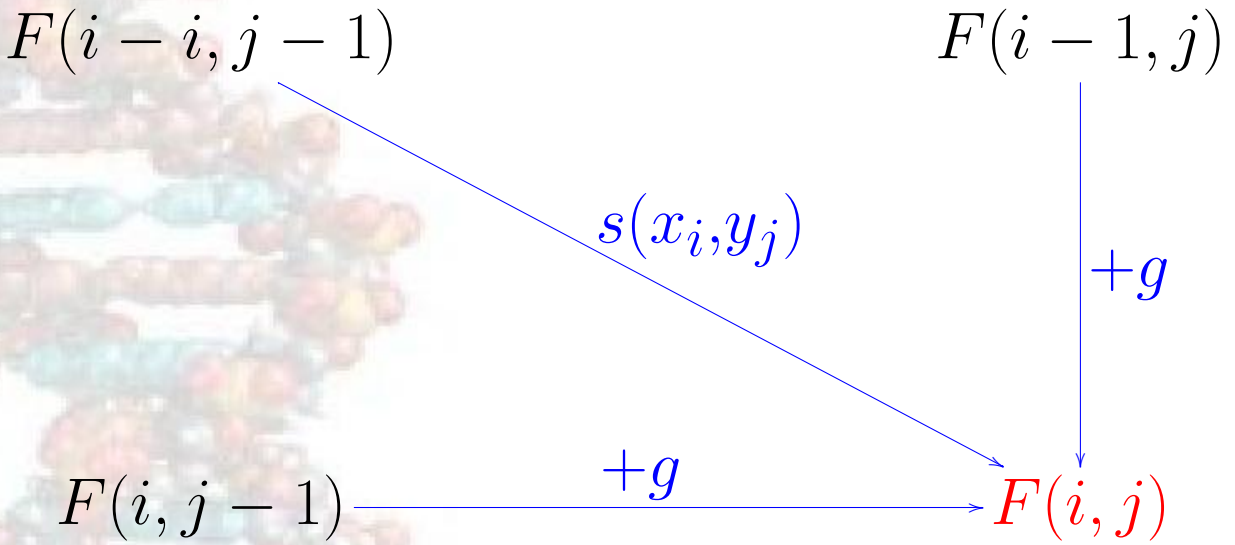
- Considere:
  1. Melhor Alinhamento dos Prefixos +
  2. Resultado do Alinhamento do par



Programação Dinâmica

1. Dada uma sequência de n caracteres x e uma sequência de m caracteres y,
2. Construa uma matriz F de dimensão $(n+1) \times (m+1)$
3. $F(i,j) =$ resultado do melhor alinhamento de $x[1 \ldots i]$ com $y[1 \ldots j]$.

Programação Dinâmica: Ideia Básica

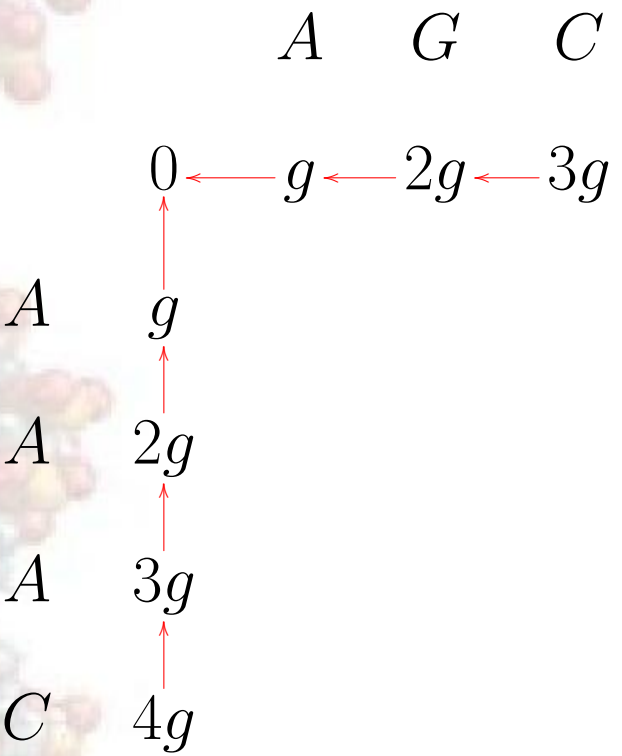


Algoritmo para Alinhamento Global com Penalização Linear de Buracos

Uma maneira é especificar a DP através da sua relação de recorrência:

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) + d \\ F(i,j-1) + d \end{cases}$$

Inicialização da Matriz



Esquema do Algoritmo

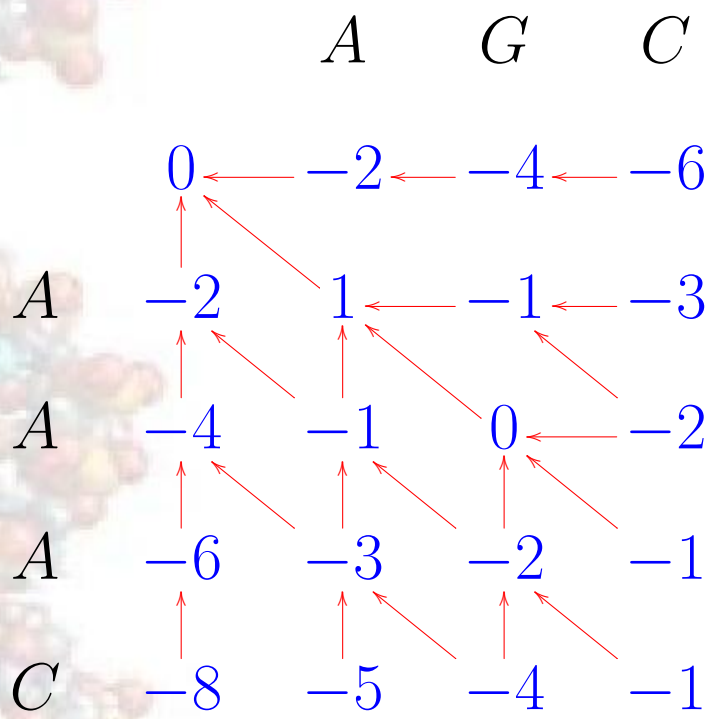
- inicializar primeira linha e coluna da matriz
- preencher o resto da matriz de cima para baixo, e esquerda para a direita
- para cada $F(i, j)$, guarde ponteiro para célula que deu o melhor resultado
- $F(m, n)$ tem a pontuação de alinhamento óptima: siga os ponteiros desde $F(m, n)$ até $F(0, 0)$ para recuperar o alinhamento.

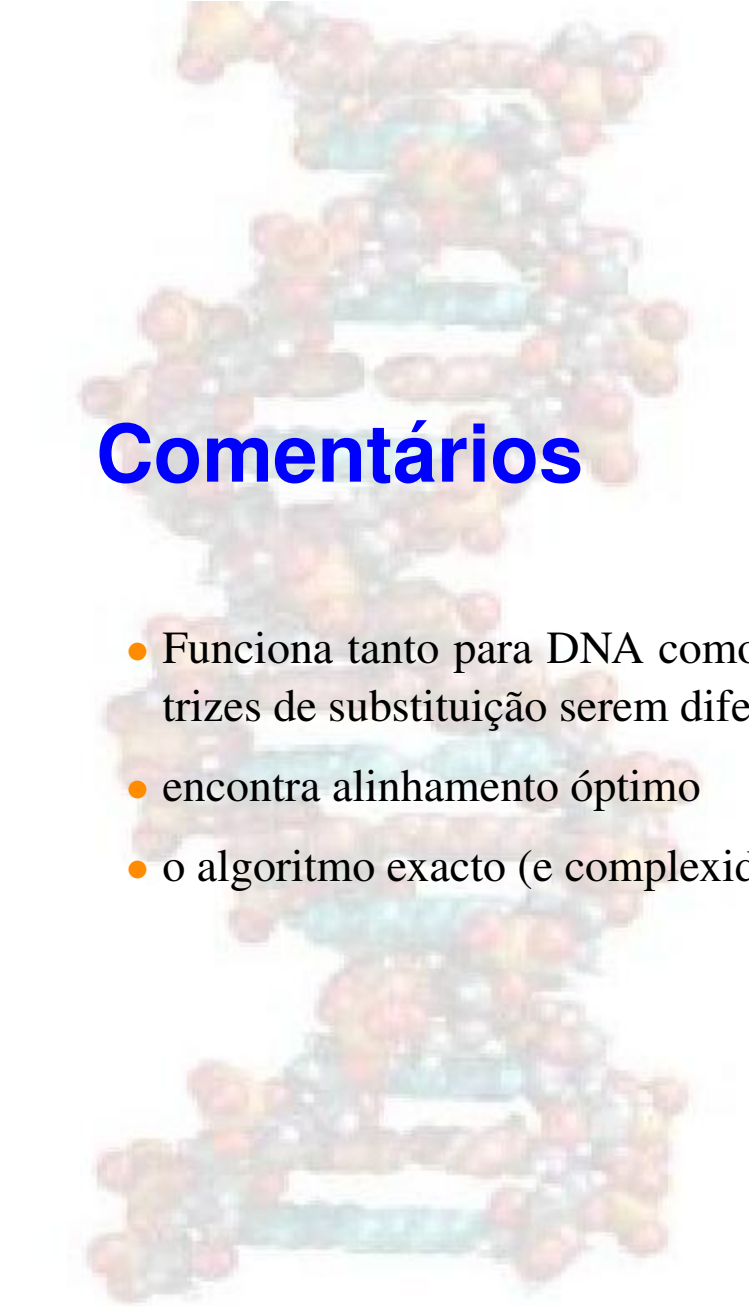
Exemplo do Esquema do Algoritmo

Imagine que escolhíamos o seguinte esquema de pontuação:

- acerto: $+1$
- erro: -1
- $g(\text{penalidade para alinhar com um buraco}) = -2$

Exemplo do Esquema do Algoritmo



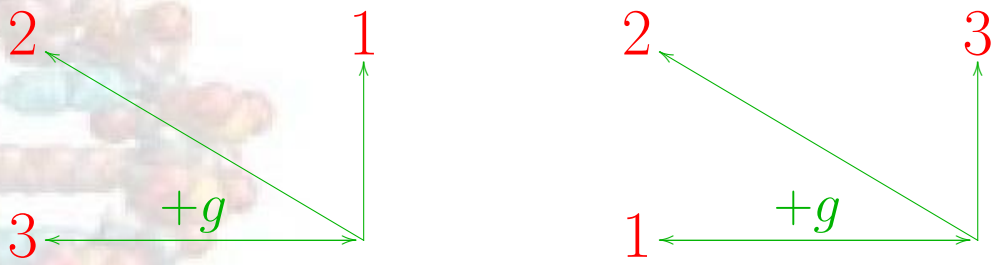


Comentários

- Funciona tanto para DNA como para sequências de amino-ácidos, apesar das matrizes de substituição serem diferentes
- encontra alinhamento óptimo
- o algoritmo exacto (e complexidade) depende da função de penalização de buracos

Alinhamentos Igualmente Óptimos

- muitos alinhamentos óptimos podem existir para um par dado de sequências
- podemos usar escolha de preferências sobre caminhos quando voltamos para trás:



- O *caminho alto* e o *caminho baixo* mostram os dois alinhamentos óptimos mais diferentes.

Análise do Algoritmo de Programação Dinâmica

- Caminho alto:

```
x:  A  A  A  C
y:  A  G  -  C
```

- Caminho baixo:

```
x:  A  A  A  C
y:  -  A  G  C
```

Análise do Algoritmo de Programação Dinâmica

- Existem

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

  alinhamentos possíveis para 2 sequências de tamanho n

- ie, duas sequências de tamanho 1000 têm $\approx 10^{600}$ alinhamentos possíveis
- mas o algoritmo DP encontra o alinhamento óptimo eficientemente.

Complexidade Computacional

- inicialização: $O(m), O(n)$
- preenchendo o resto da matriz: $O(mn)$
- voltar para trás: $O(m + n)$

- se as duas sequências tiverem o mesmo tamanho, a complexidade computacional é:

$$O(n^2)$$

Alinhamento Local

- Até agora discutimos *alinhamento globais*, onde estamos procurando o melhor emparelhamento de duas sequências desde um fim ao outro
- mais frequentemente, queremos um *alinhamento local*, o melhor alinhamento entre subsequências de x e y

Motivação

- útil para comparar sequências de proteínas que partilham um *motivo* (padrão conservado) ou *domínio* (unidade independente enrolada) mas que diferem no resto
- útil para comparar sequências de DNA que partilham um *motivo* (padrão conservado) mas que diferem no resto
- útil para comparar sequências de proteínas contra *sequências de DNA do genoma* (longos grupos de sequências não caracterizadas)
- mais preciso para comparar sequências que divergiram muito

Algoritmo de Alinhamento Local por DP

- formulação original: Smith & Waterman, *Journal of Molecular Biology*, 1981
- Interpretação das matrizes é um pouco diferente:
  - $F(i, j) =$ pontuação do melhor alinhamento de um sufixo de $x[1 \ldots i]$ e um sufixo de $y[1 \ldots j]$

Algoritmo de Alinhamento Local por DP

- Inicialização: primeira linha e coluna inicializada com 0s
- Retorno:
  - ⋆ encontrar valor máximo de $F(i, j)$; pode ser em qualquer posição da matriz
  - ⋆ parar quando encontrar uma célula com o valor 0.

Exemplo de Alinhamento Local


```
x:   A A G
y:   A A G
```

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:

$$O(n^3)$$

Matrizes

- PSSM (PSI-BLAST)
  - ⋆ Estimar uma probabilidade do AA em cada coluna
  - ⋆ Dividir pela probabilidade do AA
  - ⋆ Tirar log: se positivo é provável
- Matrizes de Substituição:
  - ⋆ PAM [Dayhoff et al, 1978]
  - ⋆ BLOSUM [Henikoff & Henikoff, 1992]

BLOSUM62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

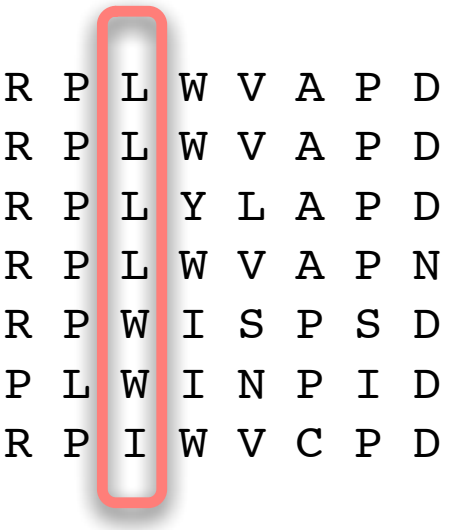
BLOSUM62

- Começar por BD com segmentos bem conservados
- Exemplo: BLOCKS

```
ID   XRODRMPGMNTB; BLOCK
AC   PR00851A; distance from previous block=(52,131)
DE   Xeroderma pigmentosum group B protein signature
BL   adapted;   width=21; seqs=8; 99.5%=985; strength=1287
XPB_HUMAN|P19447    (  74) RPLWVAPDGHIFLEAFSPVYK  54
XPB_MOUSE|P49135    (  74) RPLWVAPDGHIFLEAFSPVYK  54
P91579              (  80) RPLYLAPDGHIFLESFSPVYK  67
XPB_DROME|Q02870    (  84) RPLWVAPNGHVFLESFSPVYK  79
RA25_YEAST|Q00578   ( 131) PLWISPSDGRIILESFSPLAE 100
Q38861              (  52) RPLWACADGRIFLETFSPLYK  71
O13768              (  90) PLWINPIDGRIILEAFSPLAE 100
O00835              (  79) RPIWVCPDGHIFLETFSAIYK  86
//
```

BLOSUM: Pares

- Contar Pares por Colunas
- $L/L : 3 + 2 + 1 = 6$
- $L/W : 4 + 4 = 8$
- $L/I : 4$
- $W/W : 1$



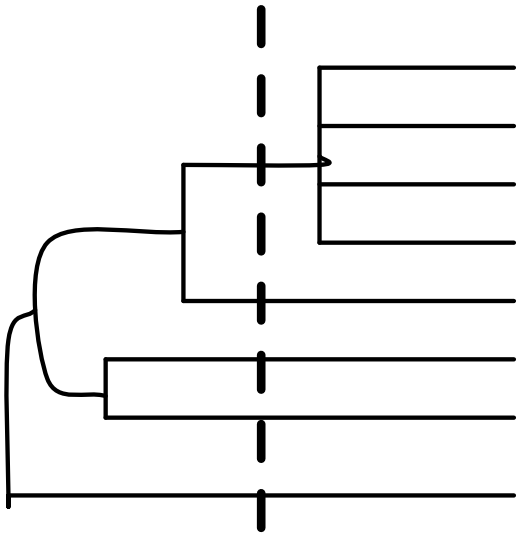
BLOSUM: Algoritmo

- Contar Pares f_{ab} em Todas Colunas da BD (> 20000)
- Calcular $p_{ab} = \frac{f_{ab}}{\sum_{i=1}^{20} \sum_{j=1}^{i} f_{ij}}$
- Calcular $q_a = \sum_{b=1}^{20} (\frac{f_{ab}}{\sum_{i=1}^{20} \sum_{j=1}^{i} f_{ij}})$
- $s(a,b) = log_2(\frac{p_{ab}}{q_a q_b})$
- Fazer arredondamento e ajustar (half-bits)

BLOSUMXX: Distância Evolutiva

- Agrupar sequências que estão perto evolucionariamente

- Juntos os elementos dos cluster $> XX$ valem 1



BLOSUM: Análise

- A melhor pontuação é para W (triptofano)
  - Relativamente raro
- O pior para * com AA, e exs como D e L
  - Ácido aspártico está a 2 mutações de Leucina
  - Polaridade e Carga Diferentes

BLOSUM vs PAM

- Menos tolerante de substituições de hidrfílicos
- Mais tolerante de mudanças entre hidrofibicidade
- Mais tolerante de falhas com cisteína e triptofan.

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 SwissProt contra uma sequência de interrogação:
  - ⋆ imaginem que a nossa pergunta tem 362 amino-ácidos
  - ⋆ SwissPROT versão 38 contém 532.146 sequências com 188.719.038 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

Motivação para Alinhamento Por Heurísticas

- ⋆ imaginem que a nossa pergunta tem 362 amino-ácidos
  - ⋆ SwissPROT versão 38 contém 532.146 sequências com 188.719.038 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

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

BLAST em Ação

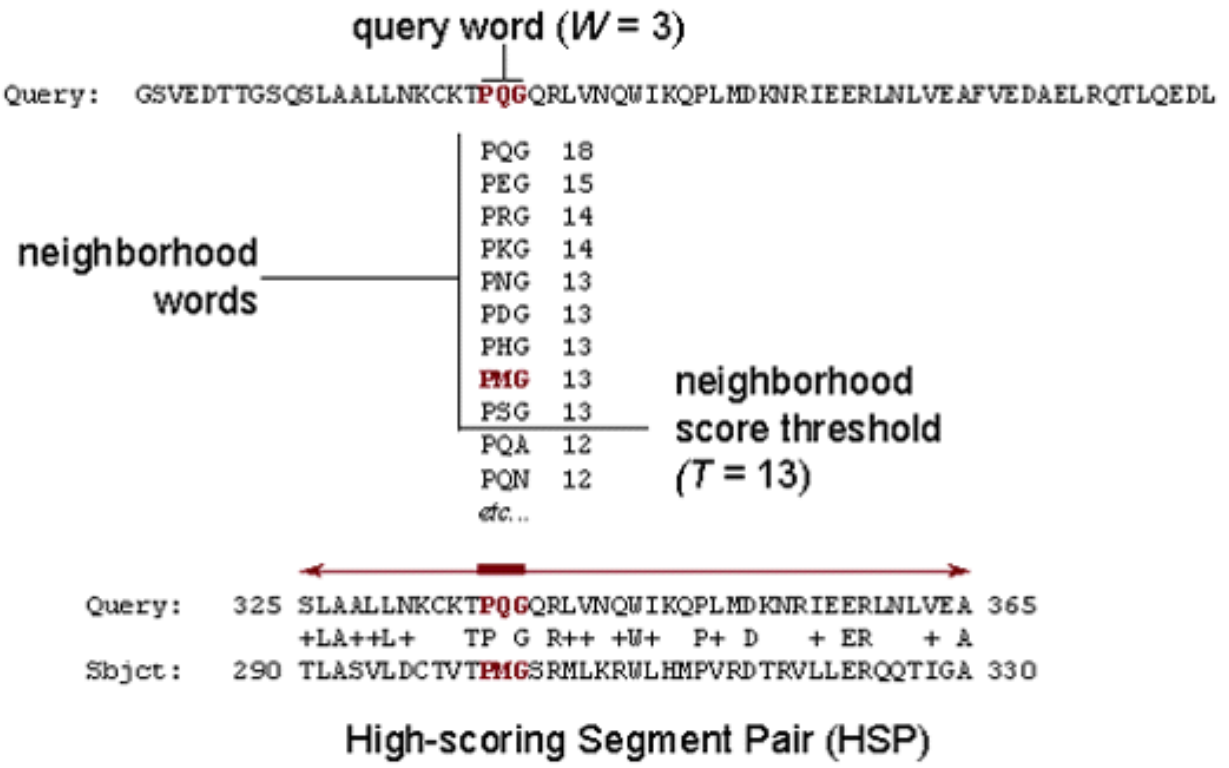
- Procurar em PDB “hen egg-white lysozyme”
- Também “Human oxyhemoglobin”
- Fazer BLAST em Swiss-Prot

Ideia Principal

- 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



High-scoring Segment Pair (HSP)

Determinação de Palavras da Interrogação

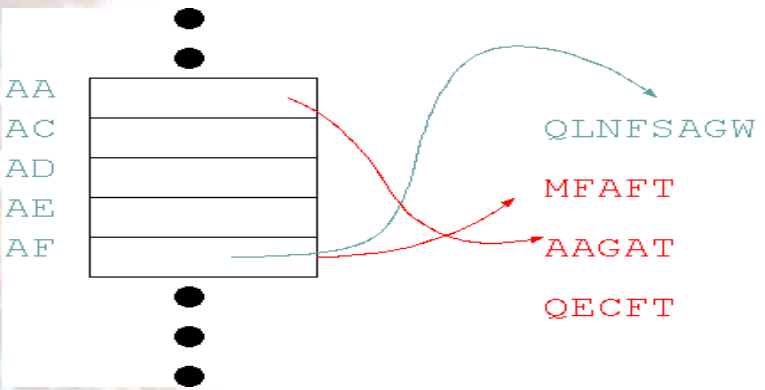
- Dada:
  - ⋆ sequência de interrogação: `QLNFSAGW`
  - ⋆ tamanho de palavra $w = 2$ (para proteína usualmente $w = 3$)
  - ⋆ limite para pontuação de palavra; $T = 8$
- Passo 1: determinar todas as palavras de tamanho w na sequência de interrogação:
  - ⋆ `QL LN NF FS SA AG GW`

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)

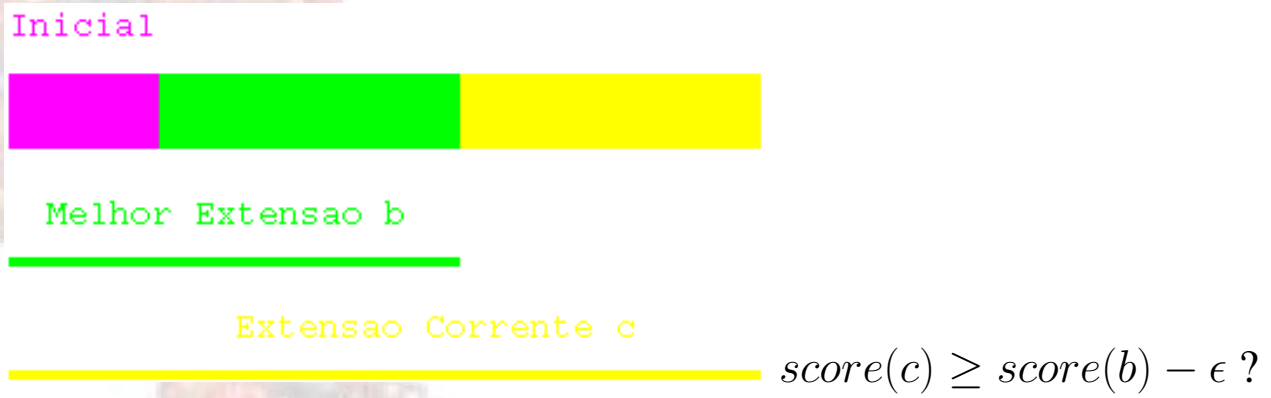
Procurar na BD

- 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

P-Value

- Se na BD $\mathcal{D}$ a sequência X tem a pontuação $S(\mathcal{D}, X) = s$, 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 $\geq$ por acaso
    - ∗ 10^{-6}: apenas 1 em 1000000!
- Como estimar *p-value* no BLAST?

Pontuação Simplificada

- Score Simplificado:
  - 1 para acerto
  - $-\infty$ para miss ou buraco
- Pontuação melhor:
  - maior alinhamento
- Se matriz de alinhamento tem tamanho $n \times m$:
  - Alinhamento pode começar $\approx n \times m$ posições

Pontuação Simplificada

- Se $P(\text{probabilidade de 2 letras à sorte serem iguais}) = p$
- Em geral, a probabilidade de alinhamento de tamanho $\geq t$:

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

  - $1-p$: se começou aqui, antes não era alinhamento
  - $p \times \ldots \times p$

- Logo, número esperado de alinhamentos de tamanho $\geq t$:

$$\approx nm(1-p)p^t$$

Pontuação Simplificada

- Aproximar Binomial por Poisson
  - $N \to \infty$ se NP fixo
- No caso:
  - $N = mn$
  - $P = (1-p)p^t$
- Podemos aproximar por Poisson com $\lambda = mn(1-p)p^t$

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

Pontuação: Explicação

- BLAST usa:

$$p - value \approx 1 - e^{-Knme^{-\lambda S}}$$

  - K e λ são parâmetros estimados.

- Valor-E:

$$E = Knme^{-\lambda S}$$

Intuição

- Dobrar o tamanho de qq sequência dobra o número de sequências para o score.
- Para obter $2 \times S$ tem que duplicar o tamanho, logo E deveria decrescer exponencialmente.
- K e λ dependem:
  - ⋆ da matriz de substituição
  - ⋆ da frequência dos amino-ácidos
- Mais intuitivo do que p-value para matches fracos
  - ⋆ E-value de 5, P-value de 0.993
  - ⋆ E-value de 10, P-value de 0.99995
- E e p-value convergem para matches fortes

Bit-Score

- $E = -Knme^{-\lambda S}$, depende de K e λ
- Bit-Score S' elimina K e λ:

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

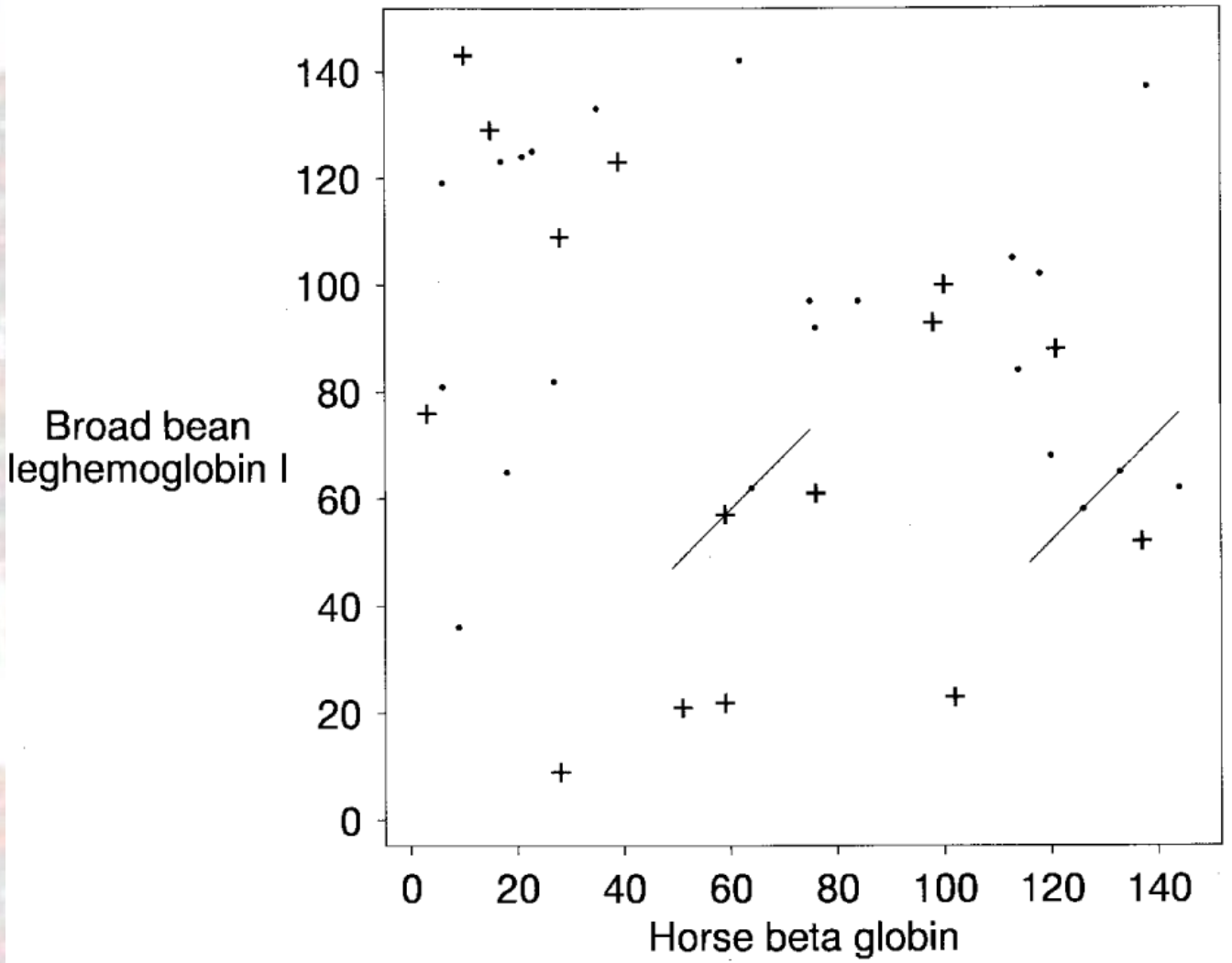
- depende apenas de m, n
- Valor-E e S':

$$E = mn2^{-S'}$$

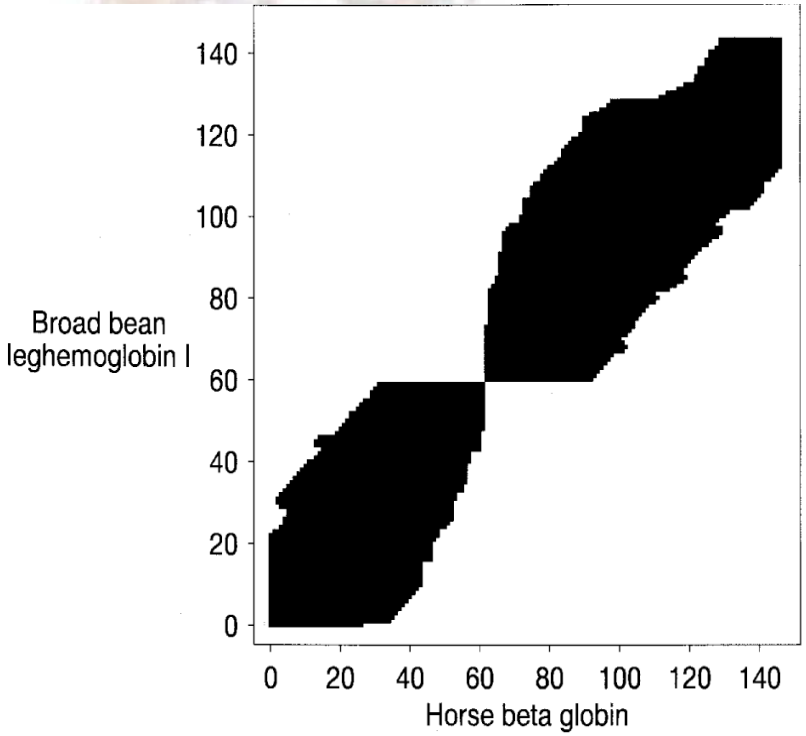
Extensões de BLAST

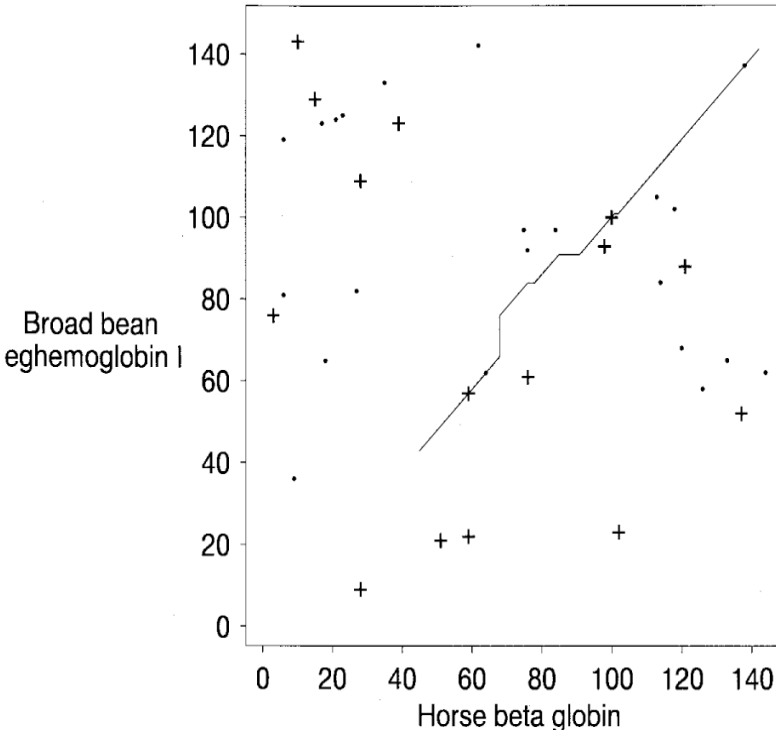
- O método *two-hit*: ampliar apenas quando há dois acertos perto e na mesma diagonal
  - ⋆ permite abaixar T
- BLAST com buracos:
  - ⋆ usar DP a partir de alinhamento com pontuação melhor
- PSI-BLAST: generalizar iterativamente a questão (fazê-la parecer mais como acertos) e voltar a procurar
- Todas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

