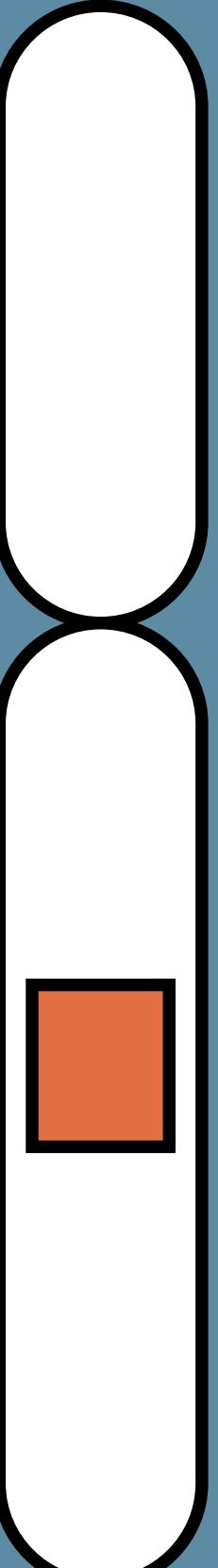


# CODEML WORKSHOP

Parte prática #2



# TÓPICOS:

- Conhecendo o codeml (Exercício 1)
- Modelando e analisando dados com codeml  
(Exercícios 3 - 5)
- Mergulhando mais fundo (Exercícios 6 - 8)

# CONHECENDO CODEML

## EXERCÍCIO 1

### Passo-a-passo do que vamos fazer:

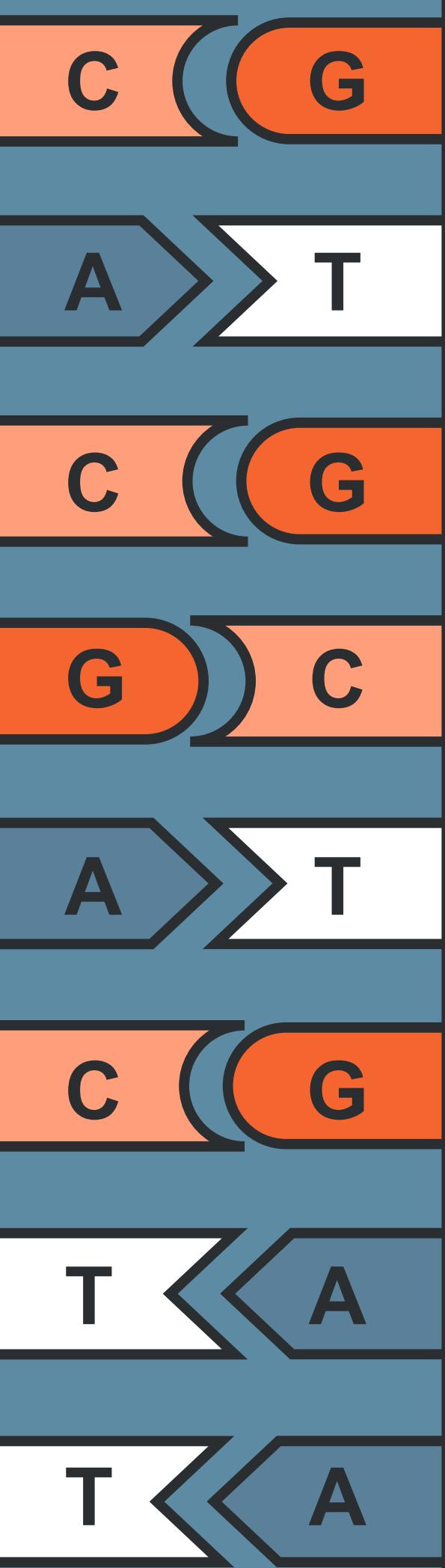
- Baixar as sequências
- Rodar o modelo one-model
- Observar o arquivo de resultados e recuperar as seguintes informações:
  - lnL, df
  - omega (total e de cada ramo)
  - comprimentos de ramos da árvore



# EXERCÍCIO 1

## BAIXANDO AS SEQUÊNCIAS

- Acesse o [repositório](#) no GitHub
- Baixe o arquivo [data1.unaln.fasta](#)
- Abra o arquivo no seu visualizador de texto  
(notepad++, sublime, bloco de notas)
- Como é o formato fasta?



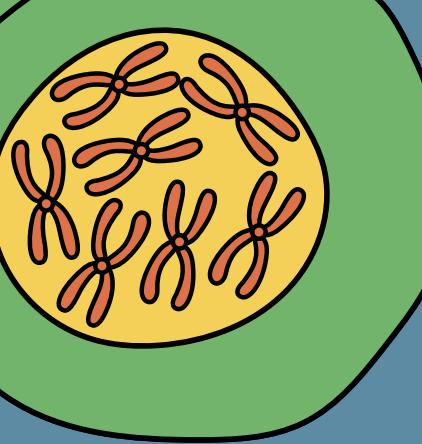
# O ARQUIVO FASTA

## ESTRUTURA

```
dataset_pratica >

1 >Chicken
2 ATGAAACAATCCATTGGTCAACCTCAGCTCAGCTTTGGATGCCATTAGTCAGATCCCAGAACAGAGAAATAGTAATGTACCACTTCTTA
GCAGAAAAGTGTCAGAGGCTGAATGACAGAGAACAGGAGATGAGGCGAGCAGCATGCAGCTGGACAACCAATATGACAGAAA
GGAAAAGCTGTCTTGAAGCTTGTCTGGTGTCTCTCTAGGGACAAGGTGTCTACTCGCTGTCTCTGGAACTTAAAC
AAAGCCAAGATATACTGGTAGCATTAGTAGGAGAACAATTTCCCTGAAATCTGTCTCTGGACGTCAGCTCCAGAC
CTGAAAGAGACAATCATTGGTAGTGTACATGTAACGTGGATATTGCAACAACAGAGGCGCTGGAAATGGCTCAAGAGGTGG
CACTCAGAAAAGTTATATGATTGAAATGTTATGGCAAATGGACTTCTGCAACGAAATTGTCTTACCTCCCAATCCAGCAAG
GGACGTATTATTAAAACCTTGCCTGCAATAGAGAAGCAACTGACTGCAACAAGCAAAGAACAGAACCTTAAAGTACACA
CTGGTTGGAAACGAAATCAGACTGTTCCAAAATCCGAGAGAGTTCGGACATGGGAGTAAAGCTCTGGAGAGCTCTGCCA
TAAAAGAGCAAATTACGAAGCTGGAGGAGCCAGCTGTCGATGCTGAACAAAGTGTATATGGTGAAGAGAAAATTTTGAGC
ATCCTGACTCAATTAAAATGGAGAGAATTATACGCAAGGATAACATCTACGCAAGATGATTAAAAGCTGCCAGGGCAGAAGG
CACTGGAGCAAGTAAACGCCTGAGCAATCAGATTCTGATCATCCTCTACTGCTTCTGACTTTGGAAATTATTCAGACAG
CAGCAGAAGTTACTGACCAGCAGAAATTAGTCACCTCAACAAAGCTACCAATACCTGGTAGACTTTAACTCTGTAG
3 >Chimpanzee
4 ATGGTTGTTCCGAAGTGGACATCGCAAAGCTGATCCAGCTGCTGCATCCCACCCCTTACTGAATGGAGATGCTAATGTGGC
TGTGGAGCAGGACCTGGCCCTGCCAGCCATGCCGTATGGGGACAGAGCTGGCAAGAGCTGTGTTGGAGGACTGTCAGG
ACCAGGACTACGAGAAATGAGATTTCGATGCTCAGAGGTAGAAAAGGAATTAAAGCCAGATGCCATGCCGGGGAGGAAT
AATCAGCCTGCTGACATTGGTATAAGATCAAGACACTCATCAAGAAGTACATCCAGAGGCGAGGACGATCAGCCTGGTGTGTC
GGTGGACAAAGGAACGTAGAGACAAGGTTGGAGCTGGTACCGAACCTCTGTTGAGAAGAGGCTACATGATCGTCAAGT
AGGAAGGAAAGGCCACGTTCTGCCCTGAGAAACCTTACAGCAGGCTCATCACACACATCTGTAATCTGCCCTGTTAG
GATAAAGTTAATGCTTAACTCAGGACATCTCTGCTCATGCAAGGAGAGGAACCTGTAGGGGAGGAAGACATTGGCTGTTAC
GTATCGTGGTAGAGAGCTGGCAGGCTTGTAATTACAGGACATTTGAGACAATCGTAAACAGCAAATCAAGGGCTGGAAAGAGC
CCAAGTCCAAAATTGAAGACATTAGAGCAGAACAGAGAGAAGGTGAGAAGCTGATCCGCTGACTTCCAGATGGAACAGATTG
CAGTCAGCTGGCAACAGACCTTCCATGGAGGAGATTTCAAGCACCTGATGGCTATCACAGGAGGCCAGAAGCGCATCTCC
CTGGCTCTGAAGGAGCGGAGGACACAGCGAACCGGAAGTCTGAAGGAGGGCTGCACGGCTGACGAGGCTCGGCCG
5 >Cow
6 ATGGTTCATTCTGACTTGGGTATCGAAGAACCTGATTCCCTGAATCCAGTCTAAATGGAAAGTGAAGATATGGAGTCAAGAGAAC
TGTATCGGGGACCAGAGCTGGCAAGAGCTGTGCTGGAGGCCCTGTCTGGGTGCCCTCCCCAGGGGAGTGGTATTGTTAC
CACAGGTGAAAAGGAATCAGTGAAGCCCAGATTGCCATCGCTGGGGAGGCACGGGATCAGTCATGAGCTGATTAGTCTGGAGG
TCTCTCATCAGGAAGTATCTTAGGCAGGAGACCATCAACTGGTGGTGTCCCTGCTAACGTGGACATCGCTACACAGAGGCG
CGTGGTGAGAACCTGGTTTCCACCTGAAAGAGGGTACATGATCGTCAAGTGGCTGGCAGCAGGACATCAAGCATCGGATGAG
AAAGACTGACCACTGAACTCATGACACATTGAAACTCTGCCCTGTTGAAAATCAAATAAAGGAGACTCACAGAGAAATAA
TCAACAATAGAAGGGAGGAATTGTTGAGCAGTATGACTCCGACTGTTACCAAAAGTACGAGGCCAGTTCTCAAATGGAGTGT
TTACAAGACGTTGAGGACATCATAAAAGCAGGTAGAGTTCTGGAAGAGGCCCTGTTGAGCATGTCACACGGTGACAGATAT
AAGAAAATGAAGCTGAGAAGTCCATCGACTACATTCCAATGGAGCAGTTGGTCTACTGCCAGGACAGGTGACCGCGTGTCA
CAACACCTGACCGCGTACCGAGGAAAGTCAGCACCCGATCTCGGCCACATCCCCGATCATCAGTTCTGTGCTCCGGACG
GAGCAGCTAAGAAGAGCATGTCAGCTGCTCAGGACAAGGACAGTACGACTGGCTGCTGAAGGAGCGCACCGACACAGAGAC
7 >Dog
8 ATGGTTAATTACAAGGAAAAACTGACTCGAATCTGTACCAATCATGTGTTACTAAACGGACTTACTGATAAGGAGAACAG
CCTGGCCCTGGCCATTGCTGTCATTGGGGACAGAGCTGGCAAGAGCTGTGCTGGAGGCTGTCAAGGAGTTGCCCTCC
AGATGGAGATTTCAGACCCCTCAGAGGTGGAAGTGGAAATCAATAAGCCAGGATGCCATTGGGGAGGACAGGGGATCAGTC
GACATCGGAGGCCAGAACACTCATTAGGAAGTACATCTTAAAGCAGGAGACATCAACTGGTGGTGTCCCTGCAACGTG
GACTGAAGGCAAGGTGGAGCTGGCACAAAACCTGCTGTCACCTGAAGAAGGGTACATGATCGTCAAGTGGCCGGGAGCAG
CCACAGTCCCCACCTGGGGAGAACACTGACCTCTGAGCTCATCACGACATCTGTAACCCCTGCTGTAGAAAATCAGATAA
GCCTTAACCAAGACATCAGCTCTTAATCCAAGGGAGGAGTCCGTGGGGAGGAGCAGAGGCCAGCTGGCTGTGACCCATC
AGAGCTGCGAGGATTGTAATTACAAGACGTTGAAATTATAAAACAAACAAATCAAAGAACACTGGAGAGCAGCTGTTGATAT
TCGAAGACATTAATTGAGCTAGAGAAGAGCTGAGAAGTCCATCGACTTCACTTCAAATGGAGCAGATCGTACTGCCAGG
GTCAACATCTCTGTCTGAATCTGGAGCACCTGTTGCCCTACCGCCAGGAGGCCACCAACGCACTCCAGGCCACATCCCCG
GCGAGTGCACCGAGCAGAGGAAAGTCCGTGAAGGAGCGCTGGCGGGCTGGCCAGGCTGGCGCCGTTAGCCAATCCCC
```

