# *Scoring multiple sequence aligments with pyMSA*

24/01/2019

cc BY SA

Antonio J. Nebro
Antonio Benítez-Hidalgo

UNIVERSIDAD DE MÁLAGA

khaos RESEARCH

## Multiple sequence alignment

- The multiple sequence alignment (MSA) problem can be defined as:
  - Finding an optimum alignment of three or more biological sequences (DNA, RNA, proteins) to identify common regions
  - These **highly-conserved regions** may be a consequence of functional, structural, or evolutionary relationships between the sequences
- Alignment procedure:
  - Insert *gaps* inside the sequences

**Unaligned sequences**

$s_1$: SKPKPIVAANWSLSELI
$s_2$: PKPIVAG
$s_3$: APPKFFVGGNWKMNGKRKSLG
$s_4$: APSRKFFVGGNW

**Aligned sequences**

$s'_1$: SK-PKPIVAANWSLSELI----
$s'_2$: ---PKPIVAG------------
$s'_3$: AP-PKFFVGGNWKMNGKRKSLG
$s'_4$: APSRKFFVGGNW----------

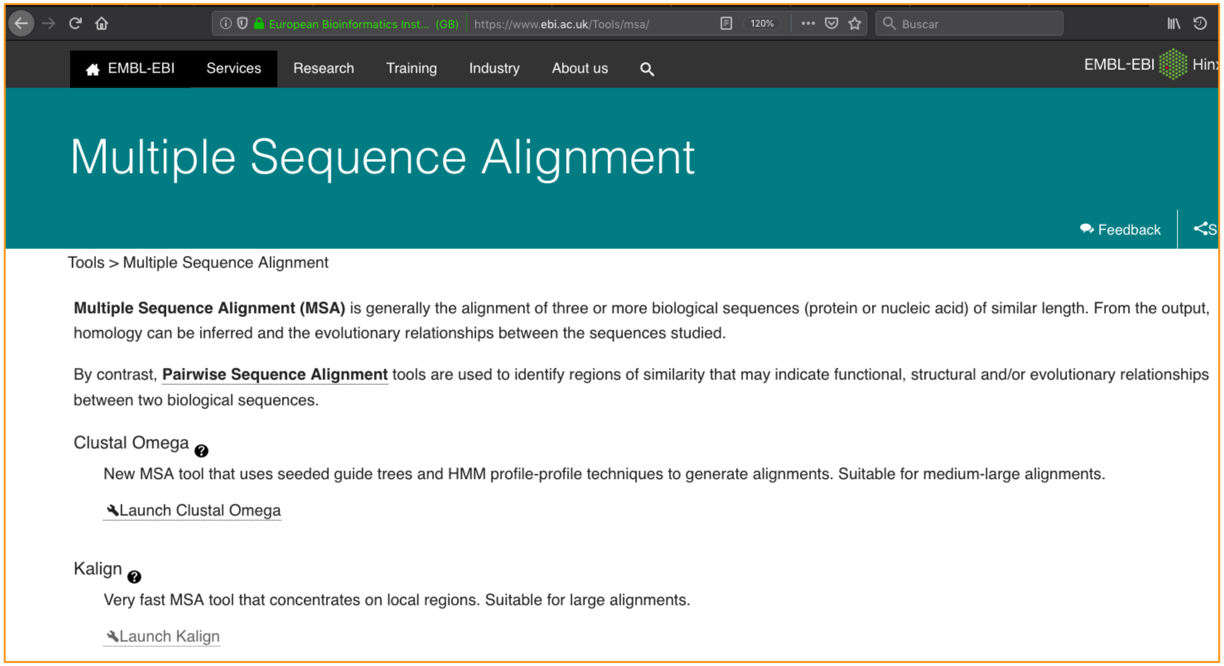
Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

# Computational complexity

- Finding the optimum of MSA problem has an NP-hard complexity
  - The computational requirements augments exponentially with the number of sequences and their length
  - The pairwise sequence alignment can be solved using exact techniques

# Tools for aligning MSAs

- Heuristic algorithms:
    - Clustal omega
    - MUSCLE
    - MAFFT
    - Kalign
    - T-Coffee



Multiple Sequence Alignment

Tools > Multiple Sequence Alignment

**Multiple Sequence Alignment (MSA)** is generally the alignment of three or more biological sequences (protein or nucleic acid) of similar length. From the output, homology can be inferred and the evolutionary relationships between the sequences studied.

