Présentation mini-projets

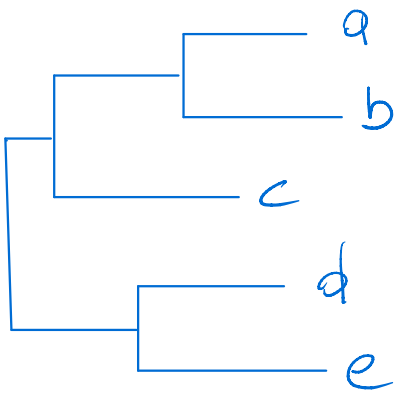
Goal : build a Python module to calculate the likelihood of a phylogenetic tree

Step 1 : find a way to store in a Python object
1) the tree topology and branch lengths
2) an alignment

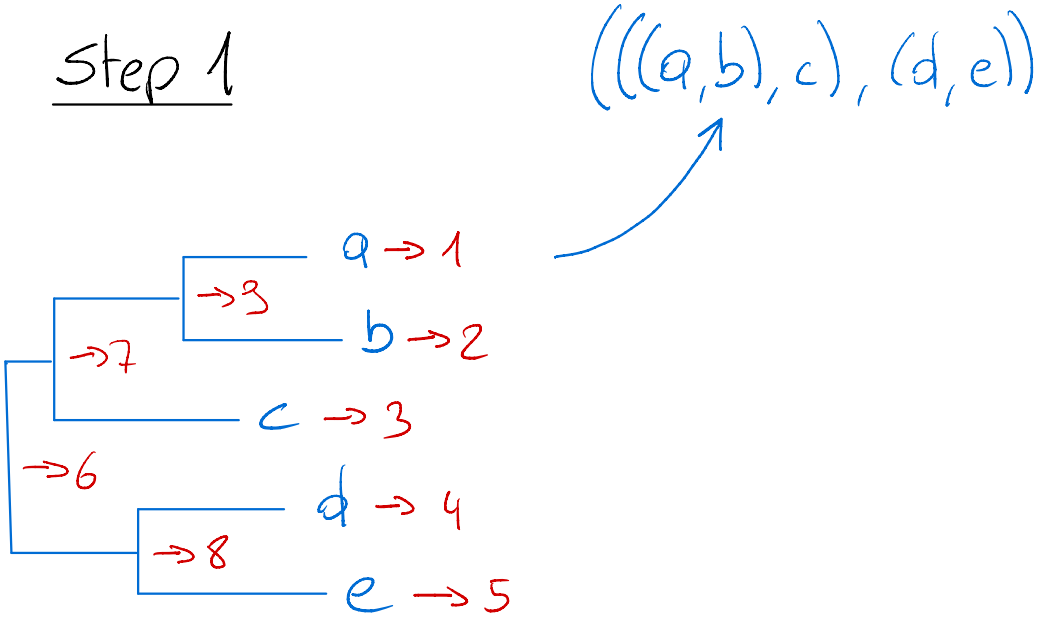
Step 2 : calculate the probability of the alignment given the tree (and a model of evolution)

Step 1

((a,b),c),(d,e))

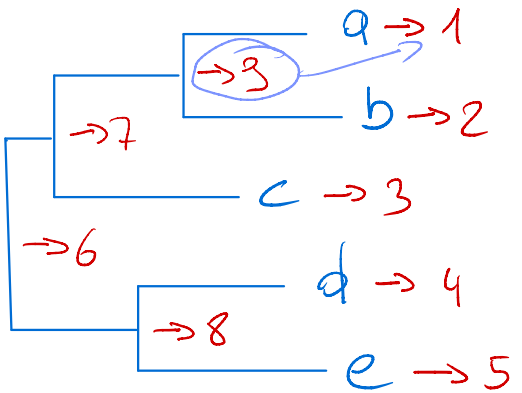


Step 1



Step 1

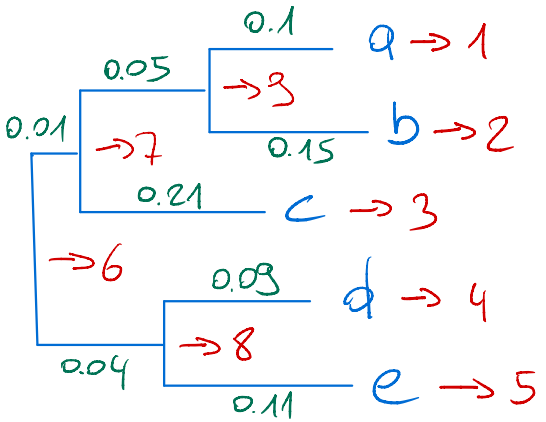
$$(((a,b),c),(d,e))$$



Parent	Child
9	1
9	2
8	4
8	5
7	3
7	9
6	7
6	8

Step 1

$$(((a,b),c),(d,e))$$



Parent	Child		Branch lengths
9	1		0.1
9	2		0.15
8	4		0.09
8	5	+	0.11
7	3		0.21
7	9		0.05
6	7		0.01
6	8		0.04

