

TUTORIAL 8

1. Programas de alinhamento geram topologias com custos distintos? (Tab. 8.1)
2. Há algum critério que possa ser utilizado de alinhamentos?
3. Parâmetros de alinhamento influenciam o resultado de sua análise?
4. Há algum critério que possa ser utilizado para a escolha de parâmetros?
5. Árvores-guias influenciam o resultado do alinhamento → topologia?
6. Como manter a consistência lógica interna de sua análise?
7. Há evidências de que alinhamento, escolha de parâmetros e inferência filogenética são interdependentes?

Alinhamentos – homologia estática:

Procedimento analítico: 2 etapas distintas e interdependentes.

Alinhamentos múltiplos:

Função objetiva: maximiza similaridade

Dependente de:

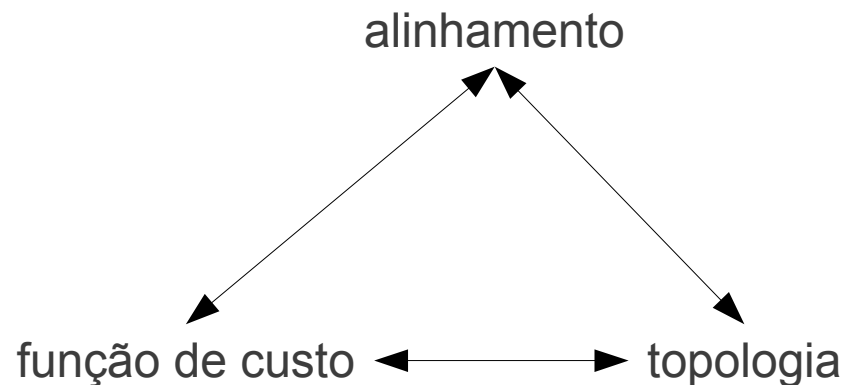
Função de custo

Árvore-guia

Inferência filogenética:

1. Via de regra viola consistência interna pela adoção de funções objetivas distintas

2. Dependente do alinhamento



Árvores e alinhamento:

TAP: *tree alignment problem*

Identificar o alinhamento, dado uma topologia, com o menor custo.

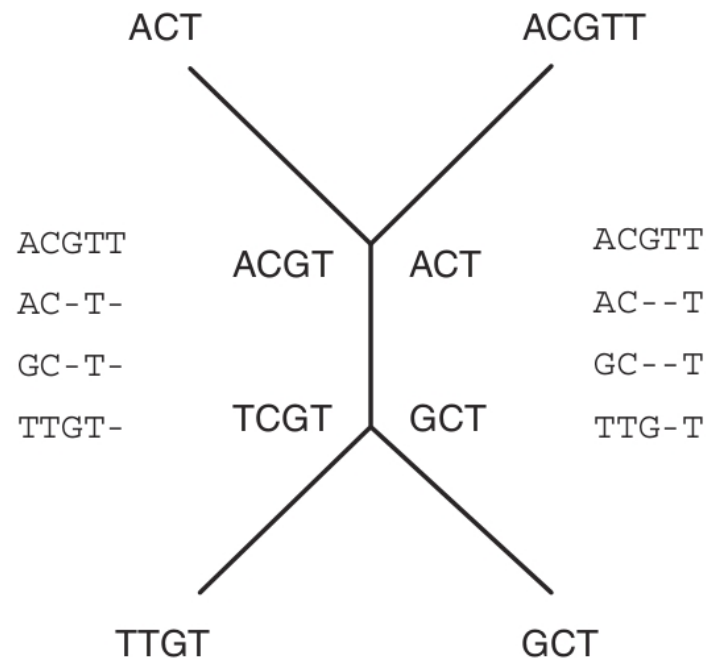


Figure 8.8: Example tree alignment. Either set of medians (ACGT, TCGT; ACT, GCT) yields a cost of 6 transformations (indels and substitutions). The two alignments also result in a tree cost of 6.

Tree alignment problem:

Soluções exatas: virtualmente não-realista

m	2	3	4	5
$n = 1$	3	13	75	541
2	13	409	23917	2244361
3	63	16081	10681263	14638756721
4	321	699121	5552351121	117629959485121
5	1683	32193253	3147728203035	1.05×10^{18}
10	8097453	9850349744182729	3.32×10^{26}	1.35×10^{38}

Table 8.1: Number of multiple sequence alignments for *very* small data sets (m sequences of length n).

Alinhamento Múltiplo Heurístico:

Implementação: operadores comuns

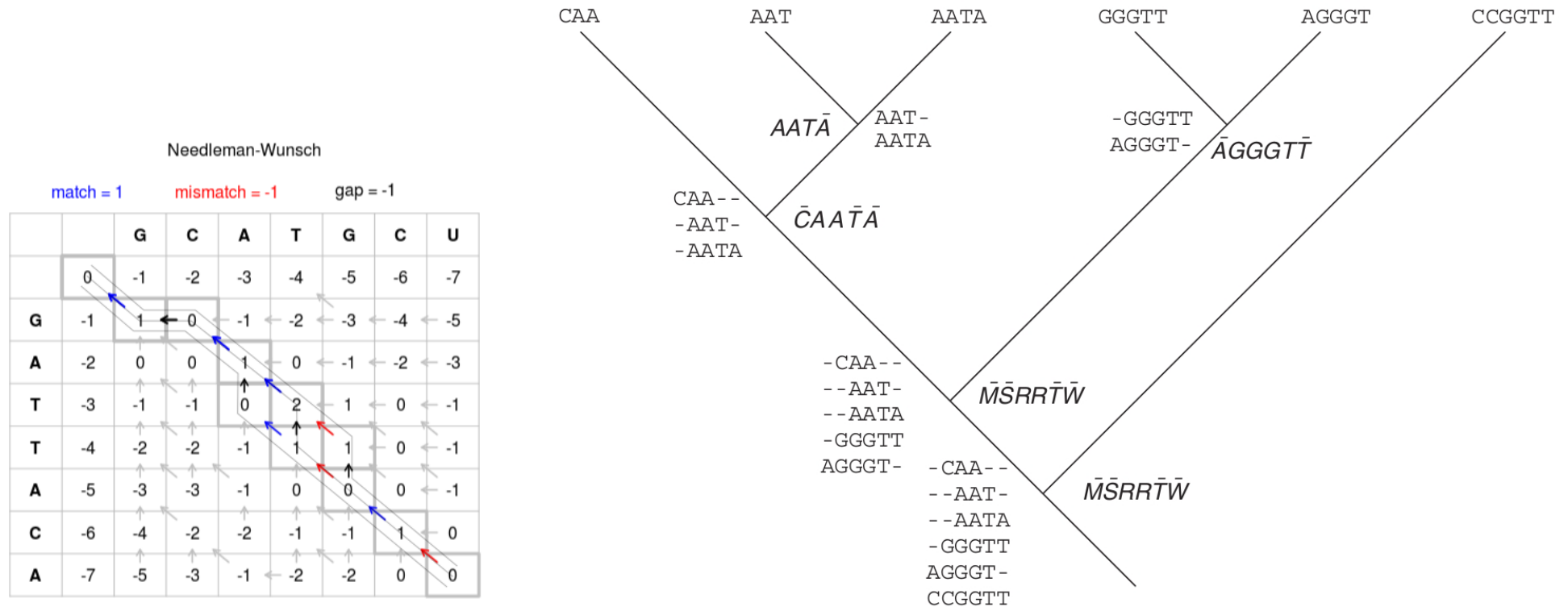


Figure 8.10: Progressive alignment of Feng and Doolittle (1987) where pairwise alignments are performed via post-order tree traversal at non-leaf vertices to create partial MSAs following “once a gap, always a gap” (left of vertices) or profile sequences (right of vertices) of IUPAC symbols for each column with a “bar” if a gap is also present in that aligned column.

Alinhamento Múltiplo Heurístico:

Implementação:

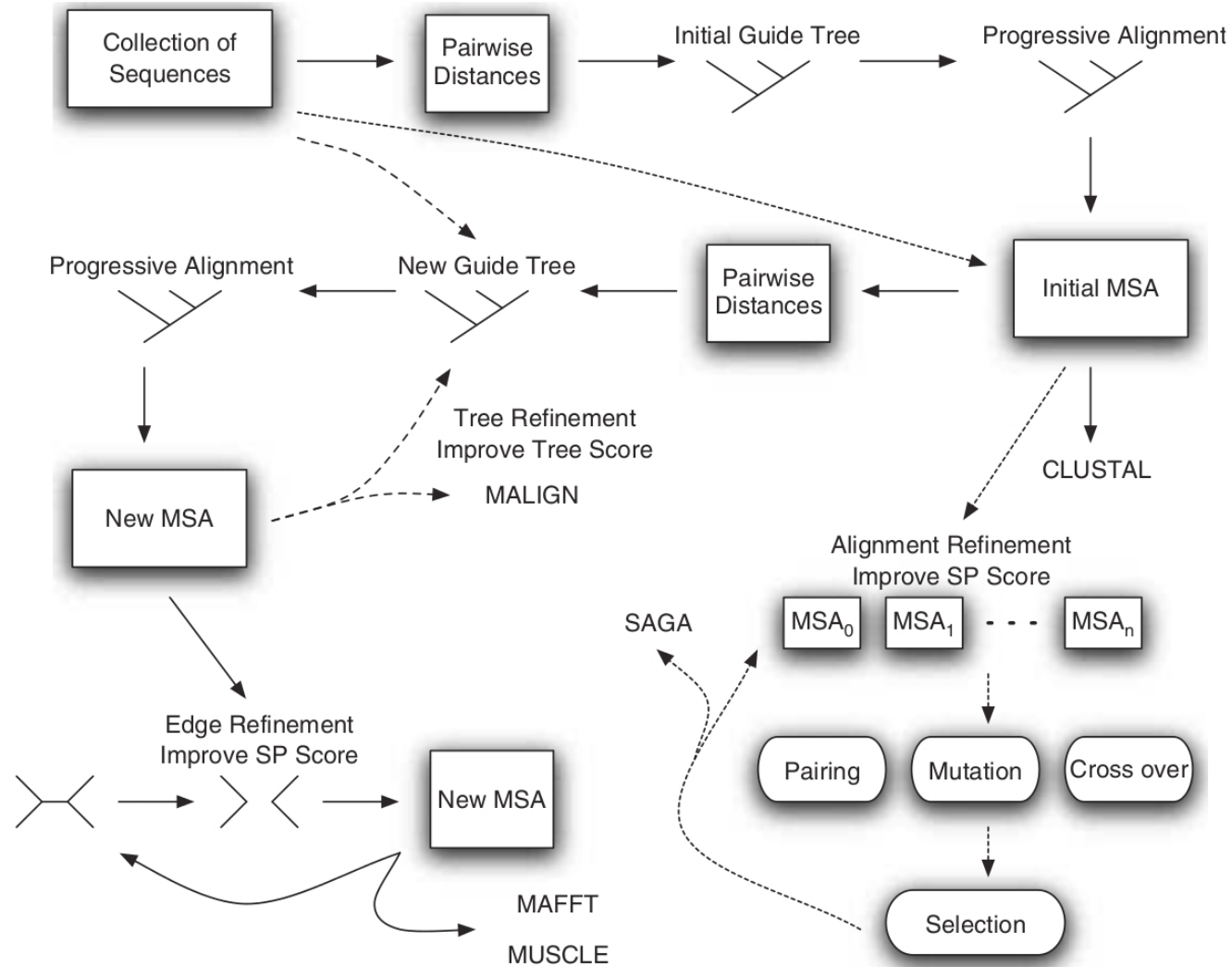
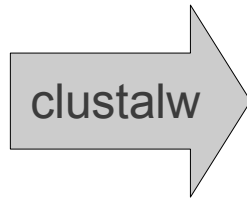


Figure 8.11: Procedural relationships among heuristic MSA implementations.