By contrast, **Pairwise Sequence Alignment** tools are used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences.

Clustal Omega

New MSA tool that uses seeded guide trees and HMM profile-profile techniques to generate alignments. Suitable for medium-large alignments.

Launch Clustal Omega

Kalign

Very fast MSA tool that concentrates on local regions. Suitable for large alignments.

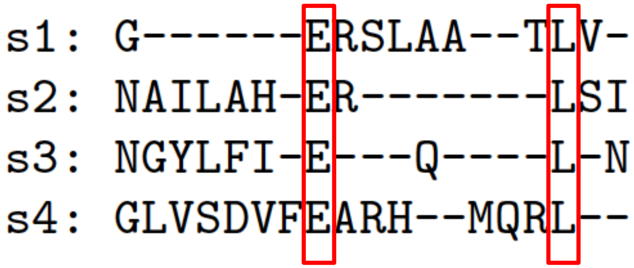
Launch Kalign

https://www.ebi.ac.uk/Tools/msa/

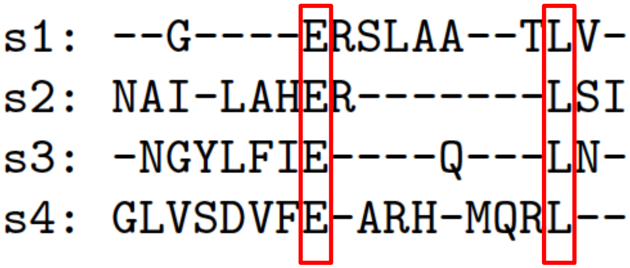
## How to score a MSA?

- Intuitively:
  - The larger the number of aligned columns the better
  - The shorter the number of gaps the better

- But this is not enough



```
s1: G------ERSLAA--TLV-
s2: NAILAH-ER-------LSI
s3: NGYLFI-E---Q----L-N
s4: GLVSDVFEARH--MQRL--
```



```
s1: --G----ERSLAA--TLV-
s2: NAI-LAHER-------LSI
s3: -NGYLFIE----Q---LN-
s4: GLVSDVFE-ARH-MQRL--
```

% Aligned columns: 10.526

% Non-gaps: 63.1579

Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

## How to score a MSA?

- Different types of methods:
  - From the scratch (assuming **independence between the columns**)
    - Percentage of totally conserved columns (TC), percentage of non-gaps (NonGaps), entropy (H)
  - By means of a **substitution matrix**
    - Sum of pairs (SP), weighted sum of pairs (wSP), star
  - Using **structural information** (e.g., tridimensional protein structure)
    - STRIKE (http://www.tcoffee.org/Projects/strike/)

# Percentage of totally conserved columns

- Count the number of (totally) conversed columns

$$TC(S) = 100 * \sum_{l=1}^{L} \frac{ConservedColumn(s_l)}{L}$$

$s_l$ residues in column $l$

```
col 123456789
    ATAATCG–G
    TTATIGGG–
    CCACFIG–R
    ACACGAG–G
    ATAWCGGTA
```

$$TC(S) = 100 * \frac{2}{9}$$
$$= 22{,}22\ \%$$

# Percentage of non-gaps

- Number of residues in regard to the number of gaps in the alignment

$$NonGap(S) = 100 * \sum_{i=1}^{k} \sum_{j=1}^{L} \frac{IsNonGap(s_{ij})}{k * L}$$

