Aligning Circularly Ordered Lists

Why would anyone want to do that?

Peter F. STADLER

Bioinformatics, Department of Computer Science

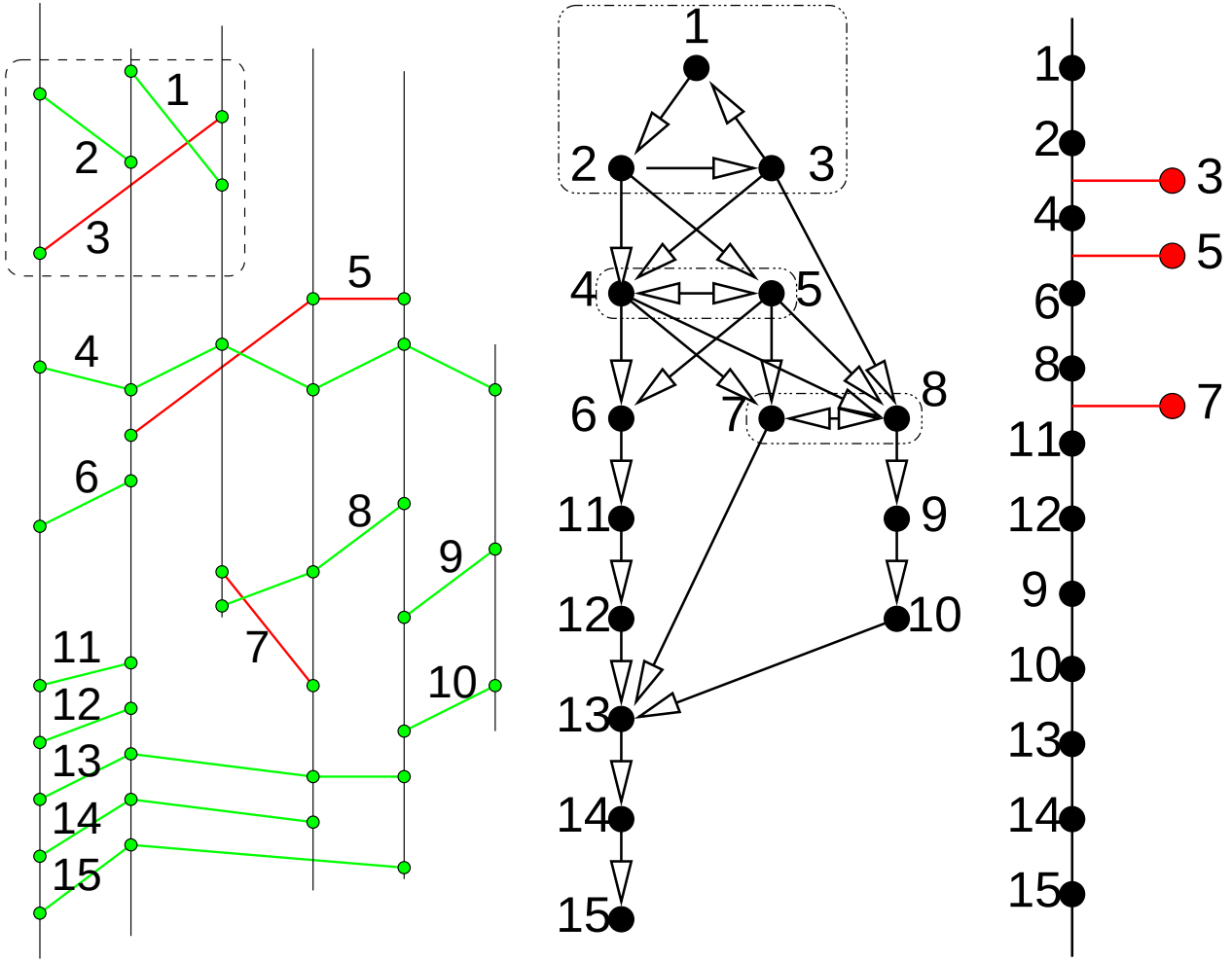
University of Leipzig, Germany

External Faculty:
Institute f. Theoretical Chemistry & Structural Biology, University of Vienna, Austria
Santa Fe Institute

