MAFFT

Multiple sequence Alignment using Fast Fourier Transform

Advanced Bioinformatics 1 – Fall 2021

Dongwook Kim
27 Oct, 2021

Outline

- Multiple Sequence Alignment
- MAFFT Algorithm
- Utilizing MAFFT

Features

MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform

Kazutaka Katoh, Kazuharu Misawa[1], Kei-ichi Kuma and Takashi Miyata*

Nucleic Acids Research, 2002

MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability

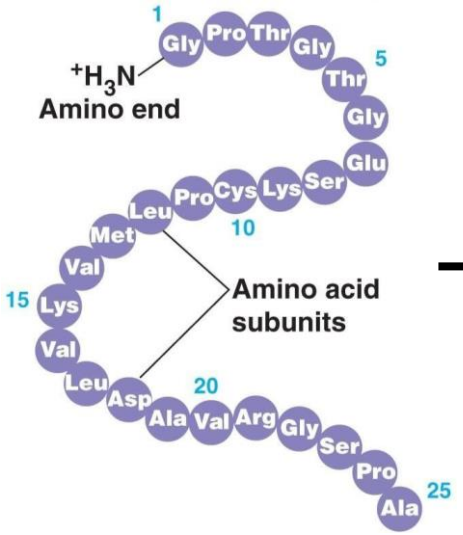
Kazutaka Katoh*[,1,2] and Daron M. Standley[1]

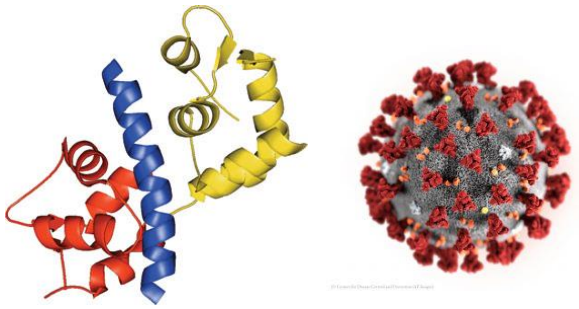
Mol. Biol. Evol., 2013

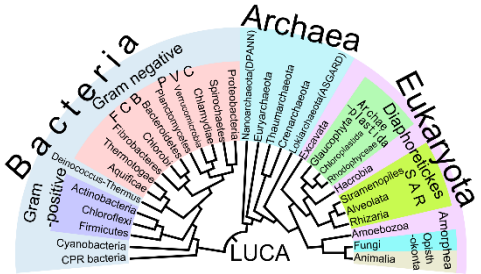
Multiple Sequence Alignment

About multiple sequence alignment

- Multiple sequence alignment (MSA) is a crucial technique for fields of computational biology, including:
  - Homology search, protein domain finding, evolutionary variation finding, *etc*.







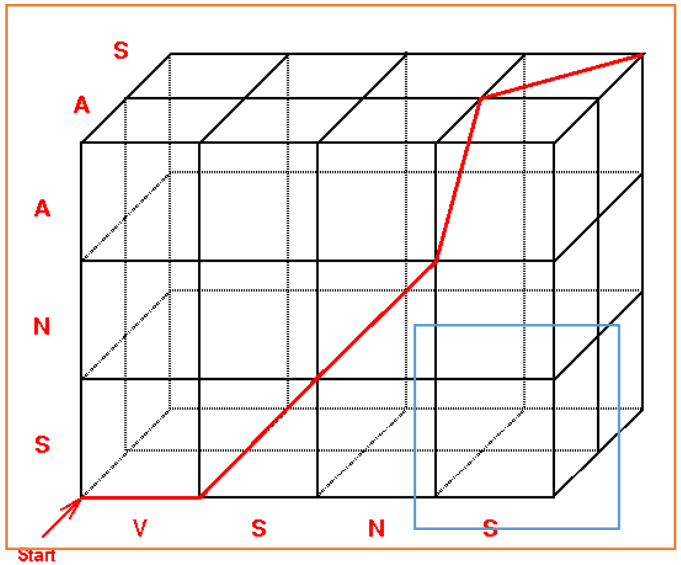
- However…

https://publish.illinois.edu/msaevaluation/

MSA is a challenging task

- Finding “global-optimum” of MSA task, which requires N-dimensional dynamic programming (DP), consumes excessive resources.

- Example for 3 short protein sequences:



http://bioinfo3d.cs.tau.ac.il/Education/CS99b/class_notes/class3.html

```
V S N _ S
_ S N A _
_ _ _ A S
```

- N sequences with length L

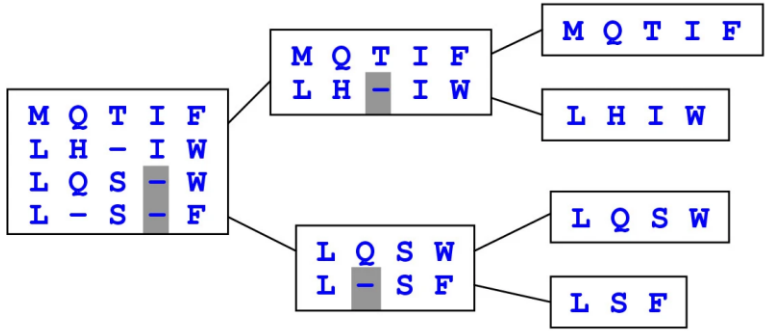
- $O(L^N)$-sized DP matrix

- Each block requires $O(2^N)$ previous blocks

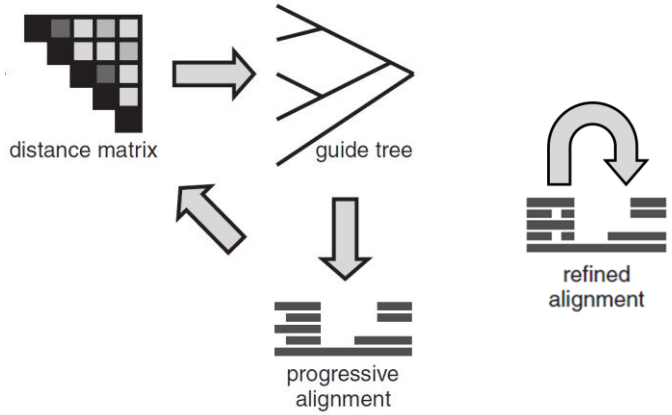
→ Time complexity of $O(2^N L^N)$

MSA can be sped up by heuristics

- Necessity of trading between time and accuracy
- Heuristics based on the concept of “guide tree” can be introduced:



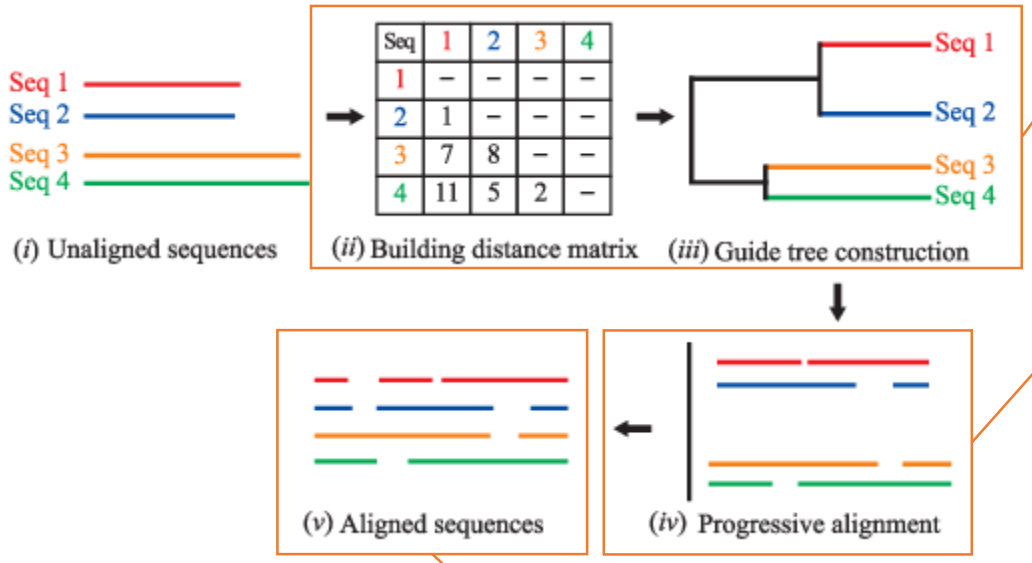
Progressive method



Iterative refinement

Edgar, Robert C. "MUSCLE: multiple sequence alignment with high accuracy and high throughput." *Nucleic acids research* 32.5 (2004): 1792-1797.
http://ai.stanford.edu/~chuongdo/papers/alignment_review.pdf

