Lecture 2: Understanding alignments and trees

Jadranka Rota and Niklas Wahlberg

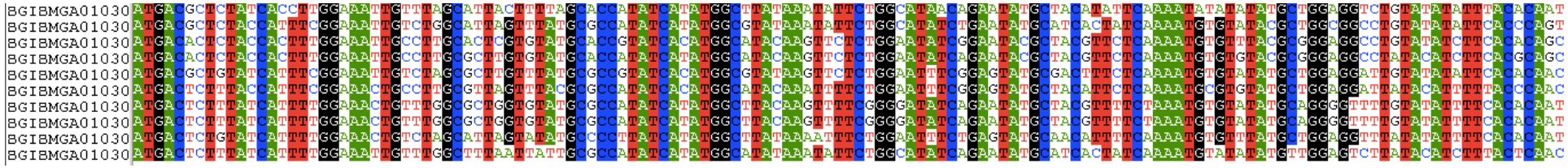
Systematic Biology Group

Department of Biology

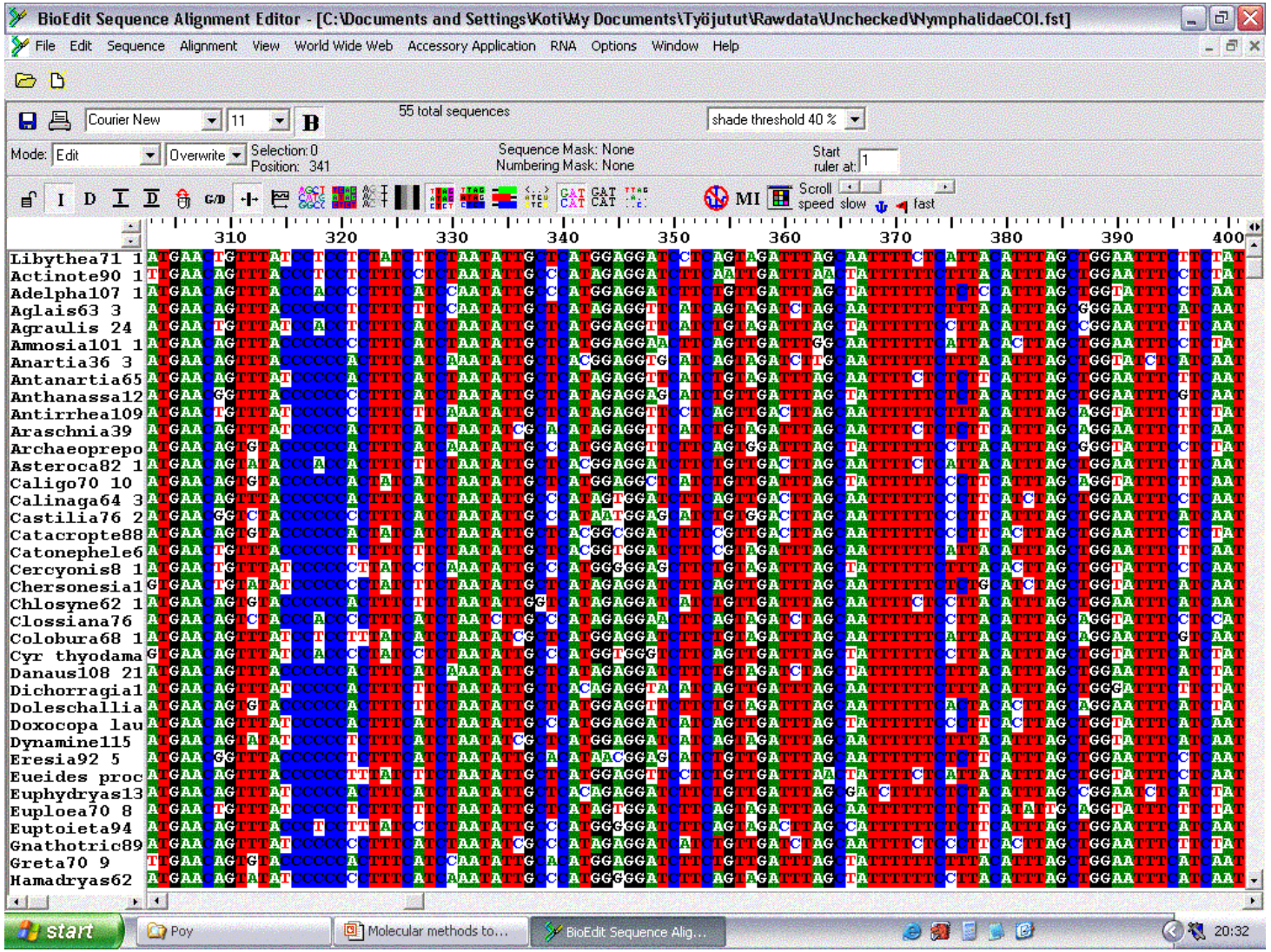
Lund University

LUND UNIVERSITY

Multiple Sequence Alignment



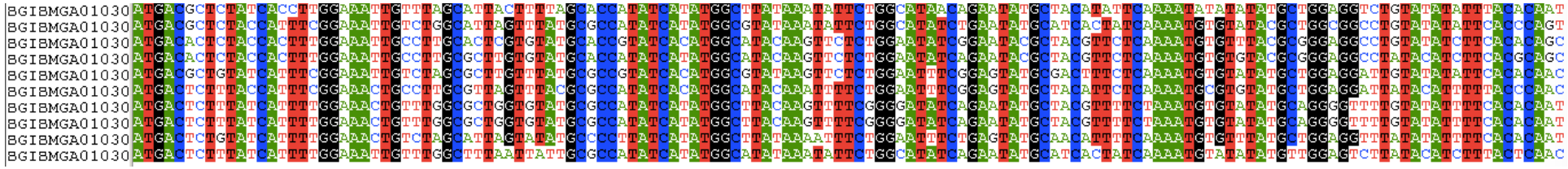
Alignment can be easy…



…or difficult

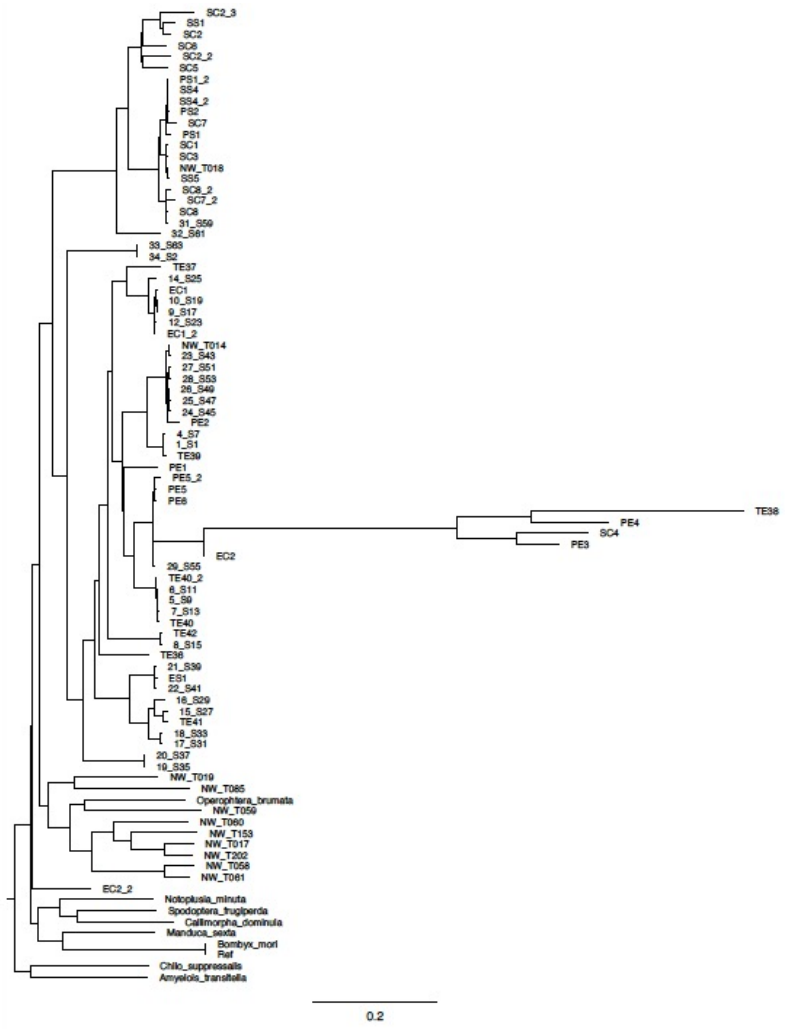
Homology: Definition

- **Homology: similarity that is the result of inheritance from a common ancestor - identification and analysis of homologies is central to phylogenetic systematics**
- **An alignment is a hypothesis of positional homology between bases/amino acids**