http://www.bioinf.uni-leipzig.de/~studla

Bled, Villa Plemelj, Feb. 23-28, 2004

The Footprint Sorting Problem

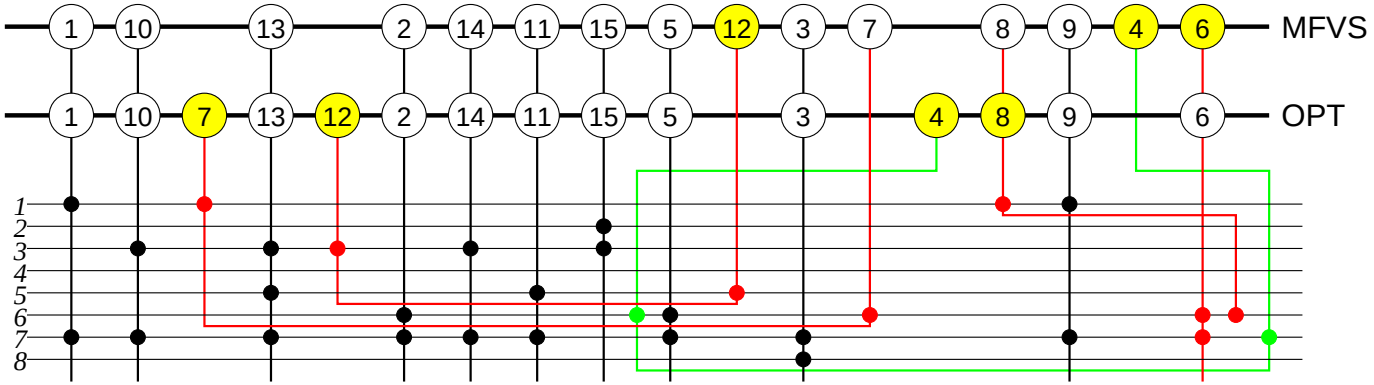


Linearly ordered list of objects (footprints in this case).

No unique solution to sorting problem
$\Longrightarrow$ different heuristic approximations.

Comparison of linearly ordered lists π' and π'':
Insertion, deletion, exact match as only edit operations:
$\Longrightarrow$ Simple Alignment Problem

$$D_{ij} = \min \begin{cases} D_{i,j-1} + 1 \\ D_{i-1,j} + 1 \\ D_{i-1,j-1} \end{cases} \quad \text{whenever } \pi'(i) = \pi''(j)$$



The Circular Case: Mitochondrial Genomes

Arrangement of 13 protein coding genes, 2 rRNAs and 22tRNAs

```
> NC_000834.cgi Branchiostoma floridae
CO1 -S2 D CO2 K ATP8 ATP6 CO3 ND3 R ND4L ND4 H S L1 ND5 G -ND6
-E CYTB T -P 12S F V 16S L2 ND1 I M -Q ND2 -N W -A -C -Y
```

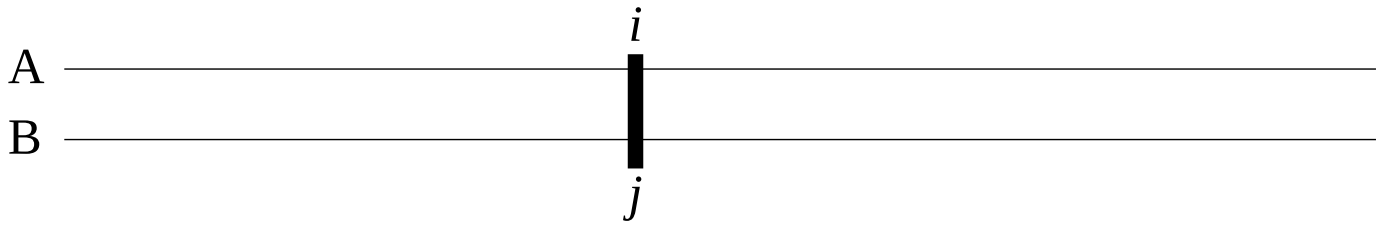
Traditional approaches:

- Break-point distances
- "Sorting by reversals"

$\xi = \pi' \circ \pi''^{-1}$ and compute a *length function* of ξ, i.e. number of reversals necessary to convert ξ to the identity permutation

Circular Alignments

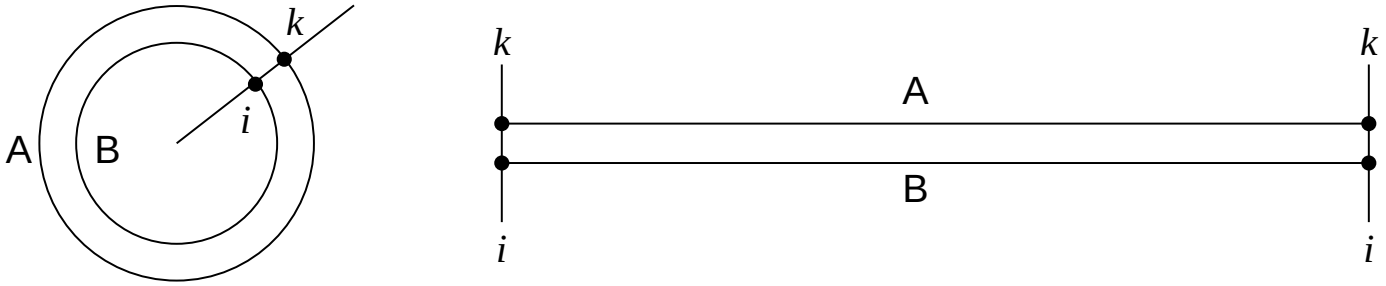
Basic Property of Alignments:



Match (i, j) separates alignment into two independent parts.

If (i, j) is part of the optimal alignment, then the two parts can be optimized independently.