Método Two-Hits



Gapped Blast



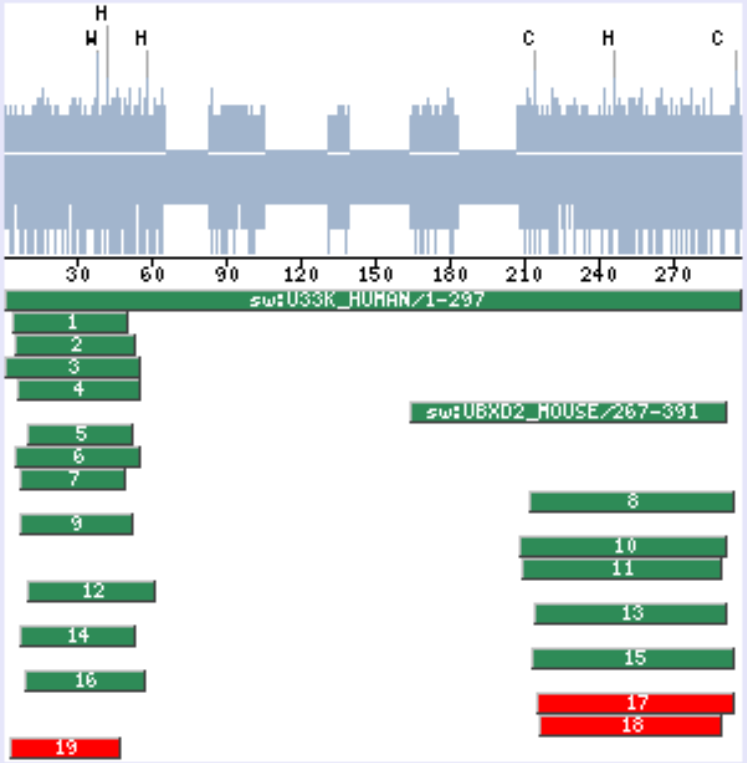



```
Leghemoglobin  43 FSFLKDSAGVVDSPKLGAHAEKVFGMVRDSAVQLRATGEVV--LDGKDGS------  90
                  F  L +   V+ +PK+ AH +KV          L + GE V  LD   G+
Beta globin    45 FGDLSNPGAVMGNPKVKAHGKKV----------LHSFGEGVHHLDNLKGTFAALSE  90


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



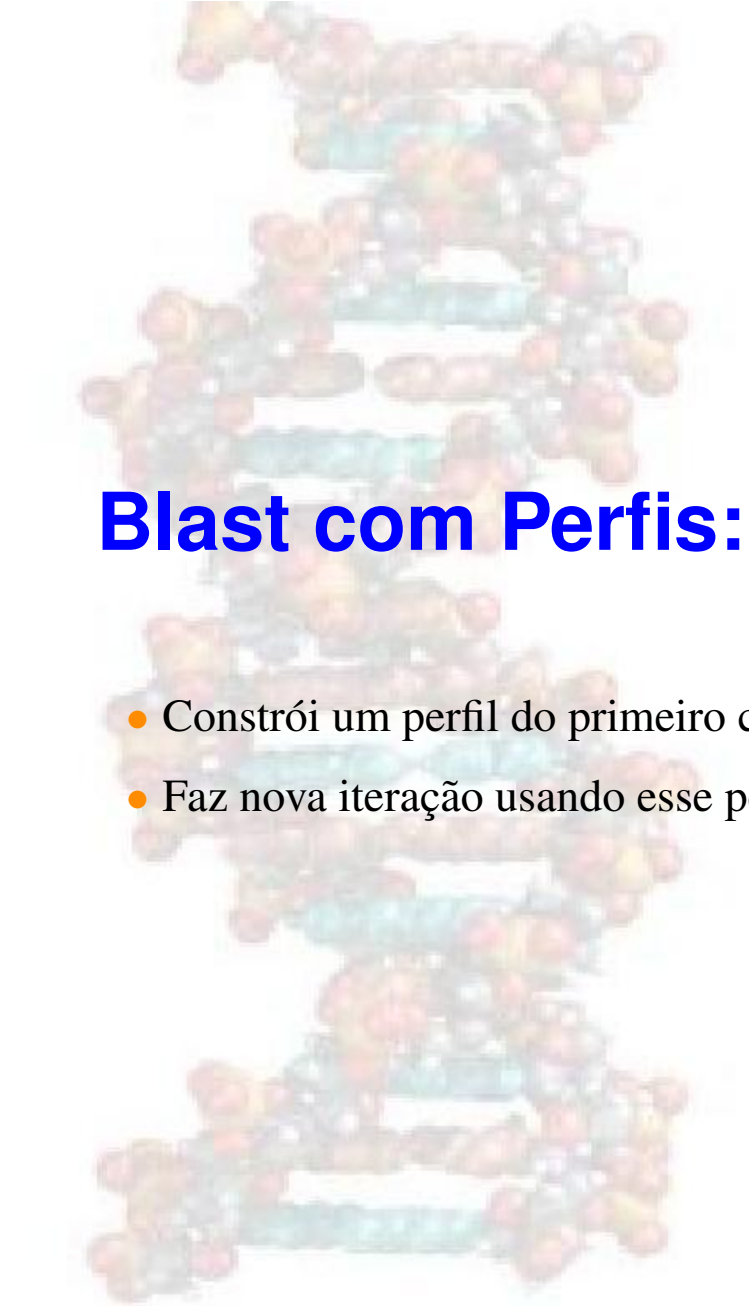
Legends: **1**, sw:UAS3_DROME/22-68; **2**, sw:UBP14_SCHPO/580-628; **3**, sw:UBP5_MOUSE/654-708; **4**, sw:UBP14_YEAST/612-661; **5**, sw:UBP14_ARATH/622-664; **6**, sw:UBP13_HUMAN/656-706; **7**, sw:UBPA_DICDI/634-676; **8**, sw:YOJ8_CAEEL/281-359; **9**, sw:UAS3_HUMAN/25-70; **10**, sw:UBX7_YEAST/211-290; **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.

Blast com Perfis

- um perfil é uma descrição de um conjunto de sequências
- colunas representam posições em sequências
- linhas representam caracteres em sequências
- elementos representam a abundância de um caracter numa posição

aminoacidos

	1	2	3	4	5	6	7	8	...
A			0						
R			0						
D			0.5						
N			0.2						
C			0						
			⋮						



Blast com Perfis: PSI-BLAST

- Constrói um perfil do primeiro conjunto de alinhamentos
- Faz nova iteração usando esse perfil.

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.
- Zhang, Z., Schwartz, S., Wagner, L. Miller, W. A greedy algorithm for aligning DNA sequences. J. Computational Biology (2000) 7:203-214.
- 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.

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 nucleótidos (1)
- N: perca por mismatch de nucleótidos (-2)
- Buracos: usa linear (nucleótidos) e 11/1 (AA)
- Matriz BLOSUM62
- Tamanho de palavra:
  - ⋆ 28 para nucleótidos
  - ⋆ 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.

Modelos Para Alinhamentos

- Na aula passada usamos pontuação como $\equiv$ comprimento
- Não faz sempre sentido
- Como avaliar alinhamento?
  - ⋆ Probabilidades do Alinhamento
- Vamos assumir que cada posição é independente

Modelos de Sequências

- Independentes:

$$Pr(x, y|U) = \prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}$$

- Dependentes:

$$Pr(x, y|R) = \prod_{i=1}^{n} p_{x_i, y_i}$$

Comparando Modelos

- Qual o mais provável?
- Comparar a verosimilhança dos 2:

$$\frac{Pr(x,y|R)}{Pr(x,y|U)} = \frac{\prod_{i=1}^{n} p_{x_i,y_i}}{\prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} bq_{y_i}} = \prod_{i=1}^{n} \frac{p_{x_i,y_i}}{q_{x_i} q_{y_i}}$$

- É mais fácil trabalhar com o log:

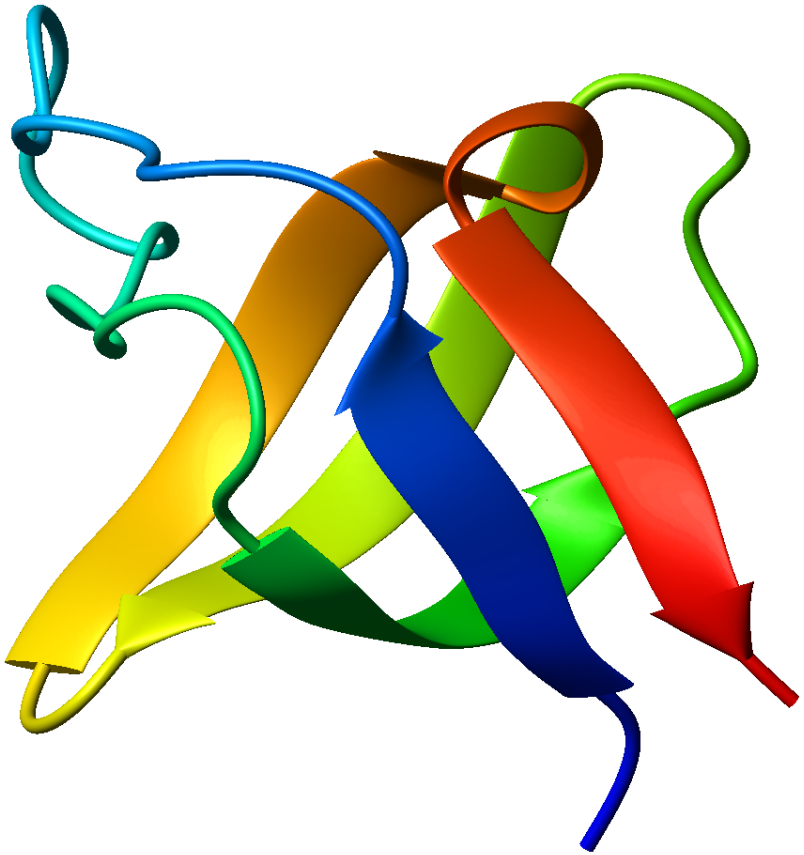
$$log \frac{Pr(x,y|R)}{Pr(x,y|U)} = \sum_{i} \frac{p_{x_i,y_i}}{q_{x_i} q_{y_i}}$$

Alinhamentos Múltiplos

- caracterizar um conjunto de sequências (ie, uma classe de sinais DNA)
- caracterizar uma família de proteínas:
  - ⋆ o que é conservado
  - ⋆ o que varia
- geração de perfis para procura

SH3

- Domínio envolvido em tradução de sinal para cito-esqueleto, RAS
- Estrutura típica de barril-beta parcialmente aberto
- ≈ 60 AAs



Alinhamento Múltiplo em SH3