Multiple sequence alignment – goals

- **To generate a concise, information-rich summary of sequence data**
- **Alignments can be treated as models that can be used to test hypotheses**
- **Does this model of events accurately reflect known biological evidence?**



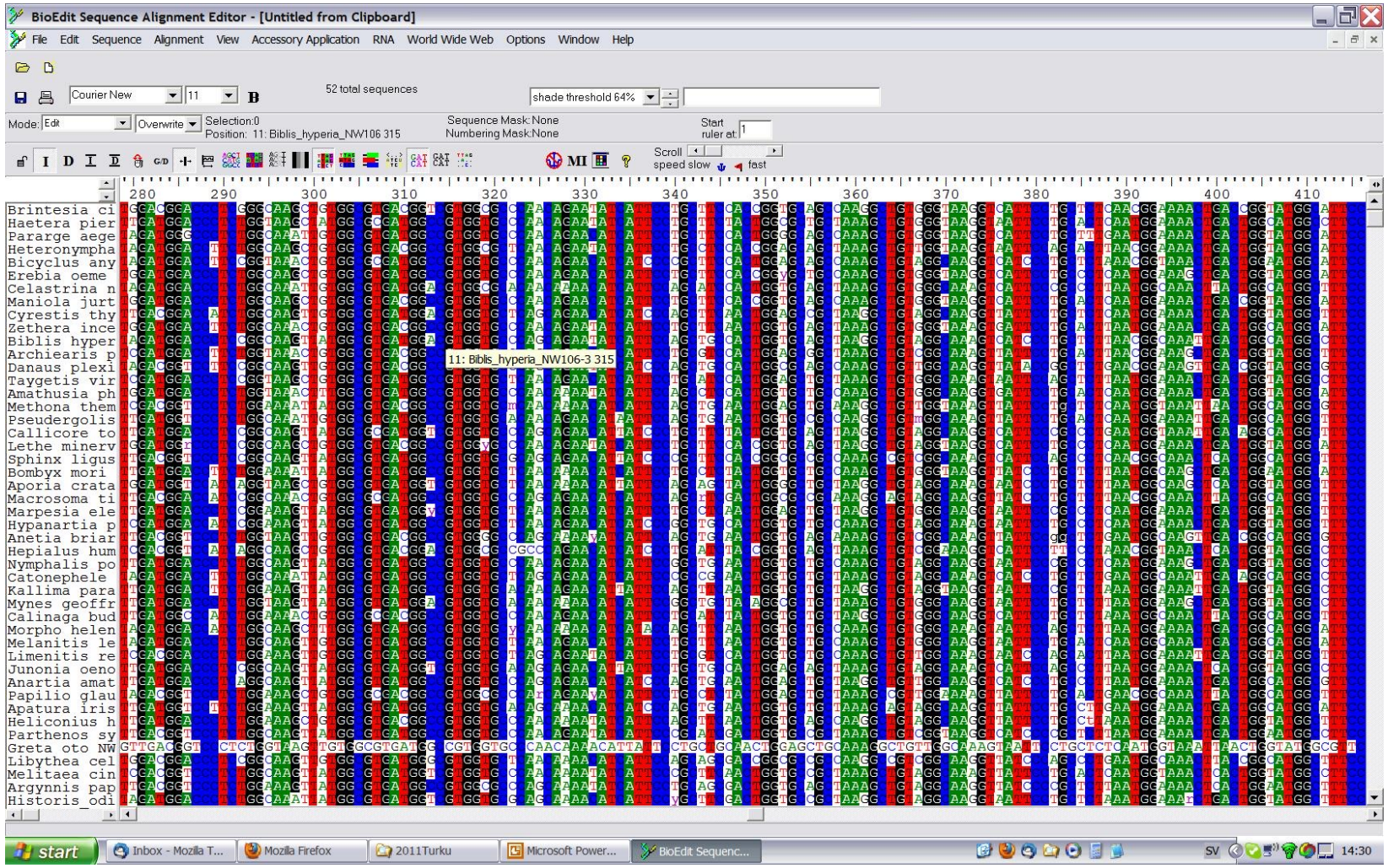
Multiple sequence alignment

- **Manual**
- **Dynamic programming**
- **Heuristic methods**
  - **Progressive alignment**
  - **Consistency-based scoring**
  - **Iterative refinement methods**

Manual alignment – reasons

- **Might be carried out because:**
  - **Alignment is easy**
  - **There is some extraneous information (structural)**
  - **Automated alignment methods have encountered a local minimum problem**
  - **An automated alignment method can be “improved”**

Protein-coding genes can often be manually aligned



How to align these sequences:

AGGGCTTTAA
AGGCTA
AATGGCTCTAA
GGAGCCCTAA

How to align these sequences:

```
A-GGGCTTTAA
A--GGCT--A-
AATGGCTCTAA
GGAG-CCCTAA
```

How to align these sequences:

```
-AGGGCTTTAA
-A-GGC--TA-
AATGGCTCTAA
-GGAGCCCTAA
```

Multiple sequence alignment

- **Is not easy! How to be objective?**
- **Dynamic programming**
  - **Works only for a very small number of sequences**
  - **Doesn’t use phylogeny and doesn’t incorporate an evolutionary model**
- **Heuristic methods**
  - **Progressive alignment**
    - **E.g. Clustal**
  - **Consistency-based scoring**
    - **Used in e.g. T-Coffee**
  - **Iterative refinement methods**
    - **Used in Mafft**

Heuristics: Progressive alignment

- **Devised by Feng and Doolittle in 1987**
- **A heuristic method and as such is not guaranteed to find the ‘optimal’ alignment**
- **Most successful implementation is Clustal**
  - **ClustalW**
  - **ClustalX**

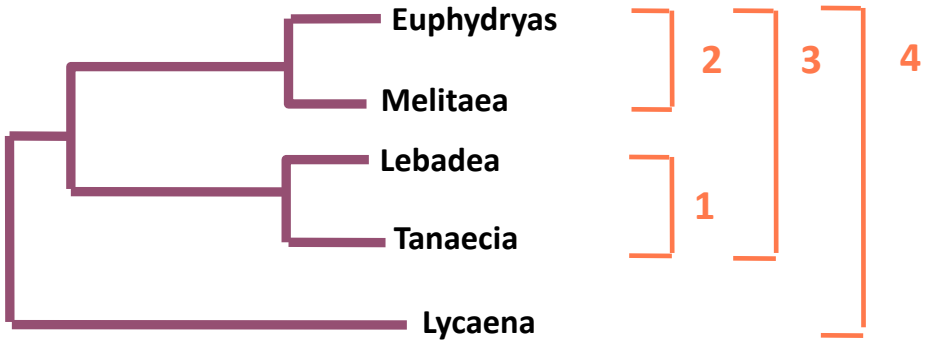
Clustal – pairwise alignments

- **First perform all possible pairwise alignments between each pair of sequences**
- **Calculate the 'distance' between each pair of sequences based on these isolated pairwise alignments**
- **Generate a distance matrix**

Taxon	*Euphydryas*	*Melitaea*	*Lebadea*	*Tanaecia*	*Lycaena*
Euphydryas	-				
Melitaea	0.17	-			
Lebadea	0.59	0.60	-		
Tanaecia	0.59	0.59	0.13	-	
Lycaena	0.77	0.77	0.75	0.75	-