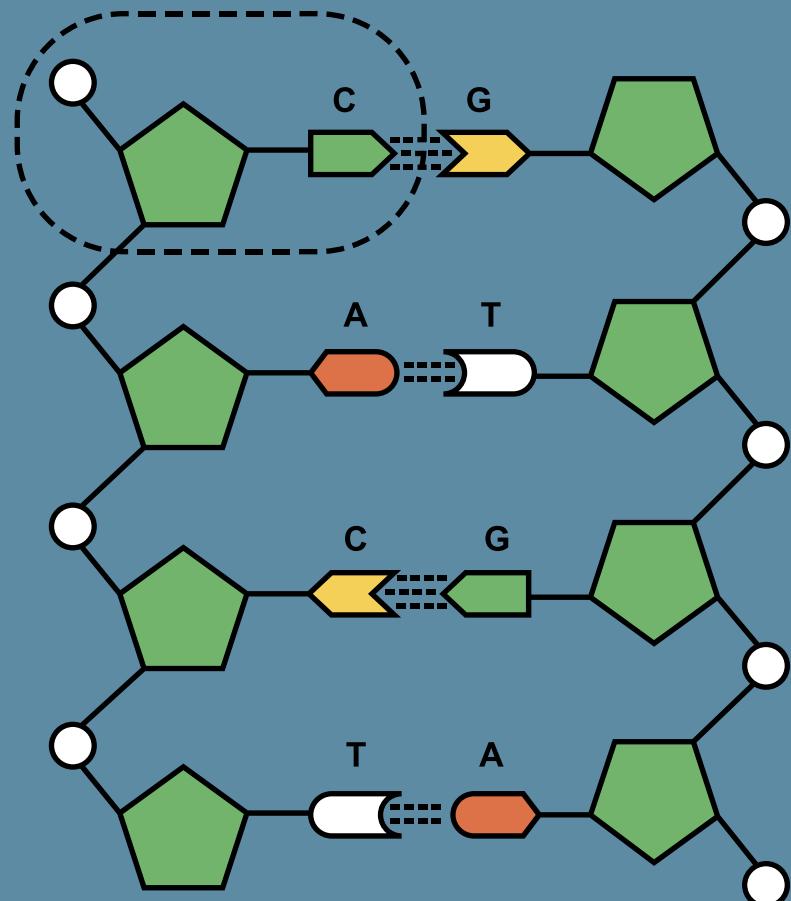
- Nome da sequência começando com >
- Quebra de linha
- Sequência



# EXERCÍCIO 1

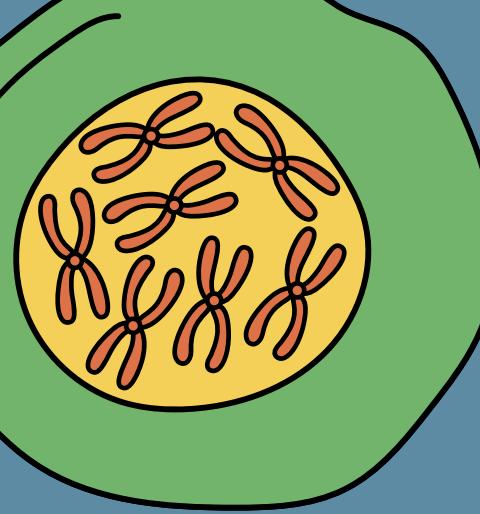
## ALINHANDO AS SEQUÊNCIAS

- Abra o arquivo [data1\\_unaln.fasta](#) no Aliview
- Você vê algum padrão? É fácil ou difícil entender o que está acontecendo?



O que muda se a gente alinhar essa sequência?





# EXERCÍCIO 1

## ALINHANDO AS SEQUÊNCIAS

- Acesse o MAFFT
- Insira seu arquivo fasta de sequências
- Vá até o fim da página e clique em submit

Multiple sequence alignment and NJ / UPGMA phylogeny

**Input:**  
Paste protein or DNA sequences in fasta format. [Example](#)

```
>Chicken
ATGAACAATCCATGGTCCAACCTCAGCTCAGCTTTGGATGTCCCATTAGATCCAAAGCAGAATAG
>Chimpanzee
ATGGTTGTTCCGAAGTGGACATCGAAAAGCTGATCCAGCTGCTGCATCCCACCCCTTATTACTGAA
>Cow
ATGGTTCATCTGACTTGGGTATCGAAGAACTTGATTCACCTGAATCCAGTCTAAATGGAAGTGAAGA
>Dog
ATGGTTAACCAAGGAAAAACTGACTCGAATCCTGTACCAATCATGTGTTACTAACGGACT
>Duck
ATGACTACTCAGCGTAACACAGACAAGCCACATAGCAAGCCAGAAGACCAGTGGAACATGTACAACAG
>Human
ATGGTTGTTCCGAAGTGGACATCGAAAAGCTGATCCAGCTGCTGCATCCCACCCCTTATTACTGAA
>Mouse
ATGGATTCTGTGAATAATCTGTGCAGGCCTATGAGGAGAAGGTGCGGCCCTGTATTGACCTCATCGA
>Orangutan
```

or upload a plain text file:  No file chosen

Use [DASH](#) to add homologous structures (protein only)

Output original plus DASH sequences    Output original sequences only

Give structural alignment(s) externally prepared

Allow unusual symbols (Selenocysteine "U", Inosine "i", non-alphabetical characters, etc.) [Help](#)

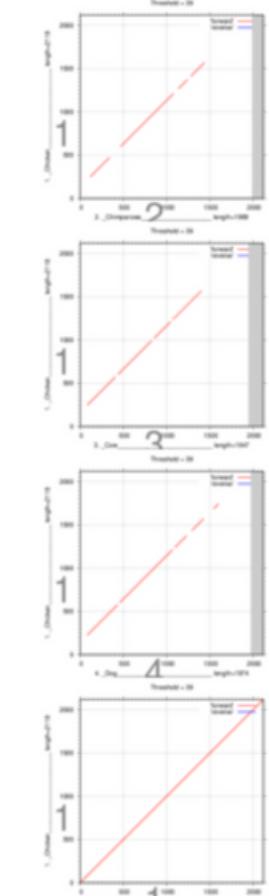
# EXERCÍCIO 1

## ALINHANDO AS SEQUÊNCIAS

- Clique em **Fasta format**, no topo da página, para baixar os resultados em formato FASTA
- Abra o arquivo no Aliview

LAST hits (score>39) between the top sequence and the others.

[Open all plots](#)



Be careful if there are **blue lines**. By default, MAFFT considers similarities in forward strands (**red**) only, but ignores similarities in reverse strands (**blue**). If **blue lines** are seen around diagonal regions in the plots above, try the "Adjust" button.

[Clustal format](#) | [Fasta format](#) | [MAFFT result](#) | [View](#) | [Tree](#) | [Refine dataset](#) | [Return to home](#)

[View](#)

[Reformat](#) to GCG, PHYLP, MSF, NEXUS, uppercase/lowercase, etc. with Readseq

[GUIDANCE2](#) computes the residue-wise confidence scores and extracts well-aligned residues.

[Refine dataset](#)

[Phylogenetic tree](#)

### MAFFT-L-INS-i Result

CLUSTAL format alignment by MAFFT (v7.511)

Chicken	atgaacaatcc-
Duck	atgactactcagcgtAACACAGACAAGCCACATAGCAAGCCAGAAGACCAGTGGAAACATG
Chimpanzee	atggtttttc-
Human	atggtttttc-
Orangutan	atggttcttc-
Rhesus_mac	atggttcttc-
Cow	atggttcattc-
Sheep	atggtttttc-
Pig	atggtttatttc-
Dog	atggtaatttc-
Mouse	-
Rat	atg-----

