

Introducción a la Filoinformática: Pan-genómica y Filogenómica –  
Diplomado en Bioinformática, Univ. Autónoma de Sinaloa, Oct. 2021  
[http://econtinua.uas.edu.mx/diplomados/2510-2700-18\\_001.htm](http://econtinua.uas.edu.mx/diplomados/2510-2700-18_001.htm)

Pablo Vinuesa (vinuesa[at]ccg . unam . mx; @pvinmex )  
Centro de Ciencias Genómicas, CCG-UNAM, Cuernavaca, México  
<http://www.ccg.unam.mx/~vinuesa/>

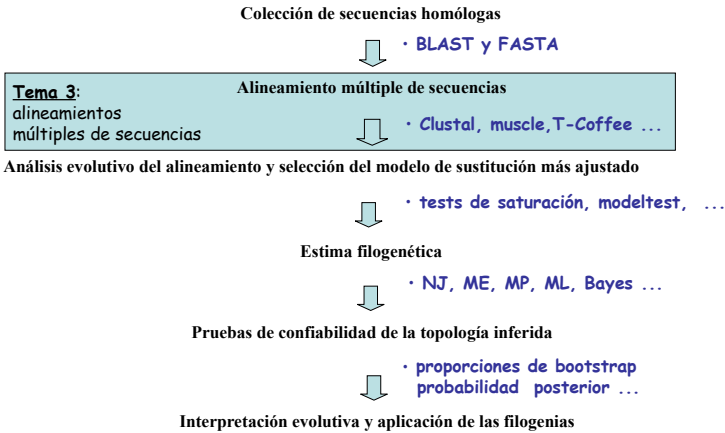


Todo el material del curso (presentaciones, tutoriales y datos) lo encontrarás en:  
[https://github.com/vinuesa/intro-filoinfo\\_UAS](https://github.com/vinuesa/intro-filoinfo_UAS)

• Tema 3 Alineamientos múltiples

1. Alineamientos múltiples y el problema de las repeticiones, sustituciones e indeles
2. Alineamientos múltiples progresivos usando programas de la familia Clustal
3. Formatos de secuencia
4. Alineamiento de secuencias codificadoras de proteínas usando RevTrans
5. Alineamiento de genes ribosomales usando RDP-II y GreenGenes
6. Aln. múltiples usando clustal-omega (clustalo)

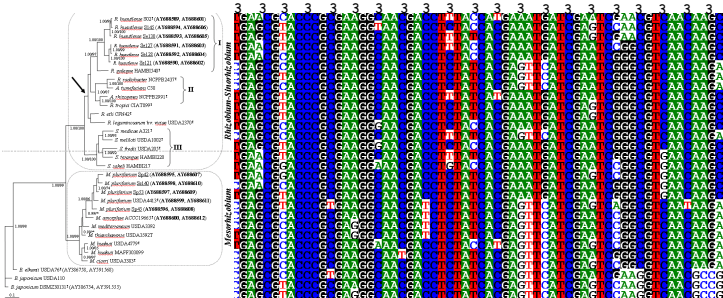
Protocolo básico para un análisis filogenético de  
secuencias moleculares



• Cualquier estudio de filogenético o de evolución molecular basado en secuencias necesita de un **alineamiento múltiple para determinar las correspondencias de homología a nivel de los residuos individuales o caracteres**.

• **El alineamiento representa una hipótesis sobre la homología de los caracteres**

• La mejor manera de representar estas homologías entre caract. es escribiendo los residuos homólogos en columnas, generándose una matriz de m x n (secs. x posic) residuos, en la que **cada columna contiene a residuos o caracteres homólogos**

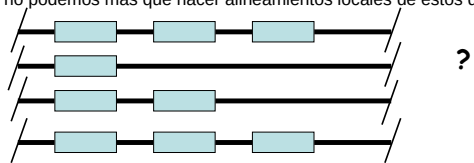


• Comparar los aln. múltiples en el contexto de una filogenia nos puede revelar mucho acerca de los patrones y tasas de sustitución, haciendo posible la identificación de estados de caracter ancestrales y derivados, etc.

Generación de alineamientos múltiples – consideraciones generales

• El problema de las repeticiones

Muchas **proteínas multidominio** pueden presentar diverso grado de **repetición de dominios** particulares. Puede llegar a ser muy complejo o imposible hacer el alineamiento global de las proteínas si difieren en el número y orientación de estas regiones repetidas. A veces no podemos más que hacer alineamientos locales de estos dominios.

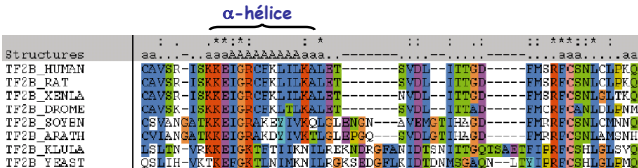


A nivel de DNA se dan también regiones repetidas, muchas veces involucrando a unos poco nts. como es el caso de los **microsatélites y otras regiones repetidas**. Con frecuencia estas regiones son imposibles de alinear objetivamente. Suelen acumularse en regiones no codificantes del genoma, incluyendo intrones, o en regiones codificantes hipervariables como espaciadores intergénicos transcritos o regiones reguladoras (UTRs).

