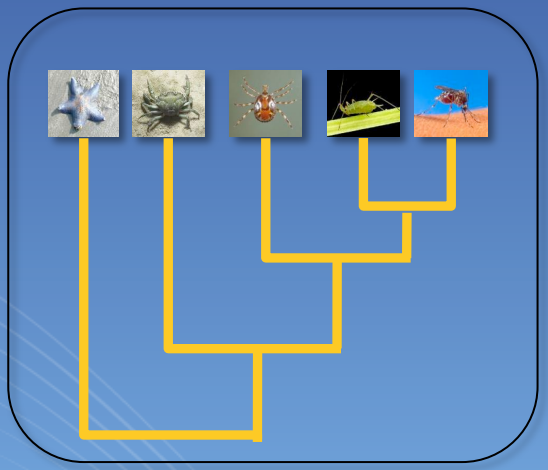
CIBIV MFPL

Applied Bioinformatics

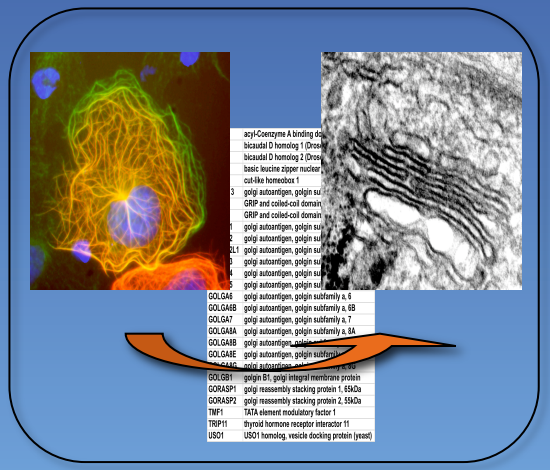
Ingo Ebersberger and Tina Koestler

CIBIV MFPL

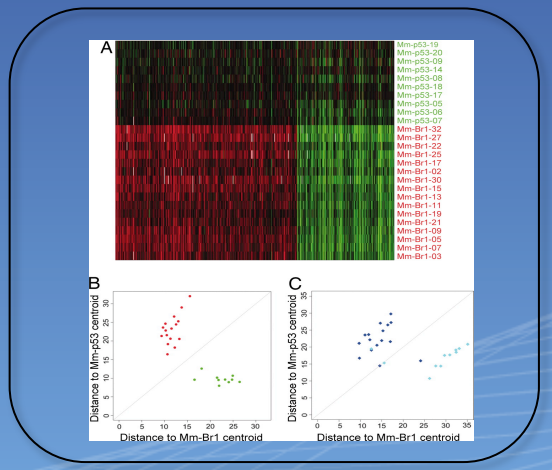
Typical applications of bioinformatics



Reconstruction
(Phylogeny)



Prediction
(Function)

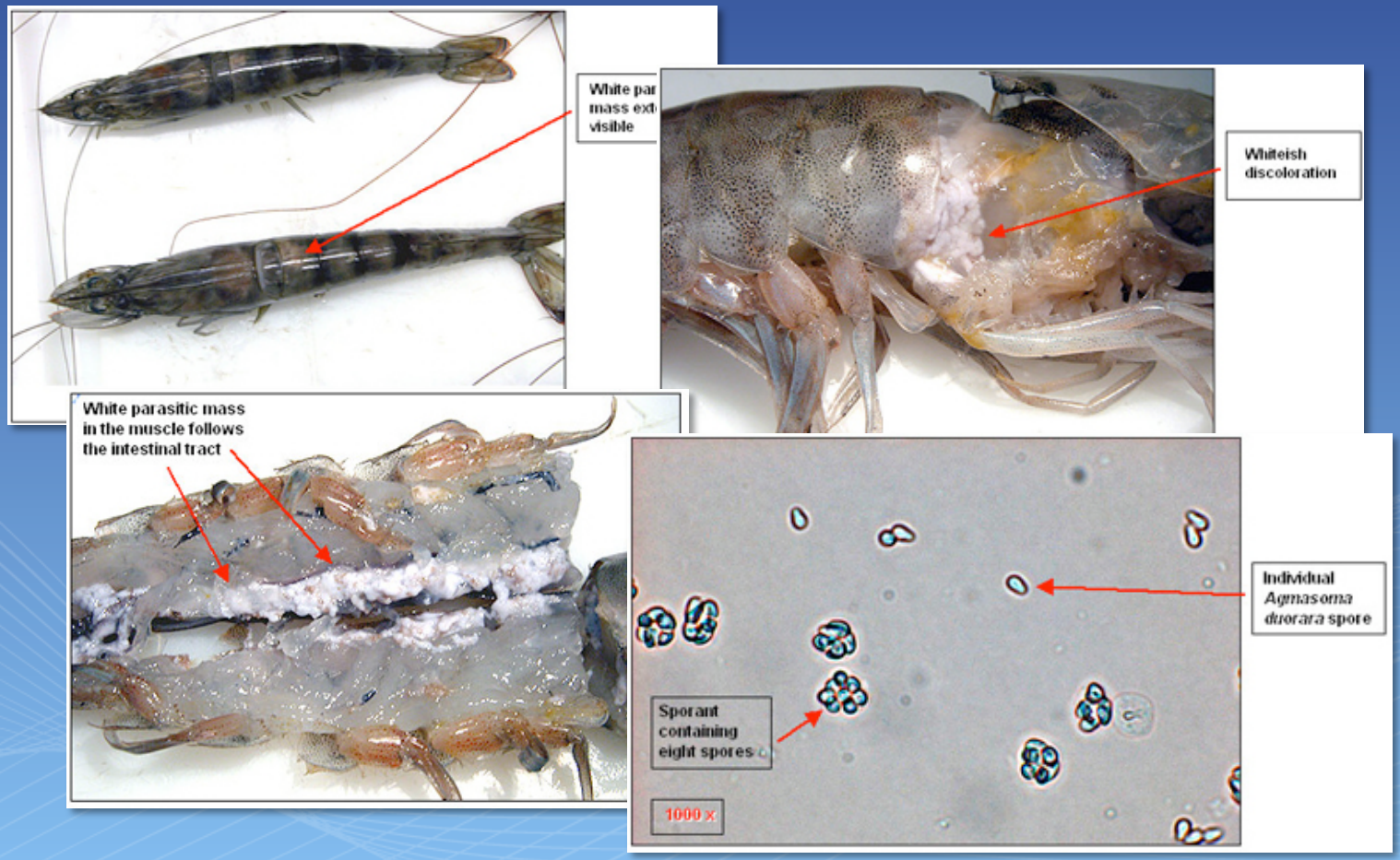


Characterization
(Gene sets)

Datamanagement/Visualization

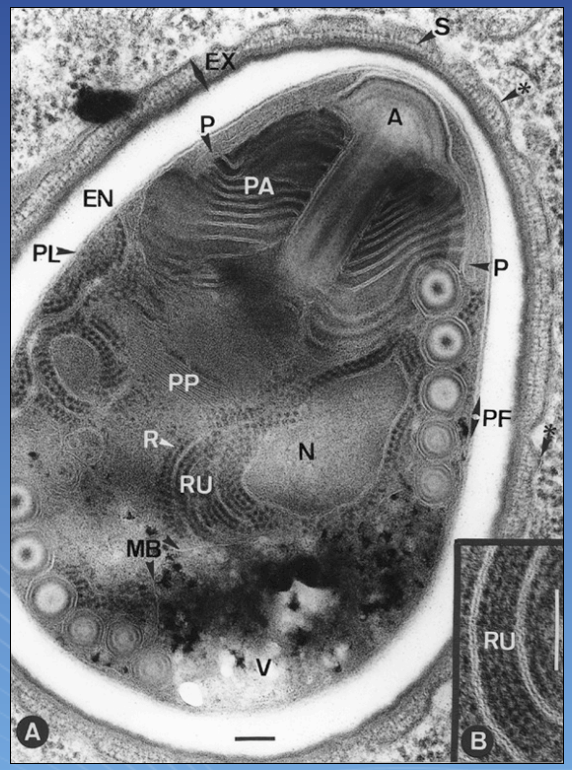
CIBIV MFPL

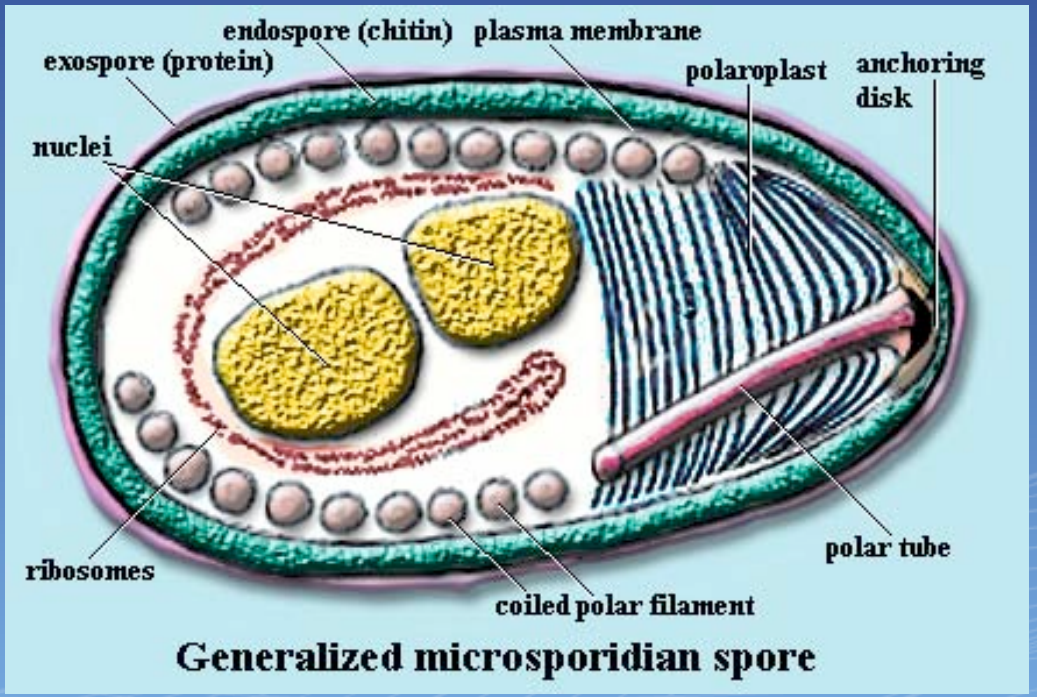
Our research object



CIBIV MFPL

Our research object

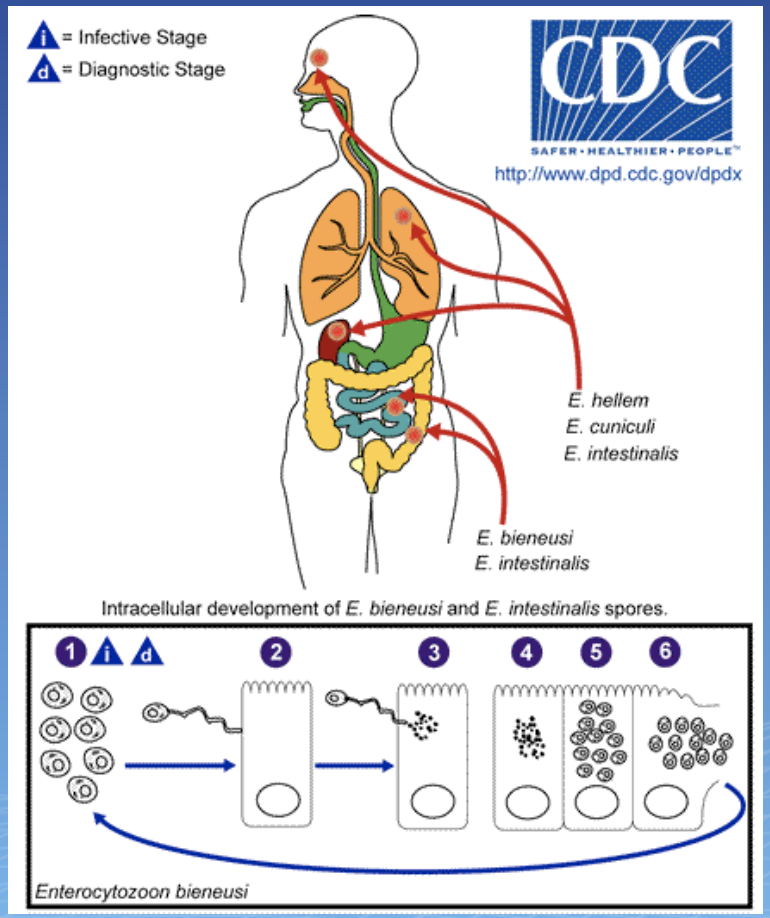




MICROSPORIDIA

CIBIV MFPL

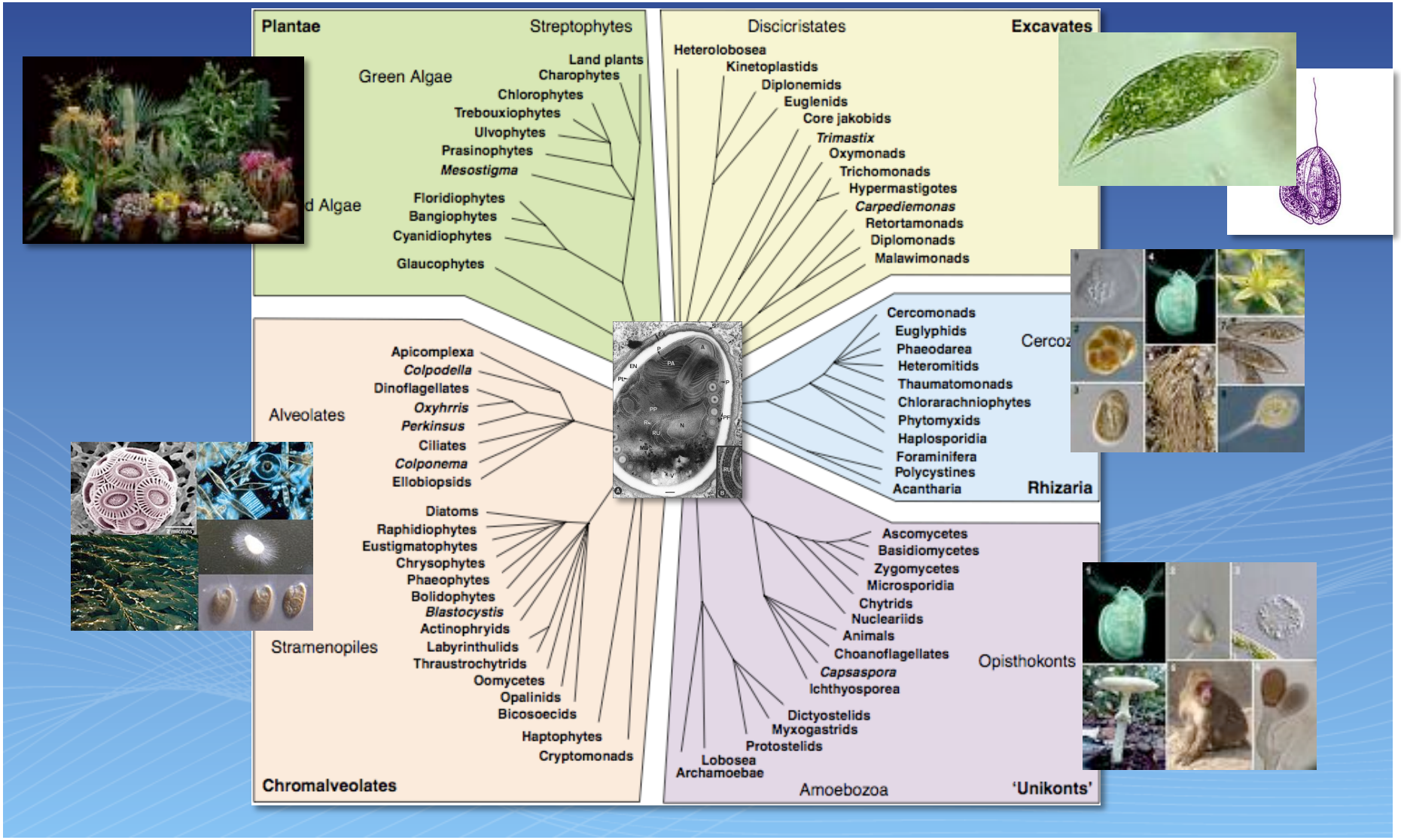
Our research object



Life cycle of Microsporidia

CIBIV MFPL

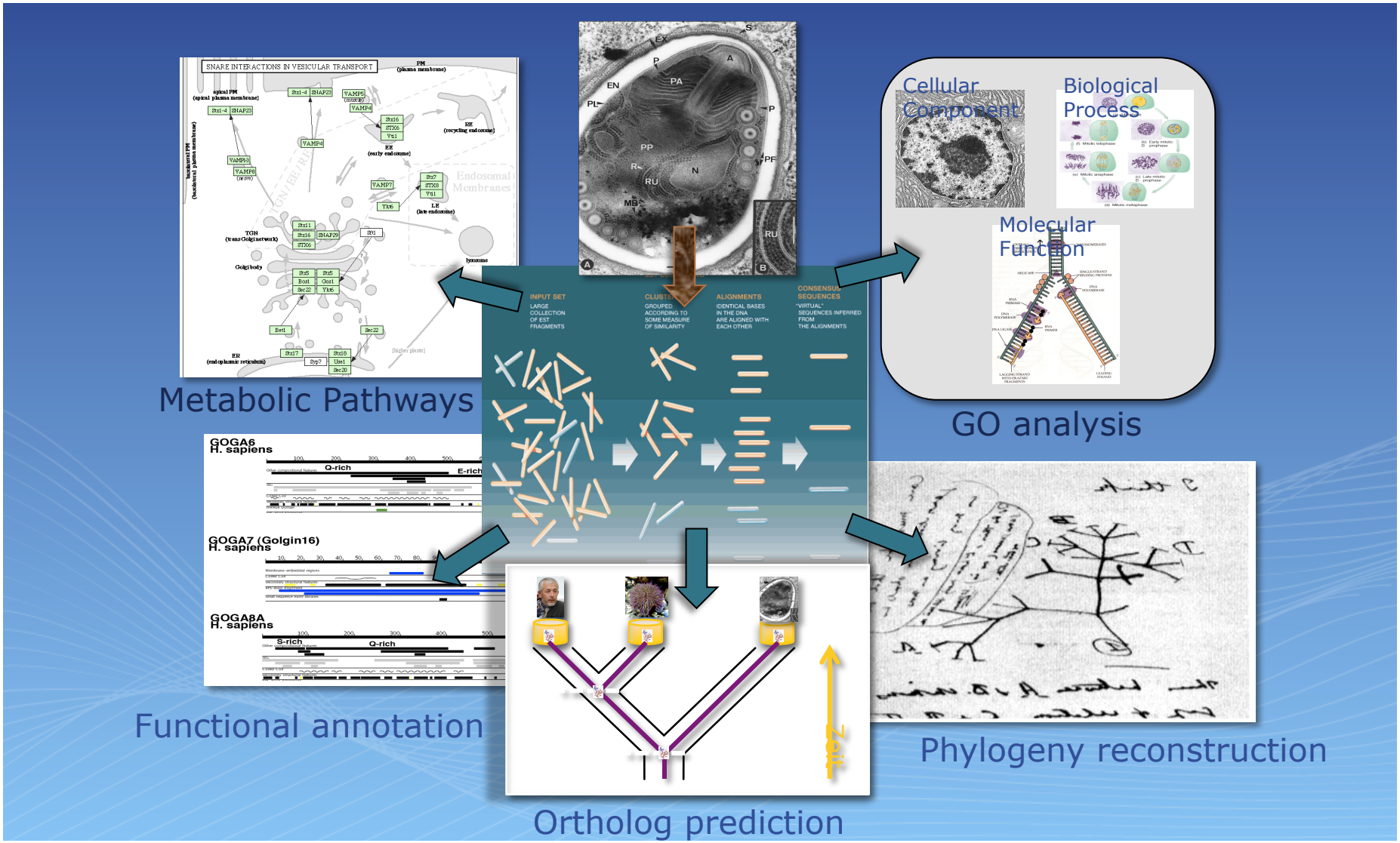
What are Microsporidia?



Modified from Keeling et al., (2005) TRENDS in Ecology and Evolution Vol.20 No.12