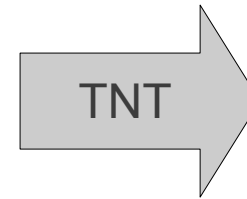
Homologia estática:

Implementação:

T1 CTGGCTACGT
T2 TGGTAAGT
T3 CCTAAGT
T4 CCTGATGC



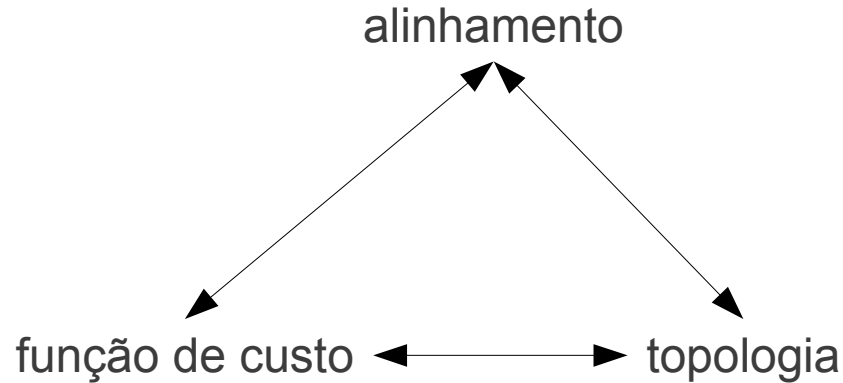
T1 CTGGCTACGT
T2 -TGG-TAAGT
T3 ---CCTAAGT
T4 CCTGATGC--



Tree 0:
 ,-- T1
 |--| ,-- T4
 `--| ,-- T3
 `----- T2

Implicit enumeration, 1 trees found, score 12.

Otimização direta – homologia dinâmica



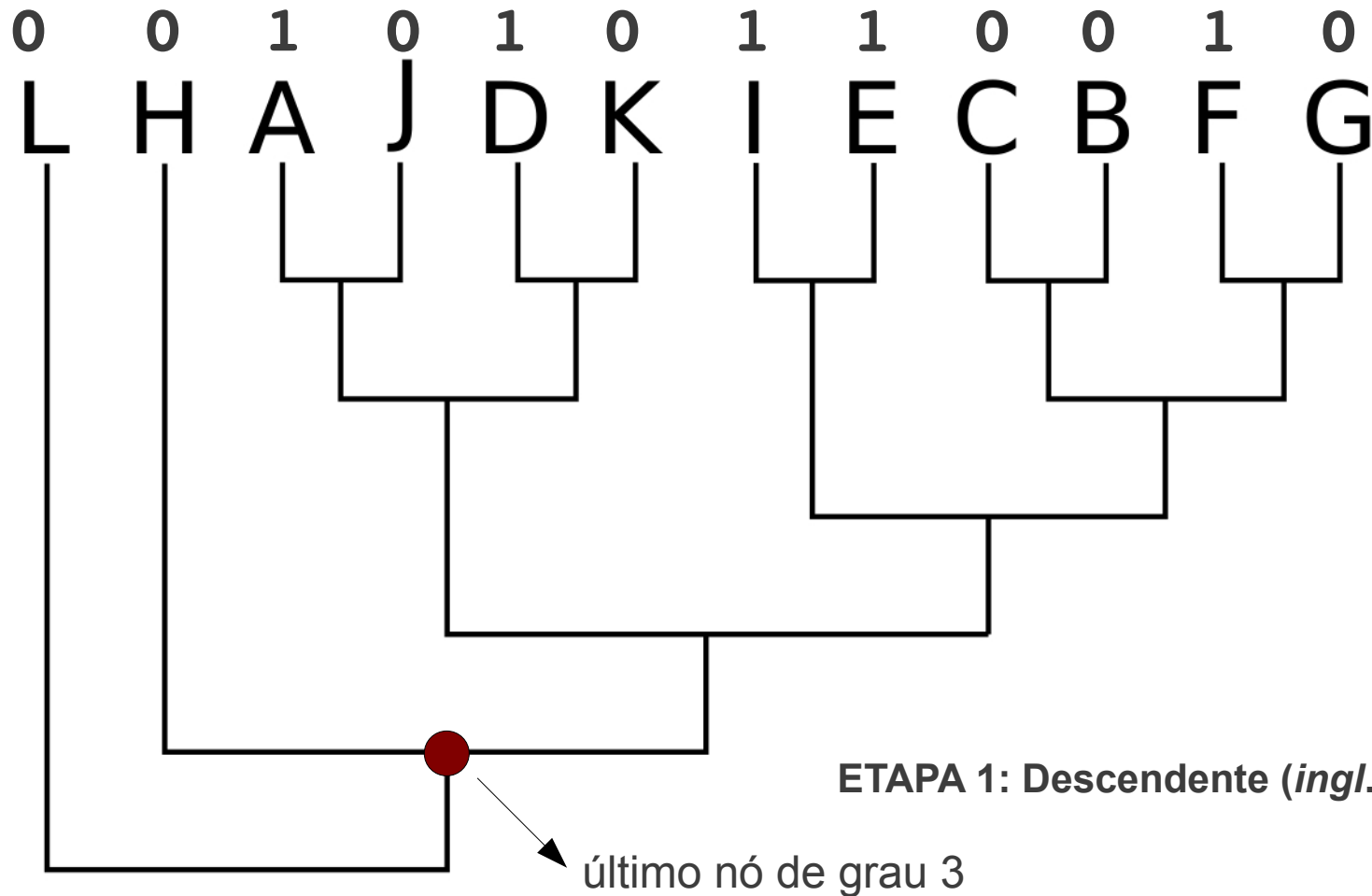
Procedimento analítico único: busca e alinhamento

Inferência filogenética:

1. Manutenção da consistência interna da análise
2. Incorpora todas as evidências simultaneamente
3. Alinhamento (implícito) é subproduto da análise

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

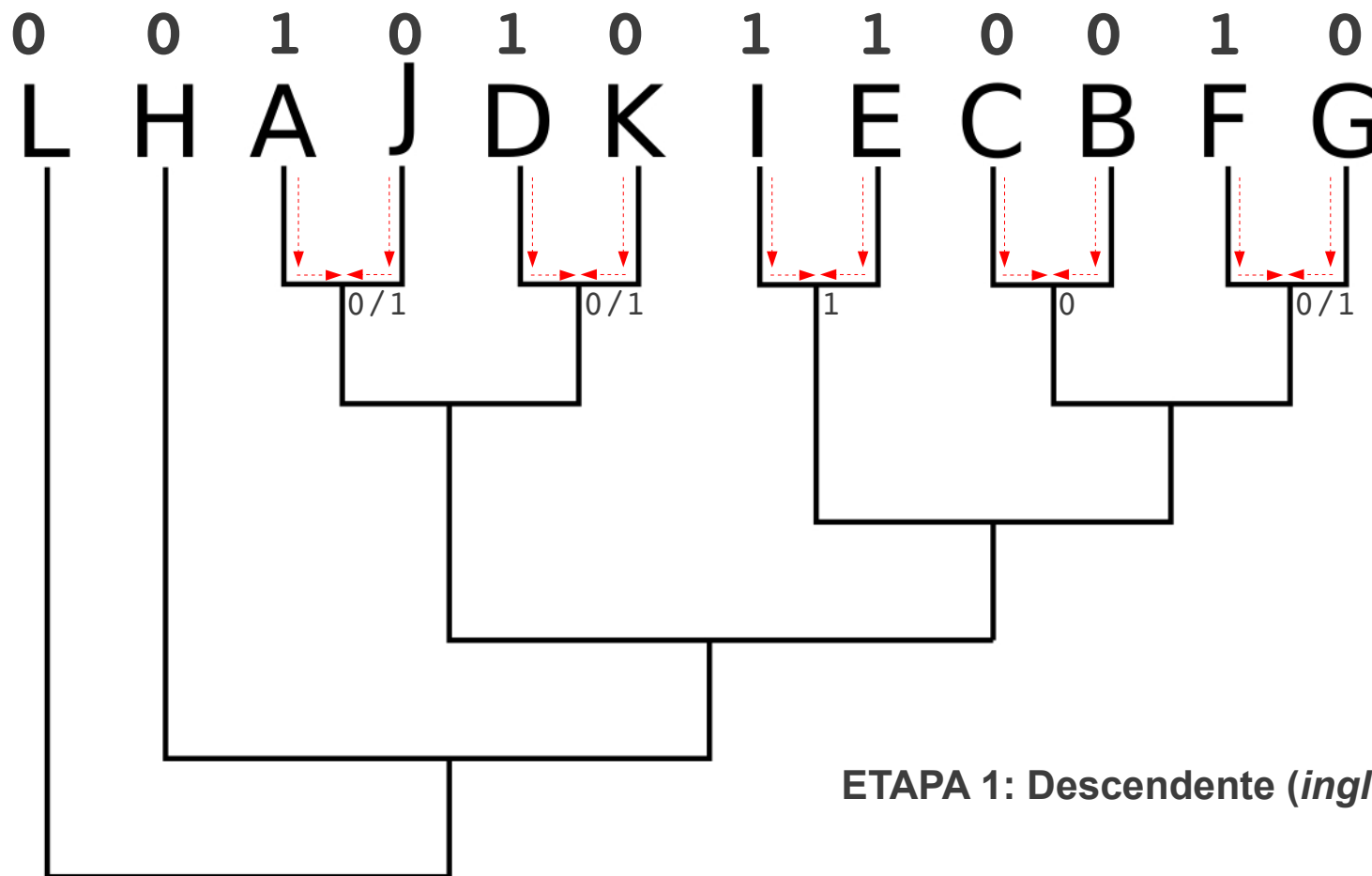


Regras:

1. se $1 \& 1$ ou $0 \& 0$ ou $0/1 \& 0/1 \rightarrow$ atribui-se ao nó 1 ou 0 ou 0/1, respectivamente.
2. se $0 \& 1 \rightarrow$ atribui-se ao nó 0/1.
3. se $0/1 \& 1$ ou $0/1 \& 0 \rightarrow$ atribui-se ao nó 1 ou 0, respectivamente.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:



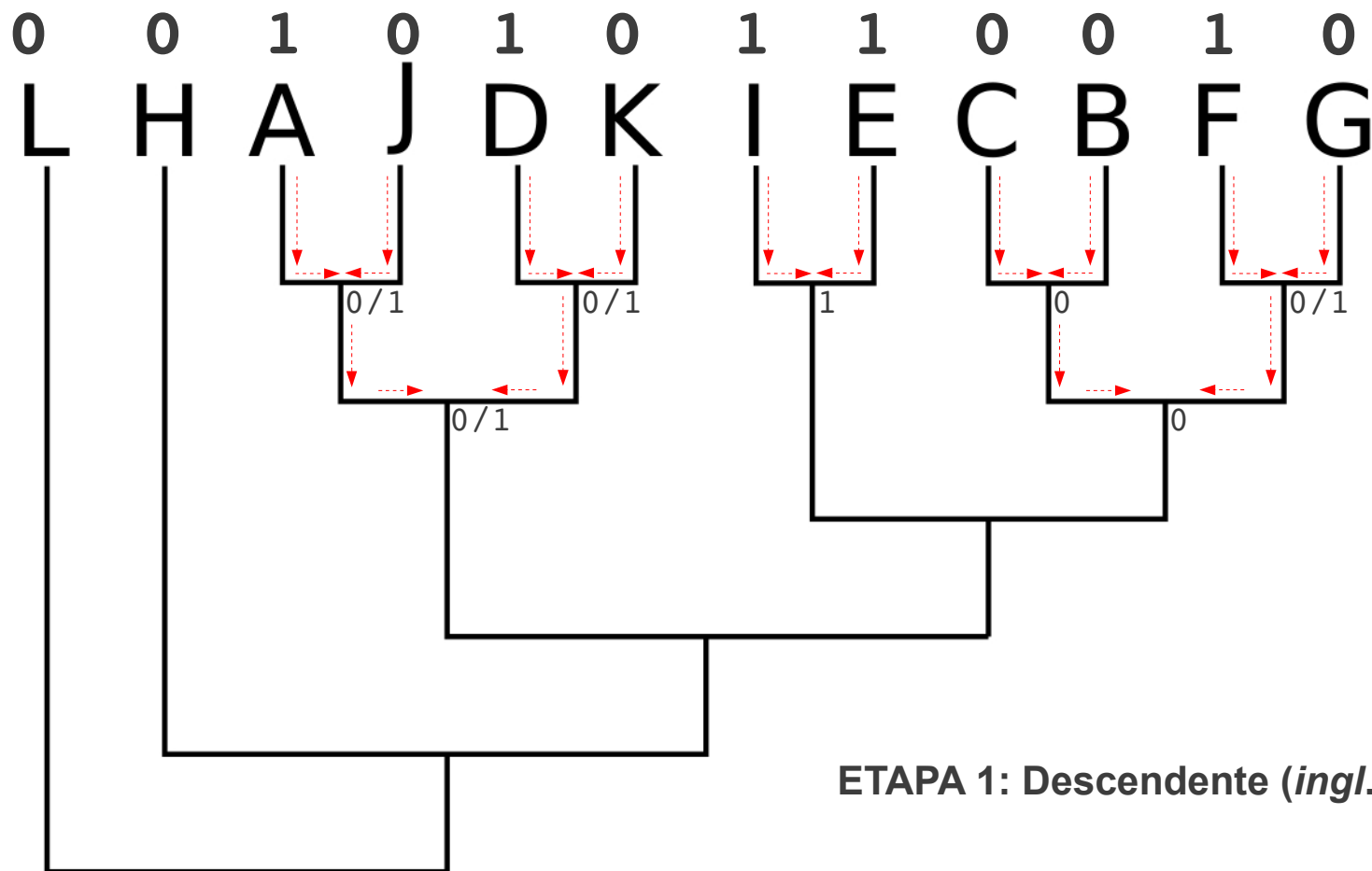
ETAPA 1: Descendente (*ingl. down pass*)