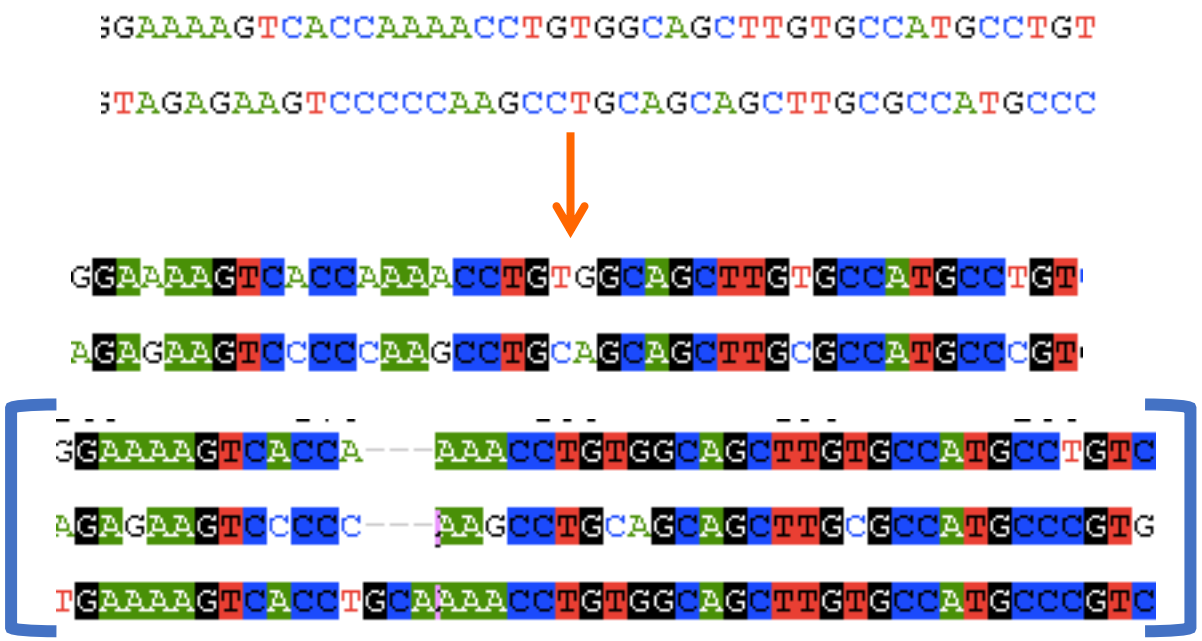
Clustal – guide tree

- **Generate a Neighbour-Joining 'guide tree' from these pairwise distances**
- **This guide tree gives the order in which the progressive alignment will be carried out**



Multiple alignment – first pair

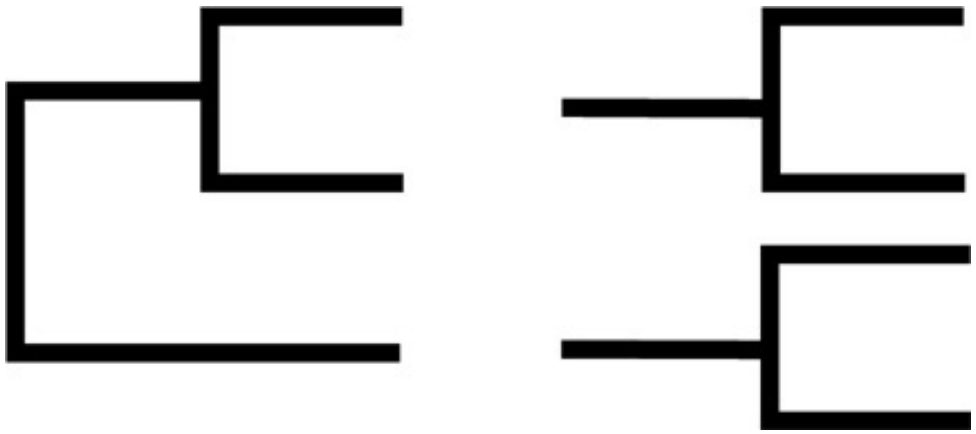
- **Align the two most closely-related sequences first**
- **This alignment is then 'fixed' and will never change**
- **If a gap is to be introduced subsequently, then it will be introduced in the same place in both sequences, but their relative alignment remains unchanged**



Clustal – decision time

- **Consult the guide tree to see what alignment is performed next**
  - **Align a third sequence to the first two**

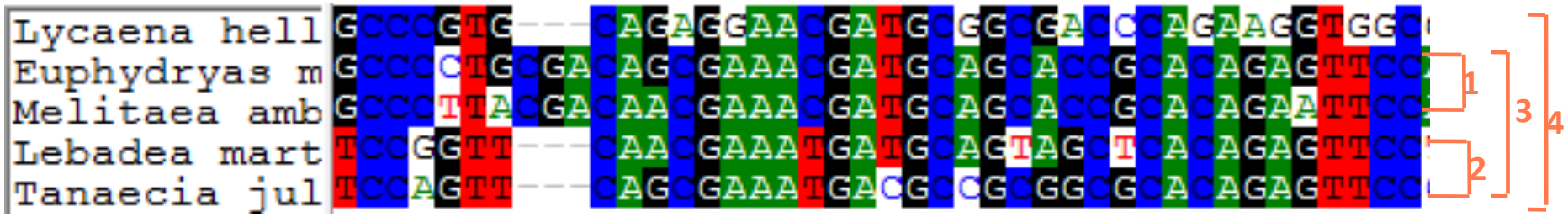
  or
  - **Align two entirely different sequences to each other**



Option 1 **Option 2**

Clustal – progression

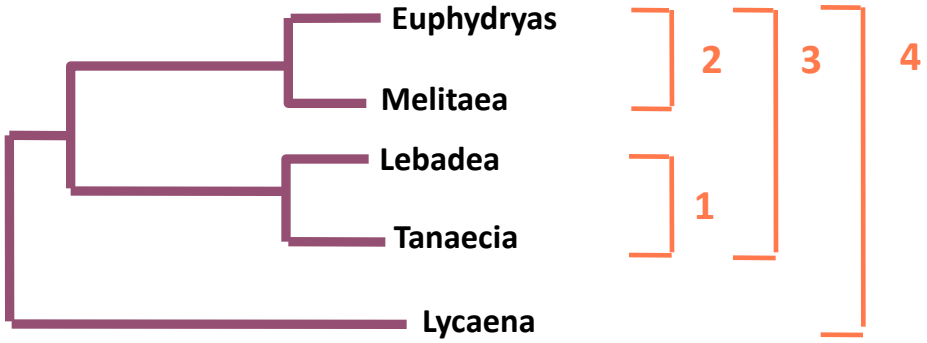
- **The alignment is progressively built up in this way**
- **Each step is treated as a pairwise alignment, sometimes with each member of a ‘pair’ having more than one sequence (e.g., pairs 1 and 2 become a new ‘pair’ in step 3)**



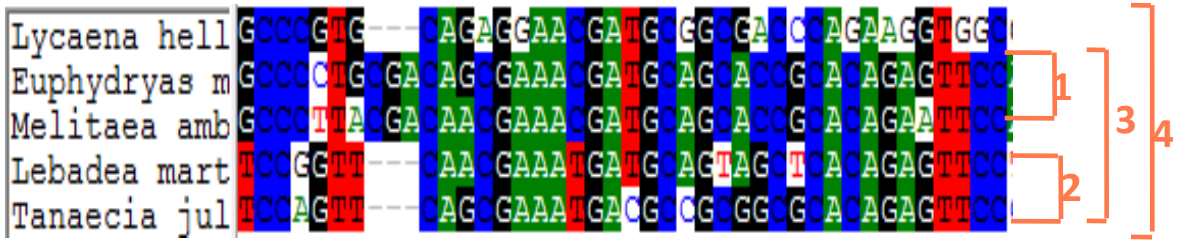
Overview of Clustal procedure

Taxon	*Euphydryas*	*Melitaea*	*Lebadea*	*Tanaecia*	*Lycaena*
Euphydryas	-				
Melitaea	0.17	-			
Lebadea	0.59	0.60	-		
Tanaecia	0.59	0.59	0.13	-	
Lycaena	0.77	0.77	0.75	0.75	-

Quick pairwise alignment: calculate distance matrix



Neighbour-joining tree (guide tree)