Brief introduction to the progressive method



- Distance matrix and guide tree
- Group-to-group alignment
- Suboptimal result

Lalwani, Soniya, et al. "Efficient discrete firefly algorithm for Ctrie based caching of multiple sequence alignment on optimally scheduled parallel machines." *CAAI Transactions on Intelligence Technology* 4.2 (2019): 92-100.

MSA can be sped up by heuristics

- Progressive method improves computational efficacy quite dramatically, with decent accuracy.
- Implemented on various tools such as CLUSTAL, MUSCLE, *etc*.

Table 2: Complexity of **MUSCLE**. Here we show the big-O asymptotic complexity of the elements of **MUSCLE** as a function of L, the typical sequence length, and N, the number of sequences, retaining the highest-order terms in N with L fixed and vice versa.

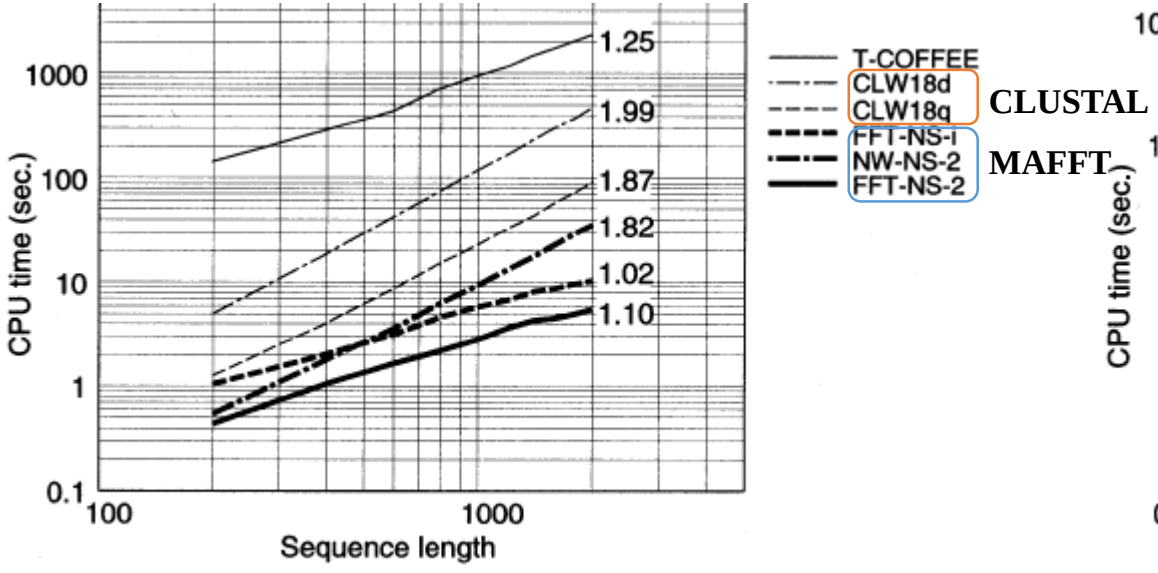
Step	O(Space)	O(Time)
K-mer distance matrix	$N^2 + L$	N^2L
UPGMA	N^2	N^2
Progressive (one iteration)	$L_p^2 = NL + L^2$	$L_p^2 = N^2 + L^2$
Progressive (root alignment)	$NL_p = N^2 + NL$	$NL_p \log N = N^2 \log N + NL \log N$
Progressive (N iterations + root)	$N^2 + NL + L^2$	$N^3 + NL^2$
Refinement (one edge)	$NL_p + L_p^2 = N^2 + L^2$	$N^2L_p + L_p^2 = N^3 + L^2$
Refinement (N edges)	$N^2 + L^2$	$N^4 + NL^2$
TOTAL	$N^2 + L^2$	$N^4 + NL^2$

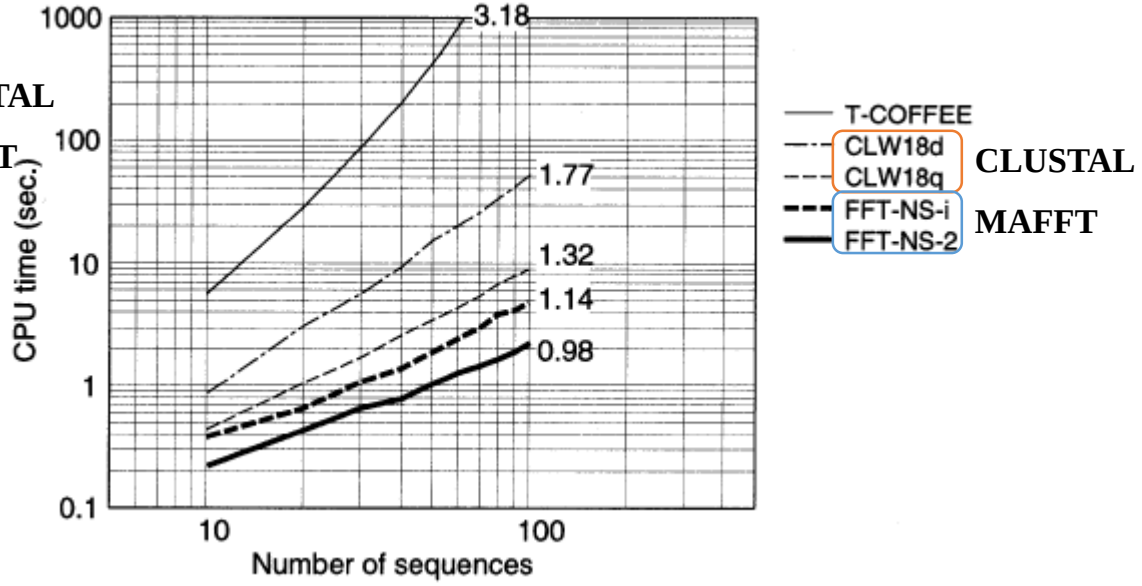
- ***N*** sequences with length ***L***
- $\boldsymbol{O(N^4+NL^2)}$ <<<< $\boldsymbol{O(2^N L^N)}$

Saeed, Fahad, and Ashfaq Khokhar. "An Overview of Multiple Sequence Alignment Systems." *arXiv preprint arXiv:0901.2747* (2009).

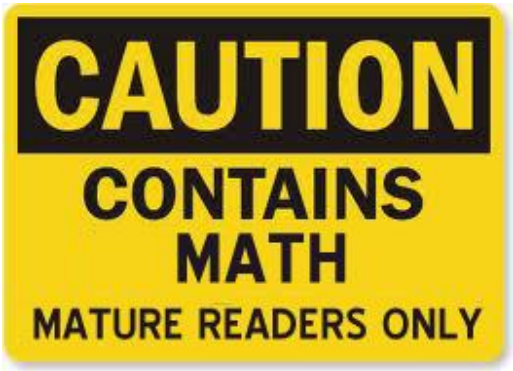
These tools are not fast enough

- Tools with naïve heuristics are still slow for massive analyses.
- MAFFT dramatically improved this performance… but how?