Regras:

1. se $1 \& 1$ ou $0 \& 0 \rightarrow$ atribui-se ao nó 1 ou 0, respectivamente.
2. se $0 \& 1 \rightarrow$ atribui-se ao nó 0/1.
3. se $0/1 \& 1$ ou $0/1 \& 0 \rightarrow$ atribui-se ao nó 1 ou 0, respectivamente.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

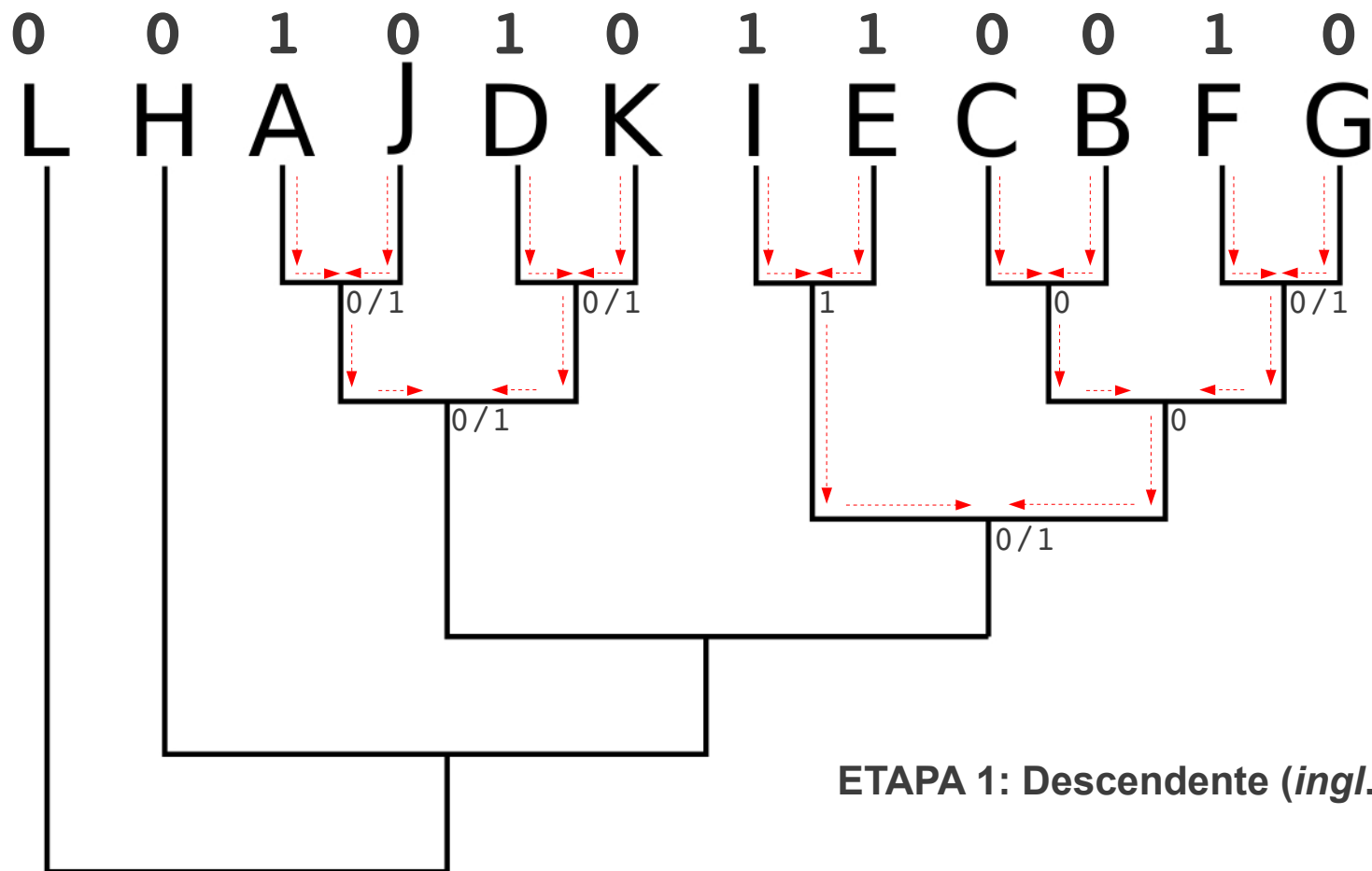


Regras:

1. se 1 & 1 ou 0 & 0 ou 0/1 & 0/1 → atribui-se ao nó 1 ou 0 ou 0/1, respectivamente.
2. se 0 & 1 → atribui-se ao nó 0/1.
3. se 0/1 & 1 ou 0/1 & 0 → atribui-se ao nó 1 ou 0, respectivamente.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

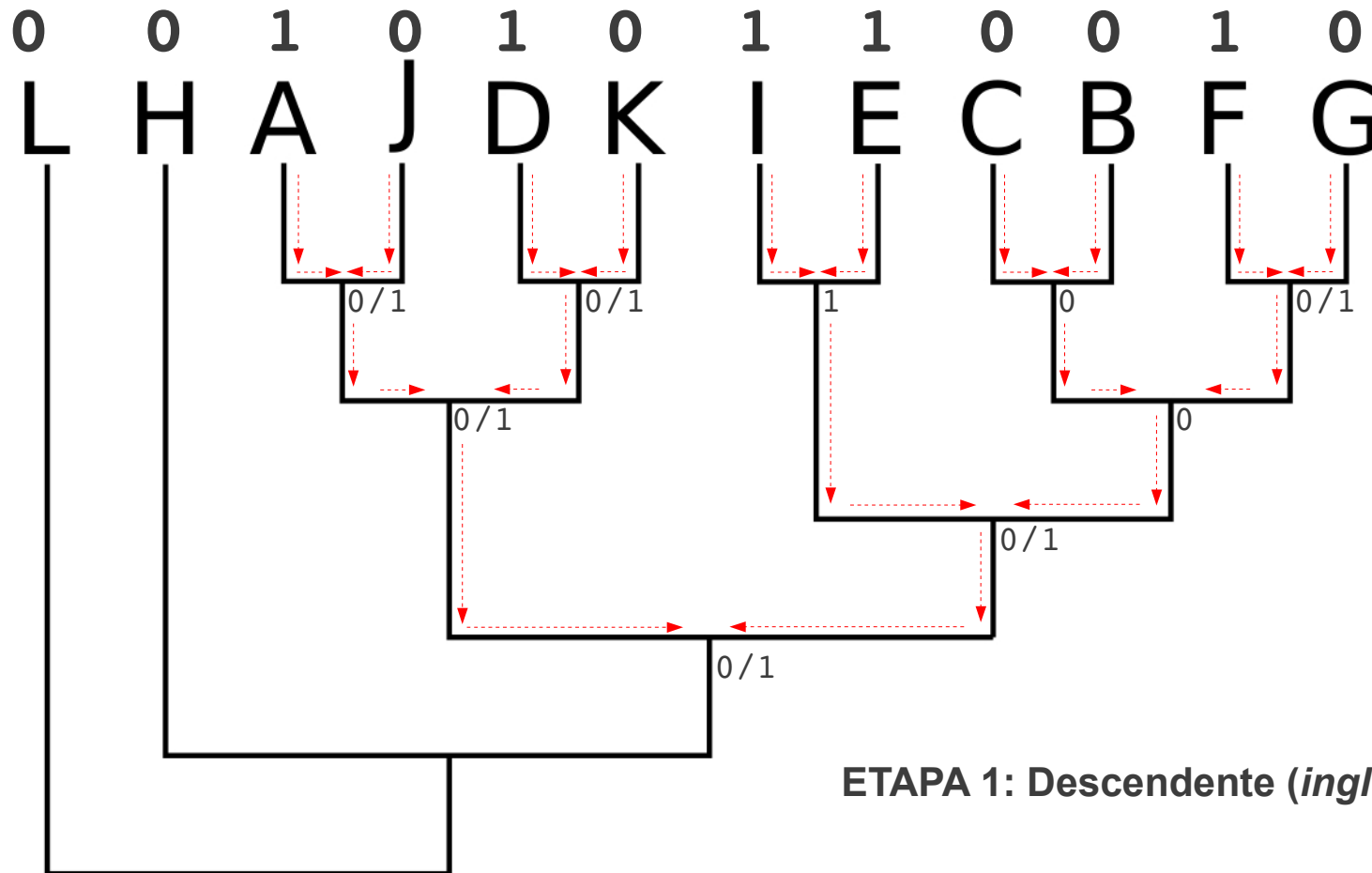


Regras:

1. se 1 & 1 ou 0 & 0 ou 0/1 & 0/1 → atribui-se ao nó 1 ou 0 ou 0/1, respectivamente.
2. se 0 & 1 → atribui-se ao nó 0/1.
3. se 0/1 & 1 ou 0/1 & 0 → atribui-se ao nó 1 ou 0, respectivamente.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

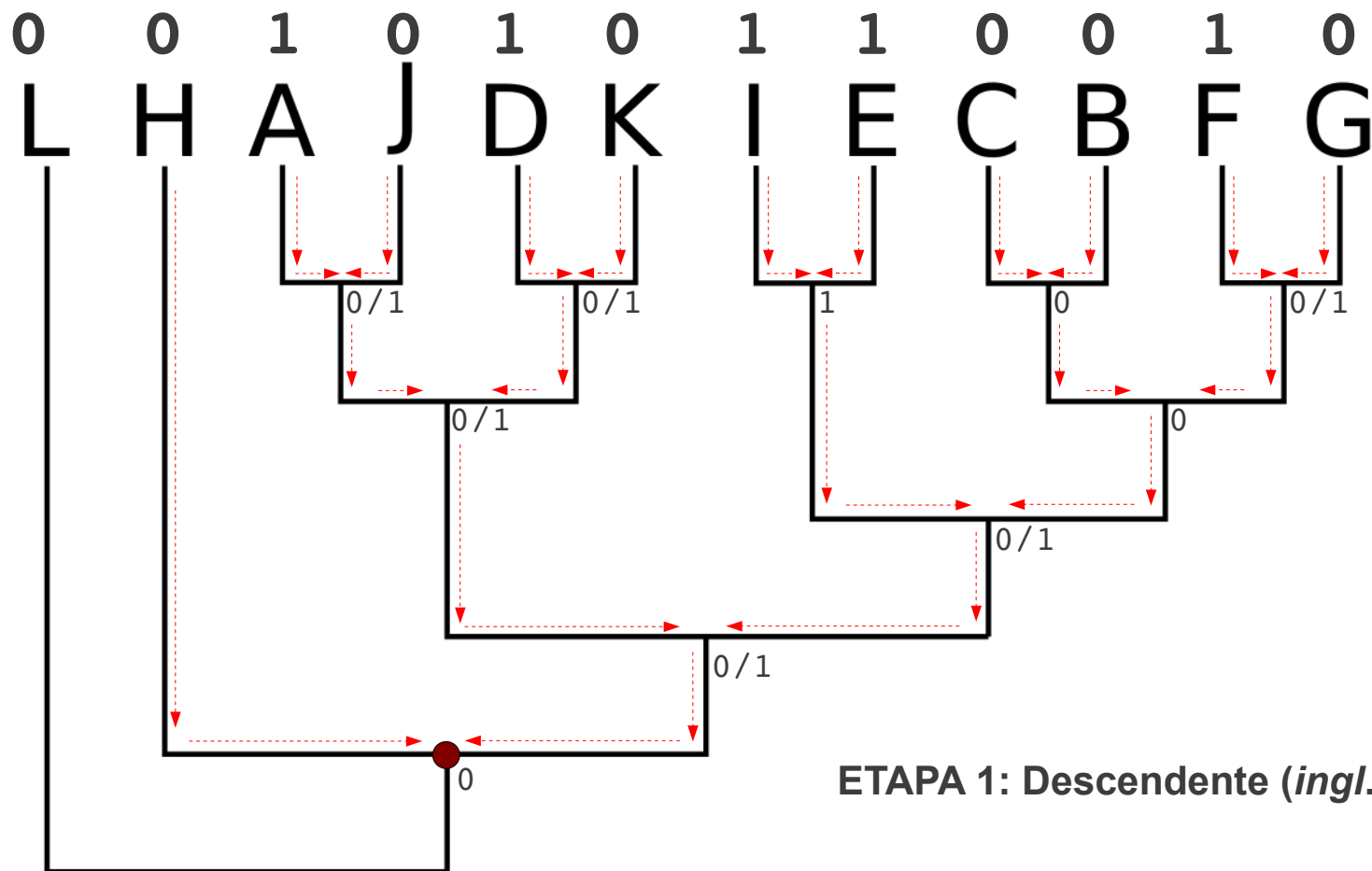


Regras:

1. se 1 & 1 ou 0 & 0 ou 0/1 & 0/1 → atribui-se ao nó 1 ou 0 ou 0/1, respectivamente.
2. se 0 & 1 → atribui-se ao nó 0/1.
3. se 0/1 & 1 ou 0/1 & 0 → atribui-se ao nó 1 ou 0, respectivamente.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

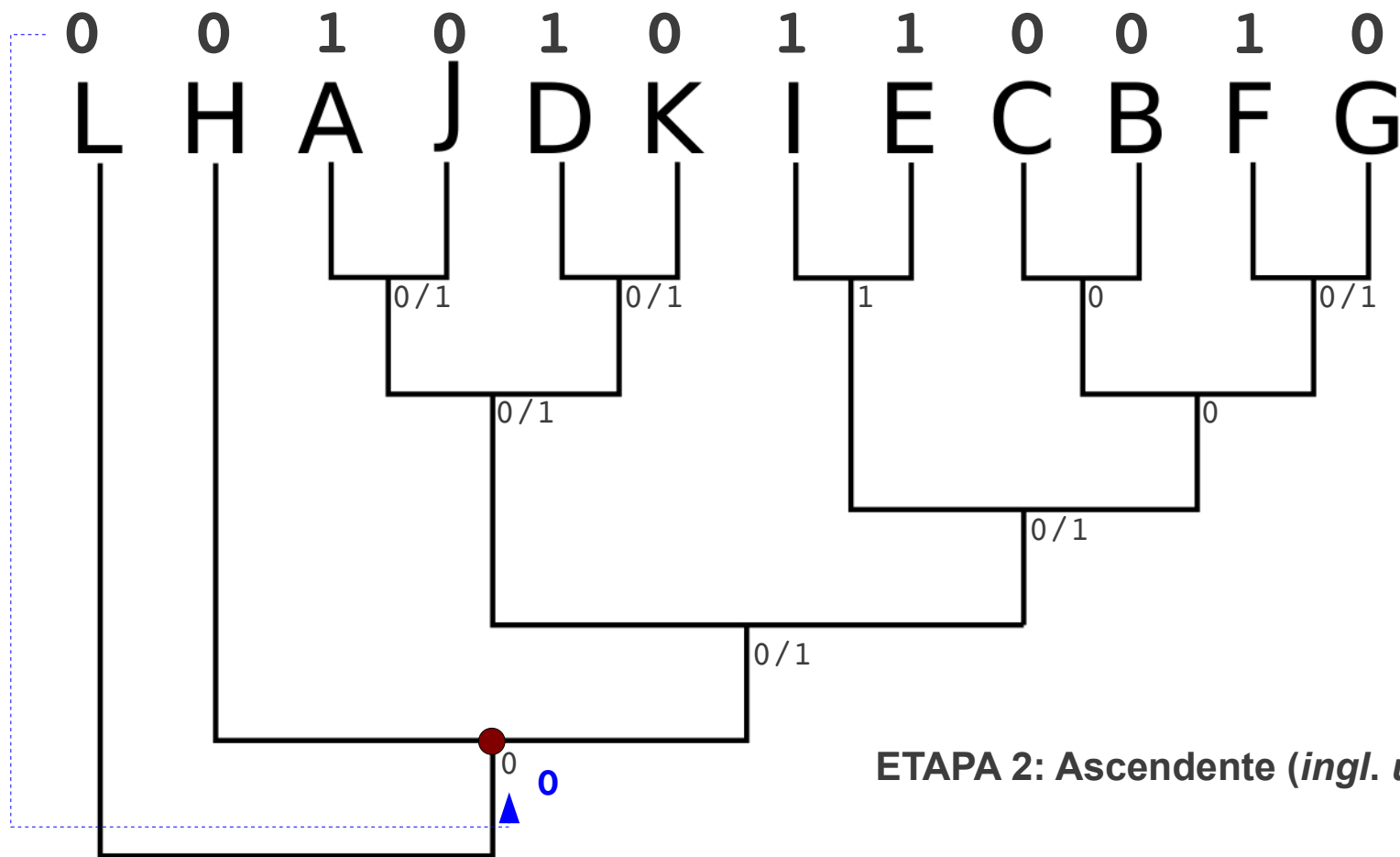


Regras:

1. se 1 & 1 ou 0 & 0 ou 0/1 & 0/1 → atribui-se ao nó 1 ou 0 ou 0/1, respectivamente.
2. se 0 & 1 → atribui-se ao nó 0/1.
3. se 0/1 & 1 ou 0/1 & 0 → atribui-se ao nó 1 ou 0, respectivamente.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

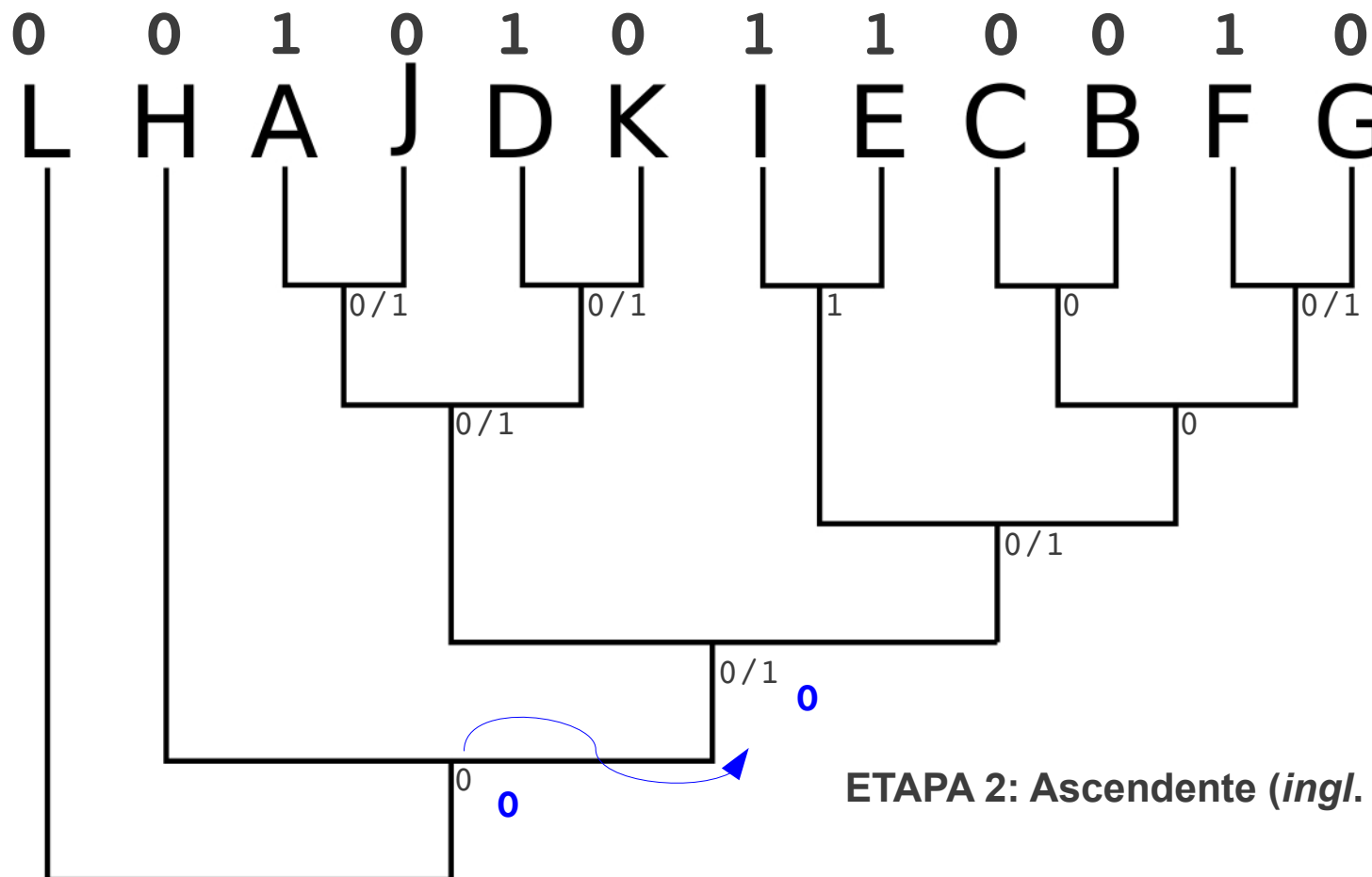


Regras:

1. se 1 & 1 ou 0 & 0 → atribui-se ao nó 1 ou 0, respectivamente.
2. se 0 & 1 → atribui-se ao nó o estado presente no **nó superior**.
3. se 1 & 0/1 ou 0 & 0/1 → atribui-se ao nó o estado presente no **nó inferior**.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

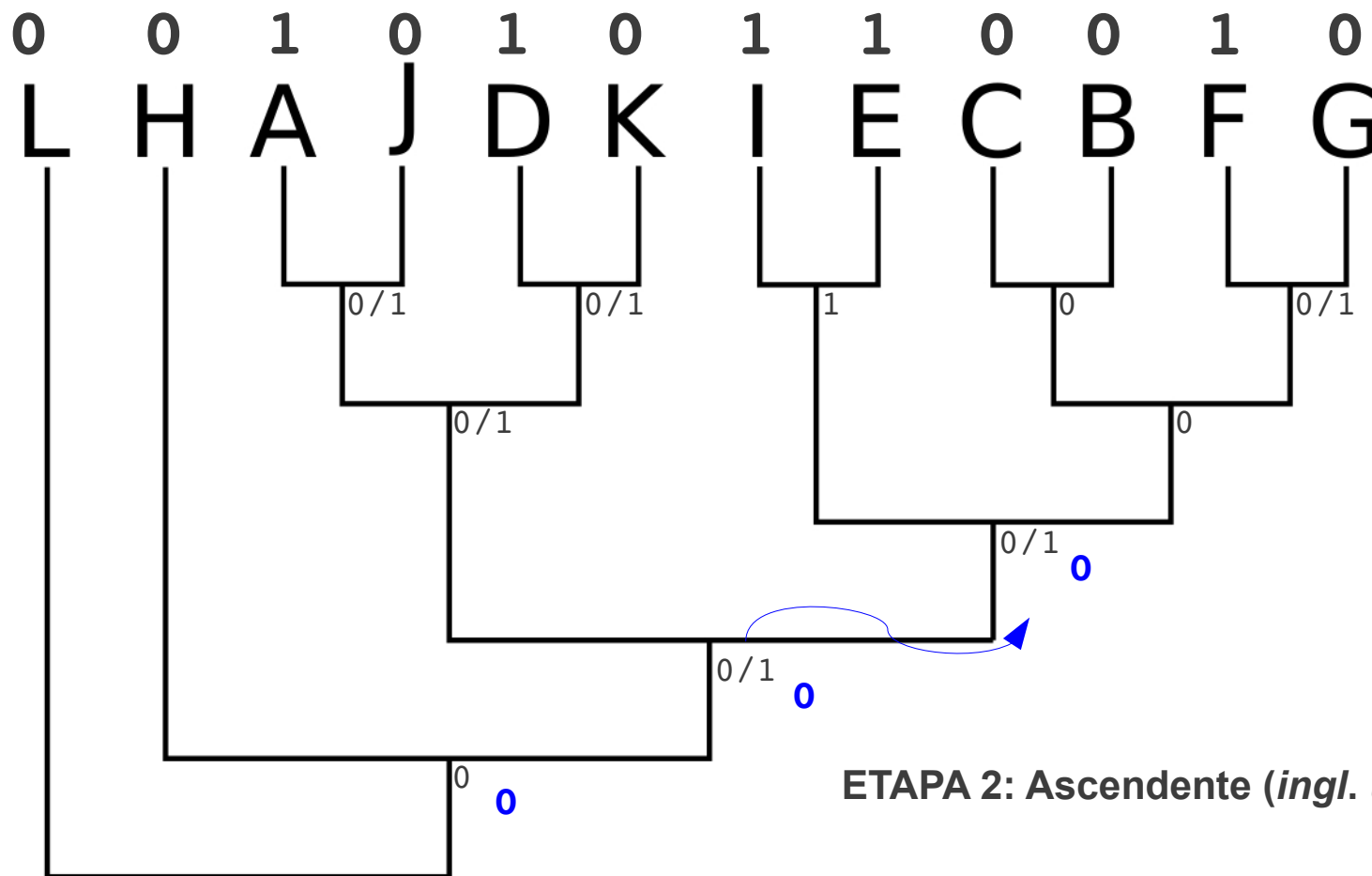


Regras:

1. se 1 & 1 ou 0 & 0 → atribui-se ao nó 1 ou 0, respectivamente.
2. se 0 & 1 → atribui-se ao nó o estado presente no **nó superior**.
3. se 1 & 0/1 ou 0 & 0/1 → atribui-se ao nó o estado presente no **nó inferior**.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:



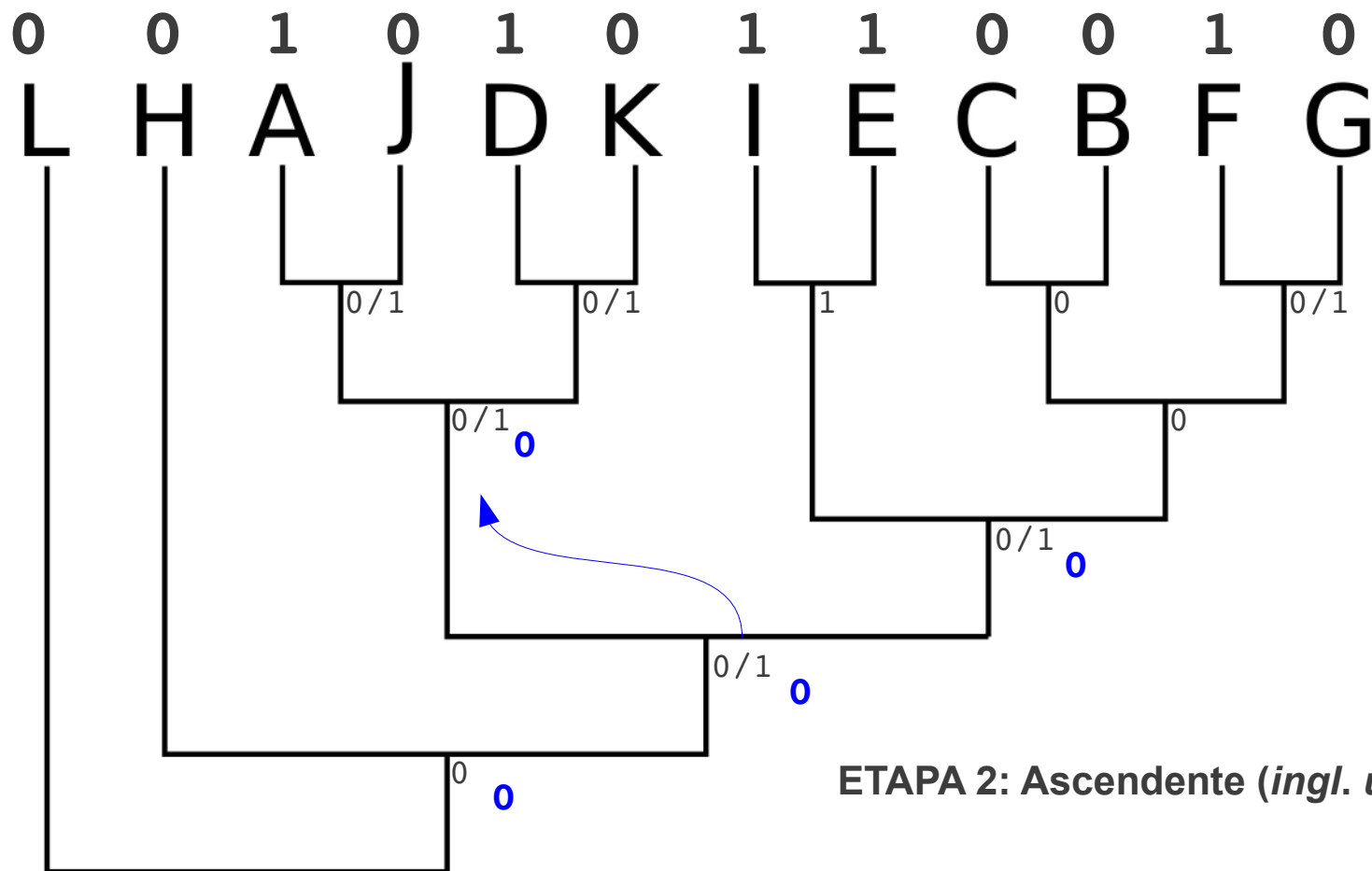
ETAPA 2: Ascendente (*ingl. up pass*)

Regras:

1. se 1 & 1 ou 0 & 0 → atribui-se ao nó 1 ou 0, respectivamente.
2. se 0 & 1 → atribui-se ao nó o estado presente no **nó superior**.
3. se 1 & 0/1 ou 0 & 0/1 → atribui-se ao nó o estado presente no **nó inferior**.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:



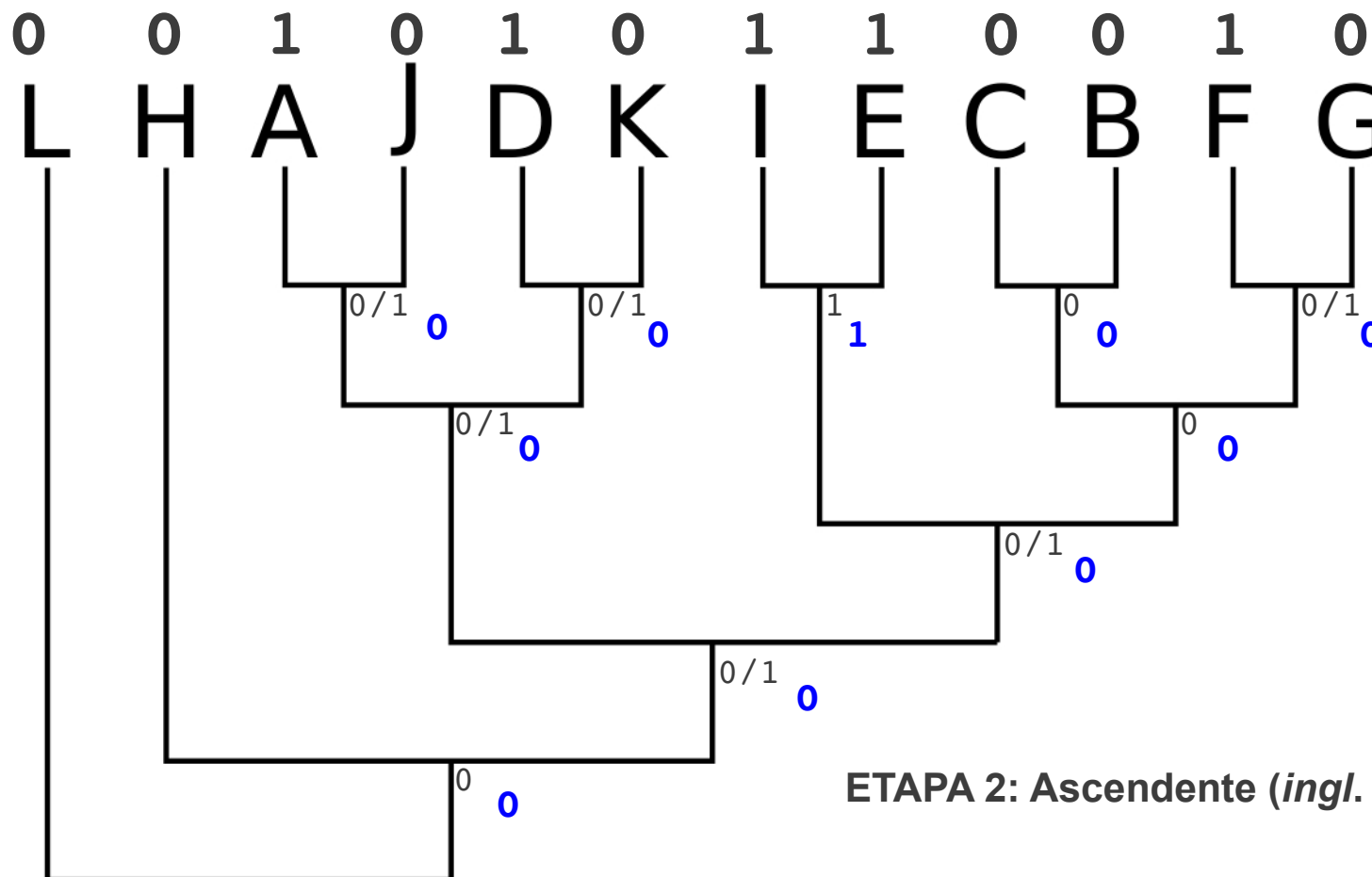
ETAPA 2: Ascendente (*ingl. up pass*)

Regras:

1. se 1 & 1 ou 0 & 0 → atribui-se ao nó 1 ou 0, respectivamente.
2. se 0 & 1 → atribui-se ao nó o estado presente no **nó superior**.
3. se 1 & 0/1 ou 0 & 0/1 → atribui-se ao nó o estado presente no **nó inferior**.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:

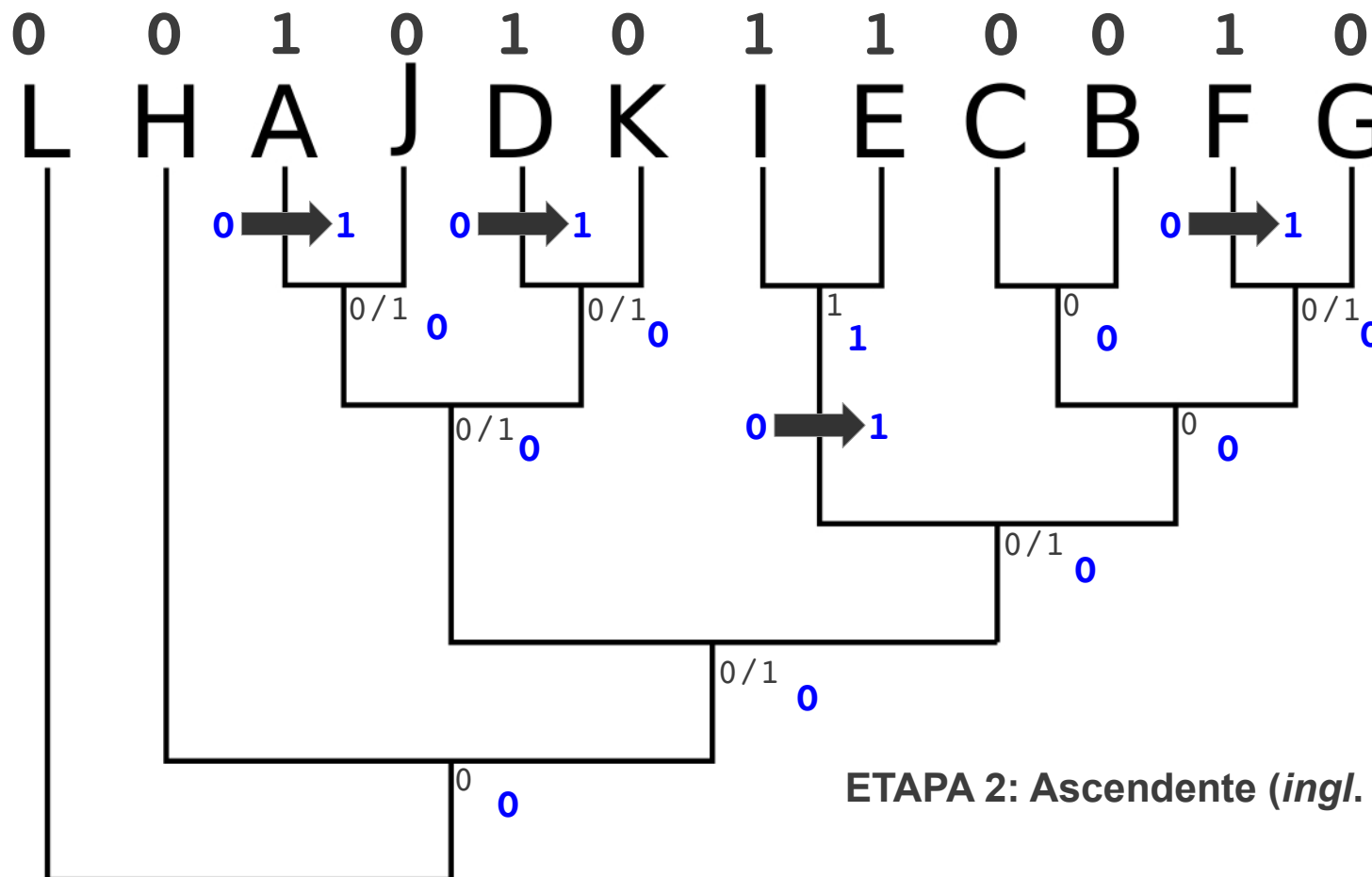


Regras:

1. se 1 & 1 ou 0 & 0 → atribui-se ao nó 1 ou 0, respectivamente.
2. se 0 & 1 → atribui-se ao nó o estado presente no **nó superior**.
3. se 1 & 0/1 ou 0 & 0/1 → atribui-se ao nó o estado presente no **nó inferior**.