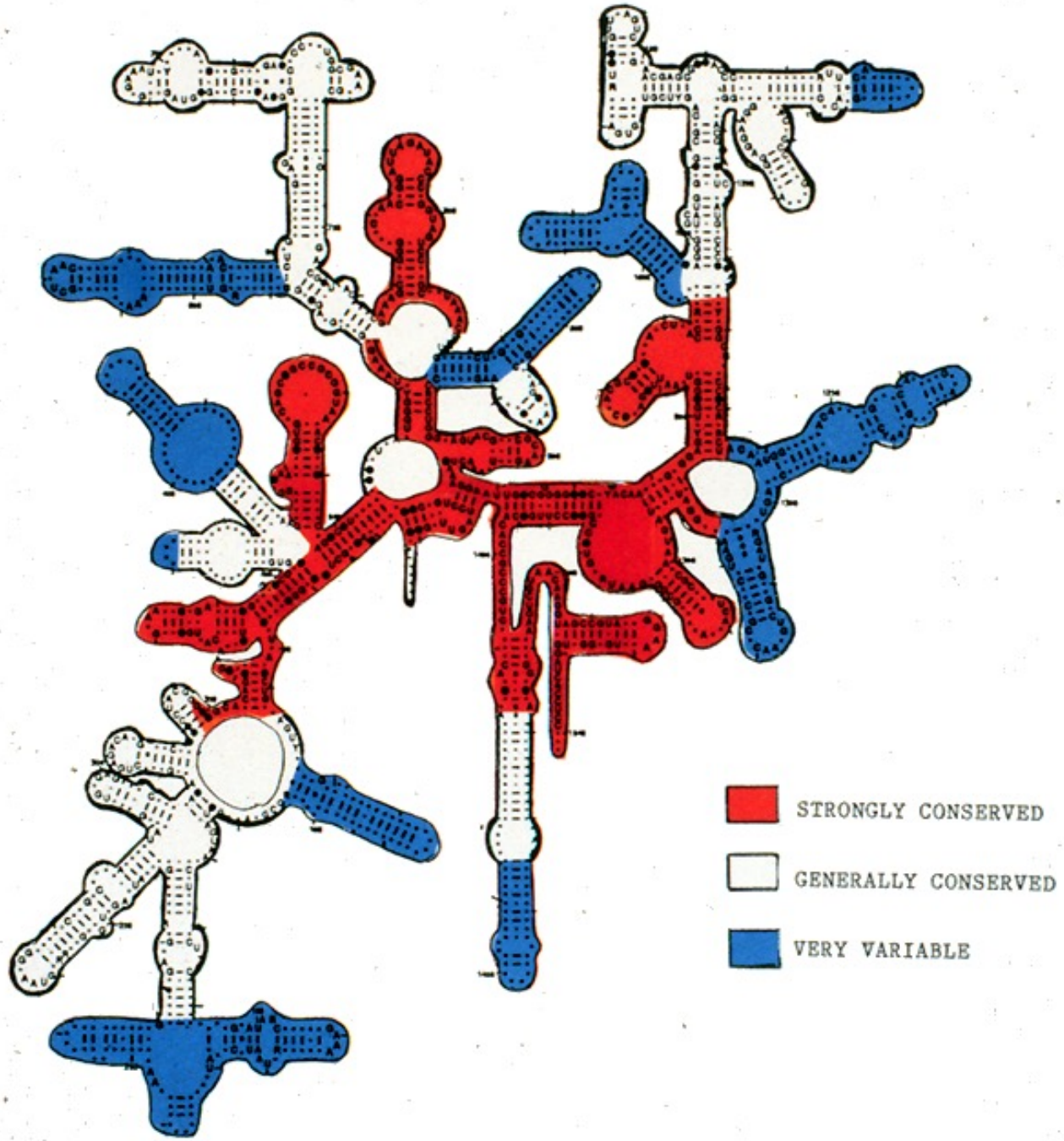
Progressive alignment following guide tree

Clustal – good points/bad points

- **Advantages:**
  - **Speed**
- **Disadvantages:**
  - **Hierarchic structure introduced that is not necessarily phylogenetic**
  - **No way of quantifying whether or not the alignment is good**
  - **No way of knowing if the alignment is 'correct'**
  - **Local minimum problem. If an error is introduced early in the alignment process, it is impossible to correct this later in the procedure**
  - **Arbitrary alignment**

Increasing the sophistication of the alignment process

- **Should we treat all the sequences in the same way?**
  - **some sequences are closely related and some sequences are distant relatives.**
- **Should we treat all positions in the sequences as though they were the same?**
  - **they might have different functions and different locations in the 3-dimensional structure**
  - **codon structure – how to retain this?**


STRONGLY CONSERVED
GENERALLY CONSERVED
VERY VARIABLE

Iterative refinement methods

- **Initial alignments split into two groups randomly**
- **Within groups the alignment is kept fixed**
- **Dynamic programming used to align the two groups to each other**
- **This is repeated until score converges**
- **Used in e.g. Muscle and MAFFT**

Using models in alignment

- **New methods are being developed all the time**
- **Latest methods include using a Bayesian statistic framework, DNA evolutionary models and alignment concomitantly with estimation of phylogentic relationships**
- **Still not feasible with a moderately sized dataset**

Bottom line

- **Alignments are extremely important in phylogenetics**
- **A bad alignment means many wrong statements of homology, which means pure rubbish as output**
- **A good alignment can be hard to attain**

The Tree

Finding the optimal trees

Numbers of possible trees for N taxa

1	1
2	1
3	1
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025
11	34459425
12	654729075
13	13749310575
14	316234143225
15	[illegible]
[illegible]	[illegible]1875 (2×10^{20})
50	3×10^{74}

How can we find the most optimal tree?

Astronomers estimate that the universe could contain **up to one septillion stars** – that's a one followed by 24 zeros. Our Milky Way alone contains more than 100 billion, including our most well-studied star, the Sun.

NASA

Finding optimal trees – exact solutions

- **Exact solutions can only be used for small numbers of taxa**
- **Exhaustive search examines all possible trees**
- **Branch and bound does not examine all trees, but will find optimal tree(s)**
- **Typically used for problems with 10–20 taxa**

Finding optimal trees – heuristics

- **The number of possible trees increases faster than exponentially with the number of taxa making exhaustive searches impractical for many data sets (an NP-complete problem)**
- **Heuristic methods are used to search tree space for optimal trees by building or selecting an initial tree and swapping branches to search for better ones**
- **The trees found are not guaranteed to be optimal – they are best guesses**

NP-complete problem: any of a class of computational problems for which no efficient solution algorithm has been found.
https://www.britannica.com/science/NP-complete-problem

Finding optimal trees – heuristics

- **Stepwise addition**

  Asis - the order in the data matrix

  Closest - starts with shortest 3-taxon tree, adds taxa in order that produces the least increase in tree length (greedy heuristic)

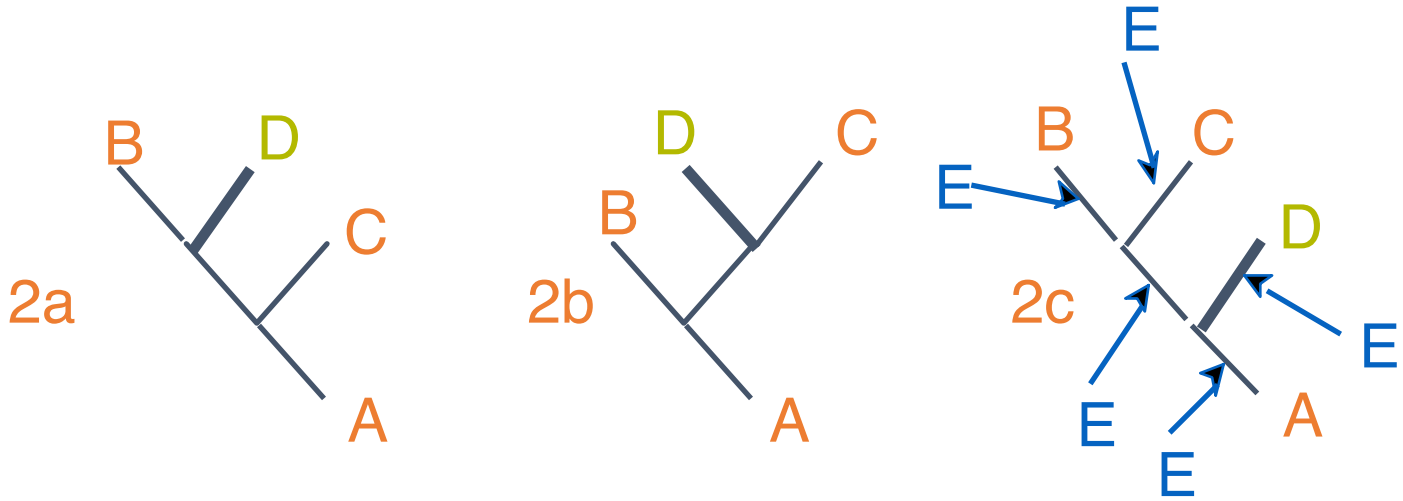
  Simple - the first taxon in the matrix is taken as a reference - taxa are added to it in the order of their decreasing similarity to the reference

  Random - taxa are added in a random sequence, many different sequences can be used