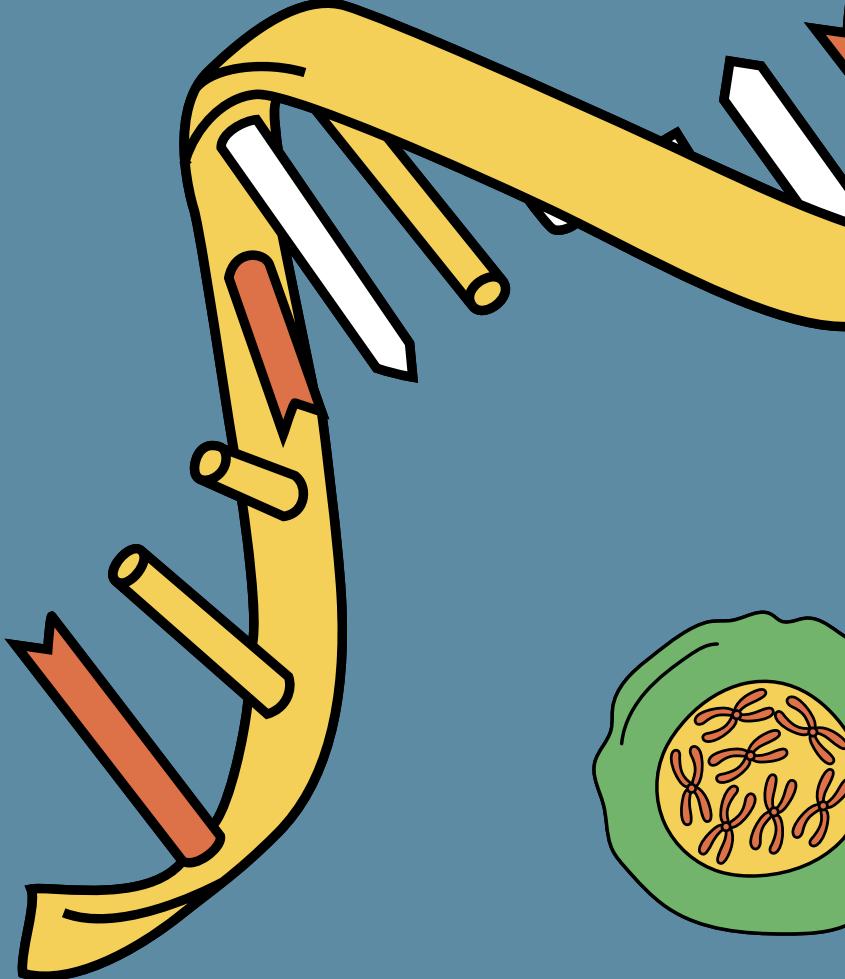
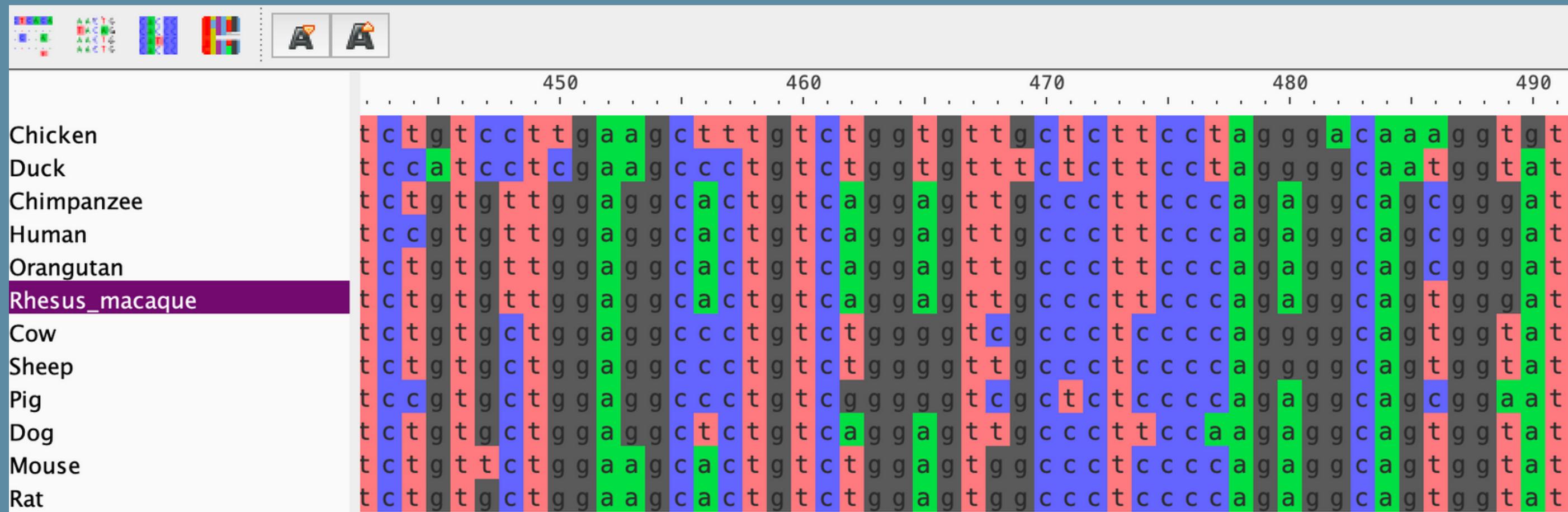
  

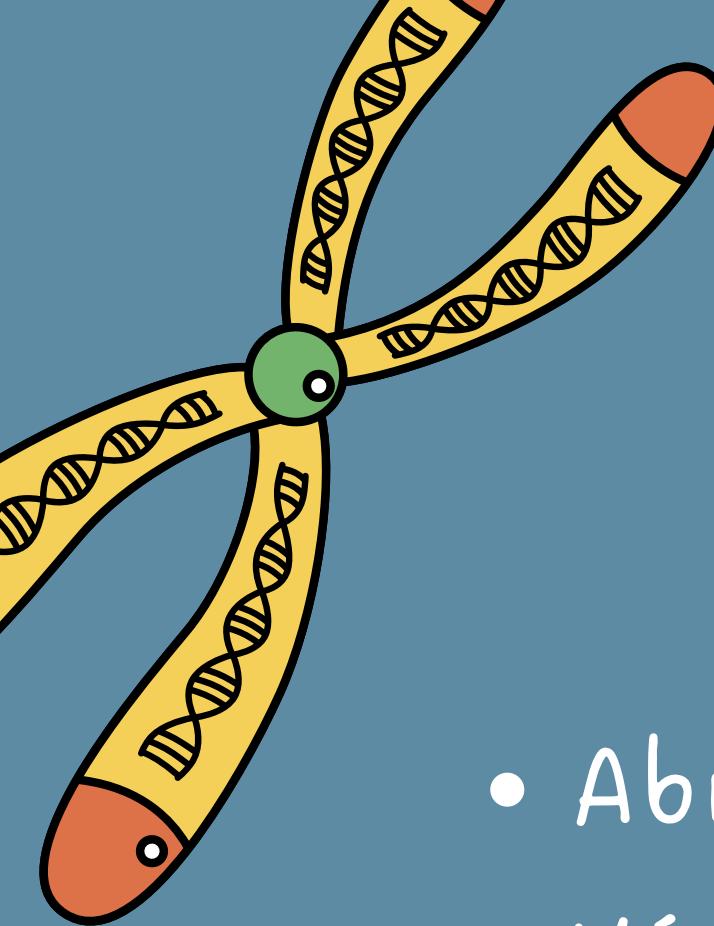
Chicken	-----atggtccaaccttcagctcagctttggatgtcccattcagatccccaaaggcag
Duck	tacaacagaaaatcccaagttcaaagcaacggccaaaagatgcagtccaaactgtgaaa
Chimpanzee	-----
Human	-----
Orangutan	-----
Rhesus_mac	-----
Cow	-----
Sheep	-----
Pig	-----
Dog	-----
Mouse	-----
Rat	-----

# EXERCÍCIO 1

## VISUALIZANDO O ALINHAMENTO

- Vamos explorar o Aliview!
- O que cada um dos botões no topo faz?





# EXERCÍCIO 1

## TRADUZINDO PARA PROTEÍNAS

- Abra o arquivo [data1\\_unaln.fasta](#) no aliview
- Vá na aba [View](#) e clique em [Show as translation](#)
- Vá em File e clique em [Save translated alignment as Fasta](#)

The screenshot shows the Aliview software interface with a multiple sequence alignment of protein translations from different species. The alignment is color-coded by amino acid type. The species listed on the left are: Chicken, Chimpanzee, Cow, Dog, Duck, Human, Mouse, Orangutan, Pig, Rat, Rhesus\_macaque, and Sheep. The alignment shows the first 100 amino acids for each species. The top of the window has a toolbar with various icons for file operations, sequence analysis, and alignment tools. The bottom right corner shows the reading frame settings: "Reading frame: 1. Standard code".

Species	Sequence (Amino Acids)
Chicken	MNNPWSNFSSAFGCPIQIPKQNSNVPPSLPVPVPGVFGVPLRSGCSNQMAFCAPELTDRKPEHEQKVSKRLNDREEDKDEAAACSLDNQYDRKIQPCIDL
Chimpanzee	MVVSEVDIAKADPAAAASHPLLNGDANVAQKNPGSVAENNLCSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Cow	MVHSDLGIEEELDSPESSSLNGSEDMEKSNLYSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Dog	MVNNSQGKITDSNPVPNHNLLNGLTDKAEKKNQGIGNSLCSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Duck	MTTQRNTDKPHSKPEDQWNMYNRNPKFATAKRCSPNLMKDGFQSLSSPVCIEASAVPLPPDSDDIEIYFPVPEQTTKESQHEQKVSMKLHEEQDVOAAE
Human	MVVSEVDIAKADPAAAASHPLLNGDATVAQKNPGSVAENNLCSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Mouse	MDSVNNLCRHYYEKVRPCIDLIDTLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Orangutan	MVLSSEVDIAKADPAAAASHPVLLNGDANVAQKNLGSVAENNLCSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Pig	MVYSSCESKEPDVSASNHLLLNGNDELVEKSHKTGPENNLYSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Rat	MKEQTSAKRHTPQKH PDTSEESQAMESVDNLCSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Rhesus_macaque	MVLSEVDIVKADPAAAASQPLLLNGDADVAQKSPGSVAENNLCSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS
Sheep	MVLSDLDIKEPDSPESGLNGSDDMVRHEHETESKGNLYSQYEKKVRPCIDLIDSLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGS



# EXERCÍCIO 1

## ALINHANDO AS PROTEÍNAS

- Acesse o [MAFFT](#)
- Insira seu arquivo  
fasta de proteínas
- Vá até o fim da  
página e clique em  
submit

Multiple sequence alignment and NJ / UPGMA phylogeny

**Input:**

Paste protein or DNA sequences in fasta format. [Example](#)

```
>Chicken
MNNPWSNFSSAFGCPIQIPKQNSNVPPSLPVPGVFGVPLRGCSNQMAFCAPELDRKPEHEQKVSK
>Chimpanzee
MVVSEVDIAKADPAAASHPLLNGDANVAQKNPGSVAENNLCSQYEEKVRPCIDLIDSLRALGVEQDL
>Cow
MVHSDLGIEELDSPESSENLNGSEDMESKSNLYSQYEEKVRPCIDLIDSLRSLGVEQDLALPAIAVIGDQ
>Dog
MVNSQGKITDSNPVPNHVLLNGLTDKAENQGIGNSLCSQYEEKVRPCIDLIDSLRALGVEQDLALPA
>Duck
MTTQRNTDKPHSKPEDQWNMYNRNPFKATAKRCSPNLMKDFQSLSSPVCIEASAVPLPPDSDEIY
>Human
MVVSEVDIAKADPAAASHPLLNGDATVAQKNPGSVAENNLCSQYEEKVRPCIDLIDSLRALGVEQDL
>Mouse
MDSVNNLCRHYEEKVRPCIDLIDTLRALGVEQDLALPAIAVIGDQSSGKSSVLEALSGVALPRGSGIV
>Orangutan
```

or upload a plain text file:  [Choose File](#) No file chosen

Use [DASH](#) to add homologous structures (protein only)

Output original plus DASH sequences  Output original sequences only

Give structural alignment(s) externally prepared

Allow unusual symbols (Selenocysteine "U", Inosine "i", non-alphabetical characters, etc.) [Help](#)



# EXERCÍCIO 1

## ALINHANDO AS PROTEÍNAS

- Clique em **Fasta format**, no topo da página, para baixar os resultados em formato FASTA
- Abra o arquivo no Aliview

Clustal format **Fasta format** MAFFT result | [View](#) | [Tree](#) | [Refine dataset](#) | [Return to home](#)

[View](#)

[Reformat](#) to GCG, PHYLIP, MSF, NEXUS, uppercase/lowercase, etc. with Readseq

[GUIDANCE2](#) computes the residue-wise confidence scores and extracts well-aligned residues.