Step 1

1) create a tree object to store the phylogenetic tree
2) import the parent-child table and the branch lengths vector into Python
3) populate the tree object with each node of the tree

Tips : a tree is a list of nodes, and nodes have attributes like branch length, parent, children, etc...

Step 1

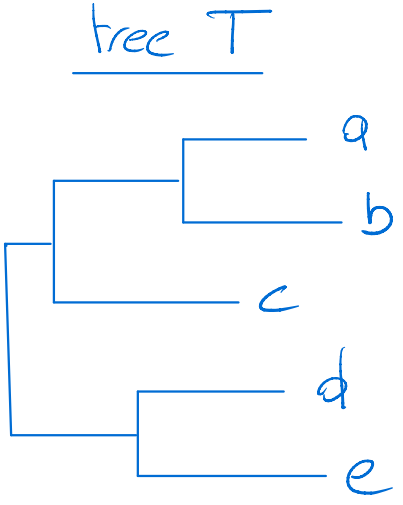
4) each node will have data coming from an alignment

a A C C G T
b A T C C T
c A C G G A
d C G A C C
e C T A C G

5) import the alignment and store it in the nodes

Step 2:

1) once you have a tree object, calculate the probability of the alignment given the tree



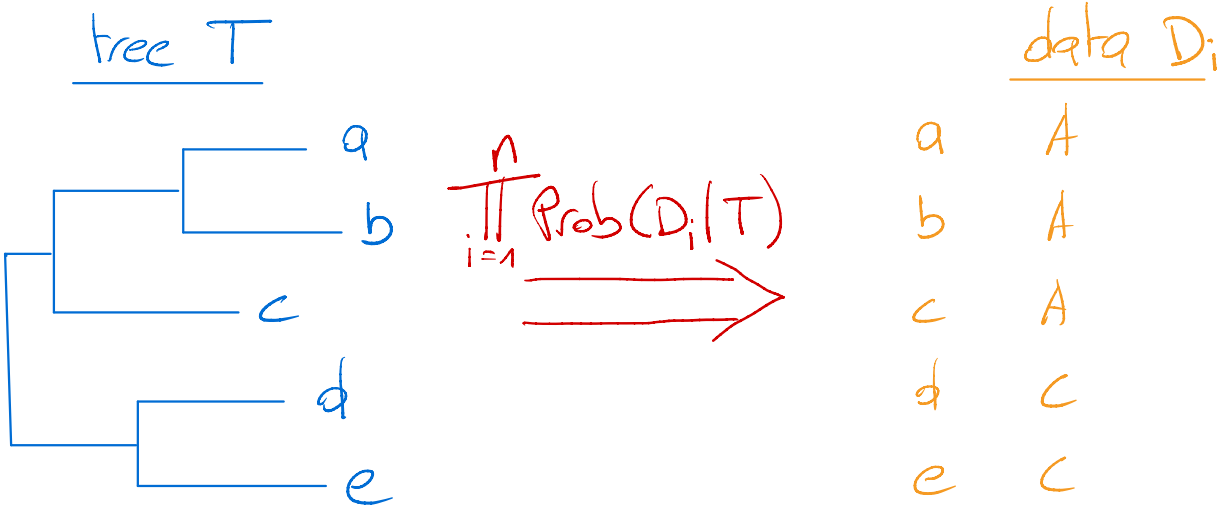
Prob(D|T)

data D

a	ACCGT
b	ATCCT
c	ACGGA
d	CGACC
e	CTACG

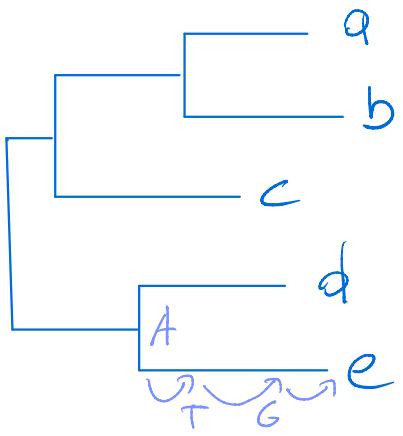
Step 2:

1) once you have a tree object, calculate the probability of the alignment given the tree



Step 2 : How to calculate the probability ?

tree T



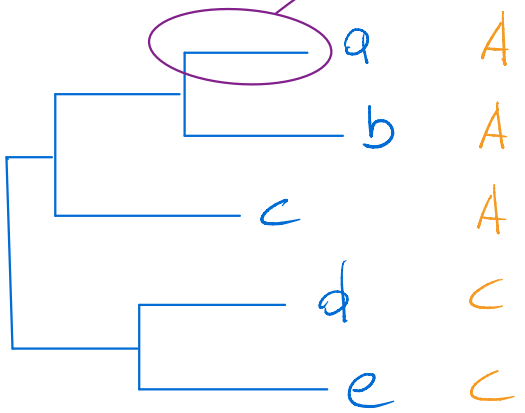
D

A
A
A
C
C

model of evolution Q

	A	C	G	T
A	-3μ	μ	μ	μ
C	μ	-3μ	μ	μ
G	μ	μ	-3μ	μ
T	μ	μ	μ	-3μ

Step 2



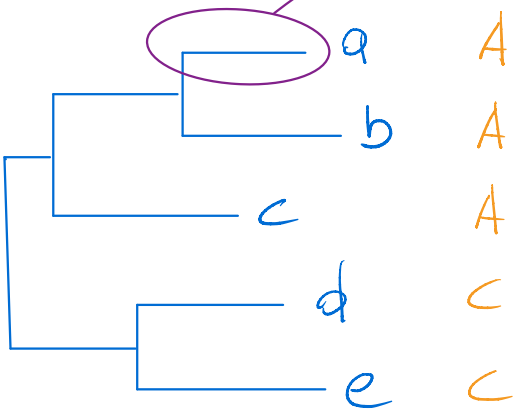
Probability of evolving a A along this branch if we do not know the nucleotide of the ancestor

$$P(A|x) = e^{Q \cdot t} \times \underbrace{\begin{matrix} A & C & G & T \\ [1, & 0, & 0, & 0] \end{matrix}}_{A}$$

Q → JC

t ← branch length

Step 2



Probability of evolving a A along this branch if we do not know the nucleotide of the ancestor

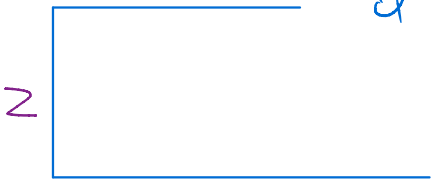
$$P(A|x) = e^{Q \cdot t} \times \underbrace{[1,0,0,0]}_{A}$$

$[0,1,0,0] = C$

etc...