Finding optimal trees – exhaustive search

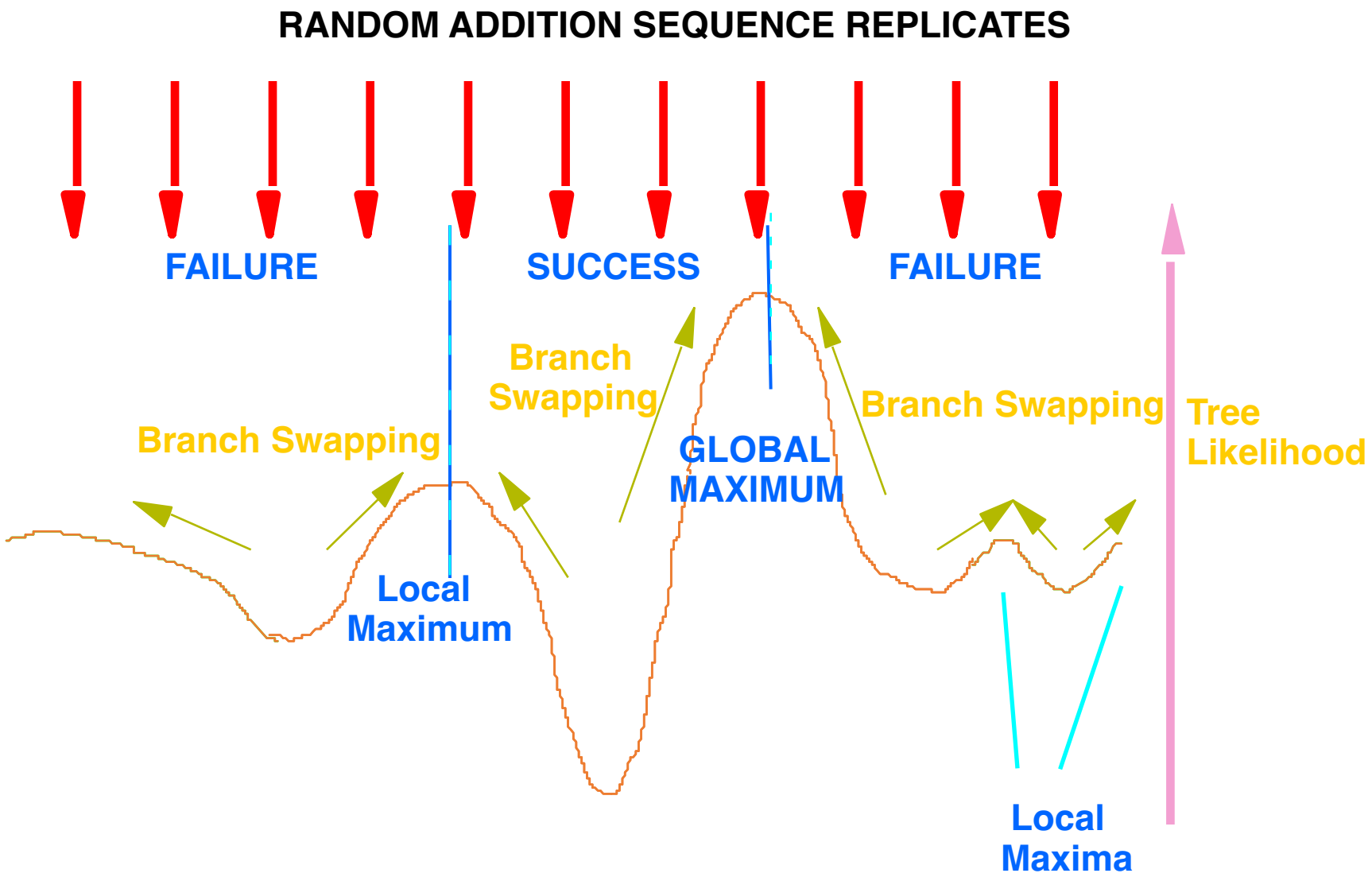
Starting tree, any 3 taxa

Add fourth taxon (D) in each of three possible positions -> three trees



Add fifth taxon (E) in each of the five possible positions on each of the three trees -> 15 trees, and so on ....

Tree space may be populated by local optima and islands of optimal trees

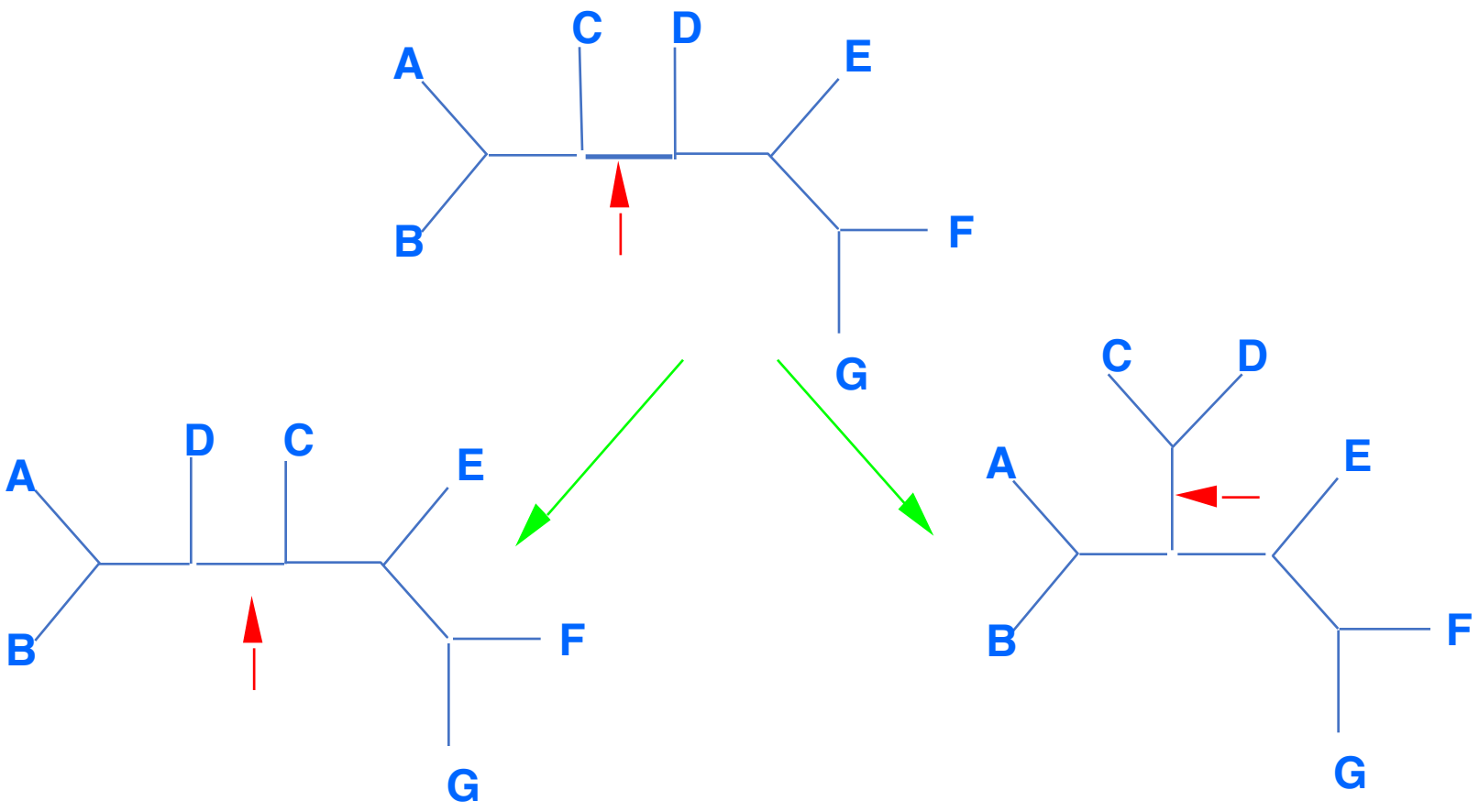


Finding optimal trees – branch swapping

- **Nearest neighbor interchange (NNI)**
- **Subtree pruning and regrafting (SPR)**
- **Tree bisection and reconnection (TBR)**