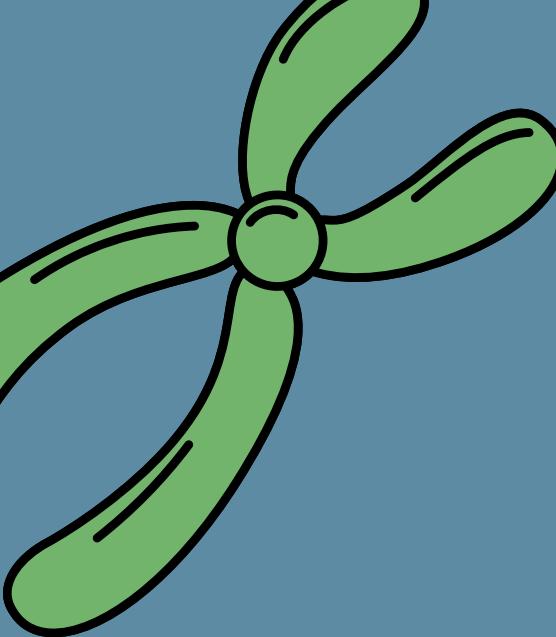
[Refine dataset](#)

[Phylogenetic tree](#)

### MAFFT-L-INS-i Result

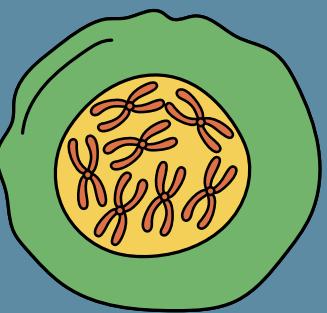
CLUSTAL format alignment by MAFFT (v7.511)

	Chicken	Duck	Chimpanzee	Human	Orangutan	Rhesus_mac	Cow	Sheep	Pig	Dog	Mouse	Rat
	M-----NNXXWSNFSSAFGCPIQIPKQNSNVPPSLPVPVGVFGVPL	MTTQRNTDKPHSKPEDQWNMYNRNPFKATAKRCSPNLMKDGFQSLSS-PVCIEASAVPL	MVVXXEVD-	MVVXXEVD-	MVLXXEVD-	MVLXXEVD-	MVHXXDLG-	MVLXXDLD-	MVYXXSCE-	MVNXXQGK-	M-----	M-----*



# EXERCÍCIO 1

## ALINHANDO CÓDONS NO PAL2NAL



- Acesse o site do [PAL2NAL](#)
- Insira seus alinhamentos de nucleotídeo e proteína
- Vá até a última opção, Output format, e selecione PAML
- Clique em Submit

[Run PAL2NAL \(try with an example\)](#)

INPUT FILE 1

Put your multiple sequence alignment of proteins (either in CLUSTAL or in FASTA format)

- The server automatically detects the file format.
- In the CLUSTAL format, you can select some specific positions by '#' under the alignment; see [the example](#).
- [How to specify stop codons and frame shifts in the alignment](#).

```
QDQVYKETLKTIREKEAKEKT-KALINPATFQNNSQFPQKGLTTTEMTOQHLKAYYQECKR  
RNIGRQIPLIIQYFILKTFGEEIEKMMQLLQDTSKCSWFLEEQSDFREKKFLKRRLLR  
LDEAQRLAKFSD-X  
>Rat  
M-----  
-----KEQTSACRHQTQKHPDTSEESQAMESVDNLCSQYEKVRPCIDL  
DSLRLALGVQEQLALPAIAVIGDQSSGKSSVLEALSGVALPRGSIVTRCPVLVLLKQLQ  
-GEKWSGKVVIYKDTEIEISHPSLVEREINKAQNLIAGEGLKISSDLISLEVSSPHVPDLT  
LIDLPGITRVAVGDQPADIEHKIKRLITEYIQKQBTINLVVVPNSNDIATTEALKMAQEVE  
DPQGDRTIGILTAKPDVLVDRGTEDKVVDVVRNLVCHLKKGYMIVKCRGQQDIQEQLSIAEA  
LQKEQVFFKEHPQFRVLLEDGKATVPCLAKRRTMELTSHICKSLPILENQINVNHQIASE  
ELQKYGADIPEDDSKRLSFLMNKINVFNKDILSLVQAQENISWEESRLFTKLNEFLAWN  
DYIEEHFKKTLSSEKHSQMEKFESHYRGRELPGFVDYKAFENIIKKEVKALEEPALNML  
HRVTTMVKNAPTKVSSNNFGDFLNLLHSTAKSKIEDIRFSQEKEAEKLIRLHFQMEHIVYC  
QDQAYKKALKEIREKEAKEKS----TGFVFQQN-XXPRKELTTTEMTOQHLNAYYQEKG  
RNIGRQIPLIIQYSILQTFGQEMEKAMLQLQDTSKCNWFLTEQSDSREKKFLKRRLLR  
LDEAQRLAKFSN-X
```

or

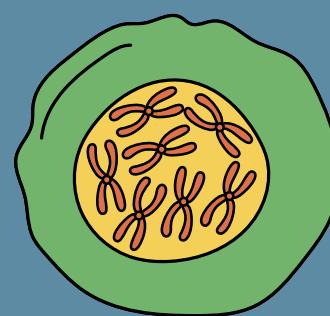
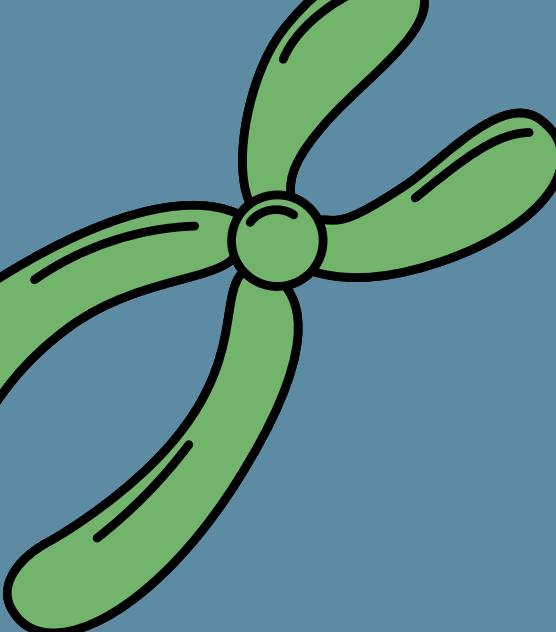
upload

Choose File No file chosen

INPUT FILE 2

Put your DNA (or mRNA) sequences (in FASTA format)

```
ggcagtggattgttaccagggtgcctctgggtggaaattgaagcaactgaagcag--  
ggagaaaaatggagccgcaaggtcattataaggacaccgagattgagatctcacaccct  
tcacttgttagaaggaaatcaataaagcccagaacttgcattgtggggaaagggttgaag  
attagtcgtatcttcattagcttgaggtagctccacatgtccccagacccgtactcg  
attgacccctgtgtatcacaaggatgtgggtgaccagcctgcagacatcgaacac  
aagatcaagagacttatcactgaaatccagaacacaggagaccatcaaccctgggtgg  
gtccccagcaatgtggcattgcacccacagaggccctgaaaatggctcaggaggtggac  
cctcaaggggatagaaccataggatctgaccaagcccgatctgtggacagaggactg
```

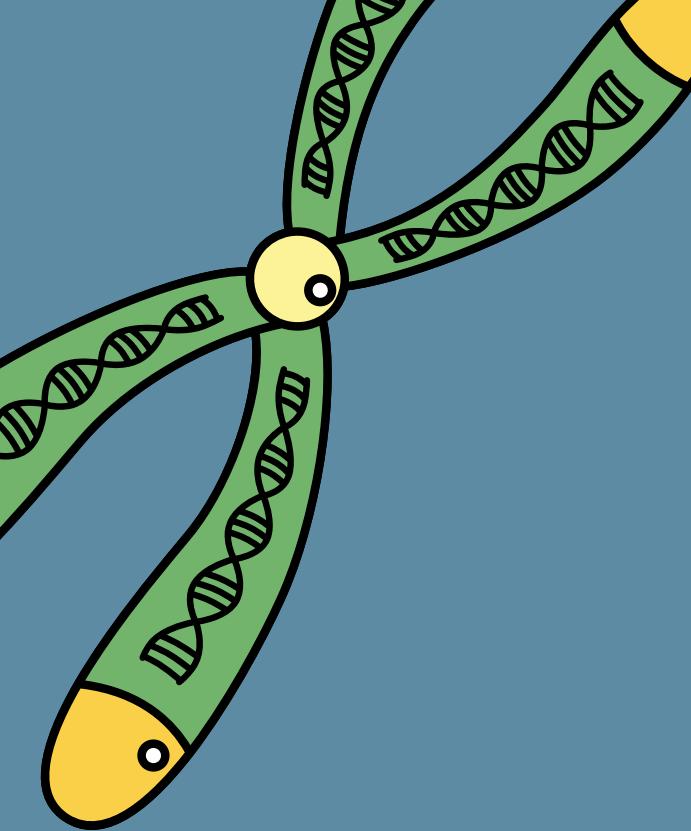


# EXERCÍCIO 1

- Salve o alinhamento PAL2NAL com a extensão .nuc (ou .phy, .txt, o que quiser)
  - A primeira linha deve conter o número de sequências (12) e de posições (2205) - pode apagar as anteriores!