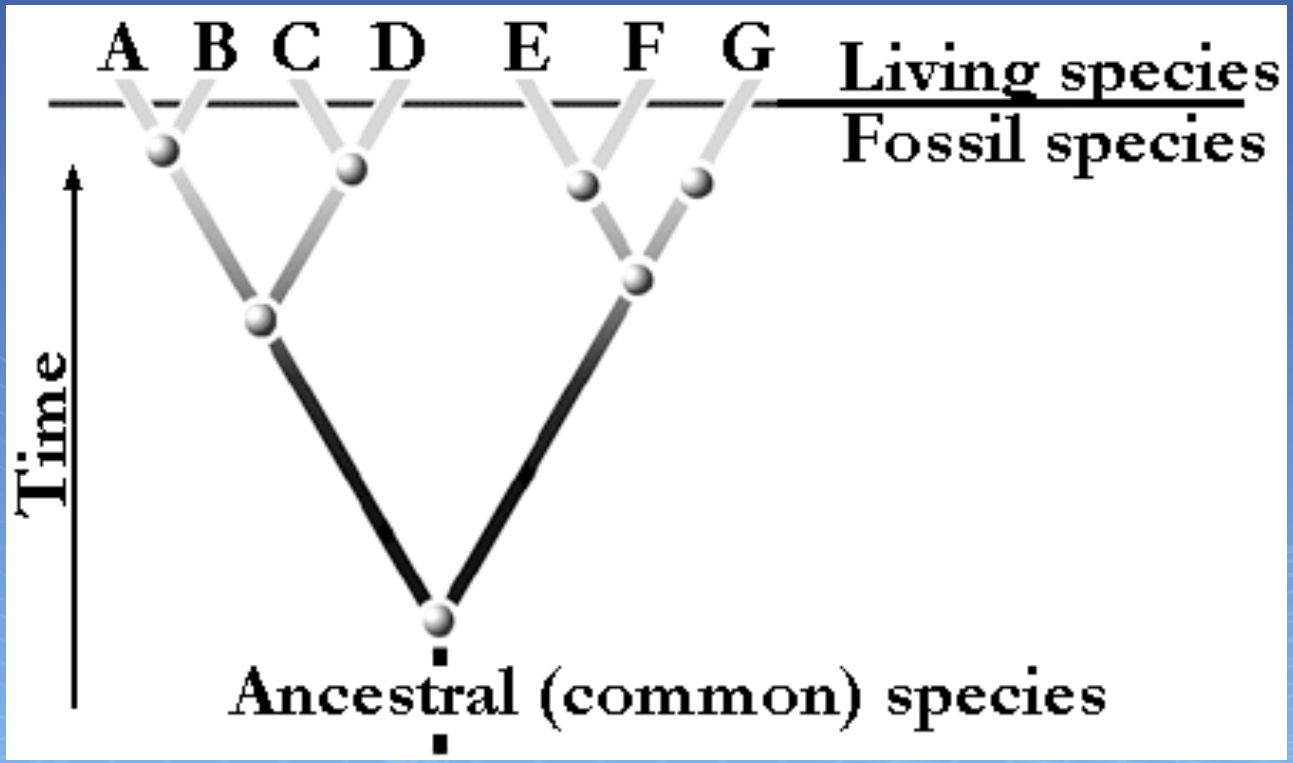
CIBIV MFPL

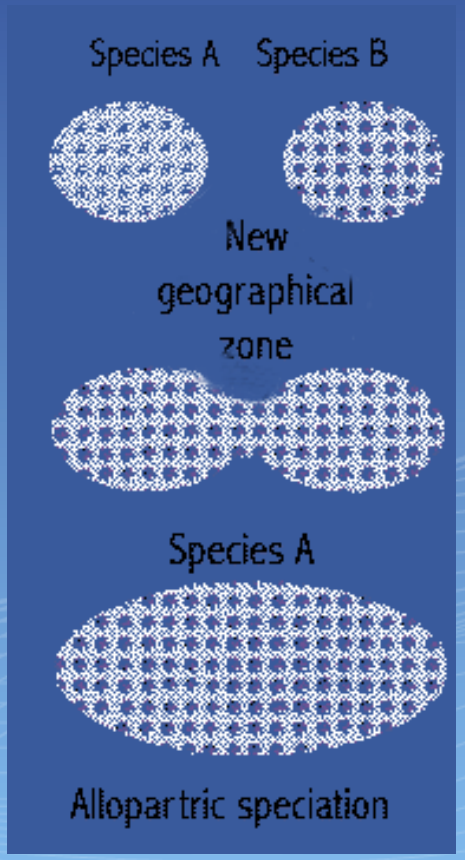
Our Workplan



CIBIV MFPL

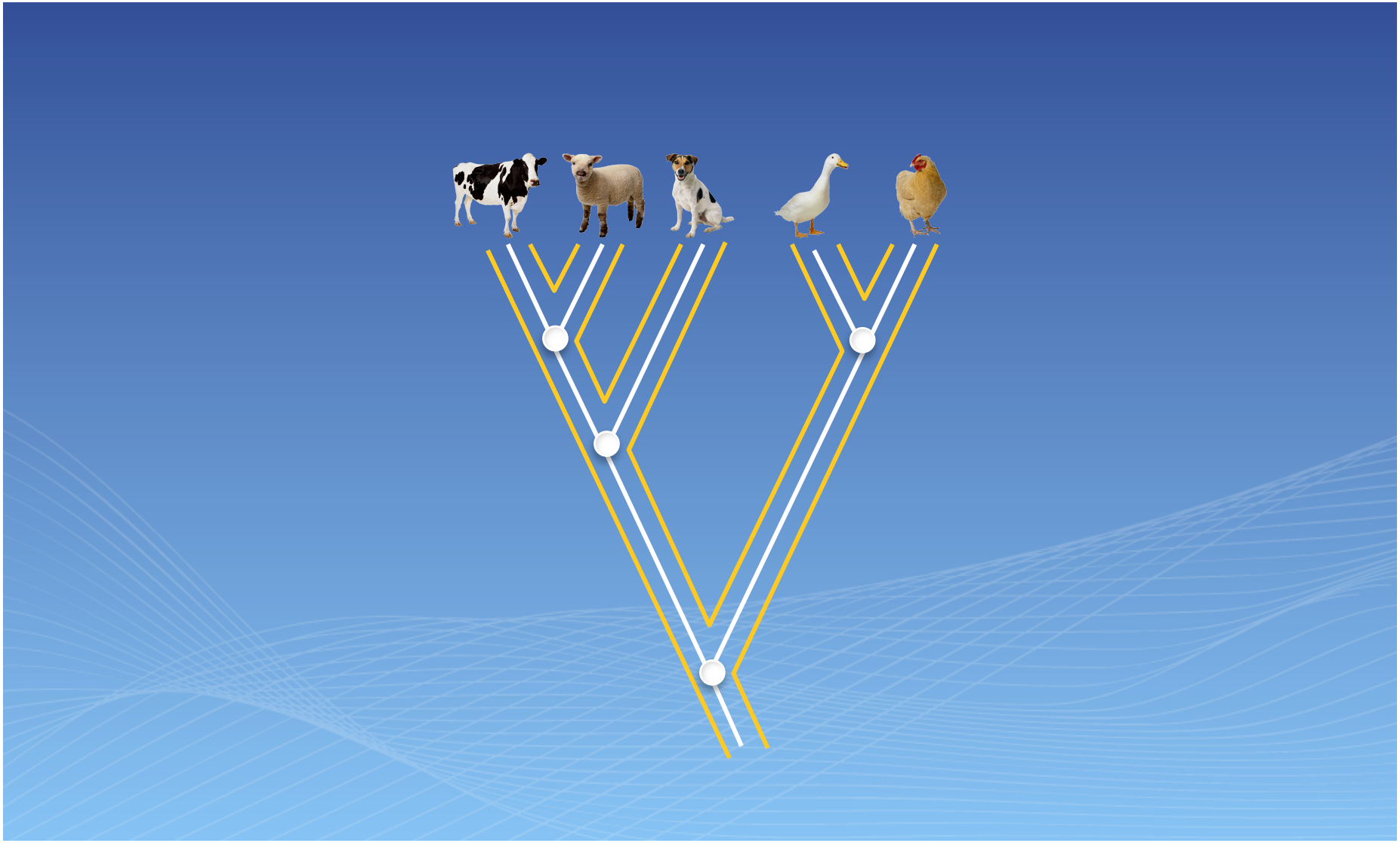
Introduction into Molecular Evolution





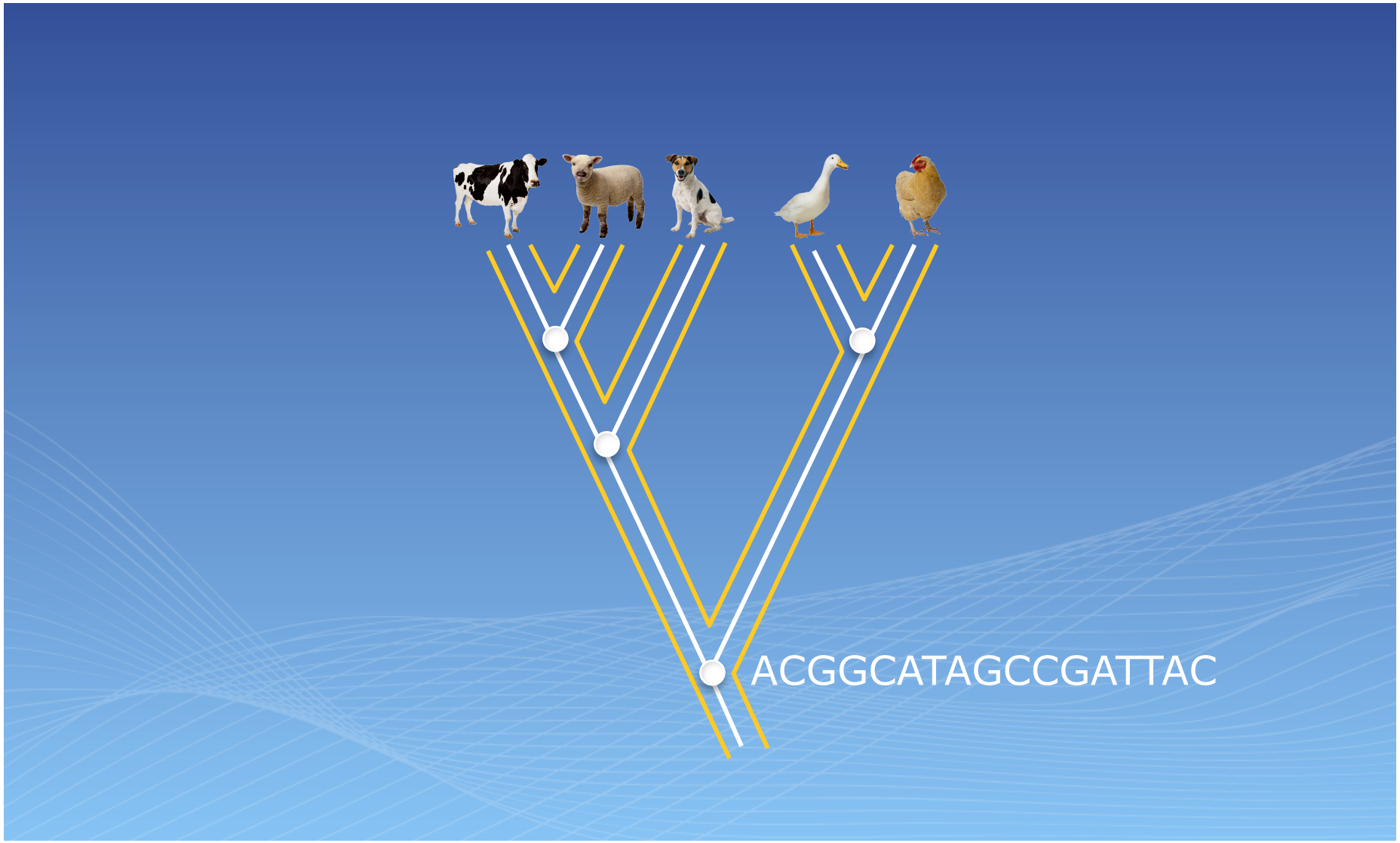
CIBIV MFPL

The evolution of biological sequences



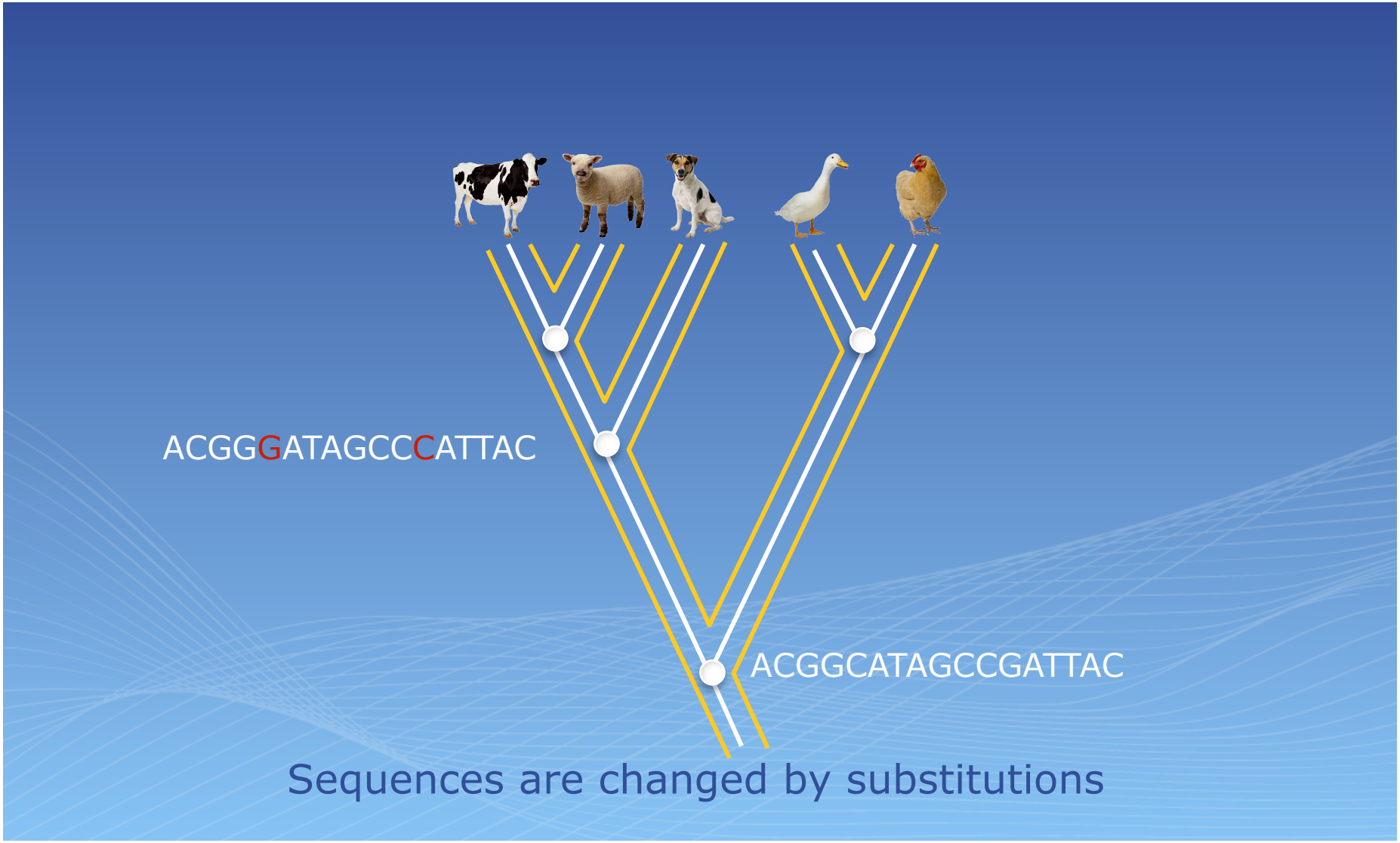
CIBIV MFPL

The evolution of biological sequences



CIBIV MFPL

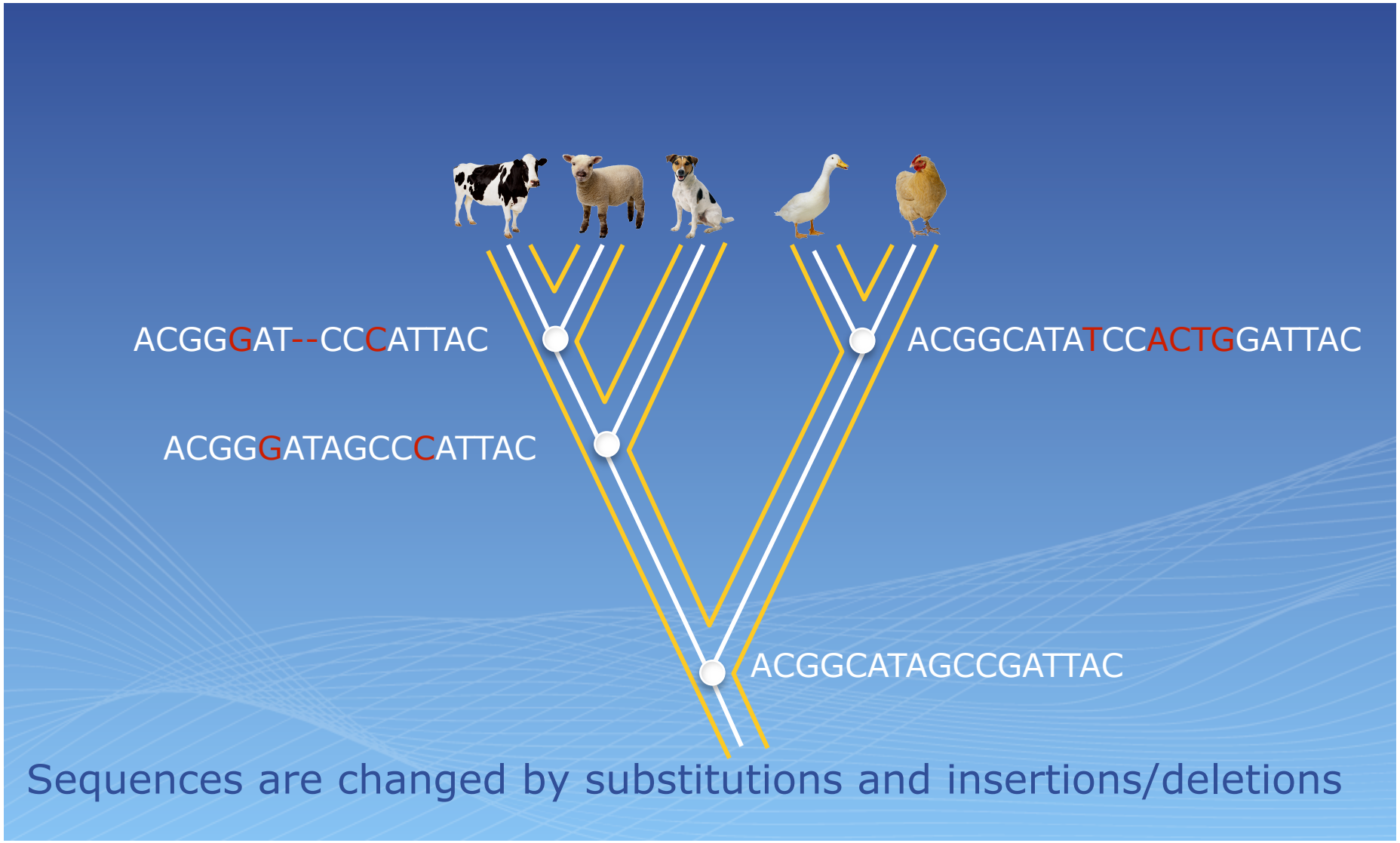
The evolution of biological sequences



Sequences are changed by substitutions

CIBIV MFPL

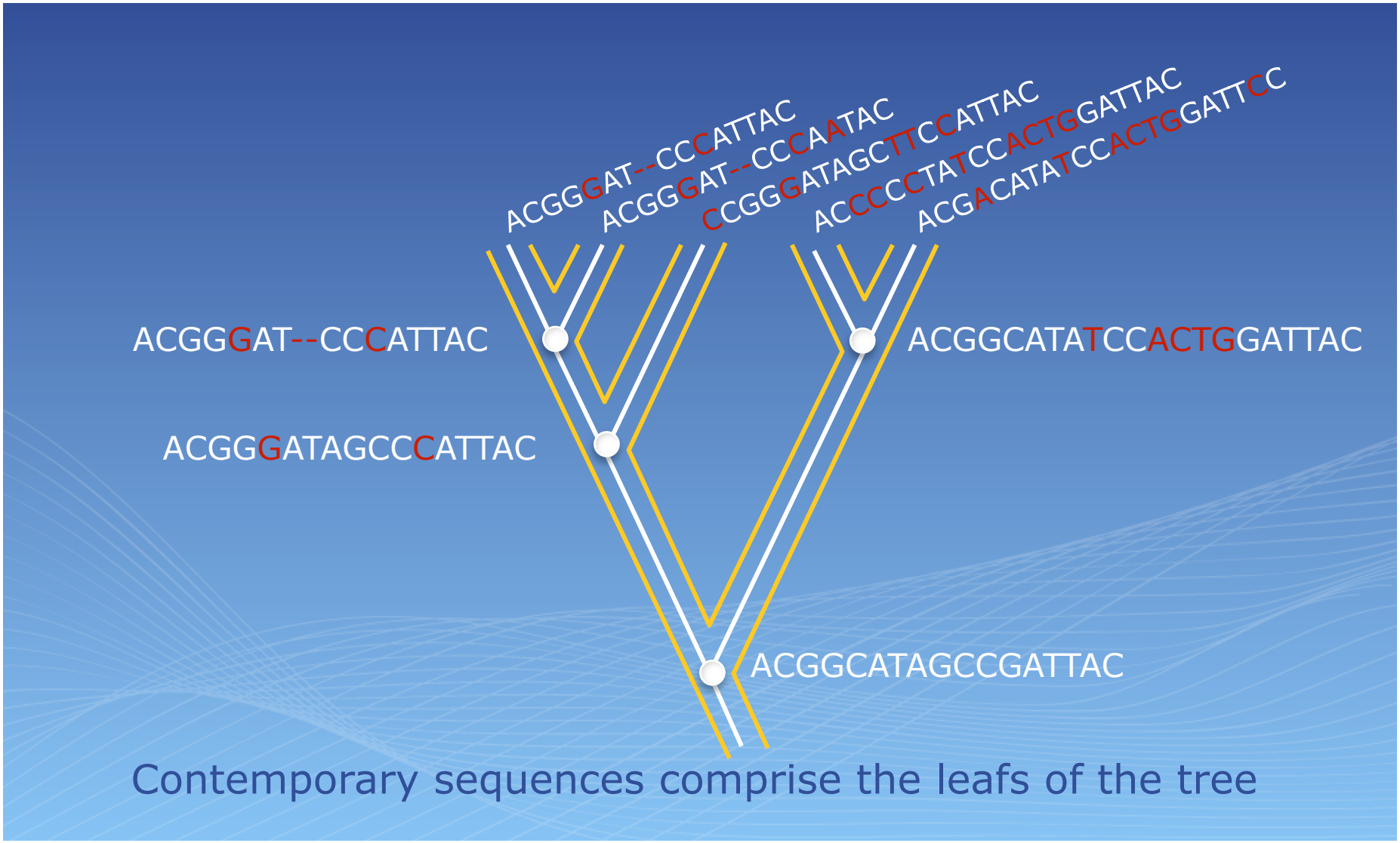
The evolution of biological sequences



Sequences are changed by substitutions and insertions/deletions

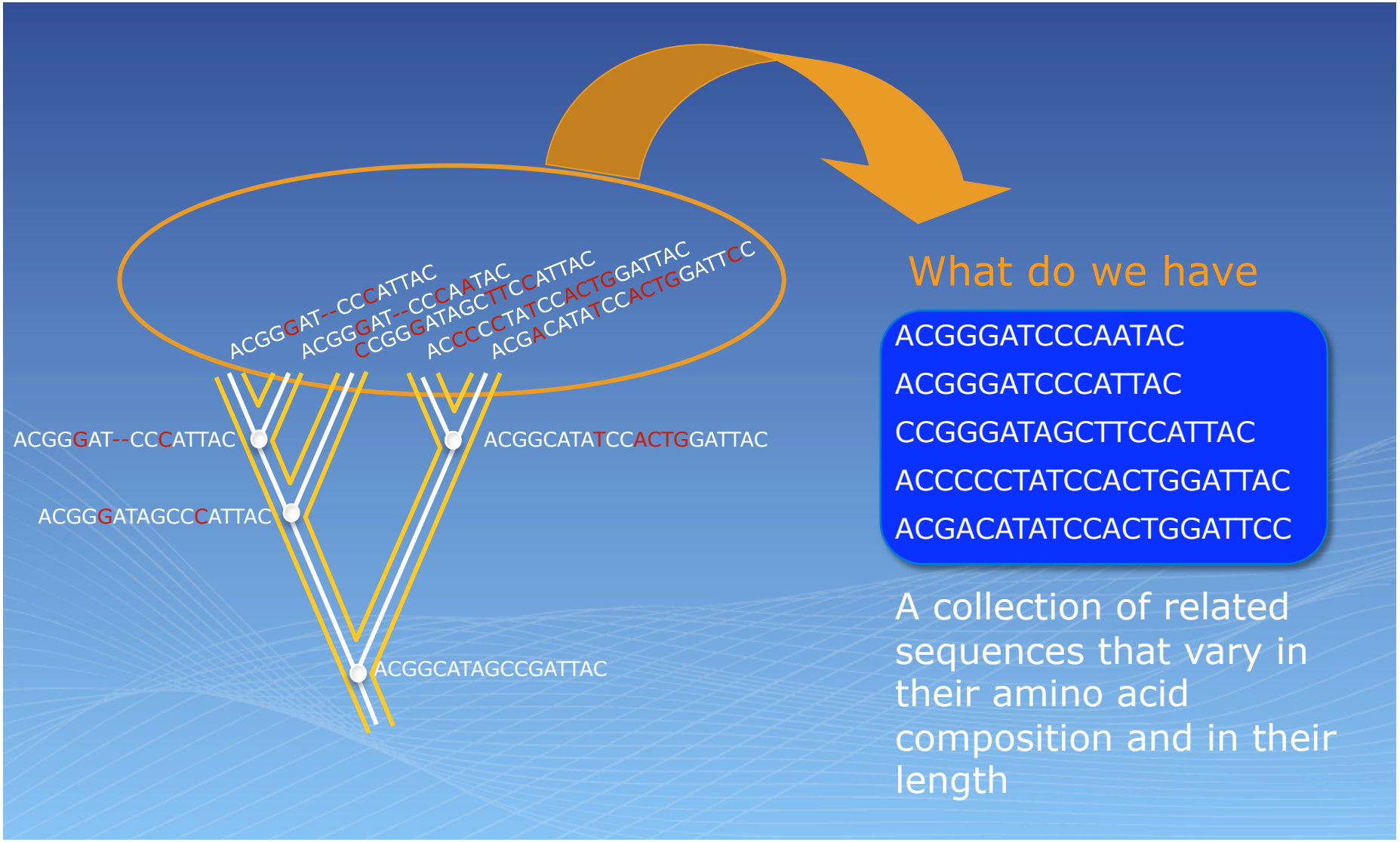
CIBIV MFPL

The evolution of biological sequences



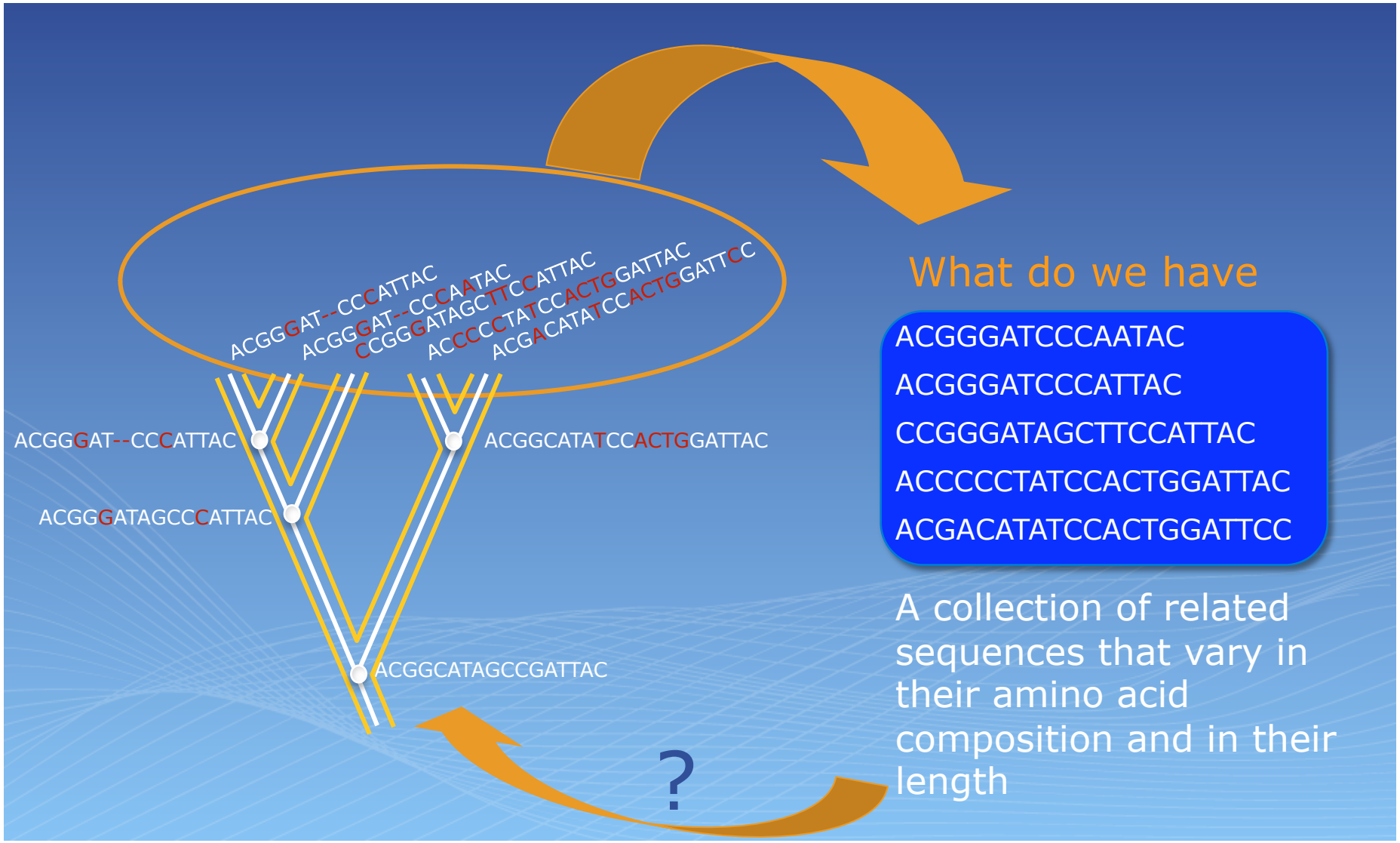
CIBIV MFPL

The evolution of biological sequences



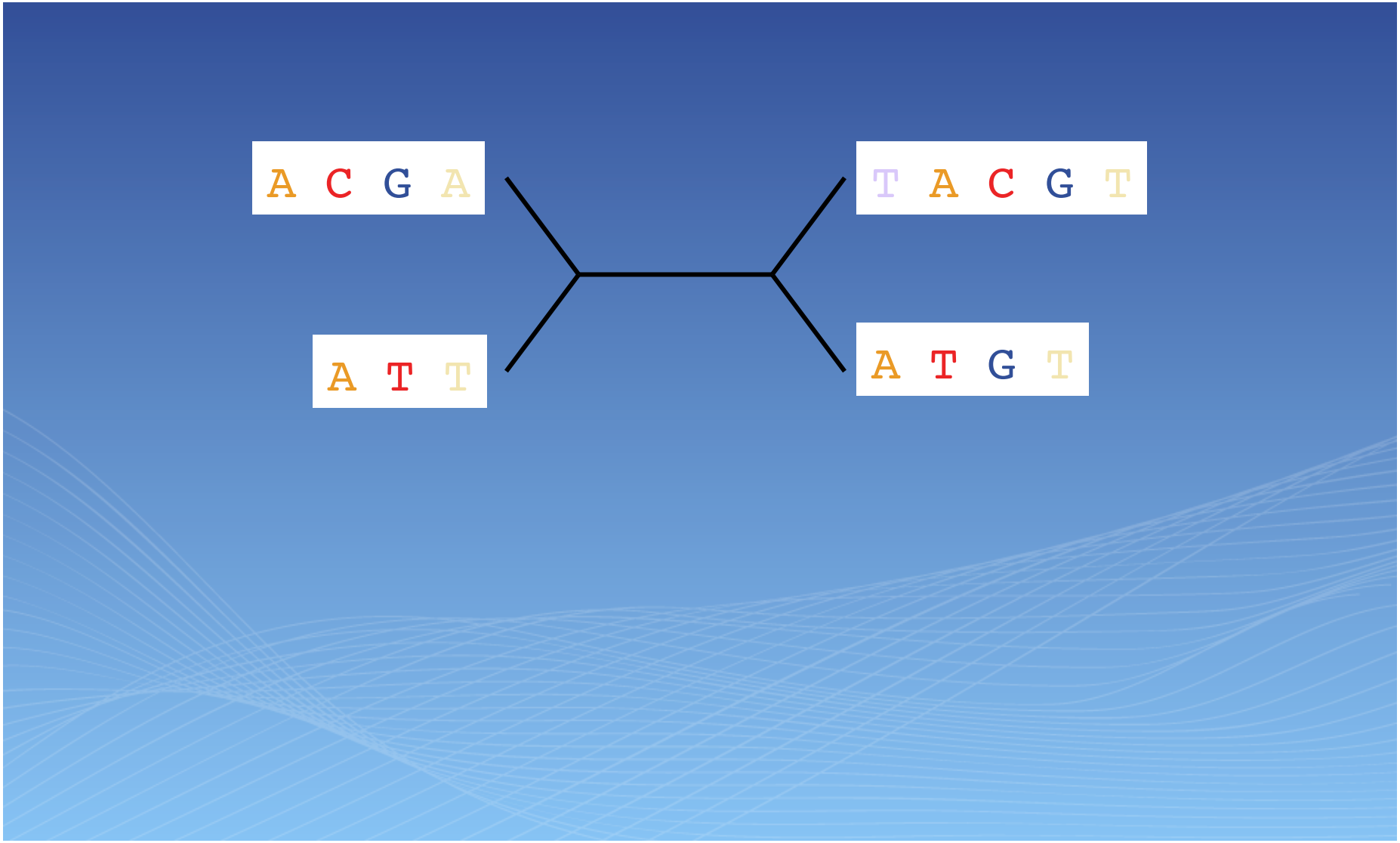
CIBIV MFPL

The evolution of biological sequences



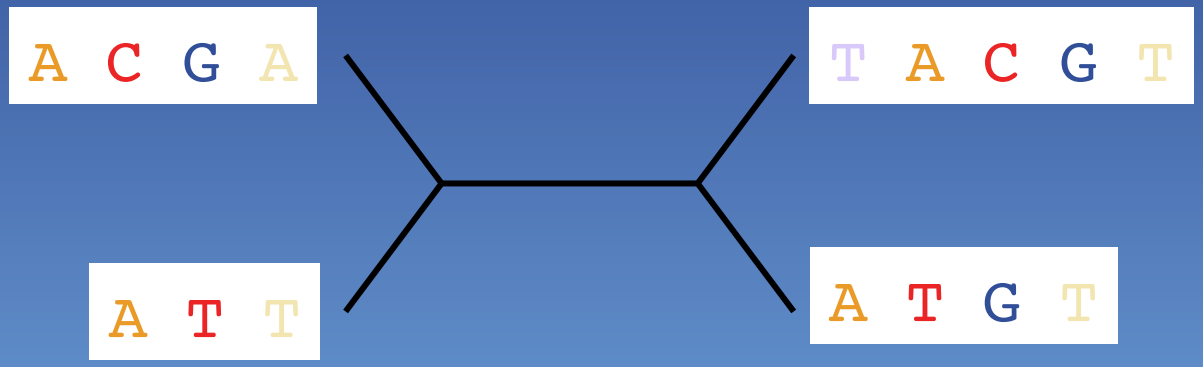
CIBIV MFPL

Finding the homologous positions



CIBIV MFPL

Finding the homologous positions




```
Seq1: - A C G A
Seq2: T A C G T
Seq3: - A T - T
Seq4: - A T G T
```

Sequence Alignment

CIBIV MFPL

Sequence Alignment

- What should a biologically correct alignment look like?
- To what extent can we define and formalize its properties?

CIBIV MFPL

Sequence Alignment

- What should a biologically correct alignment look like?
- To what extent can we define and formalize its properties?

Objective Function:

A mathematical function to measure the biological quality of an alignment...

CIBIV MFPL

Sequence Alignment

A mathematical function to measure the biological quality of an alignment α...

$$\sigma(\alpha) = \sum_{i=1}^{n} S(a_i, b_i)$$

Objective: find α that maximizes $\sigma(\alpha)$!

CIBIV MFPL

Sequence Alignment

Given two sequences A $=\{a_1,a_2,....,a_n\}$ and $B=\{b_1,b_2,....,b_m\}$ and a scoring function *S* such that

$$S(a_i,b_j)=\begin{cases}+5, if\ a_i=b_j\\ -2, if\ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

then we look for that alignment, that gives us the highest score by summing up the column scores $S(a_i,b_j)$ for all columns of the alignment.

CIBIV MFPL

Sequence Alignment

Given two sequences A $=\{a_1,a_2,....,a_n\}$ and $B=\{b_1,b_2,....,b_m\}$ and a scoring function *S* such that

$$S(a_i,b_j)=\begin{cases}+5, if\ a_i=b_j\\ -2, if\ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

then we look for that alignment, that gives us the highest score by summing up the column scores $S(a_i,b_j)$ for all columns of the alignment.

For example:

T	G	C	T	C	G	T	A	
T	-	-	T	C	A	T	A	
+5	-6	-6	+5	+5	-2	+5	+5	= 11

Many other scoring functions exist…

Blosum62:

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

PAM250:

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C Cys	12																			
S Ser	0	2																		
T Thr	-2	1	3																	
P Pro	-3	1	0	6																
A Ala	-2	1	1	1	2															
G Gly	-3	1	0	-1	1	5														
N Asn	-4	1	0	-1	0	0	2													
D Asp	-5	0	0	-1	0	1	2	4												
E Glu	-5	0	0	-1	0	0	1	3	4											
Q Gln	-5	-1	-1	0	0	-1	1	2	2	4										
H His	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F Phe	-4	-3	-3	-5	-5	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W Trp	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

CIBIV MFPL

Alignment algorithms

- There are far too many alignments for evaluating every possibility
  - number of possible pair-wise alignments: $\binom{2n}{n}$
  - for two sequences of length N=300 there are 10^{179} possibilities

Hence, we need a smart way to cut the computation short!

Dynamic programming approach for pair-wise alignments by *Needleman and Wunsch* (1970).

CIBIV MFPL

Alignment algorithms

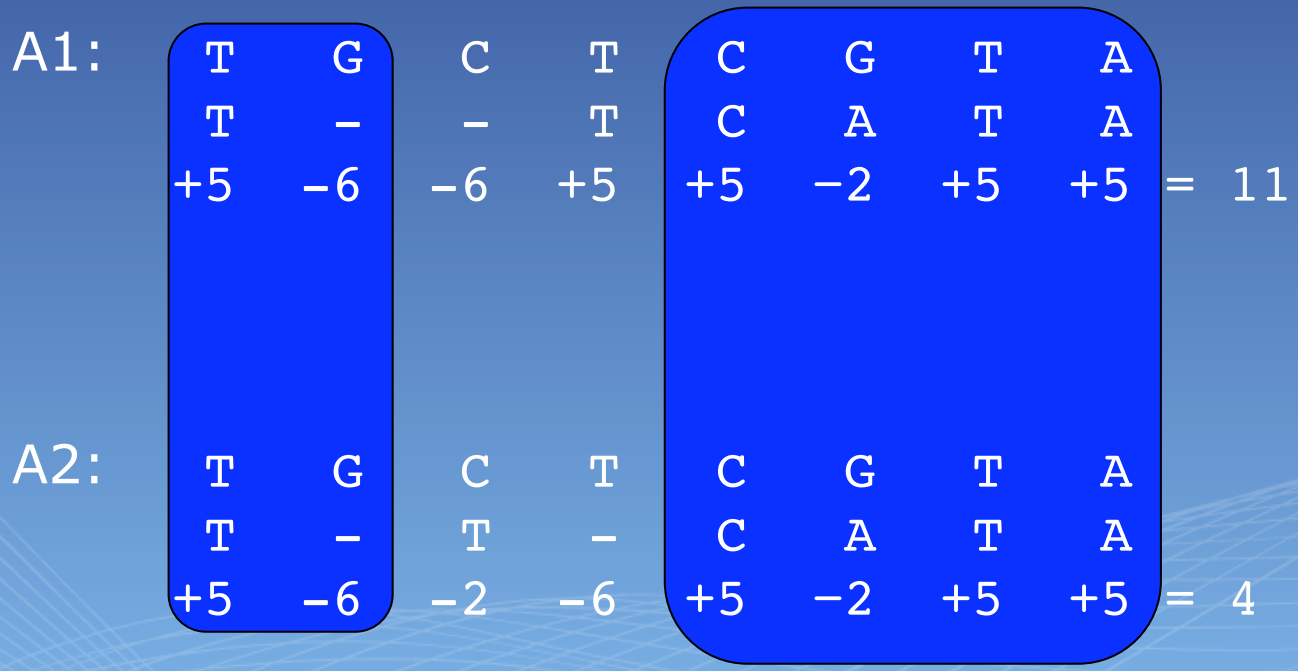
A **dynamic programming** approach usually includes:

- A mathematical description of the (biological) quality of an solution, i.e. an recursive objective function
- The computation of all intermediate values needed to obtain the globally optimal solution, thereby avoiding double-computations
- The reconstruction of the globally optimal solution from the values obtained in the previous step (backtracking)

CIBIV MFPL

Alignment algorithms

Underlying principle: Avoid redundant computation



etc...

CIBIV MFPL

Alignment algorithms

Sequence B

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0										
1	T									
2	T									
3	C									
4	A									
5	T									
6	A									

Sequence A

Scoring function

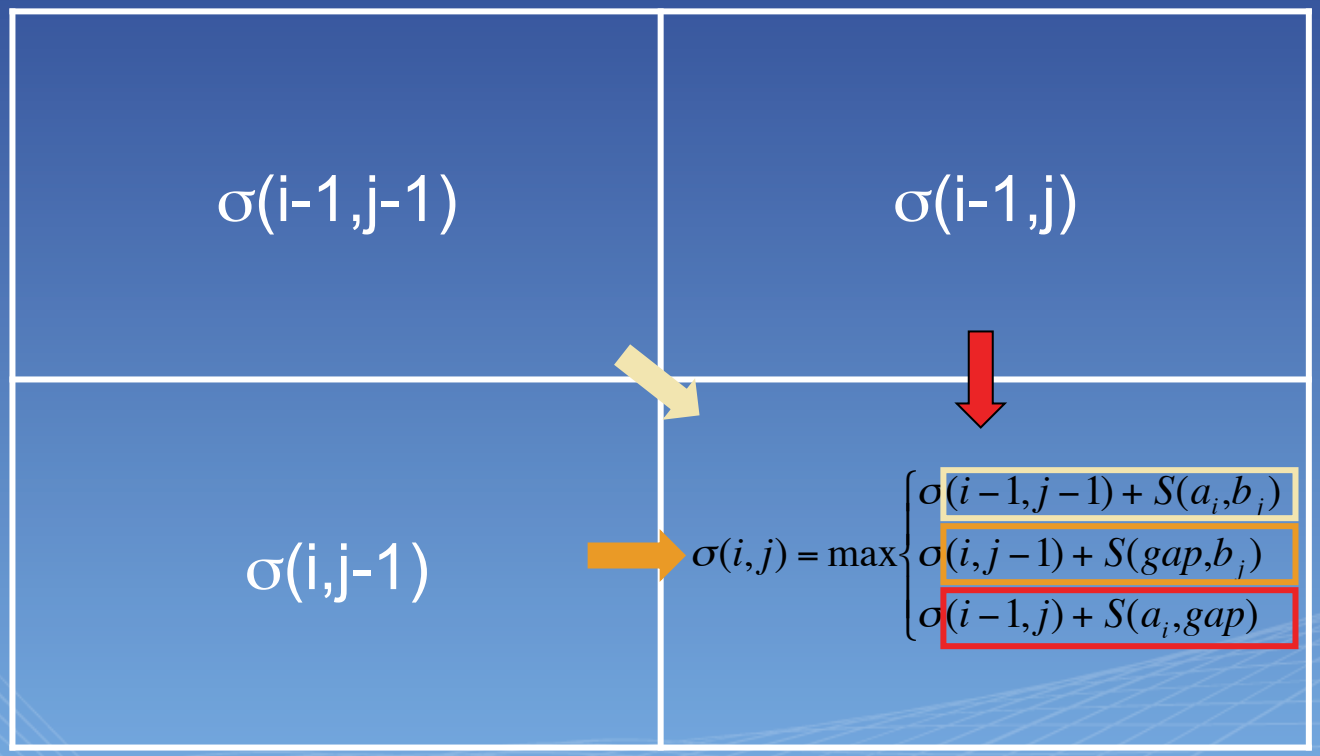
$$S(a_i,b_j)=\begin{cases}+5, if\ a_i=b_j\\ -2, if\ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

Objective function

$$\sigma(i,j)=\max\begin{cases}\sigma(i-1,j-1)+S(a_i,b_j)\\ \sigma(i,j-1)+S(gap,b_j)\\ \sigma(i-1,j)+S(a_i,gap)\end{cases}$$

CIBIV MFPL

Alignment algorithms



σ(i,j) is the optimal alignment score up to and including a_i and b_j

Alignment algorithms

Needleman-Wunsch algorithm: Initialization

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6								
2	T	−12								
3	C	−18								
4	A	−24								
5	T	−30								
6	A	−36								

$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

CIBIV MFPL

Alignment algorithms

Needleman-Wunsch algorithm: Recursion

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6	5							
2	T	−12								
3	C	−18								
4	A	−24								
5	T	−30								
6	A	−36								

$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

CIBIV MFPL

Alignment algorithms

Needleman-Wunsch algorithm: Recursion

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	–6	–12	–18	–24	–30	–36	–42	–48
1	T	–6	5	–1						
2	T	–12								
3	C	–18								
4	A	–24								
5	T	–30								
6	A	–36								

$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

CIBIV MFPL

Alignment algorithms

Needleman-Wunsch algorithm: Recursion

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6	5	−1	−7	−13	−19	−25	−31	−37
2	T	−12	−1	3	−3	−2	−8	−14	−20	−26
3	C	−18	−7	−3	8	2	3	−3	−9	−15
4	A	−24	−13	−9	2	6	0	1	−5	−4
5	T	−30	−19	−15	−4	7	4	−2	6	0
6	A	−36	−25	−21	−10	1	5	2	0	11

$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

Alignment algorithms

Needleman-Wunsch algorithm: Backtrack

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6	5	−1	−7	−13	−19	−25	−31	−37
2	T	−12	−1	3	−3	−2	−8	−14	−20	−26
3	C	−18	−7	−3	8	2	3	−3	−9	−15
4	A	−24	−13	−9	2	6	0	1	−5	−4
5	T	−30	−19	−15	−4	7	4	−2	6	0
6	A	−36	−25	−21	−10	1	5	2	0	11

*

*

CIBIV MFPL

Alignment algorithms

Needleman-Wunsch algorithm: Backtrack

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6	5	−1	−7	−13	−19	−25	−31	−37
2	T	−12	−1	3	−3	−2	−8	−14	−20	−26
3	C	−18	−7	−3	8	2	3	−3	−9	−15
4	A	−24	−13	−9	2	6	0	1	−5	−4
5	T	−30	−19	−15	−4	7	4	−2	6	0
6	A	−36	−25	−21	−10	1	5	2	0	11

A*

A*

CIBIV MFPL

Alignment algorithms

Needleman-Wunsch algorithm: Backtrack

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6	5	−1	−7	−13	−19	−25	−31	−37
2	T	−12	−1	3	−3	−2	−8	−14	−20	−26
3	C	−18	−7	−3	8	2	3	−3	−9	−15
4	A	−24	−13	−9	2	6	0	1	−5	−4
5	T	−30	−19	−15	−4	7	4	−2	6	0
6	A	−36	−25	−21	−10	1	5	2	0	11

TA*

TA*

CIBIV MFPL

Alignment algorithms

Needleman-Wunsch algorithm: Backtrack

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	T	-6	5	-1	-7	-13	-19	-25	-31	-37
2	T	-12	-1	3	-3	-2	-8	-14	-20	-26
3	C	-18	-7	-3	8	2	3	-3	-9	-15
4	A	-24	-13	-9	2	6	0	1	-5	-4
5	T	-30	-19	-15	-4	7	4	-2	6	0
6	A	-36	-25	-21	-10	1	5	2	0	11

```
*TGCTCGTA*
*T--TCATA*
```

Alignment Score: 11

CIBIV MFPL

Alignment algorithms

Smith-Waterman pair-wise local alignment

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	0	0	0	0	0	0	0	0
1	T	0	5	0	0	5	0	0	5	0
2	T	0	5	3	0	5	3	0	5	3
3	C	0	0	3	8	2	10	4	0	3
4	A	0	0	0	2	6	4	8	2	5
5	T	0	5	0	0	7	4	2	13	7
6	A	0	0	3	0	1	5	2	7	18

$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

$$\sigma(i,j)=\max\begin{cases}\sigma(i-1,j-1)+S(a_i,b_j)\\ \sigma(i,j-1)+S(gap)\\ \sigma(i-1,j)+S(gap)\\ 0\end{cases}$$

CIBIV MFPL

Alignment algorithms

Smith-Waterman pair-wise local alignment: Backtrack

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	0	0	0	0	0	0	0	0
1	T	0	5	0	0	5	0	0	5	0
2	T	0	5	3	0	5	3	0	5	3
3	C	0	0	3	8	2	10	4	0	3
4	A	0	0	0	2	6	4	8	2	5
5	T	0	5	0	0	7	4	2	13	7
6	A	0	0	3	0	1	5	2	7	18

TCGTA
TCATA

Alignment Score: 18

CIBIV MFPL

Alignment algorithms

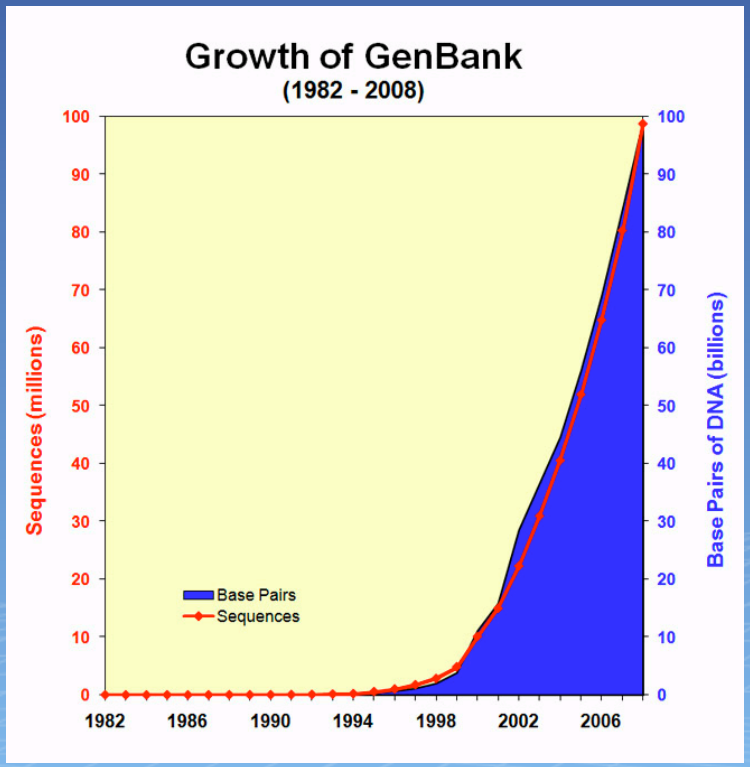
Both, Needleman-Wunsch and Smith-Waterman alignment methods are exact methods since they guarantee a globally optimal solution for the optimization problem!

