

Tutorial 8

Dados Moleculares - Alinhamento

BIZ0433 - INFERÊNCIA FILOGENÉTICA: FILOSOFIA, MÉTODO E APLICAÇÕES.

Conteúdo

Objetivo	130
8.1 Contextualização	131
8.2 Alinhamento, aplicativo e topologias	132
8.2.1 Alinhamento manual no AliView	132
8.2.2 Clustalw	133
8.2.3 Clustal- Ω	133
8.2.4 Muscle	133
8.2.5 Mafft	134
8.2.6 Avaliação dos resultados	135
8.3 Parâmetros de alinhamento	137
8.3.1 Manipulação de parâmetros em MAFFT	138
8.4 Alinhamento progressivo e árvores-guias	139
8.4.1 Examinando a influência de diferentes árvores-guias em alinhamentos múltiplos	140
8.5 Consistência interna da análise	142
8.5.1 Consistência de premissas de alinhamento	143
8.6 Questões	145
8.7 Leitura recomendada para discutir seus resultados	146
8.8 Referências	146

Objetivo

O objetivo deste tutorial é apresentar os conceitos associados ao alinhamento múltiplo de sequências nucleotídicas e subsequente análise filogenética. Neste tutorial iremos avaliar os resultados de três programas tradicionalmente utilizados para produzir alinhamentos (Clustalw, Clustal-Ω, Muscle e MAFFT) e alguns dos componentes analíticos comuns a estes programas. Este tutorial requer a execução de uma série de exercícios e a avaliação dos resultados obtidos frente a leitura do artigo de Phillips *et al.* (2000). Ao completar este tutorial, o estudante terá uma visão geral dos algoritmos e estratégias de alinhamento múltiplo disponíveis para análises filogenéticas dentro do conceito de homologia estática, bem como dos problemas associados a esta prática em inferência filogenética. Os arquivos associados a este tutorial estão disponíveis no [GitHub](#). Você baixar todos os tutoriais com o seguinte comando:

```
svn checkout https://github.com/fplmarques/cladistica/trunk/tutorials/
```

8.1 Contextualização

A maioria das análises filogenéticas convencionais baseadas em dados moleculares requer um esquema de correspondência entre os pares de base que antecede à buscas de topologias – ou seja, a análise filogenética propriamente dita [1]. Isso decorre da observação de que, em muitos casos, regiões homólogas do genoma diferem em tamanho em consequência da composição diferencial de pares de base. O processo de conversão entre sequências de tamanhos distintos em sequências de mesmo tamanho é chamado de *alinhamento* ou ainda *alinhamento múltiplo*. Considere por exemplo a seguinte observação:

```
taxon1      TTGCA
taxon2      TGGCCA
taxon3      TGCAA
taxon4      TGGCCA
```

Um possível esquema de correspondência (*i.e.*, alinhamento) entre os pares de bases das sequências nucleotídica para estes quatro terminais seria:

Alinhamento 1

```
taxon1      TTG--CA
taxon2      TGGCCA
taxon3      TG--CAA
taxon4      TGG-CTA
```

ou poderia ser:

Alinhamento 2

```
taxon1      T-TG-CA
taxon2      TGGCCA
taxon3      TG--CAA
taxon4      T-GGCTA
```

Observe que estes dois alinhamentos propõem diferentes séries de transformações para os caracteres 2 e 3, portanto assumem hipóteses de identidades históricas [senso 2] distintas. Em consequência disso, estes dois alinhamentos resultam em hipóteses filogenéticas distintas (Figura 8.1). A influência dos alinhamentos em hipóteses filogenéticas é conhecido há tempos. Morrison & Ellis [3] examinaram esta relação e concluíram que o resultado de análises filogenéticas são mais influenciadas por alinhamentos do que critérios de otimalidade.