PAL2NAL output

12 2205  
Chicken  
atg-----aacaatccatggtccaac  
ttc-----agctcaagctttggatgtcccattcaagatccaaag  
cagaatagaatgtaccaccccttcattaccagtacctgttaggagttttgggggcctctg  
cgatcaggctgcagcaatcagatggcttcgtgcaccagaactgactgacagaaaaggcct  
gagcatgagcagaaaagtgtccaagaggctgaatgacagagaagaggacaaggatgaggca  
gcagcatgcagcttgacaaccaatatgacagaaagatccaacccattgcattgatcttgatt  
gacagcctgagaaagcttgatataggaaacgacacctgtatgtgcctgcaattgcagtgtt  
ggagaccggaaactctggaaaagctgtccttgaagcttgcgtgtgccttcct  
aggacaaaagggtgtcattactcgctgccttgcacttaaactgaaaaaatgacagct  
---ccgcaggaatggaaagggttaatttattaccgcaacacagaaaatacagctccagaat  
gcatcagaggtgaagaaagcaataagaaaagcccaagatatagtggctggcactaatgg  
agcattagtggagaactaatttccttgcataatctggcttcgtacgtcccagacactgaca  
ctaattgatcttccttgcattgcagagaggccgtgggaaccagccacaagataatggc  
caacagatcaaaacactacttaaaaaatattggctgcaaaagagacaatcattgtggta  
gtgttaccatgttaacgtggatattgcaacaacagaggcgtgaaaatggctcaagaggtg  
gatcccacaggagaaaaggacgcttgggtcctcactaaaccagacactgtgaacgaagga  
actgaagagactgtcctaagataatacAAAacgaggcatcccactcagaaaaggttat  
atgattgtgaaatgttatggcaatggacttctgcaacgaattgtccttcaccccgca  
atccagcaagagagagaattcttgcagactcacaaacattcagcactctctggatgaa  
aataaggctactatccacatctggcaaataagcttacagatgaaacttgtggacgtt  
ataaaaacttgcctgcaatagagaagcaagtacatgatgcactgcaacaagcaaaagaag  
gaactccaaaagtacacacaaaacacacacccactgtcagcgtacgatgtt  
gtgggggtgatcaaagcgttaatgaaagacatctctcagacaatgcatggaaaggaatcc  
tggggaaaacgaaatcagactgtttccaaaatccgcagagatgttgcacatggga  
gtaaagctcctggagagactctgcaaa-----gttgaagaaatcgatgcagtaatt  
cccaaataatgaaagaccagtaccgcggacggagttccagactttcagactgt  
tttggagacattataaaagagcaattacgaaagactgtggaggagccagctgtgcgt  
aacaaaagtgtatgttgcagatggggacttgcagctggtaacaagcgttt  
aattttcagaacttaaaaacacgcgtctcaggccagaattgggtcattagt  
gacagacactgcggaaaattgcattctgactcaattttggagagaattataactgc  
caggataacatctacgcagatgatttaaaagactgcgcaggccagaaggcatc  
acaaaatcaaaagacactgttgcatttttttttttttttttttttttttttttttt  
tgtccagcttgcctggaaatgggttctcagtaaaggcctatttcactggagca  
aacgcctgagcaatcagattccctgtatcatccctctactgtccttcatgactt  
aatttttgcagacactcaatgttgcattcttgcagggaaaagaagaaaataactat  
ctccaggaagatcatgaaagactgcataaccaggcagaagttactgaccagg  
ctcaacaaaaggctaccaatacctggtagactttaagtctctgttag



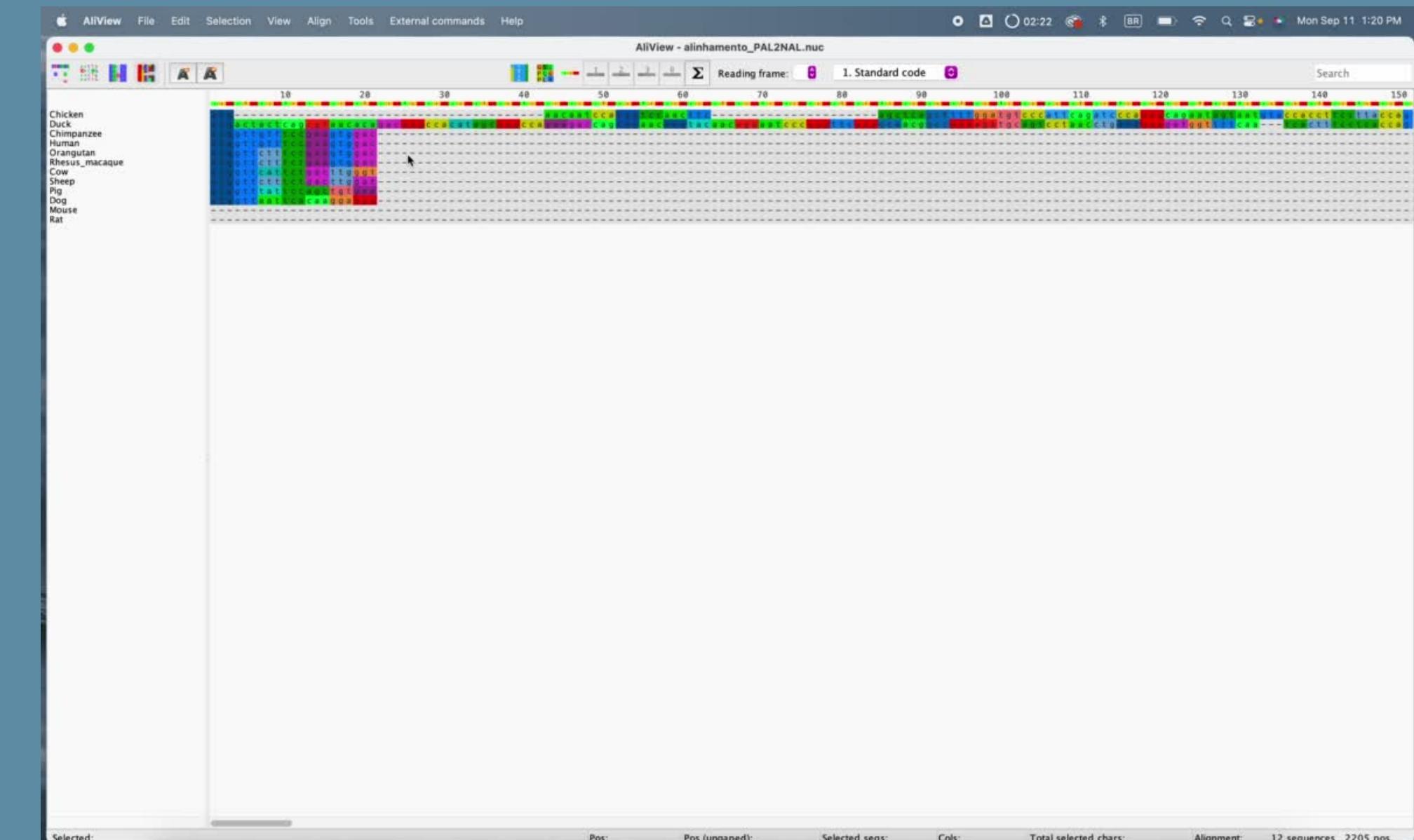
# EXERCÍCIO 1

## EDITANDO O ALINHAMENTO

Pra completar, precisamos fazer duas coisas:

- remover gaps
- remover stop codons

Por que esses passos são importantes?





# EXERCÍCIO 1

## VERIFICANDO O ALINHAMENTO

- Não colocar mais que 30 caracteres no nome
- Separar o nome e sequência por pelo menos 2 espaços

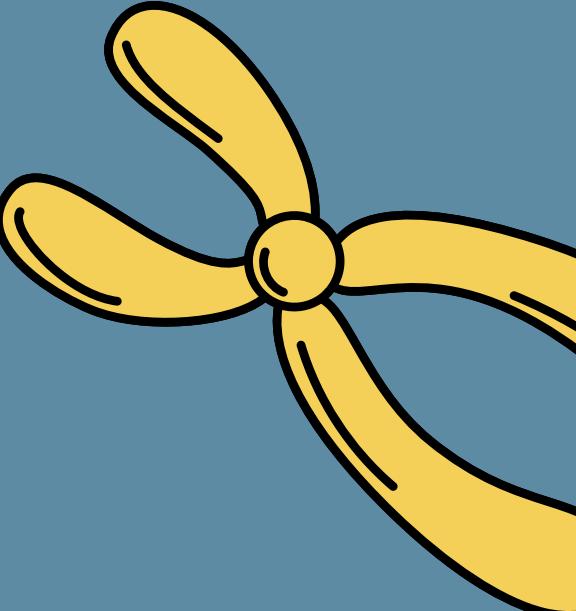
12/2007

Chicken  atg-----gcaccagaactgactgacaga  
agcctgagaaagcttcatataggaaacgacacctgatgtgcctgcatttcgttgg  
atggaaagggttaatttattaccgaacacagaaatacagctccagaatgcattcagagg  
ttgccatggccgtggggaccaggccacaagataatggccaacagatcaaaacacta  
gtcctcactaaaccagacactgtgaacgaaggaactgaagagactgtcctaagataata  
acatttcagcactttctggatgaaaataaggctactatcccacatctggcaaataagct  
ataagacgattttccttgtgggttcatcaagcgatcatggatgttttttttttttttttt  
agtaaattgccccaaatataaagaccgttaccgcggacggatgtcccaacttttttt  
taattttcagaactaaacaacgctgctcaggccagaattgggtgcatttagtgacagaca  
tcaaagaccttgct-----tttggatgtgcttca-----cgtcaatgtccca

Dica: use as ferramentas de contagem de caracteres

# EXERCÍCIO 1

## ESTIMANDO UMA ÁRVORE DE GENE



Acesse o [IQTree Webserver](#):

- Faça o upload do seu alinhamento de códon
- Selecione a opção códon e Standard/Universal

**Input Data**

**Alignment file :** C:\fakepath\alinhamento\_PAL2N/ [Browse...](#) [Show example >](#)

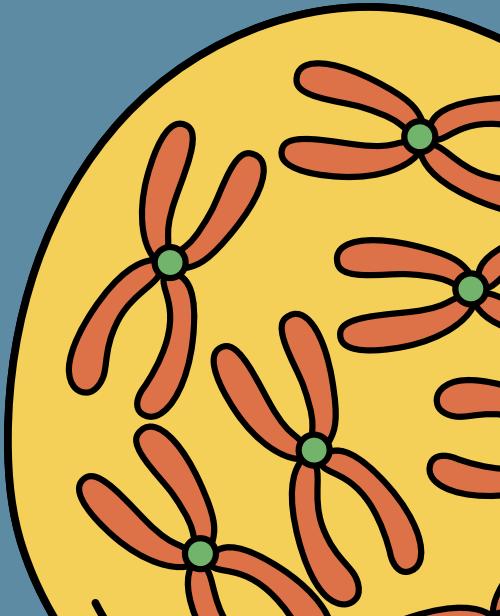
**Use example alignment:**  Yes [?](#)