Otimização de diagramas enraizados:

Considere a seguinte topologia e distribuição de estados:



Regras:

1. se 1 & 1 ou 0 & 0 → atribui-se ao nó 1 ou 0, respectivamente.
2. se 0 & 1 → atribui-se ao nó o estado presente no **nó superior**.
3. se 1 & 0/1 ou 0 & 0/1 → atribui-se ao nó o estado presente no **nó inferior**.

Algoritmo de Needleman & Wunsch (1970)

A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins

SAUL B. NEEDLEMAN AND CHRISTIAN D. WUNSCH

*Department of Biochemistry, Northwestern University, and
Nuclear Medicine Service, V. A. Research Hospital
Chicago, Ill. 60611, U.S.A.*

(Received 21 July 1969)

A computer adaptable method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins. This information is used to trace their possible evolutionary development.

The maximum match is a number dependent upon the similarity of the sequences. One of its definitions is the largest number of amino acids of one protein that can be matched with those of a second protein allowing for all possible interruptions in either of the sequences. While the interruptions give rise to a very large number of comparisons, the method efficiently excludes from consideration those comparisons that cannot contribute to the maximum match.

Comparisons are made from the smallest unit of significance, a pair of amino acids, one from each protein. All possible pairs are represented by a two-dimensional array, and all possible comparisons are represented by pathways through the array. For this maximum match only certain of the possible pathways must be evaluated. A numerical value, one in this case, is assigned to every cell in the array representing like amino acids. The maximum match is the largest number that would result from summing the cell values of every pathway.

1. Introduction

The amino acid sequences of a number of proteins have been compared to determine whether the relationships existing between them could have occurred by chance. Generally, these sequences are from proteins having closely related functions and are so similar that simple visual comparisons can reveal sequence coincidence. Because the method of visual comparison is tedious and because the determination of the significance of a given result usually is left to intuitive rationalization, computer-based statistical approaches have been proposed (Fitch, 1966; Needleman & Blair, 1969).

Direct comparison of two sequences, based on the presence in both of corresponding amino acids in an identical array, is insufficient to establish the full genetic relationships between the two proteins. Allowance for gaps (Braunitzer, 1965) greatly multiplies the number of comparisons that can be made but introduces unnecessary and partial comparisons.

2. A General Method for Sequence Comparison

The smallest unit of comparison is a pair of amino acids, one from each protein. The maximum match can be defined as the largest number of amino acids of one protein that can be matched with those of another protein while allowing for all possible deletions.

Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Considere as sequências:

TGCAA

TGGCTA

	-	T	G	C	A	A
-	0					
T						
G						
G						
C						
T						
A						

Considere os seguintes custos:

Gap = 10

Match = 0

Mismatch = 1

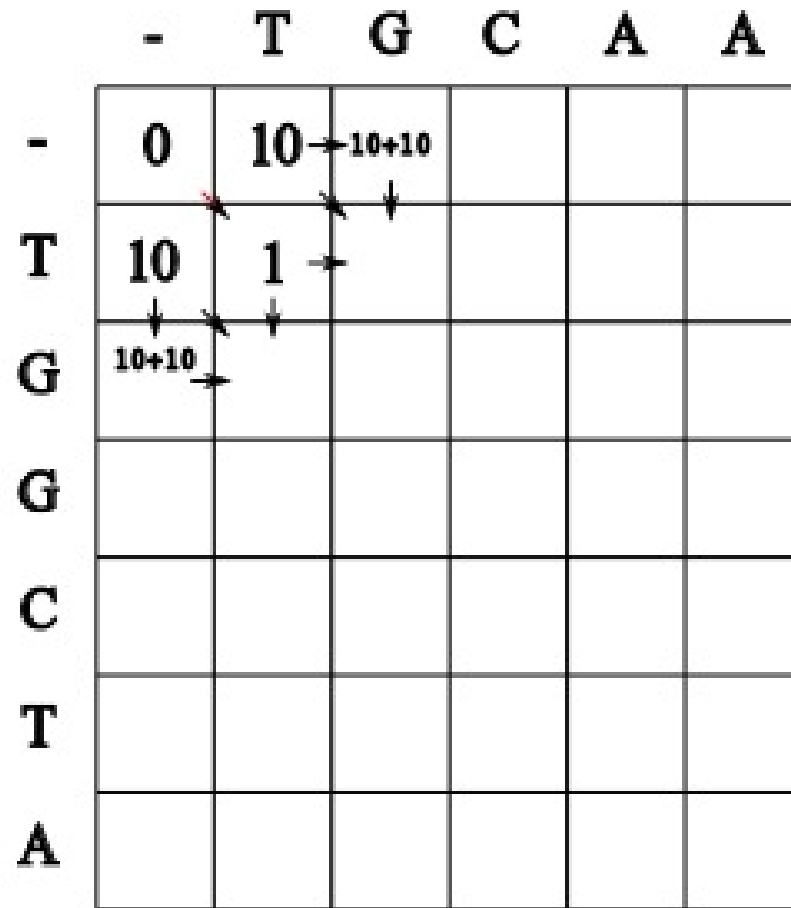
Down wave:

	-	T	G	C	A	A
-	0 → 0+10					
T	↓ 0+10					
G						
G						
C						
T						
A						

Down wave:

	-	T	G	C	A	A
-	0	10				
T	10	$10+10$ $0+1$ $10+10$				
G						
G						
C						
T						
A						

Down wave:



Down wave:

	-	T	G	C	A	A
-	0	10	20			
T	10	1	$20+10$ $10+1$ $10+1$			
G	20	$1+10$ $10+1$ $20+10$				
G						
C						
T						
A						

Down wave:

		-	T	G	C	A	A
-	0	10	20	30	40	50	
T	10	1	11				
G	20	11	$\begin{matrix} 11+10 \\ 1+0 \\ 11+10 \end{matrix}$				
G	30						
C	40						
T	50						
A	60						

Down wave:

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	30+10 20+1 11+10	40+10 30+1 21+10	50+10 40+1 31+10
G	20	11	1	21+10 11+1 1+10	31+10 21+1 11+10	41+10 31+1 21+10
G	30	11+10 20+1 30+10	1+10 11+0 21+10	11+10 1+1 11+10	21+10 11+1 2+10	31+10 21+1 12+10
C	40	21+10 30+1 40+10	11+10 21+1 31+10	2+10 11+0 21+10	12+10 2+1 11+10	22+10 12+1 3+10
T	50	31+10 40+0 60+10	21+10 31+1 40+10	11+10 21+10 31+10	3+10 11+1 21+10	13+10 3+1 12+10
A	60	40+10 50+1 60+10	31+10 40+1 50+10	21+10 31+10 41+10	12+10 21+0 31+10	4+10 12+0 21+10

Down wave:

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	30+10 20+1 11+10	40+10 30+1 21+10	50+10 40+1 31+10
G	20	11	1	21+10 11+1 1+10	31+10 21+1 11+10	41+10 31+1 21+10
G	30	11+10 20+1 30+10	1+10 11+0 21+10	11+10 1+1 11+10	21+10 11+1 2+10	31+10 21+1 12+10
C	40	21+10 30+1 40+10	11+10 21+1 31+10	2+10 11+0 21+10	12+10 2+1 11+10	22+10 12+1 3+10
T	50	31+10 40+0 60+10	21+10 31+1 40+10	11+10 21+10 31+10	3+10 11+1 21+10	13+10 3+1 12+10
A	60	40+10 50+1 60+10	31+10 40+1 50+10	21+10 31+10 41+10	12+10 21+0 31+10	4+10 12+0 21+10

Down wave:

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

Down wave:

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

Down wave:

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

Up wave:

Gap = 10
Match = 0
Mismatch = 1

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

A

A

Up wave:

Gap = 10
Match = 0
Mismatch = 1

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

AA

TA

Up wave:

Gap = 10
Match = 0
Mismatch = 1

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

CAA

CTA

Up wave:

Gap = 10
Match = 0
Mismatch = 1

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

-CAA

GCTA

Up wave:

Gap = 10
Match = 0
Mismatch = 1

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

G-CAA

GGCTA

Up wave:

Gap = 10
Match = 0
Mismatch = 1

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

TG-CAA
TGGCTA = 10 + 1 = 11

Up wave:

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

TG-CAA

TGGCTA

custo=11

	-	T	G	C	A	A
-	0	10	20	30	40	50
T	10	1	11	21	31	41
G	20	11	1	11	21	31
G	30	21	11	2	12	22
C	40	31	21	11	3	13
T	50	40	31	21	12	4
A	60	50	41	31	21	12

T-GCAA

TGGCTA

custo=11

Homologia dinâmica:

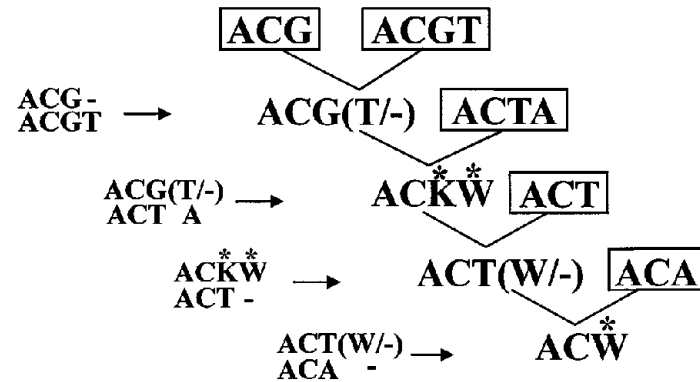


Fig. 2. Schematic of the optimization method presented here. The topology of the sequences requires three nucleotide changes and two indels. Symbols as in Fig. 1 with parentheses denoting indels.

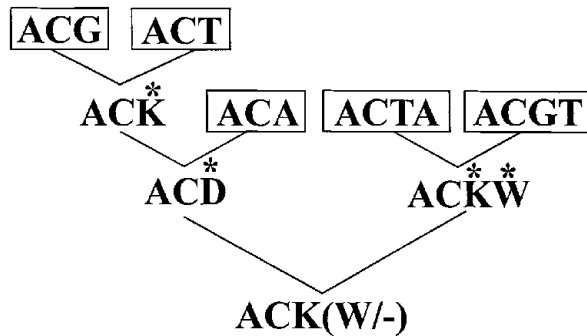


Fig. 3. Diagnosis of alternate topology. The topology of the sequences requires four nucleotide changes and a single indel. Symbols as in Fig. 2.

OPTIMIZATION ALIGNMENT: THE END OF MULTIPLE SEQUENCE ALIGNMENT IN PHYLOGENETICS?

Ward Wheeler

Department of Invertebrates, American Museum of Natural History,
Central Park West at 79th St., New York, NY 1024±5192, U.S.A.