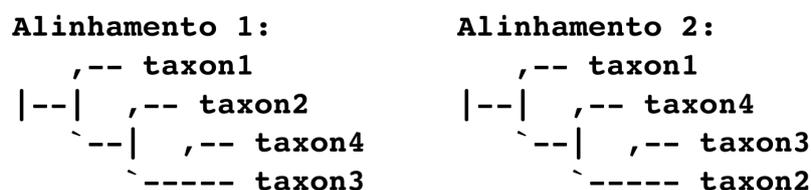


Figura 8.1: Hipóteses filogenéticas mais parcimoniosas resultantes dos alinhamentos 1 e 2, acima.

Um dos conceitos mais importantes relacionados aos procedimentos de alinhamento é compreendê-lo como um processo de inferência similar às estimativas de hipóteses filogenéticas. Pares de bases (*i.e.*, A, C, G e T) são observações, mesmo que indiretamente apresentadas pelas plataformas de sequenciamento. Os *gaps*, por outro lado, são inseridos durante o processo de alinhamento para traçar as correspondências necessárias – dentro do contexto de homologia estática [4] – para proceder com a análise filogenética. Igualmente importante é reconhecer que a inserção de *gaps* postulam eventos de inserção e deleções (*i.e.*, **INDELS**), eventos históricos que consequentemente possuem informação filogenética. Isso é relevante, pois requer que *gaps* sejam tratados como um quinto estado de caráter em análises filogenéticas de sequências nucleotídicas.

Neste tutorial iremos explorar os principais conceitos associados ao procedimento de alinhamento de sequências nucleotídicas. O tutorial seguirá um formato distinto do que vocês estão acostumados até então. Durante o tutorial vocês farão uma série de exercícios cujo objetivo é apresentar a vocês algumas das ferramentas disponíveis para alinhamentos múltiplos, entender os componentes metodológicos associados aos alinhamentos e verificar como as premissas de alinhamento impactam os resultados de análises filogenéticas. No final deste tutorial você encontrará uma série de perguntas que deverão ser respondidas para a próxima aula em referências ao artigo de Phillips *et al.* [5].

8.2 Alinhamento, aplicativo e topologias

8.2.1 ALINHAMENTO MANUAL NO ALIVIEW

Não é incomum encontrarmos publicações nas quais o alinhamento foi efetuado integralmente ou parcialmente à mão. Os chamados alinhamentos “*by eye*” são menos frequentes atualmente, isso é fato, mas seus defensores argumentavam que nosso cérebro era capaz de tomar decisões mais prudentes que os algoritmos disponíveis para essa tarefa. Há uma série de problemas com essa prática que iremos discutir na próxima aula [veja 5].

Exercício 8.1

Você deverá fazer o alinhamento manual das sequências no arquivo `seqdata1.fas` utilizando AliView. Em AliView, para inserir um gap em determinada região, basta posicionar o cursor na região desejada e pressionar a tecla de espaço. Note que o AliView tende a inserir gaps no final da sequência que deverão ser removidos ao final de seu alinhamento manual. É importante que você tente fazer o melhor alinhamento que considerar possível – embora devemos reconhecer que a noção de “melhor” é extremamente subjetiva –, mas gaste alguns bons minutos ponderando as decisões que o levaram a inserir um *gap* em determinadas regiões. Após finalizar este alinhamento você deverá salvá-lo sob o nome de “`seqdata1_man.fas`”.

8.2.2 CLUSTALW

Um dos aplicativos mais utilizados para alinhamento múltiplo é o Clustalw [6]. A documentação do programa pode ser encontrada em <http://www.clustal.org/clustal2/>. Não é o objetivo deste tutorial fazer qualquer explanação sobre os algoritmos e estratégias utilizadas pelo Clustalw, mas em Phillips *et al.* [5] você encontrará parte dessas informações. O Clustalw está disponível no sistema de vocês. Caso esteja usando uma outra instalação ou sistema operacional, certifique-se que o programa está instalado.

Exercício 8.2

Neste exercício você deverá fazer um alinhamento das sequências do arquivo `seqdata1.fas` utilizando Clustalw. Para fazê-lo, você deverá executar o Clustalw em um terminal utilizando a seguinte linha de comando:

```
$ clustalw -align -output=fasta -outorder=input -infile=seqdata1.fas
-outfile=seqdata1_cltw.fas
```

Esta linha de comando, a opção “-align” faz com que Clustalw execute o alinhamento múltiplo para as sequências existentes em “-infile=seqdata1.fas” obedecendo a ordem dos terminais (“-outorder=input”) quando imprimir os resultados no arquivo de saída “-outfile=seqdata1_clt.fas” no formato “-output=fasta”.