```
G G W W R G d y . g g k k q L W F P S N Y V
I G W L N G y n e t t g e r G D F P G T Y V
P N W W E G q l . . n n r r G I F P S N Y V
D E W W Q A r r . . d e q i G I V P S K - -
G E W W K A q r . . t g q e G F I P F N F V
G D W W L A r s . . s g q t G Y I P S N Y V
G D W W D A e l . . k g r r G K V P D N Y L
- D W W E A r s l s s g h r G Y V P S N Y V
G D W W Y A r s l i t n s e G Y I P S T Y V
G E W W K A r s l a t r k e G Y I P S N Y V
G D W W L A r s l v t g r e G Y V P S N F V
G E W W K A q t . k n g q . G W V P S N Y I
S D W W R V v n l t t r q e G L I P L N F V
L P W W R A r d . k n g q e G Y I P S N Y I
R D W W E F r s k t v y t p G Y Y E S G Y V
E H W W K V k d . q l g n v G Y I P S N Y V
I H W W R V q d . r n g h e G Y V O S S Y L
K D W W K V e v . . n d r q G F V P A A Y V
V G W M P G l n e r t r q r G D F P G T Y V
P D W W E G e l . . n g q r G V F P A S Y V
E N W W N G e i . . g n r k G I F P A T Y V
E E W L E G e c . . k g k v G I F P K V F V
G G W W K G d y . g t r i q Q Y F P S N Y V
D G W W R G s y . . n g q v G W F P S N Y V
Q G W W R G e i . . y g r v G W F P A N Y V
G R W W K A r r . a n g e t G I I P S N Y V
G G W T Q G e l . k s g q k G W A P T N Y L
G D W W E A r s n . t g e n G Y I P S N Y V
N D W W T G r t . . n g k e G I F P A N Y V
```

Avaliação de Alinhamentos Múltiplos

- Questão Principal: como estimar a qualidade de um alinhamento entre sequências múltiplas?
- Assumimos que as *colunas* individuais de alinhamentos são independentes.
- Discutiremos dois métodos:
  - ⋆ Soma de Pares (SP)
  - ⋆ Entropia Mínima

Soma de Pares (SP)

- Computar a soma das pontuações entre pares:

$$Score(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

- m_i^k = caracter da sequência k e coluna i
- s = matriz de substituição

Entropia Mínima

- Ideia: **minimizar** a *entropia* de cada coluna
- Ou coluna que pode ser apresentada com menos bits de informação é melhor
- Teoria da Informação: um código óptimo usa $-\log_2 p$ para codificar uma mensagem de probabilidade p.

Entropia Mínima

- Neste caso as mensagens são os caracteres numa certa coluna
- a entropia de uma coluna é dada por:

$$Score(m_i) = -\sum_a c_{ia} \log_2(p_{ia})$$

- m_i = a coluna i de um alinhamento m
- c_{ia} = número de caracteres a na coluna i
- p_{ia} = probabilidade do caracter a na coluna i

Programação Dinâmica

- Pode-se encontrar alinhamentos óptimos usando programação dinâmica
- Generalização de métodos para alinhamento de pares:
  - ⋆ Matriz de dimensão-k para k sequências (substituindo uma matriz bidimensional)
  - ⋆ cada entrada na matriz representa um alinhamento para k subsequências (em vez de 2 subsequências)
- dadas k sequências de tamanho n
  - ⋆ Complexidade espacial é:

$$O(n^k)$$

Programação dinâmica

- Dadas k sequências de tamanho n:
  - ⋆ Complexidade temporal é:

$$\begin{cases} O(k^2 2^k n^k) & \text{se usarmos SP} \\ O(k 2^k n^k) & \text{se as pontuações de colunas puderem ser computadas em } O(k) \end{cases}$$

Métodos Heurísticos para Alinhamento

- Como a complexidade de DP é exponencial...
- *Alinhamento Progressivo*: construa uma sucessão de alinhamentos entre pares:
  - ⋆ CLUSTALW
  - ⋆ estrela
  - ⋆ ...
- Refinamento Iterativo:
  - ⋆ dado um alinhamento (eg, dado por um método progressivo)
    - ∗ remover uma sequência, realinhá-la ao perfil de outras sequências
    - ∗ repetir até convergir.

Alinhamento em Estrela

- dadas: k sequências para serem alinhadas,

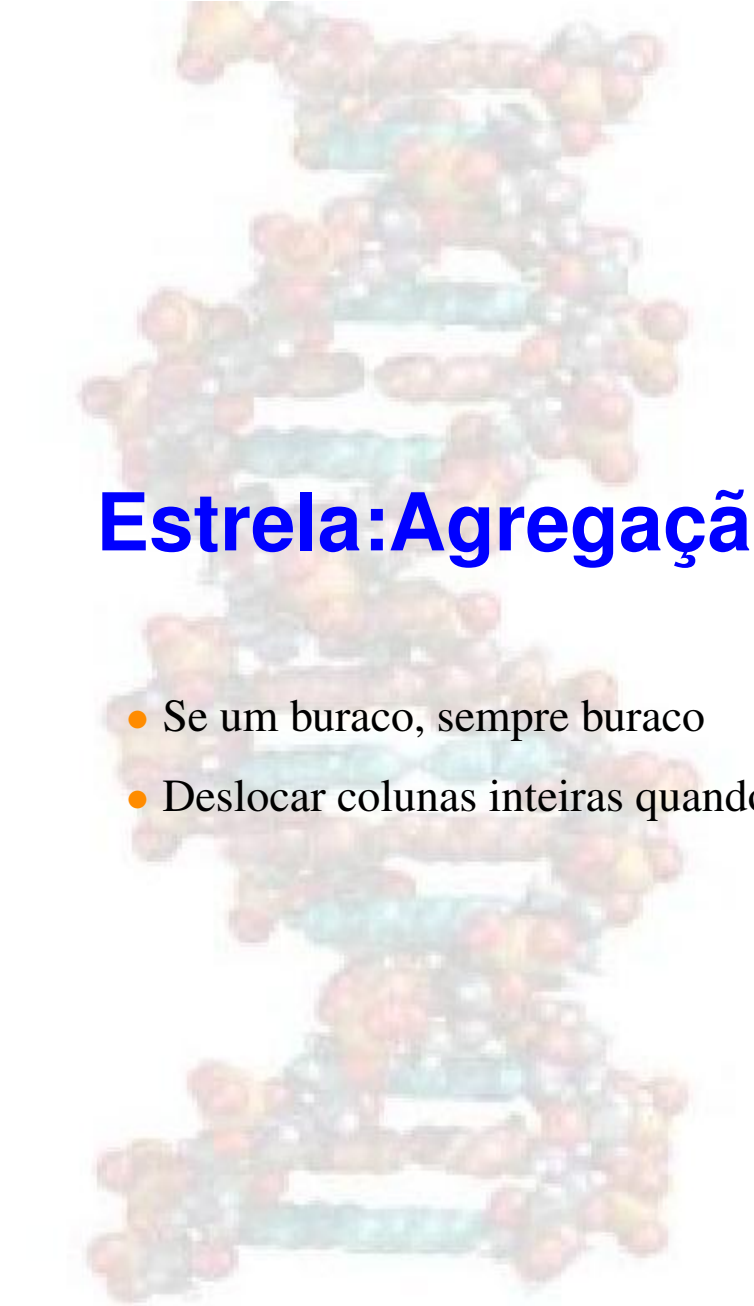
$$x_1, \ldots, x_k$$

  - ⋆ seleccione uma sequência x_c como sendo o *centro*
  - ⋆ para cada sequência x_i determine um alinhamento óptimo entre x_i e x_c
  - ⋆ agregar alinhamentos entre pares
- resultado: alinhamentos múltiplos resultando do agregado

Estrela: O Centro

- tente cada sequência como o centro, retornar o melhor alinhamento múltiplo
- computar todos os alinhamentos entre pares e seleccionar a sequência x_c que maximize:

$$\sum_{i \neq c} sim(x_i, x_c)$$



Estrela:Agregação

- Se um buraco, sempre buraco
- Deslocar colunas inteiras quando se incorporam buracos.

Estrela: Exemplo

Dados:

1. ATTGCCATT
2. ATGGCCATT
3. ATCAATTTT
4. ATCTTCTT
5. ATTGCCGATT

Estrela: Alinhamentos

1. ATTGCCATT
   ATGGCCATT
2. ATTGCCATT--
   ATC-CAATTTT
3. ATTGCCATT
   ATCTTC-TT
4. ATTGCC-ATT
   ATTGCCGATT

Estrela: Junção

1. ATTGCCATT
   ATGGCCATT

   ATTGCCATT
   ATGGCCATT

2. ATTGCCATT--
   ATC-CAATTTT

   ATTGCCATT--
   ATGGCCATT--
   ATC-CAATTTT

3. ATTGCCATT
   ATCTTC-TT

   ATTGCCATT--
   ATGGCCATT--
   ATC-CAATTTT
   ATCTTC-TT--

4. ATTGCC-ATT
   ATTGCCGATT

   ATTGCC-ATT--
   ATGGCC-ATT--
   ATC-CA-ATTTT
   ATCTTC--TT--
   ATTGCCGATT-

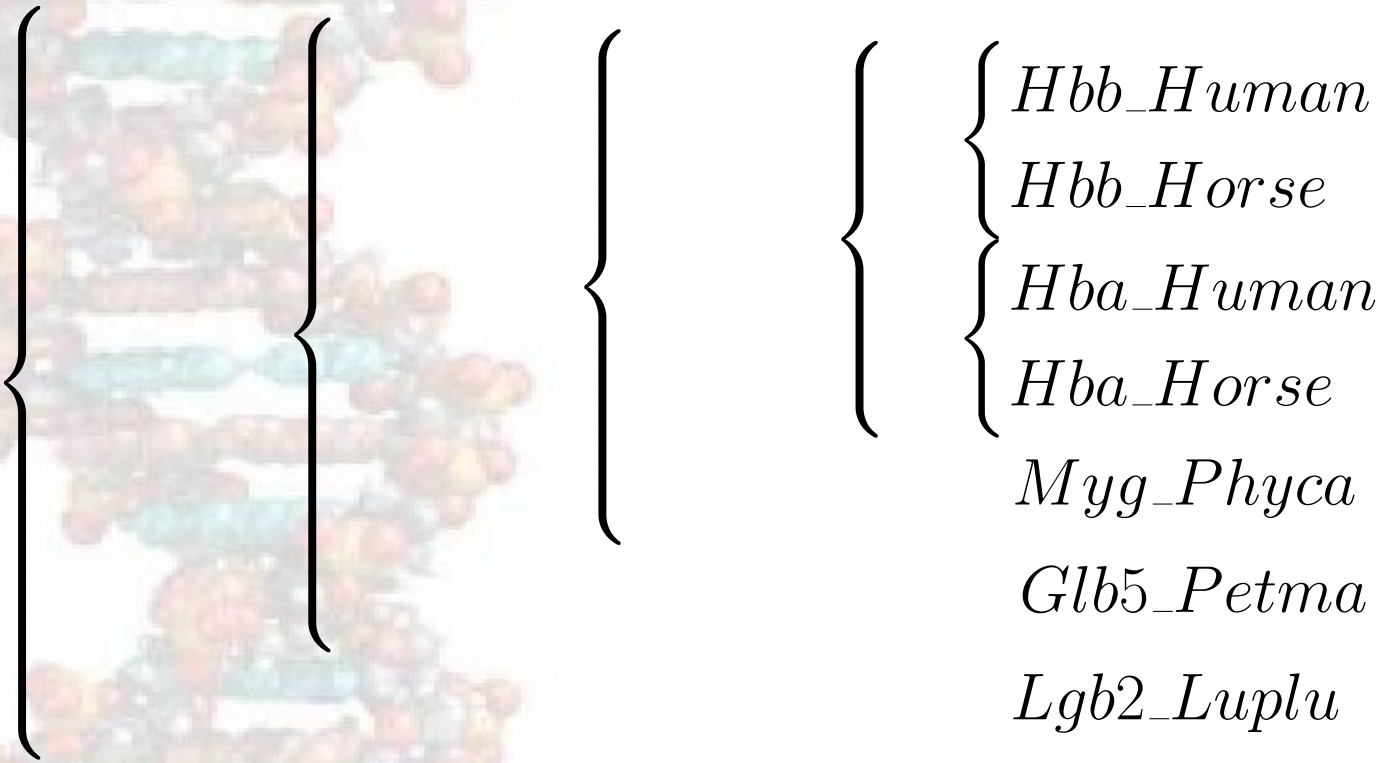
Alinhamento em Árvore

- Ideia básica: organizar alinhamentos múltiplos de sequências usando uma *árvore guia*
  - ⋆ folhas representam *sequências*
  - ⋆ nós internos representam alinhamentos
- falaremos sobre algoritmos para determinar árvores mais tarde
- determinar alinhamentos desde o fundo da árvore para cima
- variante comum: o algoritmo CLUSTALW de [Thompson et al. 1994].

Ideias de CLUSTALW

- dadas: k sequências a alinhar
  - ⋆ construir a matriz de distância de todos os pares usando DP entre os pares
  - ⋆ converter medidas de semelhança em distâncias
  - ⋆ construir uma árvore guia das distâncias
  - ⋆ alinhar os nós internos progressivamente em ordem de semelhança decrescente
- resultado: alinhamentos múltiplos na raíz da árvore

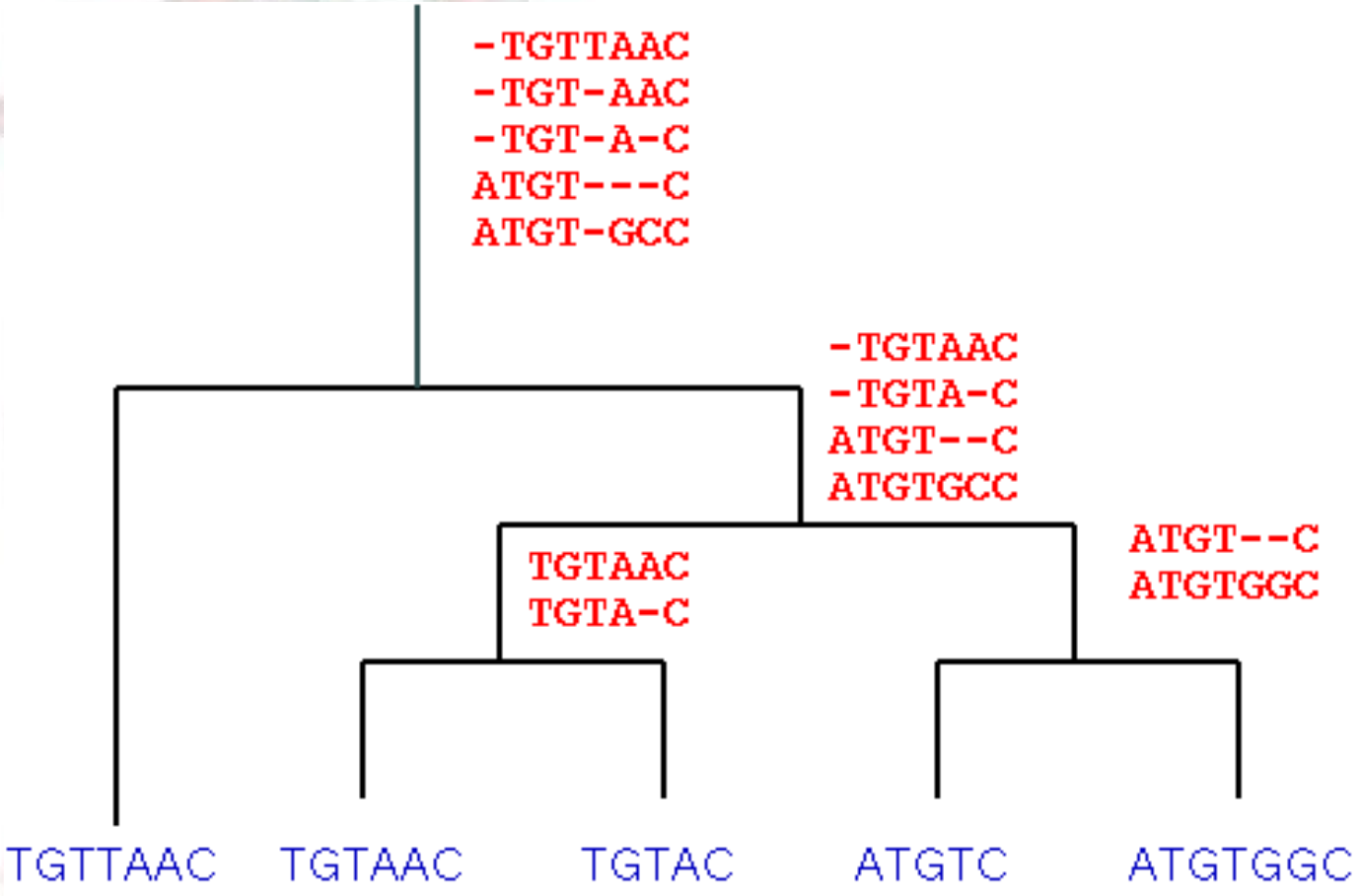
Exemplo de Árvore Guia



Alinhamento Progressivo em CLUSTALW

- dependendo do nó interno na árvore, podemos ter que alinhar:
  - ⋆ uma sequência com uma sequência
  - ⋆ uma sequência com um *perfil*
  - ⋆ um *perfil* com um *perfil*
- em todos os casos podemos usar programação dinâmica
  - ⋆ no caso de perfis, usamos Soma de Pares

SH3


-TGTTAAC
-TGT-AAC
-TGT-A-C
ATGT---C
ATGT-GCC
-TGTAAC
-TGTA-C
ATGT--C
ATGTGCC
TGTAAC
TGTA-C
ATGT--C
ATGTGGC
TGTTAAC
TGTAAC
TGTAC
ATGTC
ATGTGGC

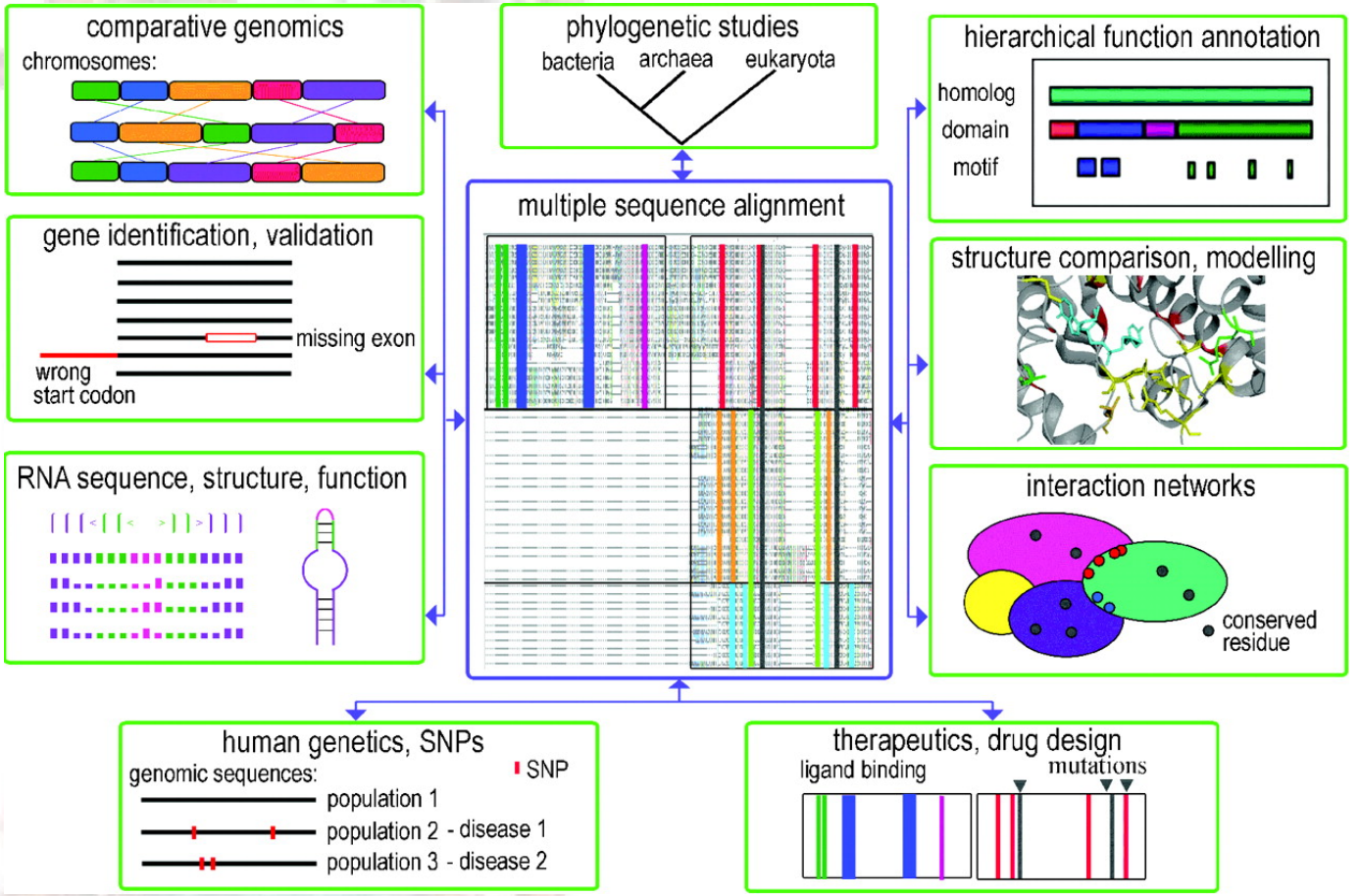
Outras Variações

- Alinhamento de perfis baseado em alinhamentos de pares
  - ⋆ considere todos os alinhamentos de pares
  - ⋆ deixar que o melhor alinhamento entre pares determine o alinhamento de sequência múltiplos
- Refinamento iterativo
  - ⋆ dado um alinhamento múltiplo
    - ∗ remover uma sequência, realinhá-la ao perfil das outras sequências
    - ∗ repetir até convergir (ou até ao computador cansar)

Métodos para Alinhamento de Múltiplas Sequências

método	**tipos de alinhamento**	**procura**
programação dinâmica multi-dimensional	global/local	programação dinâmica
Estrela	global	guloso com alinhamento de pares
CLUSTALW (árvore)	global	guloso com alinhamento de pares
HMMs com perfis	global/local	Baum-Welch (EM) para aprender modelo e Viterbi recuperar alinhamentos
EM/MEME	local	EM

Aplicações de Alinhamento de Múltiplas Sequências [Thompson et al 2005]



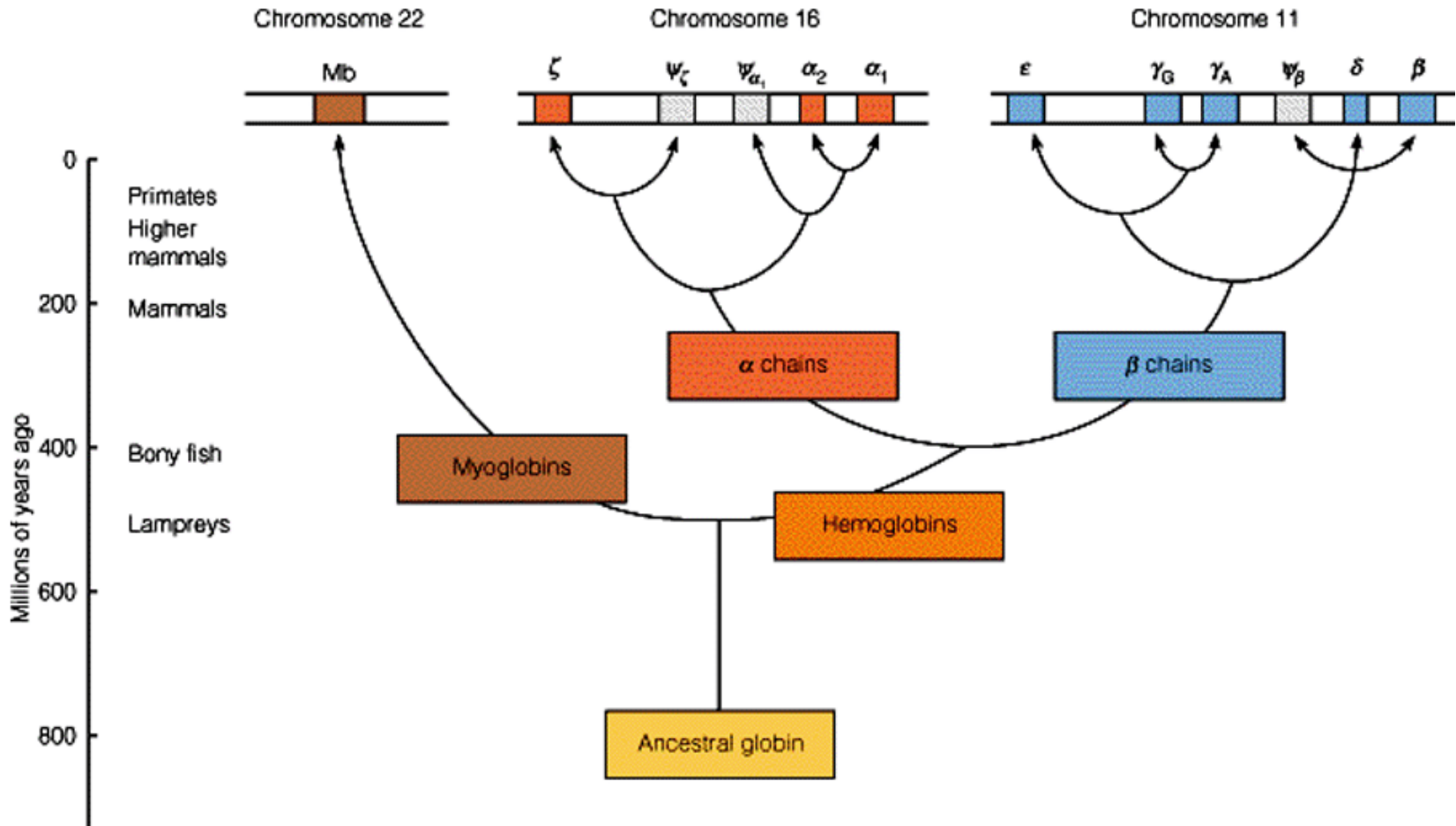
Exemplo de Sistemas

- CLUSTALW
- T-COFFEE
- ALIGN-M
- MUSCLE