**Sequence type:**  Auto-detect  DNA  Protein  Codon [?](#)  
 DNA->AA  Binary  Morphology [?](#)

**Genetic code:** 1: Standard/Universal [?](#)

**Partition file:** This field is optional. [Browse...](#) [Show example >](#)

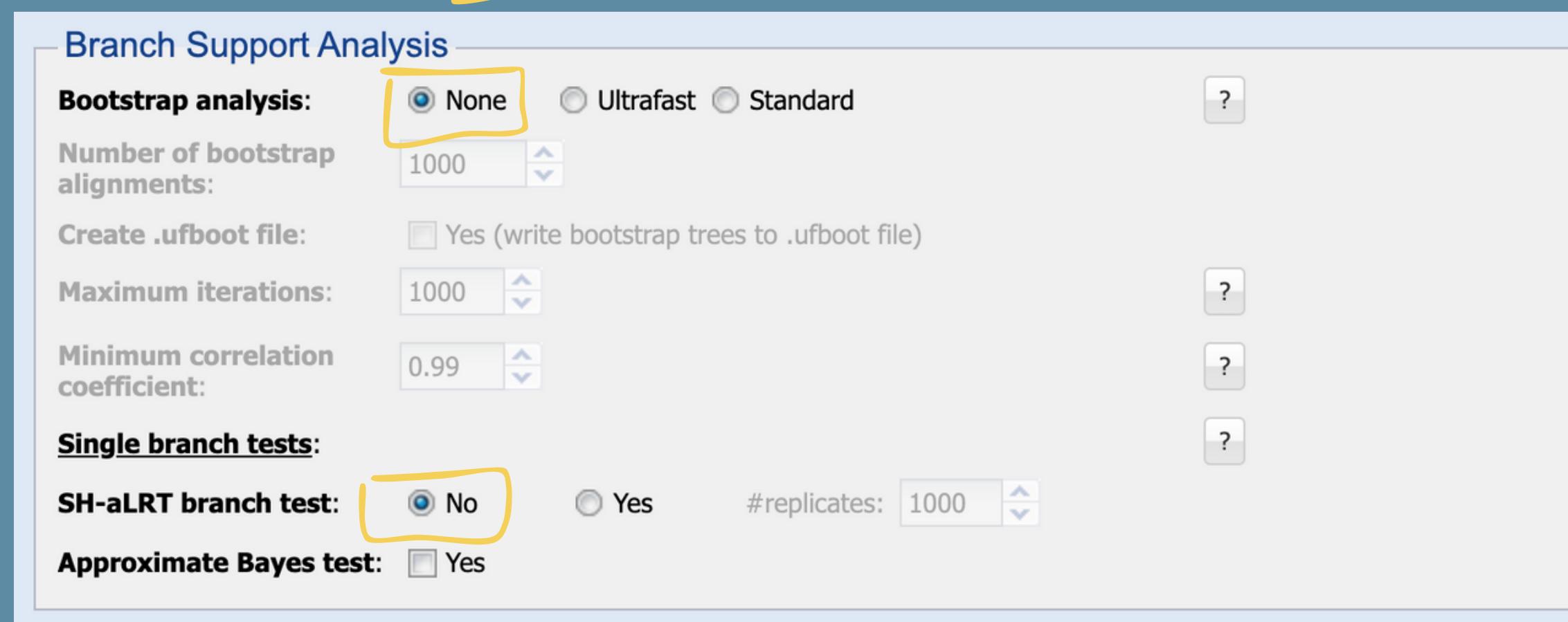
**Partition type:**  Edge-linked [?](#)  
 Edge-unlinked



# EXERCÍCIO 1

## ESTIMANDO UMA ÁRVORE DE GENE

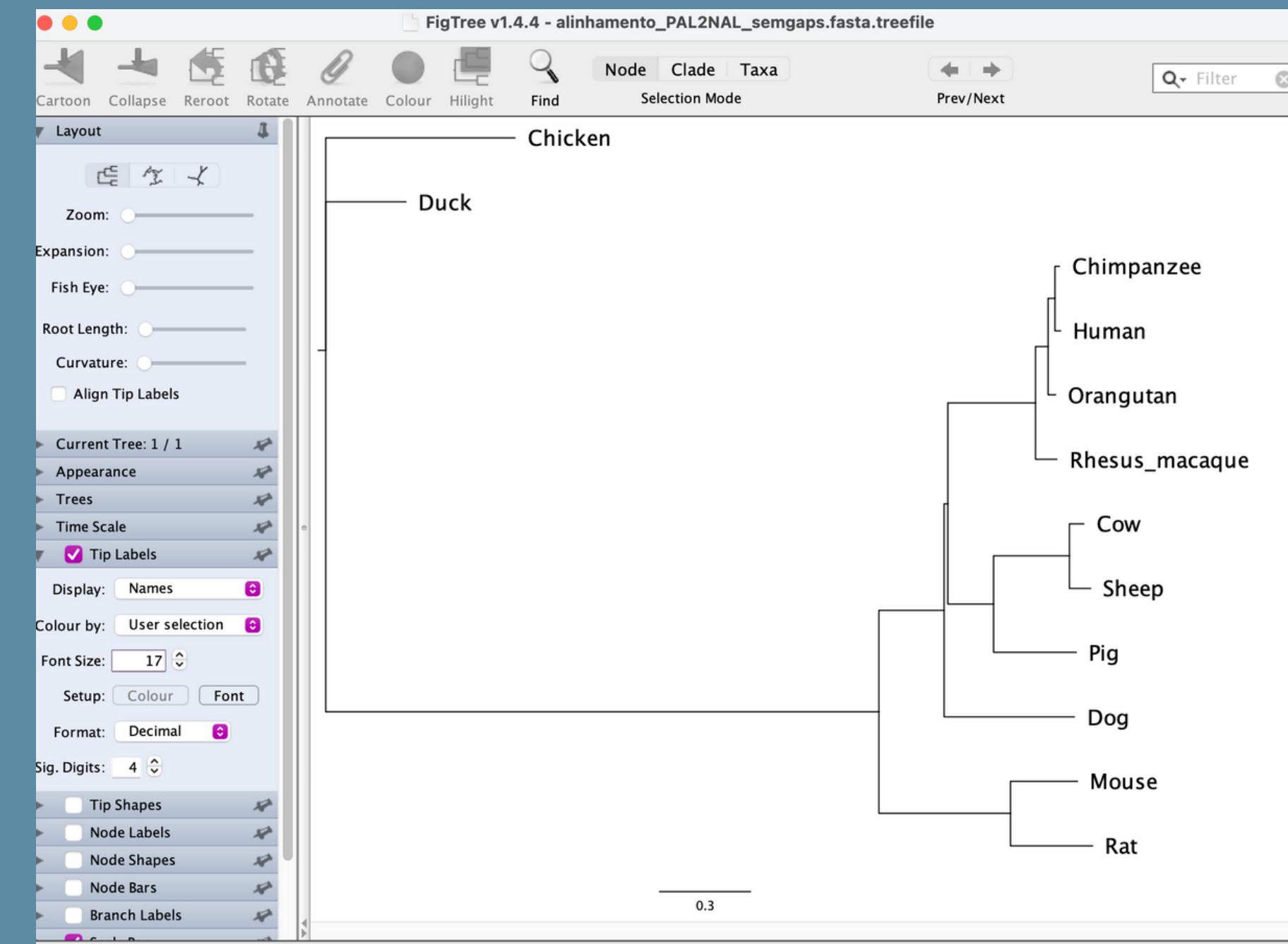
Não vamos precisar de comprimentos de ramos, então na parte de Branch support analysis, clique em "None" e "No":



# EXERCÍCIO 1

## ESTIMANDO UMA ÁRVORE DE GENE

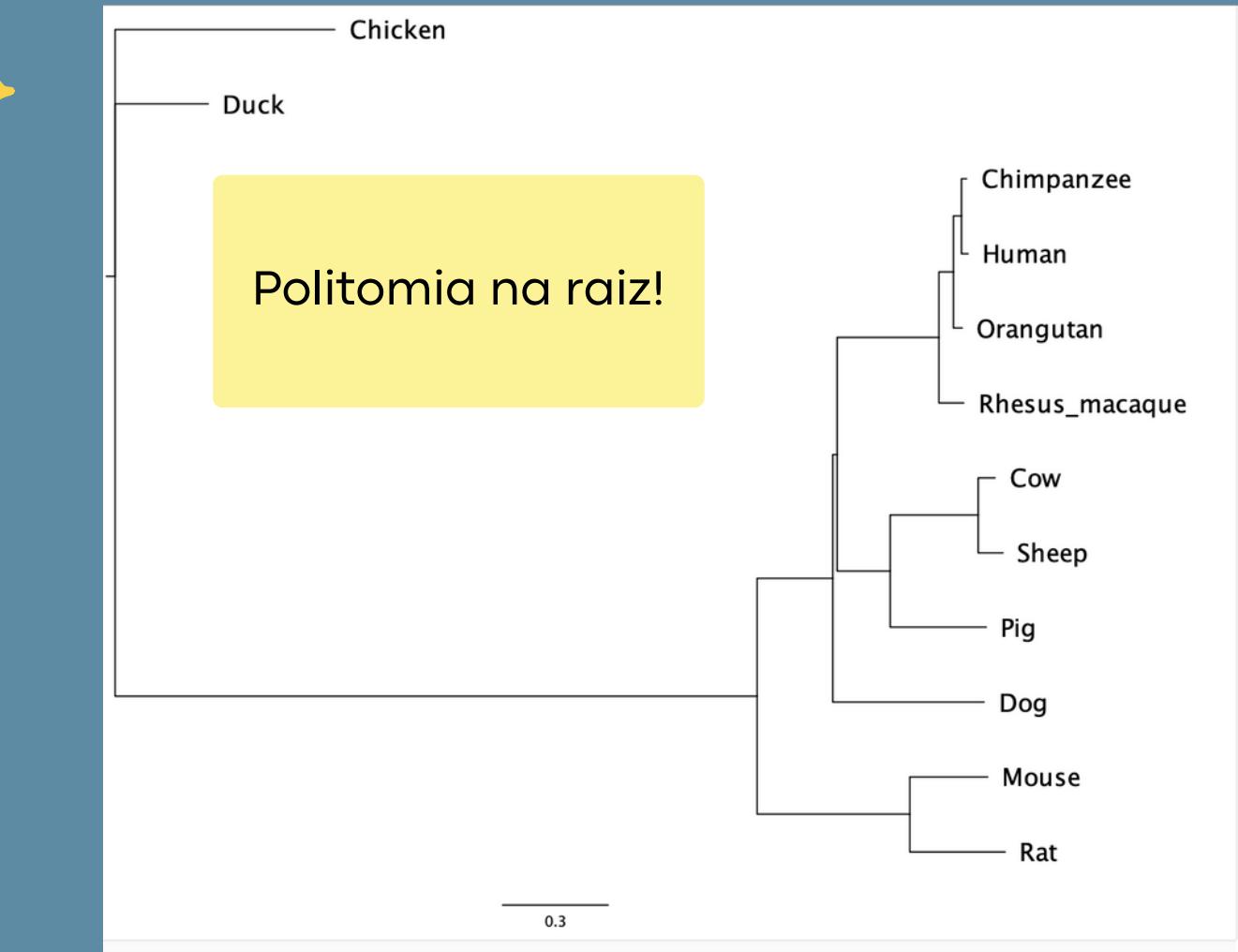
- Clique em Download selected jobs, no fim da página
- Na pasta de resultados do IQTree, localize o arquivo que termina com .treefile
- Abra esse arquivo no FigTree



# EXERCÍCIO 1

## EDITANDO A ÁRVORE DE GENE

- Cheque se sua árvore está:
  - desenraizada
  - sem comprimentos de ramos



```
(Chicken,Duck,((((Chimpanzee,Human),Orangutan),Rhesus_macaque),((Cow,Sheep),Pig)),Dog),(Mouse,Rat));
```

# EXERCÍCIO 1

## EDITANDO O ARQUIVO CONTROLE

```
seqfile = Mx_aln.phy      * Path to the alignment file
treefile = Mx_unroot.tree   * Path to the tree file
outfile = out_M0.txt        * Path to the output file
noisy = 3                   * Display moderate information on the screen
verbose = 1                 * Detailed output file
seqtype = 1                 * Codon data
Ndata = 1                   * One gene alignment
icode = 0                   * Universal genetic code
cleandata = 0               * Do not remove sites with ambiguity data
model = 0                   * One w for all branches (M0 and site models)
NSSites = 0                 * One w for all sites (M0 and branch model)
CodonFreq = 2                * Use F3x4 model
clock = 0                   * Assume no clock
fix_omega = 0                * Enables option to estimate omega
omega = 0.5                 * Initial omega value
```

Coloque aqui o nome do alinhamento de códon, árvore e arquivo de saída

Coloque aqui o modelo

# EXERCÍCIO 1

## RODANDO A ANÁLISE (AEEEE!)

- Conecte-se ao servidor!

```
ssh -p 22 awtempN@awarnach.mathstat.dal.ca
```

Senha:  
yn6bakxs

→ substitua N pelo seu número

- Depois vamos transferir o alinhamento de códons, a árvore e o arquivo controle para a pasta de análises

# EXERCÍCIO 1

## RODANDO A ANÁLISE (AEEEE!)

- Antes de rodar, precisamos ativar o ambiente onde o paml está instalado: `source activate paml`
- Confira se você está dentro da pasta contendo todos os seus arquivos, usando `ls`
- E agora sim, vamos rodar o codeml: `codeml arquivo_controle.ctl`

# EXERCÍCIO 1

## VISUALIZANDO OS RESULTADOS

- Na pasta onde você rodou as análises, localize o arquivo de resultados (qual nome você deu?)
- Abra esse arquivo usando `cat` ou `less`
  - Qual a diferença entre esses comandos?
- Vamos dar uma olhada geral!