$s_{ij}$ residue in sequence $i$ in column $j$

$k$ number of sequences $L$ length of the alignment

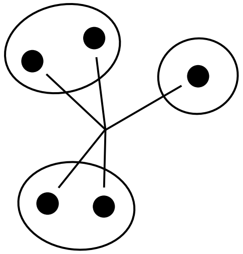
```
col 123456789
    ATAATCG-G
    TTATIGGG-
    CCACFIG-R
    ACACGAG-G
    ATAWCGGTA
```

$$NonGap(S) = 100 * \frac{3+1}{5*9}$$
$$= 8{,}88\ \%$$

Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

# Minimum entropy

- The entropy measures how diverses are the residues in a column



$$H(S) = \sum_{i=1}^{M} f_i^k \ln f_i^k$$

$f^k$ frequency of residue $k$

$M$ number of different residues

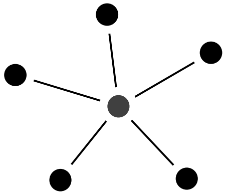
```
col 123456789
    ATAATCG–G
    TTATIGGG–
    CCACFIG–R
    ACACGAG–G
    ATAWCGGTA
```

$$H(S^4) = \overbrace{0.2\ln 0.2}^{\mathrm{A}=1/5} + \overbrace{0.2\ln 0.2}^{\mathrm{T}=1/5} + \overbrace{0.4\ln 0.4}^{\mathrm{C}=2/5} + \overbrace{0.2\ln 0.2}^{\mathrm{W}=1/5}$$
$$= -1.33$$

Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

# Star

- Considers the most repeated symbol



$$Star(S) = \sum_{i=1}^{k} \sum_{j=1}^{L} s(M_l, s_{ij})$$

$s_{ij}$ residue in sequence $i$ in column $j$

$M_l$ most repeated symbol in column $l$

|   | A | C | G | T |
|---|---|---|---|---|
| A | 5 | -4 | -4 | -4 |
| C | -4 | 5 | -4 | -4 |
| G | -4 | -4 | 5 | -4 |
| T | -4 | -4 | -4 | 5 |

Substitution matrix

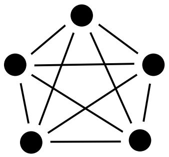
```
col 123456789
    ATAATCG–G
    TTATIGTG–
    CCGCFIG–R
    ACACGAC–G
    ATTWCGATA
```

$$\begin{aligned} Star(S^3) &= s(A,A) + s(A,A) + s(A,G) + \\ &\quad s(A,A) + s(A,T) \\ &= 7 \end{aligned}$$

Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

# Sum of pairs

- Scores each column according to a sum of pairs (SP) function
  - Requires a substitution scoring matriz (e.g., PAM250, BLOSUM62)



$$SP(S) = \sum_{i=1}^{L} \sum_{l=1}^{N-1} \sum_{j=l+1}^{N} ScoreMatrix(s_{il}, s_{ij})$$

$s_{ix}$ residue in sequence $i$ in column $x$

|   | A | C | G | T |
|---|---|---|---|---|
| A | 5 | -4 | -4 | -4 |
| C | -4 | 5 | -4 | -4 |
| G | -4 | -4 | 5 | -4 |
| T | -4 | -4 | -4 | 5 |

Substitution matrix

```
col 123456789
    GTASQLP-G
    GTASNIGTG
    PRSWFIG-R
```

A score is computed for each column, using substitution matrices and **gap penalties**

$SP(S^3) = s(A,A) + s(A,S) + s(A,S)$

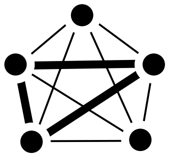
$SP(S^8) = s(-,T) + s(-,-) + s(T,-)$

The score is the sum of the column scores

$SP(S) = S_1 + S_2 + \cdots + S_9$

# Weighted sum of pairs

- Similar to SP, but applying weights



$$wSP(S) = \sum_{i=1}^{L} \sum_{l=1}^{N-1} \sum_{j=l+1}^{N} w_{lj} ScoreMatrix(s_{il}, s_{ij})$$

|   | A | C | G | T |
|---|---|---|---|---|
| **A** | 5 | -4 | -4 | -4 |
| **C** | -4 | 5 | -4 | -4 |
| **G** | -4 | -4 | 5 | -4 |
| **T** | -4 | -4 | -4 | 5 |