- PROBCONS

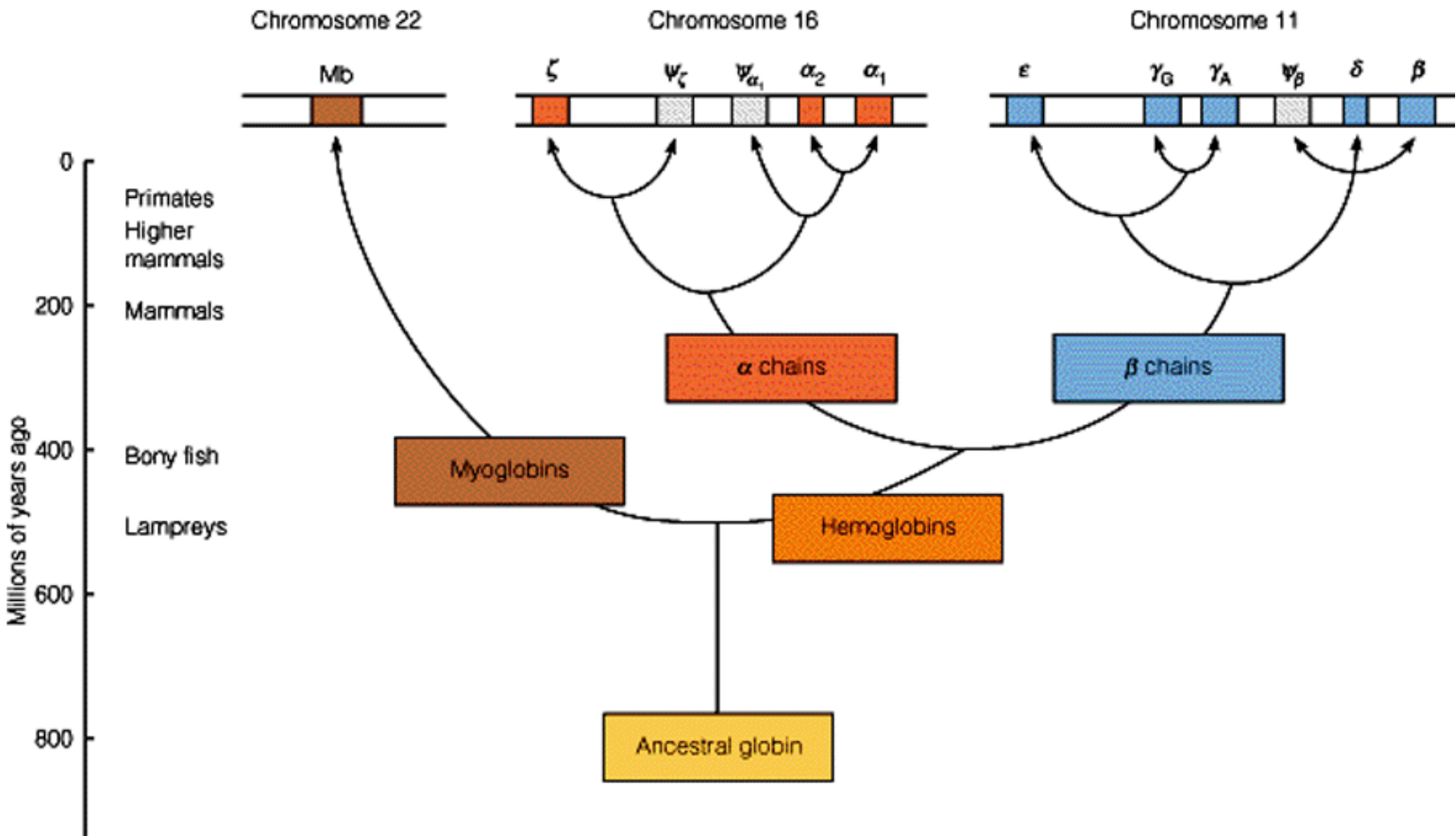
Árvores Filogenéticas

- *Árvore Filogenética*: diagrama mostrando a linha evolucionaria de espécies ou de genes
- Porquê usar árvores:
  - ⋆ para entender a ascendência de várias espécies
  - ⋆ para compreender como várias funções evoluíram
  - ⋆ para informar sobre alinhamentos múltiplos

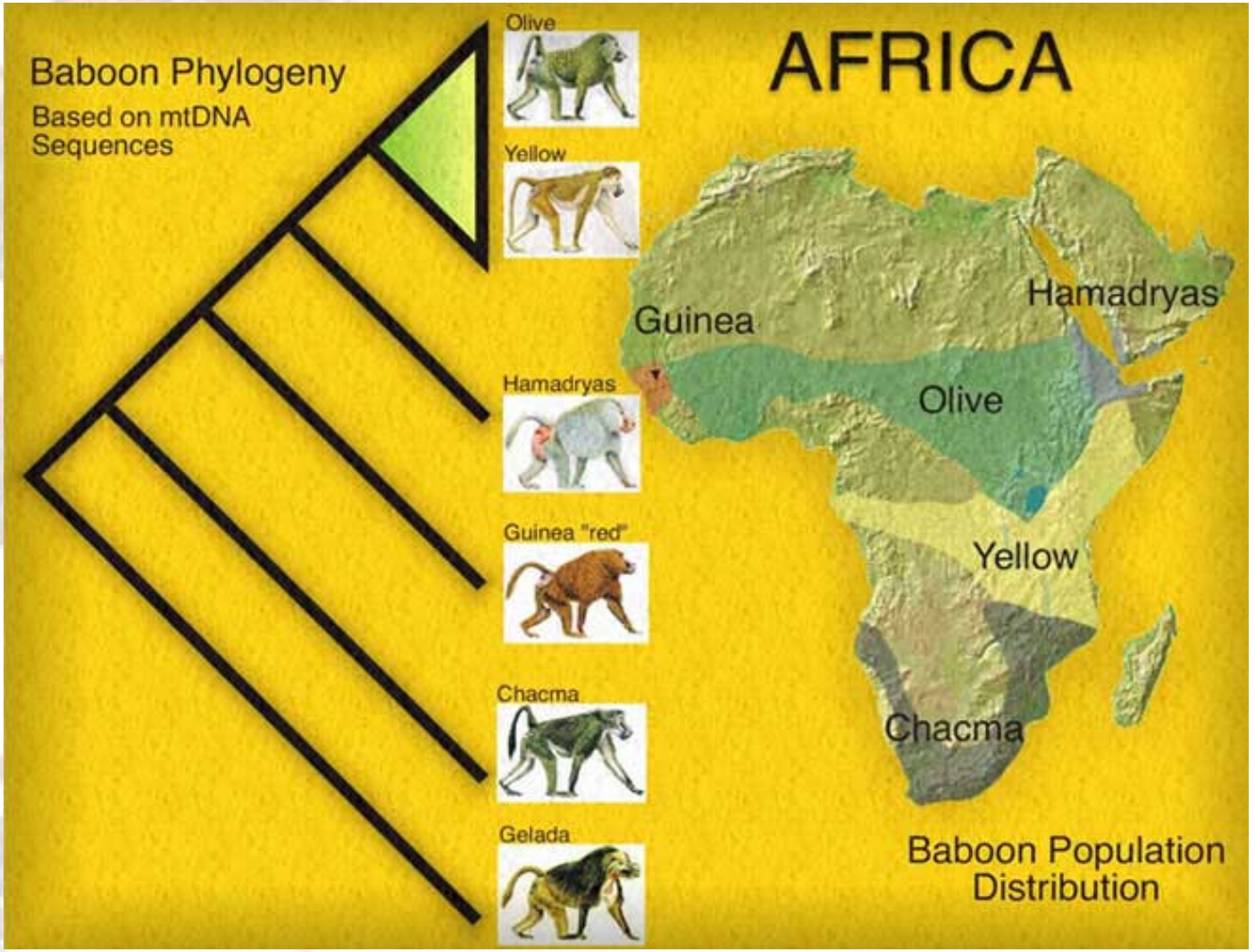
Globin evolution and expression



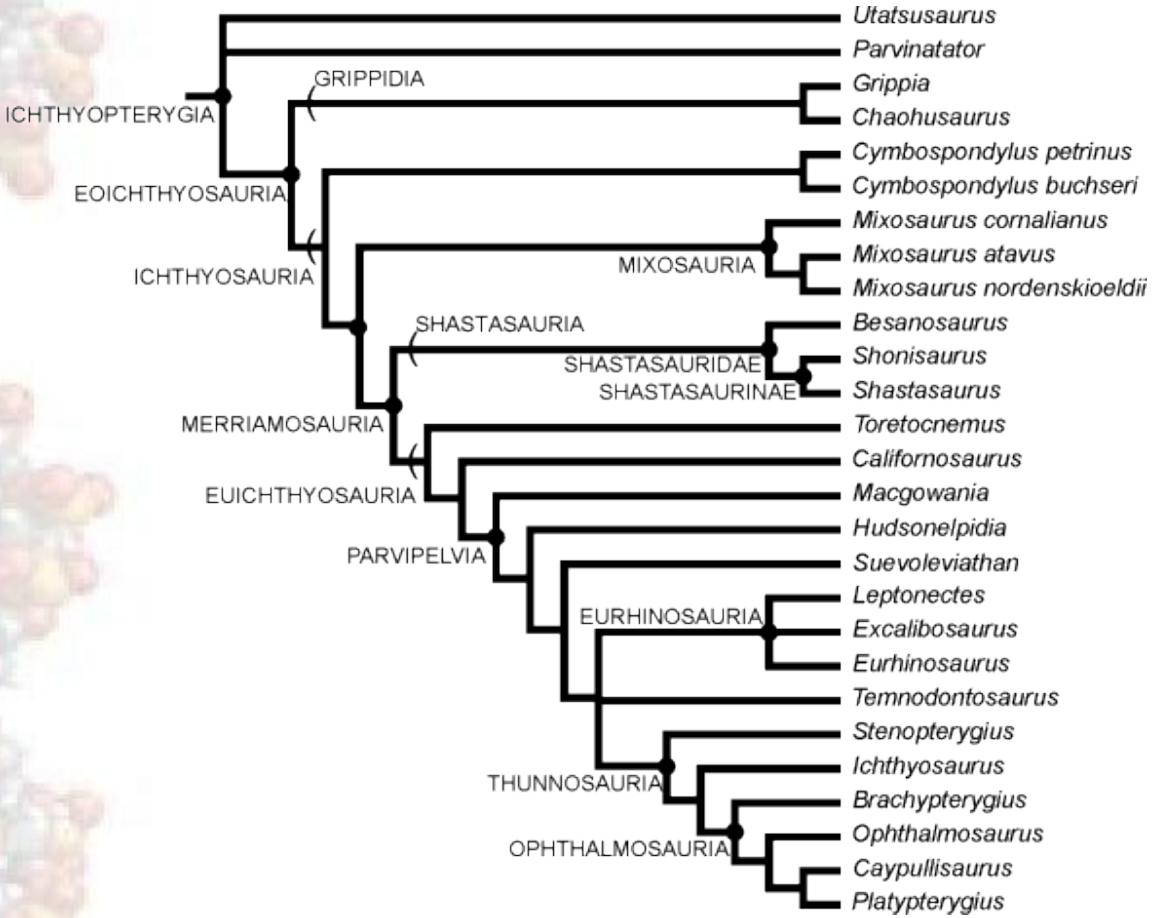
Globin evolution and expression



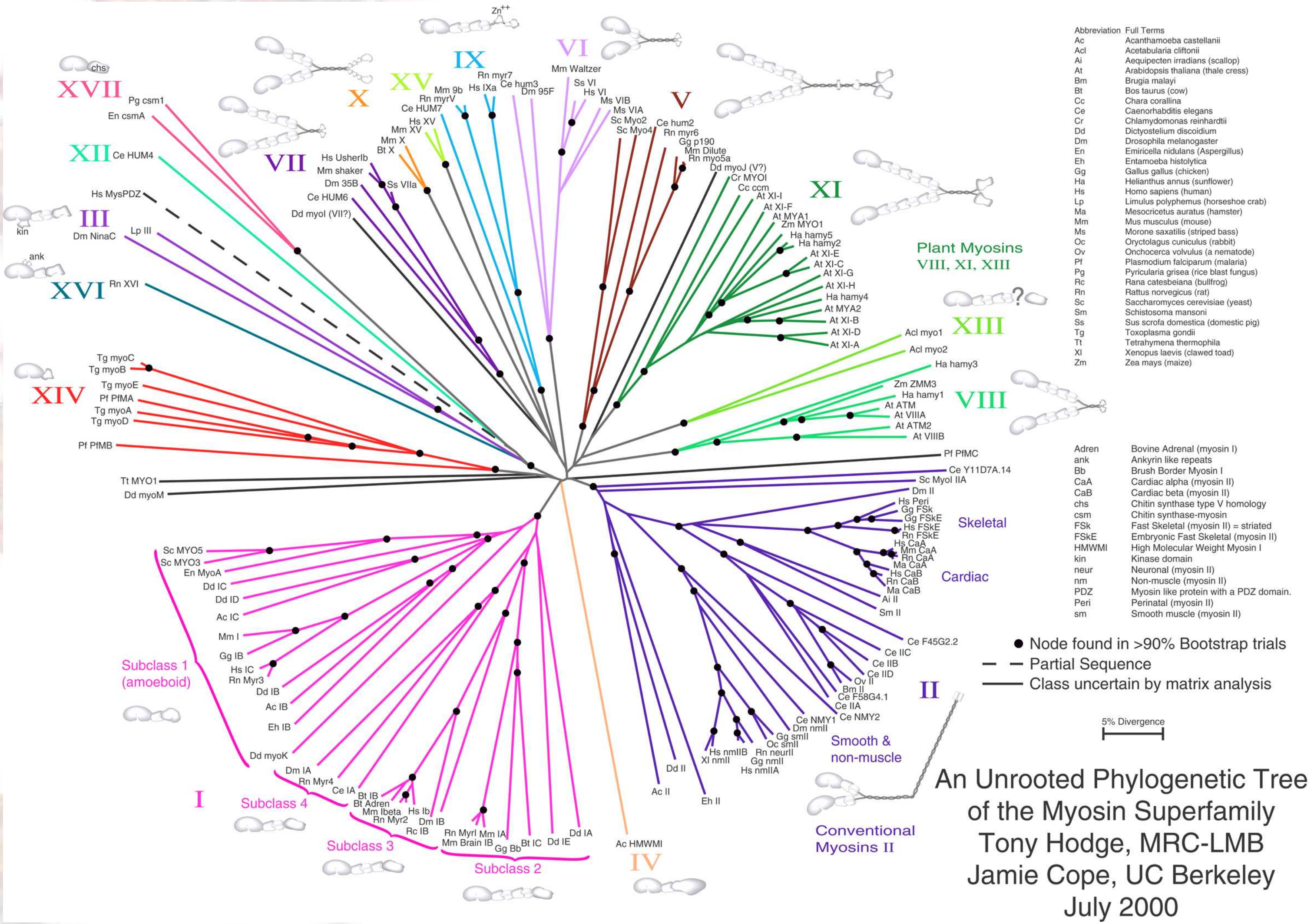
Exemplo de Filogenia: Babuínos



Exemplo de Filogenia: Myosin



Exemplo de Filogenia: Myosin



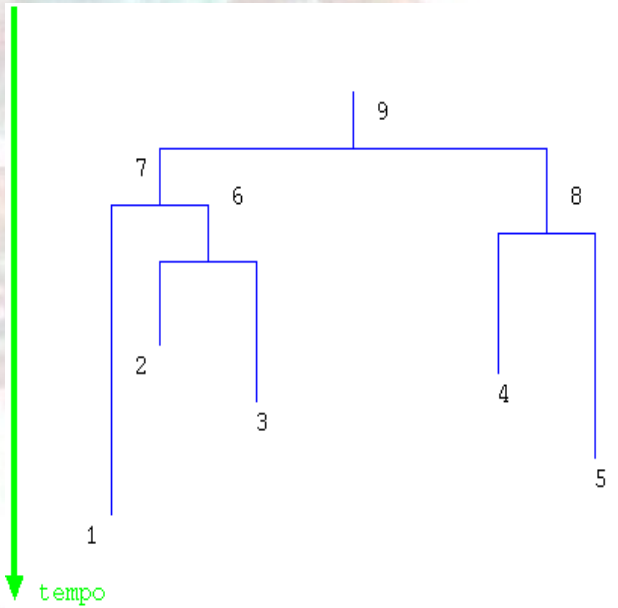
Árvores Filogenéticas: Ideias Básicas

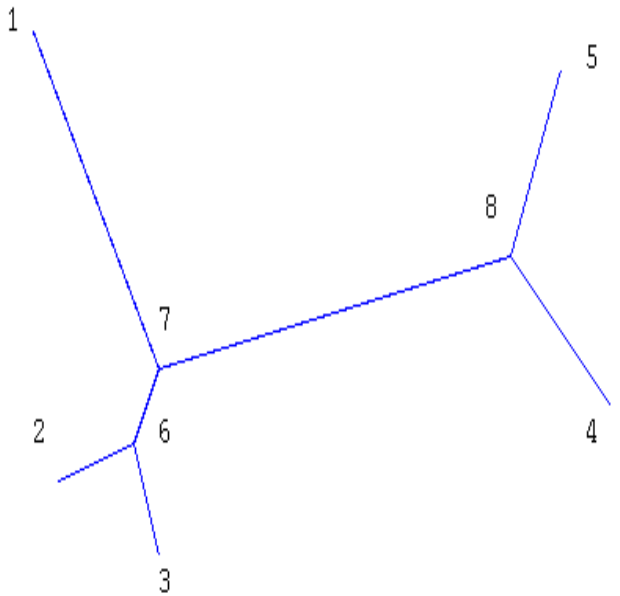
- Folhas representam coisas (genes, indivíduos/famílias, espécies) sendo comparadas
  - o termo *taxão* é usado para referir a esses elementos quando representam espécies e classificações mais amplas de organismos
  - vamos chamá-las de sequências
- nós internos são hipotéticos antepassados
- numa árvore enraizada, um caminho desde a raíz até a um nó representa um caminho evolucionário
- uma árvore não-enraizada representa relações entre coisas, mas não caminhos evolucionários

Dados para Construir Árvores

- Árvores podem ser construídas de vários tipos de dados:
  - ⋆ *baseados em distâncias*: medidas de distâncias entre espécies/genes
  - ⋆ *baseados em caracteres*: traços morfológicos (eg, pernas), sequências de DNA/proteínas
  - ⋆ *ordem de genes*: ordem linear de genes ortológicos encontrados em genomas dados

Árvores Enraizadas e Não-Enraizadas





Número de Árvores Possíveis

- dadas n sequências, existem $\prod_{i=3}^{n}(2i-5)$ árvores não-enraizadas possíveis
- e $(2n-3)\prod_{i=3}^{n}(2i-5)$ árvores enraizadas

Número de Árvores Possíveis

# sequências (n)	# árvores não-enraizadas	# árvores enraizadas
4	3	15
5	15	105
6	105	945
8	10,395	135,135
10	2,027,025	34,459,425

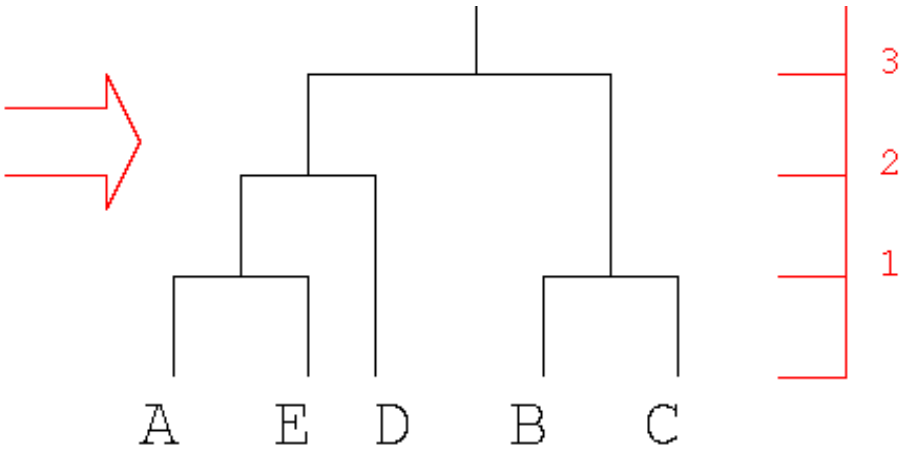
Construção de Árvores Filogenéticas

- Três tipos de métodos gerais:
  - ⋆ *distância*: encontrar uma árvore que explique as distâncias evolucionárias estimadas
  - ⋆ *parcimónia*: encontrar a árvore que requer o número mínimo de alterações para explicar os dados
  - ⋆ *máxima verosimilhança*: encontrar uma árvore que maximize a verosimilhança dos dados

Métodos Baseados em Distância

- **Dados:** uma matriz $n \times n$ M onde M_{ij} é a distância entre os objectos i e j
- **faça:** construa uma árvore pesada nas arestas tal que a distância entre as folhas i e j corresponda a M_ij

	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0



O Método UPGMA

- Unweighted Pair Group Method using Arithmetic Averages
- Ideia básica:
  - ⋆ Iterativamente tirar duas sequências/clusters e agregá-los
  - ⋆ criar novo nó na árvore para o cluster agregado
- a distância d_{ij} entre os clusters C_i e C_j de sequências é definida como:

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d_{pq}$$

ou distância média entre pares de sequências de cada cluster

Algoritmo UPGMA

- Dar a cada sequência o seu próprio cluster
- definir uma folha para cada sequência e colocar na altura 0
- enquanto há mais de 2 clusters:
  - ⋆ determinar dois clusters i, j com o menor d_{ij}
  - ⋆ defina um novo cluster $C_k = C_i \cup C_j$
  - ⋆ defina um nó k com filhos i e j, coloque-o na altura $d_{ij}/2$
  - ⋆ substitua os clusters i e j com k
- junte os últimos dois clusters, i e j, pela raíz na altura $d_{ij}/2$

UPGMA

- dado um novo cluster C_k formado pela agregação de C_i e de C_j
- podemos calcular a distância entre C_k e qualquer outro cluster C_i como segue:

$$d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$$

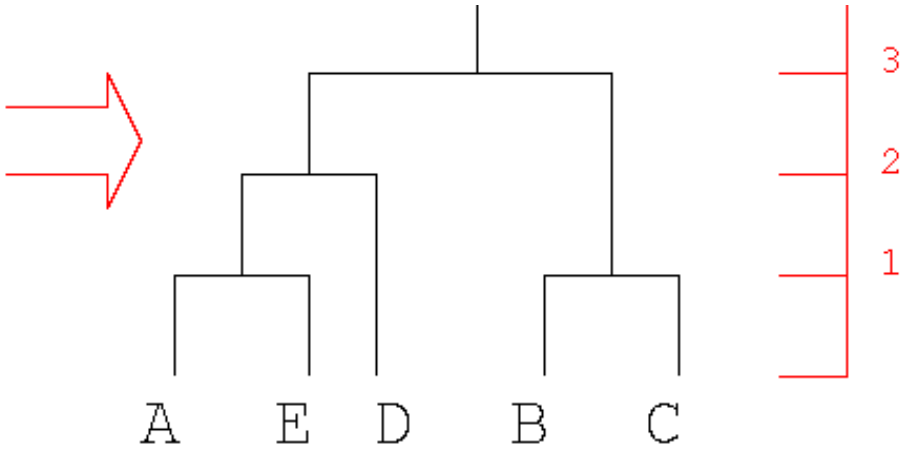
A Premissa do Relógio Molecular e Dados Ultramétricos

- A *premissa do relógio molecular*: divergência das sequências é assumida ocorrer à mesma velocidade em todos os pontos da árvore
- esta premissa não é verdade em geral: pressões evolucionárias variam de acordo com o tempo, organismos, genes num organismo e regiões num gene
- se podemos assumir esta premissa, os dados são chamados de *ultramétricos*

Dados Ultramétricos: Condição Necessária e Suficiente

- Dados Ultramétricos: para qualquer tripla de sequências i, j, k as distâncias ou são todas iguais, ou duas são iguais e a restante é menor.

	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0



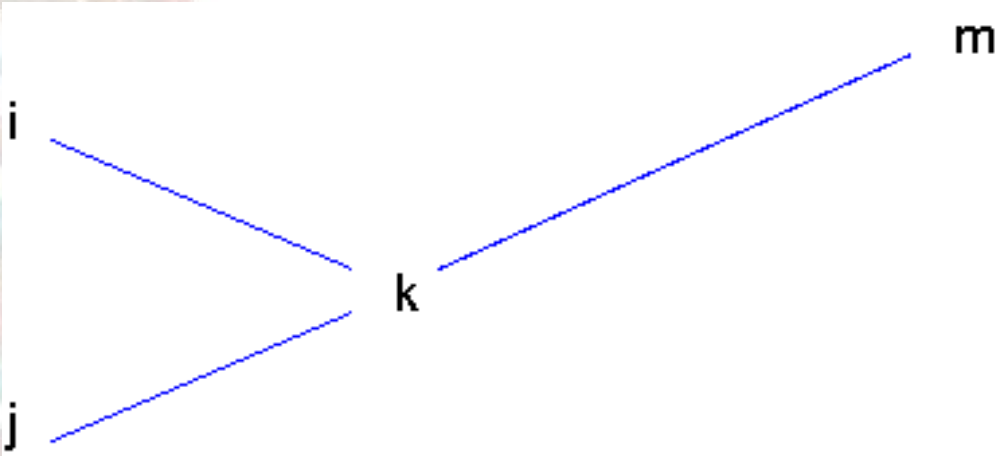
Junção de Vizinhos

- com em UPGMA, construímos uma árvore juntando iterativamente sub-árvores
- diferente de UPGMA:
  - ⋆ não assumimos o relógio molecular
  - ⋆ produz árvore não enraizada
- assuma *aditividade*: a distância entre dois pares de folhas é a soma dos comprimentos dos vértices que fazem a ligação.

Distâncias em Junção de Vizinhos

- dado um novo nó interno k, a distância para outro nó m é dada por:

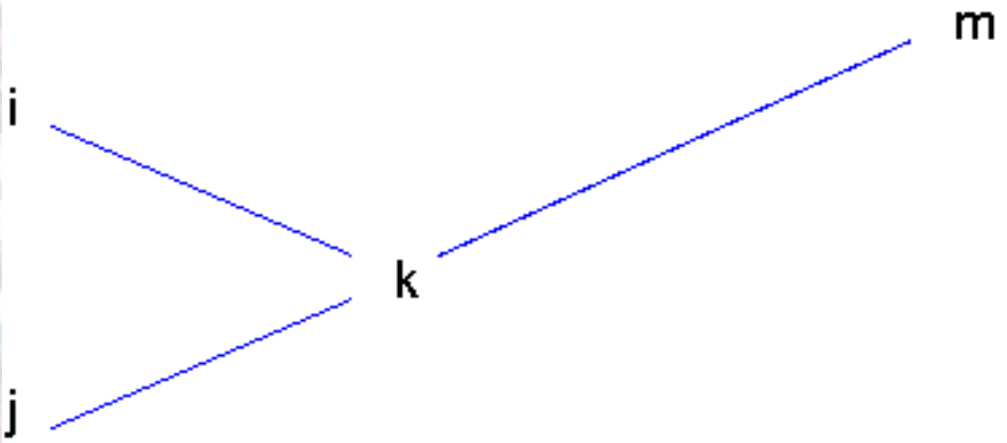
$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$$



Distâncias em Junção de Vizinhos

- Podemos calcular a distância de uma folha para o nó pai na seguinte forma:

$$d_{ik} = \frac{1}{2}(d_{ij} + d_{im} - d_{jm})$$



$$d_{jk} = d_{ij} - d_{ik}$$

Distâncias em Junção de Vizinhos

- Podemos generalizar esta regra de forma a tomar em conta a distância para todas as outras folhas:

$$d_{ik} = \frac{1}{2}(d_{ij} + r_i - r_j)$$

onde

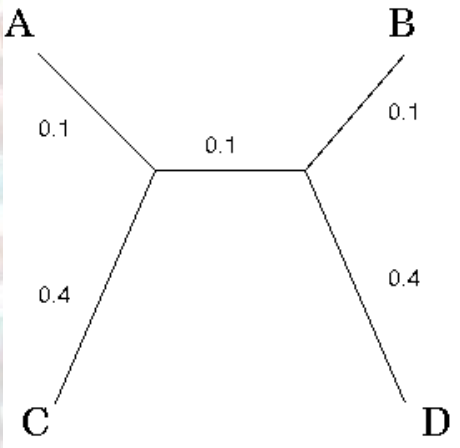
$$r_i = \frac{1}{|L| - 2} \sum_{m \in L} d_{im}$$

e L é o conjunto das folhas

- isto é mais robusto se os dados não forem estritamente aditivos

Juntar que Nós?

- Em cada passo escolhemos um par de nós para juntar. Devemos escolher os nós com o menor d_{ij}?
- Suponhamos que a árvore verdadeira parece como isto e que estamos a escolher os primeiros nós para juntar:



$$d_{AB} = 0.3$$
$$d_{AC} = 0.5$$

- Decisão errada em juntar A e B: precisamos de considerar distância do par até outras folhas.

Juntar que Nós?

- Para evitar o problema escolha o par de nós baseado nas distâncias baseado em D_{ij}:

$$D_{ij} = d_{ij} - (r_i + r_j)$$

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik}$$

Algoritmo de Junção de Vizinhos

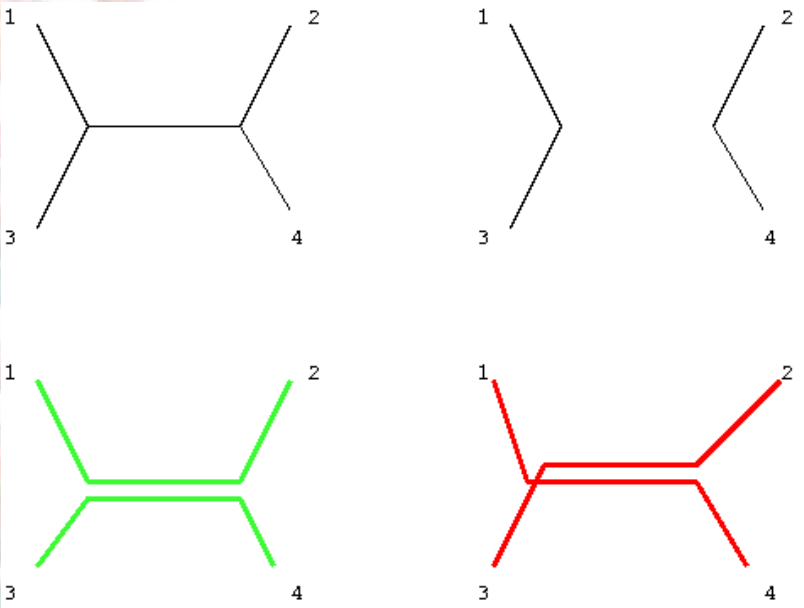
- defina a árvore T como o conjunto de nós folhas
- $L = T$
- enquanto há mais que duas sub-árvores em T:
  - ⋆ escolha o par i, j em L com D_{ij} mínimo
  - ⋆ adicione a T um novo nó agregando i e j
  - ⋆ determine novas distâncias:

$$d_{ik} = \frac{1}{2}(d_{ij} + r_i - r_j)$$
$$d_{jk} = d_{ij} - dik$$
$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij}) \text{para todos os outros} m \in L$$

  - ⋆ remova i e j de L e insira k (processe-o como se uma folha)
- junte as duas árvores restantes, i e j com um vértice de comprimento d_{ij}

Testando Aditividade

- Para qualquer conjunto de qualquer folhas i, j, k, l duas das distâncias $d_{ij} + d_{kl}$, $d_{ik} + d_{jl}$ e $d_{il} + d_{jk}$ devem ser iguais e maiores que a terceira distância