Received for publication 10 December 1994; accepted 20 June 1995

Abstract - A method is described to assess directly the number of DNA sequence transformations, evolutionary events, required by a phylogenetic topology without the use of multiple sequence alignment. This is accomplished through a generalization of existing character optimization procedures to include insertion and deletion events (indels) in addition to base substitutions. The crux of the model is the treatment of indels as processes as opposed to the patterns implied by multiple sequence alignment. The results of this procedure are directly compatible with parsimony-based tree lengths. In addition to the simplicity of the method, it appears to generate more efficient (simpler) explanations of sequence variation than does multiple alignment.

© 1996 The Willi Hennig Society

Introduction

Phylogenetic analysis of nucleotide sequences presents problems not frequently found in other forms of character data. Although each base position presents one of four identical states (A, C, G or T), the number of these positions is likely to vary, that is homologous nucleotide sequences may differ in length. These differences may be small to non-existent in protein coding regions or extensive in non-translated sequences such as ribosomal DNAs and introns. This sequence length variation has led to the development of procedures (multiple sequence alignment) to line-up these bases via the insertion of gaps. These alignment gaps allow the nucleotide base correspondences to be interpreted as putative homologues and phylogenetic analysis to proceed.

Many methods have been proposed to accomplish multiple sequence alignment either simultaneously or sequentially, but the process is computationally intensive and there is not yet consensus on what defines a "good" or "best" multiple alignment. Methods based on the dynamic programming algorithm of Needleman and Wunsch (1970) have generated alignments based on similarity (Feng and Doolittle, 1987, 1990; Higgins and Sharp, 1988, 1989), inferred phylogenetic order (Hein, 1989, 1990), external knowledge of phylogeny (Mindell, 1991), and parsimony (Sankoff and Cedergren, 1983; Wheeler and Gladstein, 1992, 1994). In each case, alignments are created with the expressed purpose of constructing historical schemes of evolutionary relationship.

Methods which use multiple alignment follow a general pattern of observation (sequence data) to alignment (insertion of gaps) to phylogeny reconstruction (parsimony or other method). During this process, sequence "gaps" are created and treated as fifth nucleotide base although they are not observations but rather

Otimização direta:

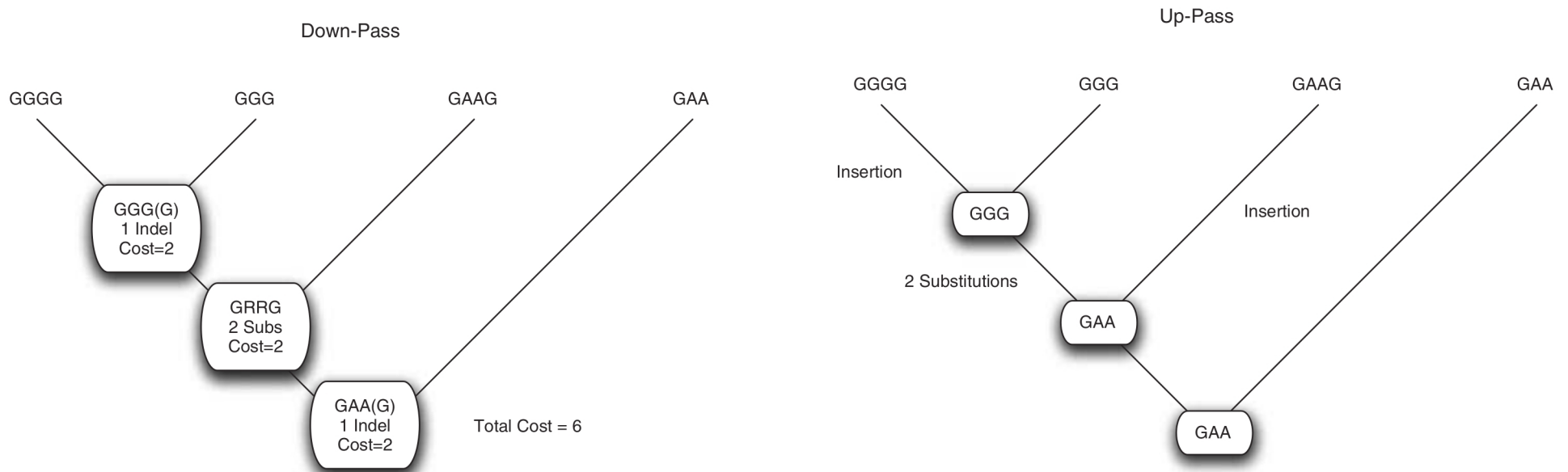
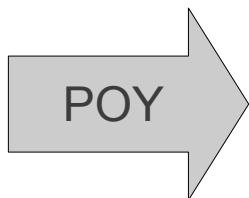


Figure 10.14: Direct Optimization (Wheeler, 1996) down-pass (left) and up-pass (right). In this example, all substitutions cost 1 and indels 2.

Homologia dinâmica:

Otimização direta:

T1 CTGGCTACGT
 T2 TGGTAAGT
 T3 CCTAAGT
 T4 CCTGATGC



Tree 0:
 ,-- T1
 |--| ,-- T2
 `--| ,-- T4
 `----- T3

T1 CTGGCT-ACGT
 T2 -T-GG- AAGT
 T3 ---CC- AAGT
 T4 ---CCTGATGC

9

Tree 1:
 ,-- T1
 |--| ,-- T3
 `--| ,-- T4
 `----- T2

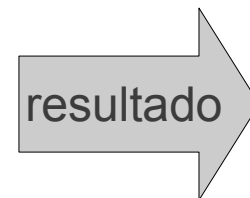
T1 -CTGGCT-ACGT
 T2 TG---GT-AAGT
 T3 -C---T-AAGT
 T4 -C---CTGATGC

10

Tree 2:
 ,-- T1
 |--| ,-- T4
 `--| ,-- T3
 `----- T2

T1 -CTGGCT-ACGT
 T2 TG---GT-AAGT
 T3 -C---T-AAGT
 T4 -C---CTGATGC

10

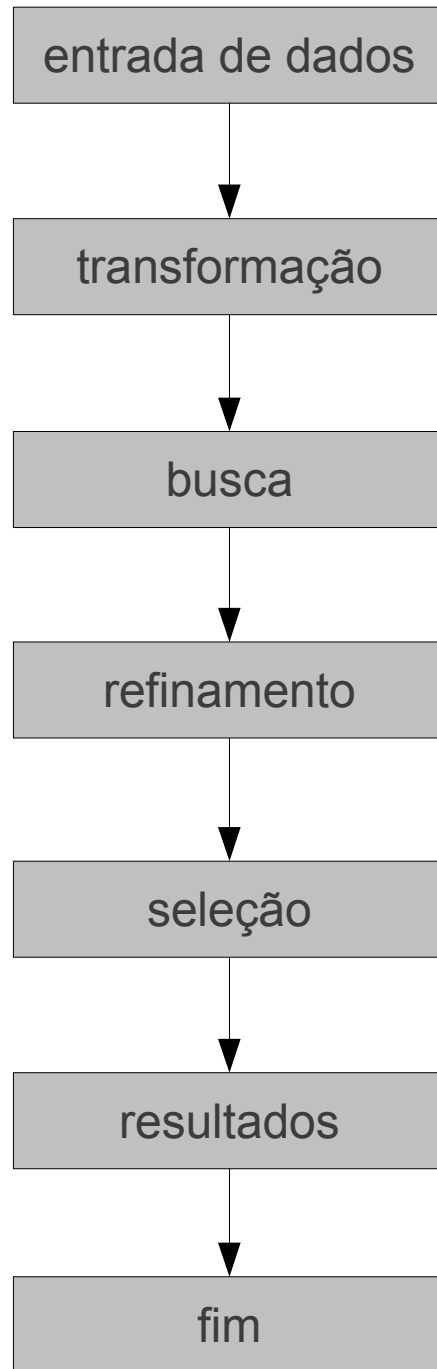


Tree 0:
 ,-- T1
 |--| ,-- T2
 `--| ,-- T4
 `----- T3

Alinhamento implícito:

T1 CTGGCT-ACGT
 T2 -T-GGT-AAGT
 T3 ---CC-TAAGT
 T4 ---CCTGATGC

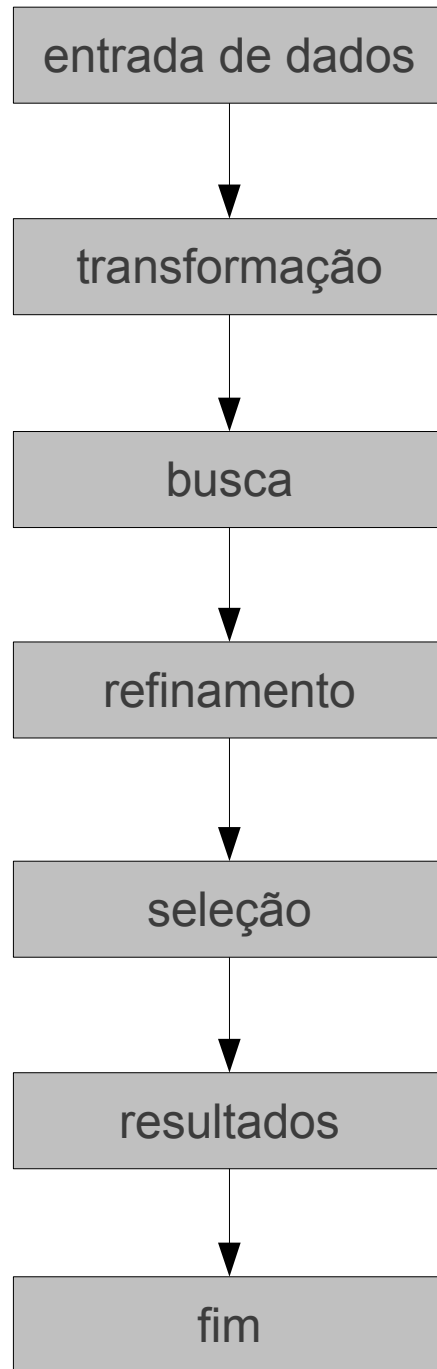
Estrutura analítica em POY:



Simple script:

```
read("data_file.fas")
transform(tcm:(1,2))
build(10)
swap()
select()
report(asciitrees)
exit()
```

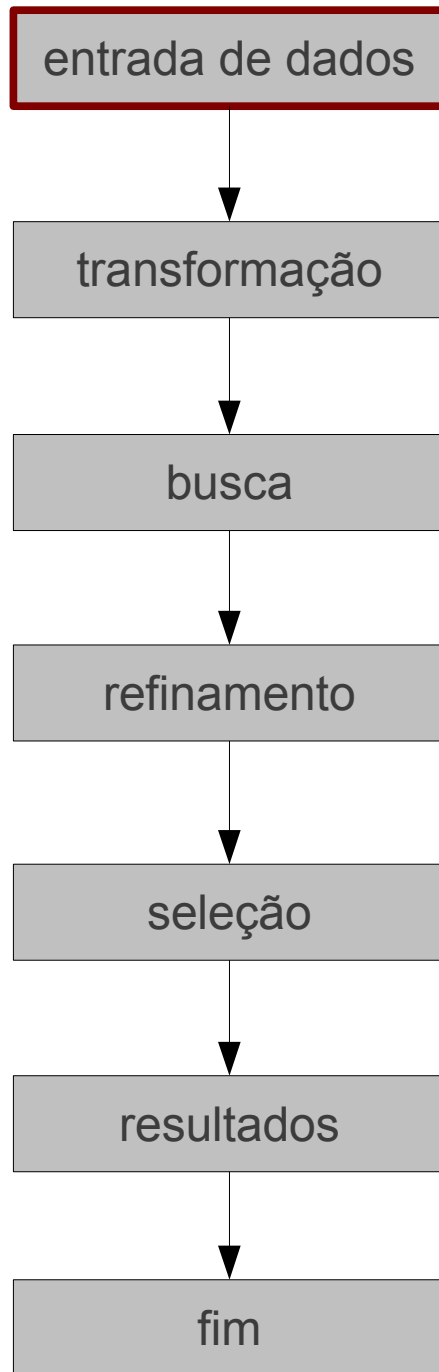
Estrutura analítica em POY:



Script mais complexo:

```
read("12s.fas","16s.fas","28s.fas")
read(prealigned:( "coi.fas", tcm:
("m0111")))
transform((names:
("12s.fas","16s.fas","28s.fas"),
(tcm:( "m0111"),gap_opening:0)))
set(root:"opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre",trees:
(nomargin),"score_do_m0111.sts",tree
stats,"search_statistics.txt",searchs
tats)
exit()
```