Drawback: Computational expensive, i.e. O(nm) in time and memory

CIBIV MFPL

Alignment algorithms: BLAST*

BLAST uses several heuristics to reduce search space



*Gapped Blast, Altschul *et al.* (1997) Nucleic Acids Res. 25:3389-3402.

CIBIV MFPL

BLAST heuristics

1. Given a query *q* and a target sequence find substrings of length *k* (*k*-mers) of score at least *t*. *k* is normally 3 to 5 for amino acids and 12 for nucleotides.

2. Extend each hit to a *locally maximal segment*. Terminate the extension when the reduction in score exceeds a pre-defined threshold

3. Report maximal segments above score *S*.

BLAST heuristics

Finding k-mers quickly

Preprocess the database of sequences:

For each sequence in the database store all k-mers in hash-table.

This takes linear time

Query sequence:

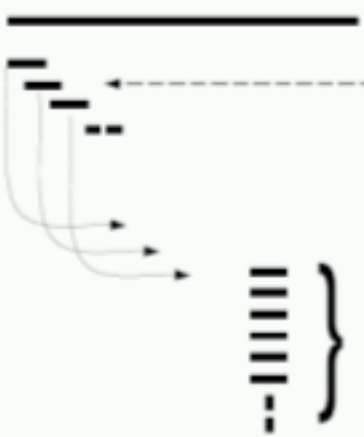
For each k-mer in the query sequence look up the hash table of the target database to see if it exists.

Also takes linear time

CIBIV MFPL

BLAST heuristics

(1) For the query find the list of high scoring words of length w.

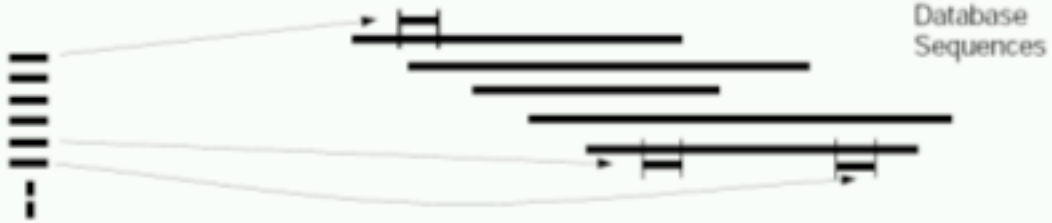


Query Sequence of length L

Maximum of L-w+1 words (typically w = 3 for proteins)

For each word from the query sequence find the list of words that will score at least T when scored using a pairscore matrix (e.g. PAM 250). For typical parameters there are around 50 words per residue of the query..

(2) Compare the word list to the database and identify exact matches.



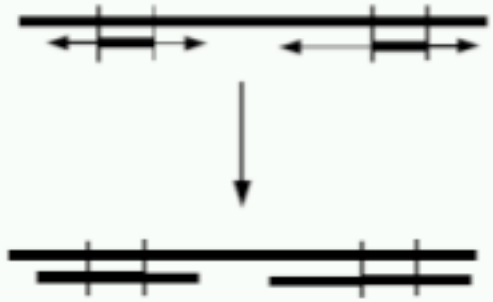
Database Sequences

Exact matches of words from word list

CIBIV MFPL

BLAST heuristics

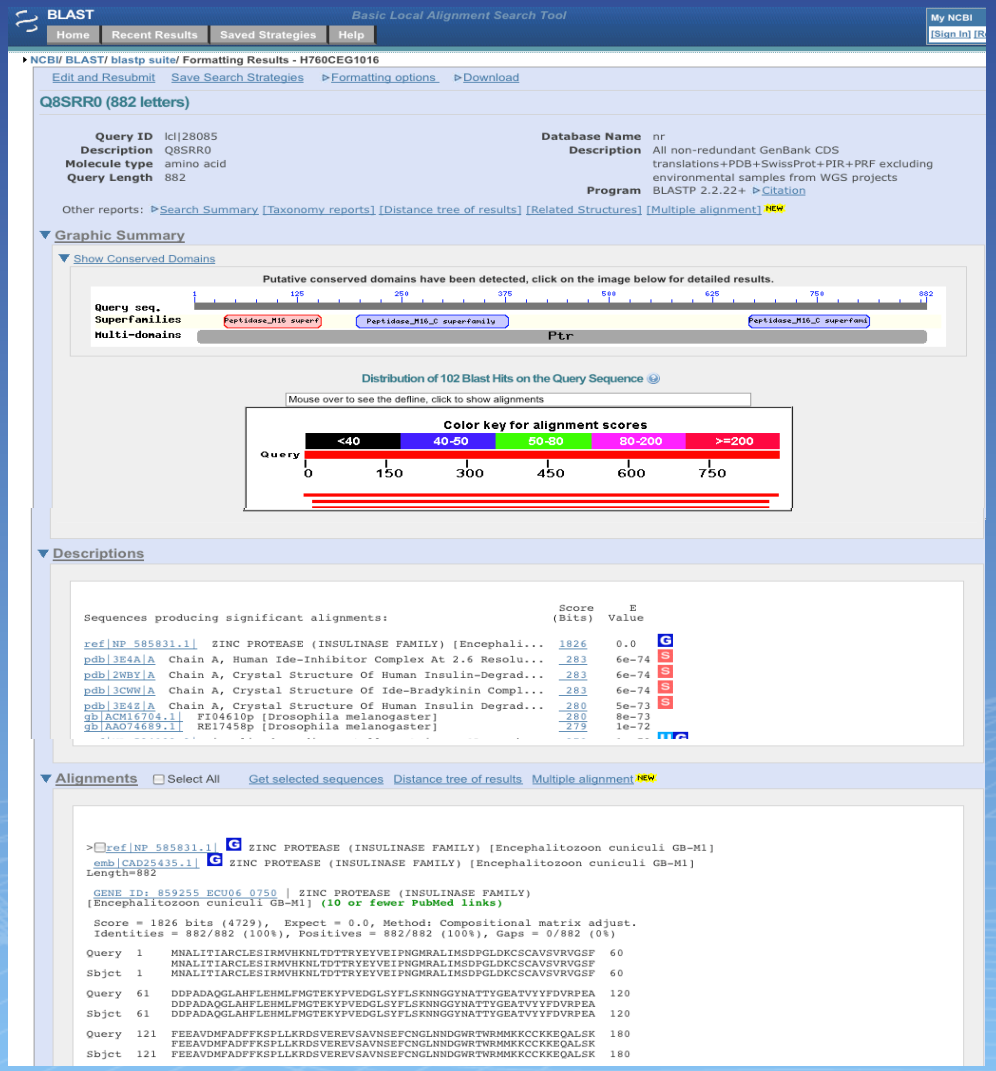
(3) For each word match, extend alignment in both directions to find alignments that score greater than score threshold S.



Maximal Segment Pairs (MSPs)

CIBIV MFPL

BLAST Searches*



*http://blast.ncbi.nlm.nih.gov/Blast.cgi

CIBIV MFPL

Different BLAST programs

The NCBI **BLAST** family of programs includes:

blastp

compares an amino acid query sequence against a protein sequence database

blastn

compares a nucleotide query sequence against a nucleotide sequence database

blastx

compares a nucleotide query sequence translated in all reading frames against a protein sequence database

tblastn

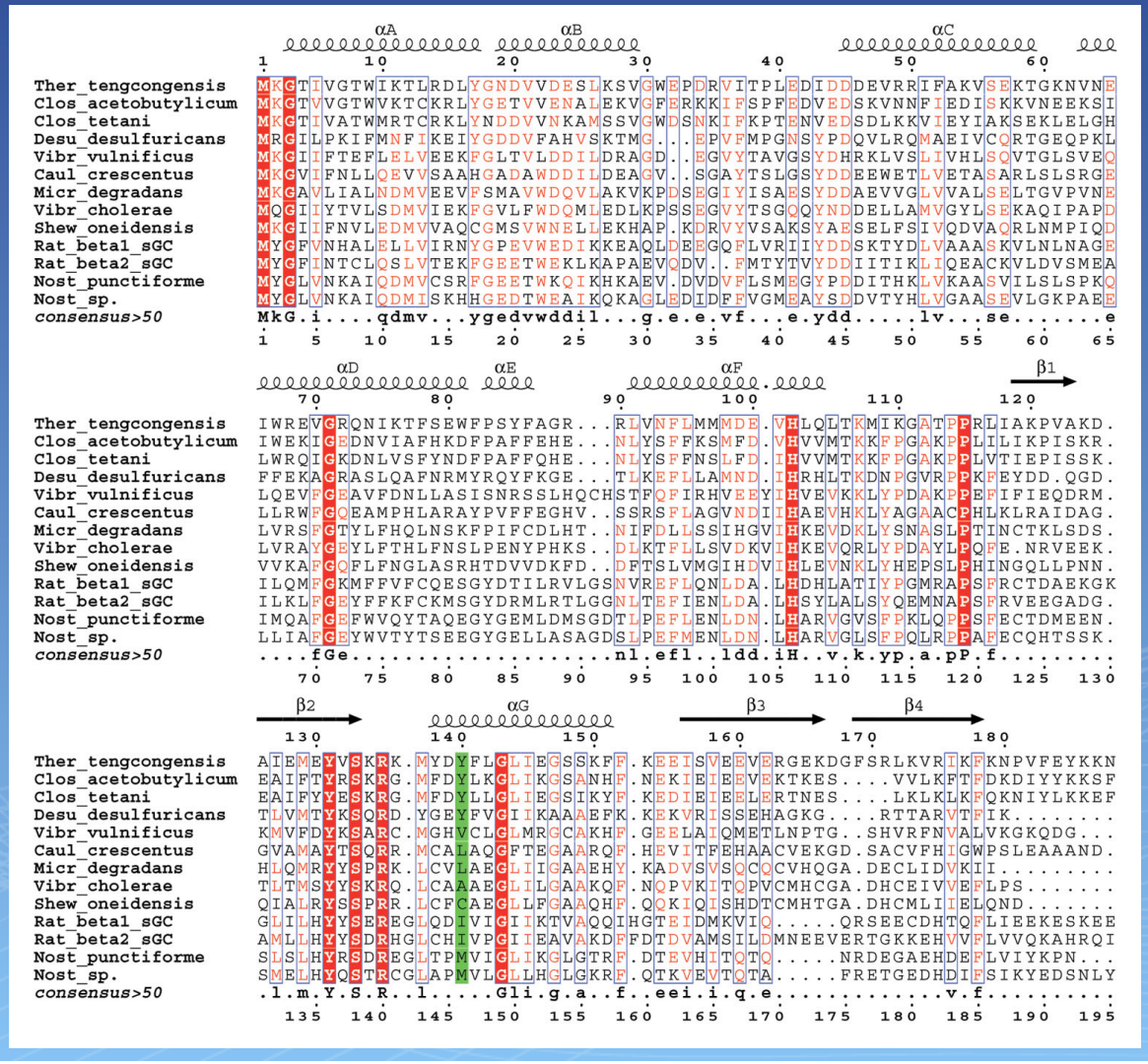
compares a protein query sequence against a nucleotide sequence database dynamically translated in all reading frames

tblastx

compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database. The tblastx program cannot be used with the nr database on the BLAST Web page.

CIBIV MFPL

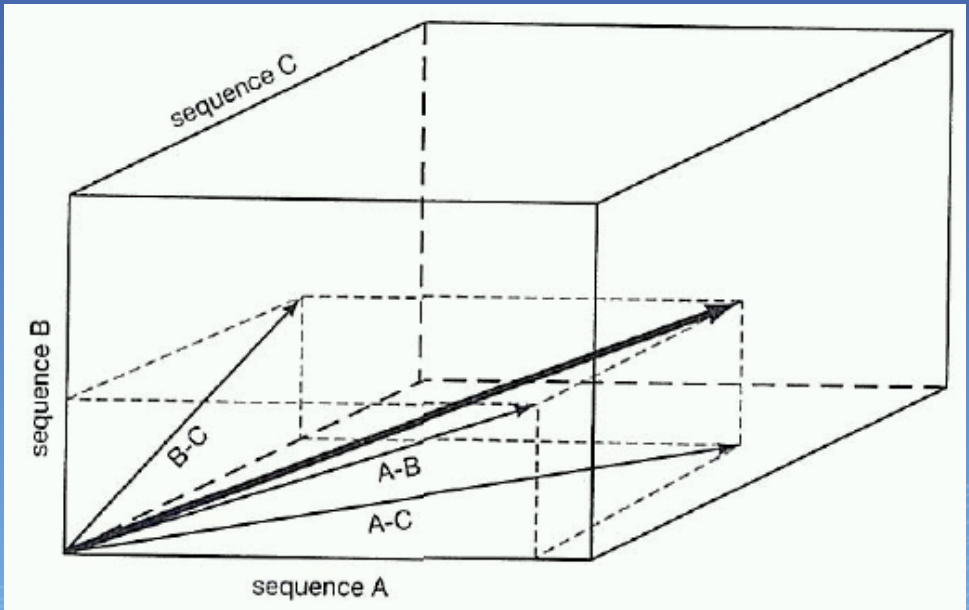
Multiple Sequence Alignments



CIBIV MFPL

Multiple Sequence Alignments

Optimal Solution:
Extend Needleman-Wunsch or Smith-Waterman to multiple sequences



But $O(n^m)$ in time and memory:
Computationally not feasible... 4 sequences of length 1000 -> 1TB RAM

CIBIV MFPL

Multiple Sequence Alignments
Sum Of Pairs

```
Seq1: AGA--CTA
Seq2: G-A--CTT
Seq3: AGAAACTT
```

```
Seq1: AGA--CTA      Seq1: AGA--CTA      Seq2: G-A--CTT
Seq2: G-A--CTT      Seq3: AGAAACTT      Seq3: AGAAACTT
```

$$S(a_i,b_j)=\begin{cases}+5, if\ a_i=b_j\\ -2, if\ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$

```
Seq1: AGA--CTA      Seq1: AGA--CTA      Seq2: G-A--CTT
Seq2: G-A--CTT      Seq3: AGAAACTT      Seq3: AGAAACTT
Score: +5           Score: +11          Score: 0
```

SUM OF PAIRS SCORE: 16

CIBIV MFPL

Multiple Sequence Alignments
Progressive alignment strategy

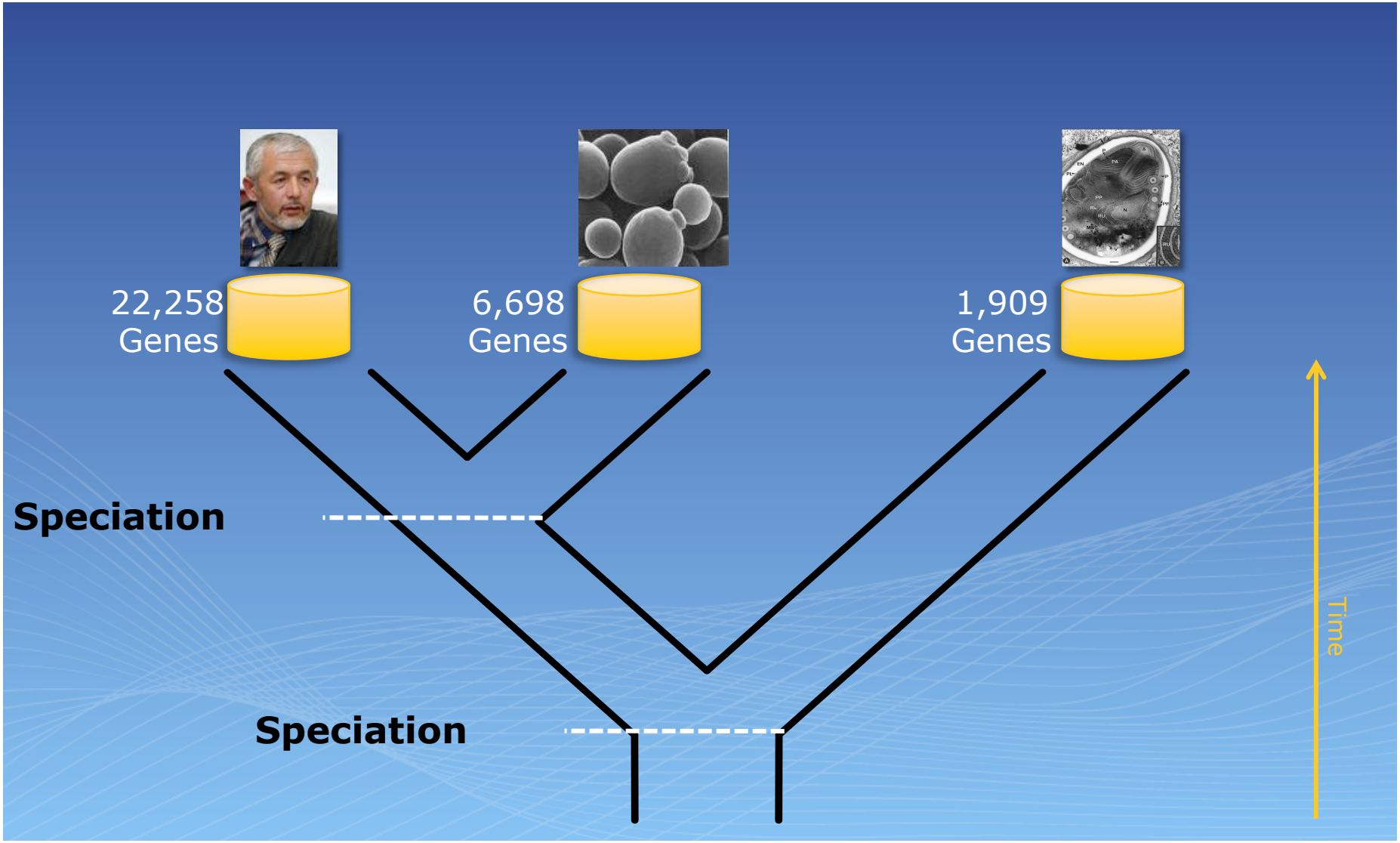
- The sequences are added stepwise. Thus, never more than two sequences (or multiple sequence alignments) are simultaneously aligned
- Sequences or MSAs are aligned using Dynamic Programming

$$\sigma(a^i, b^j) = \frac{1}{n \times m} \sum_{x=1}^{n} \sum_{y=1}^{m} S(a_x^i, b_y^j) \times \omega_x \times \omega_y$$

$\sigma(a^i,b^j)$:	score for aligning column i from alignment (or sequence) ***a*** to column j from alignment or sequence ***b***
n,m	number of sequences in alignments ***a*** and ***b***, respectively
$S(a_x^i, b_y^j)$	score for aligning position ***i*** in sequence ***x*** from alignment ***a*** to position ***j*** in sequence ***y*** from alignment ***b***
ω_x, ω_y	respective weights of the sequences ***x*** and ***y***