$$d_{13} + d_{24} < (d_{14} + d_{23} = d_{12} + d_{34})$$

Escolhendo Raízes

- Escolher uma raíz para árvores não-enraizadas é muitas vezes feita usando um "out-group"
- *Outgroup* é uma espécie que se sabe ser mais diferentes das outras espécies do que elas são entre elas.
- o ponto onde o outgroup se junta ao resto da árvore é o melhor candidato para a raíz.

Comentários Sobre Métodos Baseados em Distância

- Se os dados de distância são ultramétricos (e as distâncias são distâncias genuínas), então UPGMA encontra a árvore certa
- Se os dados são aditivos (e as distâncias são distâncias genuínas), então junção de vizinhos identifica a árvore correcta
- senão, os métodos podem não recuperar a árvore correcta, mas são boas heurísticas

Construção de Árvores Filogenéticas

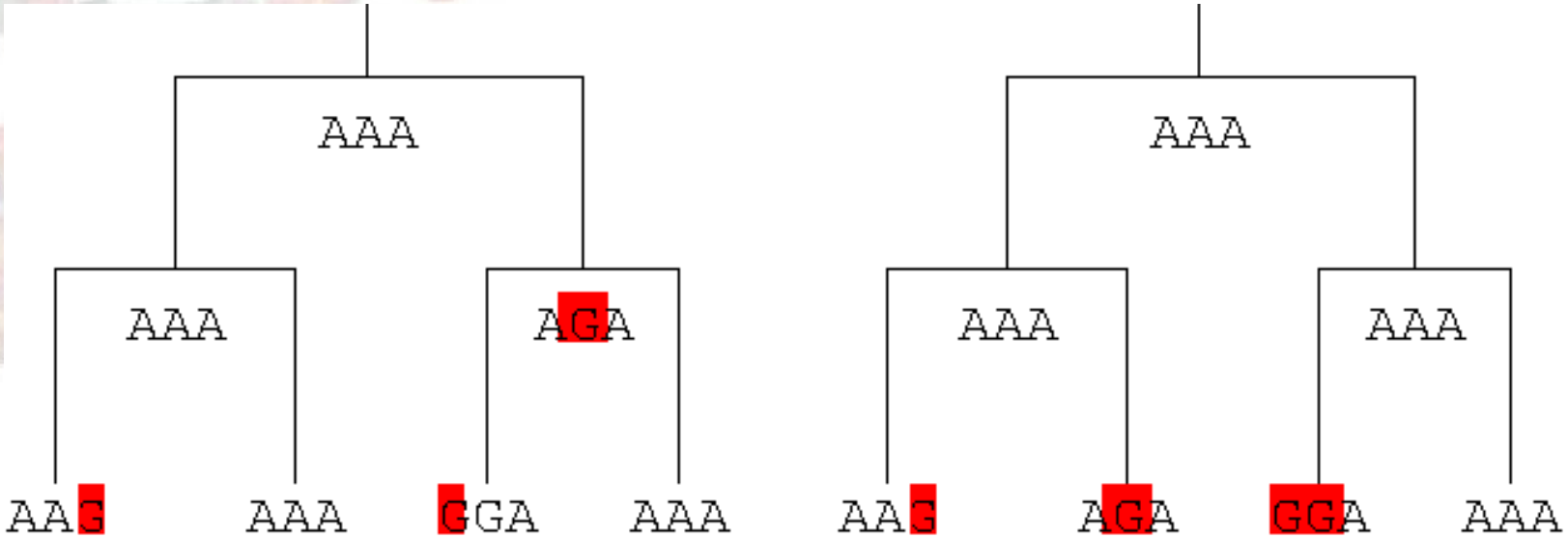
- Três tipos de métodos gerais:
  - ⋆ *distância*: encontrar uma árvore que explique as distâncias evolucionárias estimadas
  - ⋆ *parcimónia*: encontrar a árvore que requer o número mínimo de alterações para explicar os dados
  - ⋆ *maximum likelihood*: encontrar uma árvore que maximize a verosimilhança dos dados

Métodos Baseados em Parcinómia

- *dado*: dados baseados em caracteres
- *faça*: encontrar árvore que explique os dados com o número mínimo de alterações.

Exemplo de Parcinómia

- existem muitas árvores que podem explicar a filogenia das sequências seguintes: AAG, AAA GGA, AGA.



- parcimónia prefere a primeira árvore porque requer menor número de substituições

Métodos Baseados em Parcimónia

- habitualmente estes métodos envolvem dois componentes:
  - ⋆ uma procura pelo espaço das árvores
  - ⋆ um processamento para explicar o menor número de mudanças necessárias para explicar os dados (para uma dada topologia).

Encontrar Menor Número de Mudanças Numa Árvore

- Algoritmo de Fitch [1971]:
  - ⋆ assume qualquer estado (nucleotído, amino-ácido) e pode converter para qualquer outro estado
  - ⋆ assume que as posições são independentes

Algoritmo de Fitch

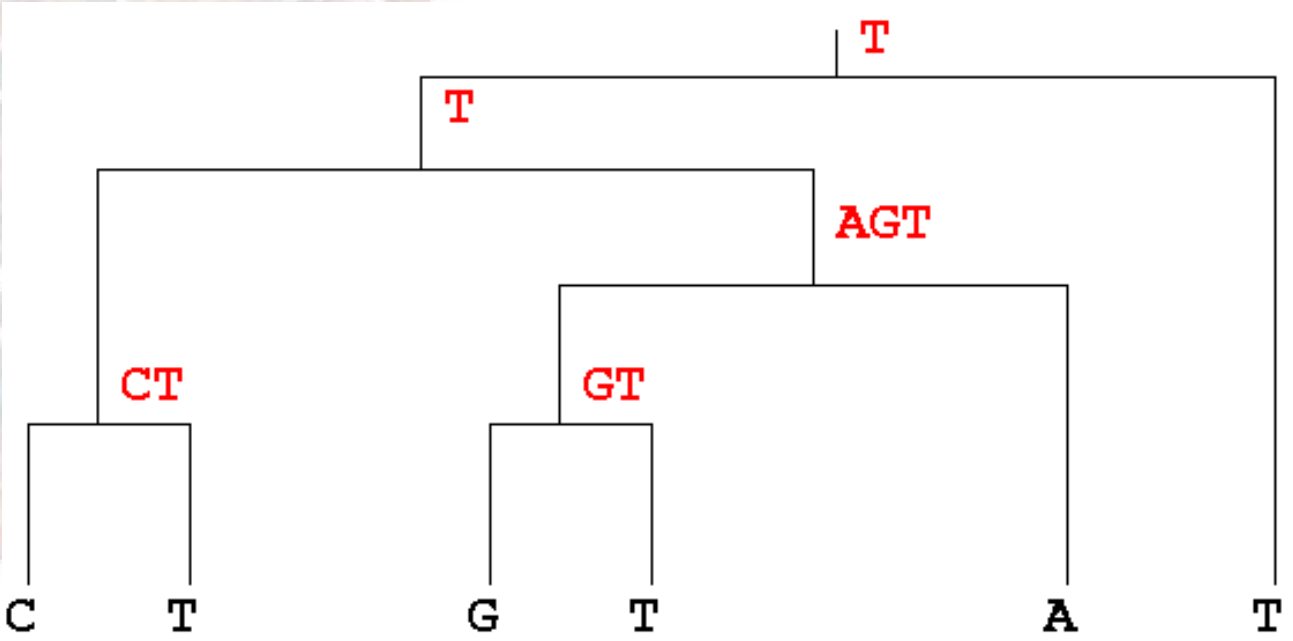
- atravessa a árvore desde as folhas até à raíz determinando o número possível de *estados* (eg, nucleotídos) que podem ser tomados por cada nó interno.
- atravessa a árvore desde a raíz até às folhas estabelecendo os estados para os nós internos.

Passo 1: Estado Possível para os Nós Internos

- atravesse a árvore em pós-ordem (desde as folhas até à raíz)
- determinar os estados possíveis R_i do nó interno i com filhos j e k:

$$R_i = \begin{cases} R_j \cup R_k, & \text{se} R_j \cap R_k = \emptyset \\ R_j \cap R_k, & \text{senão} \end{cases}$$

O Algoritmo de Fitch: Passo 1



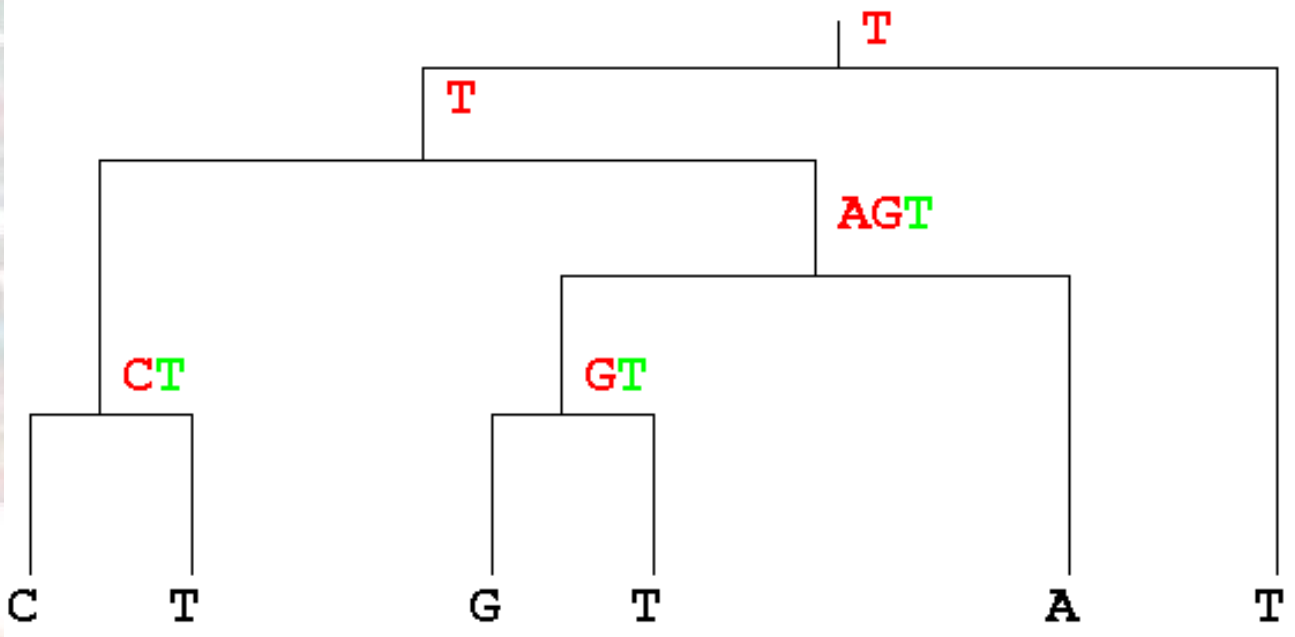
- # de mudanças = # de uniões

O Algoritmo de Fitch: Passo 2

- atravesse a árvore em pré-ordem (desde a raíz até às folhas)
- seleccionar um estado r_j do nó interno j com pai i:

$$r_j = \begin{cases} r_i, & \text{se} r_i \in R_j \\ \text{estado arbitrário} \in R_j, & \text{senão} \end{cases}$$

O Algoritmo de Fitch: Passo 2



O Algoritmo de Sankoff

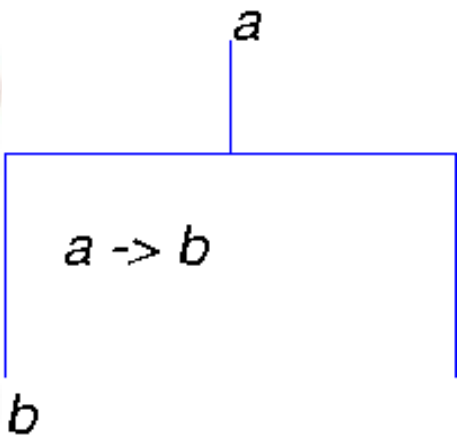
- Sankoff & Cedergren [1983]
- Em vez de assumir que todos as mudanças de estado são igualmente prováveis, use custos diferentes $S(a, b)$ para mudanças diferentes
- primeiro passo do algoritmo é propagar custos subindo na árvore:

$$a \rightarrow b$$

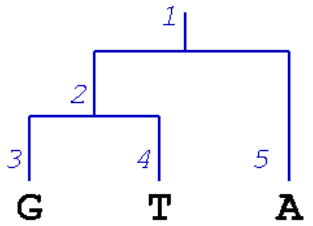
O Algoritmo de Sankoff

- para um nó interno i com filhos j e k

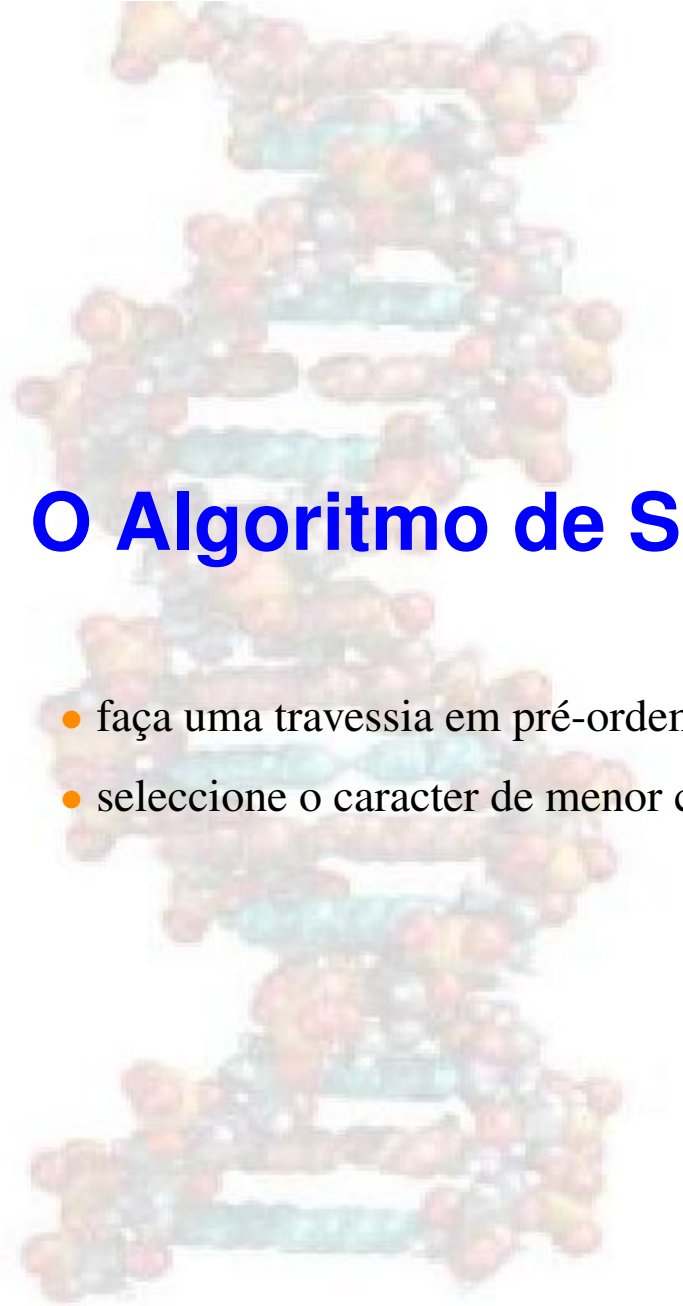
$$R_i(a) = min_b(R_j(b) + S(a,b)) + min_b(R_k(b) + S(a,b))$$



O Algoritmo de Sankoff

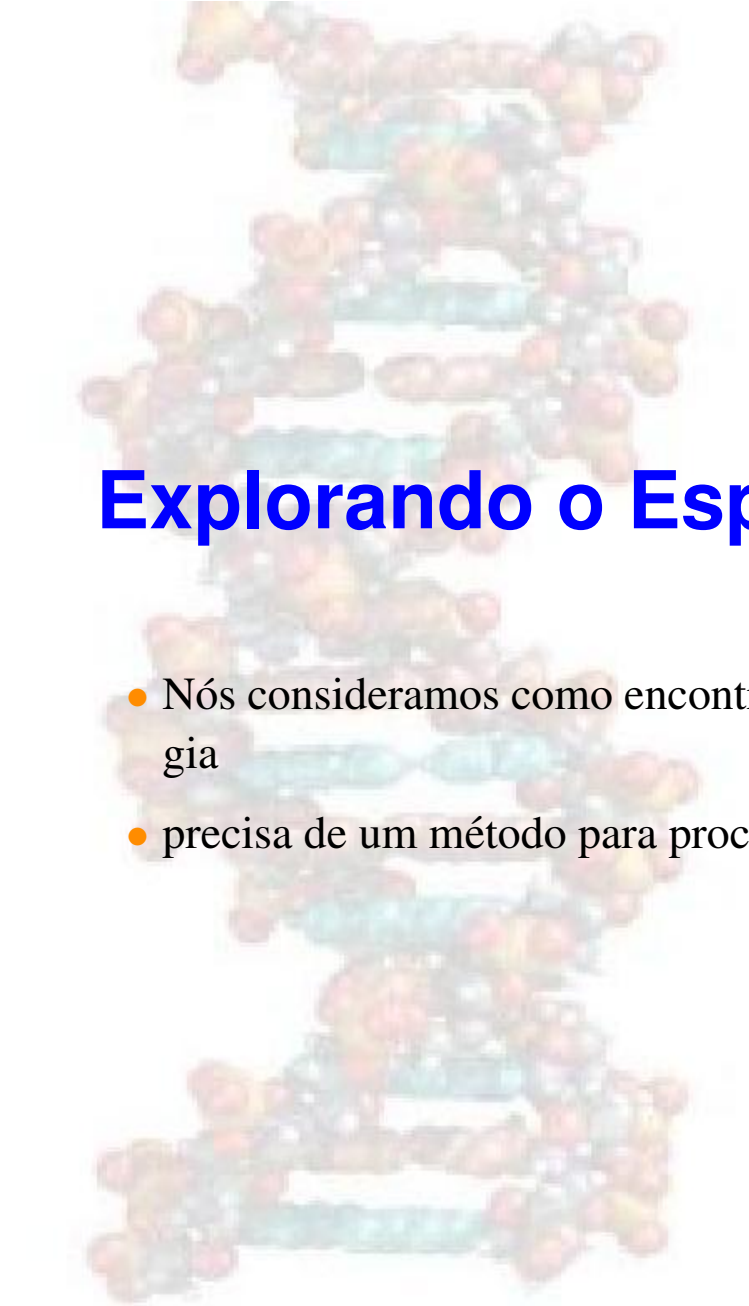


- $R_3[\mathtt{A}] = \infty, R_3[\mathtt{C}] = \infty, R_3[\mathtt{G}] = 0, R_3[\mathtt{T}] = \infty$
- $R_4[\mathtt{A}] = \infty, R_4[\mathtt{C}] = \infty, R_4[\mathtt{G}] = \infty, R_4[\mathtt{T}] = 0$
- $\begin{cases} R_2[\mathtt{A}] = R_3[\mathtt{G}] + S(\mathtt{A},\mathtt{G}) + R_4[\mathtt{T}] + S(\mathtt{A},\mathtt{T}) \\ \ldots \\ R_2[\mathtt{T}] = R_3[\mathtt{G}] + S(\mathtt{A},\mathtt{T}) + R_4[\mathtt{T}] + S(\mathtt{T},\mathtt{T}) \end{cases}$
- $R_5[\mathtt{A}] = 0, R_5[\mathtt{C}] = \infty, R_5[\mathtt{G}] = \infty, R_5[\mathtt{T}] = \infty$
- $\begin{cases} R_1[\mathtt{A}] = min(R_2[\mathtt{A}] + S(\mathtt{A},\mathtt{A}), \ldots, R_2[\mathtt{T}] + S(\mathtt{A},\mathtt{T})) + R_5[\mathtt{A}] + S(\mathtt{A},\mathtt{A}) \\ \ldots \\ R_1[\mathtt{T}] = min(R_2[\mathtt{A}] + S(\mathtt{T},\mathtt{A}), \ldots, R_2[\mathtt{T}] + S(\mathtt{T},\mathtt{T})) + R_5[\mathtt{A}] + S(\mathtt{A},\mathtt{T}) \end{cases}$



O Algoritmo de Sankoff: Passo 2

- faça uma travessia em pré-ordem da árvore (desde a raíz para as folhas)
- seleccione o caracter de menor custo para cada nó



Explorando o Espaço das Árvores

- Nós consideramos como encontrar o menor número de mudanças para cada topologia
- precisa de um método para procurar no espaço das árvores

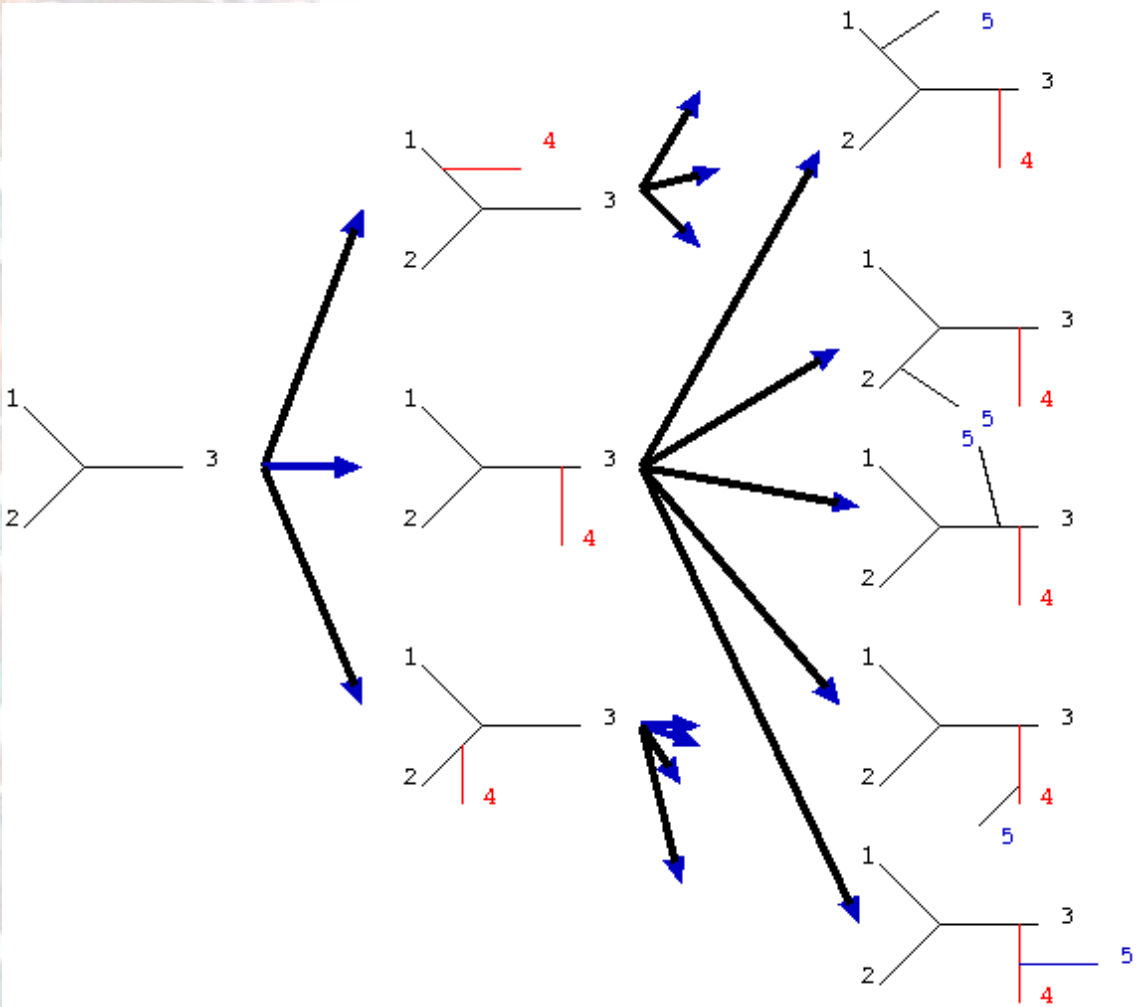
Métodos de Procura

- exaustiva
- branch & bound:
  - ⋆ encontre um árvore inicial (eg, por UPGMA ou por junção de vizinhos) e determine o custo
  - ⋆ use procura para encontrar outras árvores:
    - ∗ abandone árvores parciais cujo custo excede a árvore de menor custo até agora
- métodos gulosos: eg, troca de ramos

Procura Por Branch & Bound

- procure pelo espaço das árvores sem raíz:
  - ⋆ adicione folhas à árvore incrementalmente
  - ⋆ mantenha a árvore de custo menor completa até agora T'
  - ⋆ corte uma árvore T e os seus descendentes se $custo(T) > custo(T')$
- Propriedade Chave: adicionar folhas só pode aumentar o custo da árvore

Procura Por Branch & Bound



Algoritmo

- Para n sequências mantenha um vector de contadores:

$$[i_3][i_5][i_7]\ldots[i_{2n-5}]$$

  onde i_k toma os valores $0 \ldots k$

- uma árvore completa é representada por uma atribuição de todos os i_k a valores não-zero.
- i_k indica, com uma árvore parcial com k vértices, onde adicionar um ramo para a sequência seguinte
- $i_k = 0$ indica uma árvore parcial

Algoritmo

- Para procurar o espaço, rode contadores através dos seus valores possíveis (como se fossem odómetros):
  - ⋆ contadores mais à direita mudam mais depressa
  - ⋆ quando um contador é zero, os contadores à direita devem ser 0 também
  - ⋆ teste o custo (parcial) da árvore em cada tick
  - ⋆ faça com que o odómetro salte quando há um corte

Algoritmo

- É um método completo
  - ⋆ garantido encontrar solução óptima
- frequentemente muito mais eficiente que procura exaustiva
- no pior caso, não é melhor
- a eficiência depende da qualidade da árvore inicial

Comentários sobre Inferência de Árvores

- o espaço de procura pode ser grande, mas pode encontrar a árvore óptima eficientemente em alguns casos
- em alguns casos métodos heurísticos podem ser aplicados
- difícil avaliar filogenias inferidas: a verdade-alvo não é habitualmente sabida:
  - ⋆ podemos olhar para a concordância entre diferentes fontes de evidência
  - ⋆ quando a procura não é completa, podemos procurar repetibilidade em subamostras dos dados