8.2.3 CLUSTAL-Ω

Recentemente, Sievers *et al.* [7] desenvolveram uma nova versão de Clustal. Chamada de **Clustal-Omega** (Ω), esta versão foi considerada mais rápida e “acurada” que Clustalw. Dada à sua recente publicação, esta versão não é tão popular como a anterior, mas pode ser que isso mude ao longo dos próximos anos.

Exercício 8.3

Neste exercício você deverá fazer um alinhamento das sequências do arquivo `seqdata1.fas` utilizando Clustal- Ω . Para fazê-lo, você deverá executar o Clustal- Ω em um terminal utilizando a seguinte linha de comando:

```
$ clustalo --full --iter=16 -i seqdata1.fas -o seqdata1_clto.fas -v
```

A descrição destas opções pode ser verificada digitando “`clustalo --help`” em um terminal ou consultando a documentação do programa ([Documentação](#)).

8.2.4 MUSCLE

Muscle [8] é outro aplicativo utilizado para alinhamentos múltiplos. Em termos gerais, esse

programa é bem mais rápido que Clustalw, embora menos utilizado na literatura. A forma de alinhamento de Muscle difere razoavelmente de Clustalw. Uma breve explicação sobre estas diferenças é apresentada no [item 4.1](#) da documentação de Muscle e não será discutida aqui.

Exercício 8.4

Neste exercício você irá alinhar as mesmas sequências que alinhou com o Clustalw e Clustal-Ω utilizando Muscle. Você irá executar o Muscle utilizando seus parâmetros padrão com a seguinte linha de comando:

```
$ muscle -in seqdata1.fas -out seqdata1_mus.fas
```

Na linha de comando acima, o arquivo de entrada (`-in seqdata1.fas`) será alinhado no Muscle e o resultado impresso no arquivo de saída (`-out seqdata1_mus.fas`). No entanto, no arquivo `-out seqdata1_mus.fas` a ordem dos terminais está alterada (veja utilizando o comando “`less`”). Há uma opção em Muscle para manter a ordem dos terminais idêntica ao arquivo de entrada, porém esta opção (`-stable`) está com problemas e foi desabilitada nas versões mais recentes do programa. No entanto, o autor disponibilizou um *script* que reordena o alinhamento segundo o arquivo de entrada. Para fazer o mesmo com seus resultados, execute o seguinte comando de linha:

```
$ python ./stable.py seqdata1.fas seqdata1_mus.fas > seqdata1_mus_ord.fas
```

8.2.5 MAFFT

MAFFT [9, 10] é o último programa que iremos utilizar. Este é um dos programas mais recentes desenvolvidos e possui uma série de algoritmos muito eficientes para gerar alinhamentos. Detalhes sobre tais algoritmos podem ser encontrados nas referências citadas anteriormente ou na página do programa referente aos [algoritmos de alinhamento](#).

Exercício 8.5

Neste exercício, você deverá fazer um novo alinhamento das sequências no arquivo “`seqdata1.fas`”. Para isso, você deverá executar o MAFFT com a seguinte linha de comando:

```
$ mafft --maxiterate 1000 --preserve-case seqdata1.fas > seqdata1_mafft.fas
```

Na linha de comando acima, o termo “`--maxiterate 1000`” configura o número máximo de iterações que o MAFFT irá usar durante o processo de refinamento do alinhamento das sequências no arquivo de entrada “`seqdata1.fas`” cujo resultado será impresso no arquivo de saída “`seqdata1_mafft.fas`” preservando as letras maiúsculas dos pares de base de acordo com a

opção “--preservecase”¹.