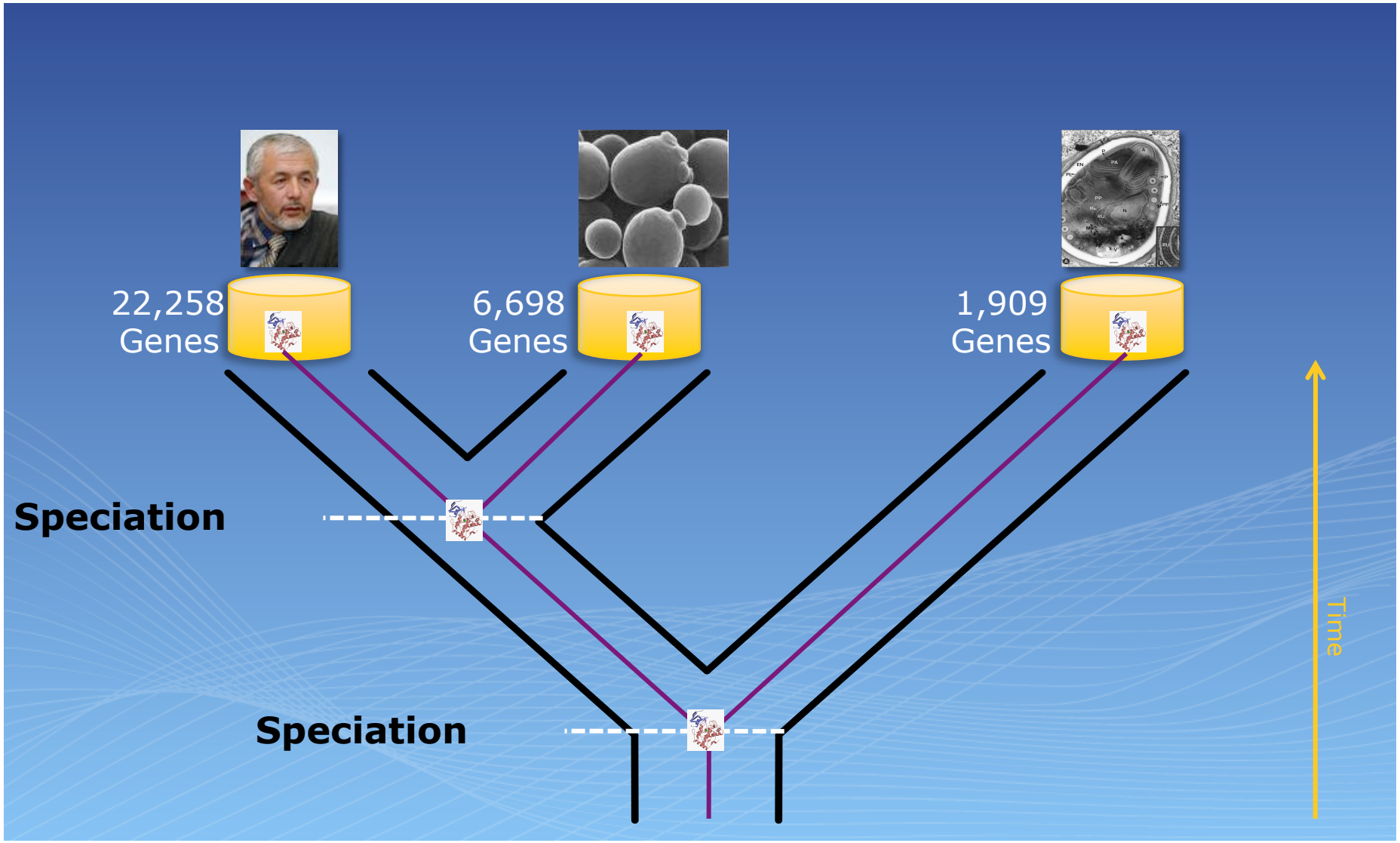
CIBIV MFPL

Evolutionary relationships of Species



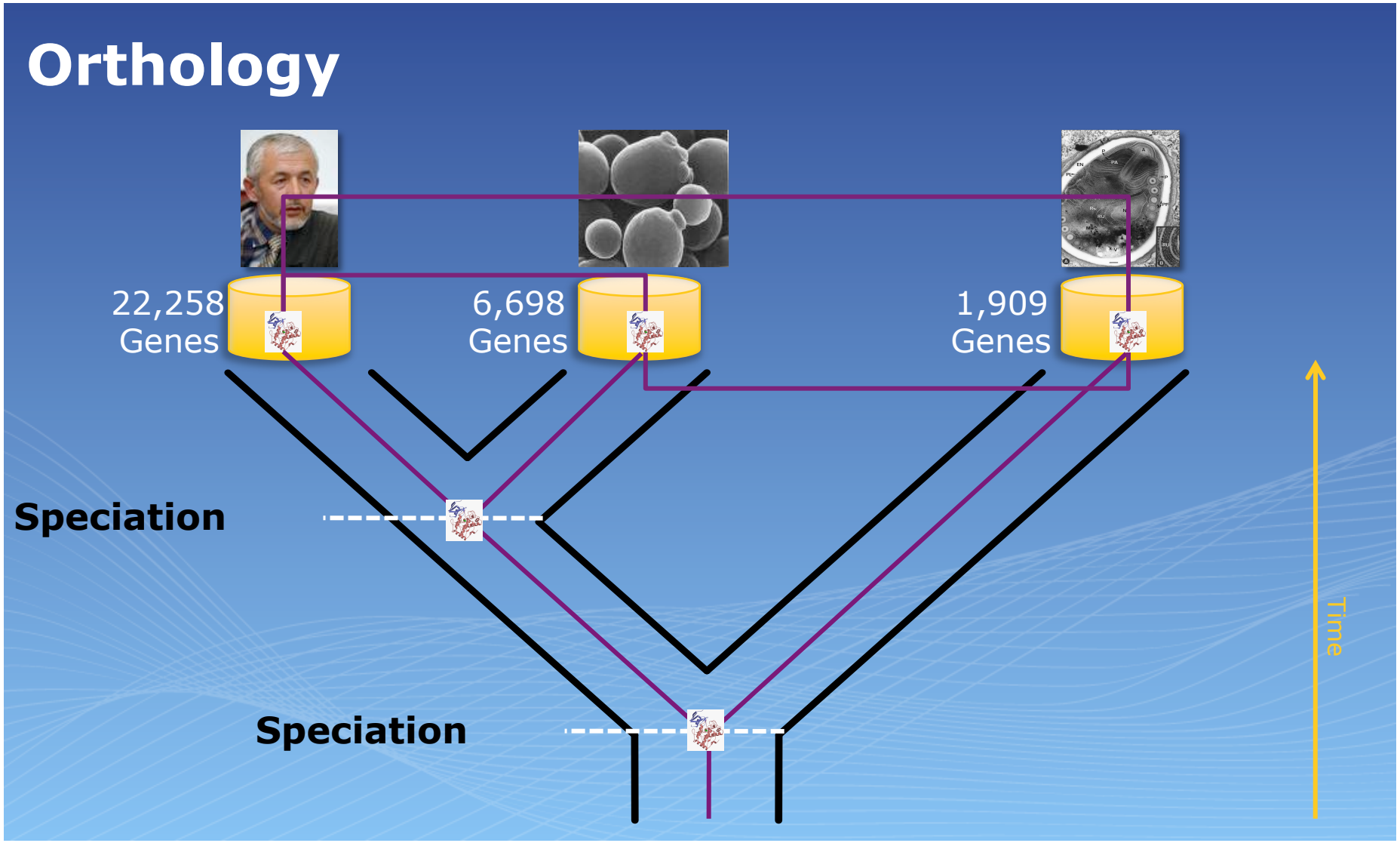
CIBIV MFPL

Evolutionary relationships of genes and gene products



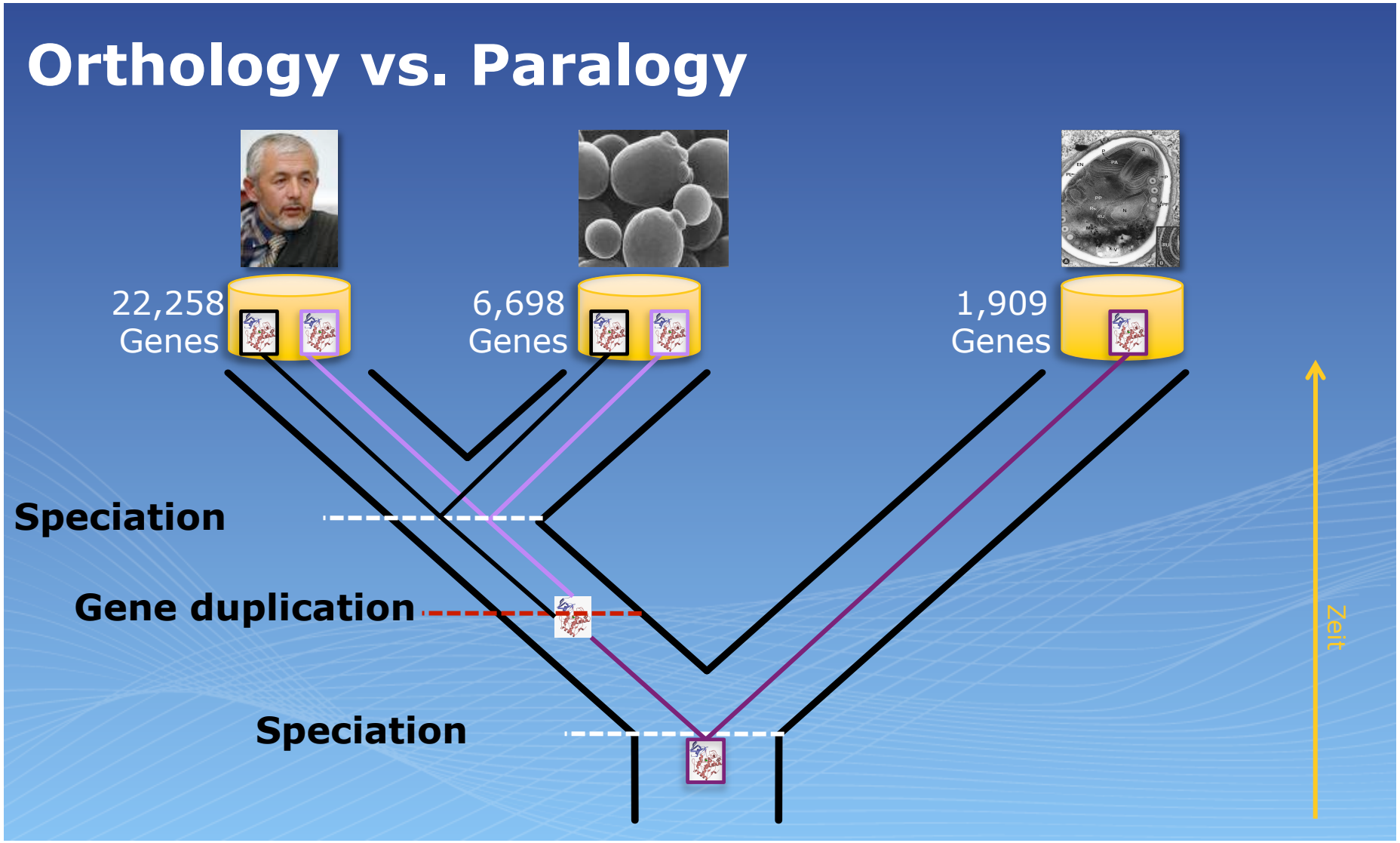
CIBIV MFPL

Evolutionary relationships of genes and gene products



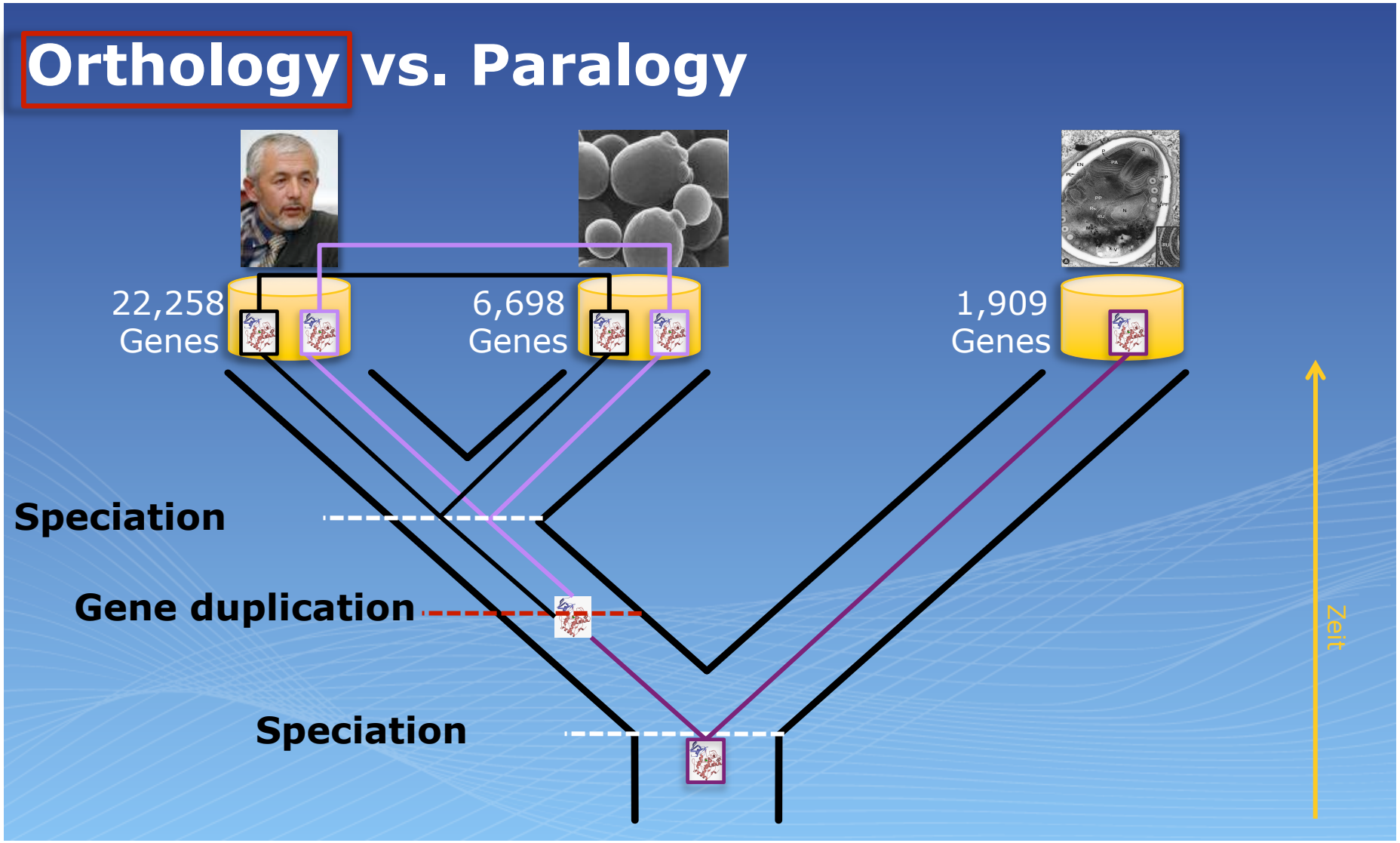
CIBIV MFPL

Evolutionary relationships of genes and gene products



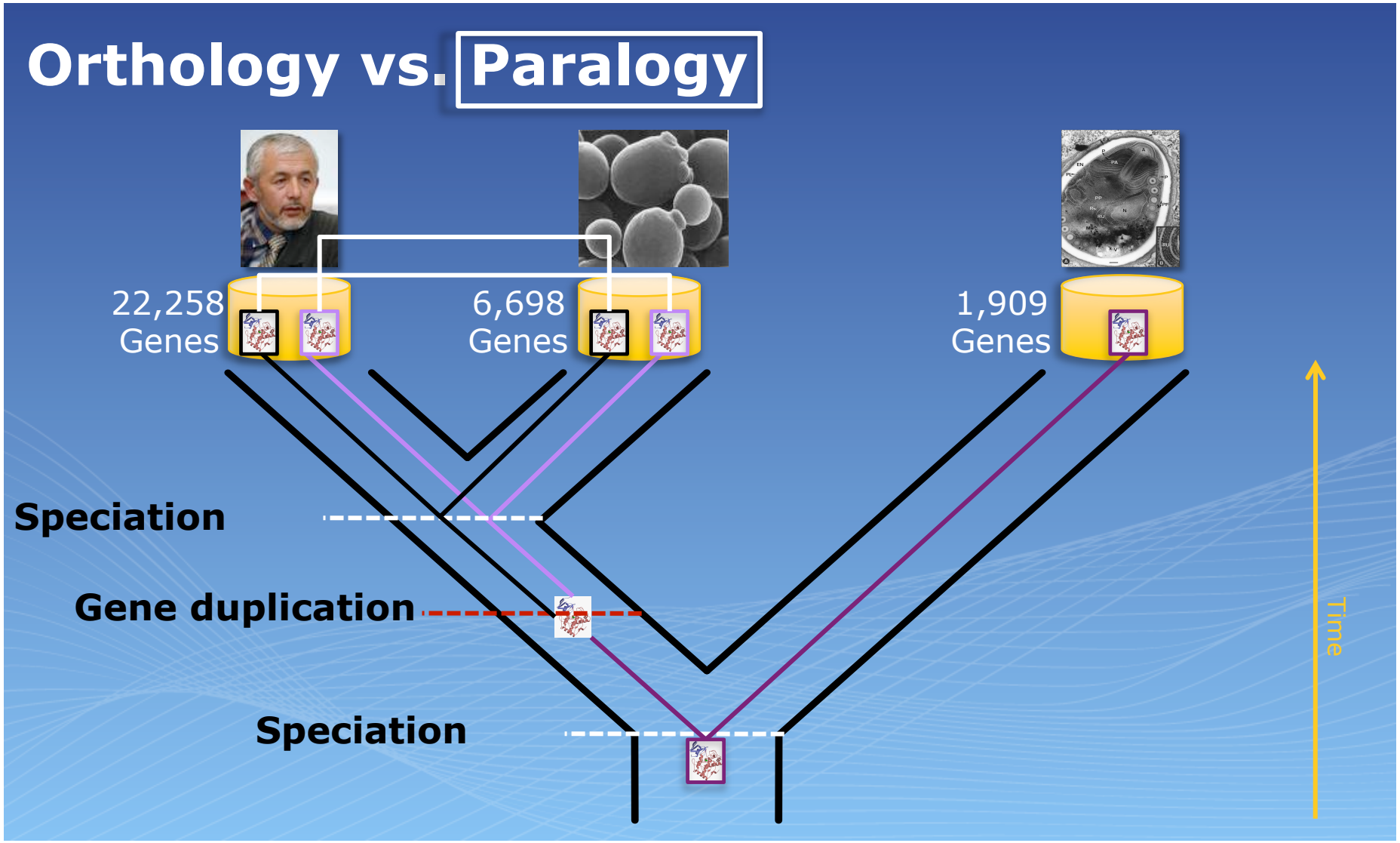
CIBIV MFPL

Evolutionary relationships of genes and gene products



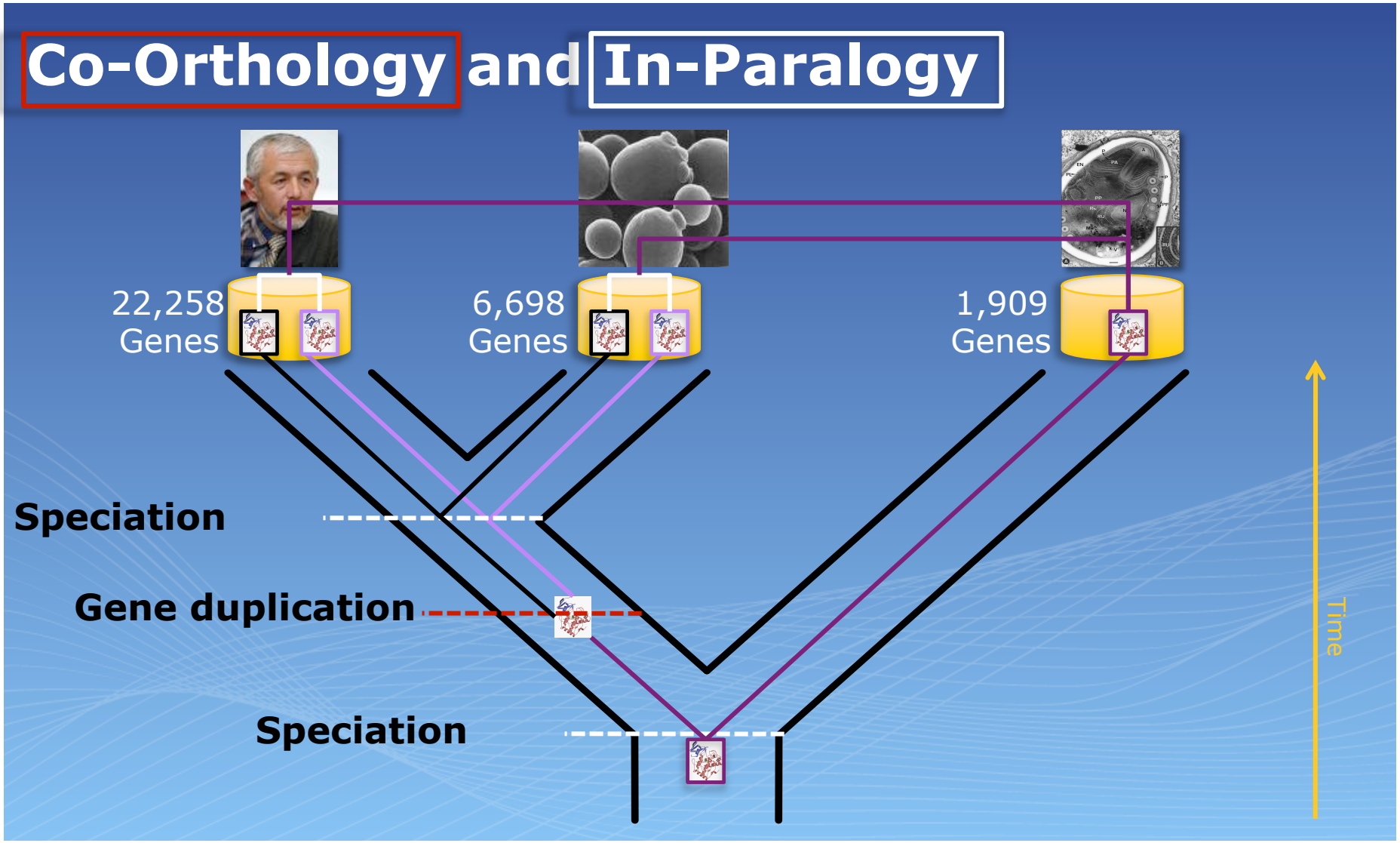
CIBIV MFPL

Evolutionary relationships of genes and gene products



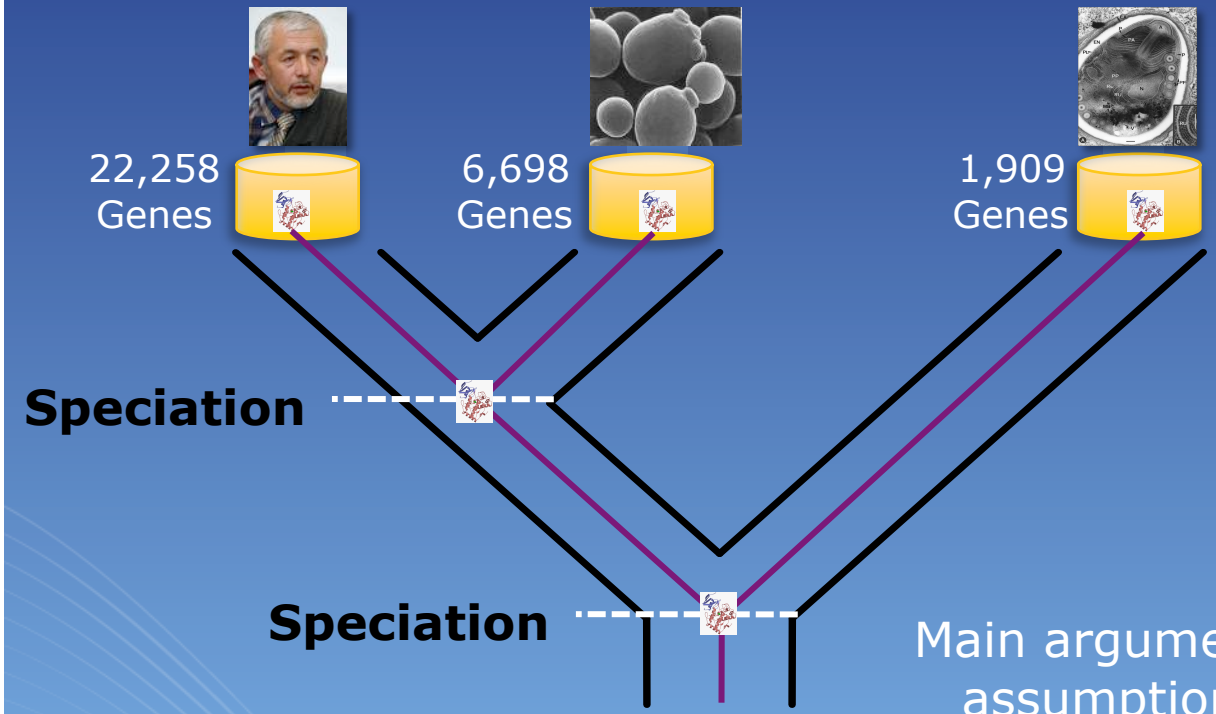
CIBIV MFPL

Evolutionary relationships of genes and gene products



CIBIV MFPL

Orthology prediction

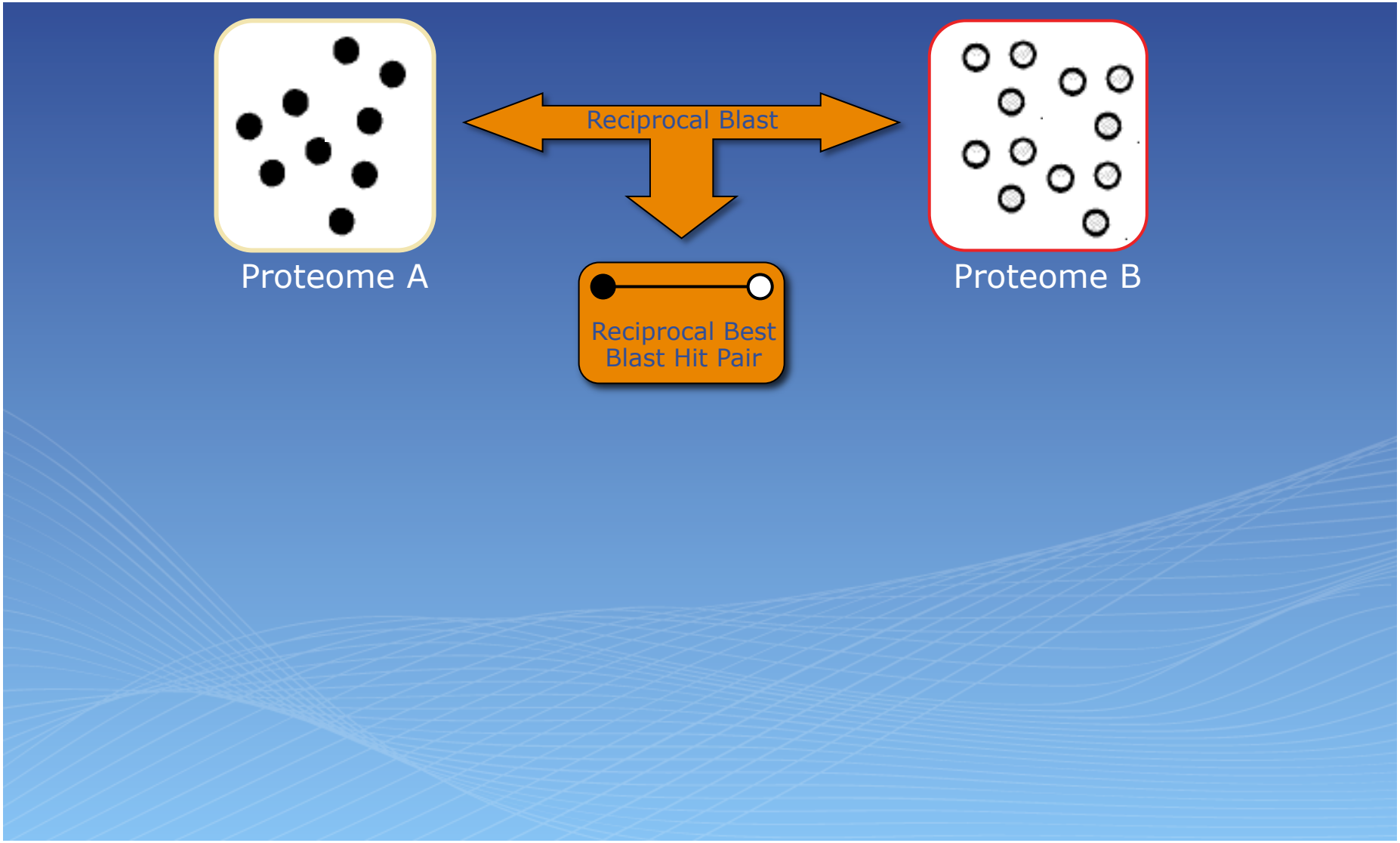


Main arguments for the orthology assumption:

- a sequence tree that is congruent to the species tree
- conservation of genomic position
- sequence similarity (typically, reciprocal best blast hit)

CIBIV MFPL

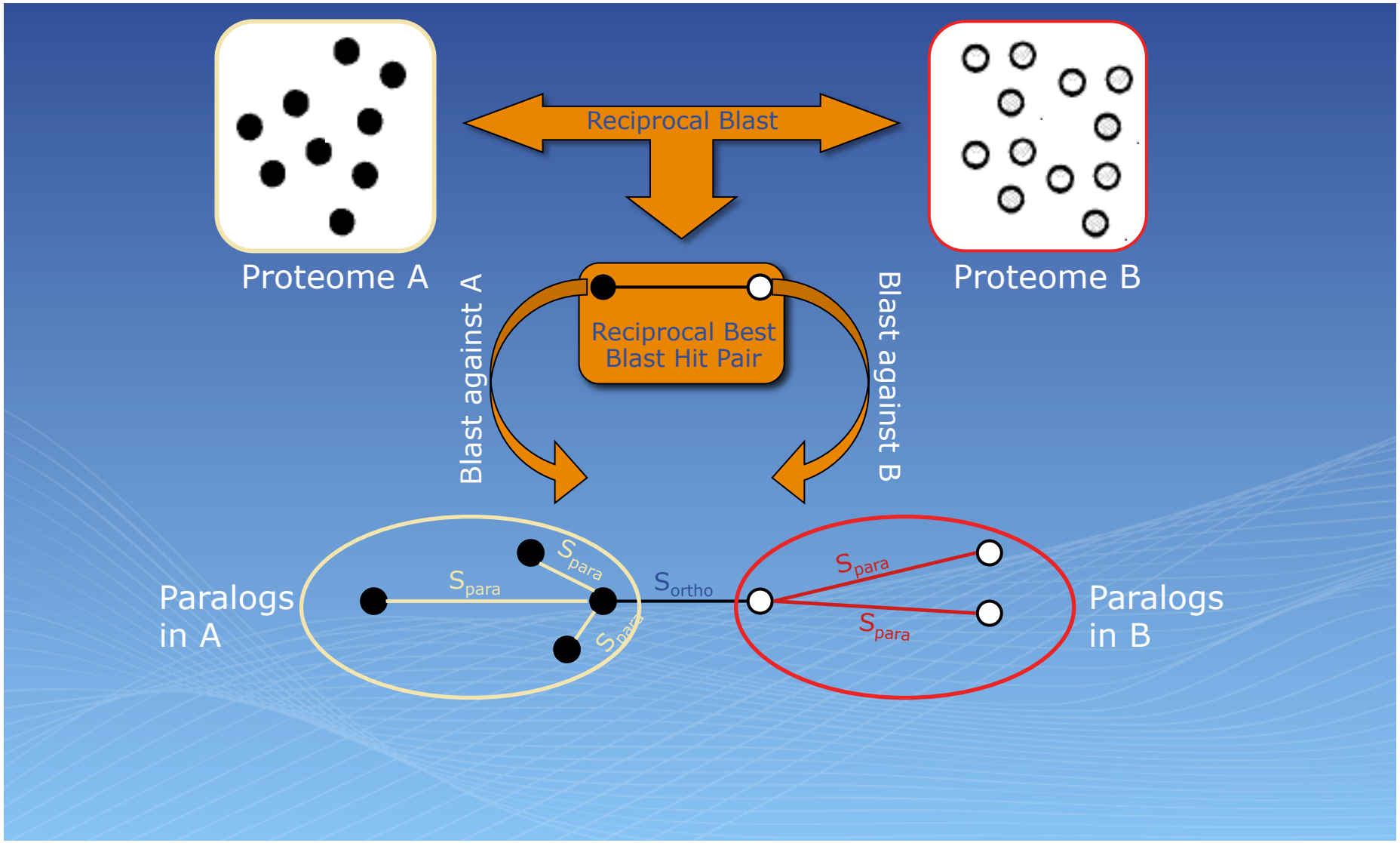
Orthology prediction: The RBH approach



Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

CIBIV MFPL

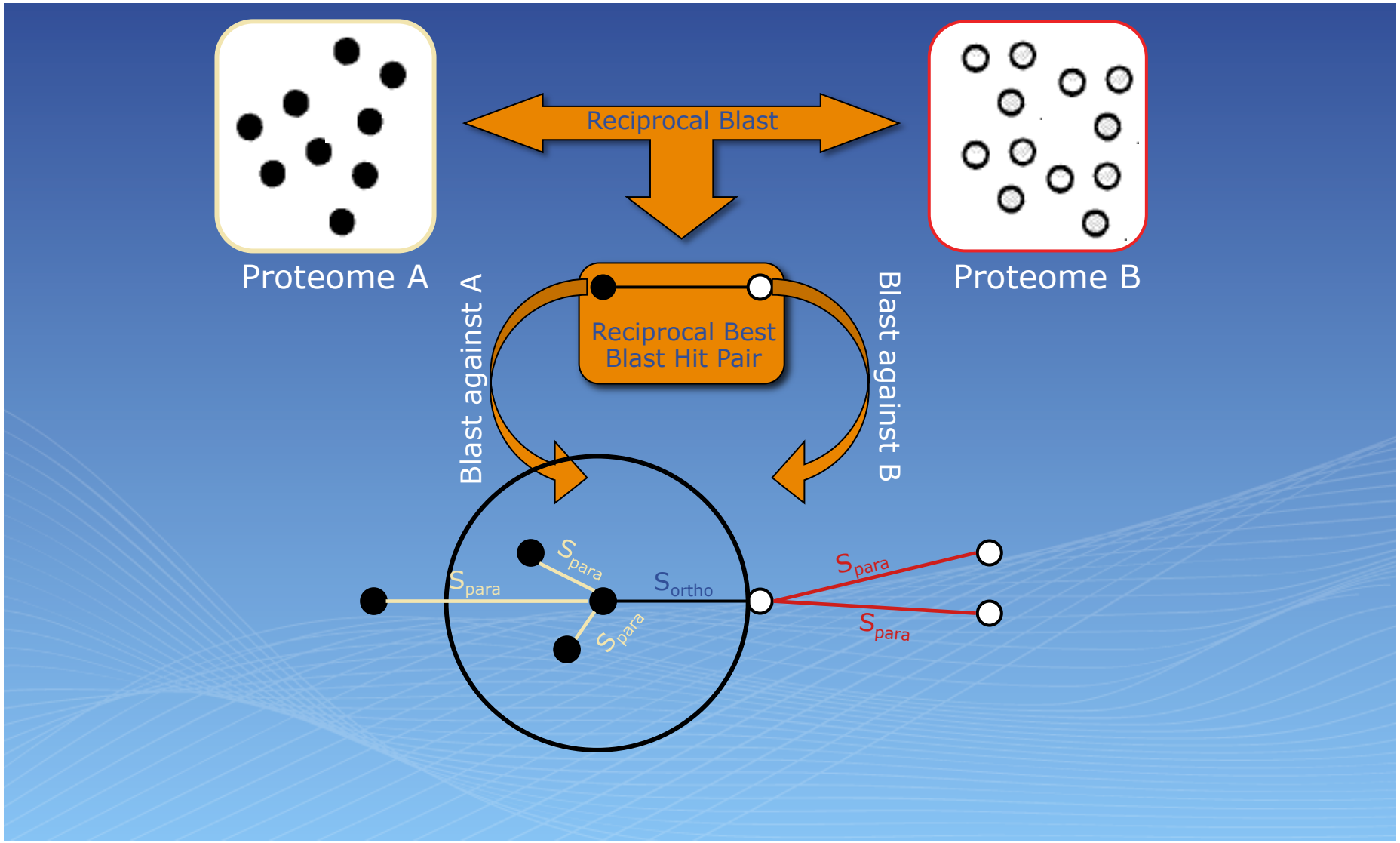
Orthology prediction: The InParanoid approach



Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

CIBIV MFPL

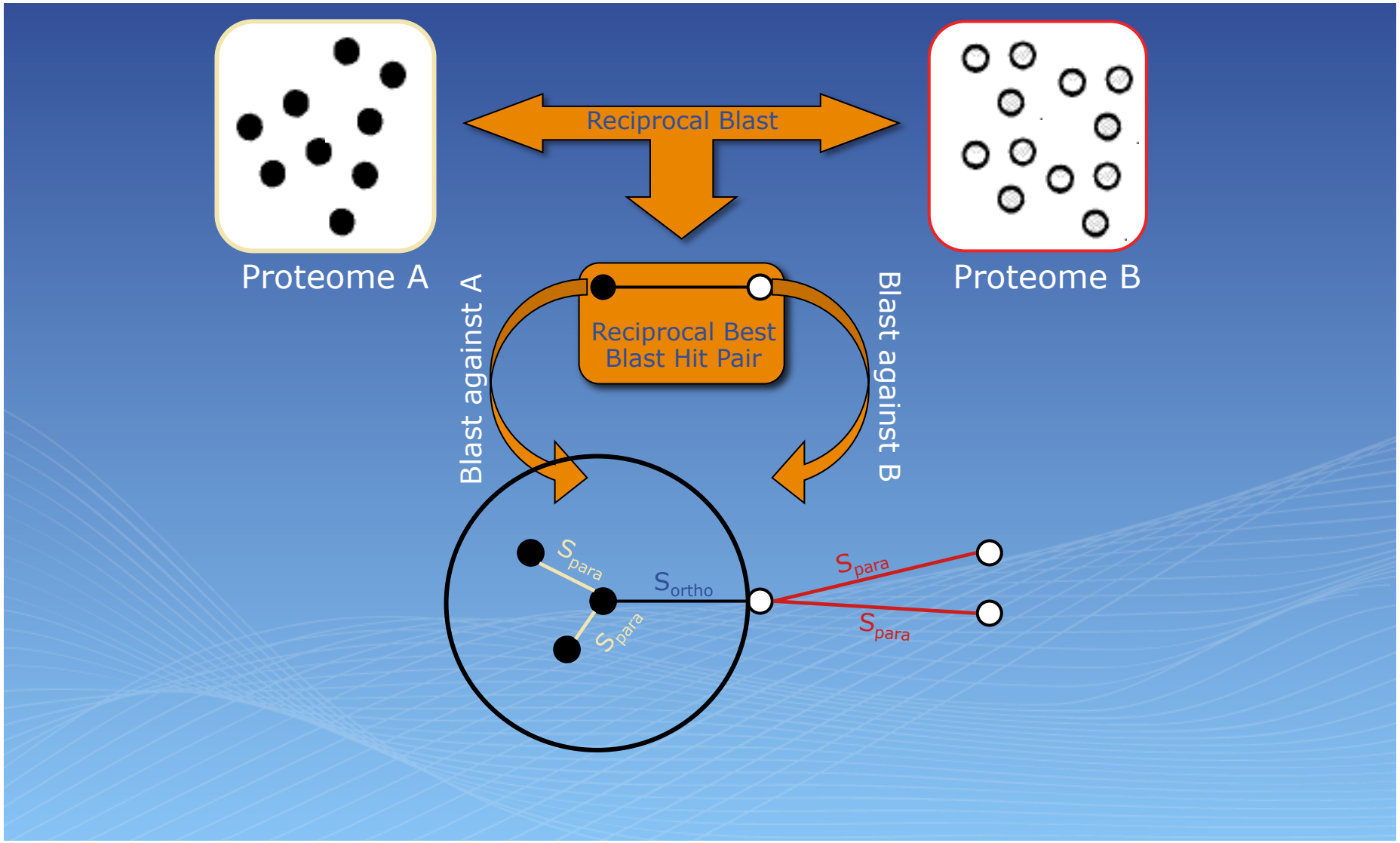
Orthology prediction: The InParanoid approach



Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

CIBIV MFPL

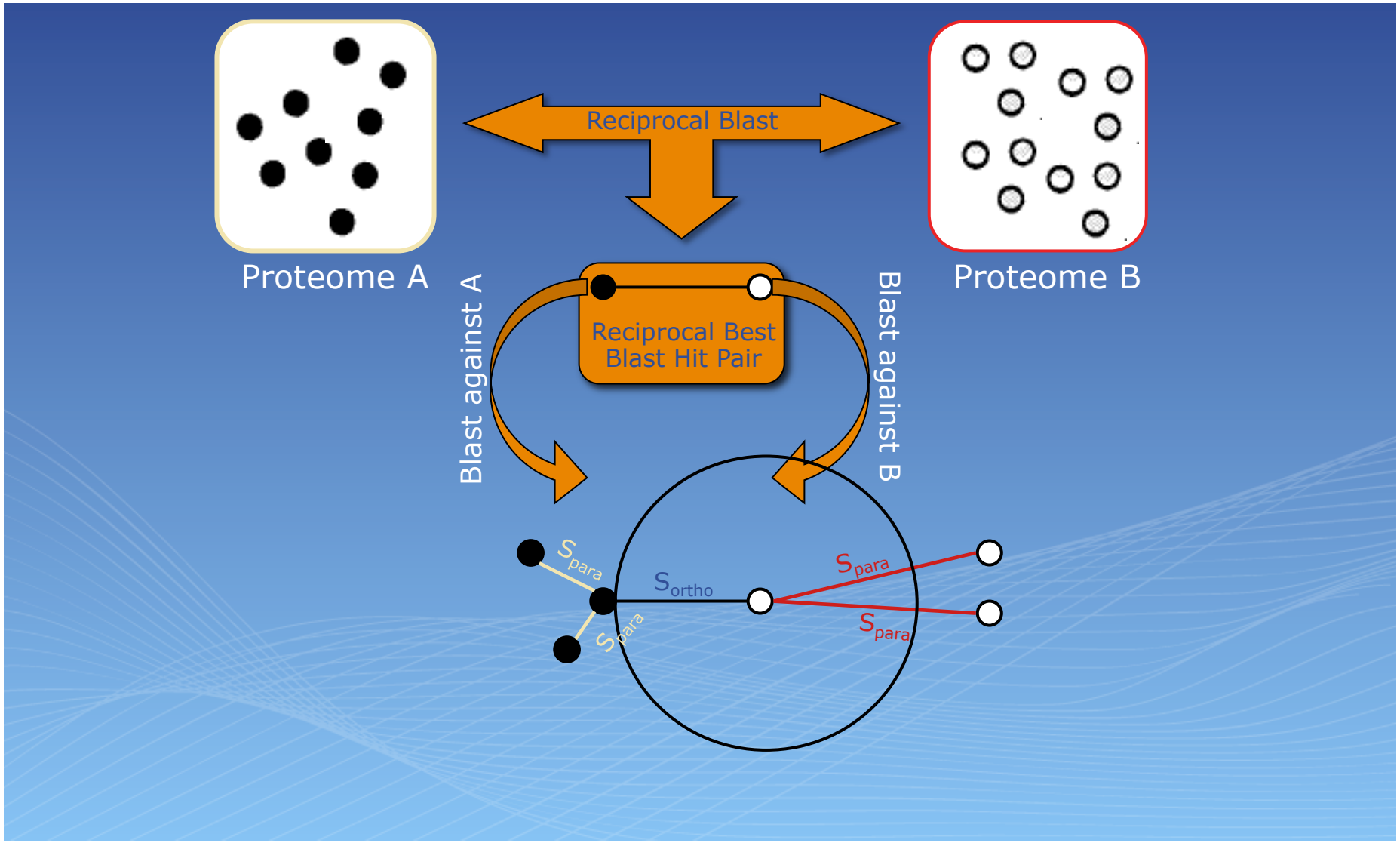
Orthology prediction: The InParanoid approach



Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

CIBIV MFPL

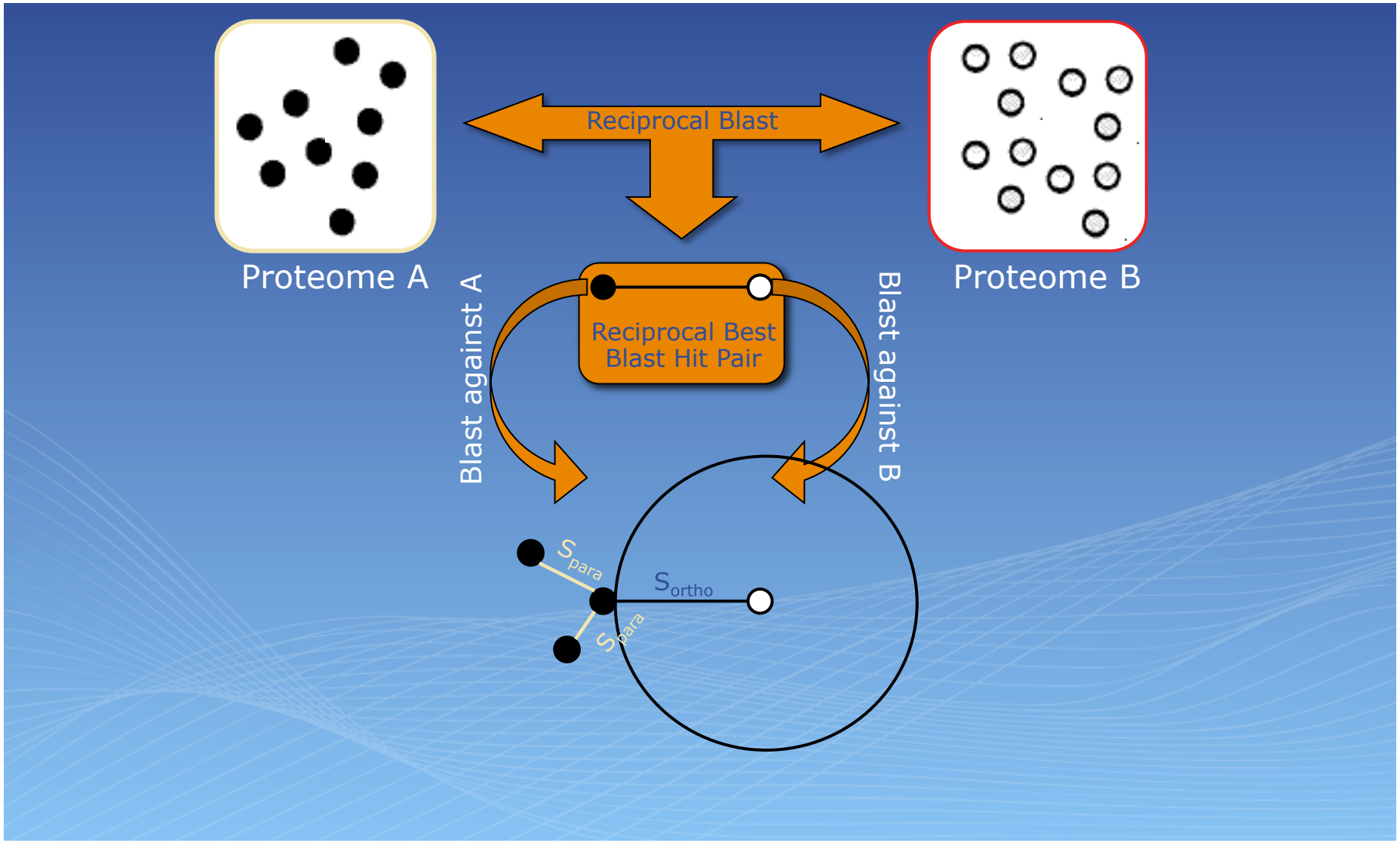
Orthology prediction: The InParanoid approach



Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

CIBIV MFPL

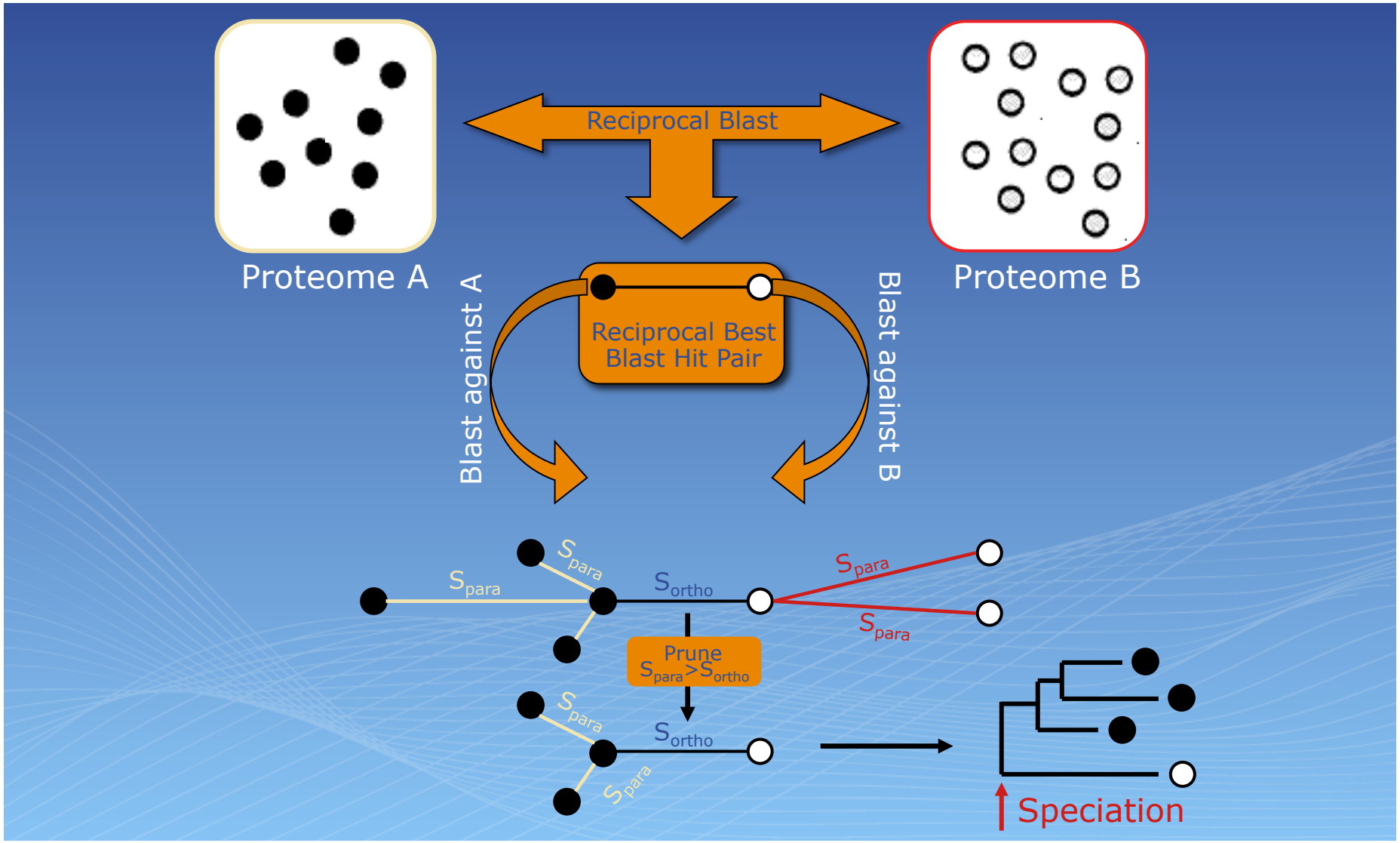
Orthology prediction: The InParanoid approach



Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

CIBIV MFPL

Orthology prediction: The InParanoid approach



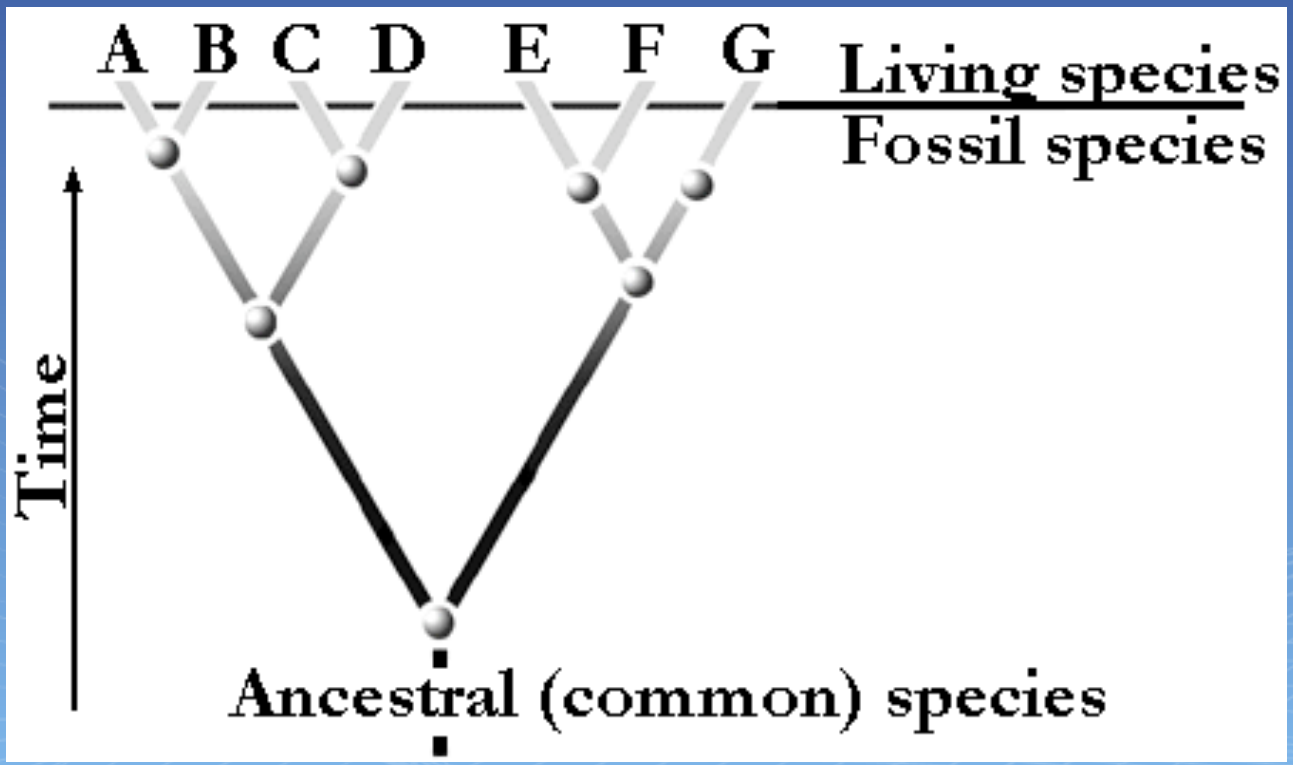
Modified from Remm et al. *J. Mol. Biol* (2001) 314: 1041-1052

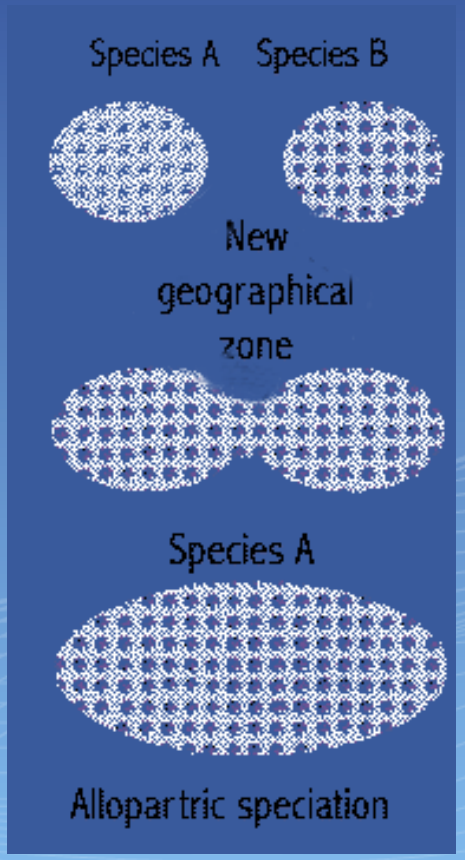
CIBIV MFPL

Tree reconstruction

CIBIV MFPL

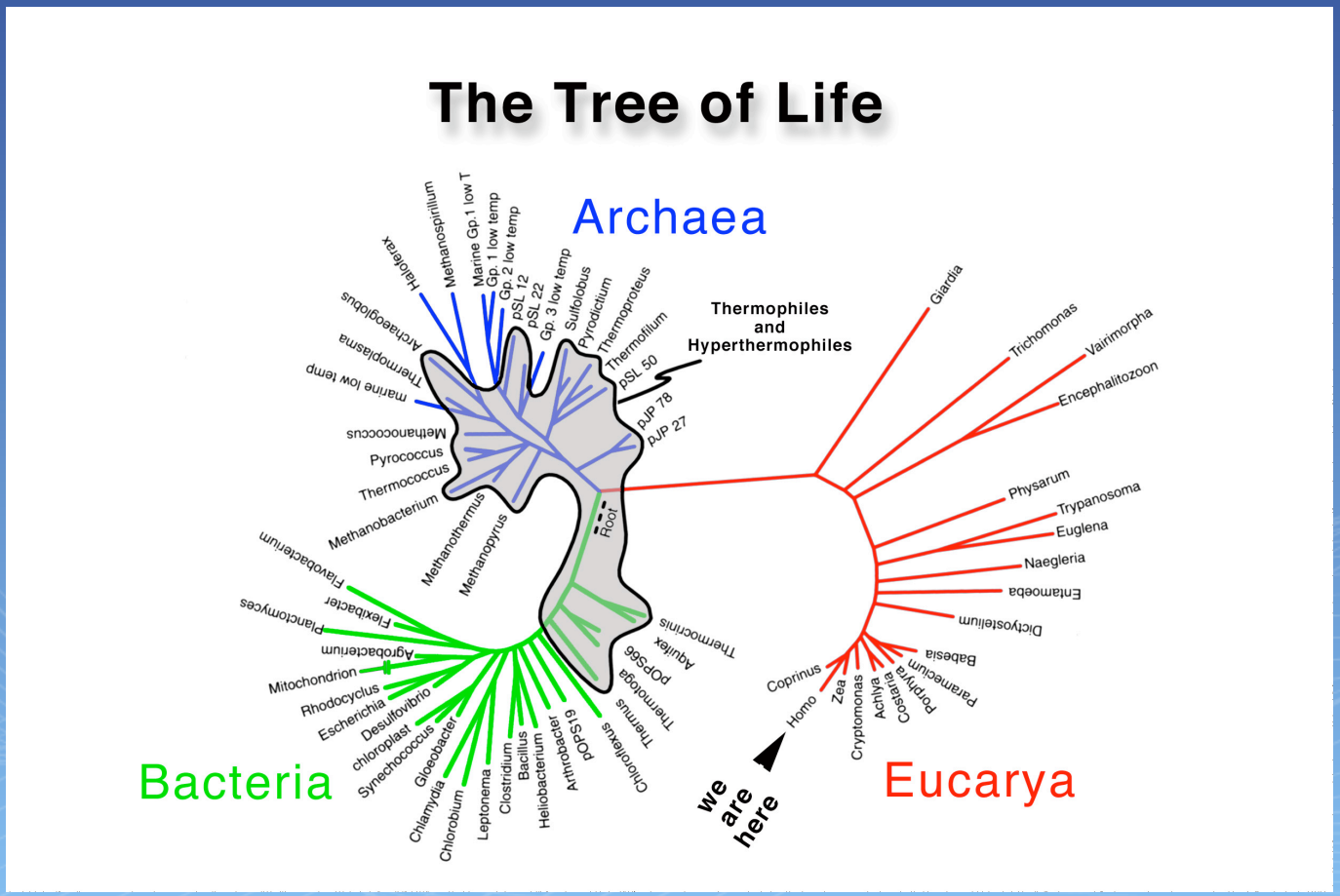
Working Hypothesis: Species evolution is tree like





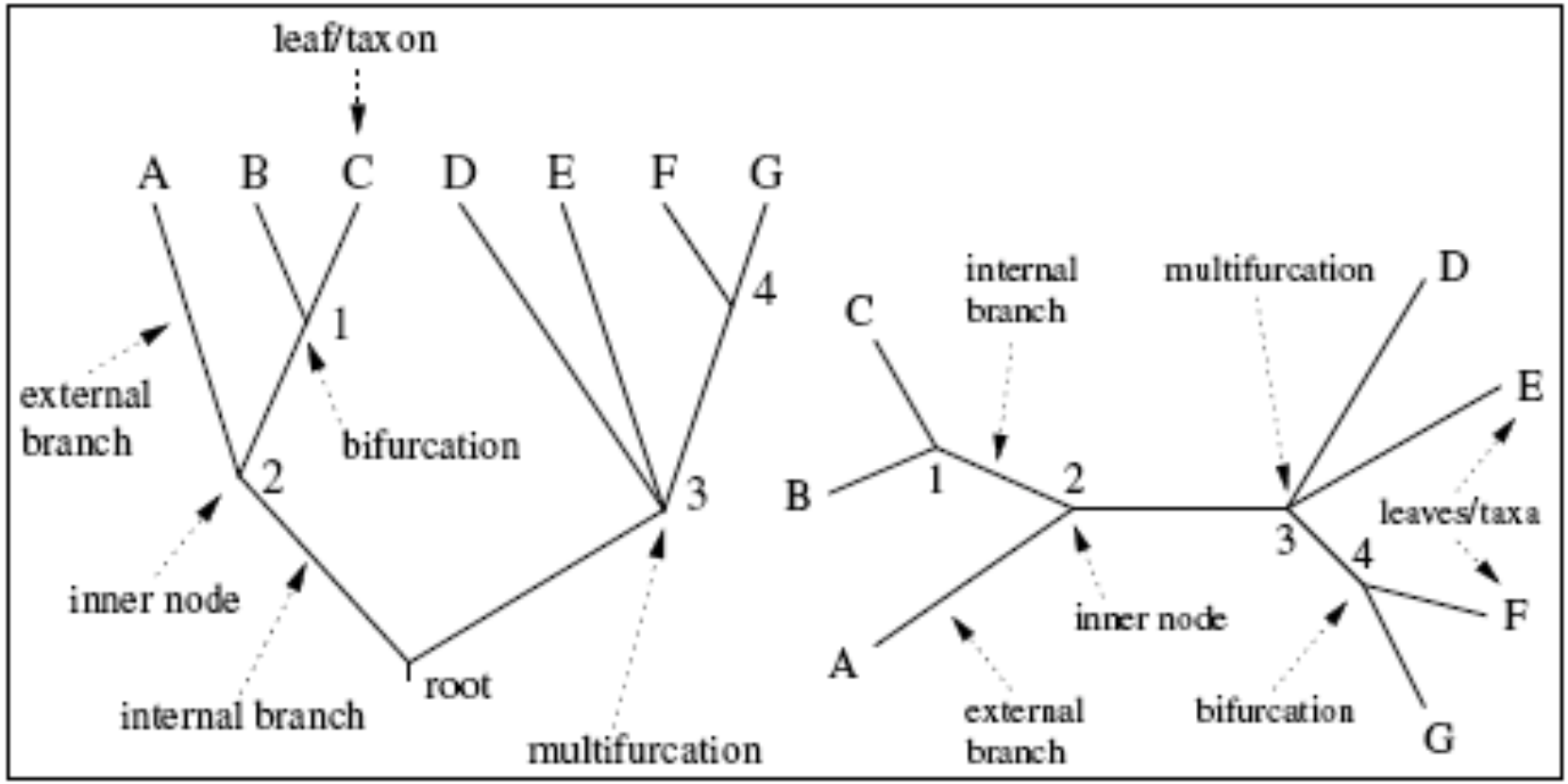
CIBIV MFPL

Tree reconstruction



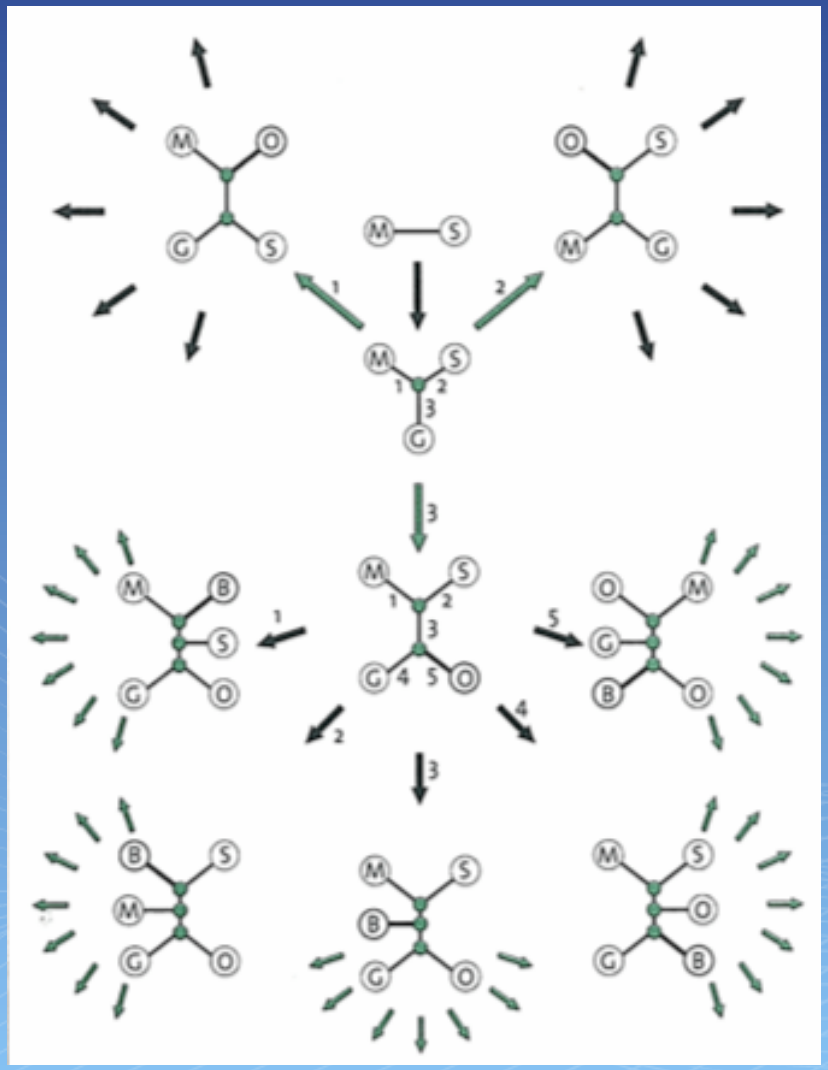
CIBIV MFPL

Tree notation



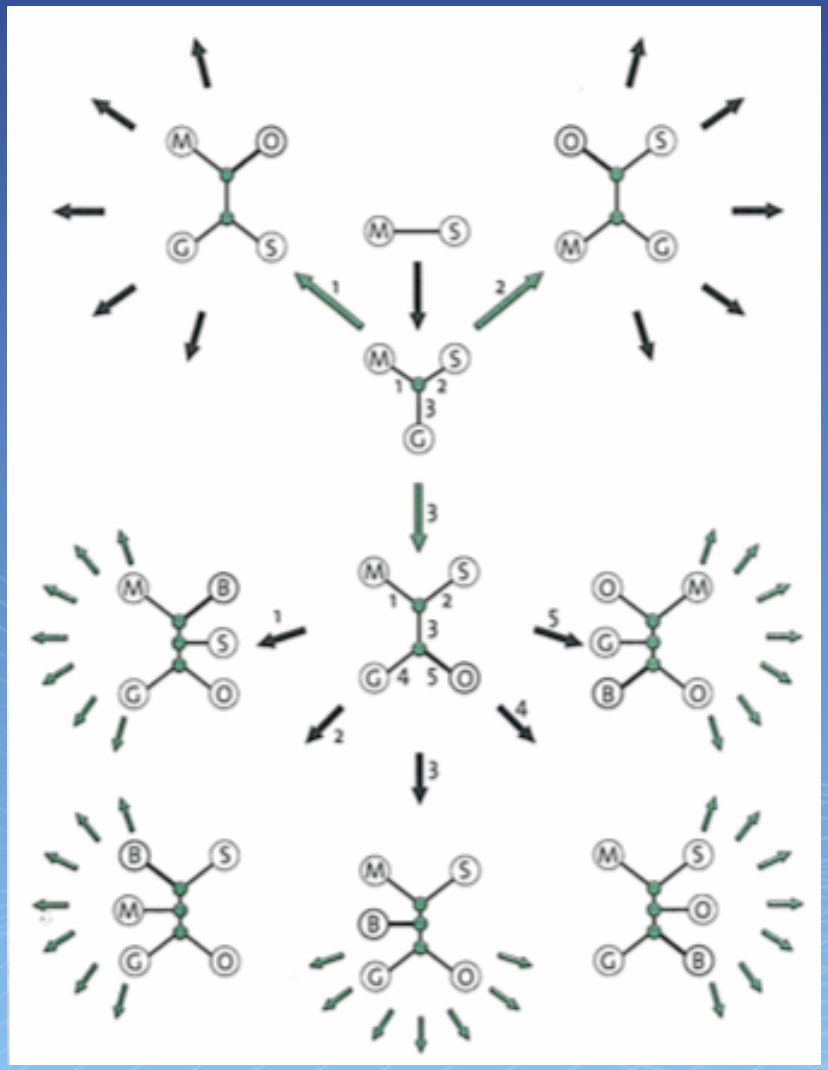
CIBIV MFPL

How many trees exist?



CIBIV MFPL

How many unrooted trees exist?



$$b(n) = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$

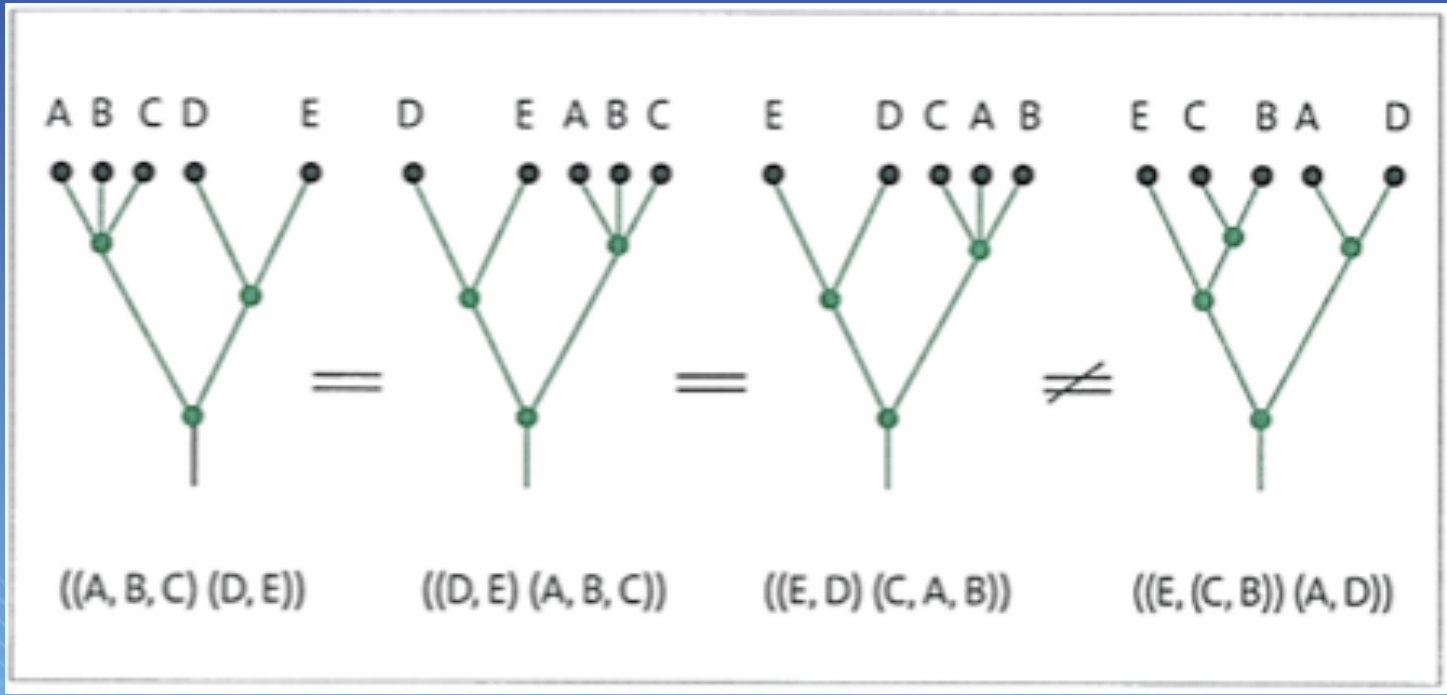
$$b(10) = 2027025$$

$$b(55) = 2.9 \times 10^{84}$$

$$b(100) = 1.7 \times 10^{182}$$

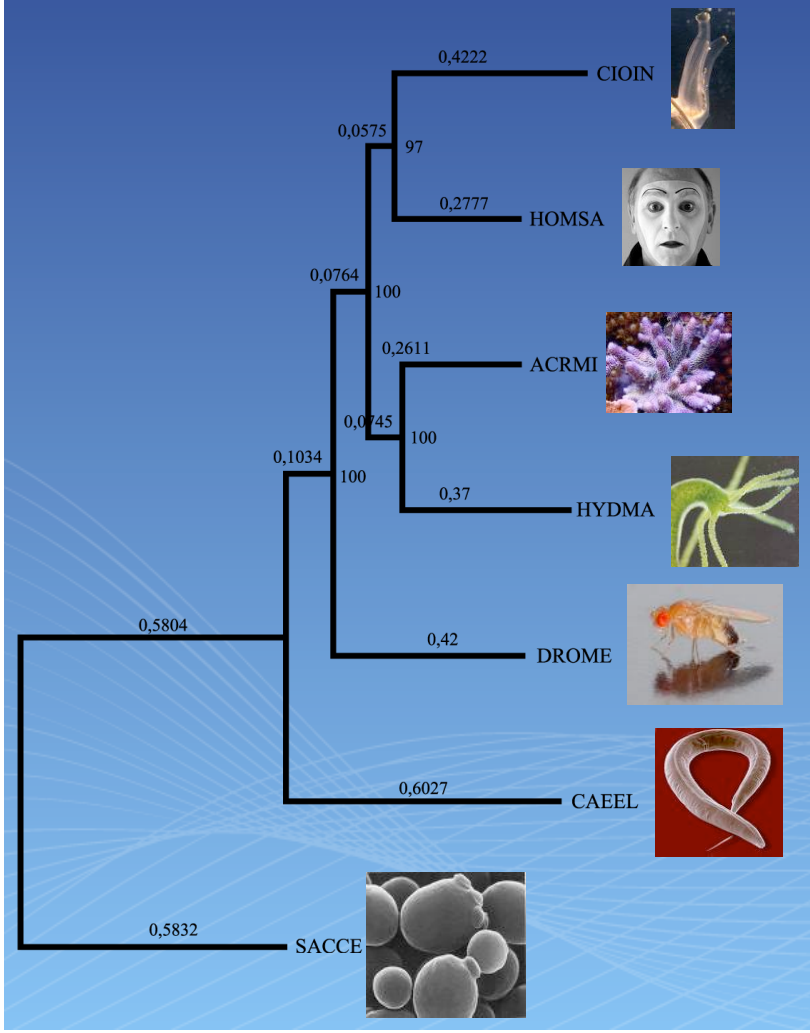
CIBIV MFPL

Tree formats



CIBIV MFPL

Three representations of the same tree

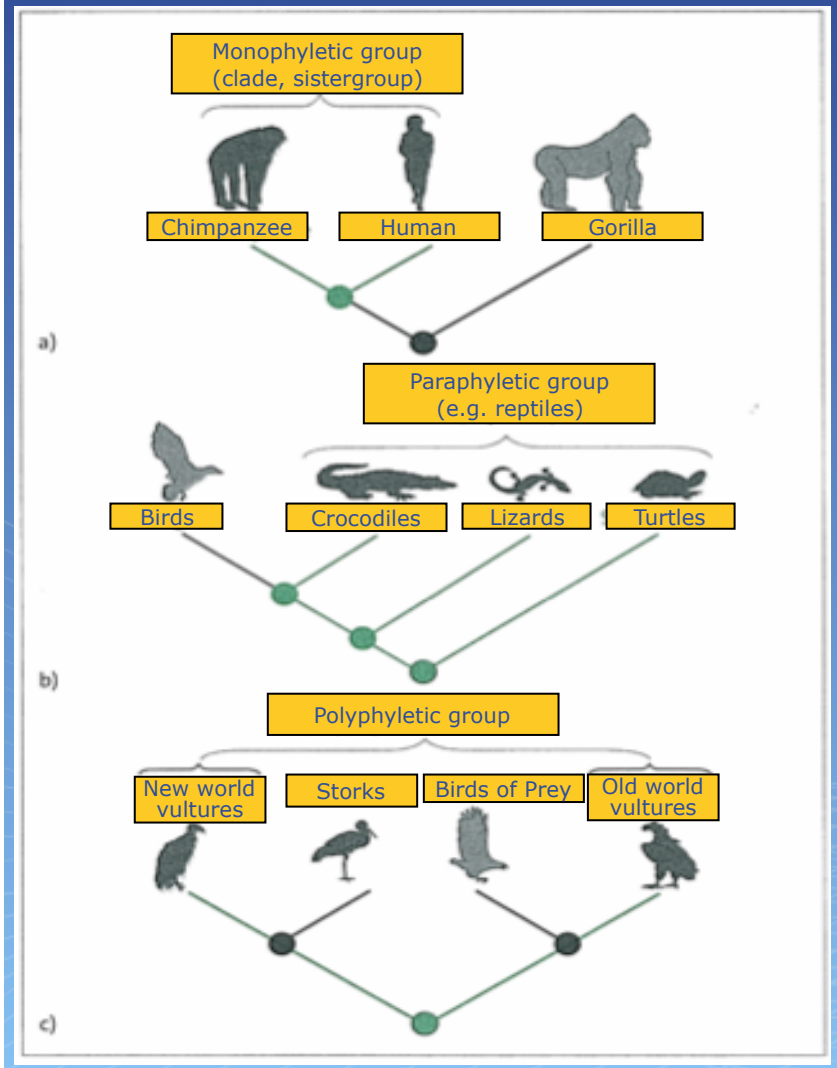



```
#NEXUS
begin taxa;
	dimensions ntax=7;
	taxlabels
	DROME
	CIOIN
	HYDMA
	SACCE
	CAEEL
	ACRMI
	HOMSA
;
end;

begin trees;
	tree [&r] tree_1 = (((((CIOIN:0.4222,HOMSA:0.2777)
[&label=97]:0.0575,
(ACRMI:0.2611,HYDMA:0.37)[&label=100]:0.0745)
[&label=100]:0.0764,
DROME:0.42)[&label=100]:0.1034,CAEEL:0.6027):0.5804,
SACCE:0.5832);
end;

begin figtree;
	set appearance.backgroundColour=#-1;
end figtree;
```

```
(((((CIOIN:0.4222,HOMSA:0.2777)97:0.0575,(ACRMI:
0.2611,HYDMA:0.3700)100:0.0745)100:0.0764,DROME:
0.4200)100:0.1034,
CAEEL:0.6027):0.5804,SACCE:0.5832);
```

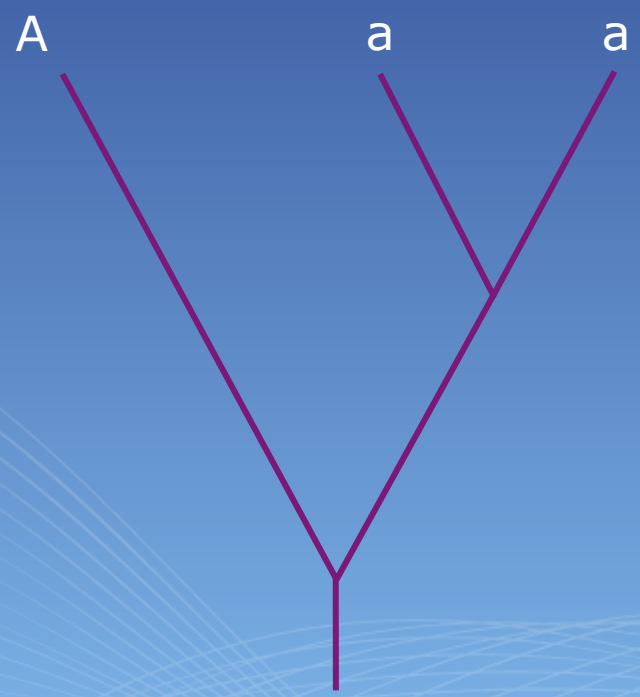
CIBIV MFPL

Some more notations



CIBIV MFPL

Some more notations

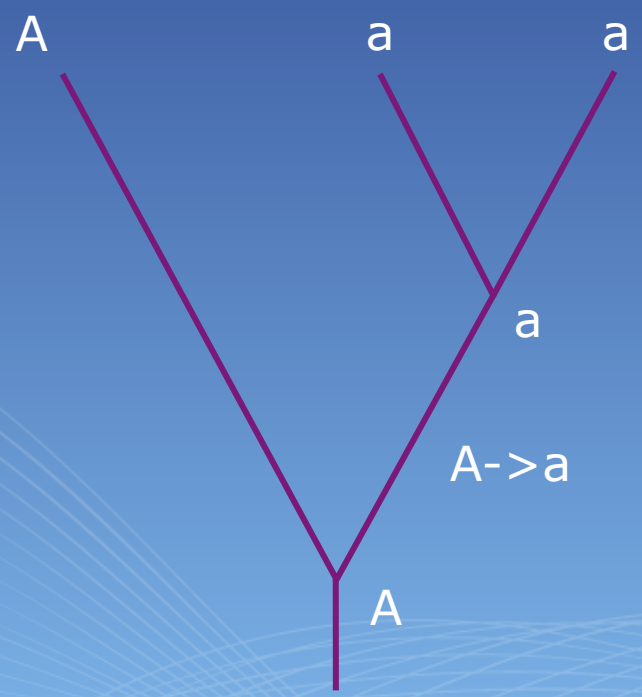


Character based phylogeny reconstruction:

- A character has to be expressed in at least two states in the taxa under study. Taxa are grouped on the basis of shared character states.