- alguns métodos novos usam dados baseados na ordem linear dos genes ortológicos no cromossoma
- filogenia de bactérias e vírus não é trivial devido a transferências laterais de material genético: *filogenias locais* podem ser mais apropriadas

Comentários sobre Inferência de Árvores

Um visão diferente:

- Aula de Felsenstein
- Aula de Shamir
- Phylip
- PAUP*
- Molphy
- PAML
- MrBayes

f

Diversidade de Métodos

- Gene Arrays: medem quantidades de RNA
- Electroforese: mede quantidade de proteínas

Gene Arrays

Uma animação

Experimentos com Clustering Hierárquico

- Eisen e outros, PNAS 1988
- Primeira aplicação de clustering para dados de expressão de genes
- S. cerevisae (fermento de padeiro)
  - ⋆ todos os genes (6200) numa única matriz
  - ⋆ medido durante vários processos
- fibroblastos humanos:
  - ⋆ 8600 transcriptos humanos numa matriz
  - ⋆ medida em 12 pontos temporais durante estimulação com soro

Os Dados

- 79 medidas para dados do fermento
- colecionado em vários pontos temporais durante:
  - ⋆ “mudança diáuxica”: desligar genes que metabolizam açúcares, e activar aqueles que metabolizam o etanol
  - ⋆ ciclo de divisão mitótico
  - ⋆ esporulação
  - ⋆ choque térmico
  - ⋆ choque de redução

Os Dados

- cada medida G_i representa

$$\log \frac{\mathbf{vermelho_i}}{\mathbf{verde_i}}$$

onde **vermelho** é o nível de expressão no teste, e **verde** é o nível de referência para o gene G no experimento i

- o perfil de expressão de um gene é o vector de medição por todos os experimentos:

$$< G_1 \ldots G_n >$$

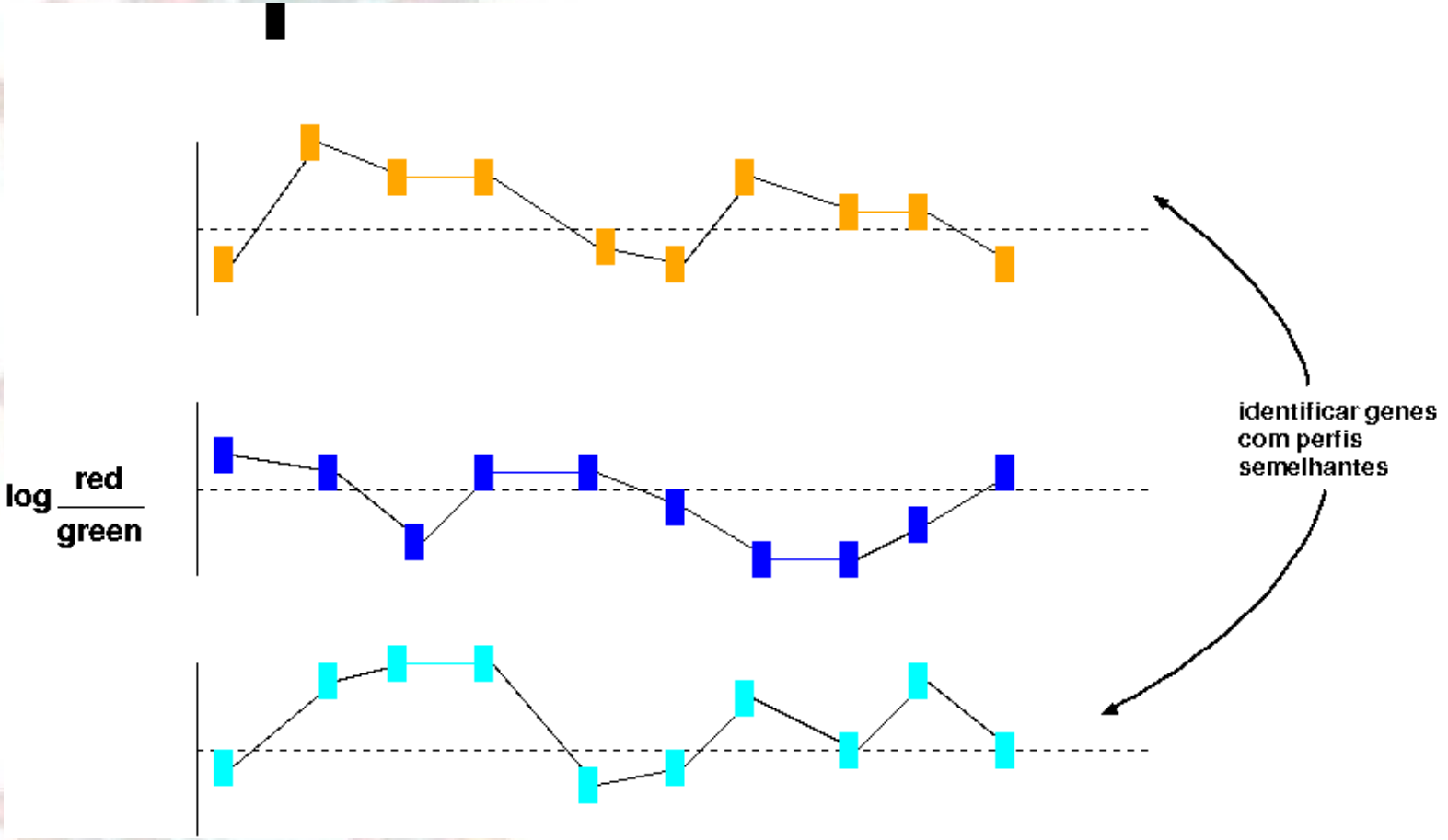
Os Dados

- m genes medidos em n experimentos

$g_{1,1}$	•	•	•	$g_{1,n}$
$g_{2,1}$	•	•	•	$g_{2,n}$
$g_{m,1}$	•	•	•	$g_{m,n}$

- cada linha é o perfil de um gene

A Tarefa



Métrica de Semelhança de Genes: Um Coeficiente de Correlação

- para determinar a semelhança entre dois genes X e Y

$$S(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - X_{\text{offset}}}{\Phi_x}\right)\left(\frac{Y_i - Y_{\text{offset}}}{\Phi_y}\right)$$

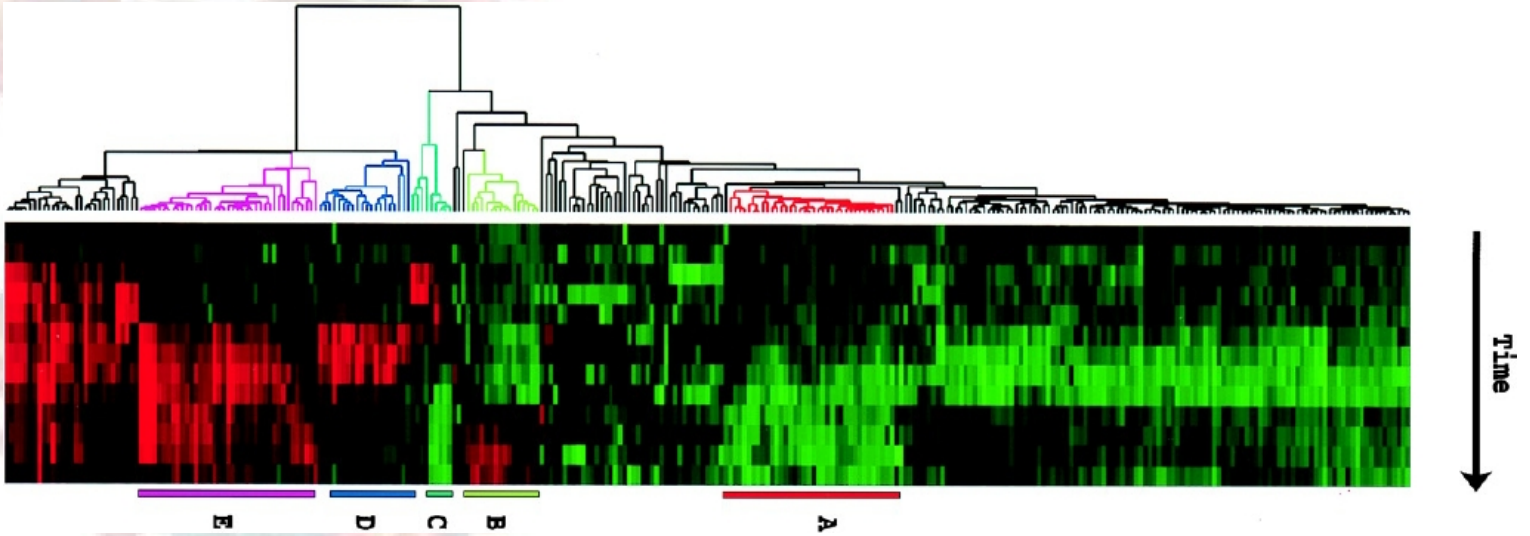
$$\Phi_G = \sqrt{\sum_{i=1}^{n}\frac{(G_i - G_{\text{offset}})^2}{n}}$$

Métrica de Semelhança de Genes

- Como há um estado assumido de referência (o nível de expressão do gene), G_{offset} é inicializado como 0 para todos os genes.

$$S(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i}{\sqrt{\sum_{i=1}^{n}\frac{X_i^2}{n}}}\right)\left(\frac{Y_i}{\sqrt{\sum_{i=1}^{n}\frac{Y_i^2}{n}}}\right)$$

Dendograma para Estimulação por Soro dos Fibroblastos



- a azul, sinalização e angiogenes;
- a verde ciclo celular,
- a vermelho colestorol e biosíntese.

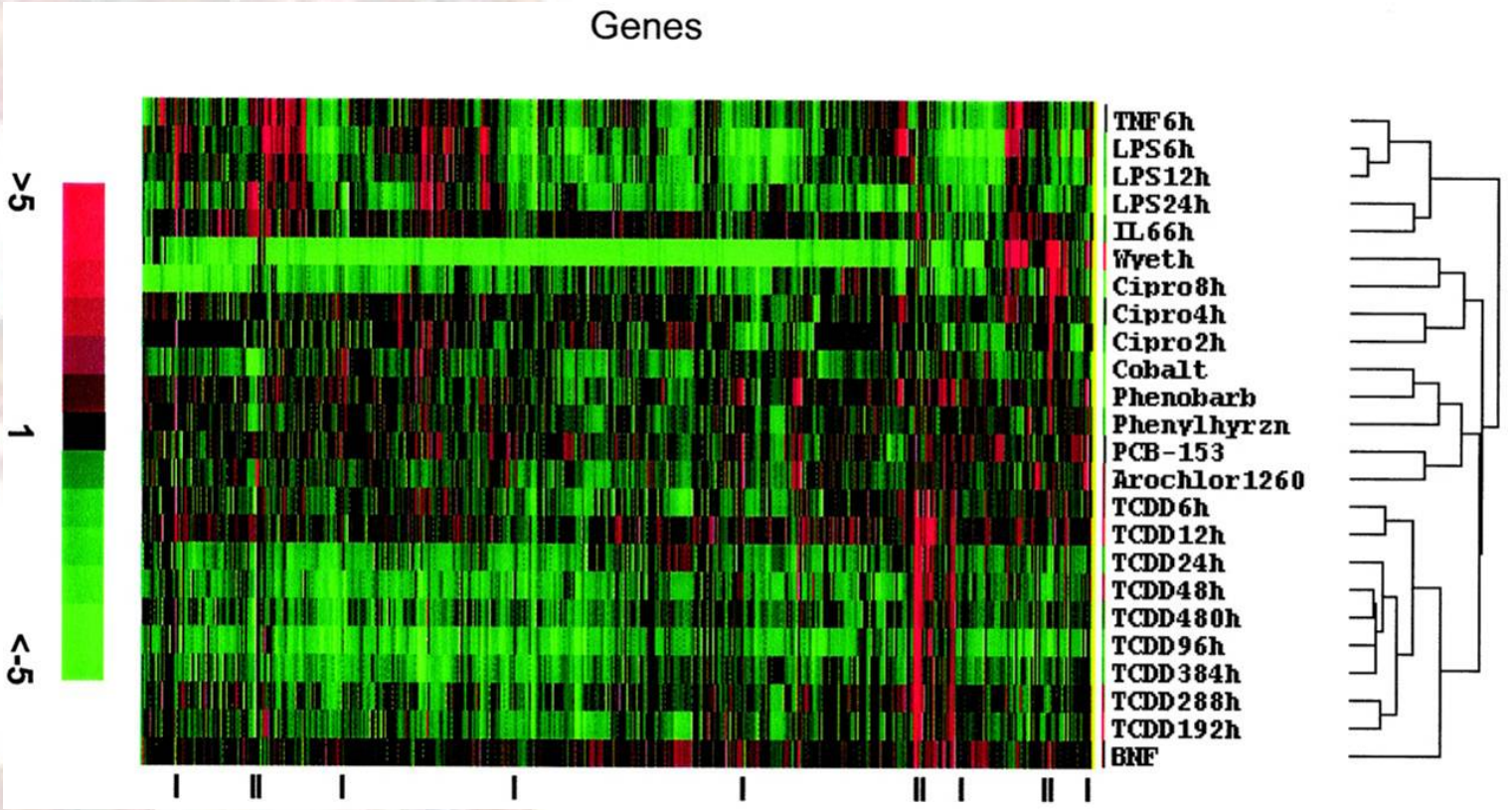
Como Ordenar Folhas

- Para qualquer dendograma de n instâncias existem 2^{n-1} permutações das folhas que são consistentes com a árvore
- Como escolher uma ordenação que seja informativo?
- Método heuristico usado por Eisen e colegas:
  - ⋆ pese cada gene usando alguma variável de interesse (e.g. nível de expressão médio ou tempo de expressão máxima)
  - ⋆ para cada nó interno, coloque o que tem menor peso é médio primeiro na ordeção
- pode ser feito óptimamente em tempo $O(n^4)$
  - ⋆ Bar-Joseph e outros, Bioinformatics 2001
  - ⋆ Noção de solução óptima: maximizar a soma de semelhança de folhas adjacentes

Resultados de Eisen

- representações redundantes de genes são agrupados em conjunto
  - ⋆ mas genes individuais podem ser distinguidos de genes relacionados por diferenças subtis na expressão
- genes com funcionalidade semelhante agrupam-se em conjunto; e.g., 126 genes regulam fortemente para desligar em resposta ao stress
  - ⋆ 112 desses genes codificam proteínas ribossomais e outras proteínas envolvidas em tradução
  - ⋆ concorda com resultados conhecidos que fermento responde a condições de crescimento favoráveis aumentando a produção dos ribossomas.
- Não existem resultados biológicos novos no artigo:
  - ⋆ mas o método confirma o que esperariam ver
  - ⋆ indica potencial para encontrar nova biologia

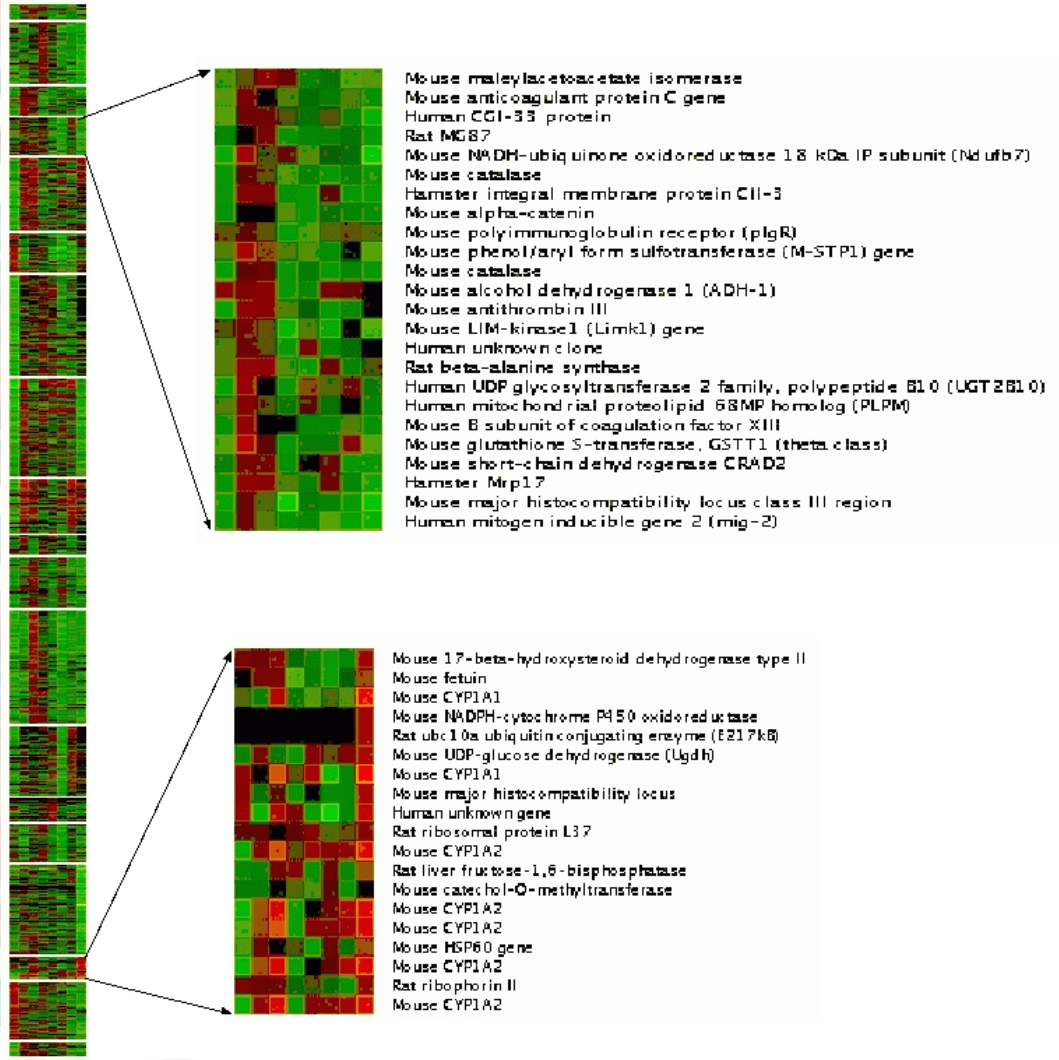
Outra Aplicação de Clustering Hierárquico



Clustering Particional

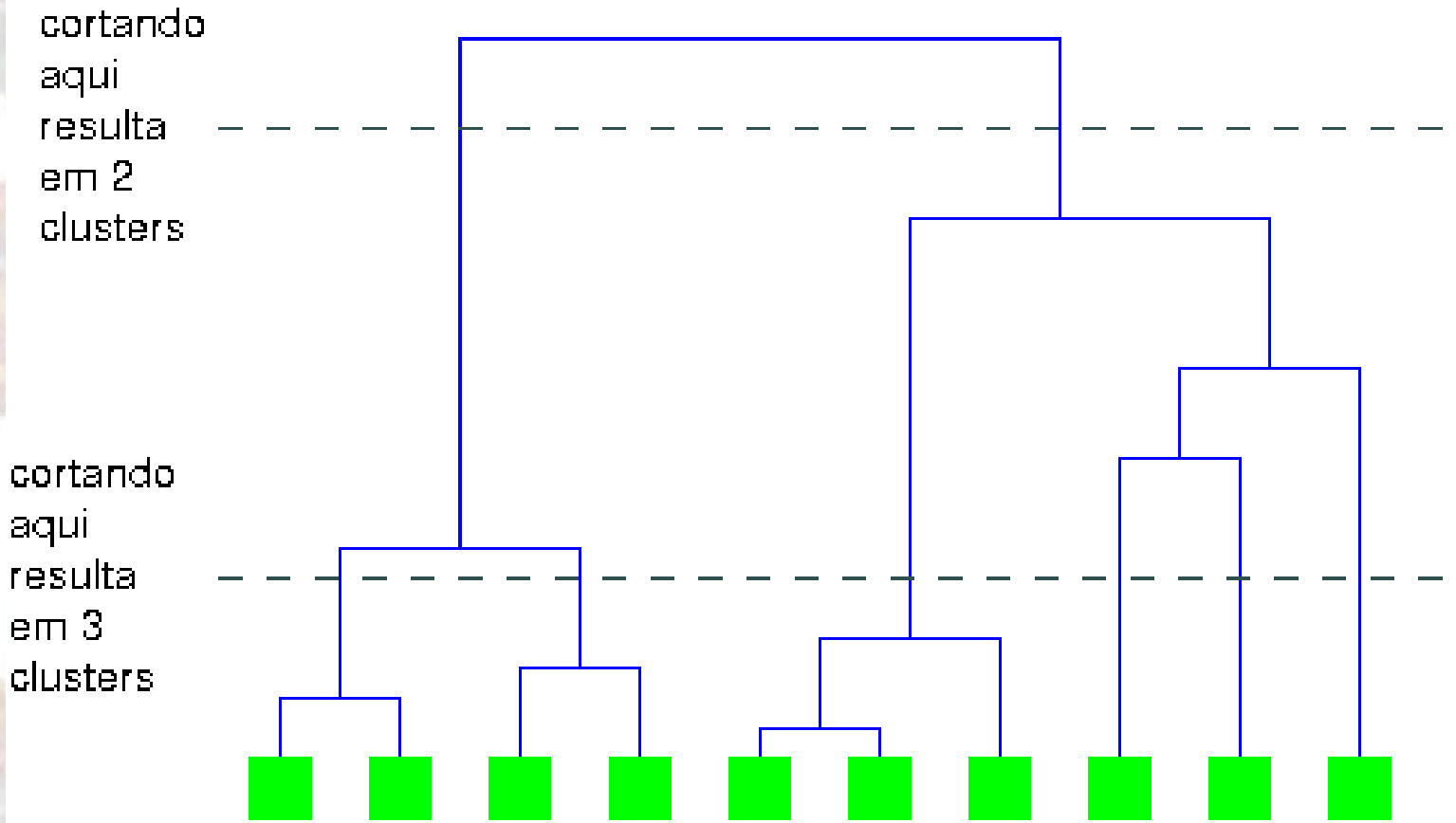
- dividir instâncis em claustros disjuntos
  - ⋆ flat ou árvore
- temas principais:
  - ⋆ quantos clusters?
  - ⋆ como representar clusters?

Exemplo de Clustering Particional



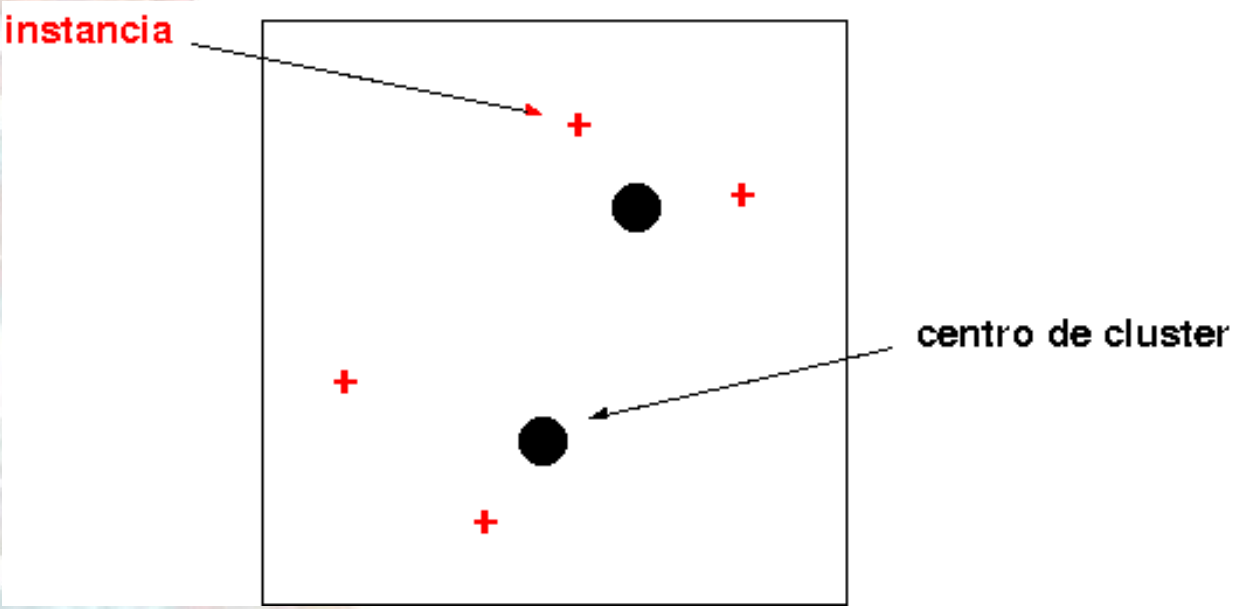
ing Hierárquico

- podemos sempre gerar clustering particional a partir de hierárquico cortando a árvore num certo nível



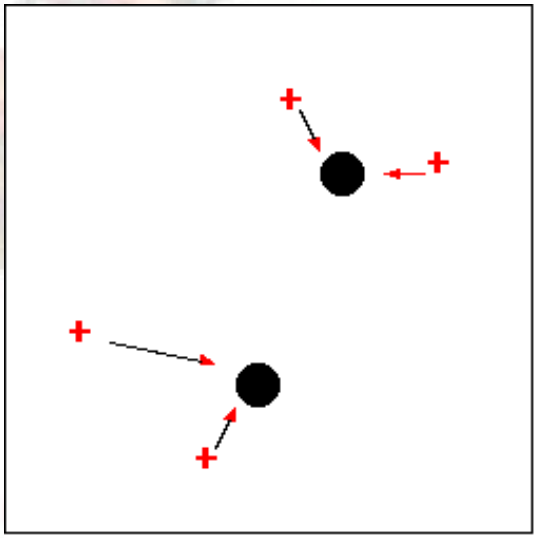
Clustering por K-Médias

- Assume que cada instância é representada por vectores de valores reais
- ponha k centros de claustros no mesmo espaço do que as instâncias
- cada cluster é representado por um vector
- considere um exemplo em que cada claustro tem duas dimensões

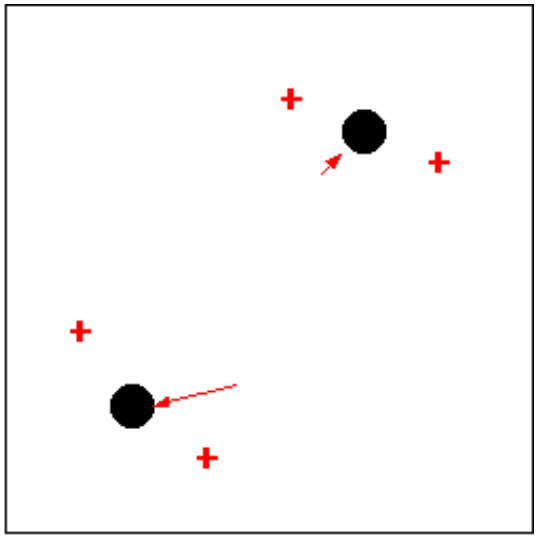


Clustering por K-Médias

- Cada iteração envolve dois passos:
  - ⋆ atribuição de instâncias a claustros
  - ⋆ recomputação dos valores médios



atribuicao



recomputacao de medias

Clustering por K-Médias: Recomputando as Médias

- Para um conjunto de instâncias que foram atribuidas a um claustro c_j, nós recomputamos a média do claustro como:

$$\mu_(c_j) = \frac{\sum_{\vec{x_i} \in c_j} \vec{x_i}}{|c_j|}$$

Clustering por K-Médias

1. dados: um conjunto $X = \{\vec{x_1} \ldots \vec{x_n}\}$ de instâncias
2. selecione k centros iniciais de claustros $\overset{\square}{f_1} \ldots \overset{\square}{f_k}$
3. Enquanto o critério de parada não se verificar, faça:
   (a) para cada cluster c_j faça

$$c_j = \{x_i | \forall \overset{\square}{f_l} \quad sim(\vec{x_i}, \overset{\square}{f_j}) \geq sim(\vec{x_i}, \overset{\square}{f_l})\}$$

   (b) para todas as médias $\overset{\square}{f_j}$ faça

$$\overset{\square}{f_j} = \mu(c_j)$$

Clustering EM

- em k-médias como descrito, instâncias são atribuidas a um e apenas a um claustro
- podemos fazer clustering de k-médias via um algoritmo de Maximização de Expectativas (EM)
  - ⋆ cada claustro representado por uma distribuição (e.g., um Gaussiano)
  - ⋆ Passo **E**: determina qual é a probabilidade de que cada claustro gere cada instância
  - ⋆ Passo **M**: muda centro de claustros de forma a maximizar o likelihood das instâncias

Clustering EM: Variáveis Escondidas

- em cada iteração de clustering por k-médias, tinhamos que atribuir cada instância a um claustro
- no método EM, vamos usar variáveis escondidas para representar essa ideia
- para cada instância $\vec{x_i}$ temos um conjunto de variáveis escondidas $z_{i1}, \ldots, z_{ik}$
- podemos considerar z_{ij} como sendo 1 se $\vec{x_i}$ fôr um membro do claustro c_j e 0 senão

Representação de Claustros

- No método EM, vamos representar cada claustro usando uma variável m-dimensional multivariada gaussiana

$$N_j(\vec{x_i}) = \frac{1}{\sqrt{(2\pi)^m ||\Sigma_j||}} exp\left[-\frac{1}{2}(\vec{x_i}-\vec{\mu_j})^T(\Sigma_j)^{-1}(\vec{x_i}-\vec{\mu_j})\right]$$

- onde:
  - ⋆ $\vec{\mu_j}$ é a média da gaussiana