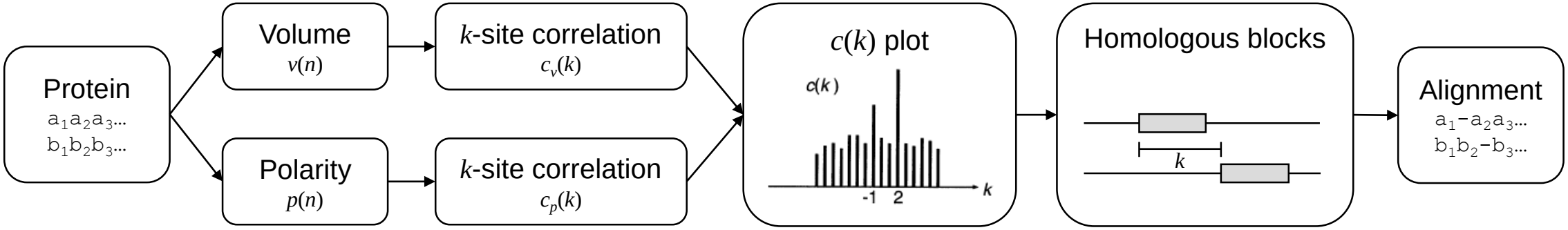


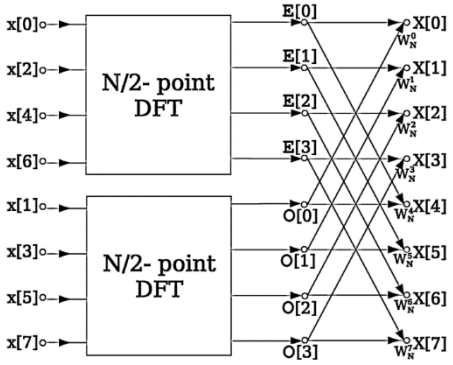
MAFFT Algorithm



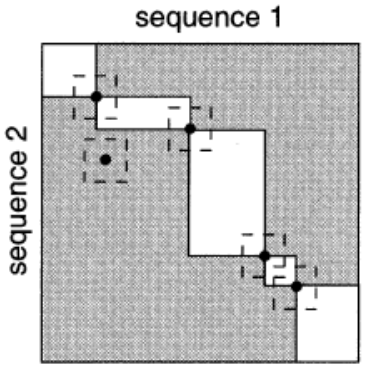
https://strawstickstone.com/blog/the-2-week-do-i-really-need-an-ipad-test/

Overview of MAFFT algorithm (pairwise)





Fast Fourier Transform



Segment-level DP

Definition of *k*-site correlation

- *k*-site correlation: The degree of <u>similarity</u> between two sequences with the <u>positional lag</u> of <u>*k*</u> sites

```
...MAGLWGKAAAEEEGLWLFFFT...
...ADKGVAAAEQEGLFVFFGGIF...
```

$k = 2$ →

```
...GLWGKAAAEEEGLWLFFFT...
...ADKGVAAAEQEGLFVFFGG...
```

$c(0) = 0.1$

$c(2) = \mathbf{2.5}$

Definition of *k*-site correlation

- *k*-site correlation: The degree of <u>similarity</u> between two sequences with the <u>positional lag</u> of <u>*k*</u> sites

```
...MAGLWGKAAAEEEGLWLFFFT...
...ADKGVAAAEQEGLFVFFGGIF...
```

$\xrightarrow{k = 2}$

```
...GLWGKAAAEEEGLWLFFFT...
...ADKGVAAAEQEGLFVFFGG...
```

$c(0) = 0.1$ $\qquad$ $c(2) = \mathbf{2.5}$

$$c(k) = \boxed{c_v(k)} + \boxed{c_p(k)}$$

Correlation of **amino acid volumes** by the positional lag of ***k***

Correlation of **amino acid polarity** by the positional lag of ***k***

$$c_v(k) = \sum_{1 \le n \le N, 1 \le n+k \le M} \boxed{\hat{v}_1(n)\hat{v}_2(n+k)}$$

High if $v_1(n)$ and $v_2(n+k)$ are similar

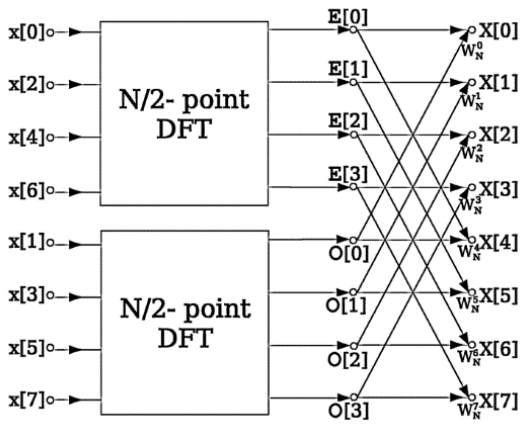
Low (even negative) otherwise

Fast Fourier transform speeds up the calculation

- Calculation of k-site correlation requires $O(N^2)$ operations.

$$c_v(k) = \sum_{1 \le n \le N, 1 \le n+k \le M} \boxed{\hat{v}_1(n)\hat{v}_2(n+k)}$$

- By applying FFT, operations drop to $O(N \log N)$.



How FFT achieves $O(N \log N)$ time complexity

- Fourier transform of $v(n)$ reshapes the <u>summation task</u> into a <u>complex vector multiplication task</u>.

- Discrete Fourier transform (DFT) will be used here.

$$c_v(k) = \sum_{1 \le n \le N, 1 \le n+k \le M} \hat{v}_1(n)\hat{v}_2(n+k)$$

Summation - $O(N^2)$

$$c_v(k) \underset{FT}{\Leftrightarrow} V_1{}^*(m) \cdot V_2(m)$$

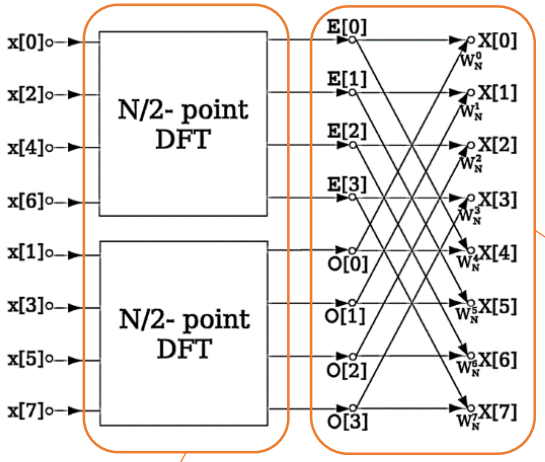
Complex vector mult. - $O(N^2)$

How FFT achieves $O(N \log N)$ time complexity

- N/2-point DFT breaks the task into half based on their parity.
- At the end (size 1), multiplication can be done with a constant amount of operation.

$$c_v(k) \underset{FT}{\Leftrightarrow} V_1^*(m) \cdot V_2(m)$$

Complex vector mult. - $O(N^2)$



By the property of Fourier matrix, this merging process can be done by $O(N)$ time complexity.

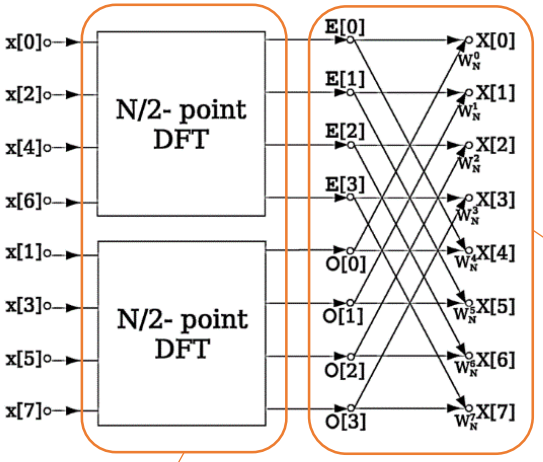
This part breaks down the task into half until the size becomes 1.