CIBIV MFPL

Some more notations

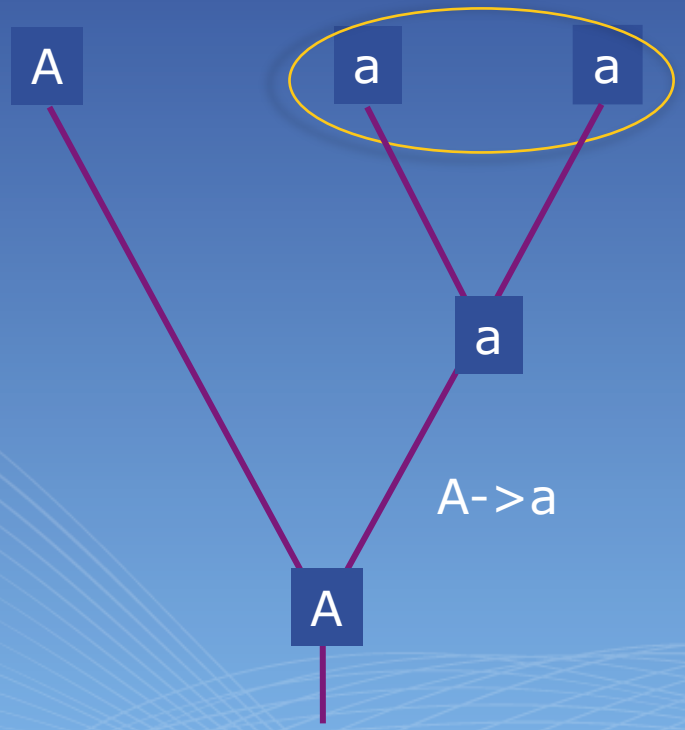


Character based phylogeny reconstruction:

- A character has to be expressed in at least two states in the taxa under study. Taxa are grouped on the basis of shared character states.
- An evolutionary derived character (state) is called an *Apomorphy*

CIBIV MFPL

Some more notations

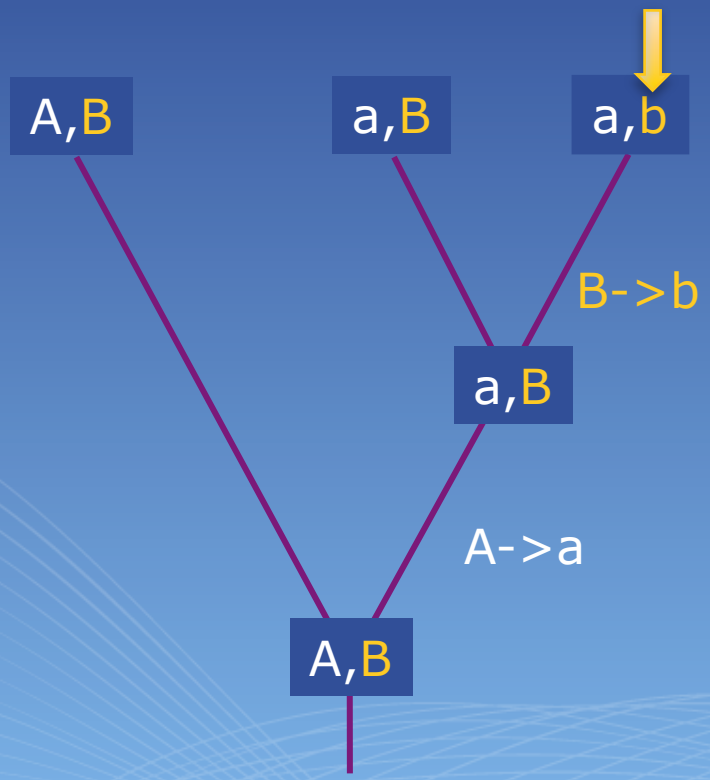


Character based phylogeny reconstruction:

- A character has to be expressed in at least two states in the taxa under study. Taxa are grouped on the basis of shared character states.
- An evolutionary derived character (state) is called an *Apomorphy*
- *Syn-Apomorphy:* an evolutionary derived character (state) shared by a group of taxa.

CIBIV MFPL

Some more notations

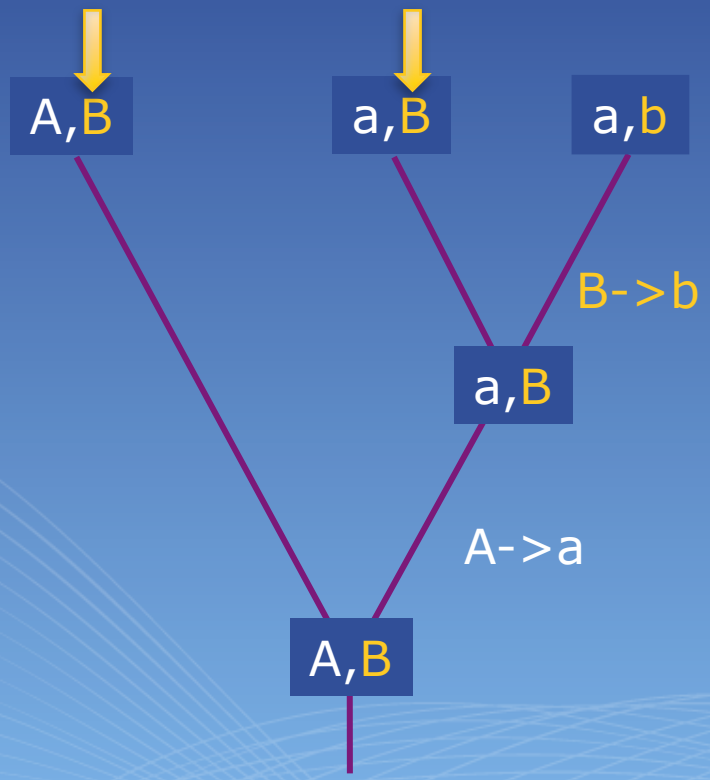


Character based phylogeny reconstruction:

- A character has to be expressed in at least two states in the taxa under study. Taxa are grouped on the basis of shared character states.
- An evolutionary derived character (state) is called an *Apomorphy*
- *Syn-Apomorphy:* an evolutionary derived character (state) shared by a group of taxa.
- *Aut-Apomorphy:* an evolutionary derived character (state) present only in a single taxon

CIBIV MFPL

Some more notations

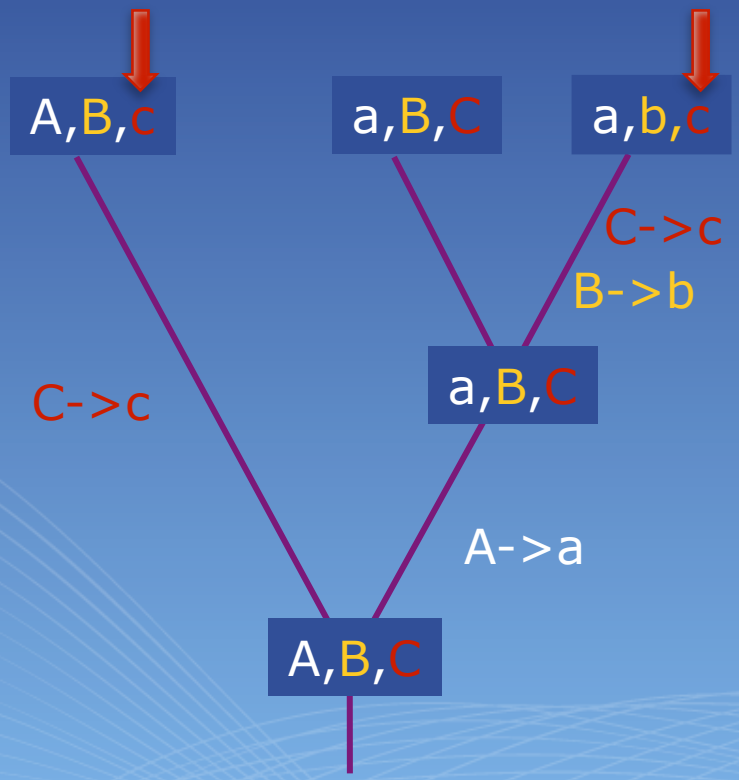


Character based phylogeny reconstruction:

- A character has to be expressed in at least two states in the taxa under study. Taxa are grouped on the basis of shared character states.
- An evolutionary derived character (state) is called an *Apomorphy*
- *Syn-Apomorphy:* an evolutionary derived character (state) shared by a group of taxa.
- *Aut-Apomorphy:* an evolutionary derived character (state) present only in a single taxon
- *Plesiomorphy:* an ancestral character (state) shared by a group of extant taxa.

CIBIV MFPL

Some more notations



Character based phylogeny reconstruction:

- A character has to be expressed in at least two states in the taxa under study. Taxa are grouped on the basis of shared character states.
- An evolutionary derived character (state) is called an *Apomorphy*
- *Syn-Apomorphy:* an evolutionary derived character (state) shared by a group of taxa.
- *Aut-Apomorphy:* an evolutionary derived character (state) present only in a single taxon
- *Plesiomorphy:* an ancestral character (state) shared by a group of extant taxa.
- *Homoplasy:* A derived character (state) that is shared for reasons other than common decent.

CIBIV MFPL

How to infer a tree from data

Data	Method	Evaluation Criterion
Characters (Alignment)	**Maximum Parsimony**	Parsimony
	Statistical Approaches: Likelihood, Bayesian	Evolutionary Models
Distances	**Distance Methods**	Evolutionary Models

CIBIV MFPL

The Maximum Parsimony Principle

William of Ockham, 1285-1347/49

Occam's Razor (law of parsimony) states:

> *Pluralitas non est ponenda sine necessitate.*
> *Plurality should not be posited without necessity.*

The principle gives precedence to simplicity; of two competing theories the simpler explanation for an observation is to be preferred.

CIBIV MFPL

The Maximum Parsimony Criterion

Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T

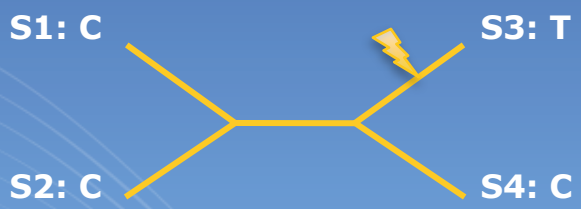
CIBIV MFPL

The Maximum Parsimony Criterion

Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T

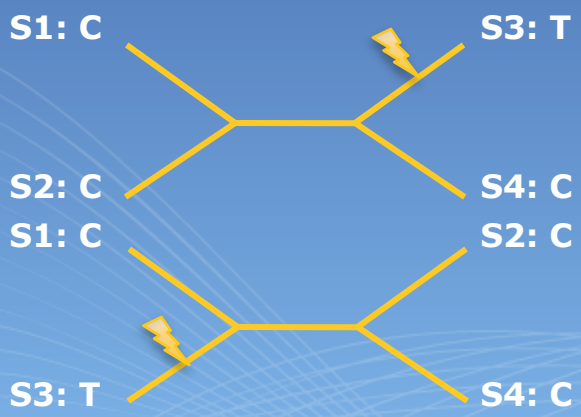
The Maximum Parsimony Criterion

Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T



The Maximum Parsimony Criterion

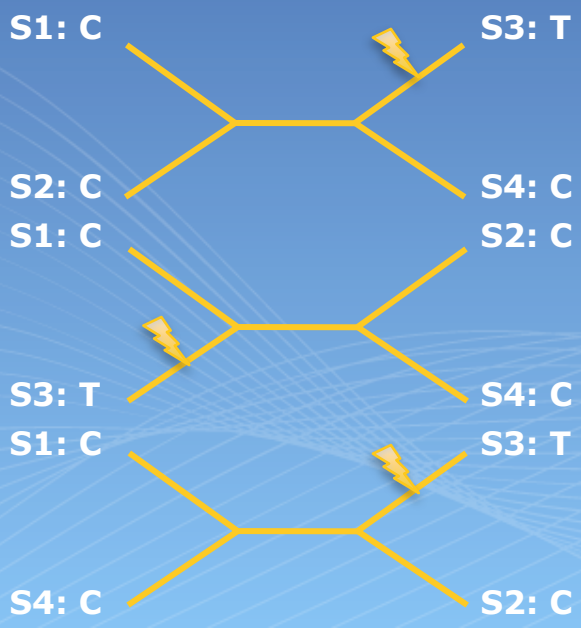
Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T



CIBIV MFPL

The Maximum Parsimony Criterion

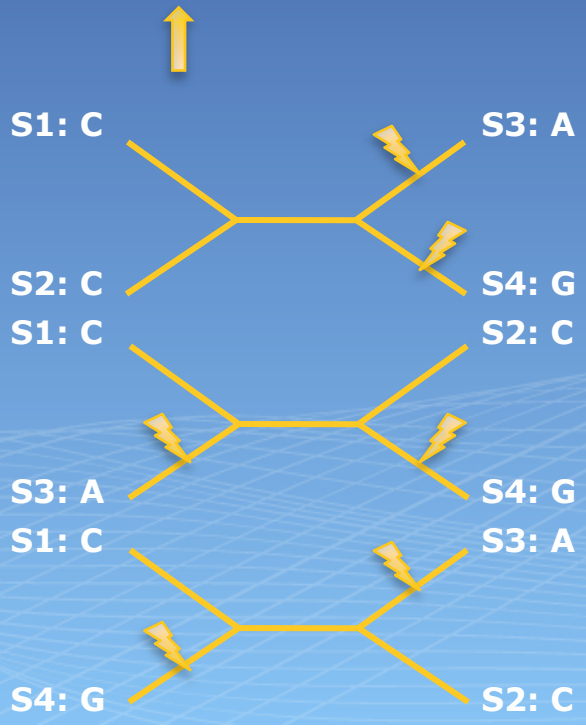
Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T



CIBIV MFPL

The Maximum Parsimony Criterion

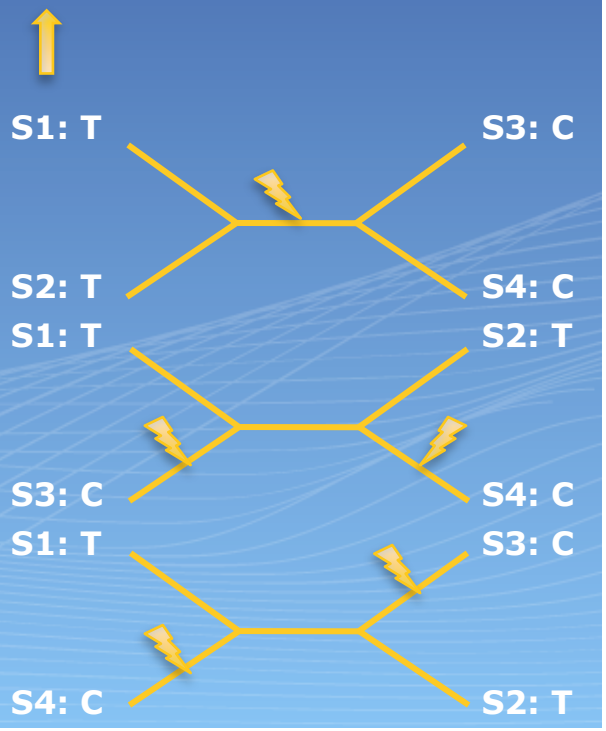
Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T



CIBIV MFPL

The Maximum Parsimony Criterion

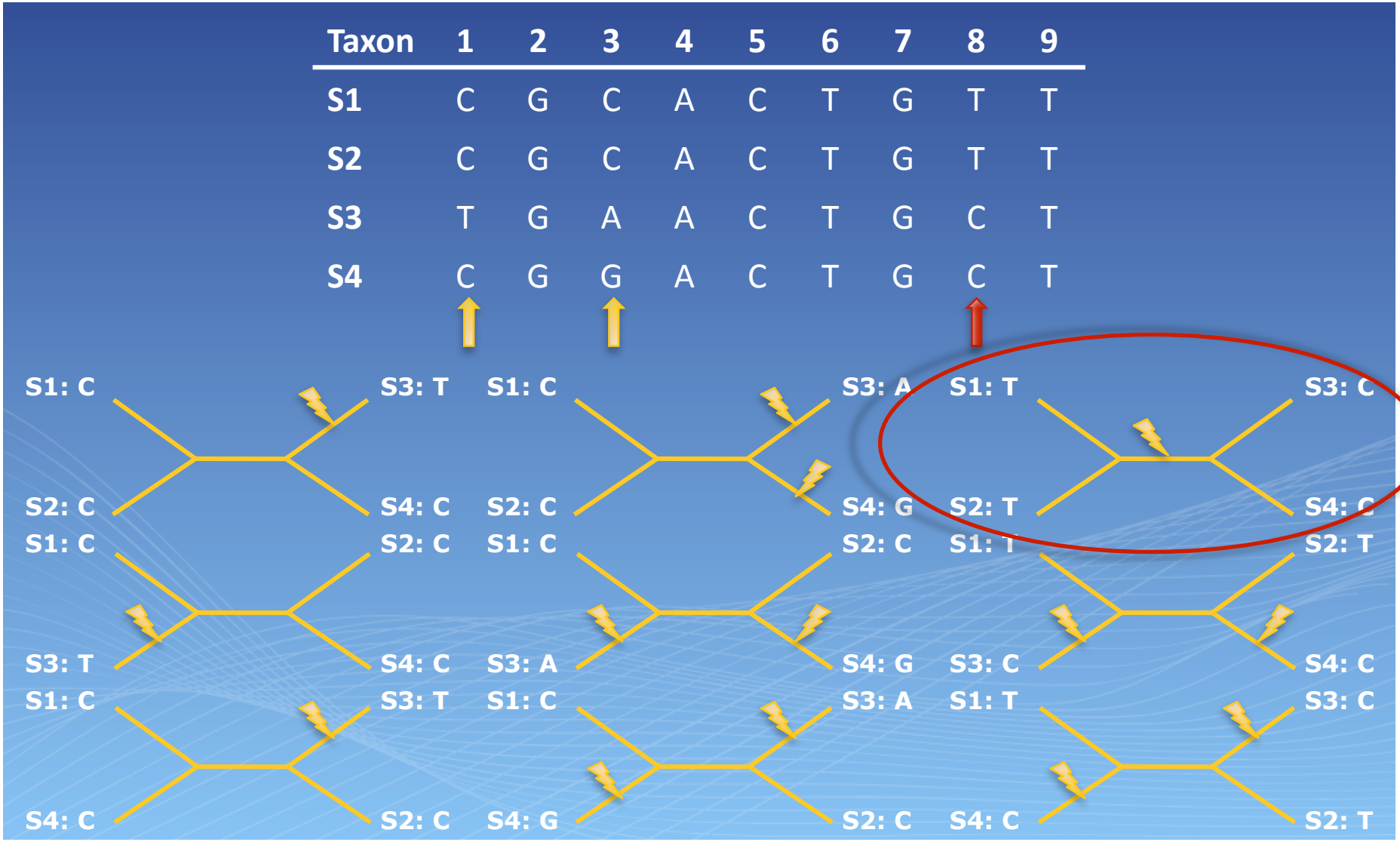
Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T



CIBIV MFPL

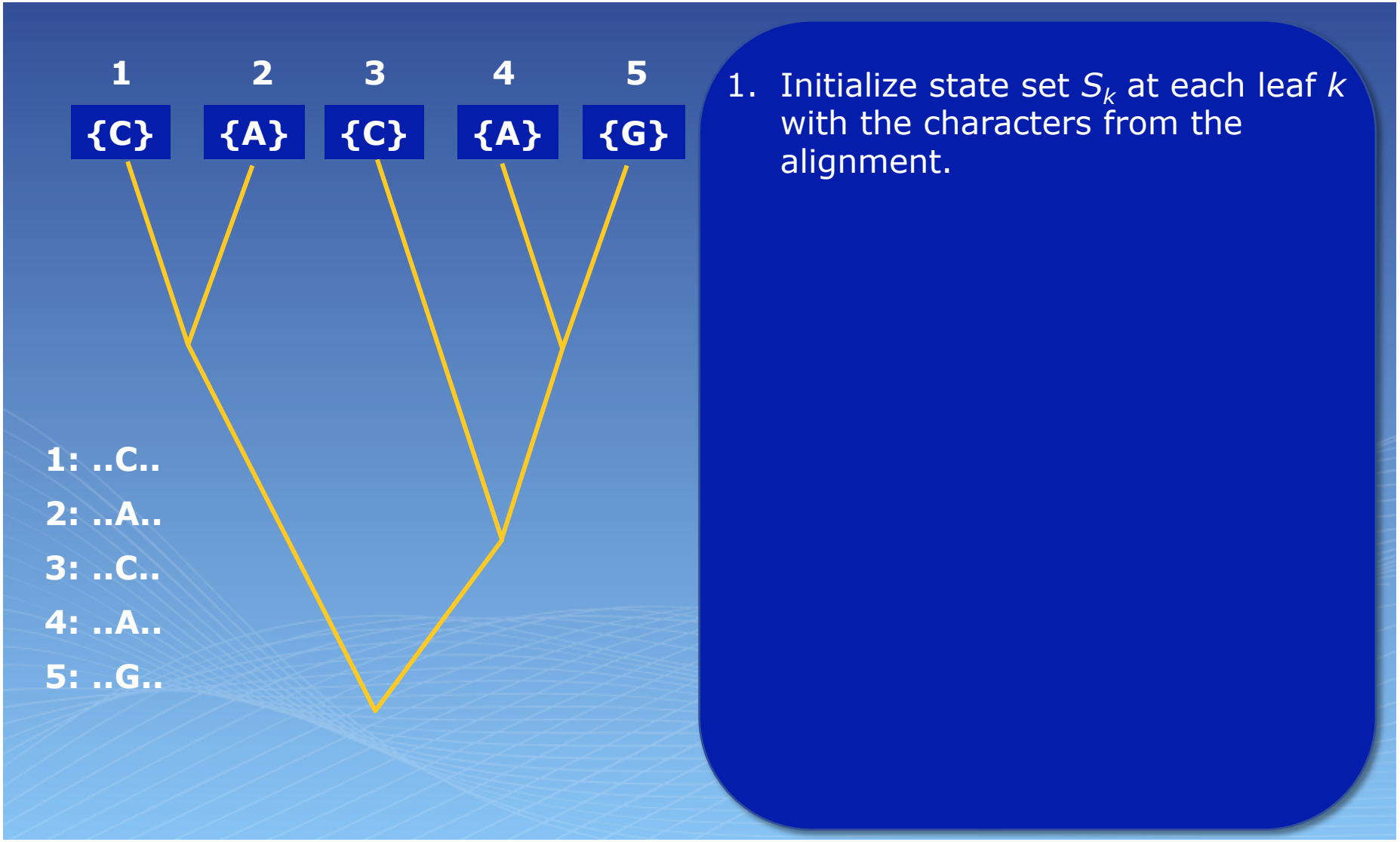
The Maximum Parsimony Tree

Taxon	1	2	3	4	5	6	7	8	9
S1	C	G	C	A	C	T	G	T	T
S2	C	G	C	A	C	T	G	T	T
S3	T	G	A	A	C	T	G	C	T
S4	C	G	G	A	C	T	G	C	T



CIBIV MFPL

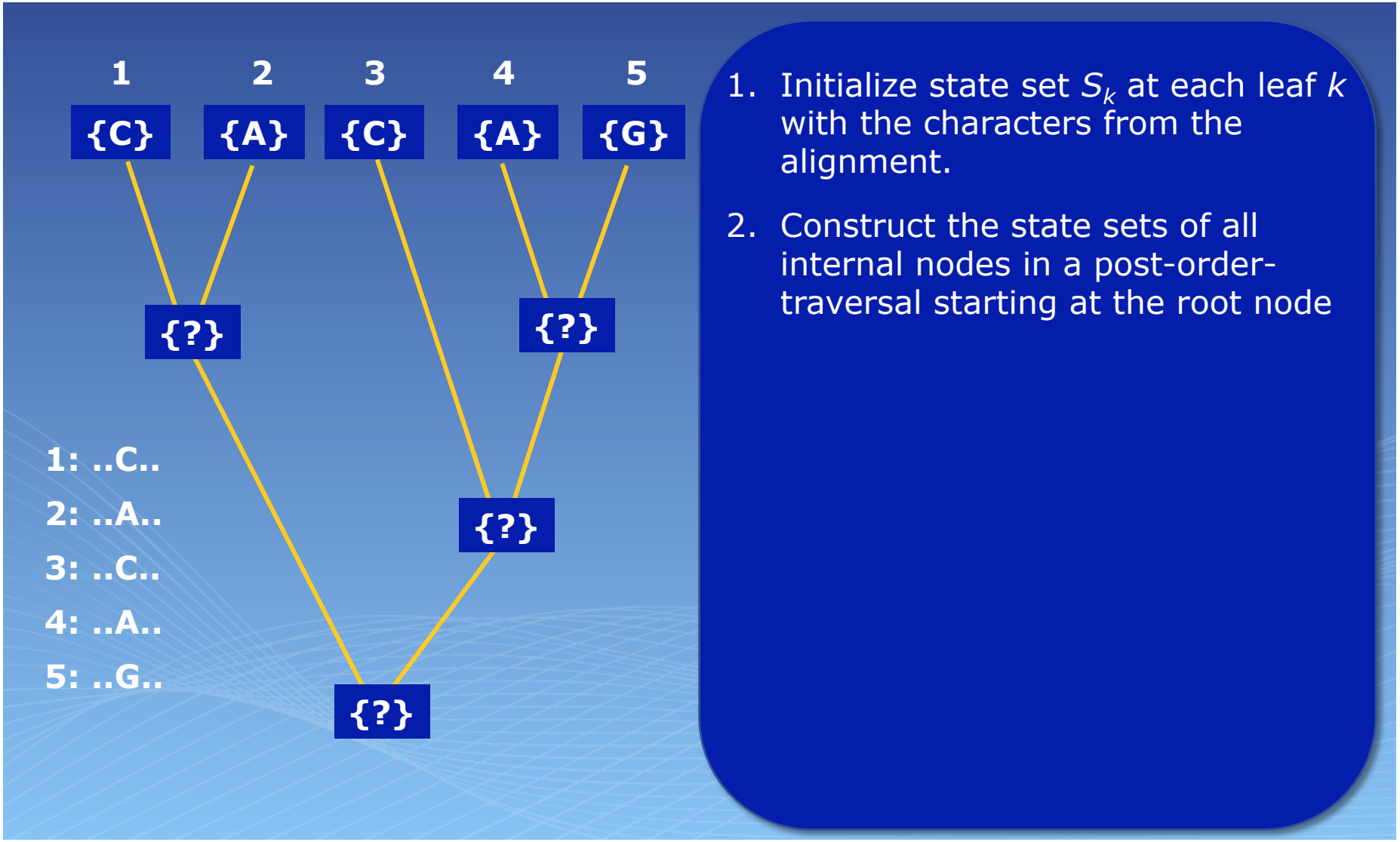
The Fitch algorithm (1970)



1. Initialize state set S_k at each leaf k with the characters from the alignment.

CIBIV MFPL

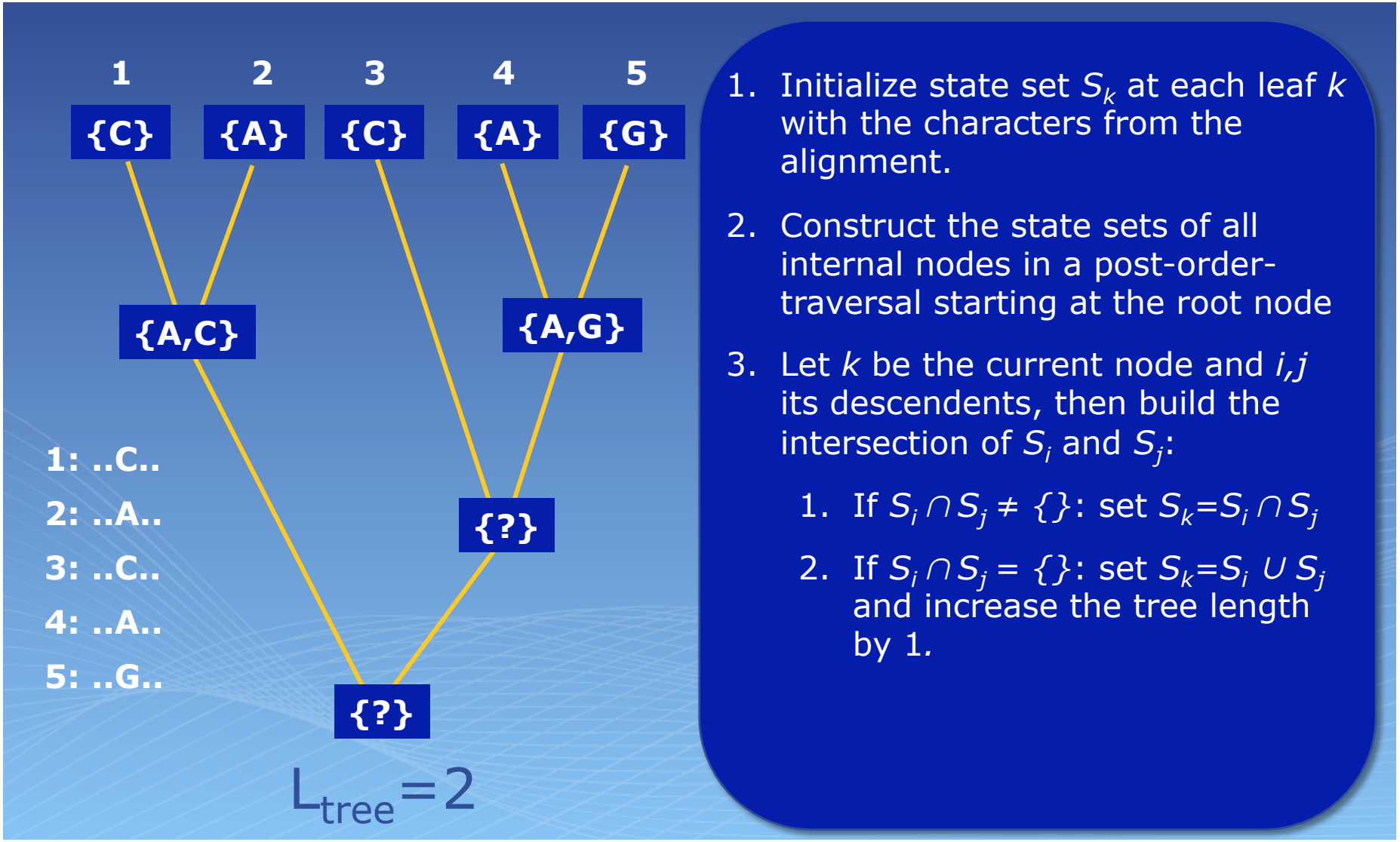
The Fitch algorithm (1970)



1. Initialize state set S_k at each leaf k with the characters from the alignment.
2. Construct the state sets of all internal nodes in a post-order-traversal starting at the root node

CIBIV MFPL

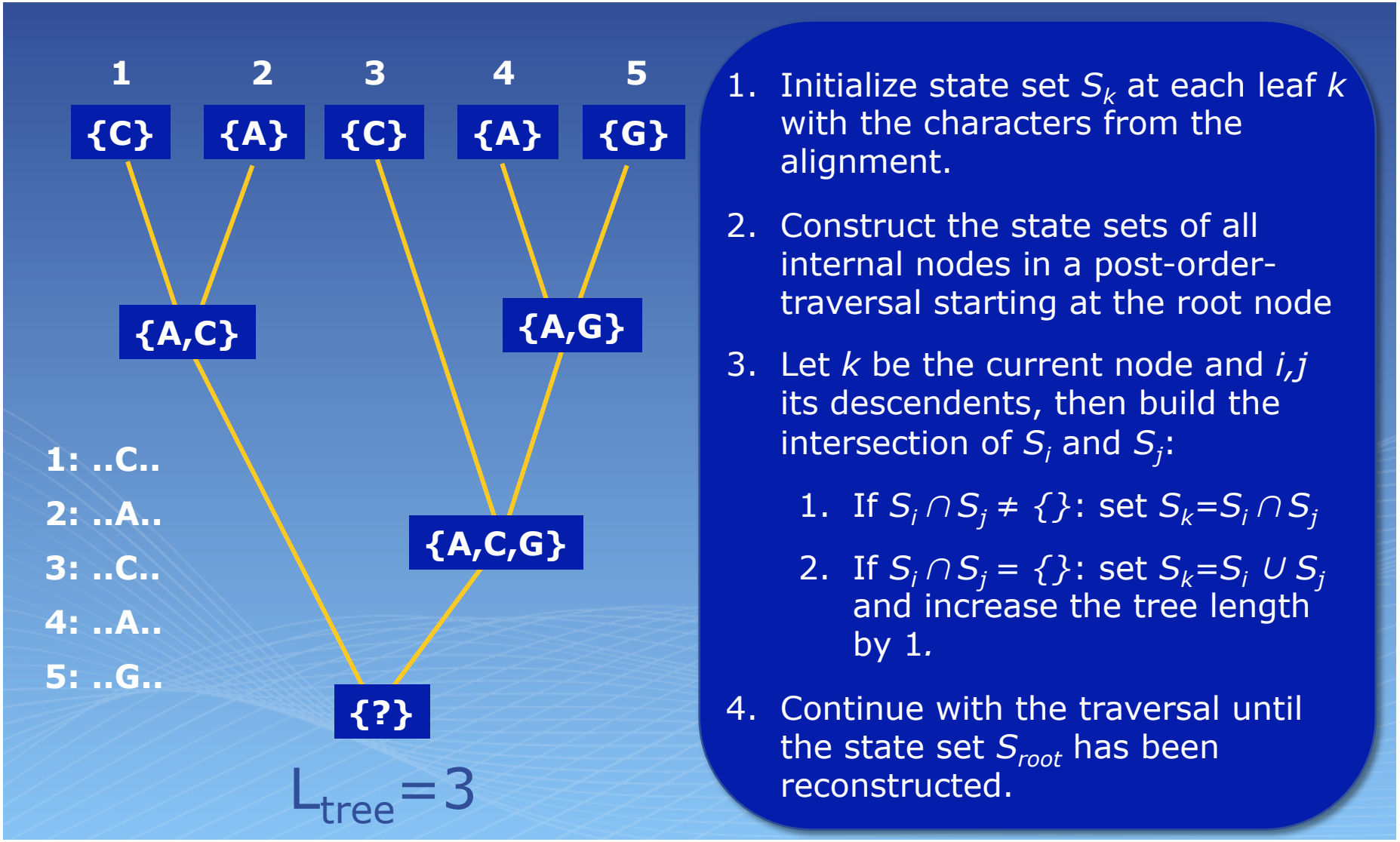
The Fitch algorithm (1970)



1. Initialize state set S_k at each leaf k with the characters from the alignment.
2. Construct the state sets of all internal nodes in a post-order-traversal starting at the root node
3. Let k be the current node and i,j its descendents, then build the intersection of S_i and S_j:
    1. If $S_i \cap S_j \neq \{\}$: set $S_k=S_i \cap S_j$
    2. If $S_i \cap S_j = \{\}$: set $S_k=S_i \cup S_j$ and increase the tree length by 1.