  - ⋆ $\sum_j$ é a matriz de covariância

Clustering EM

- O algoritmo de EM tenta colocar os parâmetros da gaussiana, Θ, para maximizar a log likelihood dos dados, X

$$\textbf{log likelihood}(X|\Theta) = log \prod_{i=1}^{n} P(\overrightarrow{x_i})$$

$$= log \prod_{i=1}^{n} \sum_{j=1}^{k} N_j(\overrightarrow{x_i})$$

$$= \sum_{i=1}^{n} log \sum_{j=1}^{k} N_j(\overrightarrow{x_i})$$

Clustering EM

- os parâmetros do modelo, Θ, includem as médias, a matriz de covariância e às vezes pesos anteriores para cada gaussiana
- aqui, vamos assumir que a matriz de covariância e os pesos prévios são fixos; vamos focar em estabelecer as médias

Clustering EM: o Passo E

- z_{ij} é uma variável escondida que vale 1 se N_j generou x_i e 0 senão
- no passo **E**, nós computamos h_{ij}, o valor esperado da variável escondida

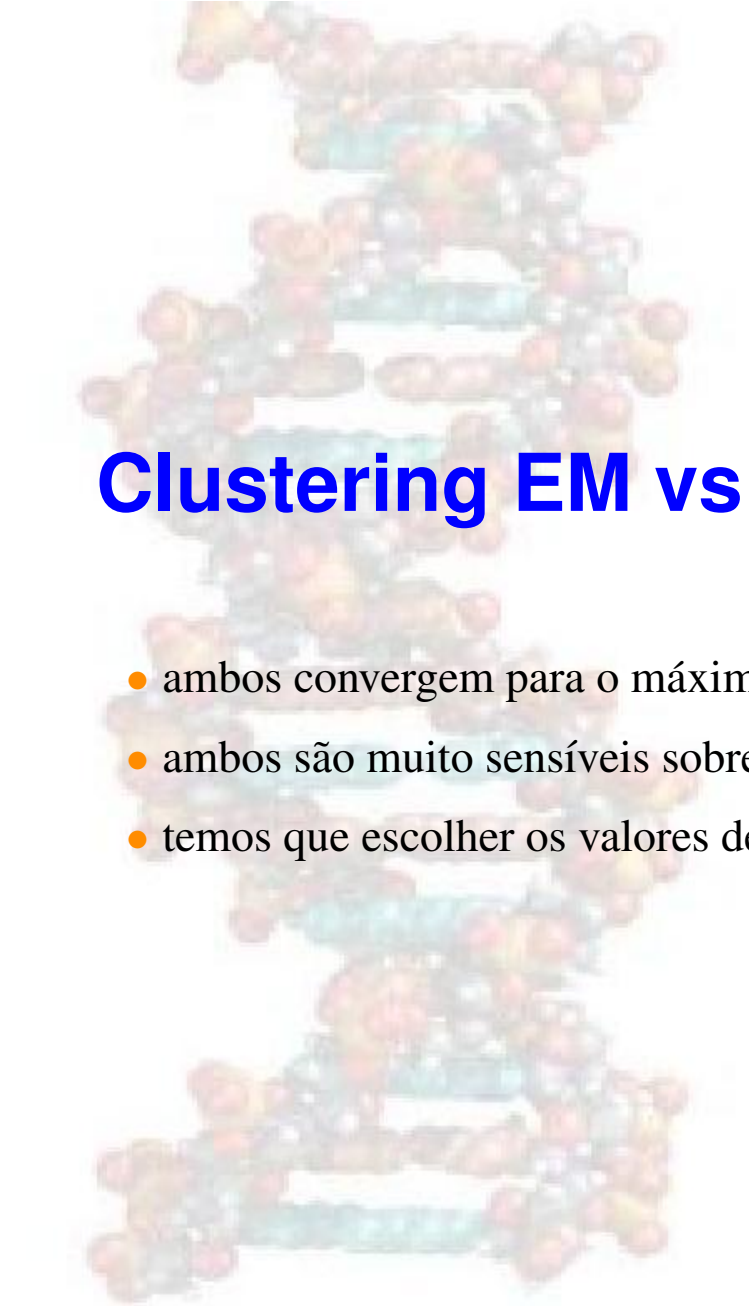
$$h_{ij} = E(z_{ij}|\vec{x_i}) = \frac{P(\vec{x_i}|N_j)}{\sum_{l=1}^{k} P(\vec{x_i}|N_j)}$$

Clustering EM: o Passo M

- dados os valores esperados h_{ij}, podemos reestimar os parâmetros da gaussiana:

$$\vec{\mu'_j} = \frac{\sum_{i=1}^{n} h_{ij} \vec{x_i}}{\sum_{i=1}^{n} h_{ij}}$$

- podemos também re-estimar a matriz de covariância e os pesos da gaussiana, se os estamos variando



Clustering EM vs K-Médias

- ambos convergem para o máximo local
- ambos são muito sensíveis sobre posições iniciais (médias) de claustros
- temos que escolher os valores de k para ambos

Avaliação de Métodos de Clustering

- mesmo tendo recebido dados aleatórios sem nenhuma estrutura, algoritmos de clustering devolvem claustros
- Avaliação:
  - ⋆ Claustros correspondem a categorias naturais?
  - ⋆ Categorias dos claustros são interessantes (há muitas maneiras de particionar os dados)?
  - ⋆ Se usarmos métodos probabilisticos (e.g., EM) podemos questionar: quais as probabilidados dos dados para teste?
  - ⋆ como é que métodos de clustering optimisam semelhança entre os claustros e dissemelhança entre claustros?

Informação Extra

- Tutorial de Baldi (ISMB02)
- Tutorial de Page & Shavlik (ICML03)
- Open-Source Clustering Software
- Array Express
- Conjuntos de Dados sobre Expressão de Genes
- PRIDE: Dados Proteómicos

Classificação Com Expressão de Genes

dados:
- Perfis de expressão para um conjunto de genes/indíviduos/pontos temporais/localizações
- Uma etiqueta para cada perfil

faça:
- *aprenda* um modelo capaz de prever etiqueta para novos perfis

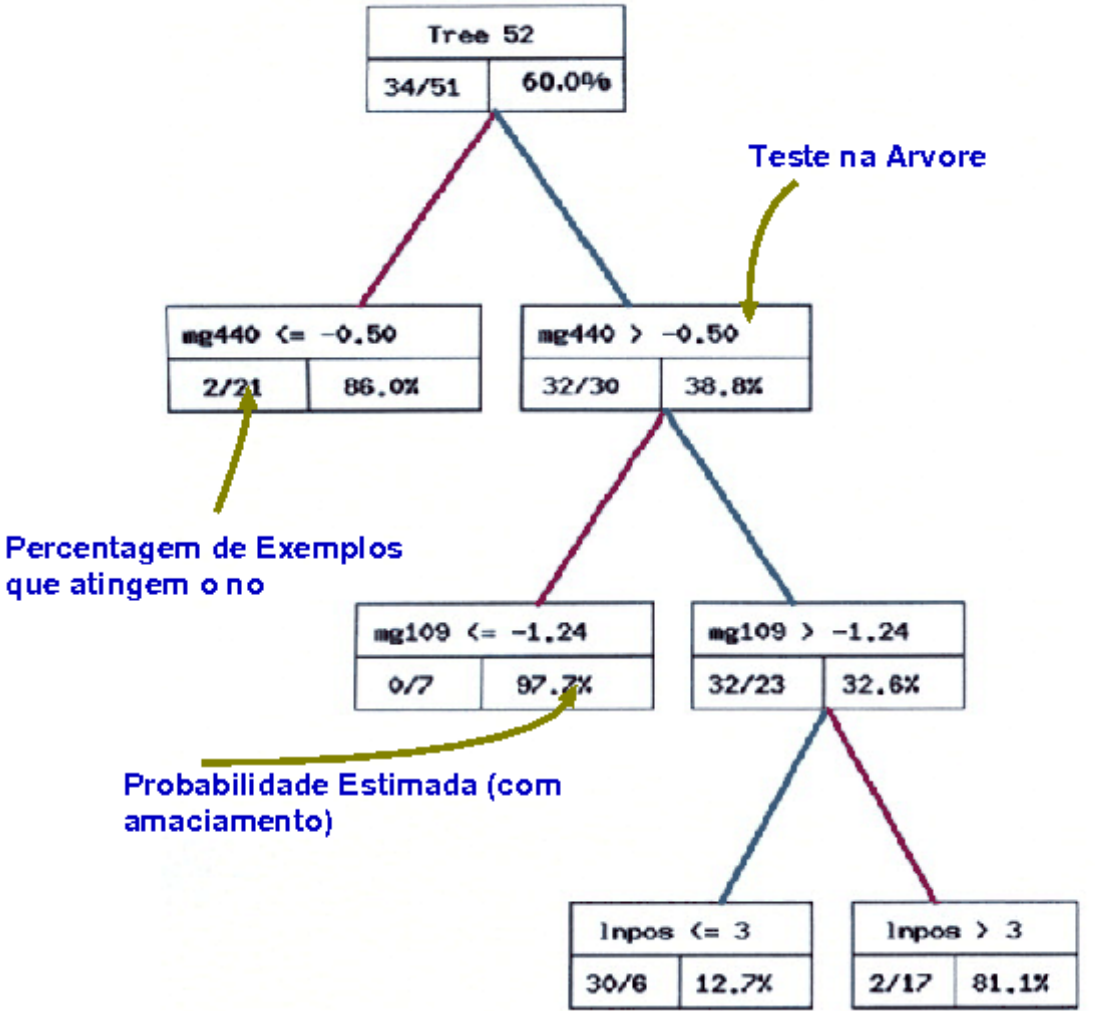
Câncro de Mama

- Obtidos por Nevins e colegas
- Dados de microarrays mais dados clínicos de 86 doentes com
  - ⋆ matrizes Affymetrix dão expressão de 12625 genes
- Objectivo:
  - ⋆ Distinguir individuos de alto risco (recorrência ≤ 5 anos) e baixo risco.

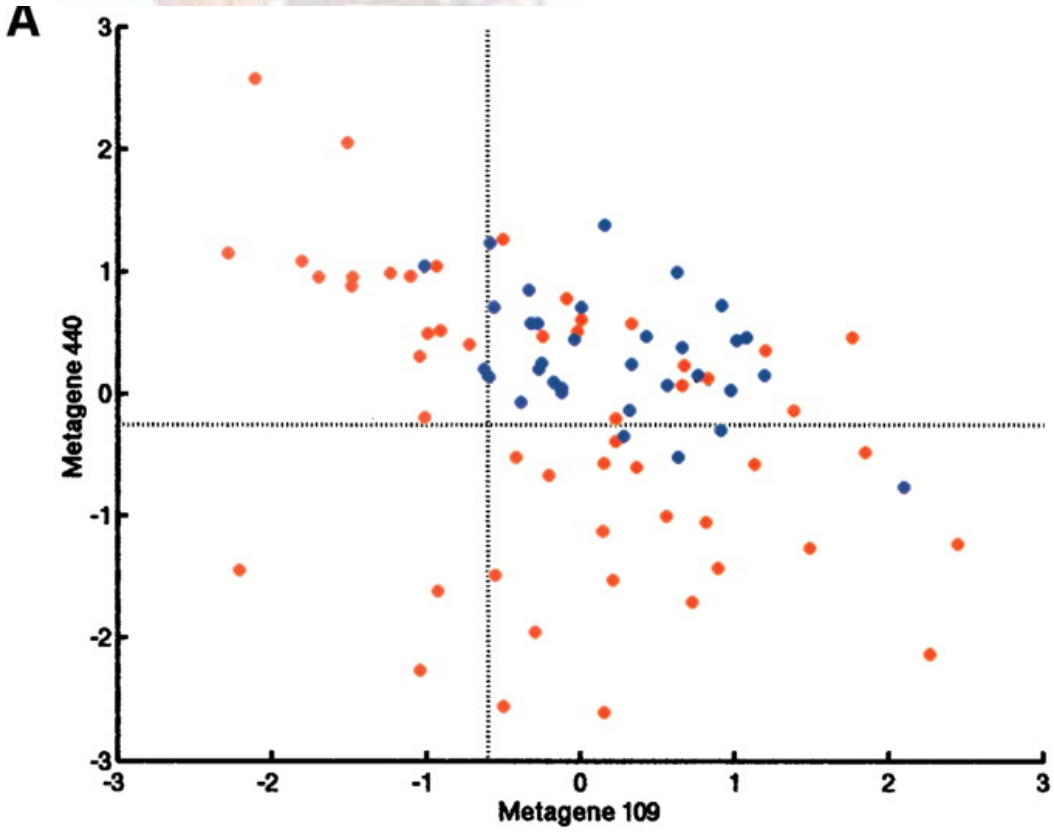
MetaGenes

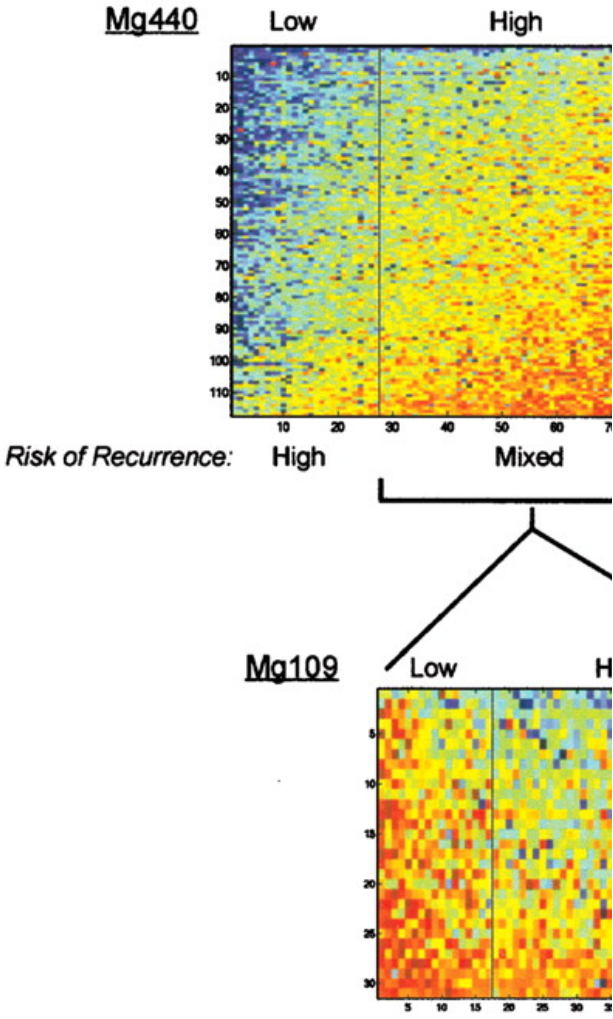
- os atributos não são genes mas sim *grupos de genes*:
  - ⋆ combinações lineares de grupos de genes
- Como?
  1. executar clustering de k-médias ($k0500$) no data-set original
  2. computar os primeiros components principais
  3. o componente principal de cada cluster é um metagene
- Identificaram 496 metagenes

Árvores de Classificação Probabílisticas



Aprender por Divisões nos Eixos





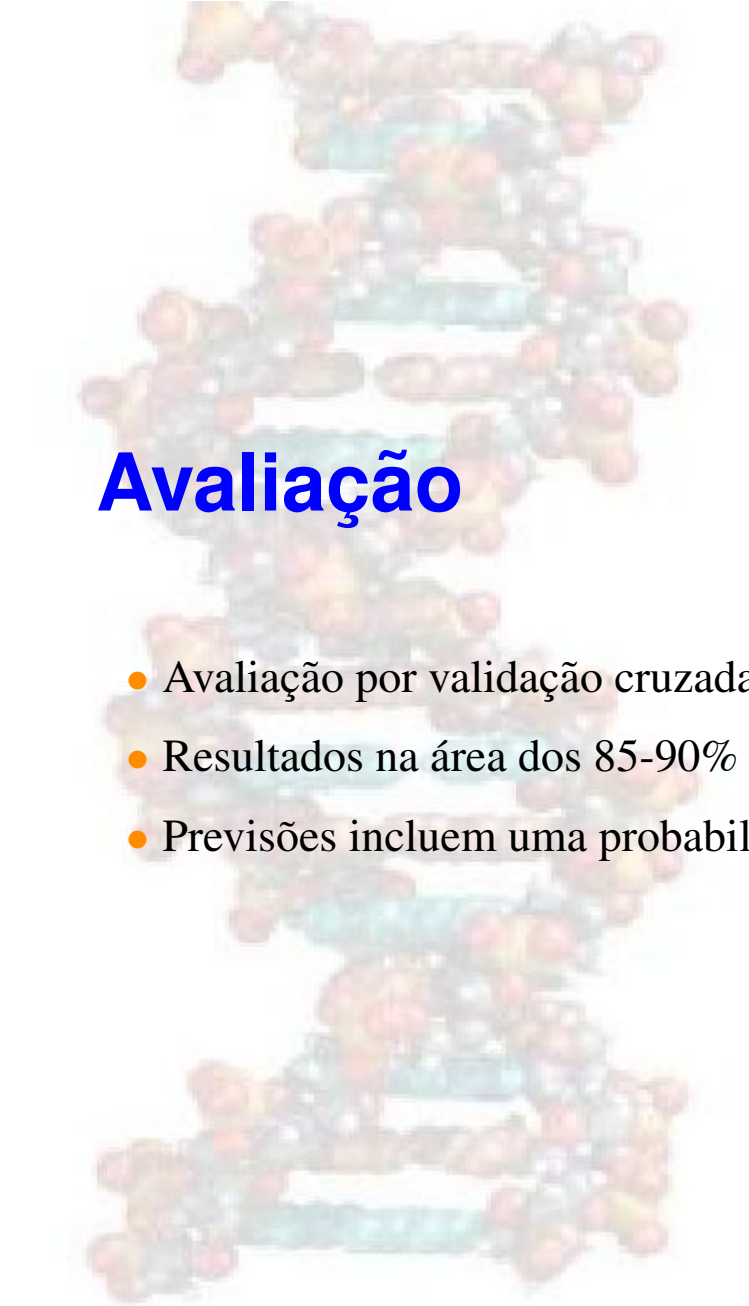
Indução de Árvores de Decisão

- Algoritmos mais populares:
  - ⋆ C4.5 de Quinlan
  - ⋆ CART de Breiman
- Nevins tem um método próprio
- Ideia geral:
  1. Crescer uma árvore recursivamente e de cima para baixo
  2. Selecionando o melhor teste

Algoritmo

$ConstruaArvore(Exemplos)$

1. **if** critério de paragem atingido:
   (a) construa uma folha N
   (b) determinar etiqueta/probabilidades/valores para N
2. **else**:
   (a) construir um nó interior
   (b) Escolher o melhor teste
   (c) Para cada resultado k do teste
      i. se $I_k =$ instâncias com resultado k
      ii. $Folha_k(N) = ConstruaArvore(I_k)$
3. **return** árvore com raíz N



Avaliação

- Avaliação por validação cruzada com *leave-one out*
- Resultados na área dos 85-90%
- Previsões incluem uma probabilidade

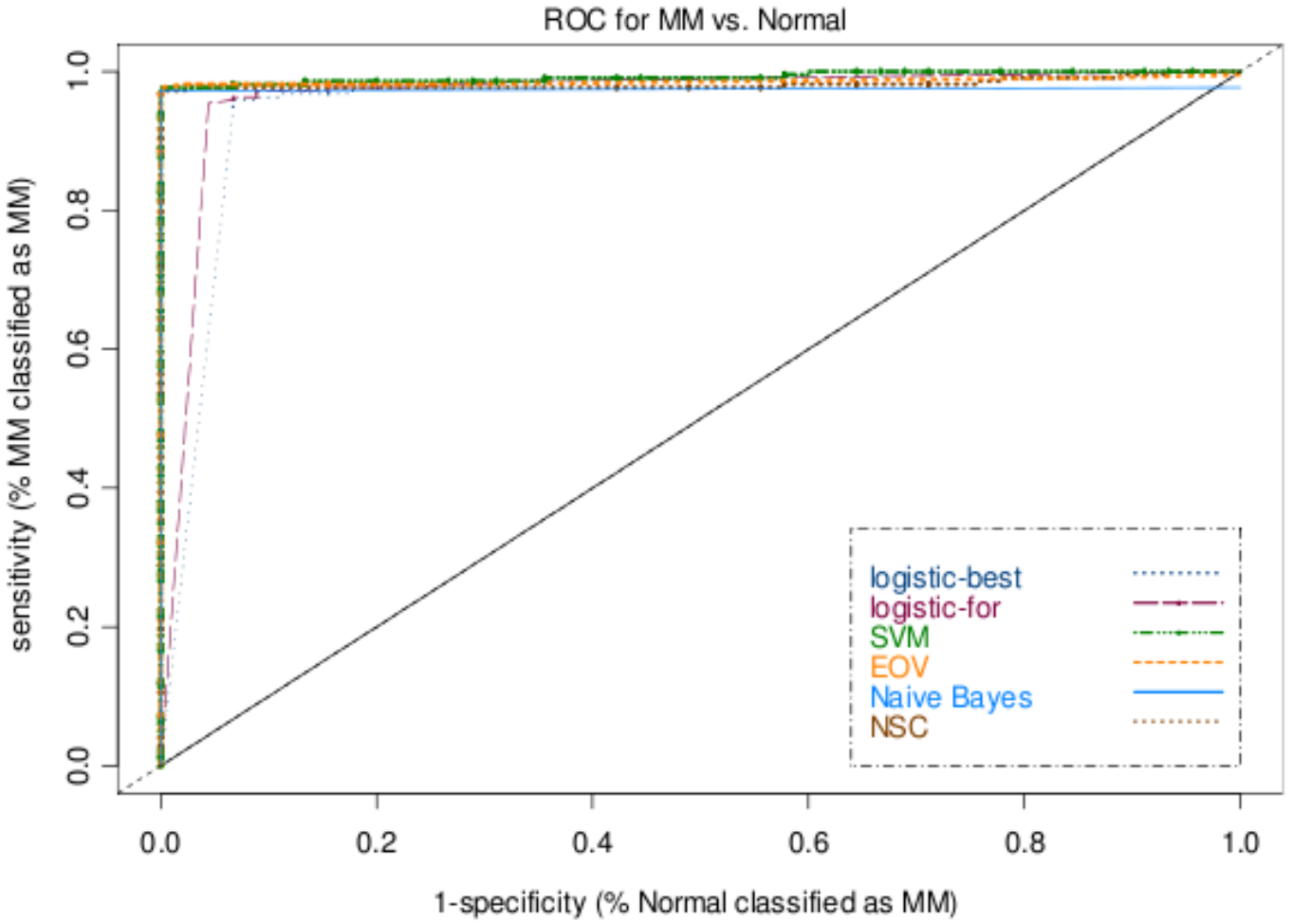
Mieloma Múltiplo e MGUS

- MGUS é uma condição de falta de uma proteina
- pode resultador em mieloma múltiplo, uma forma de cancro
- existem testes de lab para comparar
- Podemos usar expressão de genes?
- Trabalho de Hardin e colegas
- Objectivos:
  1. Permitir diagnóstico molecular
  2. Melhorar compreensão

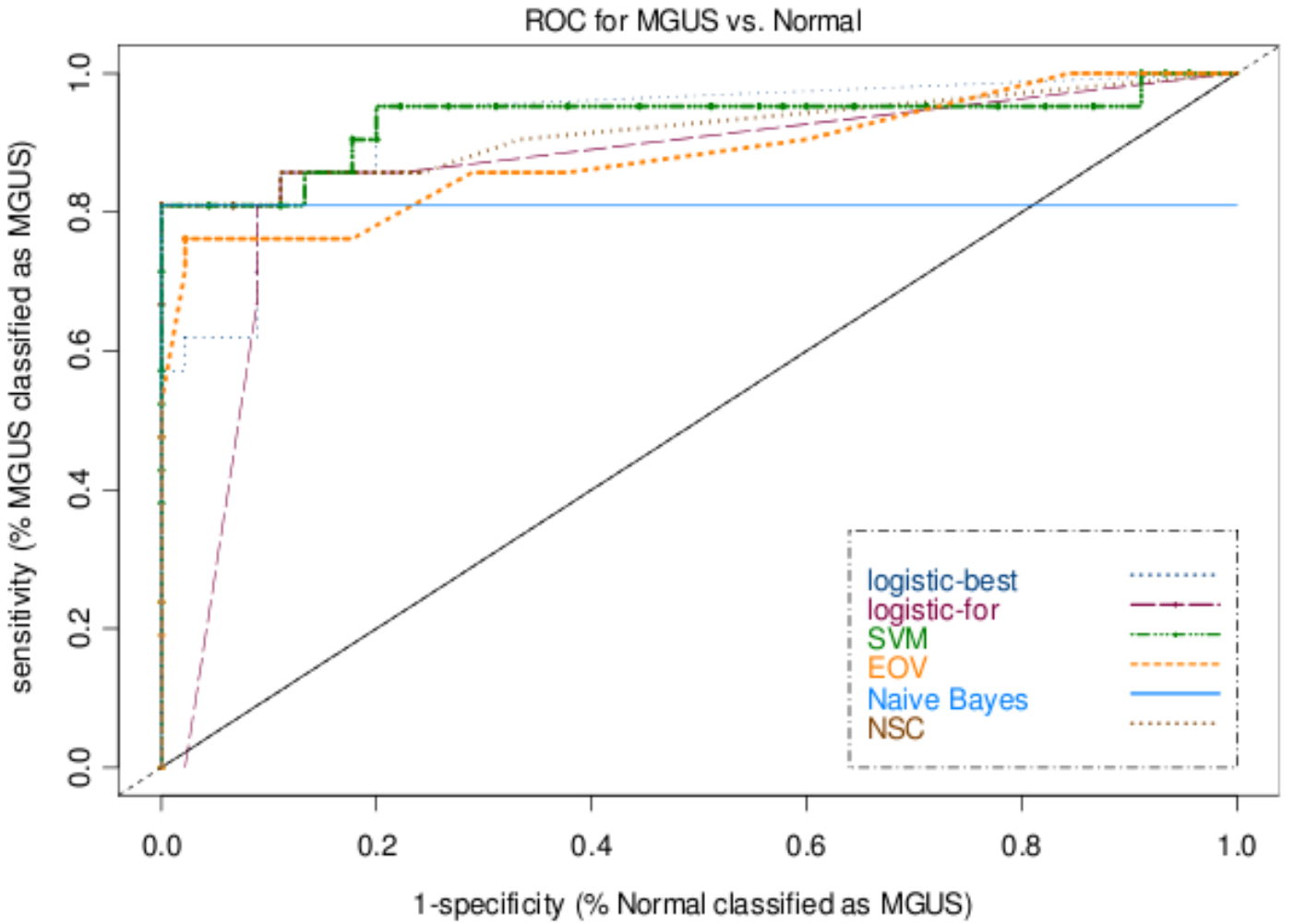
Metodologia

- Affymetrix: ≈ 12625 genes
- Seis métodos de aprendizagem:
  1. Regressão Logística
  2. Árvore de Decisão com C5.0
  3. ensembles de votos
  4. Bayes naivo
  5. centróide mais perto
  6. SVM

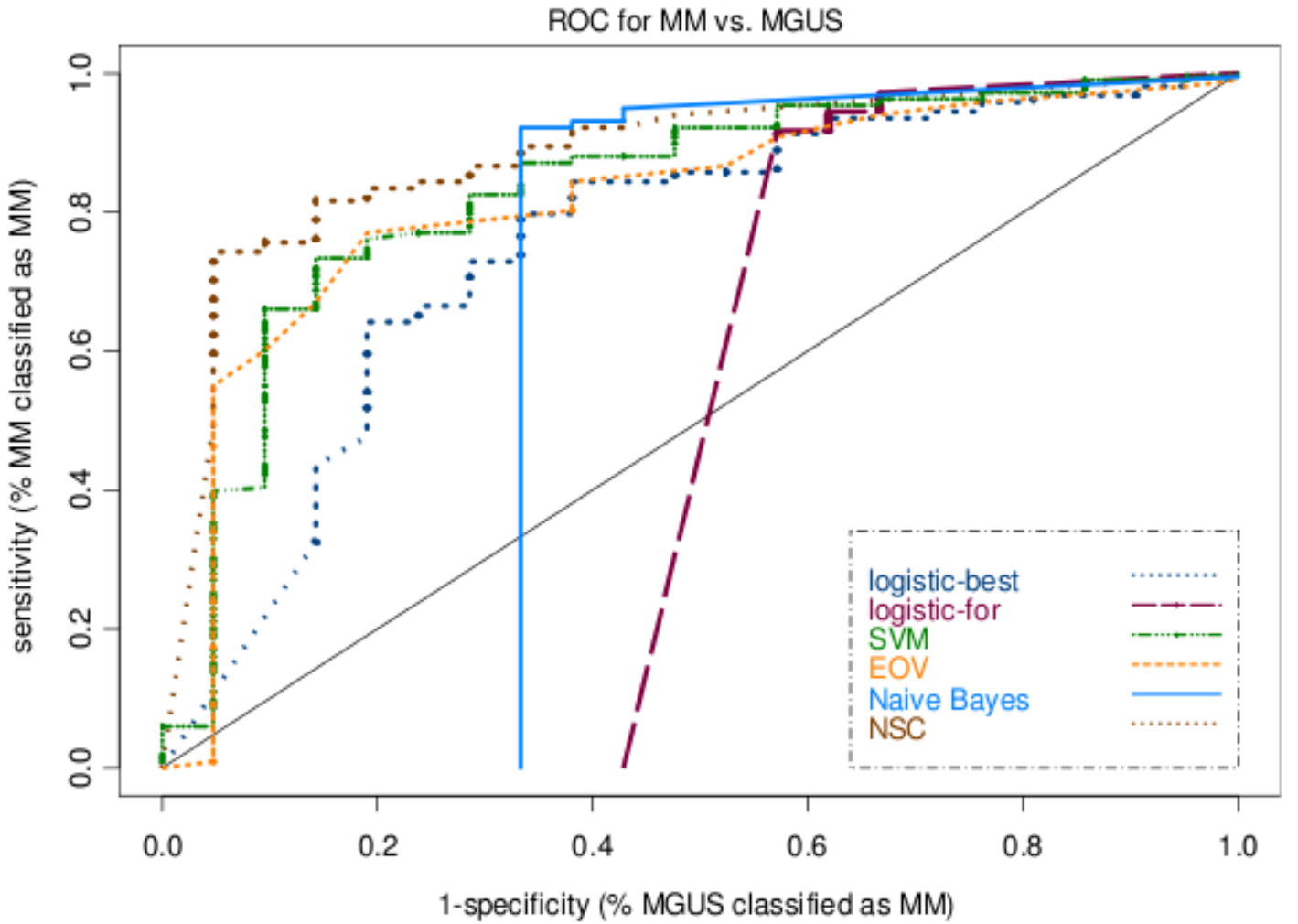
Avaliação Empírica: MM vs Normal



Avaliação Empírica: MGUs vs Normal



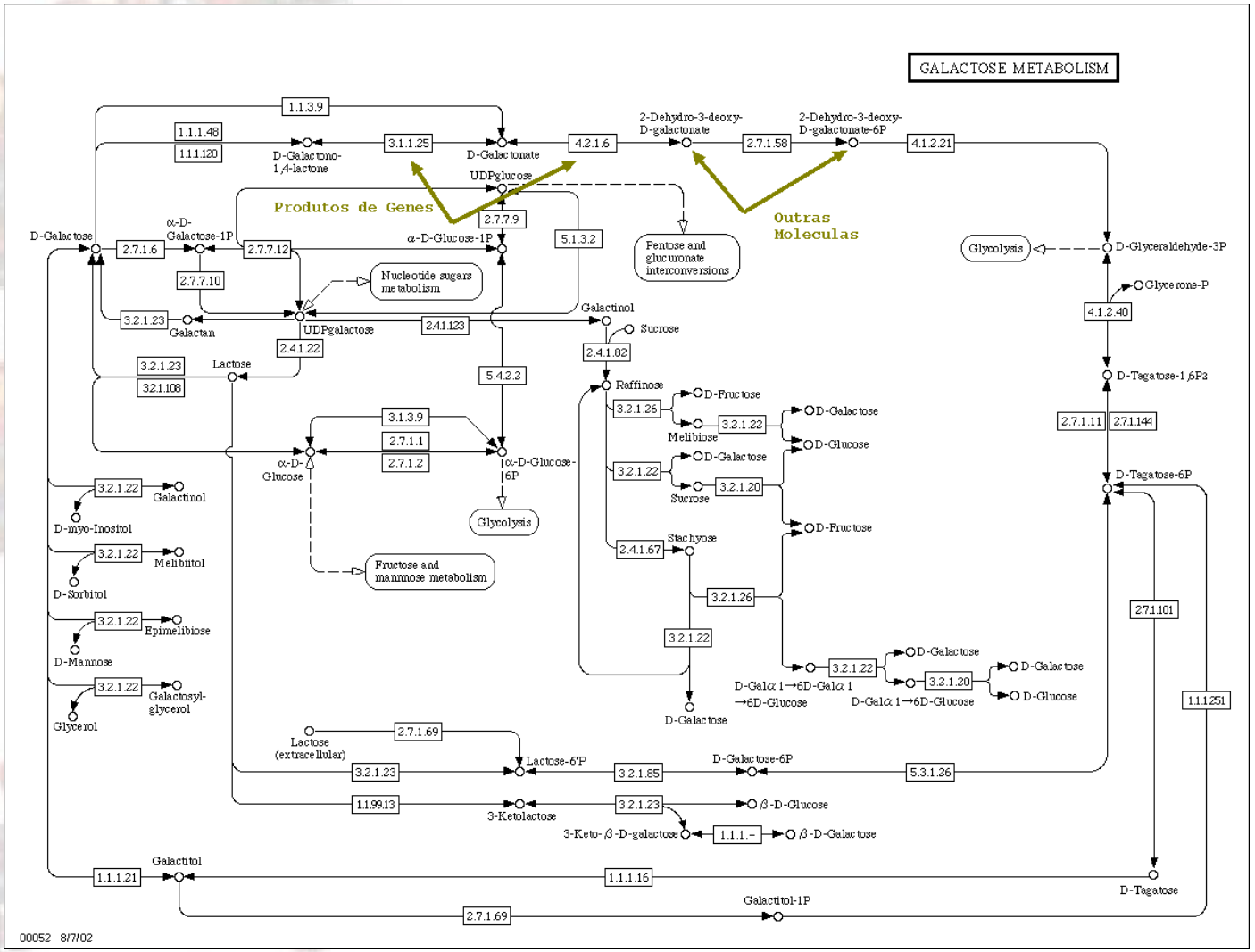
Avaliação Empírica: MGUS vs MM



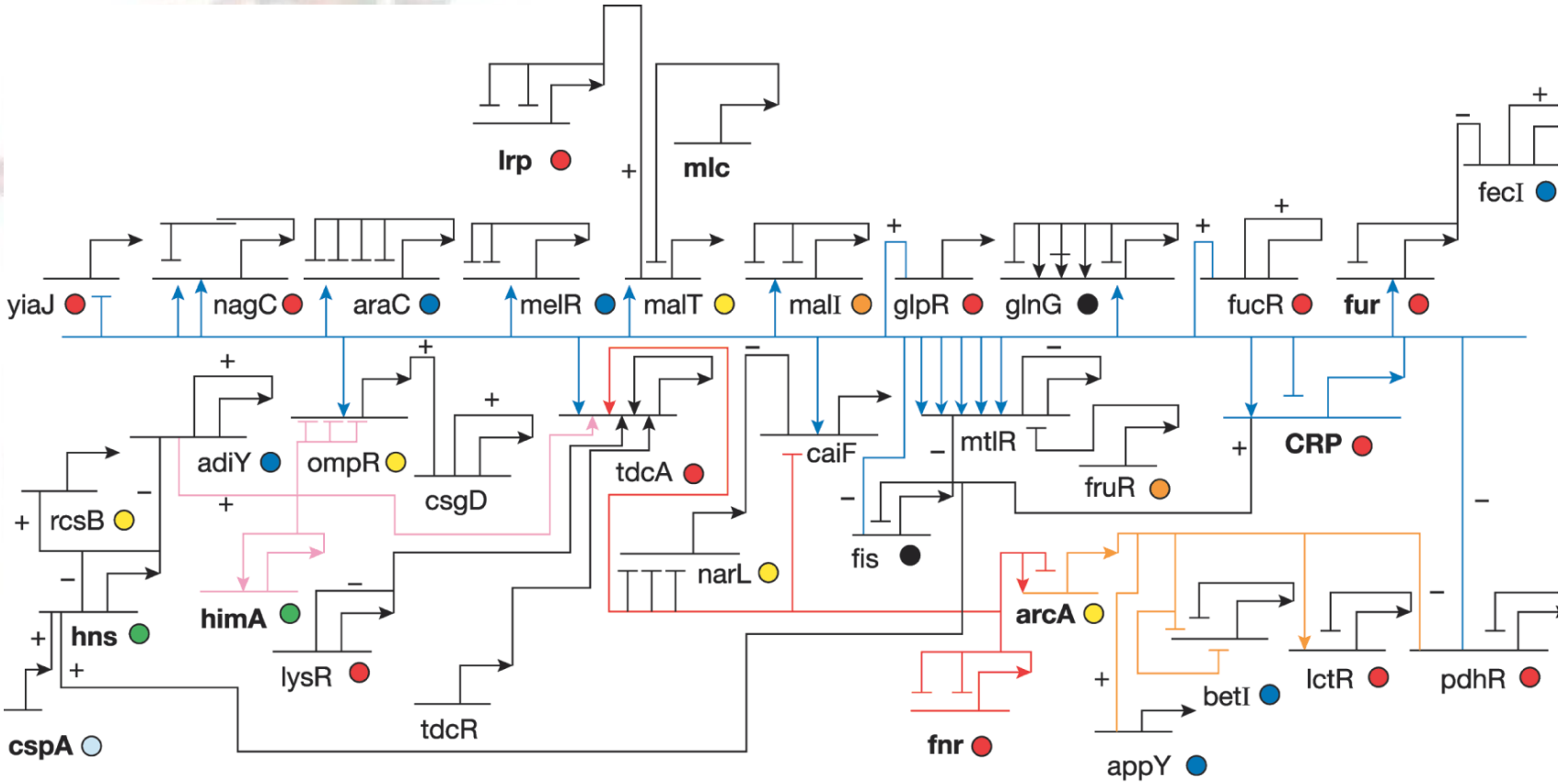
Redes Biológicas

- Metabólicas: reacções onde enzimas intervêm para controlar a conversão de substractos para produtos
- Regulatórias (genéticas): interacções que controlam expressãao de genes
- Sinalização: interacções entre proteínas e ás vezes moléculas menores que enviam sinais de controle desde fora da célula para o núcleo
- Todas estas redes estão ligadas

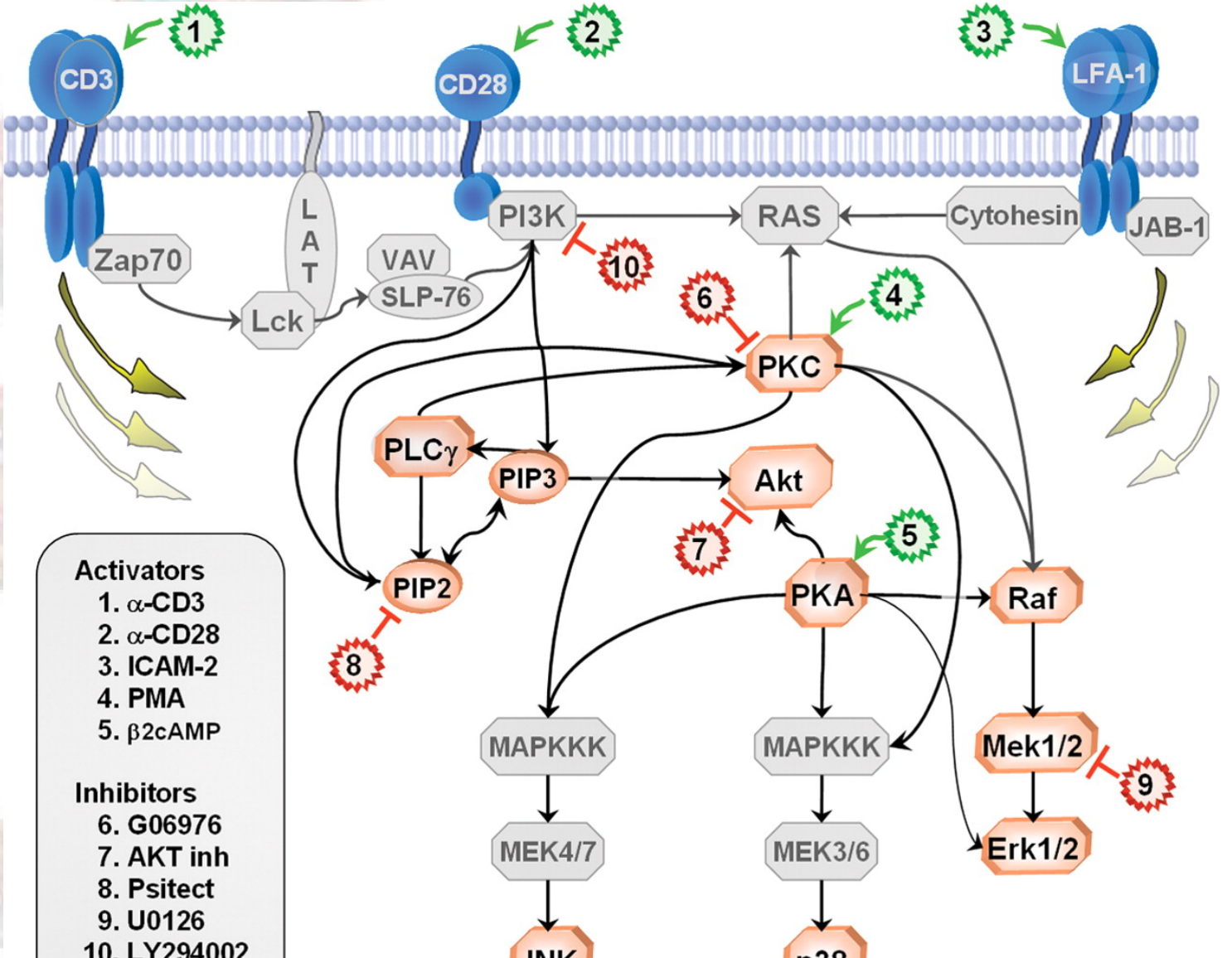
Rede Metabólica



Rede Regulatória da E Coli



Rede de Sinalização



Tarefas

Aprendizagem: dados KB e dados de expressão, tentar descobrir a estrutura da rede

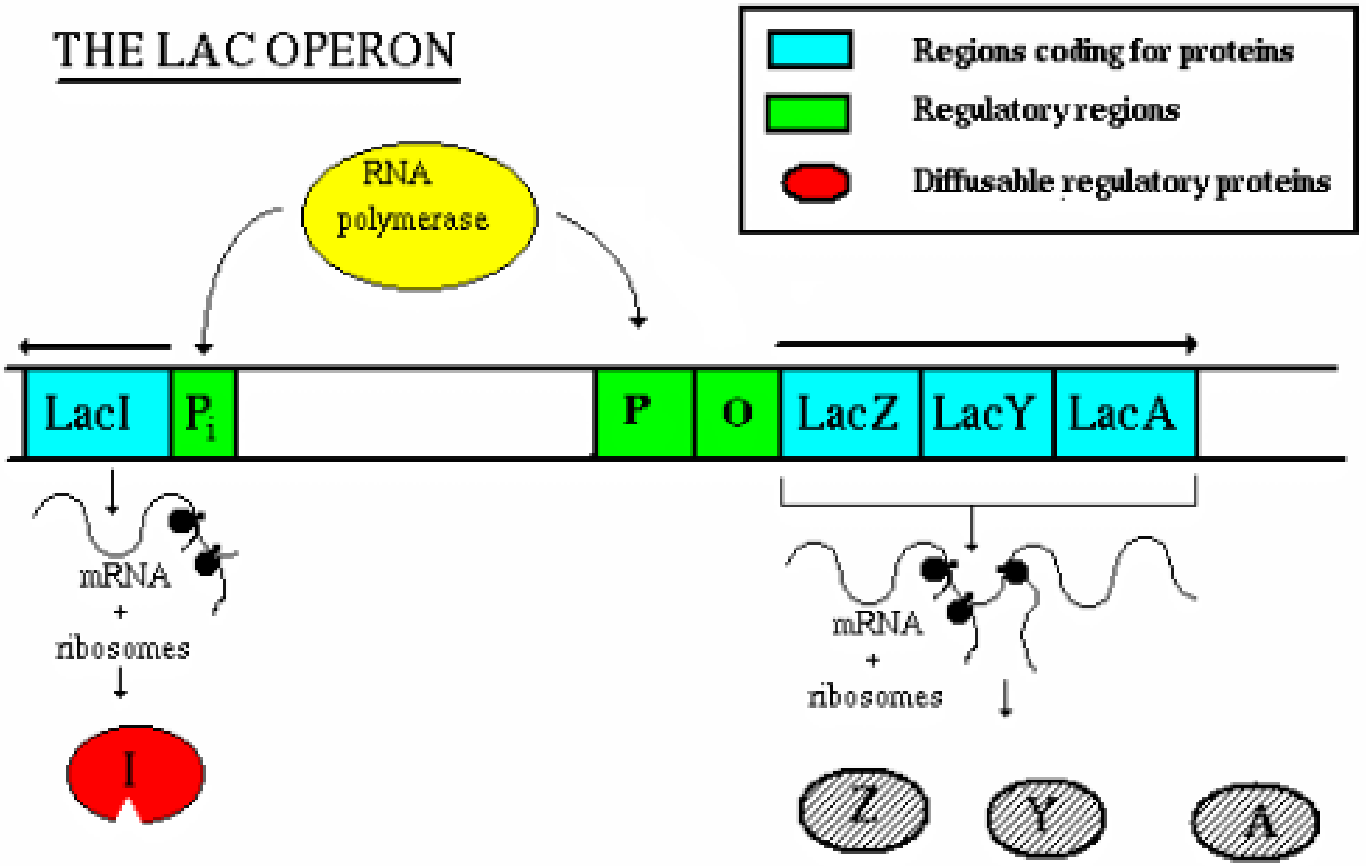
Inferência: dada uma rede (parcial) tentar prever um resultado de interesse biológico (eg, células crescem mais depressa no meio x ou y?)

Problemas:

- Ruído nos dados
- Dados incompletos
- Nem todos os factors podem ser considerados=

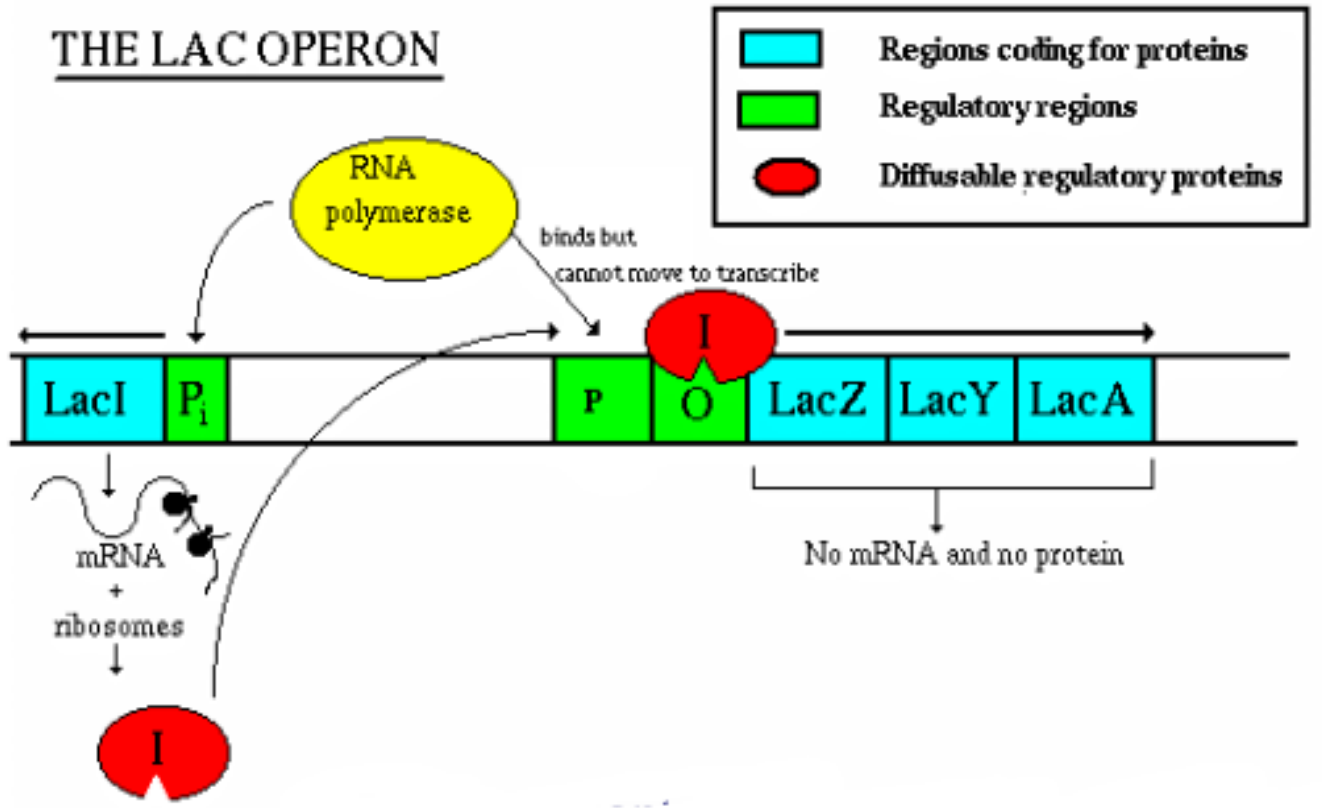
O Operão LAC

- Problema: E-coli pode usar lactose como fonte de energia, mas prefere glucose.
- Como é que ela liga a lactose?



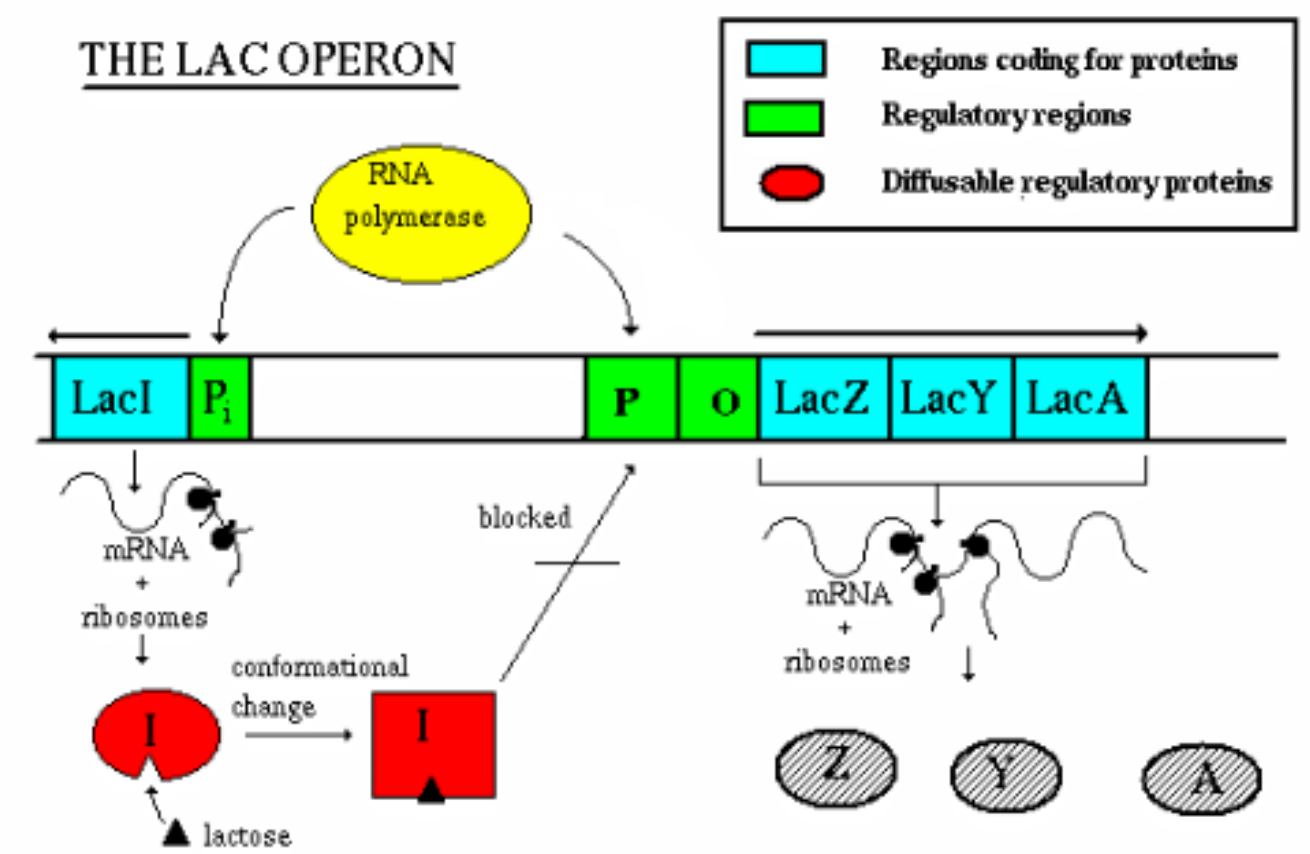
O Operão LAC: Repressão

- Se não há lactose: proteína transcripta por `LacI` reprime transcripção do operão `lac`



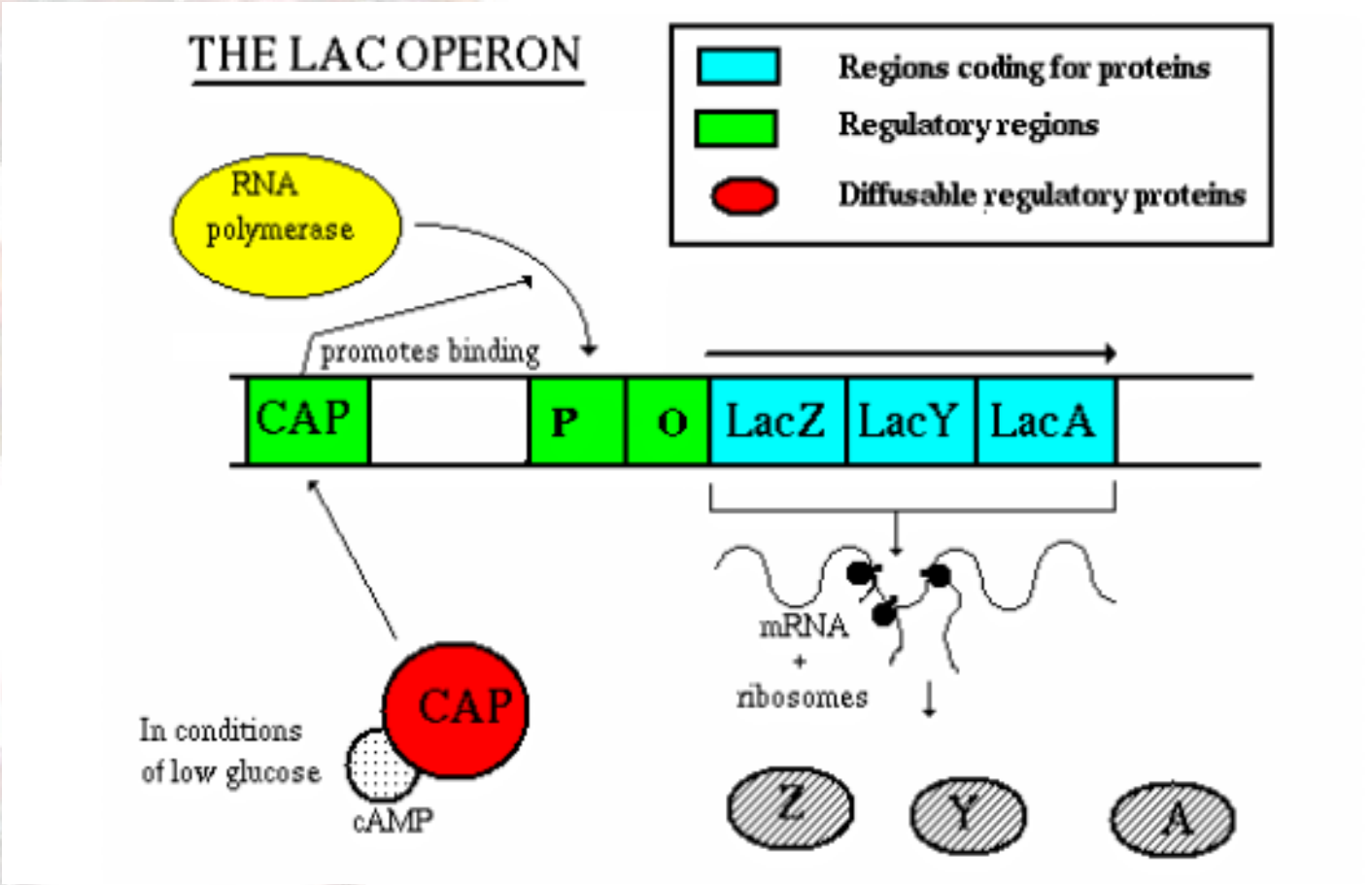
O Operão LAC: Indução

- Se há lactose: proteína transcripta por `LacI` não consegue ligar ao operador de `lac`



O Operão LAC: Activação por Glucose

- Se não há glucose: proteína `CAP` promove ligação de polimerase de RNA e aumenta transcripção.



Representação de Modelos de Rede

- Grafos Dirigidos
- Rede booleana
- Equações Diferenciais
- Redes Bayesianas e outros Modelos Gráficos
- e muito mais!

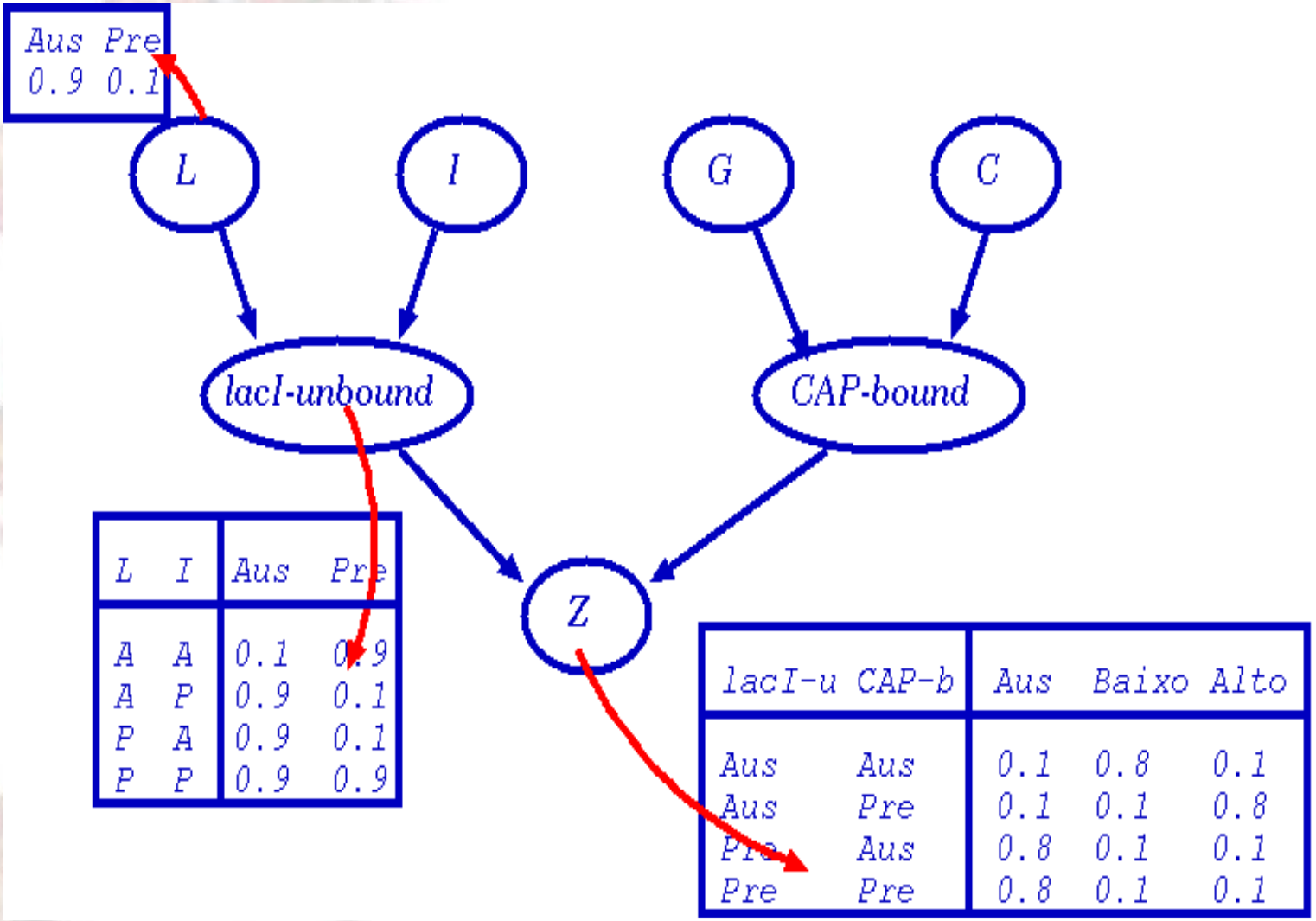
Representação de Modelos de Rede

- Vamos usar as seguintes variáveis:

L (lactose)	`presente,ausente`
G (glucose)	`presente,ausente`
I (lacI)	`presente,ausente`
C (CAP)	`presente,ausente`
$Iac1 - unbound$	`presente,ausente`
$CAP - bound$	`presente,ausente`
Z (lacZ)	`alto,baixo,ausente`

- Suponhamos que o sistema não é determínistico
- A distribuição conjunta tem $2^6 \times 3 = 192$ parâmetros

Uma Rede Bayesiana