8.2.6 AVALIAÇÃO DOS RESULTADOS

Ao completarem os exercícios acima, seu diretório de trabalho deverá conter os seguintes arquivos: `seqdata1_man.fas`, `seqdata1_cltw.fas`, `seqdata1_clto.fas`, `seqdata1_mus_ord.fas` e `seqdata1_mafft.fas`. Verifique se todos estão presentes e que tenham conteúdo, caso contrário será necessário repetir alguns dos exercícios acima.

Exercício 8.6

Sua primeira avaliação será puramente visual. Abra todos os arquivos simultaneamente em AliView e responda:

i. Todos os alinhamentos são idênticos?

ii. Se você tivesse que optar por um desses alinhamentos, qual seria? Justifique sua resposta.

iii. Abaixo você deverá fazer uma análise filogenética em TNT de todos os alinhamentos gerados, completar a Tabela 8.1 e anotar o relacionamento filogenético encontrado em cada uma das análises nos espaços específicos abaixo. Para transformar esses arquivos no formato FASTA em formato compatível com arquivos de entrada no TNT (`xread`) você poderá usar o SequenceMatrix (veja Tutorial 7) ou você poderá usar o *script* `fasta2tnt.py` com a seguinte linha de comando (exemplo):

```
$ ./fasta2tnt.py seqdata1_mafft.fas > seqdata1_mafft.tnt
```

Tabela 8.1: Análise cladística dos alinhamentos gerados por diferentes programas.

Alinhamento	Número de MPTs	Custo das MPTs
manual		
Clustalw		
Clustal-Ω		
Muscle		
MAFFT		

¹ na configuração padrão do MAFFT ele imprime os pares de base em letras minúsculas. Embora para efeitos analíticos isso não importe, letras maiúsculas são desejáveis quando se deseja inspecionar os alinhamentos em programas como AliView.

Topologias encontradas nas análises filogenéticas dos alinhamentos da Tabela 8.1:

8.3 Parâmetros de alinhamento

Todos os alinhamentos produzidos pelos programas acima baseiam-se em alinhamentos par a par computados pelo algoritmo de Needleman & Wunsch [11]. Este algoritmo está documentado no artigo de Phillips *et al.* [5] e depende de parâmetros numéricos que são utilizados no cômputo das distâncias de edição entre duas sequências alinhadas. A distância de edição entre duas sequências é definida pelo número de operações necessárias para tornar uma sequência idêntica a outra. Por exemplo, considere o seguinte alinhamento:

```
taxon1      TATG--A
             * * * *
taxon2      TGGGCCA
```

Observe que são necessárias 4 operações para tornar uma das sequências igual a outra (denotadas com ”*”). Por exemplo, se considerarmos a sequência do `taxon1` e substituirmos **A** por **G** na segunda posição, **T** por **G** na terceira posição e – por **C** na quinta e sexta posições, as sequências de ambos os terminais tornam-se idênticas. Neste caso, se o custo de cada evento de edição é atribuído a 1, o custo deste alinhamento seria 4.

Parâmetros de alinhamento podem considerar custos distintos para eventos de edição diferentes – na expectativa de acomodar algumas premissas associadas ao que sabemos sobre evolução molecular. No exemplo acima, a edição necessária na segunda posição envolve um evento de transição (*i.e.*, substituições entre purinas ou pirimidinas), ao passo que na terceira posição o evento de edição envolve uma transversão (*i.e.*, substituições de purina por uma pirimidina, ou vice versa). Em casos como esse, você poderia atribuir custos diferenciais para esses dois tipos de operações de edição. Você poderia ainda levar em consideração que o custo de um *gap* é diferente de dois *gaps* consecutivos. A justificativa para estes custos diferenciais associados à abertura e extensão de *gaps* está no argumento de que INDELs (*i.e.*, inserções e deleções) podem ocorrer em bloco [mas veja 5, 12].