How FFT achieves $O(N \log N)$ time complexity

- The entire task can be done with $O(N \log N)$ time complexity.

$$c_v(k) \underset{FT}{\Leftrightarrow} V_1^*(m) \cdot V_2(m)$$

Complex vector mult. - $O(N^2)$



We have to break the task $O(\log N)$ times until it reaches size of 1.

Each broken down step requires $O(N)$ time complexity.

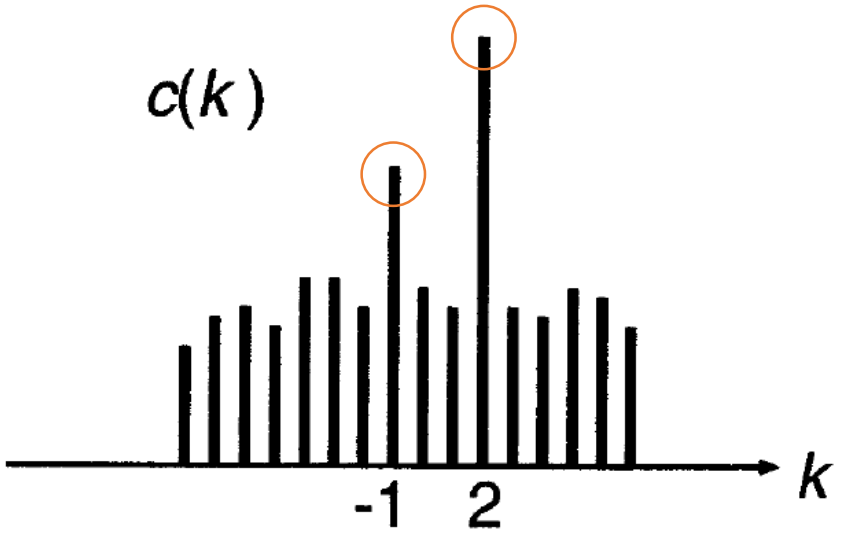
$O(\log N) \times O(N) = O(N \log N)$ ■

From correlation to homology

- k-site correlation: The degree of <u>similarity</u> between two sequences with the <u>positional lag</u> of <u>k</u> sites

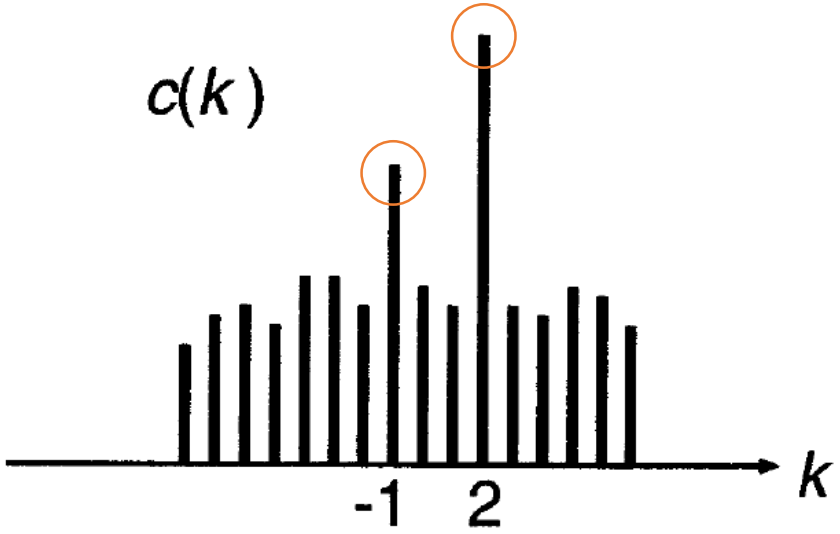
- Peaks from $c(k)$ plot represent <u>the lags with high potential of homology</u>

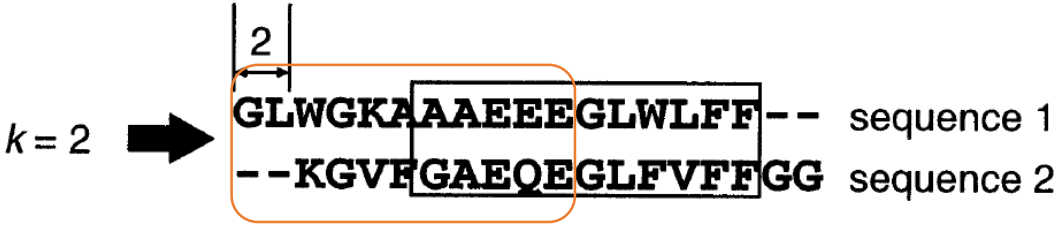
$$c(k) = c_v(k) + c_p(k)$$



Finding homologous segments with $c(k)$

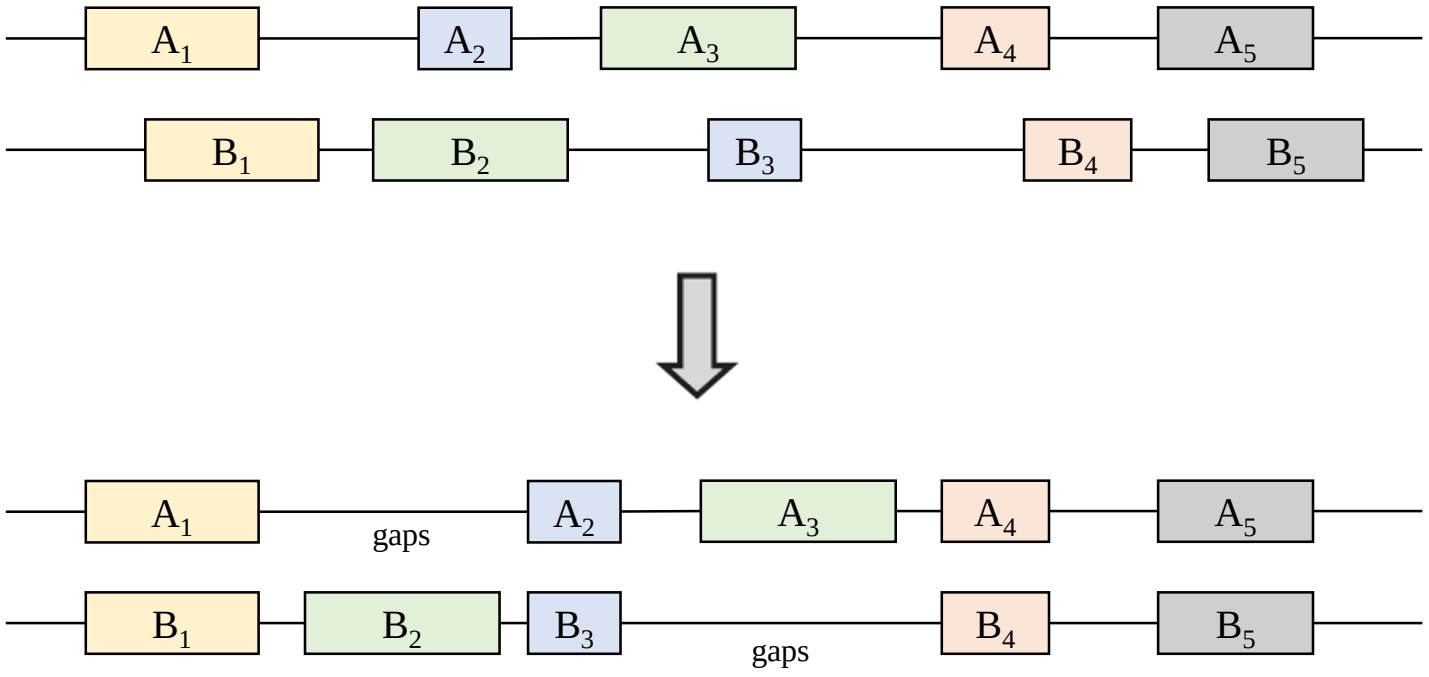
- Obtain values of k exceeding certain threshold, and align the sequences applying such positional lags
- Run sliding window analysis to find the homologous region