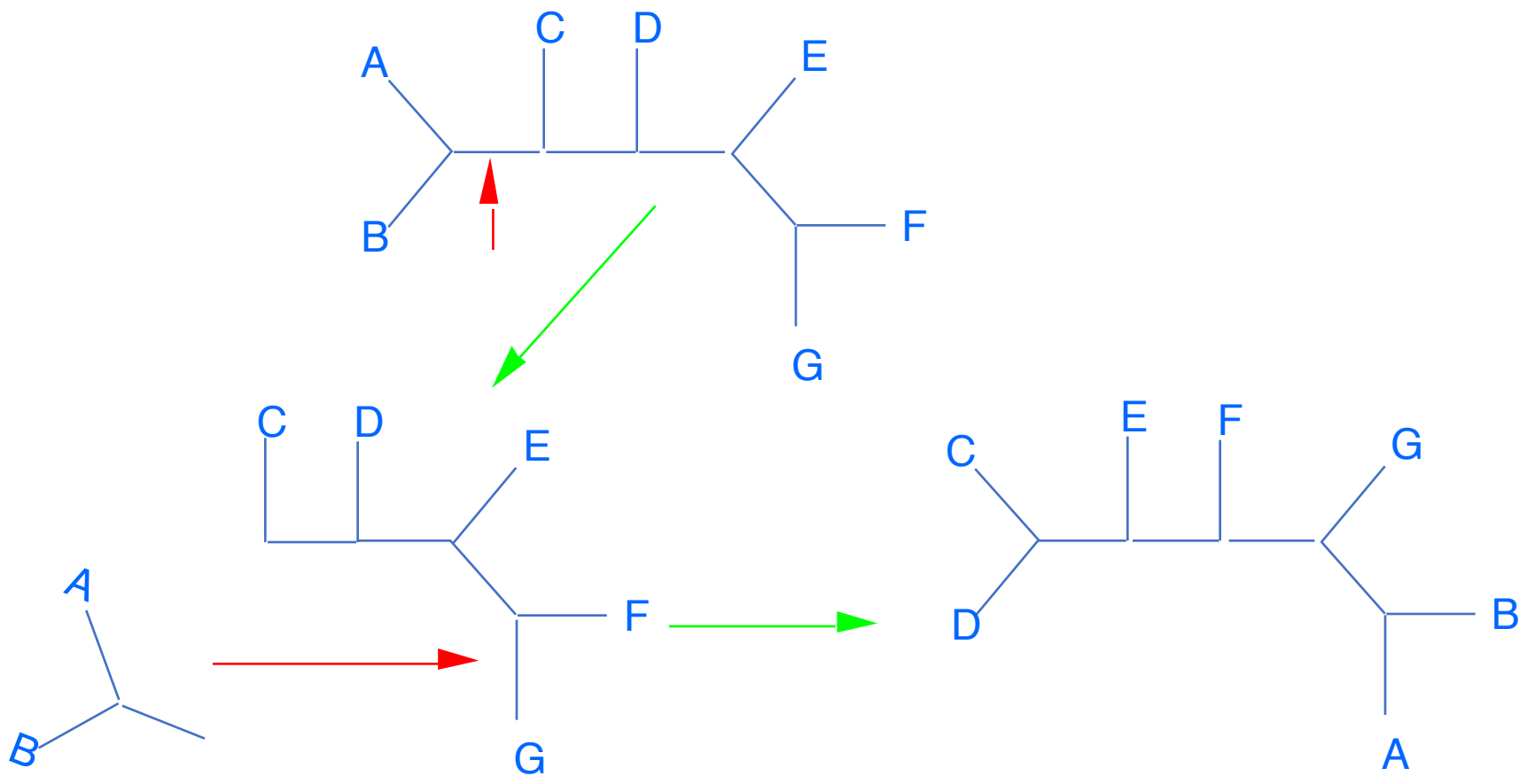
Finding optimal trees – heuristics

Nearest neighbor interchange (NNI)



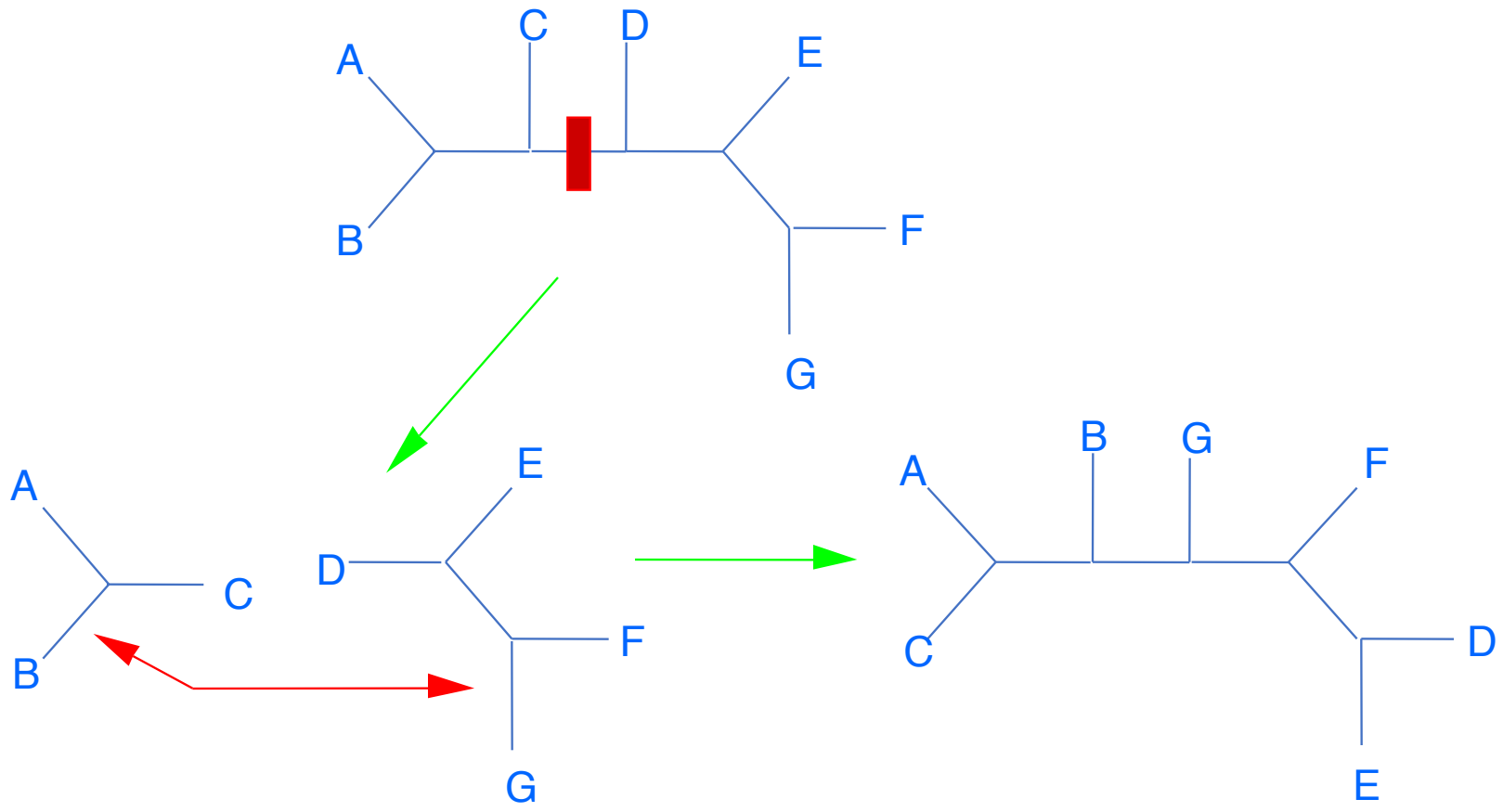
Finding optimal trees – heuristics

Subtree pruning and regrafting (SPR)



Finding optimal trees – heuristics

Tree bisection and reconnection (TBR)



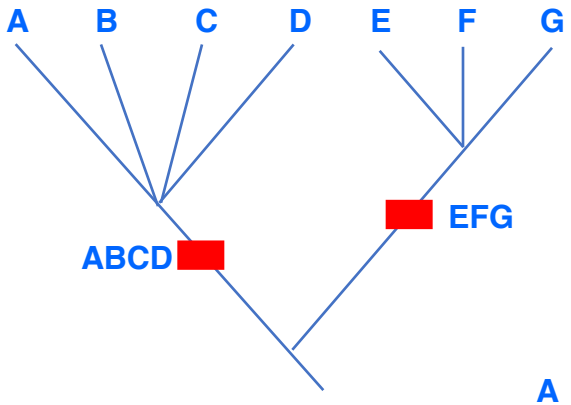
Searching with topological constraints

- **Topological constraints are user-defined phylogenetic hypotheses**
- **Can be used to find optimal trees that either:**

1. include a specified clade or set of relationships

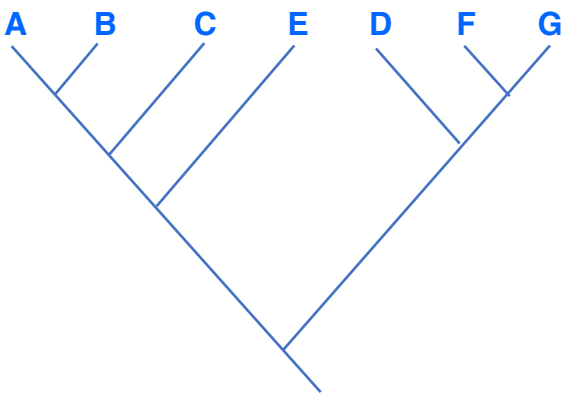
2. exclude a specified clade or set of relationships (reverse constraint)