O cálculo do custo de um alinhamento é necessário por que a função objetiva que cada programa está buscando otimizar depende dele. Todos os programas exemplificados acima tentam maximizar a semelhança entre as sequências alinhadas e, portanto, sua função objetiva minimiza a distância de edição entre elas. Considere o seguinte alinhamento:

```
taxon1      TATG----A
             * * * * *
taxon2      T--GGGCCA
```

Neste caso, o custo do alinhamento seria 6, maior que o anterior. Mas isso só é verdade se assumirmos que qualquer operação de edição possui o mesmo custo, no caso 1. No entanto, considere custos diferenciais para abertura e extensão de *gap*, no qual o primeiro *gap* recebe valor igual a 1 e os demais 0.5. Neste caso ambos alinhamentos teriam custo igual a 3,5. Adicionalmente, consideremos custos diferenciais para transições e transversões, por exemplo

1 e 2, respectivamente. Neste caso, o primeiro exemplo teria custo igual a 4,5 ao passo que o segundo permaneceria com custo igual a 3,5 – pois nenhuma operação de edição neste último alinhamento requer substituição entre pares de base. O que fizemos acima foi variar os chamados **parâmetros de custo** de alinhamentos, e como se observa, eles influenciam o resultado da função objetiva e portanto pode selecionar alinhamentos distintos.

8.3.1 MANIPULAÇÃO DE PARÂMETROS EM MAFFT

Para verificar as relações entre parâmetros de alinhamento e inferência filogenética iremos manipular dois parâmetros disponíveis em MAFFT para 4 alinhamentos distintos e posteriormente fazer uma análise cladística destes alinhamentos.

Exercício 8.7

O parâmetro “`--op`” define o que é conhecido como *gap open penalty* em programas de alinhamento e o parâmetro “`--ep`” define o custo de extensão de *gaps*. Em MAFFT, esses parâmetros recebem os valores de 1.53 e 0.123, respectivamente. Obedecendo a sintaxe das linhas de comando abaixo, você deverá produzir 4 alinhamentos distintos, sendo que para um deles será considerado os valores padrão para estes parâmetros. Para os demais serão atribuídos custos diferentes, porém idênticos para “`--op`” e “`--ep`”. Os comandos de linha para a execução destes alinhamentos são as seguintes:

```
$ mafft --maxiterate 1000 --preserve-case seqdata2.fas > seqdata2.d.fas
$ mafft --op 1 --ep 1 --maxiterate 1000 --preserve-case seqdata2.fas > seqdata2.1.fas
$ mafft --op 2 --ep 2 --maxiterate 1000 --preserve-case seqdata2.fas > seqdata2.2.fas
$ mafft --op 3 --ep 3 --maxiterate 1000 --preserve-case seqdata2.fas > seqdata2.3.fas
```

Após gerar estes alinhamentos, você deverá fazer uma análise filogenética destes dados alinhados e anotar os resultados na Tabela 8.2 e no campo disponível para ilustrar as topologias obtidas, abaixo

Tabela 8.2: Efeito de parâmetros de alinhamento em inferência filogenética para as sequências do arquivo `seqdata2.fas`.

Alinhamento	Número de MPTs	Custo das MPTs
<i>default</i>		
<code>--op 1</code>		
<code>--op 2</code>		
<code>--op 3</code>		

Topologias encontradas nas análises filogenéticas sob diferentes parâmetros de alinhamento da Tabela 8.2:

<i>default</i>	<code>--op 1</code>
<code>--op 2</code>	<code>--op 3</code>

8.4 Alinhamento progressivo e árvores-guias

O algoritmo de Needleman-Wunsch [11], utilizado pela maioria dos aplicativos de alinhamento múltiplo, foi concebido para alinhar pares de sequências. Desta forma, ele é bidimensional. Por esta razão, para que os chamados alinhamentos múltiplos sejam executados pelo alinhamento par a par das sequências envolvidas é necessário definir quais seriam os pares alinhados para que ao final tivéssemos um alinhamento completo de todas as sequências de interesse. Todos

os programas selecionam pares, sejam eles de táxons terminais (OTUs) ou de vértices internos (HTUs) de acordo com o que denominamos de árvore-guia (*i.e.*, *guide trees*). Estas árvores-guias são geradas, via de regra, na fase inicial do alinhamento [consulte 5] e determinam a ordem progressiva dos pares que serão submetidos ao algoritmo de Needleman-Wunsch. O resultado final é o chamado “alinhamento múltiplo”.