CIBIV MFPL

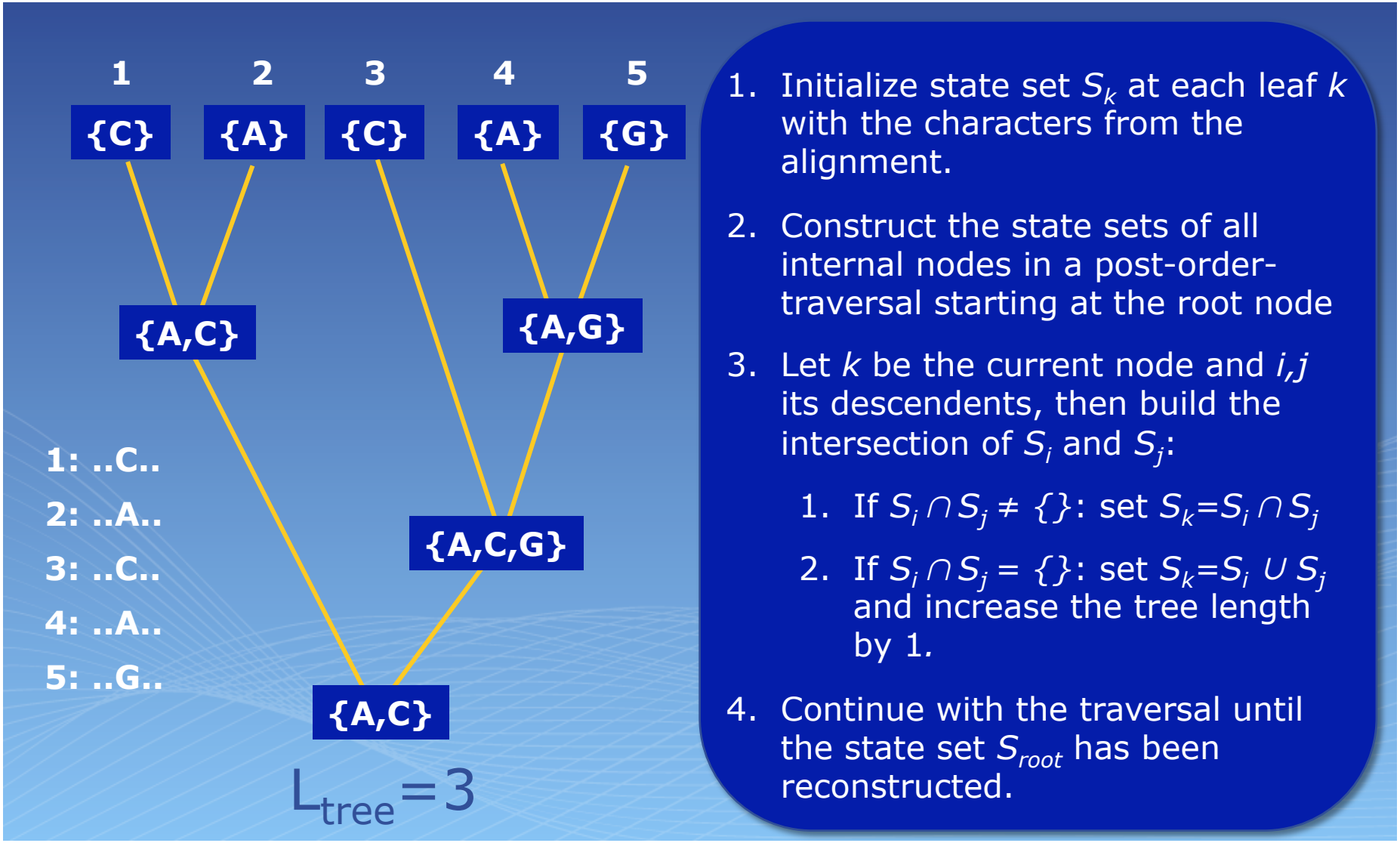
The Fitch algorithm (1970)



1. Initialize state set S_k at each leaf k with the characters from the alignment.
2. Construct the state sets of all internal nodes in a post-order-traversal starting at the root node
3. Let k be the current node and i,j its descendents, then build the intersection of S_i and S_j:
    1. If $S_i \cap S_j \neq \{\}$: set $S_k=S_i \cap S_j$
    2. If $S_i \cap S_j = \{\}$: set $S_k=S_i \cup S_j$ and increase the tree length by 1.
4. Continue with the traversal until the state set S_{root} has been reconstructed.

CIBIV MFPL

The Fitch algorithm (1970)



1. Initialize state set S_k at each leaf k with the characters from the alignment.
2. Construct the state sets of all internal nodes in a post-order-traversal starting at the root node
3. Let k be the current node and i,j its descendents, then build the intersection of S_i and S_j:
    1. If $S_i \cap S_j \neq \{\}$: set $S_k=S_i \cap S_j$
    2. If $S_i \cap S_j = \{\}$: set $S_k=S_i \cup S_j$ and increase the tree length by 1.
4. Continue with the traversal until the state set S_{root} has been reconstructed.

CIBIV MFPL

MP Objective Function

Aim: Find the tree *T* that minimizes the following function

$$L(T) = \sum_{k=1}^{B} \sum_{j=1}^{A} \omega_j * diff(x_{k'j}, x_{k''j})$$

***diff*:** Scoring matrix for changes

ω_j**:** Alignment-specific weight (often $\omega_j=1$, *for all j)*

***A*:** Alignment length

***B*:** Number of branches in *T*

***k', k''*:** Endnotes of branch *k*

CIBIV MFPL

Aspects of Maximum Parsimony

1. Parsimony is often considered model-free
2. One has no choice of a model, but the algorithm assumes that changes are rare and backmutations do not occur (model)
3. Although assumption 2 is often true for morphological data, it is certainly not true for biological sequence data

CIBIV MFPL

The criterion of distance

S1	A	G	C	T	T	A	C	C	T	T	T	T	A	C	T
S2	C	G	T	A	A	A	T	T	T	C	C	C	G	A	T
S3	C	G	C	A	A	G	T	T	T	C	C	C	G	A	T
S4	C	A	C	T	T	A	T	T	A	G	T	C	A	A	C

CIBIV MFPL

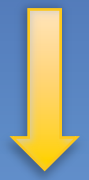
Edit distance (Hamming)

S1	A	G	C	T	T	A	C	C	T	T	T	T	A	C	T
S2	C	G	T	A	A	A	T	T	T	C	C	C	G	A	T
S3	C	G	C	A	A	G	T	T	T	C	C	C	G	A	T
S4	C	A	C	T	T	A	T	T	A	G	T	C	A	A	C

CIBIV MFPL

Edit distance (Hamming)

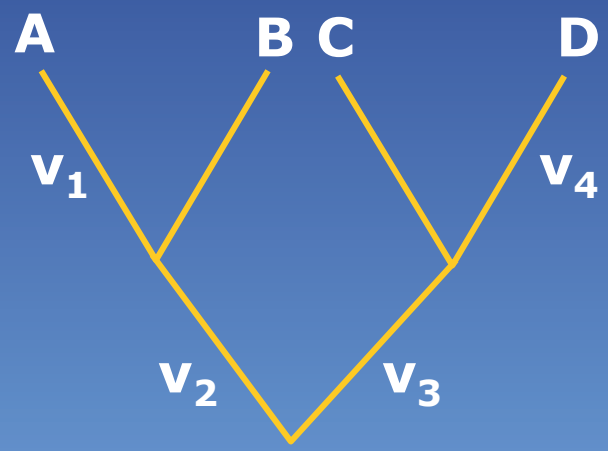
S1	A	G	C	T	T	A	C	C	T	T	T	T	A	C	T
S2	C	G	T	A	A	A	T	T	T	C	C	C	G	A	T
S3	C	G	C	A	A	G	T	T	T	C	C	C	G	A	T
S4	C	A	C	T	T	A	T	T	A	G	T	C	A	A	C



	S1	S2	S3	S4
S1	0	11	11	8
S2	11	0	2	10
S3	11	2	0	9
S4	8	10	9	0

CIBIV MFPL

Distance based tree reconstruction



Aim: Find branch lengths v_b such that the sum of the branch lengths connecting any two leaves gets close to the measured distances between all pairs of leaves, e.g.

$$d_{A,D}^{measured} \approx v_1 + v_2 + v_3 + v_4$$

CIBIV MFPL

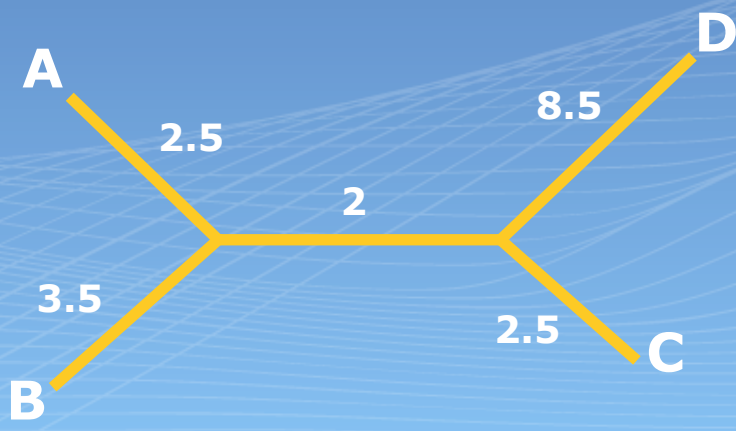
The Four-Point-Condition

Theorem: Four-Point-Condition

A distance matrix $(d_{i,j})$, $i,j=1....n$, is representable as a tree, if and only if

$$d(a,b)+d(c,d)\le \max\{d(a,c)+d(b,d),d(a,d)+d(b,c)\}$$
$$\text{for all } a,b,c,d\in\{1,2,...,n\}$$

	A	B	C
B	6		
C	7	8	
D	12	14	11



CIBIV MFPL

The ultrametric inequality

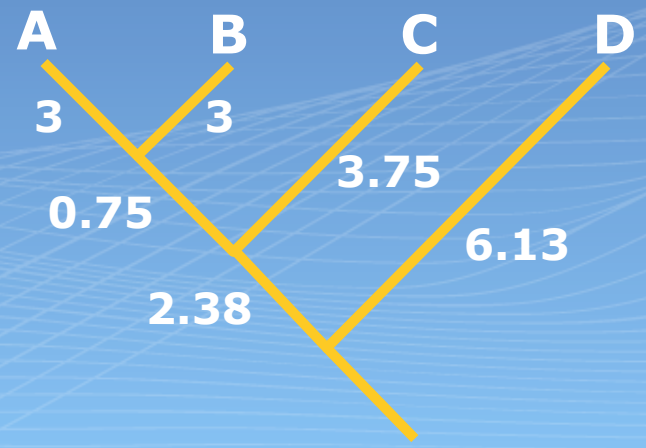
Theorem: The ultrametric inequality

A distance matrix $(d_{i,j})$, $i,j=1....n$, is representable as a clock-like tree, if and only if

$$d(A,B) \leq \max\{d(A,C),d(B,C)\}$$

for all triple (A,B,C)

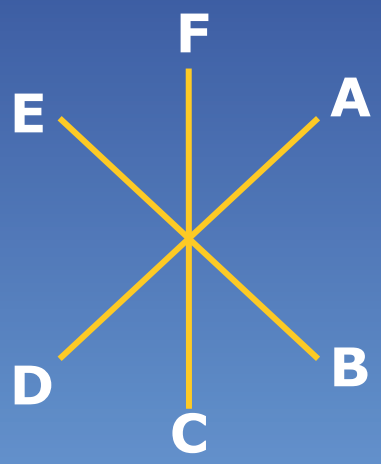
	A	B	C
B	6		
C	7	8	
D	12	14	11



CIBIV MFPL

The Neighbor Joining Algorithm*

1. begin with a star tree:



2. compute for each pair (*1,2*) the net-divergence

$$\frac{1}{2(n-2)}\sum_{k=3}^{N} D_{1k} + D_{2k}) + \frac{1}{2}D_{12} + \frac{1}{N-2}\sum_{3\le i<j} D_{ij} \quad (1)$$

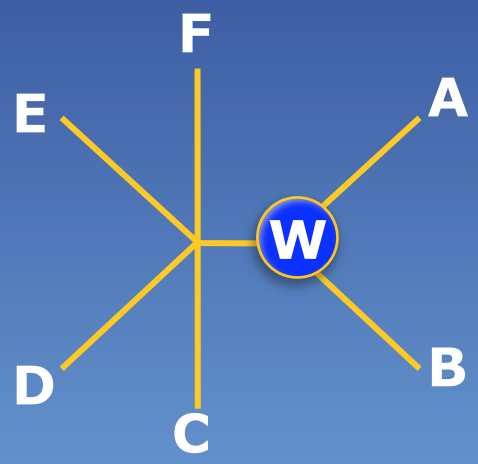
3. take the pair (*A,B*) that minimizes Eq. (1)

Saitou and Nei (1987), Mol Biol Evol 4:406-425

CIBIV MFPL

The Neighbor Joining Algorithm

4. cluster (*A,B*) and define an interior node *W*



5. compute branch lengths for the external edges:

$$v(A,W) = \frac{1}{2}\left(D(A,B) + \frac{1}{m-2}\sum_{k=3}^{m} D(A,k) - D(B,k) \right)$$

$$v(B,W) = D(A,B) - v(A,W)$$

CIBIV MFPL

The Neighbor Joining Algorithm

6. compute distance *W* to the remaining *m-2* leaves:

$$D(W,k) = \frac{1}{2}\left(D(A,k) + D(B,k) - D(A,B)\right)$$

7. continue with step 1 with the reduced set of leaves

CIBIV MFPL

The Maximum Likelihood criterion

Modelling sequence evolution

CIBIV MFPL

The Maximum Likelihood criterion

Modelling sequence evolution

1. **First order Markov process**
   The evolutionary process is memory less, i.e. sequence S_2 mutates to S_3 during time t_{n+1} independent of S_1

CIBIV MFPL

The Maximum Likelihood criterion

Modelling sequence evolution

1. **First order Markov process**
   The evolutionary process is memory less, i.e. sequence S_2 mutates to S_3 during time t_{n+1} independent of S_1
2. **Stationary**
   The overall character frequencies π_j of the nucleotides or amino acids remain constant.

CIBIV MFPL

The Maximum Likelihood criterion

Modelling sequence evolution

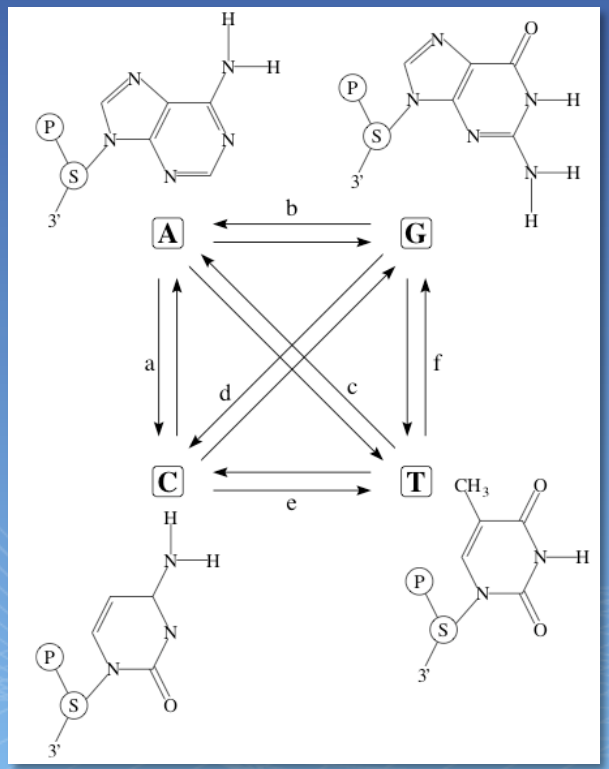
1. **First order Markov process**
   The evolutionary process is memory less, i.e. sequence S_2 mutates to S_3 during time t_{n+1} independent of S_1
2. **Stationary**
   The overall character frequencies π_j of the nucleotides or amino acids remain constant.
3. **Time reversible**

$$\pi_i \cdot P_{ij}(t) = P_{ji}(t) \cdot \pi_j$$

CIBIV MFPL

Modelling sequence evolution

Evolutionary models are often described using a substitution rate matrix Q and character frequencies Π.



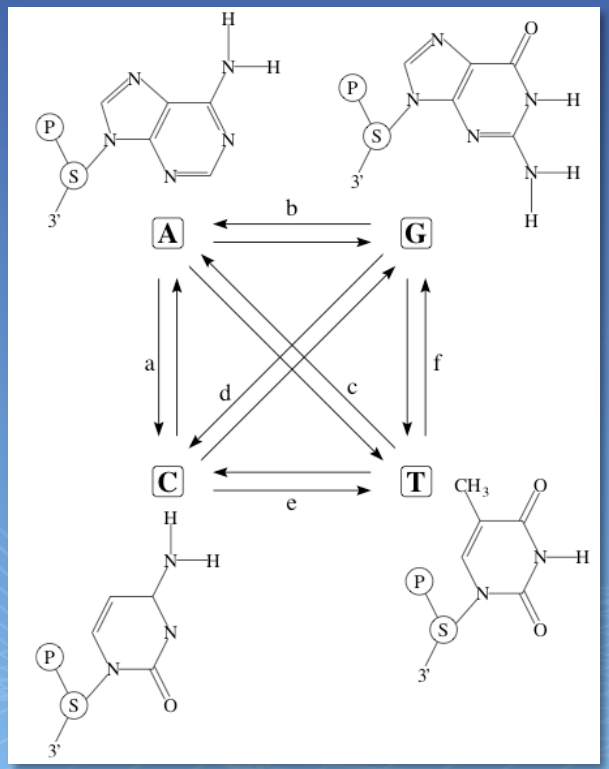
$$Q = \begin{array}{c} \begin{matrix} A & C & G & T \end{matrix} \\ \begin{pmatrix} - & a & b & c \\ a & - & d & e \\ b & d & - & f \\ c & e & f & - \end{pmatrix} \end{array}$$

$$\Pi = \left(\pi_A, \pi_C, \pi_G, \pi_T\right)$$

CIBIV MFPL

Modelling sequence evolution

Evolutionary models are often described using a substitution rate matrix Q and character frequencies Π.



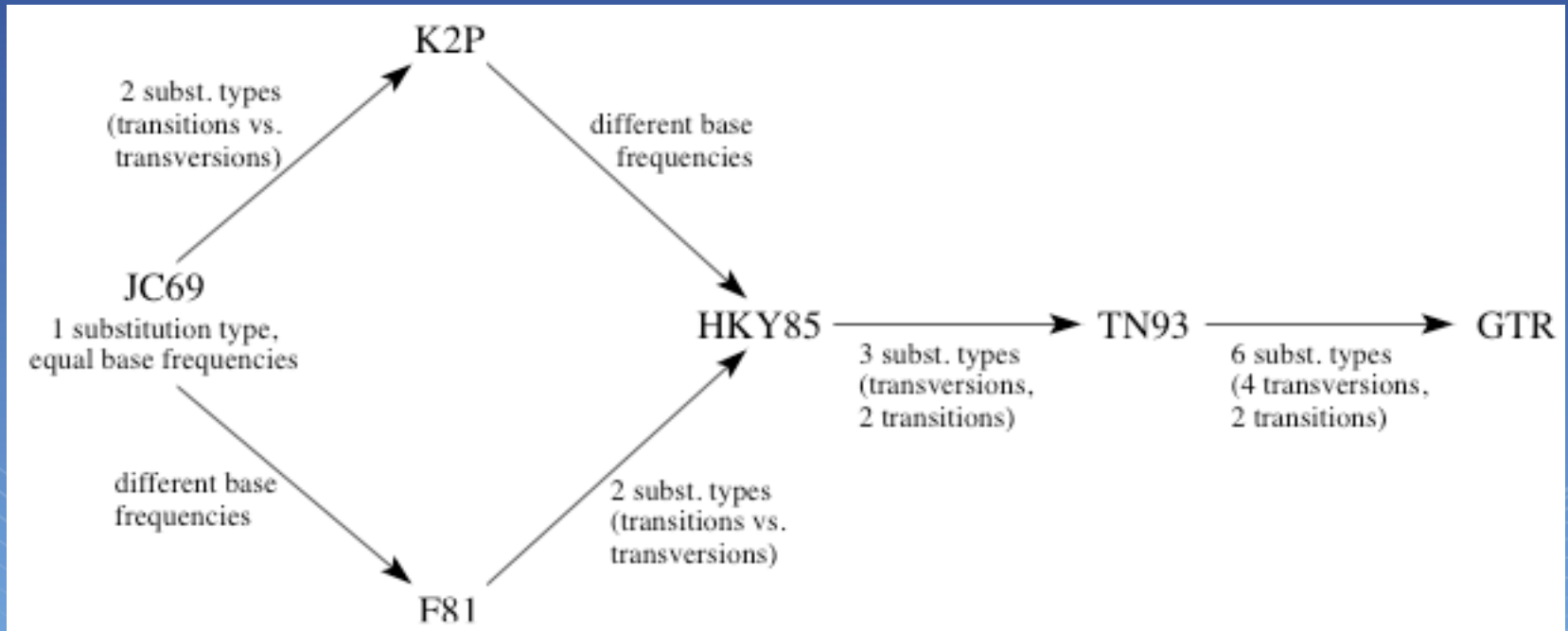
$$Q = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} - & a & b & c \\ a & - & d & e \\ b & d & - & f \\ c & e & f & - \end{pmatrix} \end{array}$$

$$\Pi = (\pi_A, \pi_C, \pi_G, \pi_T)$$

From Q and Π we reconstruct a substitution probability matrix P where $P_{ij}(t)$ is the probability of changing i to j in time t.

CIBIV MFPL

DNA sequence evolution models



Further modification:

rate heterogeneity: invariant sites, Γ-distributed rates, mixed.

CIBIV MFPL

Protein sequence evolution models

Generally this is the same for protein sequences, but with 20 × 20 matrices. Some protein models are:

- Poisson model ("JC69" for proteins, rarely used)
- Dayhoff (Dayhoff *et al.*, 1978, general matrix)
- JTT (Jones *et al.*, 1992, general matrix)
- WAG (Whelan & Goldman, 2000, more distant sequences)
- VT (Müller & Vingron, 2000, distant sequences)
- mtREV (Adachi & Hasegawa, 1996, mitochondrial sequences)
- cpREV (Adachi *et al.*, 2000, cloroplast sequences)
- mtMAM (Yang *et al.*, 1998, Mammalian mitochondria)
- mtART (Abascal *et al.*, 2007, Arthropod mitochondria)
- rtREV (Dimmic *et al.*, 2002, reverse transcriptases)
- ...

CIBIV MFPL

The Likelihood function

The likelihood of sequence s evolving to s’ in time t:

S: GGTCCTGACAGAAATAAAC

S‘: GATCCTGAGAGAAATAAAC

$$L(t \mid s \rightarrow s') = \prod_{i=1}^{m} \left(\pi_{s_i} \times P_{s_i s'_i}(t) \right)$$

CIBIV MFPL

The Likelihood function

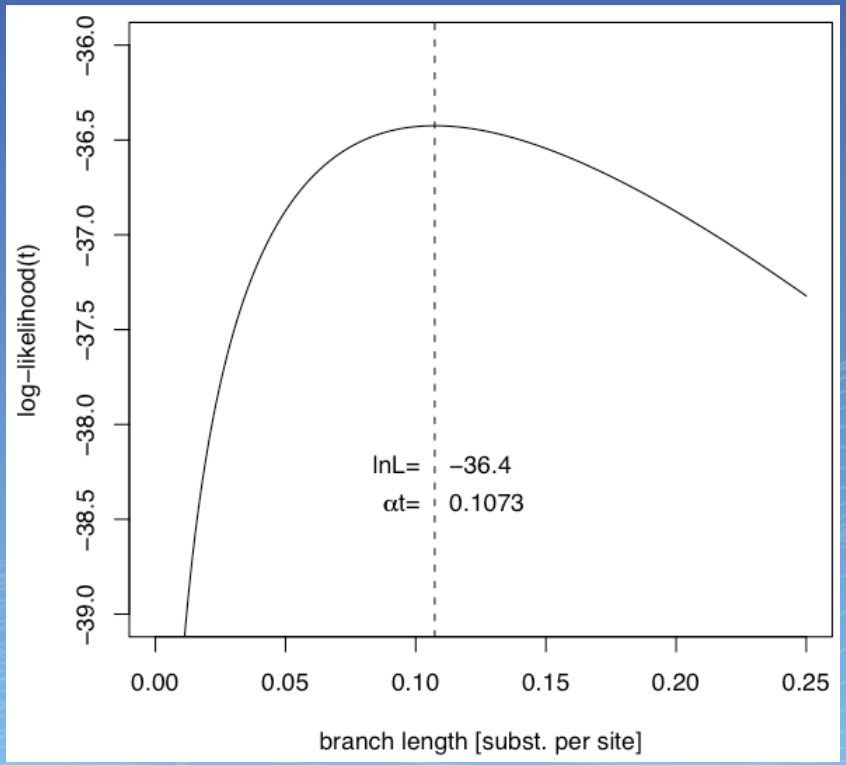
The likelihood of sequence s evolving to s’ in time t:

S: GGTCCTGACAGAAATAAAC

S’: GATCCTGAGAGAAATAAAC

$$L(t \mid s \rightarrow s') = \prod_{i=1}^{m} \left(\pi_{s_i} \times P_{s_i s'_i}(t) \right)$$

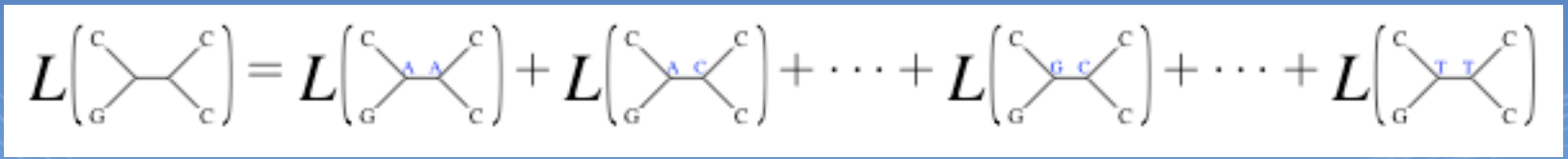
Log-Likelihood surface under JC69



CIBIV MFPL

Tree likelihoods

Given a tree with branch lengths and sequences for all nodes, the computation of likelihood values is straightforward. Usually no sequences are available for the inner nodes (ancestral sequences). Hence we have to evaluate every possible labeling at the inner nodes:

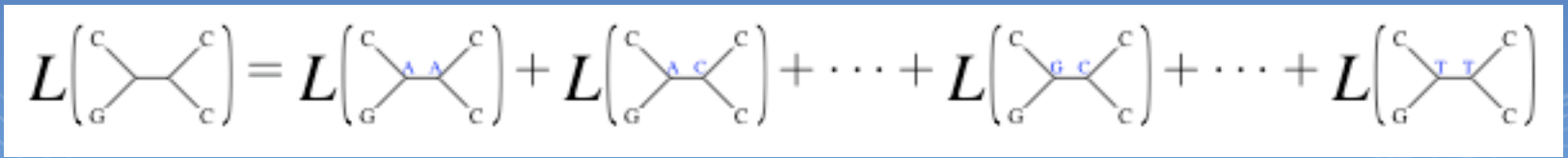


for every column in the alignment.

CIBIV MFPL

Tree likelihoods

Given a tree with branch lengths and sequences for all nodes, the computation of likelihood values is straightforward. Usually no sequences are available for the inner nodes (ancestral sequences). Hence we have to evaluate every possible labeling at the inner nodes:

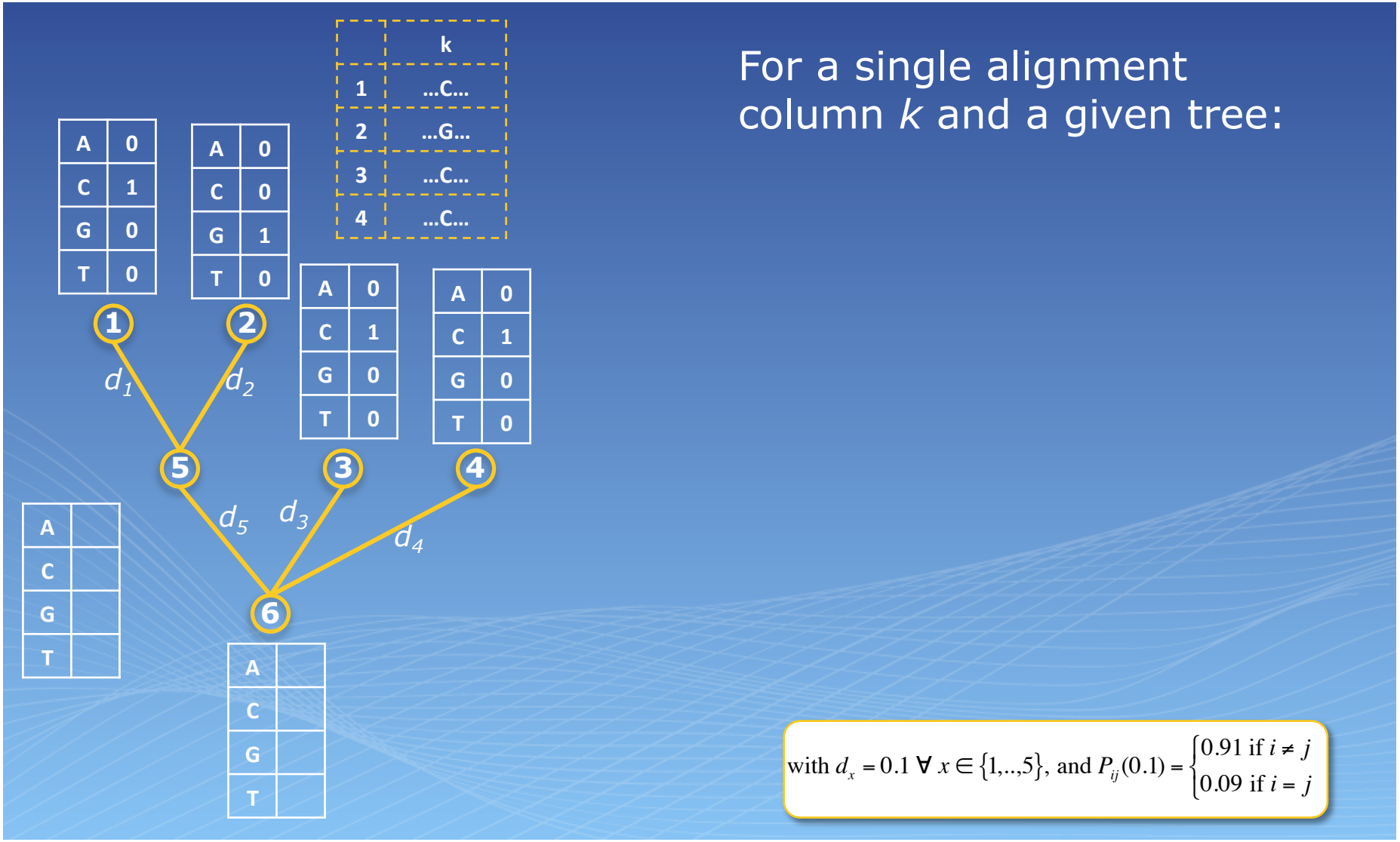


for every column in the alignment.

But there is a faster algorithm...