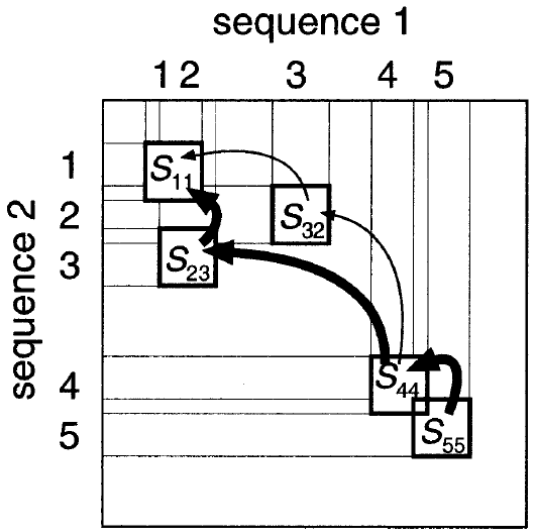




About segment-level dynamic programming

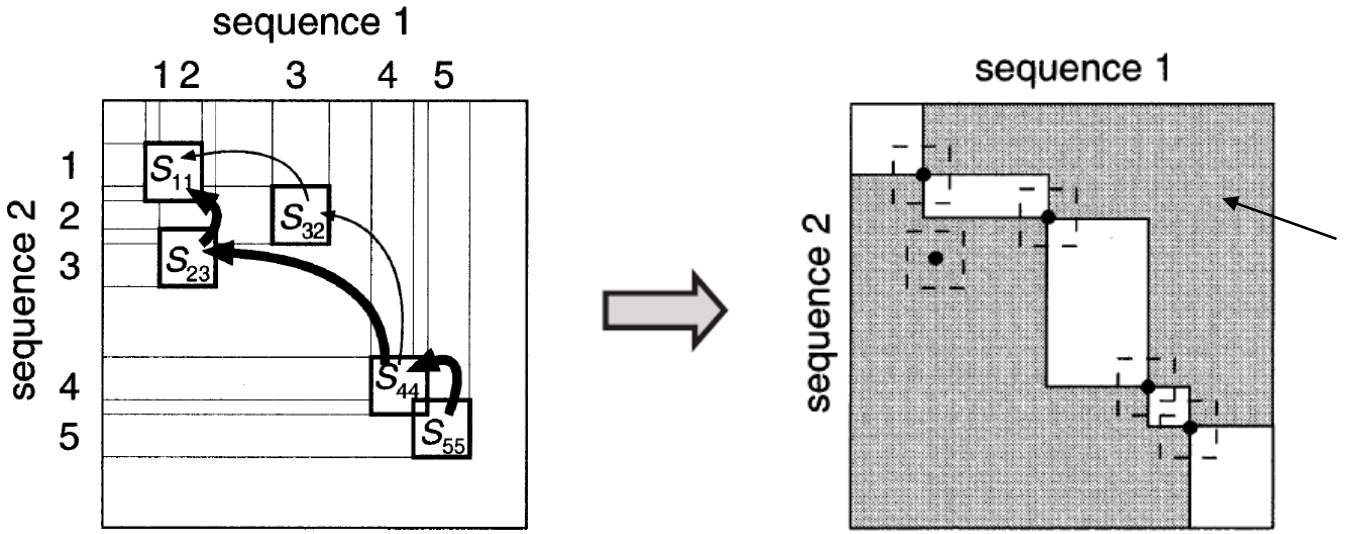
- First, align homologous segments and find optimal arrangement of the segments.





About segment-level dynamic programming

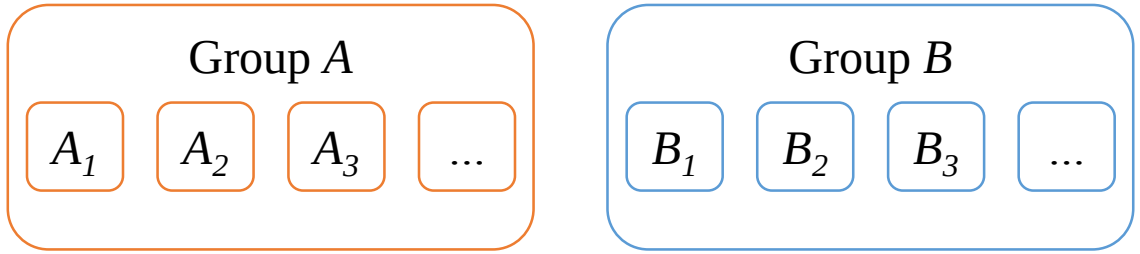
- Then, align remaining sites with reduced dynamic programming matrix.



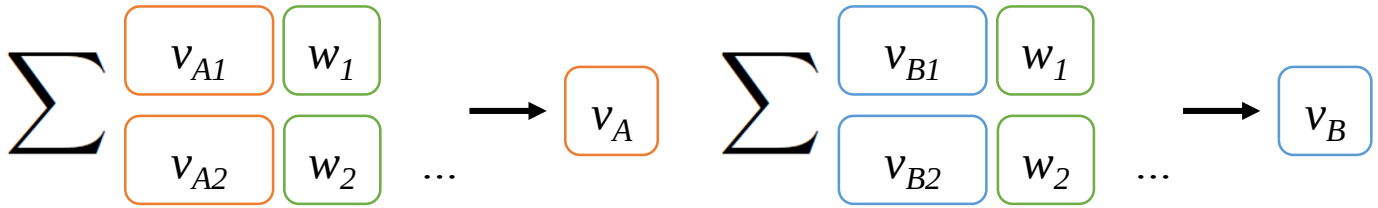
Shaded area is excluded from the calculation.

Extending pairwise to group-wise alignment

- Consider an example of performing alignment of group A and B, which consist of sequences A_1, A_2, … and B_1, B_2, …

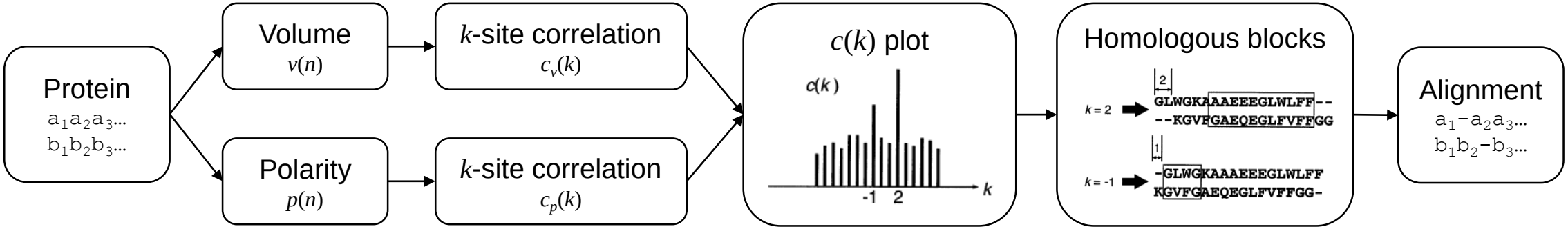


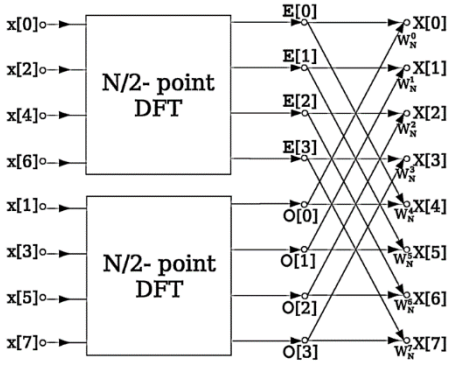
- By applying weighting factor, we can define a ‘group components’ of volume and polarity.