8.4.1 EXAMINANDO A INFLUÊNCIA DE DIFERENTES ÁRVORES-GUIAS EM ALINHAMENTOS MÚLTIPLOS

Neste componente do tutorial iremos examinar como árvores-guias influenciam os alinhamentos. Iremos considerar quatro árvores-guias para as sequências do arquivo `seqdata2.fas`² (Figura 8.2).

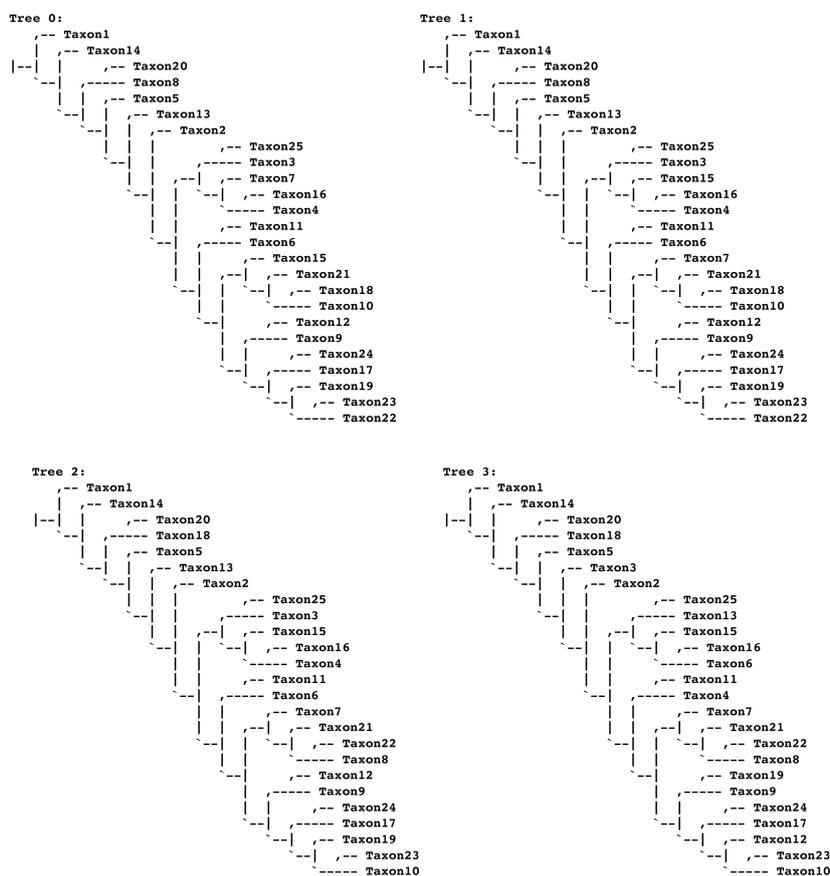


Figura 8.2: Quatro árvores-guias utilizadas no Exercício 8.8 para verificar o efeito dessas topologias nos alinhamentos múltiplos. Estas topologias estão nos arquivos `new_guide_*.tre` e foram transformados para o formato compatível com MAFFT nos arquivos `new_guide_tree*.mafft`.

² os comandos de linha para obtenção e controle da árvores-guias em MAFFT estão disponíveis em <http://mafft.cbrc.jp/alignment/software/treein.html> e <http://mafft.cbrc.jp/alignment/software/treeout.html>.

Exercício 8.8

Você deverá executar as linhas de comando abaixo, cada uma delas irá alinhar as sequências do arquivo `seqdata2.fas` considerando cada uma das árvores-guias disponíveis nos arquivos `new_guide_tree*.mafft`:

```
$ mafft --maxiterate 1000 --preservecase seqdata2.fas > seqdata2.d.fas
$ mafft --op 1 --ep 1 --treein new_guide_tree0.mafft --maxiterate 1000 --preservecase
seqdata2.fas > seqdata2.1.0.fas
$ mafft --op 1 --ep 1 --treein new_guide_tree1.mafft --maxiterate 1000 --preservecase
seqdata2.fas > seqdata2.1.1.fas
$ mafft --op 1 --ep 1 --treein new_guide_tree2.mafft --maxiterate 1000 --preservecase
seqdata2.fas > seqdata2.1.2.fas
$ mafft --op 1 --ep 1 --treein new_guide_tree3.mafft --maxiterate 1000 --preservecase
seqdata2.fas > seqdata2.1.3.fas
```

Após gerar estes alinhamentos, você deverá fazer uma análise filogenética destes dados alinhados e anotar os resultados na Tabela 8.3 e no campo disponível para ilustrar as topologias obtidas, abaixo.