Step 2

$e^{Q \times 0.1} \times [1,0,0,0]$



A

A

$e^{Q \times 0.15} \times [1,0,0,0]$

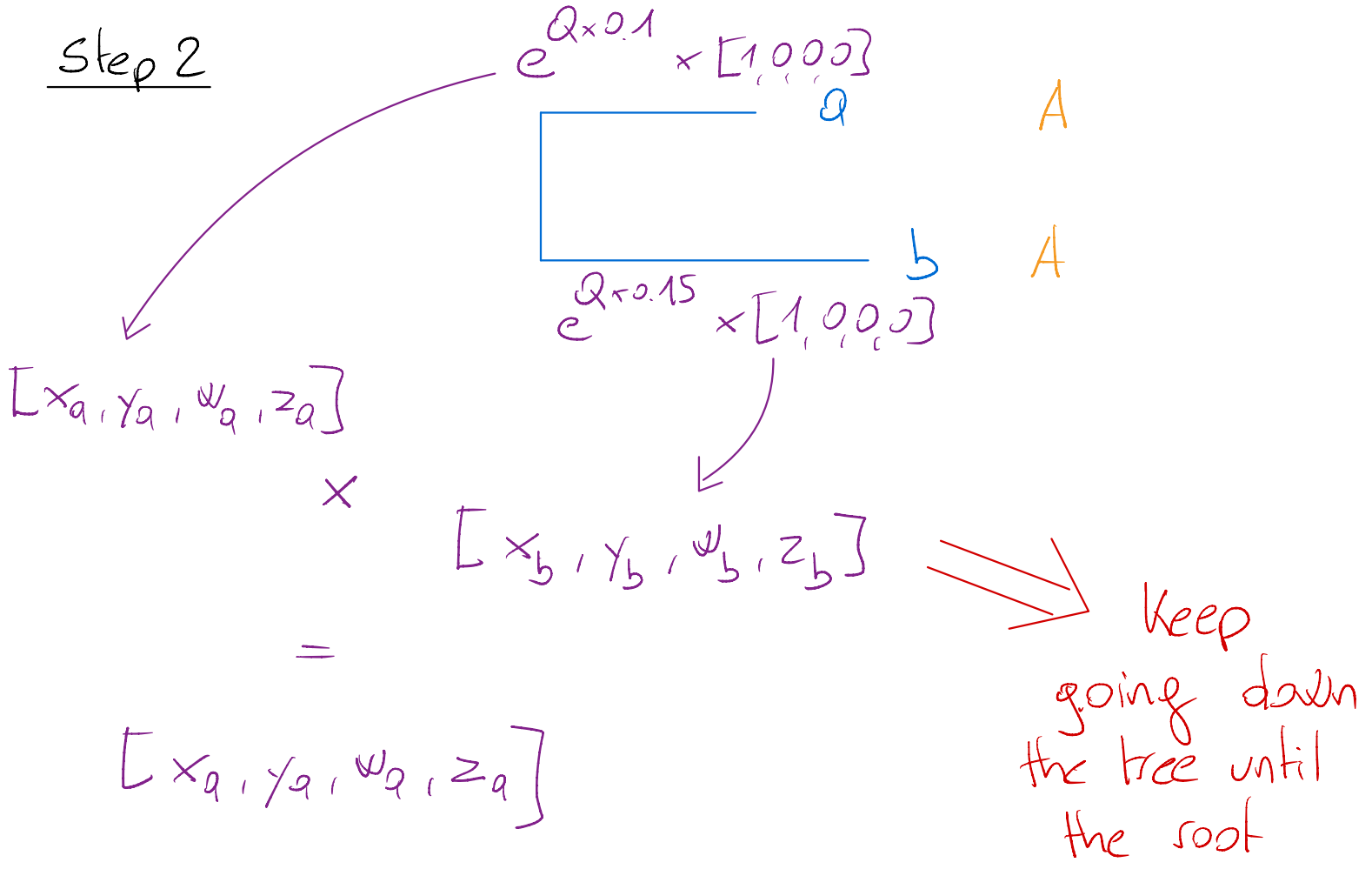
$[x_a, y_a, w_a, z_a]$

$\times$

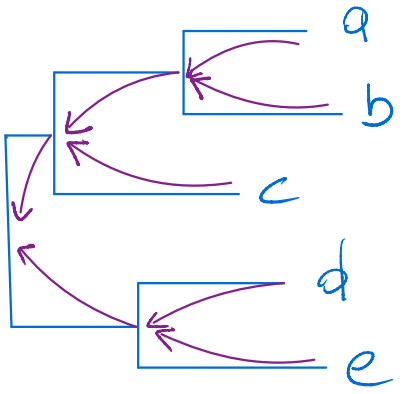
$[x_b, y_b, w_b, z_b]$

$=$

$[x_2, y_2, w_2, z_2]$


Step 2
$e^{Q \times 0.1} \times [1,0,0,0]$
a
A
b
A
$e^{Q \times 0.15} \times [1,0,0,0]$
$[x_a, y_a, w_a, z_a]$
×
$[x_b, y_b, w_b, z_b]$
=
$[x_a, y_a, w_a, z_a]$
keep going down the tree until the root

Step 2



$$[x_r, y_r, w_r, z_r]$$

$$L(D|T) = \sum [x_r, y_r, w_r, z_r]$$

→ transform into $\ln L$ and add up each $\ln L$ per site to get the global likelihood

What is expected in the mini-project :

1) find a way to store a tree and alignment in a Python object.

2) calculate the $\ln L$ for an alignment
   a) with a fixed μ parameter
   b) by estimating μ (if you can)
   c) think about the efficiency of the algorithm

3) should be able to take any tree and alignment

Schedule :

1) start working in pairs today
   → contact us by Teams anytime (@team1, ...)

2) send by Teams an overview of your approach
   → deadline : Fri Nov 8th 5pm

3) come with some code on Mon Nov 11th
   → presentation of code versioning

4) work on your own (but contact us to ask questions)

5) Nov 25th → presentation of your code + feedbacks