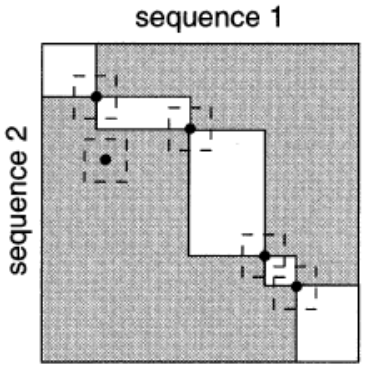
- Performing identical process on group component results in group-wise alignment. Easy!

Review on MAFFT algorithm





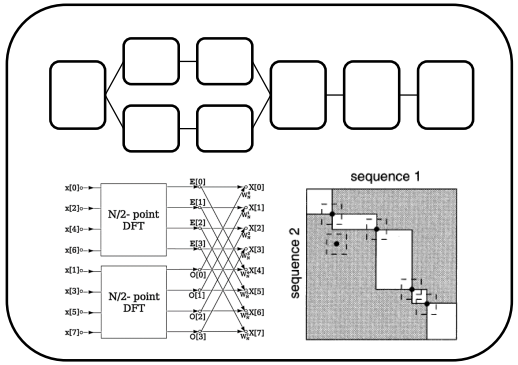
Fast Fourier Transform



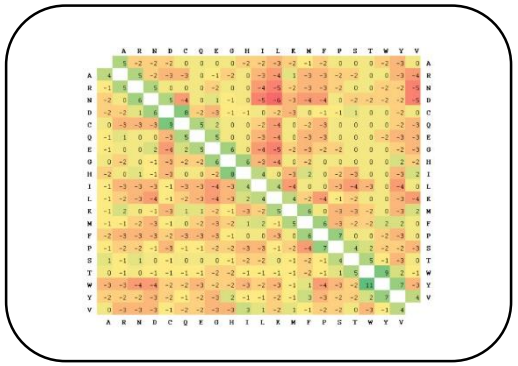
Segment-level DP

Outlining FFT-NS-2 and FFT-NS-i

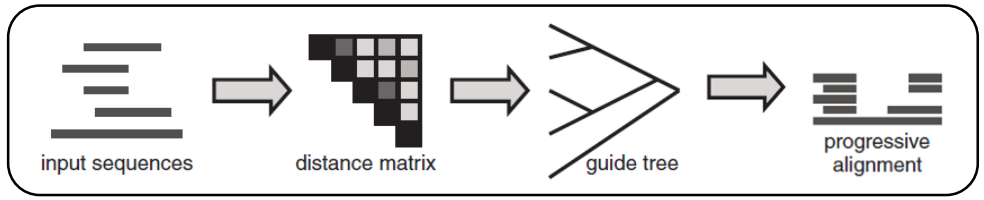
***FFT** procedure*



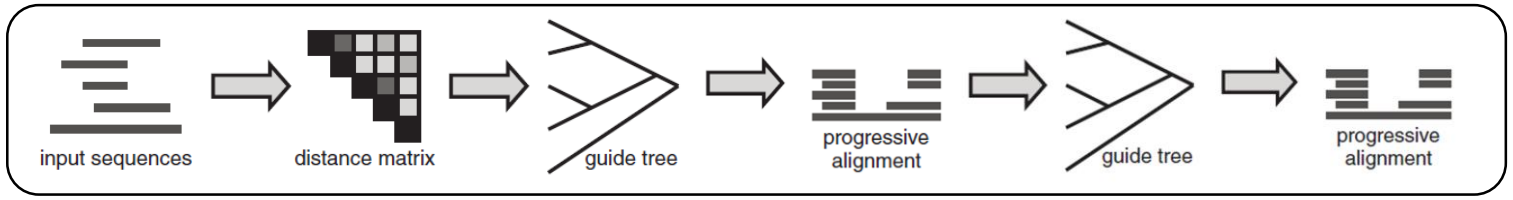
***N**ormalized **S**imilarity matrix*



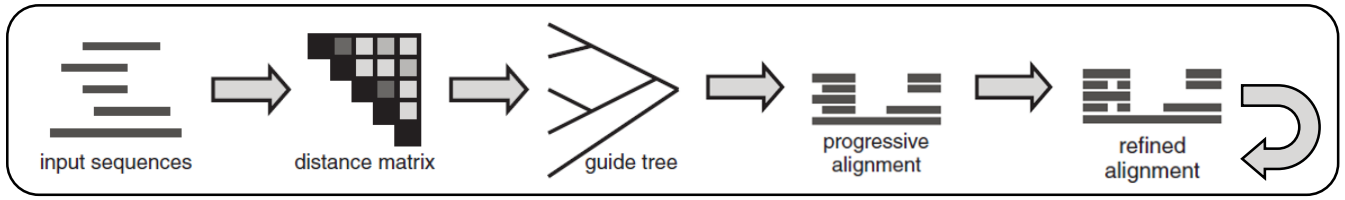
+



= FFT-NS-1



= FFT-NS-2



= FFT-NS-i

Utilizing MAFFT with CO

https://colab.research.google.com/drive/1KyKyKD2H_a60RIgFzDqVbRfKC3D_Gxo0?usp=sharing

Running MAFFT

```
!mafft/bin/mafft -h
```

```
mafft/bin/mafft: Cannot open -h.

--------------------------------------------------------------------------------
  MAFFT v7.487 (2021/Jul/25)
  https://mafft.cbrc.jp/alignment/software/
  MBE 30:772-780 (2013), NAR 30:3059-3066 (2002)
--------------------------------------------------------------------------------
High speed:
  % mafft in > out
  % mafft --retree 1 in > out (fast)

High accuracy (for <~200 sequences x <~2,000 aa/nt):
  % mafft --maxiterate 1000 --localpair  in > out (% linsi in > out is also ok)
  % mafft --maxiterate 1000 --genafpair  in > out (% einsi in > out)
  % mafft --maxiterate 1000 --globalpair in > out (% ginsi in > out)

If unsure which option to use:
  % mafft --auto in > out

--op # :         Gap opening penalty, default: 1.53
--ep # :         Offset (works like gap extension penalty), default: 0.0
--maxiterate # : Maximum number of iterative refinement, default: 0
--clustalout :   Output: clustal format, default: fasta
--reorder :      Outorder: aligned, default: input order
--quiet :        Do not report progress
--thread # :     Number of threads (if unsure, --thread -1)
--dash :         Add structural information (Rozewicki et al, submitted)
```

FFT-NS-2 (Fast; progressive)

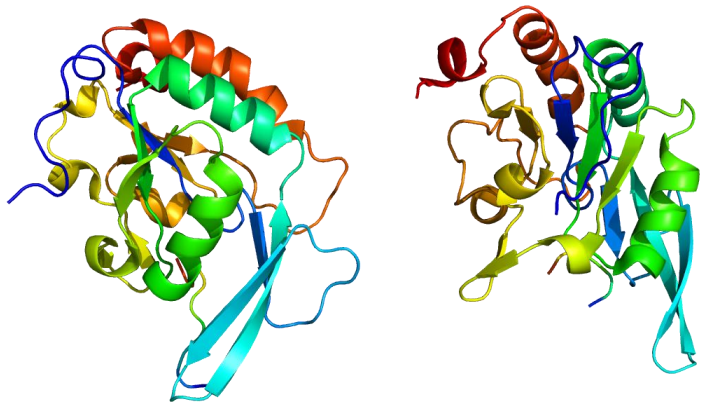
```
$ mafft --maxiterate 0 input > output
```