# EXERCÍCIO 1

## VISUALIZANDO OS RESULTADOS

- Agora vamos localizar as seguintes informações:
  - $\ln L$ ,  $df$
  - omega (total e de cada ramo)
  - comprimentos de ramos da árvore
- E vamos colocar na planilha!

# EXERCÍCIO 1

## VISUALIZANDO OS RESULTADOS

$$\ln L = -12873.814018$$

23  
parâmetros  
livres

```
TREE # 1: (1, 2, (((((3, 4), 5), 6), ((7, 8), 9)), 10), (11, 12)); MP score: -1
lnL(ntime: 21 np: 23): -12873.814018 +0.000000
13..1 13..2 13..14 14..15 15..16 16..17 17..18 18..19 19..3 19..4 18..5 17..6
5..10 14..22 22..11 22..12
0.577417 0.287022 1.639958 0.231387 0.081203 0.289313 0.043540 0.021587 0.014533 0.019238 0.023881 0.067966
363933 0.437367 0.253505 0.226870 2.358349 0.331112

Note: Branch length is defined as number of nucleotide substitutions per codon (not per nucleotide site).

tree length = 5.38464

(1: 0.577417, 2: 0.287022, (((((3: 0.014533, 4: 0.019238): 0.021587, 5: 0.023881): 0.043540, 6: 0.067966):
9: 0.243382): 0.185313): 0.081203, 10: 0.363933): 0.231387, (11: 0.253505, 12: 0.226870): 0.437367): 1.639958

(Chicken: 0.577417, Duck: 0.287022, (((((Chimpanzee: 0.014533, Human: 0.019238): 0.021587, Orangutan: 0.023881,
Cow: 0.052746, Sheep: 0.065521): 0.258956, Pig: 0.243382): 0.185313): 0.081203, Dog: 0.363933): 0.231387
: 1.639958);

Detailed output identifying parameters

kappa (ts/tv) = 2.35835
omega (dN/dS) = 0.33111
```

↓  
ômega

árvore com comprimentos  
de ramo

# EXERCÍCIO 1

## EDITANDO O ARQUIVO CONTROLE

```
seqfile = Mx_aln.phy      * Path to the alignment file  
treefile = Mx_unroot.tree  * Path to the tree file  
outfile = out_M0.txt       * Path to the output file  
noisy = 3                  * Display moderate information on the screen  
verbose = 1                 * Detailed output file  
seqtype = 1                 * Codon data  
Ndata = 1                   * One gene alignment  
icode = 0                   * Universal genetic code  
cleandata = 0               * Do not remove sites with ambiguity data  
model = 0                   * One ω for all branches (M0 and site models)  
NSSites = 0                  * One ω for all sites (M0 and branch model)  
CodonFreq = 2                * Use F3x4 model  
clock = 0                    * Assume no clock  
fix_omega = 0                * Enables option to estimate omega  
omega = 0.5                  * Initial omega value
```

Coloque aqui o nome do arquivo de saída, para cada vez que você for rodar

Mude o valor inicial do ômega: 0.005, 0.01, 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 2.0

# ANALISANDO DADOS

## EXERCÍCIO 2

Passo-a-passo do que vamos fazer:

- Criar uma nova pasta com: alinhamento, arquivo controle e árvore (marcada com labels)
- Rodar o modelo two-model
- Observar o arquivo de resultados e recuperar: lnL, df, omega (total)
- Comparar one-model e two-model: Likelihood Ratio Test



# EXERCÍCIO 2

## MARCANDO RAMOS NA ÁRVORE

- Aqui vamos usar a árvore com os comprimentos de ramos que foram estimados pelo One Model
- Depois vamos marcar com labels os ramos das aves, para os quais um ômega diferente será estimado:

(Chicken #1: 0.577417, Duck #1: 0.287022, (((((Chimpanzee: 0.014533, Human: 0.019238): 0.021587, Orangutan: 0.023881): 0.043540, Rhesus\_macaque: 0.067966): 0.289313, ((Cow: 0.052746, Sheep: 0.065521): 0.258956, Pig: 0.243382): 0.185313): 0.081203, Dog: 0.363933): 0.231387, (Mouse: 0.253505, Rat: 0.226870): 0.437367): 1.639958);

Qual o objetivo desta marcação?

# EXERCÍCIO 2

## EDITANDO O ARQUIVO CONTROLE

```
seqfile = Mx_aln.phy      * Path to the alignment file  
treefile = Mx_unroot.tree   * Path to the tree file  
outfile = out_M0.txt        * Path to the output file  
noisy = 3                  * Display moderate information on the screen  
verbose = 1                 * Detailed output file  
seqtype = 1                 * Codon data  
Ndata = 1                   * One gene alignment  
icode = 0                   * Universal genetic code  
cleandata = 0                * Do not remove sites with ambiguity data  
model = 2                   * Two ω for different sets of branches  
NSSites = 0                  * One ω for all sites (M0 and branch model)  
CodonFreq = 2                 * Use F3x4 model  
clock = 0                    * Assume no clock  
fix_omega = 0                 * Enables option to estimate omega  
omega = 0.5                  * Initial omega value
```

Coloque aqui o nome do alinhamento de códon, árvore e arquivo de saída

Troque aqui o modelo para 2, que é o branch-model two

# EXERCÍCIO 2

## RODANDO A ANÁLISE

- Antes de rodar, precisamos ativar o ambiente onde o paml está instalado: `source activate paml`
- Confira se você está dentro da pasta contendo todos os seus arquivos, usando `ls`
- E agora sim, vamos rodar o codeml: `codeml arquivo_controle.ctl`

# ANALISANDO DADOS

## EXERCÍCIO 3

Passo-a-passo do que vamos fazer:

- Criar uma nova pasta com: alinhamento, arquivo controle e árvore (normal, sem labels)
- Rodar os modelos de sítio: M1a e M2a
- Observar o arquivo de resultados e recuperar: lnL, df, omega (total), sítios sob seleção
- Comparar M1a e M2a: Likelihood Ratio Test



# EXERCÍCIO 3

## EDITANDO O ARQUIVO CONTROLE

```
seqfile = Mx_aln.phy      * Path to the alignment file  
treefile = Mx_unroot.tree  * Path to the tree file  
outfile = out_M0.txt       * Path to the output file  
noisy = 3                  * Display moderate information on the screen  
verbose = 1                 * Detailed output file  
seqtype = 1                 * Codon data  
Ndata = 1                   * One gene alignment  
icode = 0                   * Universal genetic code  
cleandata = 0               * Do not remove sites with ambiguity data  
model = 0                   * One w for all branches (M0 and site models)  
NSSites = 1                  * Site model M1a (nearly neutral)  
CodonFreq = 2                * Use F3x4 model  
clock = 0                    * Assume no clock  
fix_omega = 0                * Enables option to estimate omega  
omega = 0.5                 * Initial omega value
```

Coloque aqui o nome do alinhamento de códon, árvore e arquivo de saída

Troque aqui o modelo para 0 novamente, e o modelo de sítio (NSSites) para 1

# EXERCÍCIO 3

## EDITANDO O ARQUIVO CONTROLE

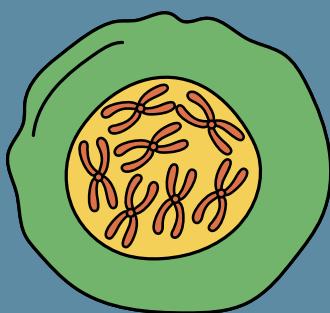
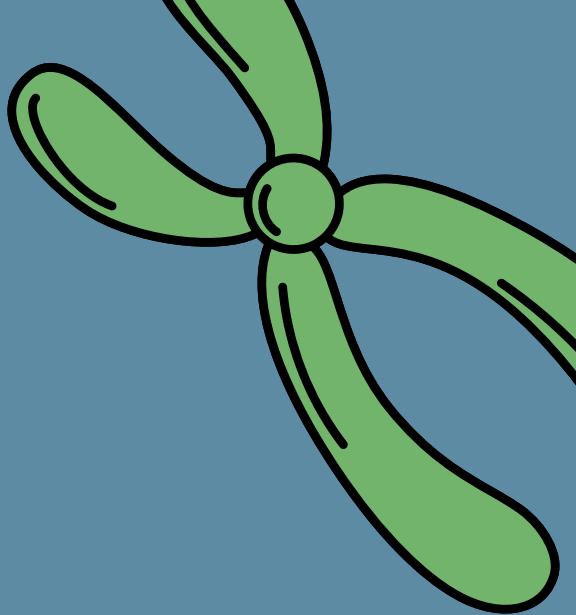
```
seqfile = Mx_aln.phy      * Path to the alignment file  
treefile = Mx_unroot.tree  * Path to the tree file  
outfile = out_M0.txt       * Path to the output file  
noisy = 3                  * Display moderate information on the screen  
verbose = 1                 * Detailed output file  
seqtype = 1                 * Codon data  
Ndata = 1                   * One gene alignment  
icode = 0                   * Universal genetic code  
cleandata = 0               * Do not remove sites with ambiguity data  
model = 0                   * Two ω for different sets of branches  
NSSites = 2                  * Site model M2a (positive selection)  
CodonFreq = 2                * Use F3x4 model  
clock = 0                    * Assume no clock  
fix_omega = 0                * Enables option to estimate omega  
omega = 0.5                 * Initial omega value
```

Coloque aqui o nome do alinhamento de códon, árvore e arquivo de saída

Troque aqui o modelo de ramos para 0, e o modelo de sitio (NSSites) para 2

# EXERCÍCIO 3

## VISUALIZANDO OS RESULTADOS



26  
parâmetros  
livres

comprimento  
total da árvore

```
TREE # 1: (1, 2, (((((3, 4), 5), 6), ((7, 8), 9)), 10), (11, 12)); MP score: -1
lnL(ntime: 21 np: 26): -12515.037921 +0.000000
 13..1 13..2 13..14 14..15 15..16 16..17 17..18 18..19 19..3 19..4 18..5 17..6
 0.728020 0.256898 2.537589 0.211655 0.069503 0.328017 0.048169 0.022410 0.014916 0.019767 0.024271 0.067498
28 0.104095 3.093702

Note: Branch length is defined as number of nucleotide substitutions per codon (not per nucleotide site).

tree length = 6.74614

(1: 0.728020, 2: 0.256898, (((((3: 0.014916, 4: 0.019767): 0.022410, 5: 0.024271): 0.048169, 6: 0.067498): 0
5, (11: 0.286782, 12: 0.250061): 0.587156): 2.537589);

(Chicken: 0.728020, Duck: 0.256898, (((((Chimpanzee: 0.014916, Human: 0.019767): 0.022410, Orangutan: 0.0242
63298): 0.193811): 0.069503, Dog: 0.419282): 0.211655, (Mouse: 0.286782, Rat: 0.250061): 0.587156): 2.537589)

Detailed output identifying parameters

kappa (ts/tv) = 2.73385

MLEs of dN/dS (w) for site classes (K=3)

p: 0.59854 0.35813 0.04333
w: 0.10409 1.00000 3.09370
```