Tabela 8.3: Efeito de árvore-guia em inferência filogenética para as sequências do arquivo `seqdata2.fas`.

Árvore-guia	Número de MPTs	Custo das MPTs
<code>new_guide_0.tre</code>		
<code>new_guide_1.tre</code>		
<code>new_guide_2.tre</code>		
<code>new_guide_3.tre</code>		

Topologias encontradas nas análises filogenéticas utilizando diferentes árvores-guias:

new_guide_0.tre	new_guide_1.tre
new_guide_2.tre	new_guide_3.tre

8.5 Consistência interna da análise

As análises executadas até o momento sofrem de pelo menos duas inconsistências analíticas severas. A primeira delas está relacionada ao fato de que os algoritmos utilizados para criar as árvores-guias possuem uma função objetiva completamente diferente daquela que você utilizou nas análises filogenéticas. Todas as árvores-guias geradas pelos programas de alinhamento múltiplo que utilizamos até o momento são geradas por algoritmos fenéticos, para os quais a função objetiva minimiza a distância fenética entre as sequências. Desta forma, árvores-

guias em Clustalw, Clustal-Ω, Muscle e MAFFT são fenogramas. No entanto, ao analisarmos os alinhamentos em TNT, nós abandonamos esse critério de otimalidade (distância fenética) e adotamos outro (distância patrística de série de transformações por parcimônia)! Não seria prudente epistemologicamente manter a consistência da função objetiva, e conseqüentemente critério de otimalidade, ao longo de toda a análise? A segunda inconsistência analítica refere-se ao fato de que os parâmetros dos parâmetros de custos de alinhamento, escolhidos de arbitrariamente – diga-se de passagem –, foram ignorados durante a etapa de inferência filogenética. Desta forma, você utilizou uma série de premissas para gerar os dados que foram desconsideradas no momento em que você utilizou estes mesmos dados para inferir relações de parentesco. O MAFFT, por exemplo, utiliza uma razão de 2:1 para transversões e transições que não foi considerada quando você fez uma análise cladística daquele alinhamento. O mesmo pode ser dito para todas as análises do Exercício 8.7. Em todos esses casos, as premissas analíticas não foram consistentes ao longo de todas as etapas de sua análise.

8.5.1 CONSISTÊNCIA DE PREMISSAS DE ALINHAMENTO

Quando se adota programas como Clustalw, Clustal-Ω, Muscle e MAFFT não é possível manter consistência dos critérios de otimização – assumindo que sua análise filogenética não utilize métodos fenéticos. Se seu critério de otimalidade em inferência filogenética é parcimônia, é necessário utilizar programas como, por exemplo, MALIGN [13] para manter a consistência interna da análise. Seja como for, o que iremos fazer a seguir é minimizar estas inconsistências analíticas implementando as premissas de custo de alinhamento em análises filogenéticas.

Para manter as premissas de alinhamento durante a análise filogenética é preciso implementar as mesmas funções de custo atribuídas para os abertura e extensão de *gaps* e também custos diferenciais de transformação durante o alinhamento na análise filogenética. MAFFT atribui penalidades distintas para transversões e transições na razão de 2:1, respectivamente. Nas análises do Exercício 8.7, nós atribuímos custos diferentes dos parâmetros padrão do programa para os *gaps*. A implementação dessas premissas em TNT é feita por matrizes de Sankoff. Consulte a seção 6.1 do Tutorial 6 para relembrar como caracteres de Sankoff são implementados em TNT.