Substitution matrix

```
col 123456789
    ATAAT–CG–G w1=1
    TTATIGTG–G w2=1
    CCGCFIG–RG w3=1
    ACACGAC–GG w4=1
    ATTWCGATAG w5=1
```

$$\begin{aligned} wSP(S^3) &= w_1w_2s(A,A) + w_1w_3s(A,G) + \\ &\quad w_1w_4s(A,A) + w_1w_5s(A,T) + \\ &\quad w_2w_3s(A,G) + w_2w_4s(A,A) + \\ &\quad w_2w_5s(A,T) + w_3w_4s(G,A) + \\ &\quad w_3w_5s(G,T) + w_4w_5s(G,A) \\ &= -5 \end{aligned}$$

Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

# Installing the package

- To download pyMSA 1.0.0, just clone the Git repository hosted in GitHub:

```
$ git clone https://github.com/benhid/pyMSA.git
$ python setup.py install
```

- Alternatively, you can install it with *pip*:

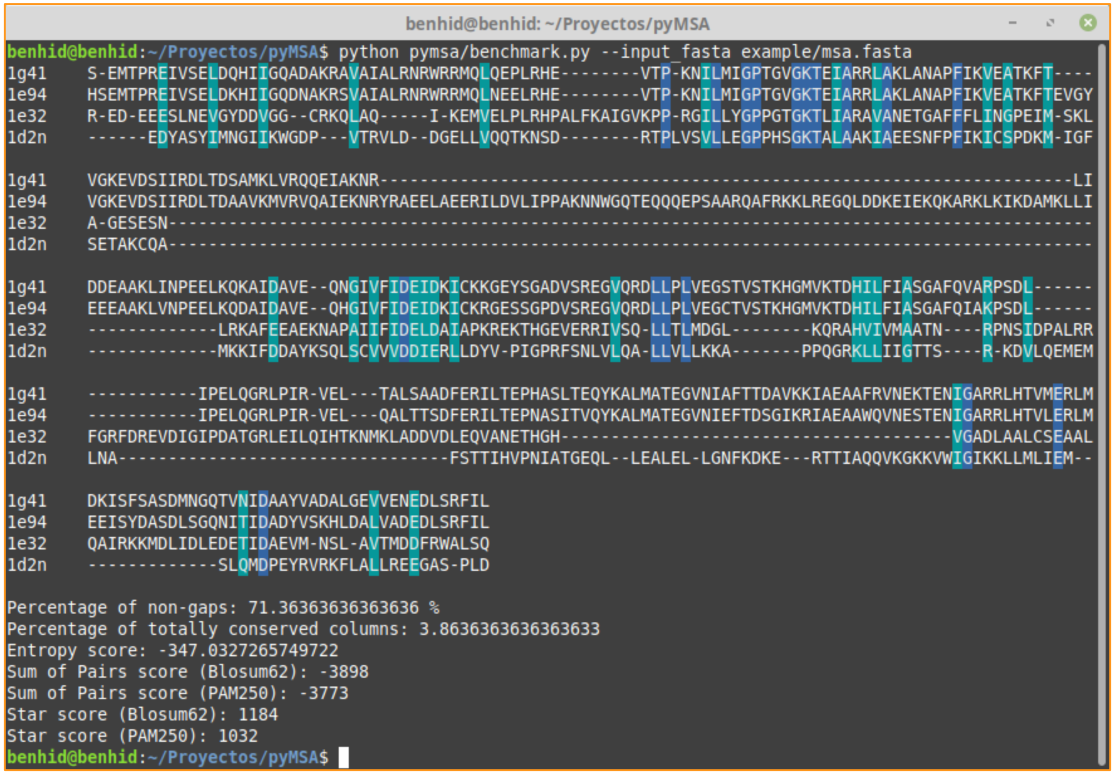
```
$ pip install pyMSA
```

Antonio J. Nebro
PhD in Computer Science
ajnebro@uma.es

# How to score an alignment

- Some examples are located in the *examples* folder
- …or run a full benchmark against a file:

```
$ python pymsa/benchmark.py --input_fasta ~/msa.txt
```



```
benhid@benhid:~/Proyectos/pyMSA$ python pymsa/benchmark.py --input_fasta example/msa.fasta
1g41     S-EMTPREIVSELDQHIIGQADAKRAVAIALRNRWRRMQLQEPLRHE--------VTP-KNILMIGPTGVGKTEIARRLAKLANAPFIKVEATKFT----
1e94     HSEMTPREIVSELDKHIIGQDNAKRSVAIALRNRWRRMQLNEELRHE--------VTP-KNILMIGPTGVGKTEIARRLAKLANAPFIKVEATKFTEVGY
1e32     R-ED-EEESLNEVGYDDVGG--CRKQLAQ-----I-KEMVELPLRHPALFKAIGVKPP-RGILLYGPPGTGKTLIARAVANETGAFFFLINGPEIM-SKL
1d2n     ------EDYASYIMNGIIKWGDP---VTRVLD--DGELLVQQTKNSD--------RTPLVSVLLEGPPHSGKTALAAKIAEESNFPFIKICSPDKM-IGF

1g41     VGKEVDSIIRDLTDSAMKLVRQQEIAKNR---------------------------------------------------------------------LI