Searching with topological constraints

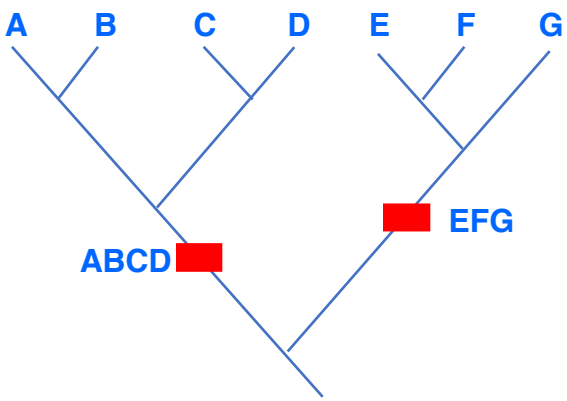


CONSTRAINT TREE

((A,B,C,D)(E,F,G))



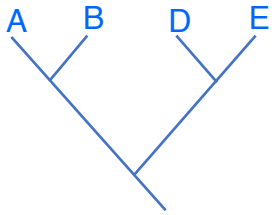
Incompatible with constraint tree



Compatible with constraint tree

Searching with topological constraints
backbone constraints

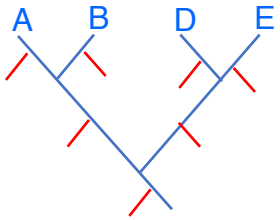
- **Backbone constraints specify relationships among a subset of the taxa**
- **Taxa A, B, C, D, E**



BACKBONE CONSTRAINT

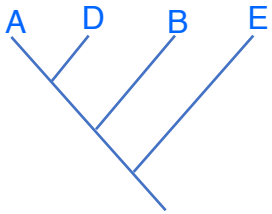
((A,B)(D,E))

relationships of taxon C are not specified



possible positions of taxon C

Compatible with backbone constraint



Incompatible with backbone constraint

Consensus methods

Multiple optimal trees

- **Many methods can yield multiple equally optimal trees**
- **We can further select among these trees with additional criteria**
- **Typically, relationships common to all the optimal trees are summarised with *consensus trees***

Consensus methods

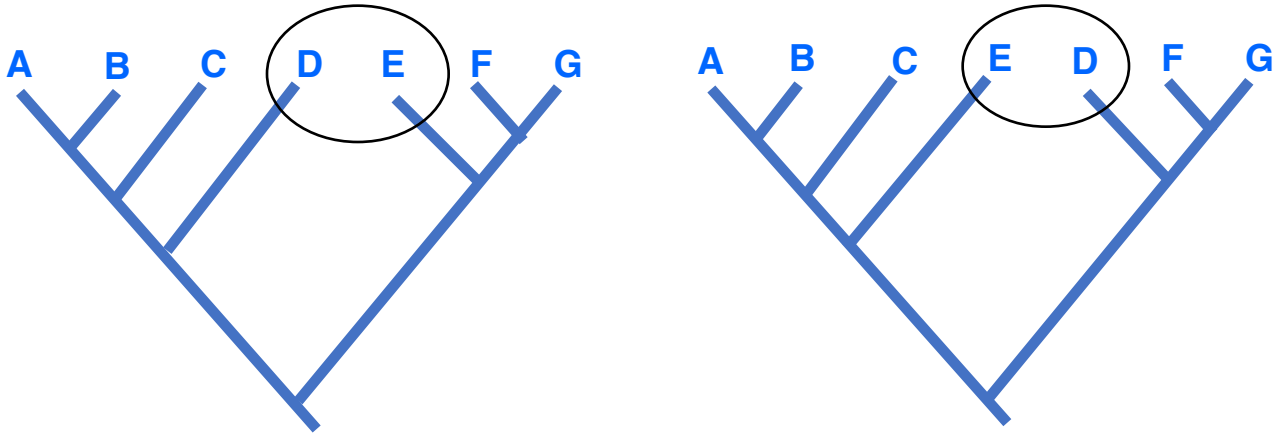
- **A consensus tree is a summary of the agreement among a set of fundamental trees**
- **There are many consensus methods that differ in:**
  1. **the kind of agreement**
  2. **the level of agreement**
- **Consensus methods can be used with multiple trees from a single analysis or from multiple analyses**

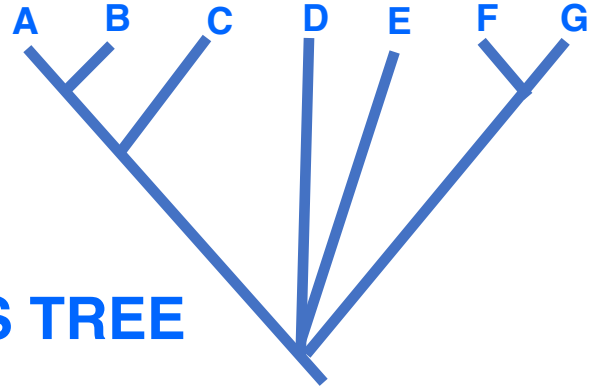
Strict consensus methods

- **Strict consensus methods require agreement across all the fundamental trees**
- **They show only those relationships that are unambiguously supported by the parsimonious interpretation of the data**
- **This method produces a consensus tree that includes all and only those full splits found in all the fundamental trees**
- **Other relationships (those in which the fundamental trees disagree) are shown as unresolved polytomies**

Strict consensus methods

TWO FUNDAMENTAL TREES





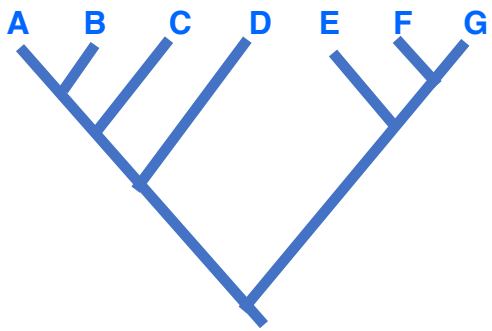
STRICT CONSENSUS TREE

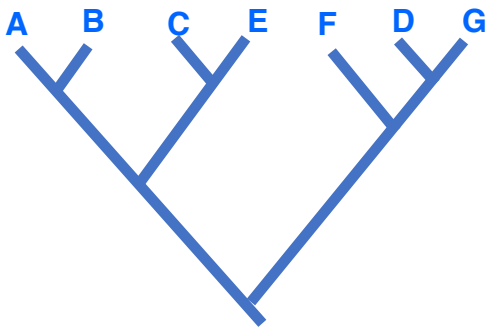
Majority-rule consensus methods

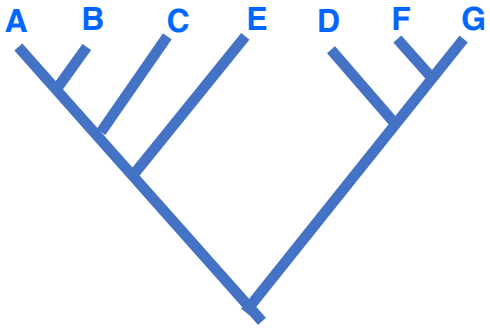
- **Majority-rule consensus methods require agreement across a majority of the fundamental trees**
- **May include relationships that are not supported by the most parsimonious interpretation of the data**
- **This method produces a consensus tree that includes all and only those full splits found in a majority (>50%) of the fundamental trees**
- **Other relationships are shown as unresolved polytomies**
- **Of particular use in bootstrapping**

Majority rule consensus

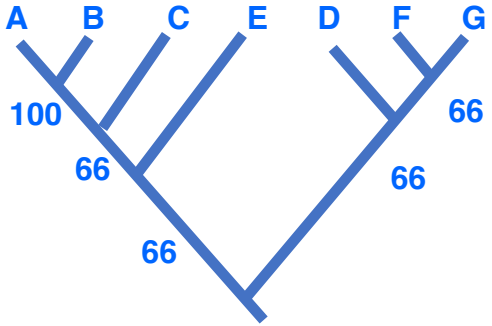
THREE FUNDAMENTAL TREES







Numbers indicate frequency of clades in the fundamental trees



MAJORITY-RULE CONSENSUS TREE

Consensus methods – use

- **Currently majority-rule methods mainly used**
  - **bootstrapping**
  - **Bayesian methods**
- **Reduced methods can be useful to identify problem taxa**
  - **E.g. RogueNaRok**
- **Strict methods mainly used in parsimony analyses**
  - **rarely used with molecular data**