Computing Circular Alignments



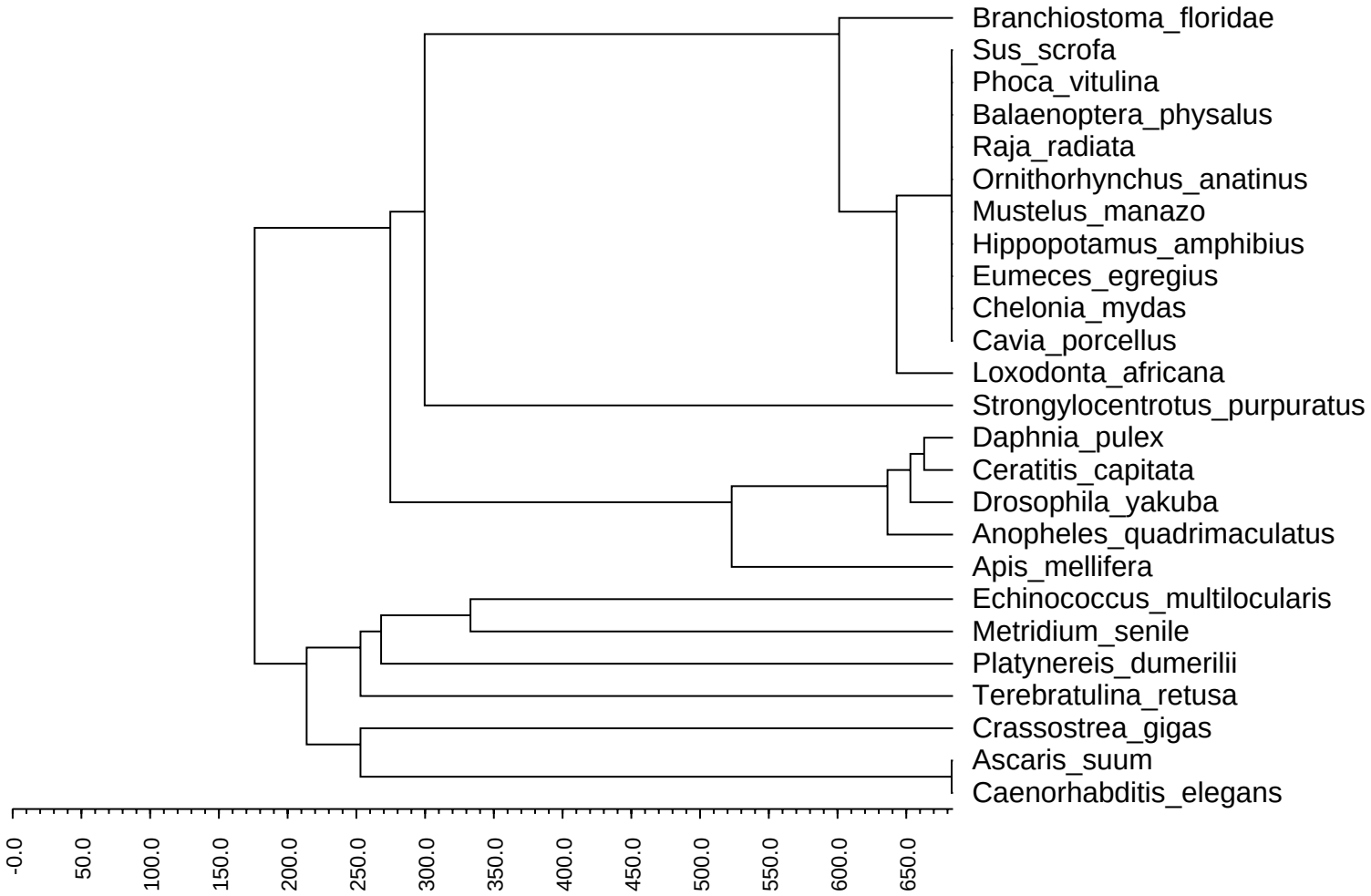
Rotate sequence A such that it starts with $i+1$: $\rightarrow A'$
Rotate sequence B such that it starts with $k+1$: $\rightarrow B'$
Compute Alignment of A' and B' with the restriction that (a) inital gaps have full costs and (b) the last entries (i.e., i and k) match.

Polynomial algorithm that runs in $n^2 \times$ Alignment(A', B') time with $\mathcal{O}(n^2)$ memory.