1e94     VGKEVDSIIRDLTDAAVKMVRVQAIEKNRYRAEELAEERILDVLIPPAKNNWGQTEQQQEPSAARQAFRKKLREGQLDDKEIEKQKARKLKIKDAMKLLI
1e32     A-GESESN--------------------------------------------------------------------------------------------
1d2n     SETAKCQA--------------------------------------------------------------------------------------------

1g41     DDEAAKLINPEELKQKAIDAVE--QNGIVFIDEIDKICKKGEYSGADVSREGVQRDLLPLVEGSTVSTKHGMVKTDHILFIASGAFQVARPSDL------
1e94     EEEAAKLVNPEELKQDAIDAVE--QHGIVFIDEIDKICRRGESSGPDVSREGVQRDLLPLVEGCTVSTKHGMVKTDHILFIASGAFQIAKPSDL------
1e32     -------------LRKAFEEAEKNAPAIIFIDELDAIAPKREKTHGEVERRIVSQ-LLTLMDGL--------KQRAHVIVMAATN----RPNSIDPALRR
1d2n     -------------MKKIFDDAYKSQLSCVVVDDIERLLDYV-PIGPRFSNLVLQA-LLVLLKKA-------PPQGRKLLIIGTTS----R-KDVLQEMEM

1g41     -----------IPELQGRLPIR-VEL---TALSAADFERILTEPHASLTEQYKALMATEGVNIAFTTDAVKKIAEAAFRVNEKTENIGARRLHTVMERLM
1e94     -----------IPELQGRLPIR-VEL---QALTTSDFERILTEPNASITVQYKALMATEGVNIEFTDSGIKRIAEAAWQVNESTENIGARRLHTVLERLM
1e32     FGRFDREVDIGIPDATGRLEILQIHTKNMKLADDVDLEQVANETHGH---------------------------------------VGADLAALCSEAAL
1d2n     LNA---------------------------------FSTTIHVPNIATGEQL--LEALEL-LGNFKDKE---RTTIAQQVKGKKVWIGIKKLLMLIEM--

1g41     DKISFSASDMNGQTVNIDAAYVADALGEVVENEDLSRFIL
1e94     EEISYDASDLSGQNITIDADYVSKHLDALVADEDLSRFIL
1e32     QAIRKKMDLIDLEDETIDAEVM-NSL-AVTMDDFRWALSQ
1d2n     -------------SLQMDPEYRVRKFLALLREEGAS-PLD

Percentage of non-gaps: 71.36363636363636 %
Percentage of totally conserved columns: 3.8636363636363633
Entropy score: -347.0327265749722
Sum of Pairs score (Blosum62): -3898
Sum of Pairs score (PAM250): -3773
Star score (Blosum62): 1184
Star score (PAM250): 1032
benhid@benhid:~/Proyectos/pyMSA$
```