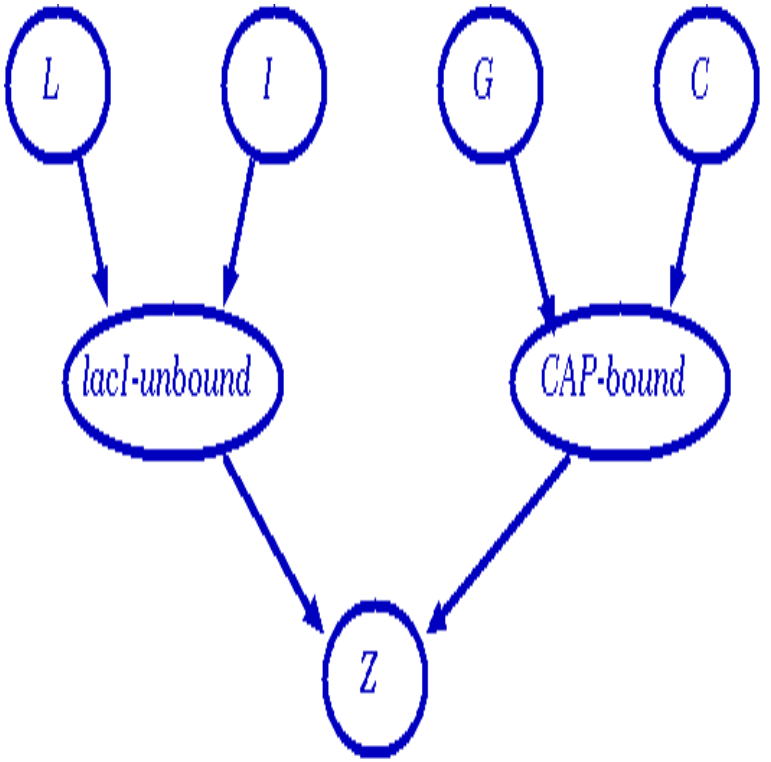
Redes Bayesianas

- São DAGs onde:
  - ⋆ Nó X é uma variáveis aleatória
  - ⋆ Cada X tem uma CPD que representa $P(X|Pais(X))$
- Intuitivamente, se arco de X para Y X influencia directamente Y
- formalmente: dados os pais de X, X é independente dos seus não descendentes.

Redes Bayesianas

- Uma BN representa uma factorização da distribuição conjunta:

$$P(L, I, LU.G, CB, Z) =$$
$$P(L) \times P(I) \times$$
$$P(LU|L, I) \times$$
$$P(G) \times P(C) \times$$
$$P(CB|G, C) \times$$
$$P(Z|LU, CB)$$

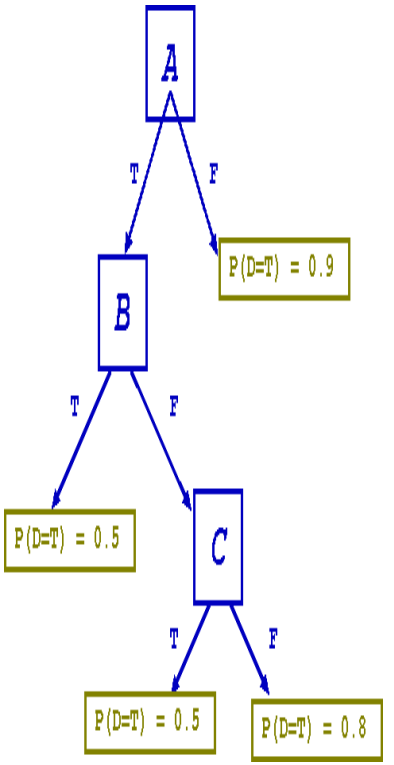


CPTs de Variáveis Contínuas

- Podemos modelas variáveis contínuas

$P(D|A, B, C)$

A	B	C	T	F
T	T	T	0.9	0.1
T	T	F	0.9	0.1
T	F	T	0.9	0.1
T	F	F	0.9	0.1
F	T	T	0.8	0.2
F	T	F	0.5	0.5
F	F	T	0.5	0.5
F	F	F	0.5	0.5



CPTs de Variáveis Contínuas

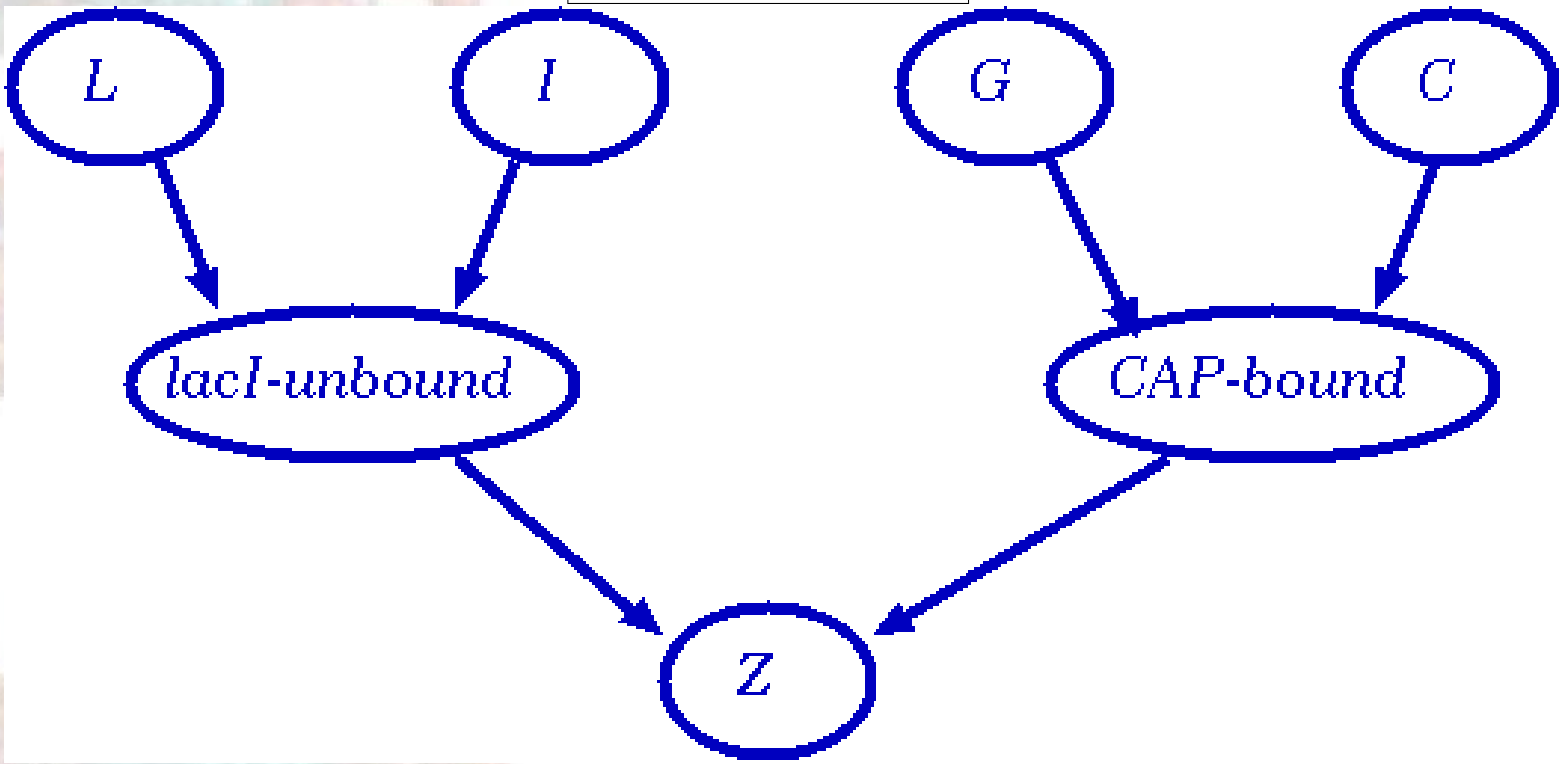
- Podemos modelas variáveis contínuas
- Modelos linares gaussianos:

$$P(X|u_1 \ldots u_n) \approx N(a_0 + \sum_i a_i \times u_i, \sigma^2)$$

- Distribuição normal sobre uma média que depende linearmente dos pais u_i

- Dado um grafo com a estrutura do BN
- E um conjunto de instâncias de treino

L	G	I	C	IU	CB	Z
P	P	P	A	A	L	
P	P	P	A	A	A	
A	P	P	P	A	AL	
			...			



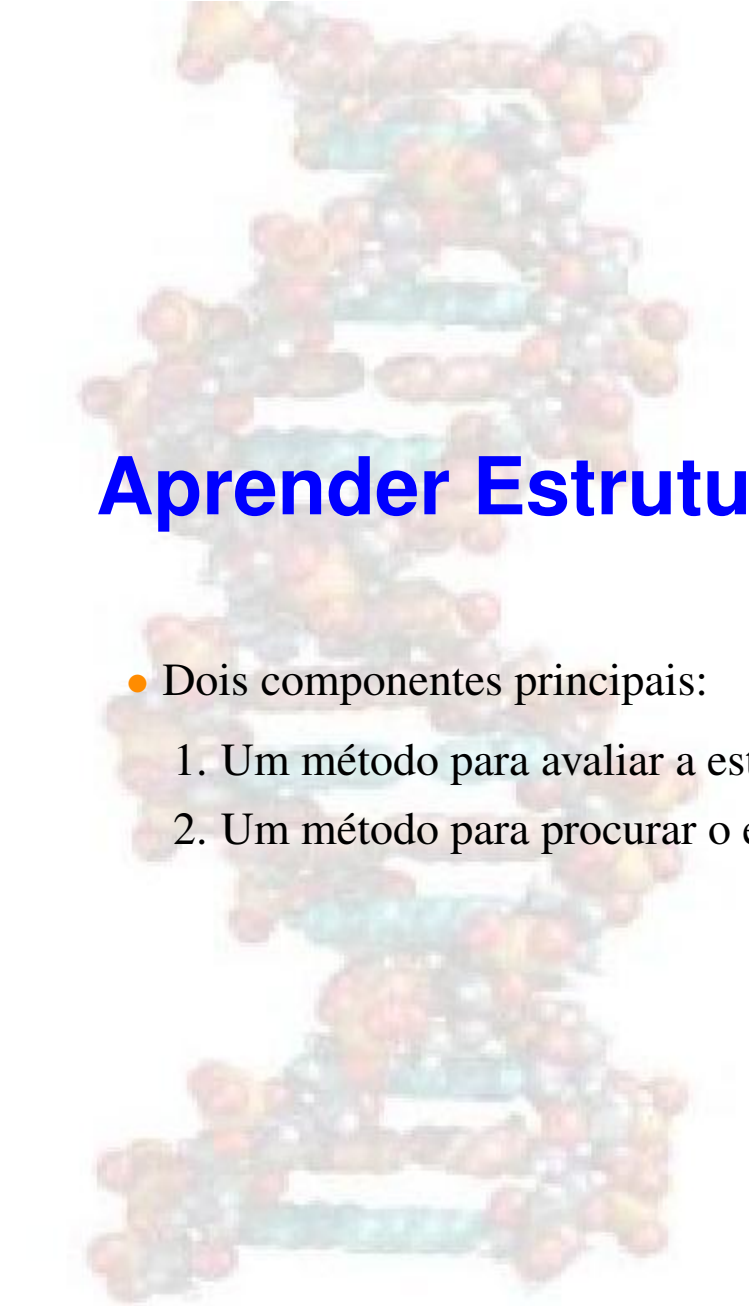
- Preencher as tabelas

Aprender Estrutura

- Dado um conjunto de instâncias de treino

L	G	I	C	IU	CB	Z
P	P	P	A	A	L	
P	P	P	A	A	A	
A	P	P	P	A	AL	
			...			

- Obter a estrutura do grafo
- e provavelmente os parâmetros



Aprender Estrutura

- Dois componentes principais:
  1. Um método para avaliar a estrutura de um BN
  2. Um método para procurar o espaço de estruturas

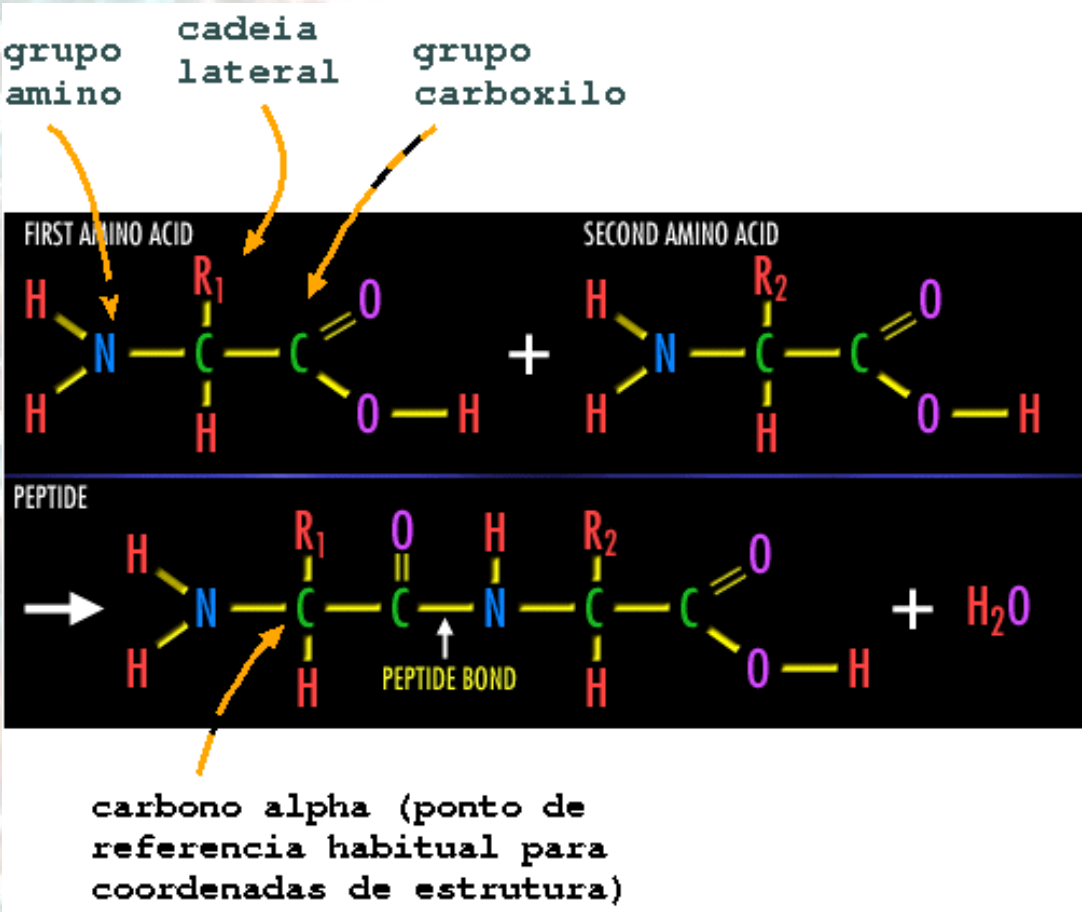
Estrutura de Proteínas

- sabemos que a função de uma proteína é determinada em muito pela sua forma 3D (dobragem, conformação)
- podemos prever a forma 3D de uma proteína dado apenas o seu código em aminoácidos?
- A resposta em geral é não!
- mas métodos que nos dão uma descripção parcial da estrutura 3D podem ajudar

Arquitectura de Proteínas

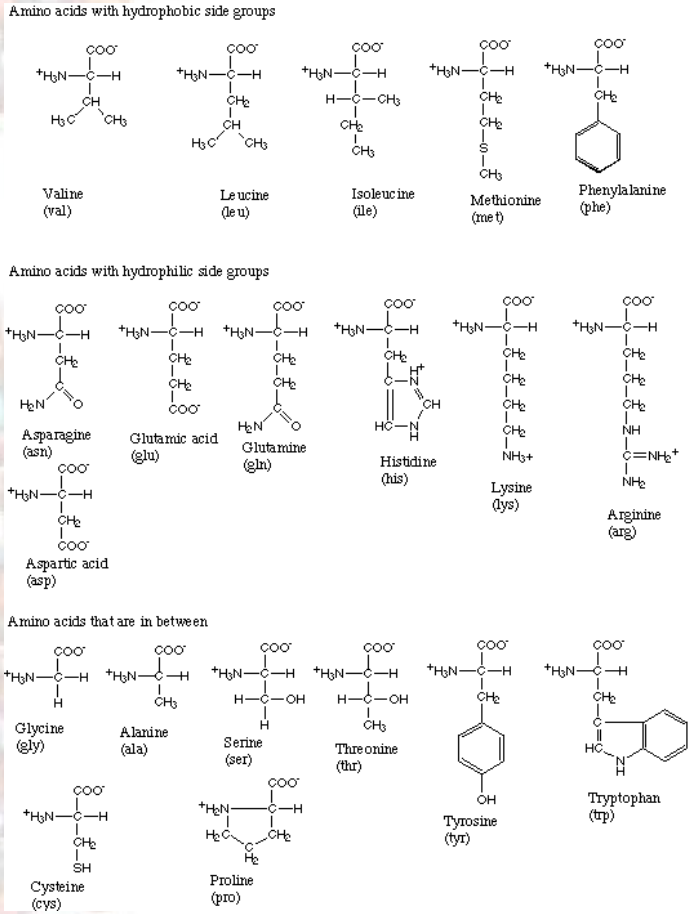
- As proteínas são polipeptideos consistindo de amino-ácidos ligados por elos entre peptideos
- cada amino-ácido consiste de:
  - ⋆ um átomo de carbono central
  - ⋆ um grupo de amino, NH_2
  - ⋆ um grupo carboxilo, COOH
  - ⋆ uma cadeia lateral
- diferências em cadeias laterais distinguem diferentes amino-ácidos.

Ligações de Peptideos



Cadeias Laterais de Amino-Ácidos

- cadeias laterais variam em: forma, tamanho, polaridade e carga.



O que determina a Conformação?

- Em geral, a sequência de amino-ácidos de uma proteína determina a sua forma 3D [Anfinsen et al.,1950s]
- mas existem excepções:
  - ⋆ todas as proteínas podem ser desnaturadas
  - ⋆ algumas proteínas são inerentemente desordenadas (i.e., não têm estrutura regular)
  - ⋆ algumas proteínas recebem ajuda para se dobrar de chaperones
  - ⋆ existem vários mecanismos a partir dos quais a conformação de uma proteína pode ser mudada ao vivo:
    - ∗ fosforilação
    - ∗ priões
    - ∗ etc.

O que determina a Conformação?

- Que propriedades físicas das proteínas determinam a sua dobragem?
  - ⋆ Rigidez do "backbone" da proteína (coluna)
  - ⋆ Interações entre amino-ácidos, incluindo:
    - ∗ interacções eletrostáticas
    - ∗ forças de van der Waals
    - ∗ ligações de hidrogénio,
    - ∗ http://en.wikipedia.org/wiki/Disulfidedisulfidos
  - ⋆ interacções dos amino-ácidos com a água

Níveis de Descrição

A estrutura das proteínas é muitas vezes descrita em quatro diferentes escalas:

- estrutura primária
- estrutura secundária
- estrutura terciária
- estrutura quaternária

Níveis de Descrição

primary structure
(amino acid sequence)

secondary struc
(α-helix)

Níveis de Descrição

tertiary structure
(folded individual peptide)

quaternary structure
(aggregation of two or more peptides)

Estrutura Secundária

- estrutura secundária refere-se a algumas estruturas que se repetem frequentemente
- é uma descrição local da estrutura
- duas estruturas secundárias comuns:
  - ⋆ hélice α
  - ⋆ linhas/folhas β
- uma terceira categoria, chamada de bobina ou laço, refere-se a tudo o resto

Classes DSSP

Convenção para estrutura secundária:

G 3-turn helix (310 helix): Tam min 3 resíduos.

H 4-turn helix (α helix). Tam min 4 resíduos.

I 5-turn helix (π helix). Tam min 5 resíduos.

T hydrogen bonded turn (3, 4 or 5 turn)

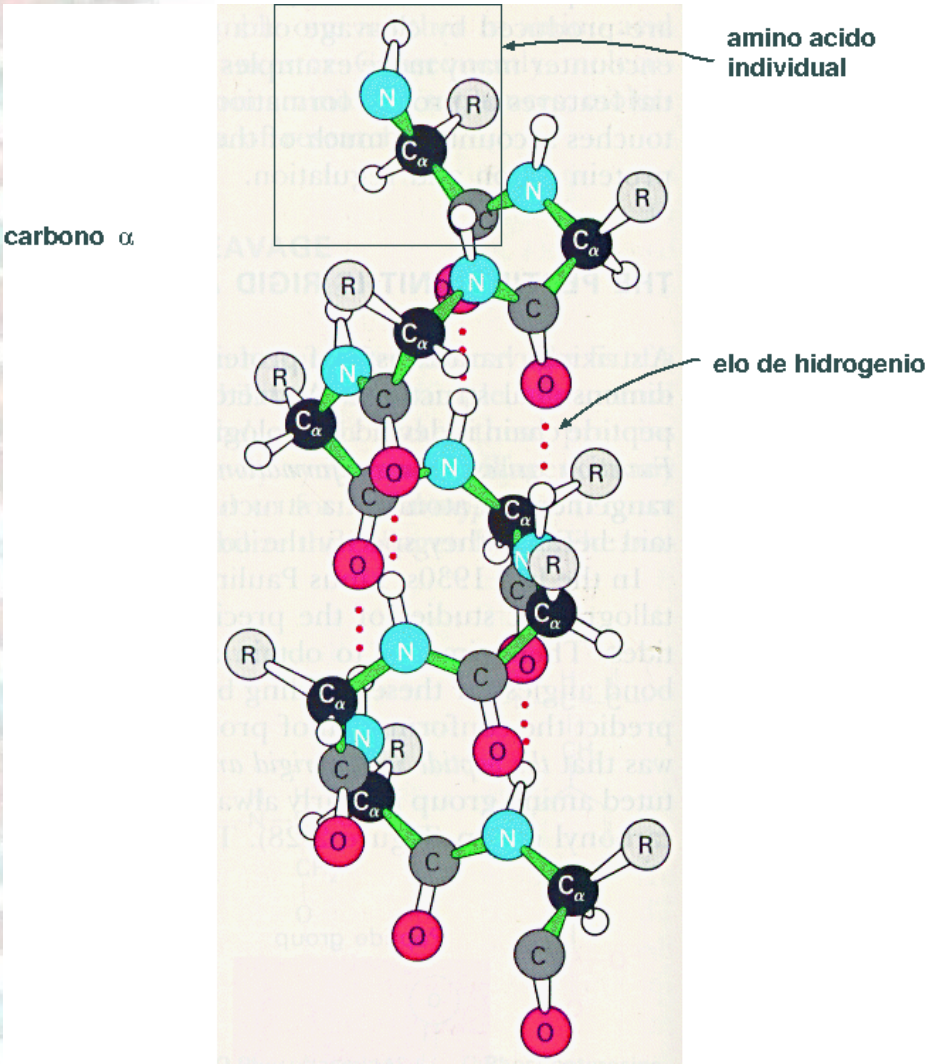
E beta sheet in parallel and/or anti-parallel sheet conformation (extended strand). Tam min 2 resíduos.

B residue in isolated beta-bridge (single pair beta-sheet hydrogen bond formation)

S bend (the only non-hydrogen-bond based assignment)

Resto como espaço, **C** (coil) ou **L** (loop)

Hélices α



Folhas β

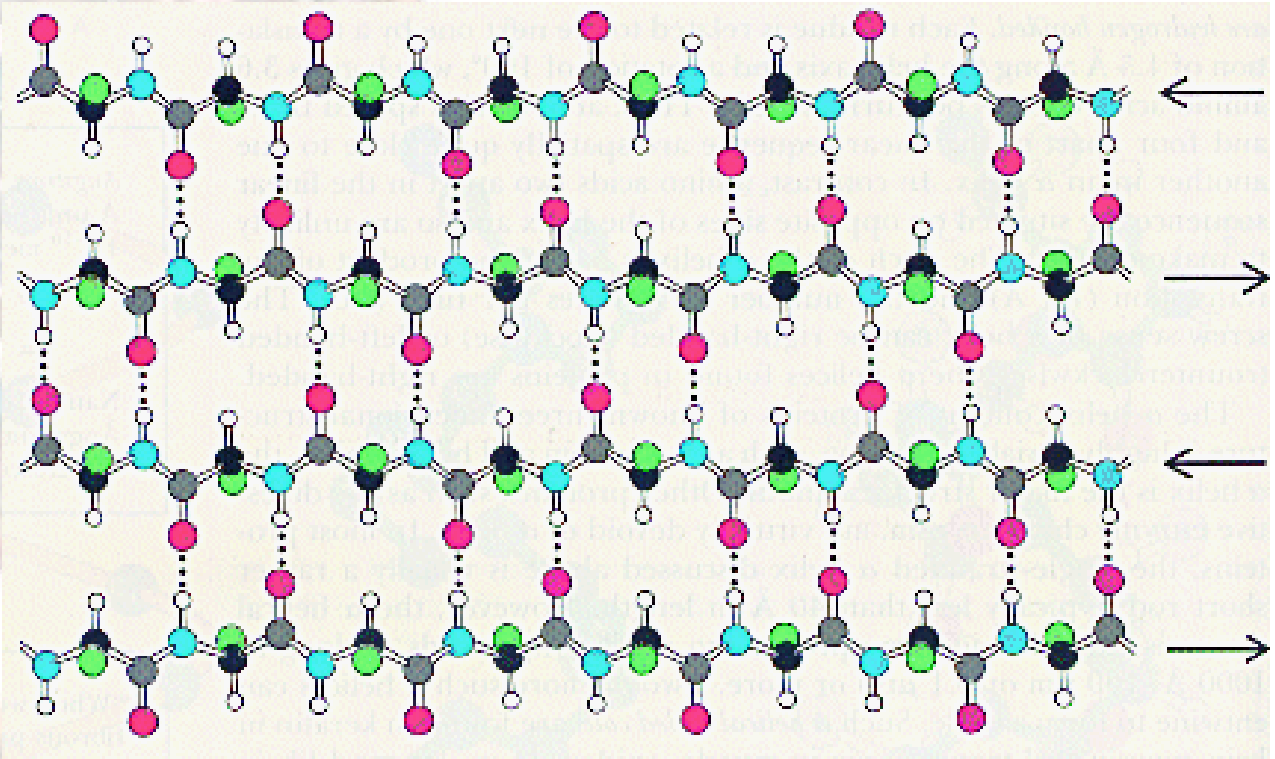
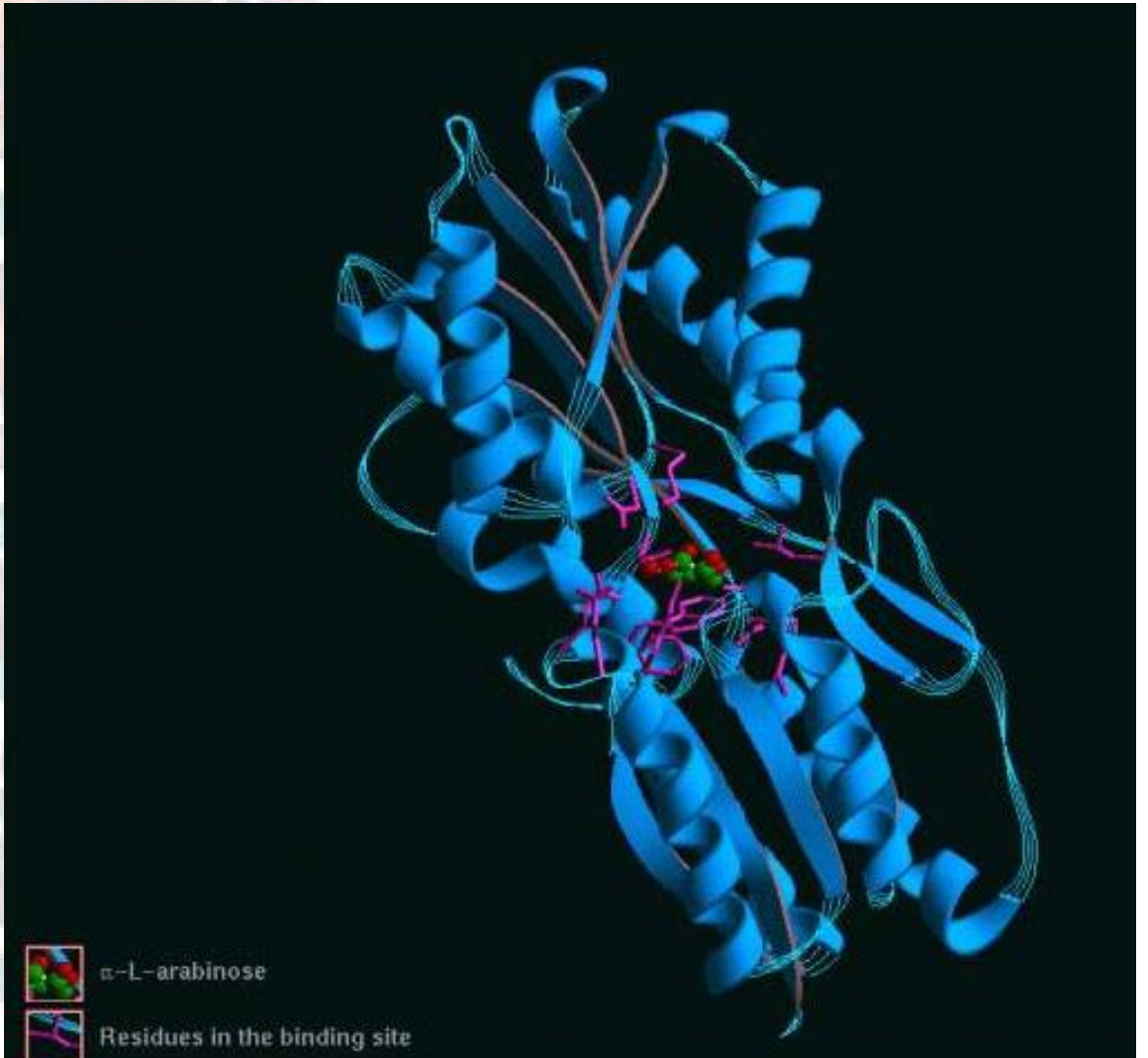


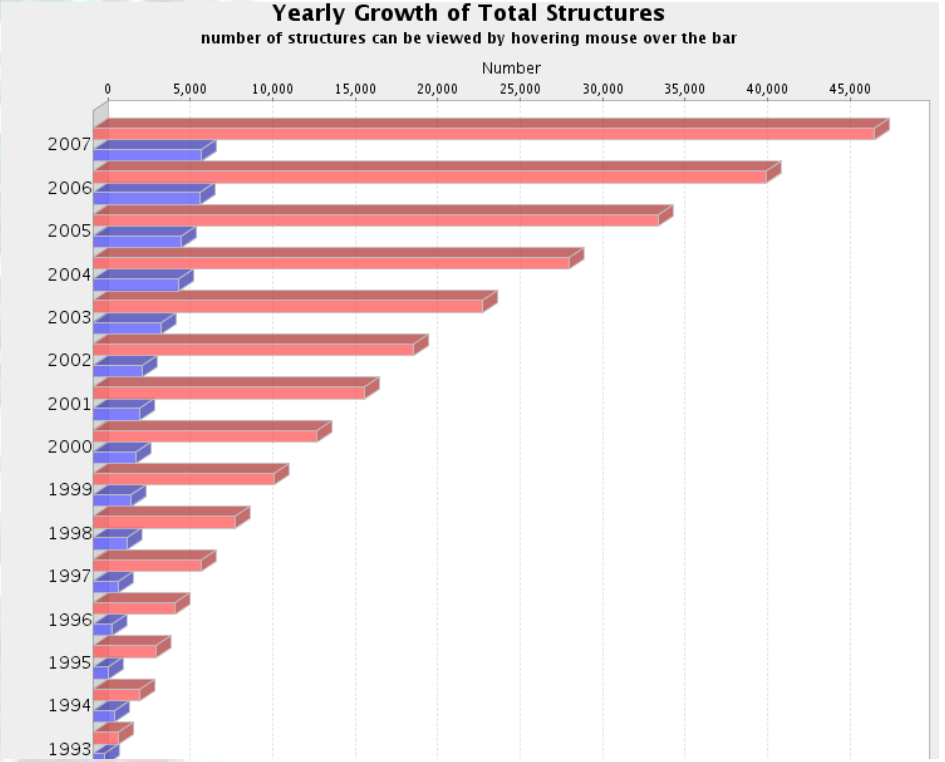
Diagrama em fita mostrando Estruturas Secundárias



Determinação de Estrutura de Proteínas

- A estrutura de proteínas pode ser determinada experimentalmente (em muitos casos) por:
  - ⋆ cristalografia de raios-x
  - ⋆ ressonância magnética nuclear (NMR)
- mas isto é muito caro e gasta muito tempo
- Questão chave: podemos prever estruturas por métodos computacionais?

Estruturas e Sequências



- a versão de 13/11/07 de SWISS-PROT, em contraste, tem entradas para 289,473 sequências de proteínas.

Aparte: Structural Genomics

- iniciativa mundial tentando determinar estruturas de proteínas com alto desempenho
- Acessível do PDB
- 8 centros na Europa

Métodos de Previsão de Estrutura de Proteínas

- Previsão em 1D:
  - ⋆ Estrutura secundária
  - ⋆ acessibilidade de solventes (que resíduos estão expostos em água, que resíduos estão enterrados)
  - ⋆ hélices transmembranas (que resíduos atravessam membranas)
- Previsão em 2D:
  - ⋆ contactos entre resíduos e fitas
- Previsão em 3D:
  - ⋆ modelação por homologia
  - ⋆ reconhecimento de dobras (e.g., via fios)
  - ⋆ previsão desde o principio (e.g., vai dinâmica molecular)

Previsão de Estrutura Secundária

- Dado: uma sequência de proteína
- faça: preveja um estado de estrutura secundária (α, β, bobina) para cada amino-ácido na sequência.
- exemplo:

KELVLALYDYQEKSPREVTMKKGDILTLLM...

ccc$\beta\beta\beta\beta$ccccccccccccc$\beta\beta\beta\beta$ccccccc$\beta\beta\beta\beta\beta\beta$...

Modelação Por Homologia

- Observação: proteínas com sequências similares tendem a dobrar em estruturas similares
- Dados: uma sequência de interrogação Q, um banco de dados de estruturas de proteínas
  - ⋆ encontrar proteína P tal que:
    - ∗ estrutura de P é conhecida
    - ∗ P tem alta semelhança com Q
- devolve estrutura de P como aproximação a estrutura de Q

Modelação Por Homologia

- A maior parte dos pares de proteínas com estrutura semelhante são homólogos remotos ($< 25\%$ de identidade entre as sequências)
- modelação por homologia em geral não funciona para homólogos remotos; a maior parte dos pares de proteínas com $< 25\%$ não estão relacionados