CIBIV MFPL

Calculating tree likelihoods

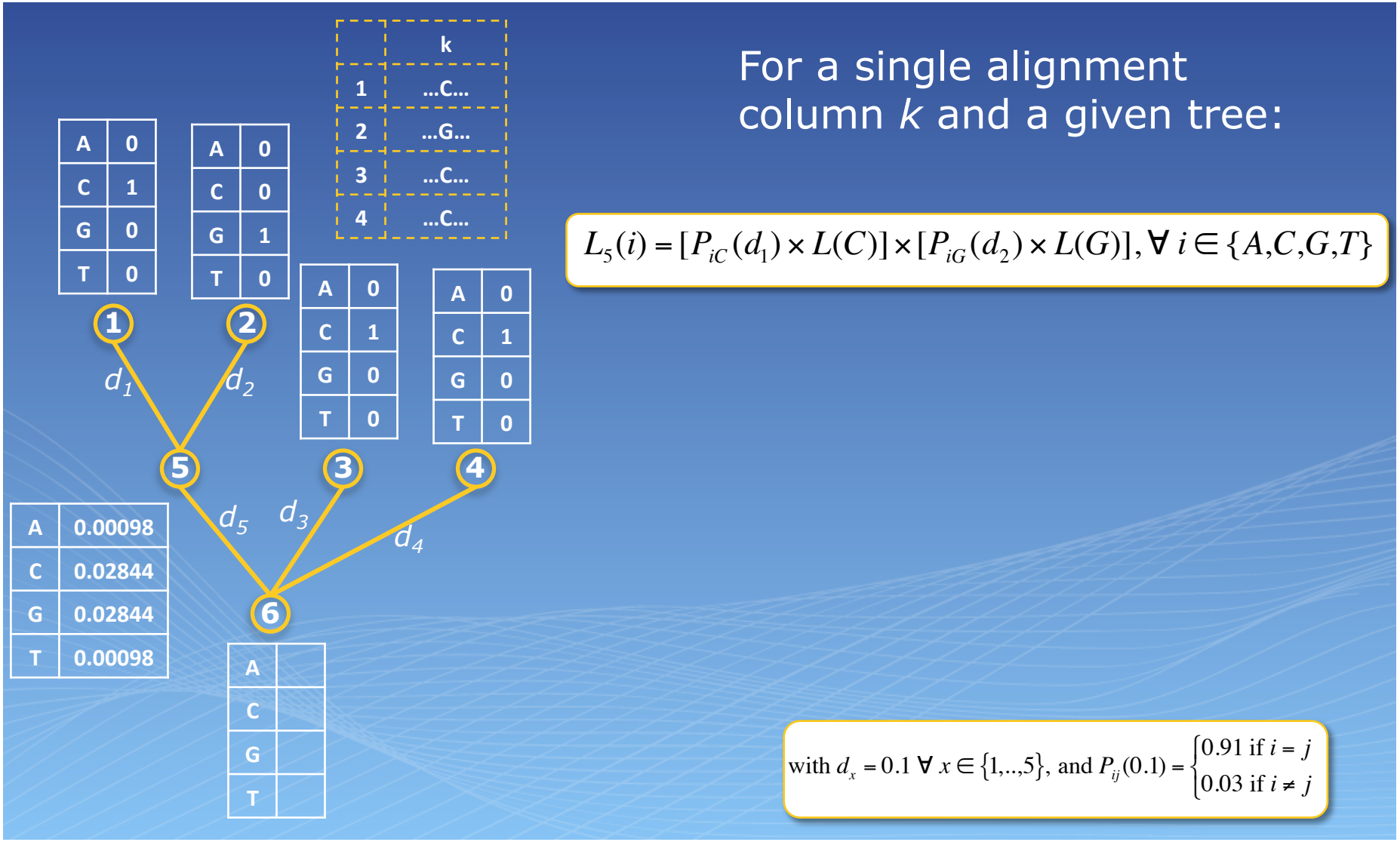


For a single alignment column *k* and a given tree:

$$\text{with } d_x = 0.1 \;\forall\; x \in \{1,..,5\}, \text{ and } P_{ij}(0.1) = \begin{cases} 0.91 \text{ if } i \neq j \\ 0.09 \text{ if } i = j \end{cases}$$

CIBIV MFPL

Calculating tree likelihoods



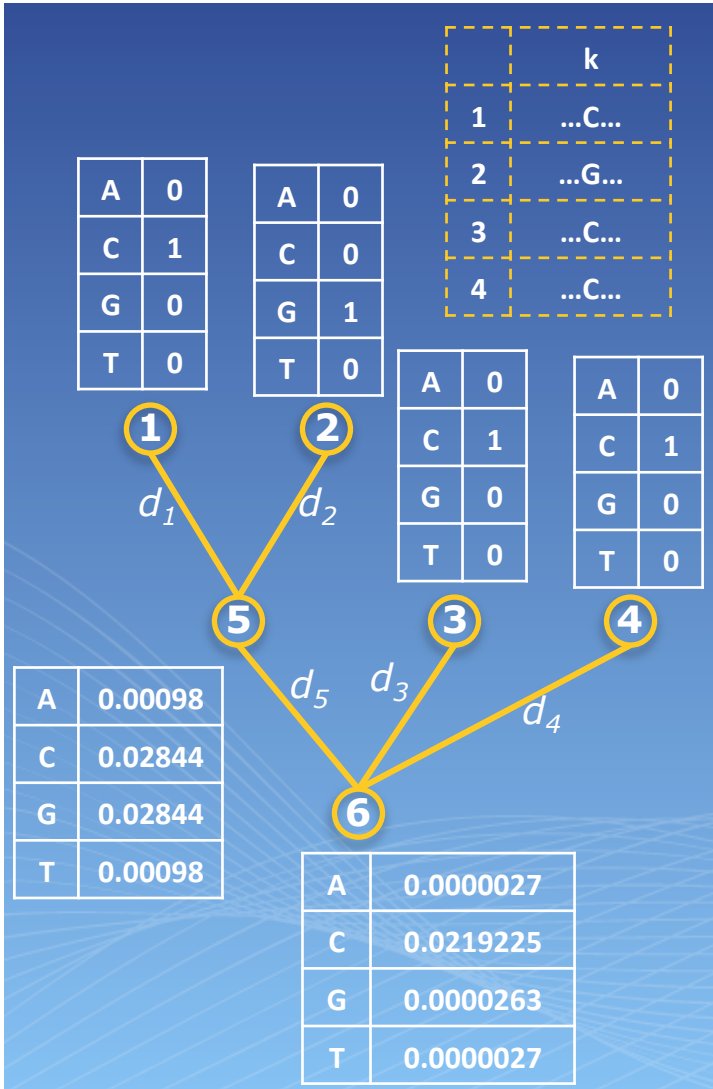
For a single alignment column *k* and a given tree:

$$L_5(i) = [P_{iC}(d_1) \times L(C)] \times [P_{iG}(d_2) \times L(G)], \forall\, i \in \{A,C,G,T\}$$

with $d_x = 0.1\ \forall\ x \in \{1,..,5\}$, and $P_{ij}(0.1) = \begin{cases} 0.91 \text{ if } i = j \\ 0.03 \text{ if } i \neq j \end{cases}$

CIBIV MFPL

Calculating tree likelihoods



For a single alignment column *k* and a given tree:

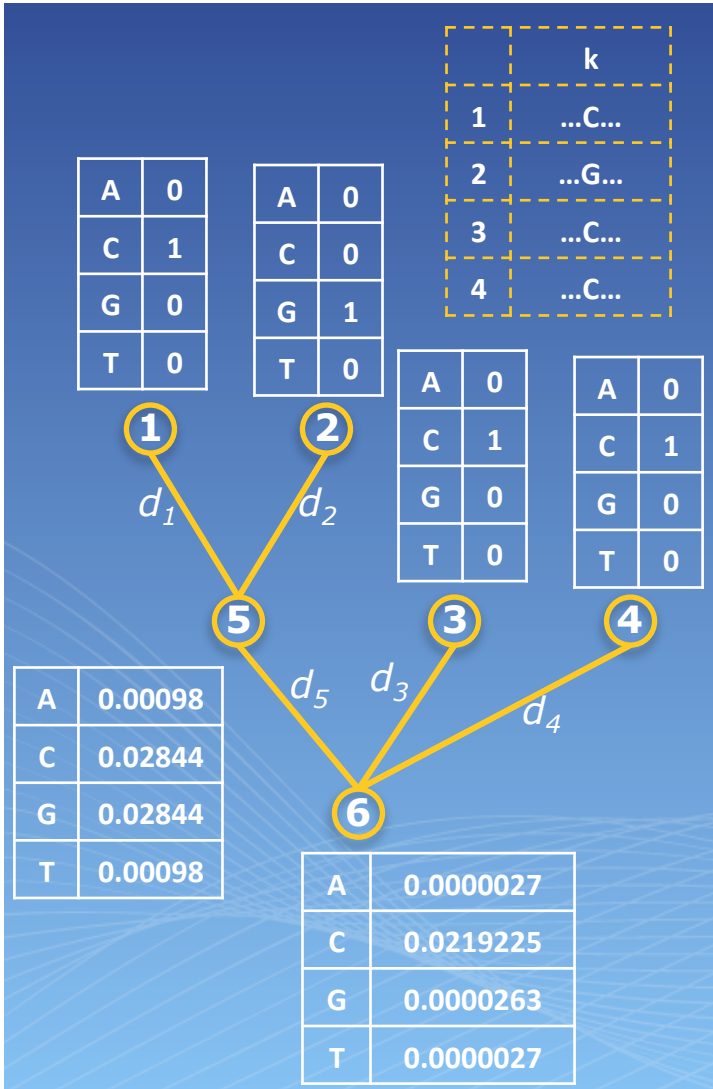
$$L_5(i) = [P_{iC}(d_1) \times L(C)] \times [P_{iG}(d_2) \times L(G)], \forall\, i \in \{A,C,G,T\}$$

$$L_6(i) = \prod_{v=\{3,4,5\}} \left[\sum_{j=\{A,C,G,T\}} P_{ij}(d_v) \times L_v(j) \right], \forall\, i \in \{A,C,G,T\}$$

$$\text{with } d_x = 0.1\ \forall\ x \in \{1,..,5\}, \text{ and } P_{ij}(0.1) = \begin{cases} 0.91 \text{ if } i = j \\ 0.03 \text{ if } i \neq j \end{cases}$$

CIBIV MFPL

Calculating tree likelihoods



For a single alignment column *k* and a given tree:

$$L_5(i) = [P_{iC}(d_1) \times L(C)] \times [P_{iG}(d_2) \times L(G)], \forall\, i \in \{A,C,G,T\}$$

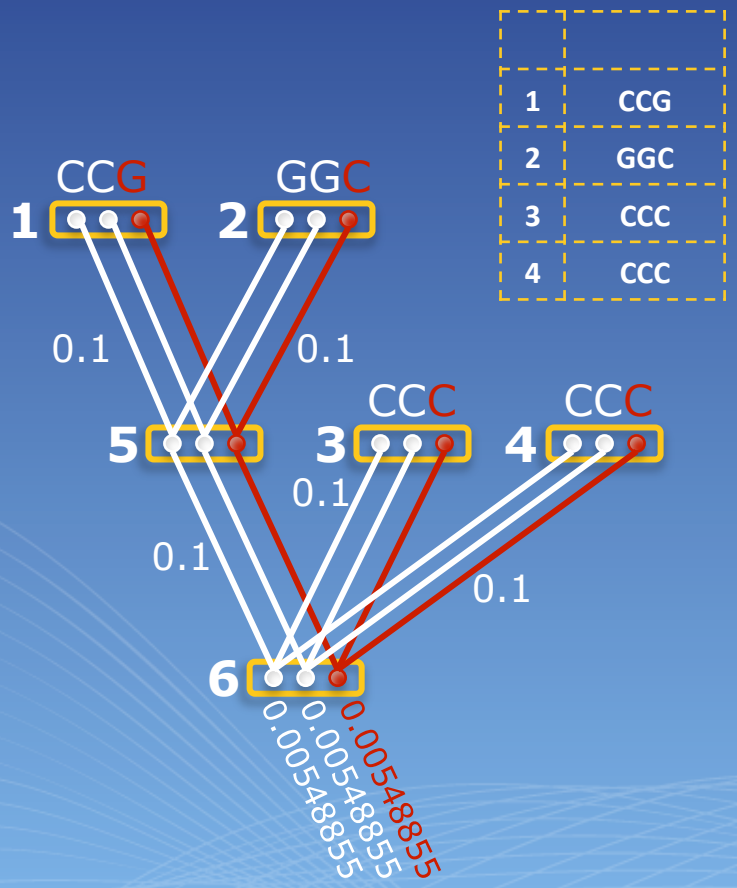
$$L_6(i) = \prod_{v=\{3,4,5\}} \left[\sum_{j=\{A,C,G,T\}} P_{ij}(d_v) \times L_v(j) \right], \forall\, i \in \{A,C,G,T\}$$

$$L^{(k)} = \sum_{i=\{A,C,G,T\}} \pi_i \times L_6(i) = 0.005489$$

with $d_x = 0.1\ \forall\ x \in \{1,..,5\}$, and $P_{ij}(0.1) = \begin{cases} 0.91 \text{ if } i = j \\ 0.03 \text{ if } i \neq j \end{cases}$

CIBIV MFPL

Calculating tree likelihoods



For an alignment of four sequences and length m=3 the likelihood is then

$$L(T) = \prod_{k=1}^{m} L^{(k)} = 0.005489^2 \times 0.005489$$
$$= 0.0000001653381$$

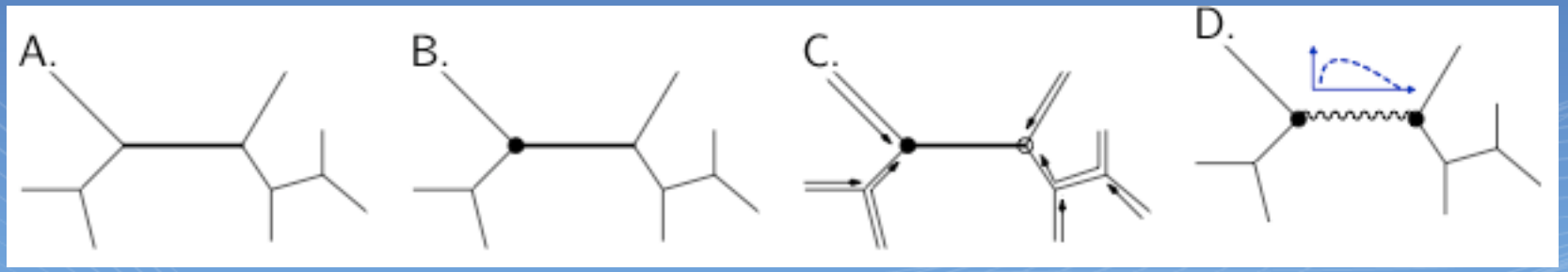
or the log-likelihood is

$$\ln L(T) = \sum_{k=1}^{m} \ln L^{(k)} = -15.61527$$

CIBIV MFPL

Optimizing branch lengths

To compute optimal branch lengths do the following. Initialize the branch lengths. Choose a branch (A). Move the virtual root to an adjacent node (B). Compute all partial likelihoods recursively (C). Adjust the branch length to maximize the likelihood value (D).



CIBIV MFPL

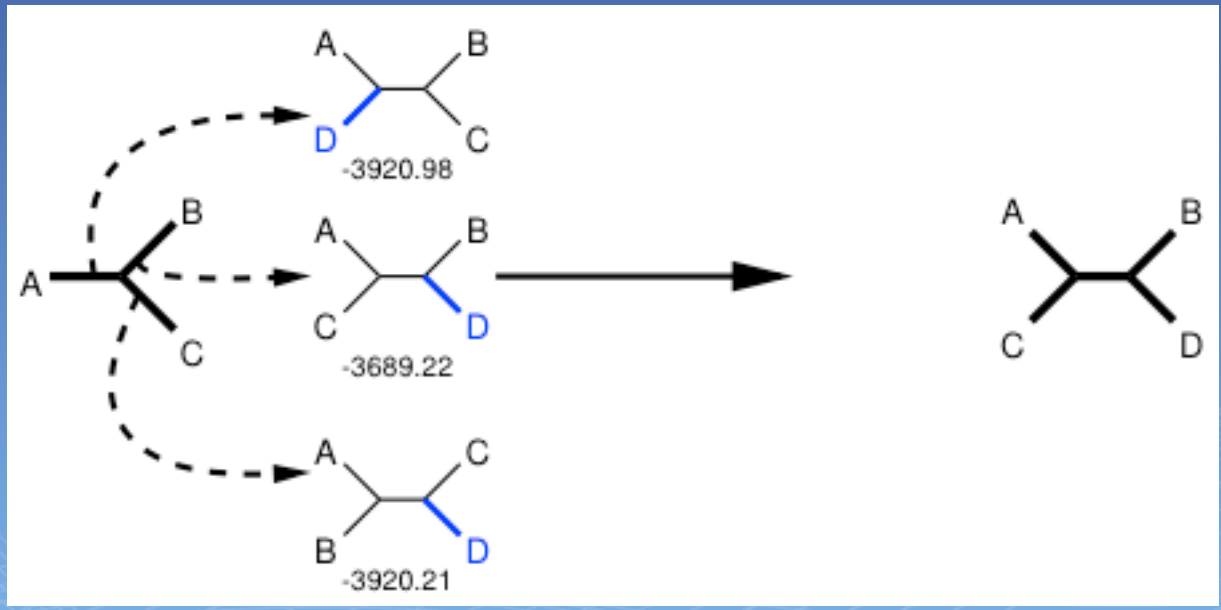
Finding the best tree

1. **Exhaustive Search:** evaluates every possible tree and hence an optimal solution is guaranteed. Limit: 10-12 taxa
2. **Branch and Bound:** excludes parts from the tree space from the search where the optimal tree cannot be found. Guarantees to find the optimal tree.
3. **Heuristics:** Can be applied to large taxon sets but does not guarantee an optimal solution

CIBIV MFPL

Finding the best tree: Heuristic search

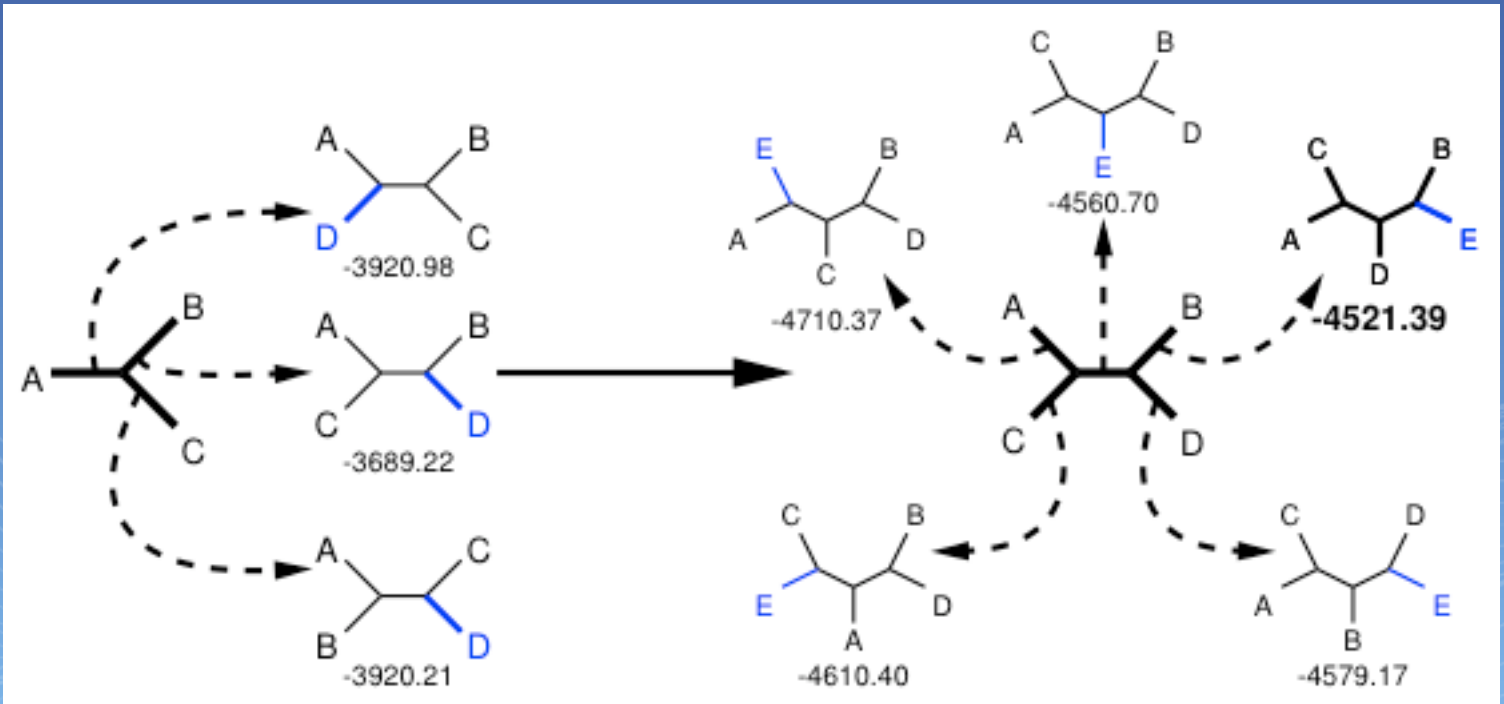
Building the tree: Stepwise insertion



CIBIV MFPL

Finding the best tree: Heuristic search

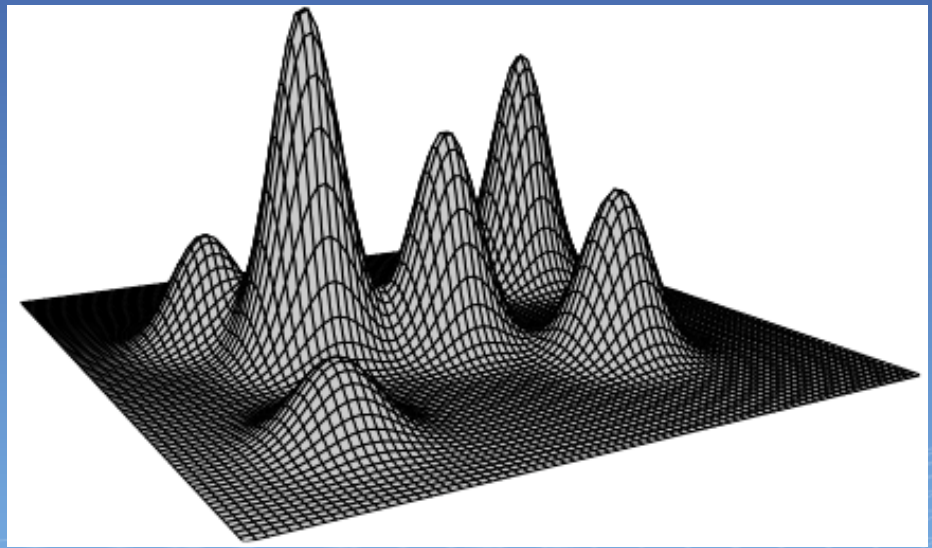
Building the tree: Stepwise insertion



CIBIV MFPL

Finding the best tree: Heuristic search

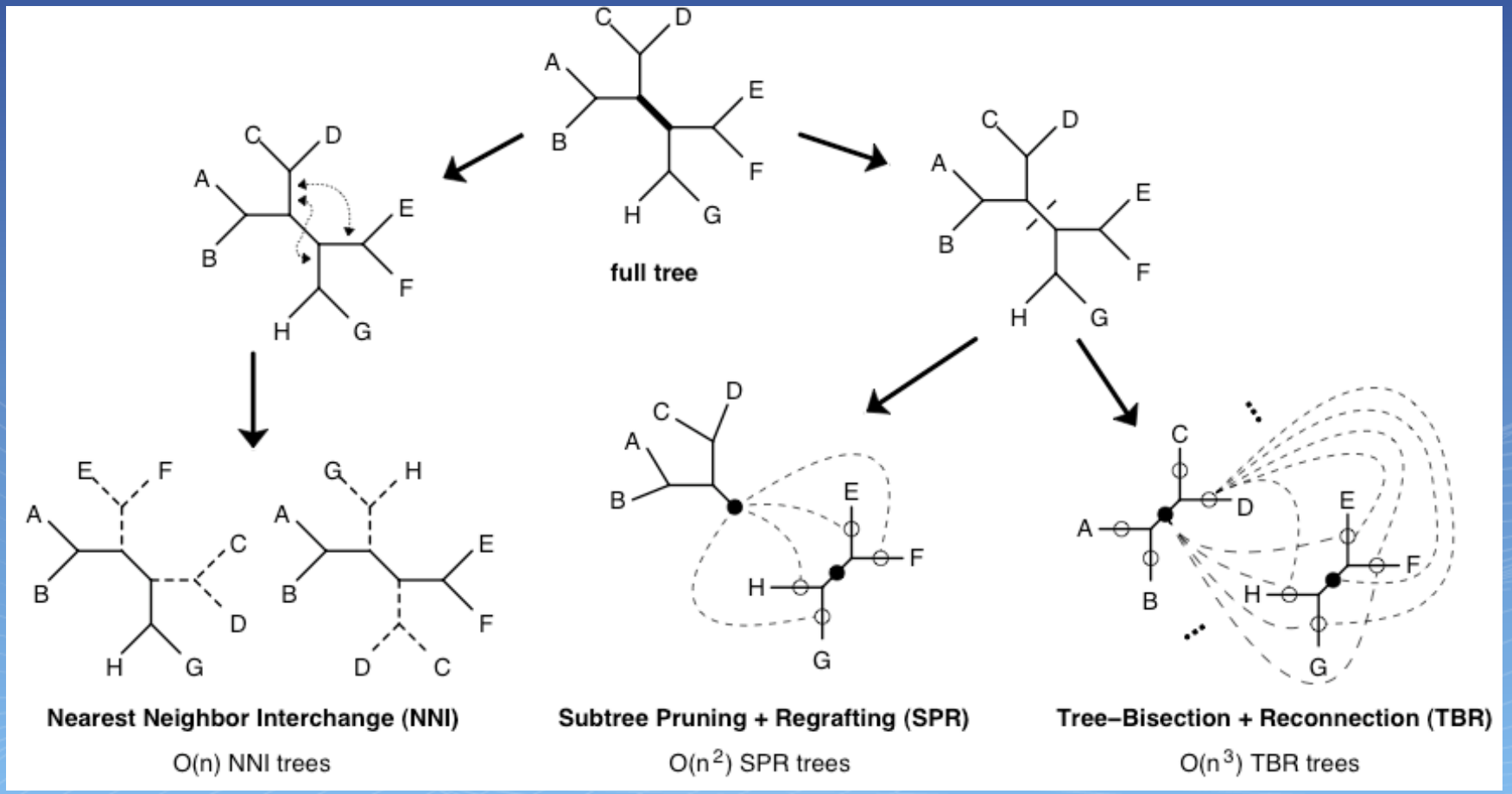
How can we deal with local maxima in the likelihood surface?



Tree rearrangements to escape local maxima.

CIBIV MFPL

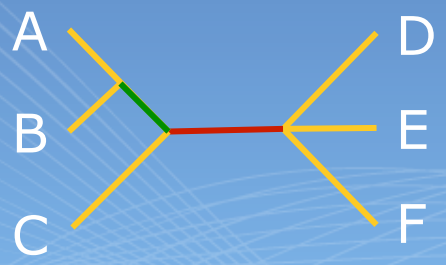
Finding the best tree: Tree rearrangements



CIBIV MFPL

Summarizing trees

Definition: A split Y|Z in the tree is a bipartition of the leaves/taxa into two subsets Y and Z induced by removing an edge from the tree.



AB|CDEF

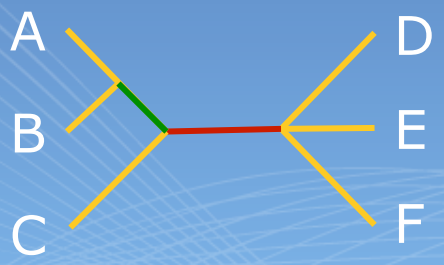
ABC|DEF

CIBIV MFPL

Summarizing trees

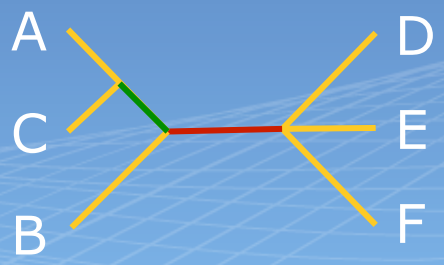
Definition: A split Y|Z in the tree is a bipartition of the leaves/taxa into two subsets Y and Z induced by removing an edge from the tree.

Definition: Two splits W|X and Y|Z are compatible, i.e. not contradictory, if at least one intersection of $W \cap Y$, $W \cap Z$, $X \cap Y$, $X \cap Z$ is empty.



AB|CDEF

ABC|DEF

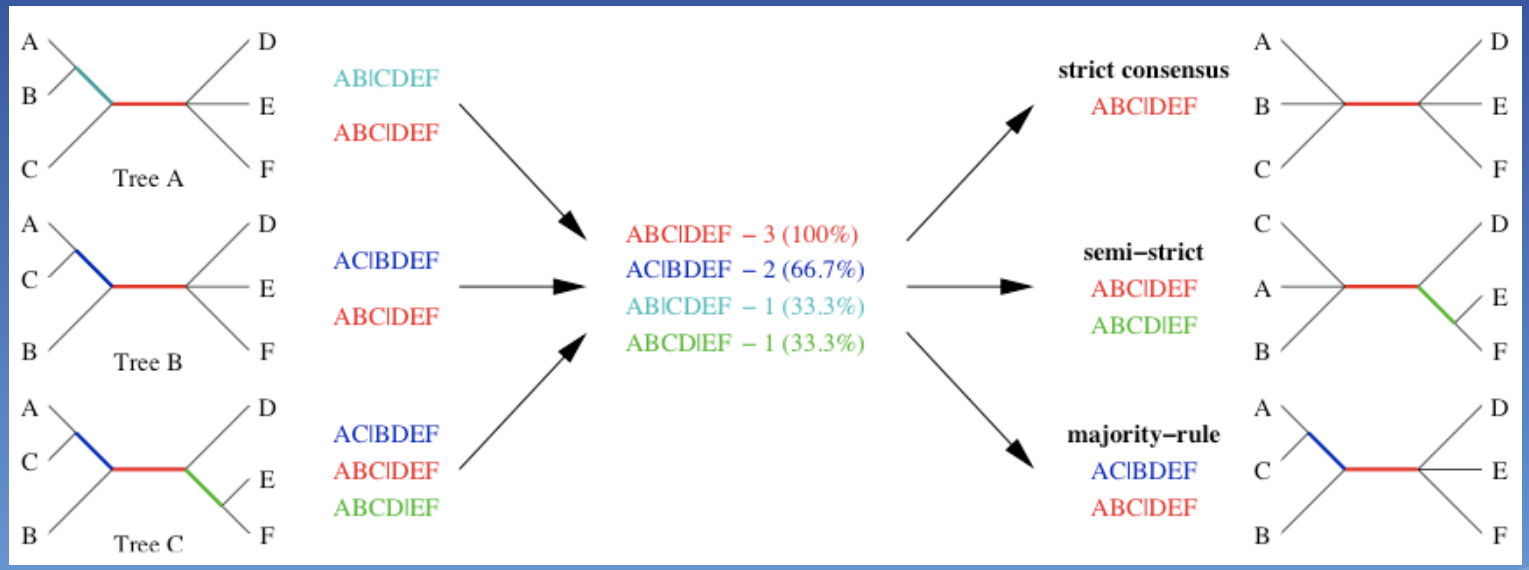


AC|BDEF

ABC|DEF

CIBIV MFPL

Summarizing trees



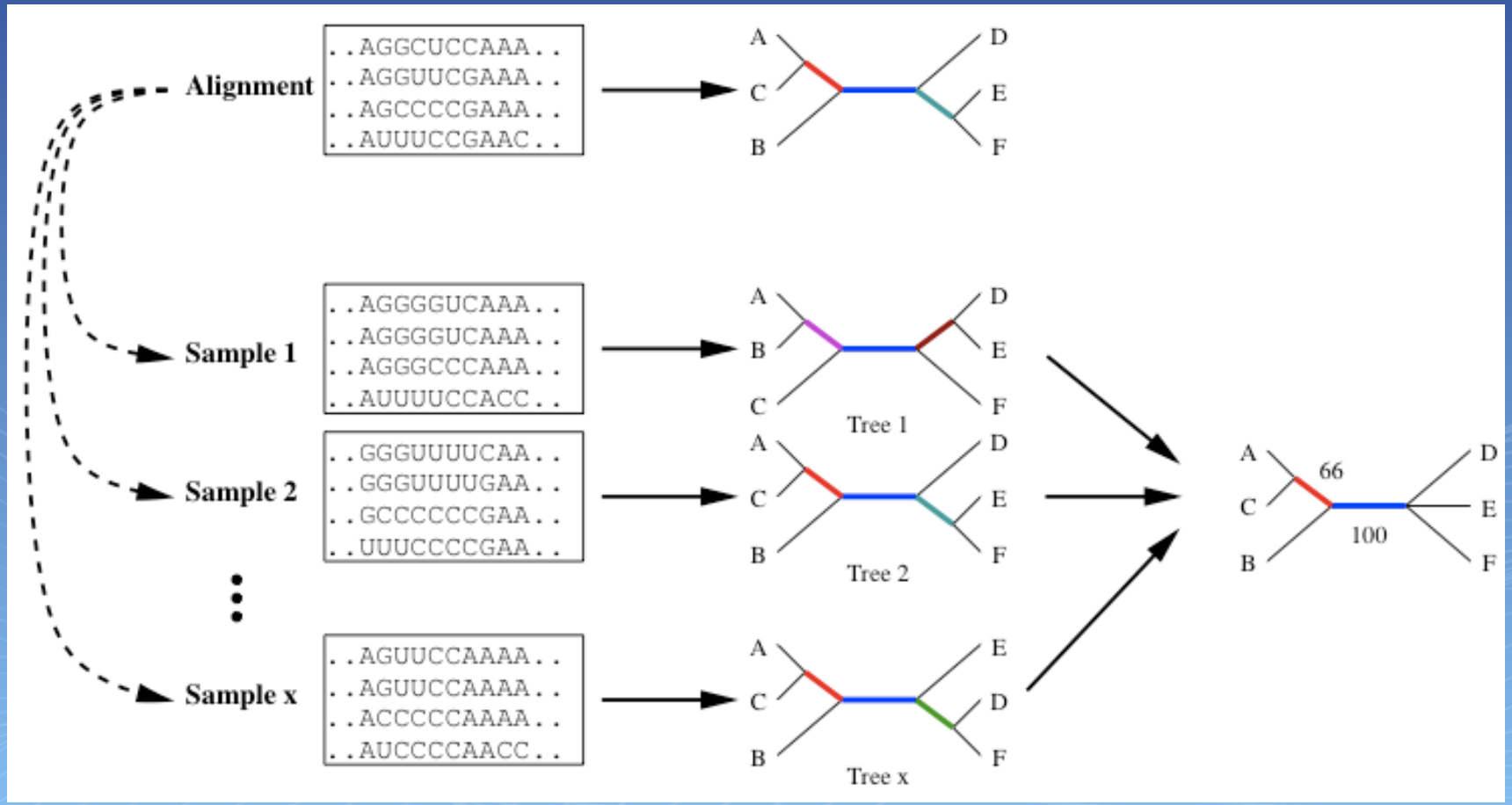
Strict consensus: contains only splits that occur in all input trees

Semi-strict consensus: contains only splits that are not contradicted by any tree

Majority-rule consensus (M_l): contains all splits that occur in more than *l* input trees, where typically *l* = *50%*.

CIBIV MFPL

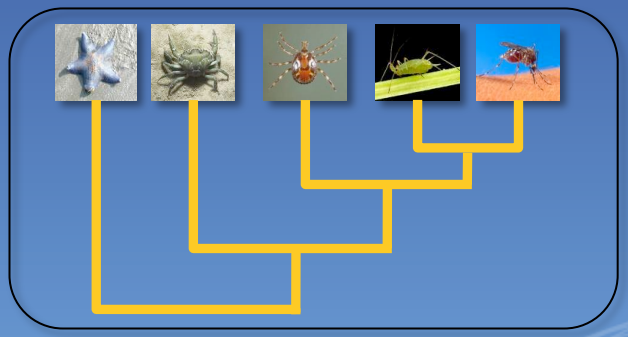
Assessing the confidence of trees: The (non-parametric) Bootstrap



CIBIV MFPL

What is the goal?

Biological Problem



Problem solution