FFT-NS-i (Iterative)

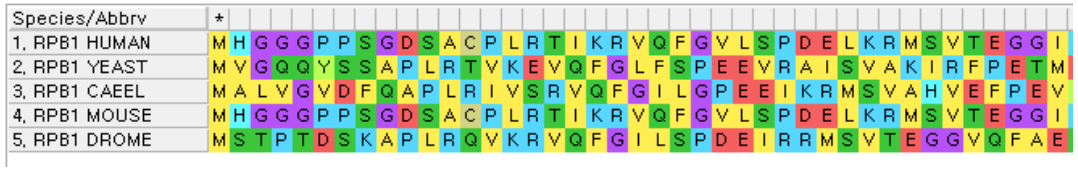
```
$ mafft --maxiterate 1000 input > output
```

Example Dataset: RPB1

- DNA-directed RNA polymerase II subunit RPB1 (POLR2A)
- Protein sequences retrieved from UniProt



https://en.wikipedia.org/wiki/POLR2A



- `RPB1_HUMAN` : *Homo sapiens*
- `RPB1_YEAST` : *Saccharomyces cerevisiae*
- `RPB1_CAEEL` : *Caenorhabditis elegans*
- `RPB1_MOUSE` : *Mus musculus*
- `RPB1_DROME` : *Drosophila melanogaster*

Running MAFFT – FFT-NS-2

```
!mafft/bin/mafft --maxiterate 0 drive/MyDrive/Colab₩ Notebooks/mafft/rpb1.fasta > rpb1_ns2.fasta
```

```
nthread = 0
nthreadpair = 0
nthreadtb = 0
ppenalty_ex = 0
stacksize: 8192 kb
rescale = 1
Gap Penalty = -1.53, +0.00, +0.00


Making a distance matrix ..
    1 / 5
done.

Constructing a UPGMA tree (efffree=0) ...
    0 / 5
done.

Progressive alignment 1/2...
STEP     4 / 4
done.

Making a distance matrix from msa..
    0 / 5
done.
```

Guide tree #1

Prog. align #1

```
Constructing a UPGMA tree (efffree=1) ...
    0 / 5
done.

Progressive alignment 2/2...
STEP     4 / 4
done.

disttbfast (aa) Version 7.487
alg=A, model=BLOSUM62, 1.53, -0.00, -0.00, noshift, amax=0.0
0 thread(s)


Strategy:
 FFT-NS-2 (Fast but rough)
 Progressive method (guide trees were built 2 times.)

If unsure which option to use, try 'mafft --auto input > output'.
For more information, see 'mafft --help', 'mafft --man' and the mafft page.

The default gap scoring scheme has been changed in version 7.110 (2013 Oct).
It tends to insert more gaps into gap-rich regions than previous versions.
To disable this change, add the --leavegappyregion option.
```

Guide tree #2

Prog. align #2

Running MAFFT – FFT-NS-i

```
!mafft/bin/mafft --maxiterate 1000 drive/MyDrive/Colab# Notebooks/mafft/rpb1.fasta > rpb1_nsi.fasta
```

```
...

Segment   1/ 35    1-  29
STEP 005-001-1  identical.
Converged.

Segment   2/ 35   29-  88
STEP 002-003-1  identical.
Converged.

Segment   3/ 35   88- 151
STEP 002-003-1  identical.
Converged.

Segment   4/ 35  151- 196
STEP 003-001-1  identical.
Converged.

Segment   5/ 35  196- 270
STEP 003-001-0  identical.
Converged.

...
```

Iterate each segment until convergence

```
...

Strategy:
 FFT-NS-i (Accurate but slow)
 Iterative refinement method (max. 16 iterations)

If unsure which option to use, try 'mafft --auto input > output'.
For more information, see 'mafft --help', 'mafft --man' and the mafft page.

The default gap scoring scheme has been changed in version 7.110 (2013 Oct).
It tends to insert more gaps into gap-rich regions than previous versions.
To disable this change, add the --leavegappyregion option.
```

Comparing results by peeking files

```
!head rpb1_ns2.fasta
```

```
>RPB1_HUMAN
MHGGGPPSGDSACPLRTIKRVQFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLM
DPRQGVIERTGRCQTCAGNMTECPGHFGHIELAKPVFHVGFLVKTMKVLRCVCFFCSKLL
VDSNNPKIKDILAKSKGQPKKRLTHVYDLCKGKNICEGGEEMDNKFGVEQPEGDEDL--T
KEKGHGGCGRYQPRIRRSGLELYAEW-KH-VNEDSQEKKI-LLSPERVHEIFKRISDEEC
FVLGMEPRYARPEWMIVTVLPVPPLSVRPAVVMQGSARNQDDLTHKLADIVKINNQLRRN
EQNGAAAHVIAEDVKLLQFHVATMVDNELPGLPRAMQKSGRPLKSLKQRLKGKEGRVRGN
LMGKRVDFSARTVITPDPNLSIDQVGVPRSIAANMTFAEIVTPFNIDRLQELVRRGNSQY
PGAKYIIRDNGDRIDLRFHPKPSDLHLQTGYKVERHMCDGDIVIFNRQPTLHKMSMMGHR
VRILPWSTFRLNLSVTTPYNADFDGDEMNLHLPQSLETRAEIQELAMVPRMIVTPQSNRP
```

```
!head rpb1_nsi.fasta
```

```
>RPB1_HUMAN
MHGGGPPSGDSACPLRTIKRVQFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLM
DPRQGVIERTGRCQTCAGNMTECPGHFGHIELAKPVFHVGFLVKTMKVLRCVCFFCSKLL
VDSNNPKIKDILAKSKGQPKKRLTHVYDLCKGKNICEGGEEMDNKFGVEQPEGDEDLTK-
-EKGHGGCGRYQPRIRRSGLELYAEWKH--VNEDSQEKKI-LLSPERVHEIFKRISDEEC
FVLGMEPRYARPEWMIVTVLPVPPLSVRPAVVMQGSARNQDDLTHKLADIVKINNQLRRN
EQNGAAAHVIAEDVKLLQFHVATMVDNELPGLPRAMQKSGRPLKSLKQRLKGKEGRVRGN
LMGKRVDFSARTVITPDPNLSIDQVGVPRSIAANMTFAEIVTPFNIDRLQELVRRGNSQY
PGAKYIIRDNGDRIDLRFHPKPSDLHLQTGYKVERHMCDGDIVIFNRQPTLHKMSMMGHR
VRILPWSTFRLNLSVTTPYNADFDGDEMNLHLPQSLETRAEIQELAMVPRMIVTPQSNRP
```

FFT-NS-2 : ...**GLELYAEW-KH-VNEDSQEKKI-LLSPER**...

FFT-NS-i : ...**GLELYAEWKH--VNEDSQEKKI-LLSPER**...