Take home messages

- **Statements of homology are the basis of phylogenetics**
- **Alignments of molecular sequences are very strong statements of positional homology**
- **Finding an optimal tree is not a trivial task**

The Data

File formats

Computer programs

- **Multitude of programs available for free!**
- **Most have their own input format**
- **Many are ”black box” programs**
- **Input files are always simple text files!!!**

No good online resource available

http://evolution.gs.washington.edu/phylip/software.html

was an attempt but not updated for a long time

Computer programs – ML

- **IQ-TREE (recommended)**
- **RAxML (recommended)**
- **PHYML**
- **GARLI**

Computer programs – Bayesian inference

- **MrBayes (recommended)**
- **BEAST (recommended)**
- **BAMBE**
- **BayesPhylogenies**
- **ExaBayes**

Viewing trees

- **FigTree (recommended)**
- **TreeView**

Three most common data formats

- **FASTA**
- **Phylip**
- **Nexus**

Input format – FASTA

```
>Papilio_glaucus_69_3
GAGaTGGAaGACAaGGTTTCGTCGACCCTGTCCGGCCTCGAGGGCGAACT
>Hamearis84_13
GGaATGGAaGAGAAaGTCTCCACAACCCTCTCCGGACTCGAAGGTGAGCT
>Danaus_plexippus108_21
GAGAtGGAGGAGAaGGTCTCCTCCACCCTCTCAGGTCTCGAAGGTGAACT
>Greta_oto70_9
GGAATGGAaGAGAaGGTCTCCTCGACCCTCTCAGGCCTTGAAGGTGAACT
>Amathusia_phidippus114_17
GGaATGGAaGACAAaGTCTCCTCAaCCCTCTCCGGTCTTGAGGGTGAACT
>Morpho_peleides66_5
GGaATGGAGAGAAAaGTCTCTACTACCCTGTCTGGCCTCGAAGGCGAACT
>BrintesiaB01
GGAATGGAaGACAAaGTCTCGTCCACCCTCTCCGGGCTGGAAGGCGAGCT
>Elymnias_casiphone121_20
GAGAwGGaAGAcaAAGTATCCTCCACCCTCTCTGGTCTTGAAGCTGAACT
>Erebia_oemeEW24_7
gGaATGGAaGACAAaGTCTCCTCGACTCTCTCTGGCCTCGAAGGCGAGCT
```

Input format – PHYLIP

```
9 50
Papilio_gl GAGaTGGAaGACAaGGTTTCGTCGACCCTGTCCGGCCTCGAGGGCGAACT
Hamearis84 GGaATGGAaGAGAAaGTCTCCACAACCCTCTCCGGACTCGAAGGTGAGCT
Danaus_ple GAGAtGGAGGAGAaGGTCTCCTCCACCCTCTCAGGTCTCGAAGGTGAACT
Greta_oto7 GGAATGGAaGAGAaGGTCTCCTCGACCCTCTCAGGCCTTGAAGGTGAACT
Amathusia_ GGaATGGAaGACAAaGTCTCCTCAaCCCTCTCCGGTCTTGAGGGTGAACT
Morpho_pel GGaATGGAGAGAAAaGTCTCTACTACCCTGTCTGGCCTCGAAGGCGAACT
BrintesiaB GGAATGGAaGACAAaGTCTCGTCCACCCTCTCCGGGCTGGAAGGCGAGCT
Elymnias_c GAGAwGGaAGAcaAAGTATCCTCCACCCTCTCTGGTCTTGAAGCTGAACT
Erebia_oem gGaATGGAaGACAAaGTCTCCTCGACTCTCTCTGGCCTCGAAGGCGAGCT
```

Input format – NEXUS

- Nexus blocks: data block, tree block, character set block, MrBayes block, …

```
#NEXUS
BEGIN DATA;
   DIMENSIONS  NTAX=9 NCHAR=50;
   FORMAT DATATYPE=DNA MISSING=? GAP=- INTERLEAVE=No;
   Matrix

[ArgKin 596]
Papilio_glaucus_69_3      GAGaTGGAaGACAaGGTTTCGTCGACCCTGTCCGGCCTCGAGGGCGAACT
Hamearis84_13             GGaATGGAaGAGAAaGTCTCCACAACCCTCTCCGGACTCGAAGGTGAGCT
Danaus_plexippus108_21    GAGAtGGAGGAGAaGGTCTCCTCCACCCTCTCAGGTCTCGAAGGTGAACT
Greta_oto70_9             GGAATGGAaGAGAaGGTCTCCTCGACCCTCTCAGGCCTTGAAGGTGAACT
Amathusia_phidippus114_17 GGaATGGAaGACAAaGTCTCCTCAaCCCTCTCCGGTCTTGAGGGTGAACT
Morpho_peleides66_5       GGaATGGAGAGAAAaGTCTCTACTACCCTGTCTGGCCTCGAAGGCGAACT
BrintesiaB01              GGAATGGAaGACAAaGTCTCGTCCACCCTCTCCGGGCTGGAAGGCGAGCT
Elymnias_casiphone121_20  GAGAwGGaAGAcaAAGTATCCTCCACCCTCTCTGGTCTTGAAGCTGAACT
Erebia_oemeEW24_7         gGaATGGAaGACAAaGTCTCCTCGACTCTCTCTGGCCTCGAAGGCGAGCT
;
end;
```

A good overview available at https://plewis.github.io/nexus/

Input format – NEXUS interleaved

```
#NEXUS
BEGIN DATA;
   DIMENSIONS  NTAX=9 NCHAR=121;
   FORMAT DATATYPE=DNA MISSING=? GAP=- INTERLEAVE=Yes;
   Matrix

[ArgKin 50 bp]
Papilio_glaucus_69_3      GAGaTGGAaGACAaGGTTTCGTCGACCCTGTCCGGCCTCGAGGGCGAACT
Hamearis84_13             GGaATGGAaGAGAAaGTCTCCACAACCCTCTCCGGACTCGAAGGTGAGCT
Danaus_plexippus108_21    GAGAtGGAGGAGAaGGTCTCCTCCACCCTCTCAGGTCTCGAAGGTGAACT
Greta_oto70_9             GGAATGGAaGAGAaGGTCTCCTCGACCCTCTCAGGCCTTGAAGGTGAACT
Amathusia_phidippus114_17 GGaATGGAaGACAAaGTCTCCTCAaCCCTCTCCGGTCTTGAGGGTGAACT
Morpho_peleides66_5       GGaATGGAGAGAAAaGTCTCTACTACCCTGTCTGGCCTCGAAGGCGAACT
BrintesiaB01              GGAATGGAaGACAAaGTCTCGTCCACCCTCTCCGGGCTGGAAGGCGAGCT
Elymnias_casiphone121_20  GAGAwGGaAGAcaAAGTATCCTCCACCCTCTCTGGTCTTGAAGCTGAACT
Erebia_oemeEW24_7         gGaATGGAaGACAAaGTCTCCTCGACTCTCTCTGGCCTCGAAGGCGAGCT

[COI 71 bp]
Papilio_glaucus_69_3      taAagAtaTTgGaACATTATACTTTATTTTTGGAATTTGAGCAAGAATATTAGGAACTTCTTTAAGTTTAT
Hamearis84_13             ???????????????????????????????????????TGAGCAGGAATAGTAGGAACATCATTAAGATTAC
Libythea_celtis71_1       ???????????????????????????????????????TGAGCAGGAATAGTAGGAACTTCATTAAGTCTAT
Danaus_plexippus108_21    ???????????????????????????????????????TGAGCAGGAATAGTTGGGACATCTTTAAGTCTTT
Greta_oto70_9             ???????????????????????????????????????TGAGCAGGAATAGTAGGAACATCTTTAAGTTTAT
Amathusia_phidippus114_17 ???????????????????????????????????????TGATCTGGAATAGTAGGAACATCCCTCAGTCTTA
Morpho_peleides66_5       ???????????????????????????????????????TGAGCCGGTATAATTGGTACATCCCTAAGTCTTA
BrintesiaB01              ???????????????????????????????????????TGAGCAGGTATAGTAGGAACATCTCTTAGTTTAA
Elymnias_casiphone121_20  ???????????????????????????????????????TGATCAGGAATAGTAGGAACTTCCCTCAGTCTTA
Erebia_oemeEW24_7         ???????????????????????????????????????TGAGCAGGTATAGTAGGTACTTCCCTTAGTCTTA
;
end;
```

Github page for the course

- **https://github.com/NymphalidNiklas/IBIO_2024/**