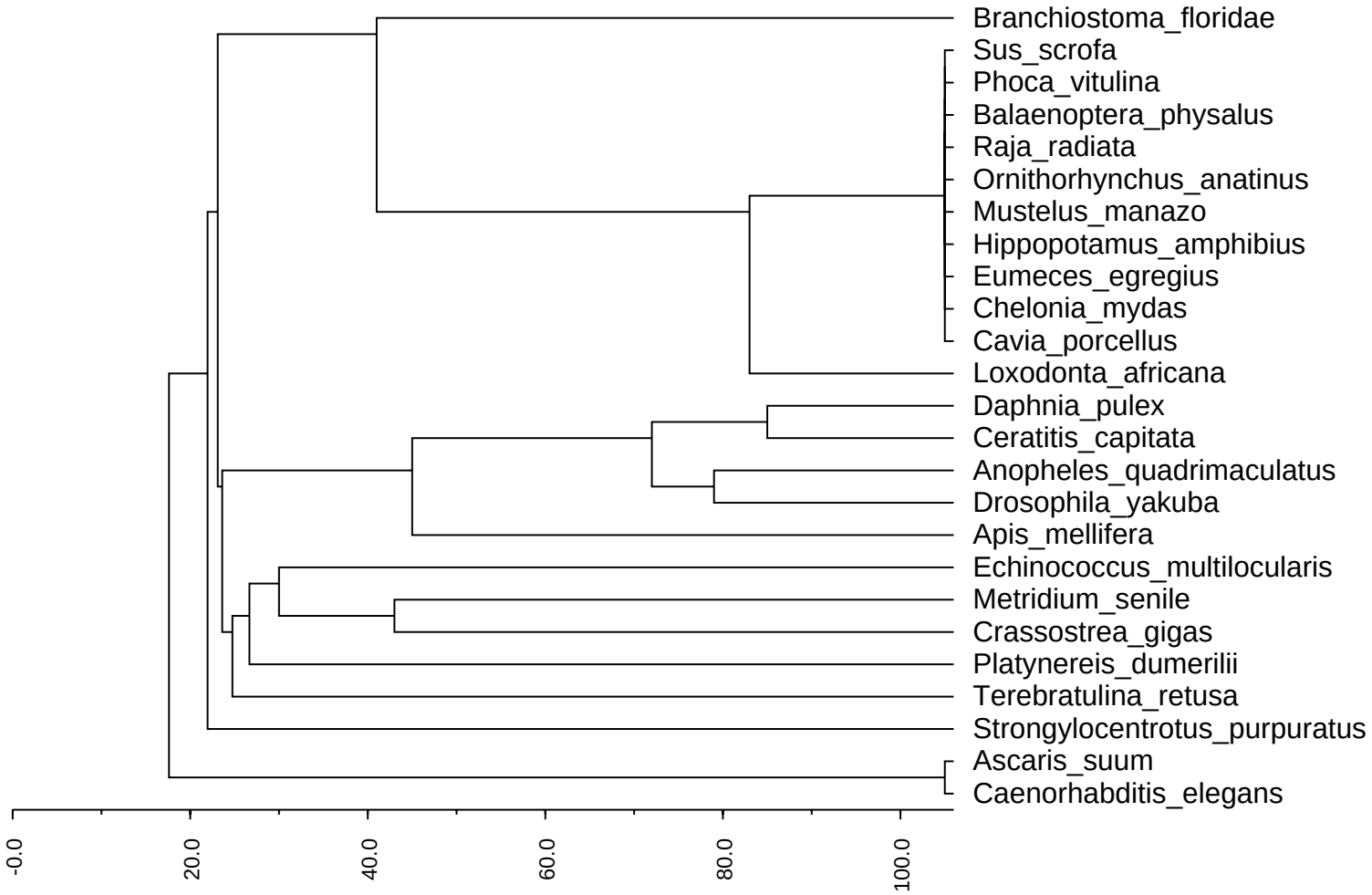
Resulting Alignments:

```
'NC_000834'   <Branchiostoma_floridae       >   1110110111010111111111100000000001111011110000000
'NC_000845'   <Sus_scrofa                   >   1110110111110111111011100000000001101111110000000
'NC_000884'   <Cavia_porcellus              >   1110110111110111111011100000000001101111110000000
'NC_000886'   <Chelonia_mydas               >   1110110111110111111011100000000001101111110000000
'NC_000888'   <Eumeces_egregius             >   1110110111110111111011100000000001101111110000000
'NC_000889'   <Hippopotamus_amphibius       >   1110110111110111111011100000000001101111110000000
'NC_000890'   <Mustelus_manazo              >   1110110111110111111011100000000001101111110000000
'NC_000844'   <Daphnia_pulex                >   1001111111111000000000001101111111000000011110 1
'NC_000857'   <Ceratitis_capitata           >   1001111111111000000000001101111111000000011011 1
'NC_000875'   <Anopheles_quadrimaculatus    >   1001111111110100000000001101111111000000011110 1
```