Exercício 8.9

Sua tarefa é implementar as premissas de alinhamento na reanálise das sequências dos arquivos `seqdata2.2.fas` e `seqdata2.3.fas` produzidas no Exercício 8.7 deste tutorial. Os resultados dessas análises devem ser compilados na Tabela 8.4 e nos campos abaixo.

Tabela 8.4: Implementação dos parâmetros de alinhamento em análises filogenéticas.

Dados	Número de MPTs	Custo das MPTs
seqdata2.2.fas		
seqdata2.3.fas		

Topologias encontradas nas análises filogenéticas para as quais você implementou as premissas de alinhamento durante a análise filogenética:

seqdata2.2.fas	seqdata2.3.fas
----------------	----------------

8.6 Questões

1. Você tem alguma objeção ao uso de alinhamentos manuais? Comente.

2. Os resultados dos exercícios da seção 8.2 lhe permitem identificar algum critério objetivo que possa ser usado para a seleção de algum dos alinhamentos que você produziu? Justifique.

3. Parâmetros de alinhamento influenciam reconstruções filogenéticas? Comente de acordo com os resultados que você obteve nos exercícios acima.

4. Você considera que esses parâmetros de alinhamento são arbitrários? Justifique.

5. Árvores-guias influenciam o resultado final do alinhamento? Comente de acordo com os resultados que você obteve nos exercícios acima..

6. Dentre as árvores-guias exploradas no Exercício 8.8, alguma(s) dela(s) gerou(am) um alinhamento que você consideraria melhor do que aqueles obtidos até o momento para as sequências do arquivo seqdata2.fas? Justifique.

7. A topologia da árvores-guias nos exercícios que você fez diferem daquelas obtidas pelas análises filogenéticas das sequências alinhadas? Por que você acha que isso ocorre?

8. A implementação dos premissas de custo de alinhamento durante a análise filogenética modificou os resultados que você havia obtido até então para esses dados? Comente de acordo com os resultados que você obteve nos exercícios acima.

8.7 Leitura recomendada para discutir seus resultados

Phillips, A. *et al.* 2000. Multiple sequence alignments in phylogenetic Anlalysis. *Molecular Phylogenetics and Evolution* **16**(3): 317–330

8.8 Referências

1. Wheeler, W. C. 2012. Systematics: a course of lectures. Malaysia: Wiley-Backwell, 2012. 426.
2. Grant, T. & Kluge, A. G. 2004. Transformation series as an idiographic character concept. *Cladistics* **20**: 23–31.
3. Morrison, D. A. & Ellis, J. T. 1997. Effects of nucleotide sequence alignment on phylogenetic estimation: a case study of 18S rDNAs of Apicomplexa. *Molecular Biology and Evolution* **14**(4): 428–441.
4. Wheeler, W. C. 2001. Homology and the optimization os DNA sequence data. *Cladistics* **17**: S3–S11.

5. Phillips, A.; Janes, D. & Wheeler, W. C. 2000. Multiple sequence alignments in phylogenetic Anlalysis. *Molecular Phylogenetics and Evolution* **16**(3): 317–330.
6. Larkin, M. A. *et al.* 2008. ClustalW and ClustalX, version 2. *Bioinformatics* **23**(21): 2947–2948.
7. Sievers, F. *et al.* 2011. Fast, scalable generation of high-quality protein multiple sequence alignment using Clustal Omega. *Molecular Systems Biology* **7**: 539.
8. Edgar, R. C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* **32**(5): 1792–1797.
9. Katoh, K.; Misawa, K.; Kuma, K. & Miyata, T. 2004. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research* **30**(14): 3059–3066.
10. Katoh, K. & Standley, D. M. 2013. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in performance and usability. *Molecular Biology and Evolution* **30**(4): 771–780.
11. Needleman, S. B. & Wunsch, C. D. 1970. A general method applicable to the search of similitaries in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**: 443–453.
12. Giribet, G. & Wheeler, W. C. 1999. On gaps. *Molecular Phylogenetics and Evolution* **13**(1): 132–143.
13. Wheeler, W. C. & Gladstein, D. L. 1994. MALIGN, version 1.93. New York, NY: American Museum of Natural History, 1994.