ômegas para cada classe de sítio

$$\ln L = -12515.037921$$

árvore com comprimentos  
de ramo

# EXERCÍCIO 3

## VISUALIZANDO OS RESULTADOS

A primeira linha (p:) mostra a proporção de sítios pertencentes a cada classe

MLEs of dN/dS (w) for site classes (K=3)

p:	0.59854	0.35813	0.04333
w:	0.10409	1.00000	3.09370

classe 1  
 $w < 1$

classe 2  
 $w = 1$

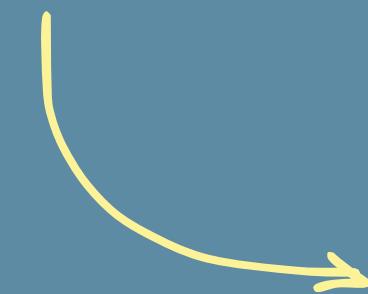
classe 3  
 $w > 1$

A segunda linha (w:) mostra os valores de ômega estimados para cada classe

# EXERCÍCIO 3

## VISUALIZANDO OS RESULTADOS

Sítios  
(potencialmente)  
sob seleção  
positiva



	Pr(w>1)	post mean +- SE for w
3 -	0.872	2.825
6 -	0.608	2.272
9 P	0.552	2.156
17 E	0.834	2.746
208 G	0.572	2.197
330 D	0.958*	3.005
363 Q	0.808	2.692
389 Q	0.585	2.224
396 S	0.936	2.960
418 K	0.657	2.375
433 C	0.818	2.713
485 E	0.575	2.203
492 N	0.632	2.323
514 D	0.823	2.723
565 -	0.826	2.730
570 C	0.766	2.604
571 A	0.959*	3.008
576 Q	0.715	2.497
577 C	0.625	2.308
579 S	0.562	2.176

	Pr(w>1)	post mean +- SE for w
3 -	0.884	2.638 +- 0.776
6 -	0.664	2.213 +- 0.963
9 P	0.621	2.119 +- 0.960
17 E	0.859	2.579 +- 0.803
32 D	0.570	2.005 +- 0.945
130 T	0.527	1.934 +- 0.952
208 G	0.649	2.158 +- 0.938
330 D	0.959*	2.776 +- 0.638
363 Q	0.839	2.536 +- 0.824
389 Q	0.661	2.173 +- 0.927
396 S	0.941	2.742 +- 0.676
418 K	0.717	2.289 +- 0.914
433 C	0.844	2.550 +- 0.820
485 E	0.645	2.158 +- 0.948
492 N	0.687	2.247 +- 0.943
514 D	0.850	2.559 +- 0.813
565 -	0.847	2.565 +- 0.826
570 C	0.802	2.471 +- 0.867
571 A	0.961*	2.780 +- 0.631
576 Q	0.759	2.388 +- 0.903
577 C	0.685	2.241 +- 0.941
579 S	0.637	2.134 +- 0.941
582 L	0.546	1.947 +- 0.926
622 S	0.501	1.865 +- 0.918

# ANALISANDO DADOS

## EXERCÍCIO 4

Passo-a-passo do que vamos fazer:

- Criar uma nova pasta com: alinhamento, arquivo controle e árvore (marcada, com labels)
- Rodar os modelos de ramo-sítio: A-model e A-model null
- Observar o arquivo de resultados e recuperar: lnL, df, omega (total), sítios sob seleção
- Comparar A-model e A-model null: Likelihood Ratio Test



# EXERCÍCIO 4

## EDITANDO O ARQUIVO CONTROLE

```
seqfile = Mx_aln.phy      * Path to the alignment file  
treefile = Mx_unroot.tree  * Path to the tree file  
outfile = out_M0.txt       * Path to the output file  
noisy = 3                  * Display moderate information on the screen  
verbose = 1                 * Detailed output file  
seqtype = 1                 * Codon data  
Ndata = 1                   * One gene alignment  
icode = 0                   * Universal genetic code  
cleandata = 0               * Do not remove sites with ambiguity data  
model = 2                   * Two ω for different sets of branches  
NSSites = 2                  * Site model M2a (positive selection)  
CodonFreq = 2                * Use F3x4 model  
clock = 0                    * Assume no clock  
fix_omega = 0                * Enables option to estimate omega  
omega = 0.5                 * Initial omega value
```

Coloque aqui o nome do alinhamento de códon, árvore e arquivo de saída

Coloque aqui o modelo de ramo como 2, e o modelo de sítio (NSSites) como 2 também

# EXERCÍCIO 4

## MARCANDO RAMOS NA ÁRVORE

- Vamos marcar com labels os ramos das aves, para os quais um ômega diferente será estimado:

(Chicken #1: 0.577417, Duck #1: 0.287022, (((((Chimpanzee: 0.014533, Human: 0.019238): 0.021587, Orangutan: 0.023881): 0.043540, Rhesus\_macaque: 0.067966): 0.289313, ((Cow: 0.052746, Sheep: 0.065521): 0.258956, Pig: 0.243382): 0.185313): 0.081203, Dog: 0.363933): 0.231387, (Mouse: 0.253505, Rat: 0.226870): 0.437367): 1.639958);

# EXERCÍCIO 4

## VISUALIZANDO OS RESULTADOS

26  
parâmetros  
livres

comprimento  
total da árvore

$$\ln L = -12471.439883$$

```
TREE # 1: (1, 2, (((((3, 4), 5), 6), ((7, 8), 9)), 10), (11, 12)); MP score: -1
lnL(ntime: 21 np: 26): -12471.439883 +0.000000
 13..1 13..2 13..14 14..15 15..16 16..17 17..18 18..19 19..3 19..4 18..5 17..6
 0.661651 0.322318 2.505996 0.196213 0.065382 0.322753 0.046727 0.022240 0.014815 0.019599 0.024085 0.067530
45 0.071866 2.603509

Note: Branch length is defined as number of nucleotide substitutions per codon (not per nucleotide site).

tree length = 6.67714

(1: 0.661651, 2: 0.322318, (((((3: 0.014815, 4: 0.019599): 0.022240, 5: 0.024085): 0.046727, 6: 0.067530):
3, (11: 0.282615, 12: 0.243514): 0.603563): 2.505996;

(Chicken: 0.661651, Duck: 0.322318, (((((Chimpanzee: 0.014815, Human: 0.019599): 0.022240, Orangutan: 0.024
59992): 0.193416): 0.065382, Dog: 0.414574): 0.196213, (Mouse: 0.282615, Rat: 0.243514): 0.603563): 2.505996

Detailed output identifying parameters

kappa (ts/tv) = 2.68364

MLEs of dN/dS (w) for site classes (K=4)

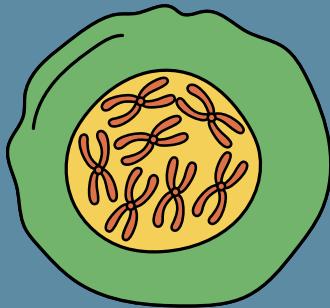
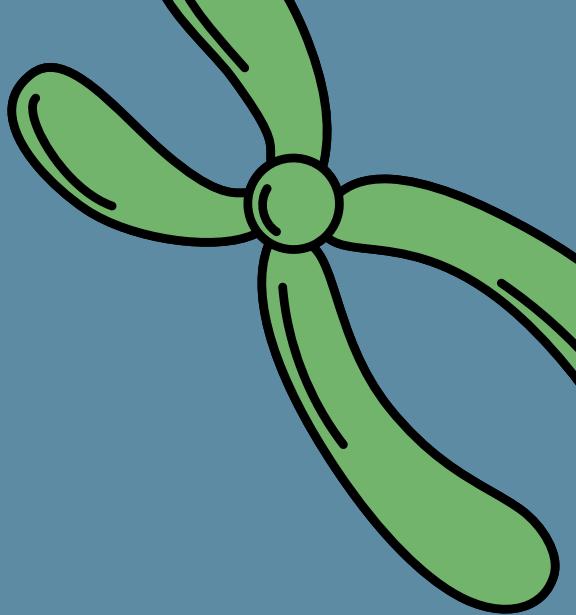
site class      0       1      2a      2b
proportion  0.47016  0.27524  0.16059  0.09401
background w  0.07187  1.00000  0.07187  1.00000
foreground w  0.07187  1.00000  2.60351  2.60351
```

ômegas para cada classe de sítio, para ramos foreground e background

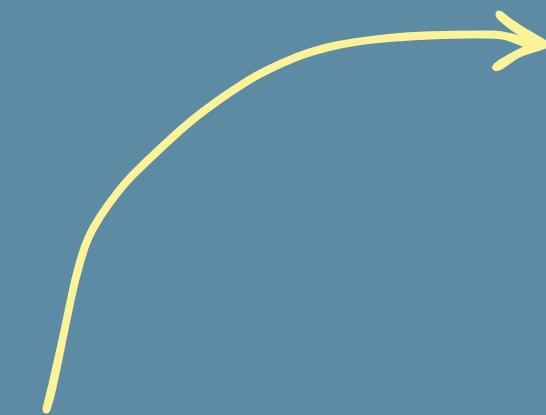
árvore com comprimentos de ramo

# EXERCÍCIO 5

## VISUALIZANDO OS RESULTADOS



Sítios  
(potencialmente)  
sob seleção  
positiva



Bayes Empirical Bayes (BEB) analysis (Yang, Wong & Nielsen 2005. Mol. Biol. Evol. 22:1107–1118)  
Positive sites for foreground lineages Prob(w>1):

39	C	0.503
40	S	0.803
46	D	0.654
47	R	0.992**
50	Q	0.827
56	V	0.812
61	K	0.868
63	D	0.903
65	G	0.788
66	N	0.761
69	M	0.991**
79	R	0.783
80	N	0.687
86	V	0.756
94	A	0.860
98	D	0.718
99	K	0.829
101	V	0.724
102	I	0.816
114	M	0.556
124	V	0.992**
131	E	0.996**
133	Q	0.992**

# INTERPRETANDO!!

## (A PARTE MAIS IMPORTANTE DO TRABALHO)

- Vamos abrir as nossas tabelas de resultados e olhar quais modelos se ajustaram melhor aos dados, para as análises de ramo, sítio e ramo-sítio.

Quais modelos foram melhores em cada caso? E o que isso significa?

Os resultados dos modelos de ramo e de sítio tem o mesmo significado?