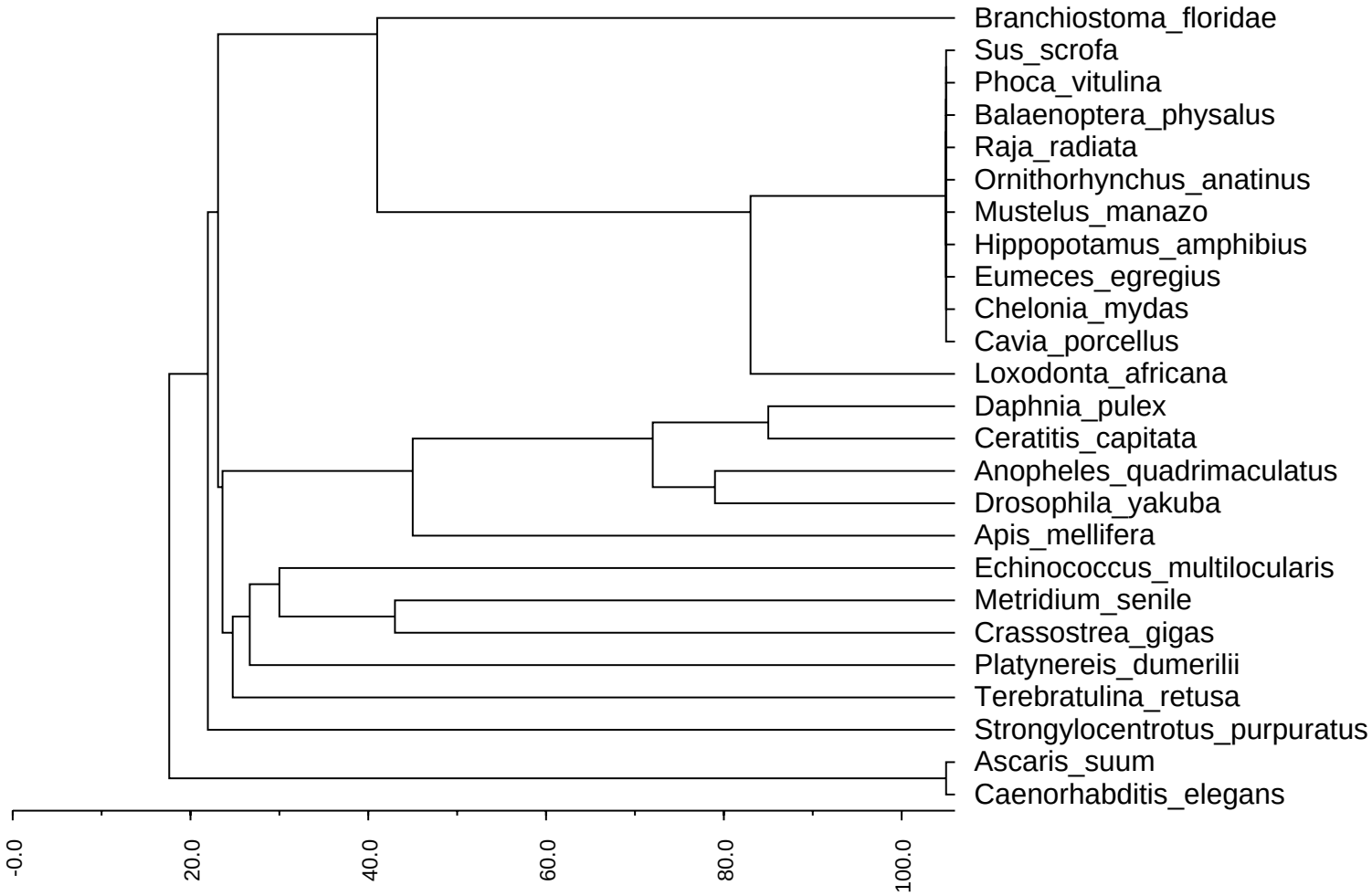
(For simplicity: representing presence/absence of genes only)



gap open = 10, gap extend = 10



gap open = 10, gap extend = 3



gap open = 10, gap extend = 1

Outlook

tRNAs and proteins "move" with different frequencies:
$\Longrightarrow$ more sophisticated scoring model: Gap costs depends on contents and length

Acknowledgements

Footprint Sorting:
Sonja J. Prohaska, Claudia Fried, Claus R. Stadler (Leipzig)
Wim Hordijk (Canterbury, NZ)

Mitochondrial Genomes:
Guido Fritzsch, Martin Schlegel