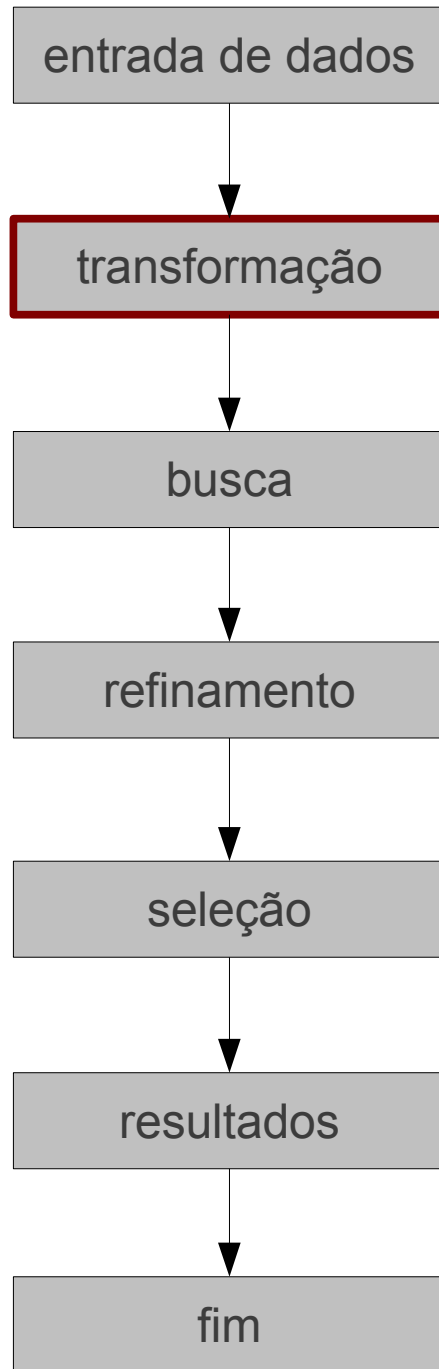
Estrutura analítica em POY:



Script mais complexo

```
read("12s.fas", "16s.fas", "28s.fas")
read(prealigned:( "coi.fas", tcm:
("m0111")))
transform((names:
("12s.fas", "16s.fas", "28s.fas"),
(tcm:( "m0111"), gap_opening:0)))
set(root:"opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre", trees:
(nomargin), "score_do_m0111.sts", tree
stats, "search_statistics.txt", searchs
tats)
exit()
```

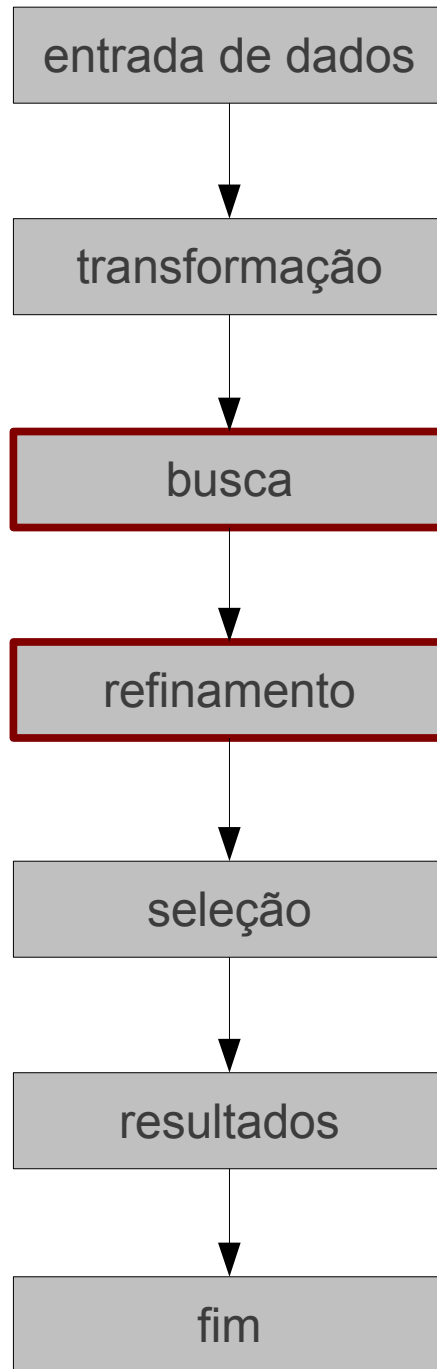
Estrutura analítica em POY:



Script mais complexo

```
read("12s.fas", "16s.fas", "28s.fas")
read(prealigned:( "coi.fas", tcm:
("m0111" )))
transform ( (names:
("12s.fas", "16s.fas", "28s.fas"),
(tcm: ("m0111"), gap_opening:0)) )
set(root: "opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre", trees:
(nomargin), "score_do_m0111.sts", tree
stats, "search_statistics.txt", searchs
tats)
exit()
```

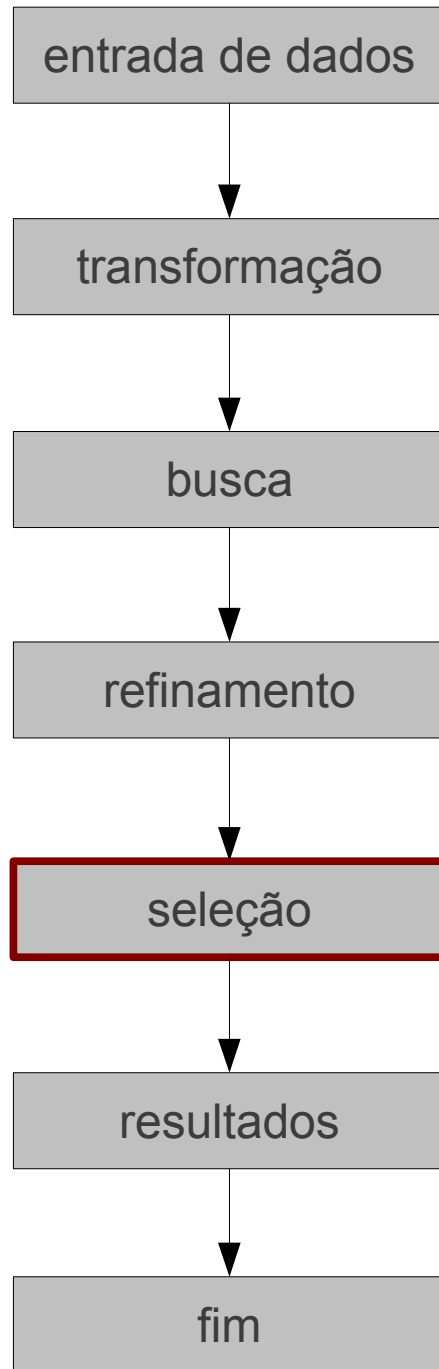
Estrutura analítica em POY:



Script mais complexo

```
read("12s.fas", "16s.fas", "28s.fas")
read(prealigned:( "coi.fas", tcm:
("m0111" )))
transform((names:
("12s.fas", "16s.fas", "28s.fas"),
(tcm:( "m0111" ), gap_opening:0)))
set(root:"opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre", trees:
(nomargin), "score_do_m0111.sts", tree
stats, "search_statistics.txt", searchs
tats)
exit()
```

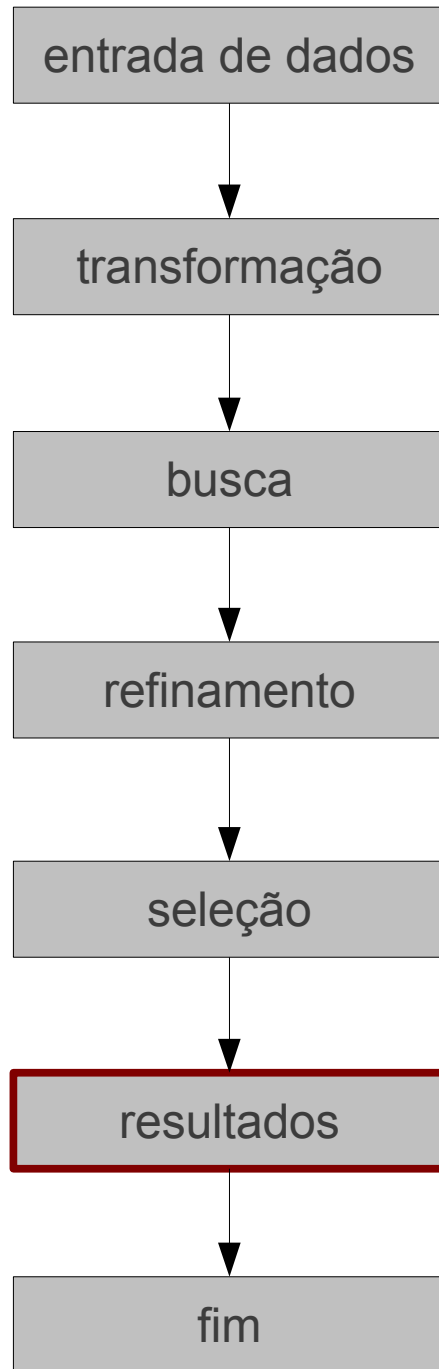
Estrutura analítica em POY:



Script mais complexo

```
read("12s.fas", "16s.fas", "28s.fas")
read(prealigned:("coi.fas", tcm:
("m0111")))
transform((names:
("12s.fas", "16s.fas", "28s.fas"),
(tcm:("m0111"), gap_opening:0)))
set(root:"opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre", trees:
(nomargin), "score_do_m0111.sts", tree
stats, "search_statistics.txt", searchs
tats)
exit()
```

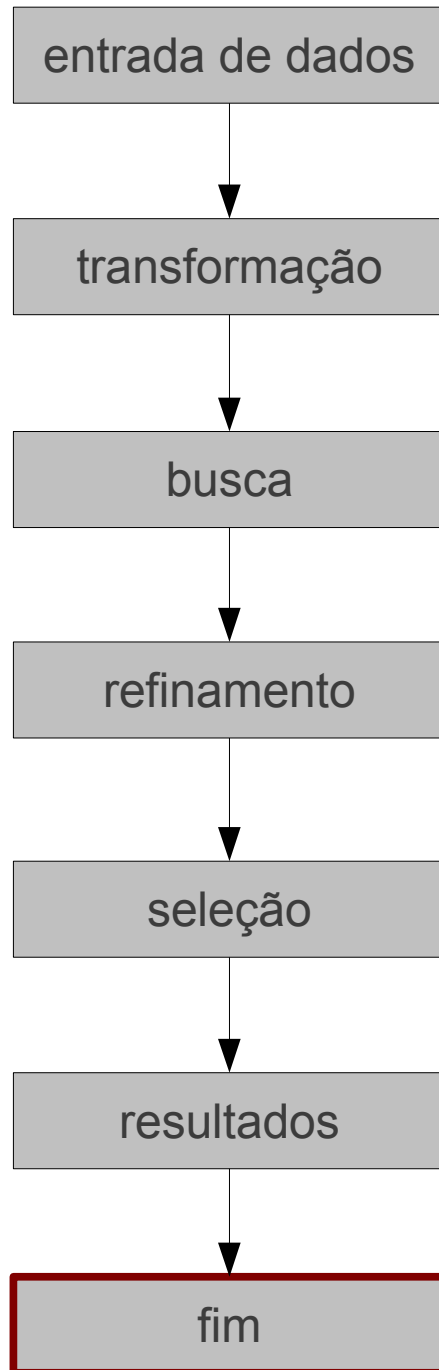

Estrutura analítica em POY:



Script mais complexo

```
read("12s.fas", "16s.fas", "28s.fas")
read(prealigned:("coi.fas", tcm:
("m0111")))
transform((names:
("12s.fas", "16s.fas", "28s.fas"),
(tcm:("m0111"), gap_opening:0)))
set(root:"opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre", trees:
(nomargin), "score_do_m0111.sts", tree
stats, "search_statistics.txt", searchs
tats)
exit()
```

Estrutura analítica em POY:



Script mais complexo

```
read("12s.fas", "16s.fas", "28s.fas")
read(prealigned:("coi.fas", tcm:
("m0111")))
transform((names:
("12s.fas", "16s.fas", "28s.fas"),
(tcm:("m0111"), gap_opening:0)))
set(root:"opilio_0969")
search(max_time:0:04:00)
select()
report("trees_do_search.tre", trees:
(nomargin), "score_do_m0111.sts", tree
stats, "search_statistics.txt", searchs
tats)
exit()
```