Comparing results with ClustalW format

```
!mafft/bin/mafft --quiet --clustalout --maxiterate 0 drive/MyDrive/Colab# Notebooks/mafft/rpb1.fasta > rpb1_ns2_cl.out
!mafft/bin/mafft --quiet --clustalout --maxiterate 1000 drive/MyDrive/Colab# Notebooks/mafft/rpb1.fasta > rpb1_nsi_cl.out
!head rpb1_ns2_cl.out
!head rpb1_nsi_cl.out
```

```
CLUSTAL format alignment by MAFFT FFT-NS-2 (v7.487)


RPB1_HUMAN      MHGGGPPSGDSACPLRTIKRVQFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLM
RPB1_YEAST      ----MVGQQYSSAPLRTVKEVQFGLFSPEEVRAISVAK--IRFPETMDETQTRAKIGGLN
RPB1_CAEEL      ---MALVGVDFQAPLRIVSRVQFGILGPEEIKRMSVAH--VEFPEVYE--NGKPKLGGLM
RPB1_MOUSE      MHGGGPPSGDSACPLRTIKRVQFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLM
RPB1_DROME      ---MSTPT-DSKAPLRQVKRVQFGILSPDEIRRMSVTEGGVQFAETME--GGRPKLGGLM
                         .*** :.,****::.*:*:: :**:.  :.:.*. :    :.*:***

CLUSTAL format alignment by MAFFT FFT-NS-i (v7.487)


RPB1_HUMAN      MHGGGPPSGDSACPLRTIKRVQFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLM
RPB1_YEAST      MVGQ----QYSSAPLRTVKEVQFGLFSPEEVRAISVAK--IRFPETMDETQTRAKIGGLN
RPB1_CAEEL      MALVG---VDFQAPLRIVSRVQFGILGPEEIKRMSVAH--VEFPEVYE--NGKPKLGGLM
RPB1_MOUSE      MHGGGPPSGDSACPLRTIKRVQFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLM
RPB1_DROME      MSTP----TDSKAPLRQVKRVQFGILSPDEIRRMSVTEGGVQFAETME--GGRPKLGGLM
                *          .*** :.,****::.*:*:: :**:.  :.:.*. :    :.*:***
```

FFT-NS-i identified initiation codon

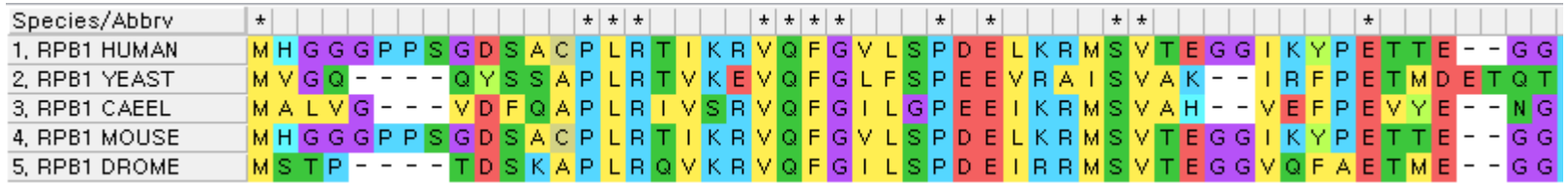
- * → Identical site
- : → Conserved substitutions
- . → Semi-conserved substitutions
- → Not conserved

Comparing results with external program

- Visualization with MEGA-X software

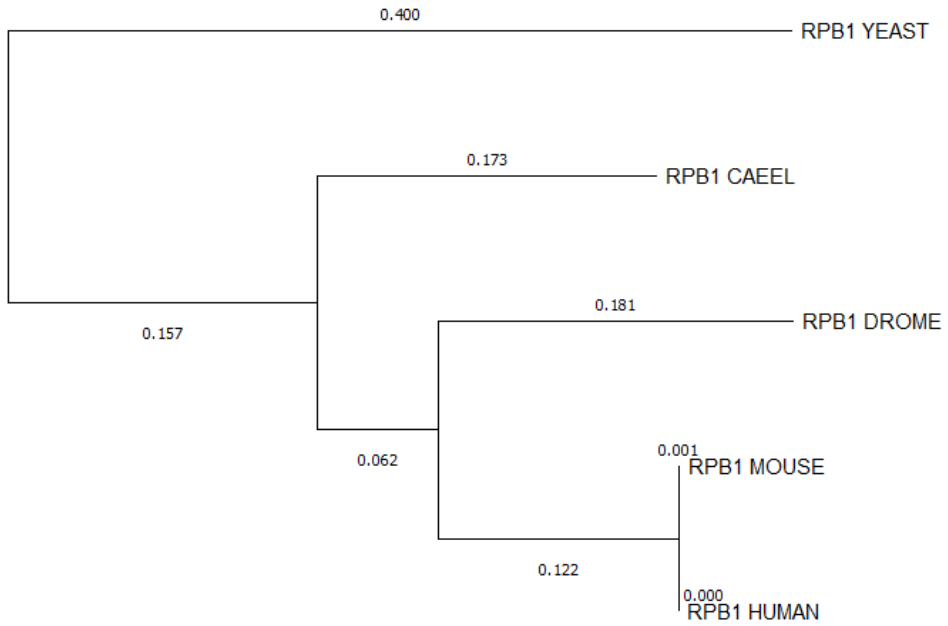
FFT-NS-2

FFT-NS-i

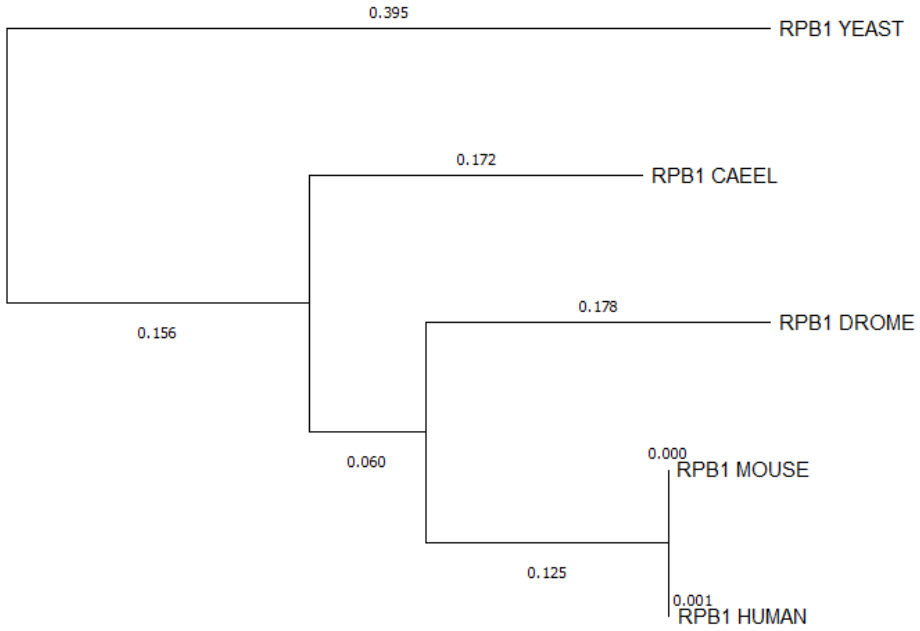


Comparing results with external program

- Maximum likelihood tree by MEGA-X software



FFT-NS-2



FFT-NS-i

References

Edgar, Robert C. "MUSCLE: multiple sequence alignment with high accuracy and high throughput." *Nucleic acids research* 32.5 (2004): 1792-1797.

Katoh, Kazutaka, et al. "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform." *Nucleic acids research* 30.14 (2002): 3059-3066.

Katoh, Kazutaka, and Daron M. Standley. "MAFFT multiple sequence alignment software version 7: improvements in performance and usability." *Molecular biology and evolution* 30.4 (2013): 772-780.

Lalwani, Soniya, et al. "Efficient discrete firefly algorithm for Ctrie based caching of multiple sequence alignment on optimally scheduled parallel machines." *CAAI Transactions on Intelligence Technology* 4.2 (2019): 92-100.

Saeed, Fahad, and Ashfaq Khokhar. "An Overview of Multiple Sequence Alignment Systems." *arXiv preprint arXiv:0901.2747* (2009).

YouTube – “The Fast Fourier Transform (FFT): Most Ingenious Algorithm Ever?” by Reducible: https://youtu.be/h7apO7q16V0

THANK YOU FOR LISTENING!