Este tipo de "repeats" cortos son poco frecuentes a nivel de aminoácidos, si bien a este nivel es común encontrar regiones o **dominios "de gran escala" repetidos**. Un ejemplo clásico de este fenómeno son las calmodulinas.

Generación de alineamientos múltiples – consideraciones generales

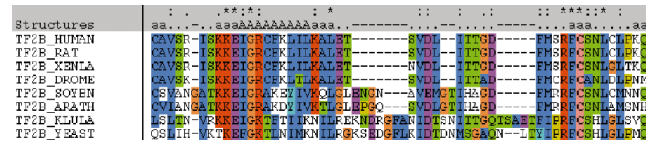
- El problema de las sustituciones
- Al examinar alns. múltiples de proteínas se observan dos patrones de sustitución:
  - Existen bloques de 5 a 20 residuos con alto nivel de identidad y similitud dispersos entre regiones de menor similitud. Estos bloques corresponden típicamente a elementos estructurales como **α-hélices y pliegues β** que evolucionan más lentamente que los loops o bucles que los interconectan



- Las **columnas alineadas** con múltiples estados de carácter **tienden a presentar residuos de características bioquímicas similares** (I, A, V, L; S, T; R, K; etc.). Esta conservación de residuos similares es particularmente patente en los bloques correspondientes a elementos de estructura secundaria, sitios activos o de unión a ligandos. La propiedad bioquímica más conservada es la de polaridad/hidrofobicidad.

Generación de alineamientos múltiples – consideraciones generales

- El problema de los indeles (inserciones/deleciones)
- Cuando por eventos de inserción o deleción (**indeles**) las secuencias homólogas presentan distintas longitudes, es necesario introducir **"gaps"** en el alineamiento para mantener la correspondencia entre sitios homólogos situados antes y después de las regiones afectadas por indeles. Estas regiones se identifican mediante guiones (-).

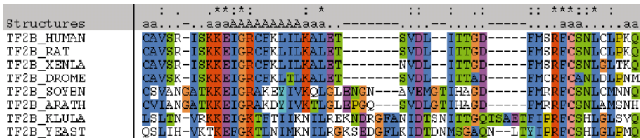


Los indeles no se distribuyen aleatoriamente en las secuencias codificadoras.

- Casi siempre aparecen **ubicados entre dominios funcionales o estructurales**, preferentemente en bucles (loops) que conectan a dichos dominios. Esto vale tanto para RNAs estructurales (tRNAs y rRNAs) como para proteínas. No suelen interrumpir el marco de lectura.
- Generalmente se usan **sistemas de penalización de gaps afines**:  $GP = gap + e(L - 1)$  en el cálculo del score de un alineamiento múltiple  
[ gap = costo apertura de gap; e = costo extensión del indel; L longitud del indel ]

Generación de alineamientos múltiples – consideraciones generales

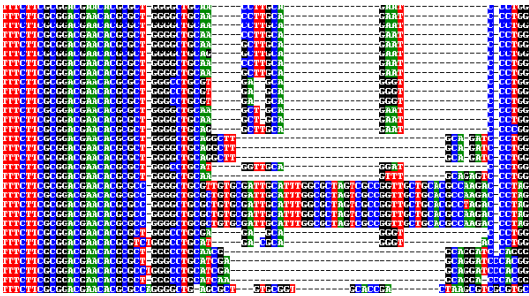
- El problema de las sustituciones
- Es importante recordar que por debajo del **20% de identidad** a nivel de sec. de AA es ya imposible que se pueda obtener un alineamiento múltiple (o pareado) confiable si nos basamos para obtenerlo sólo en la secuencia primaria, ya que entramos en la **zona de penumbra** (saturación mutacional)



- Un par de secuencias de nts al azar presentarán en promedio un 25 % de identidad.
- Por tanto, siempre que sea posible, hay que realizar los alineamientos múltiples en base a las secuencias traducidas, es decir, sobre AAs (igual que al hacer búsquedas en bases de datos de secuencia), que son mucho más informativos (20 caracteres vs. 4) y con tasa evolutiva mucho menor.

Generación de alineamientos múltiples – consideraciones generales

- A mayor **distancia genética** (evolutiva) entre un par de secuencias, mayor será el número de mutaciones acumuladas. Dependiendo del tiempo de separación de los linajes y la tasa evolutiva del locus, puede llegar a ser imposible alinear ciertas regiones debido a fenómenos de **saturación mutacional o acúmulo de indeles**. En loci de evolución muy rápida como intrones o espaciadores intergénicos, los fenómenos de saturación mutacional se observan incluso cuando se comparan secuencias de organismos evolutivamente próximos (mismo género o familia).



**¡Las regiones de homología dudosa deben de ser excluidas de un análisis filogenético!**  
**Debemos de maximizar a toda costa la relación entre señal/ruido**

Alineamientos múltiples - algoritmos

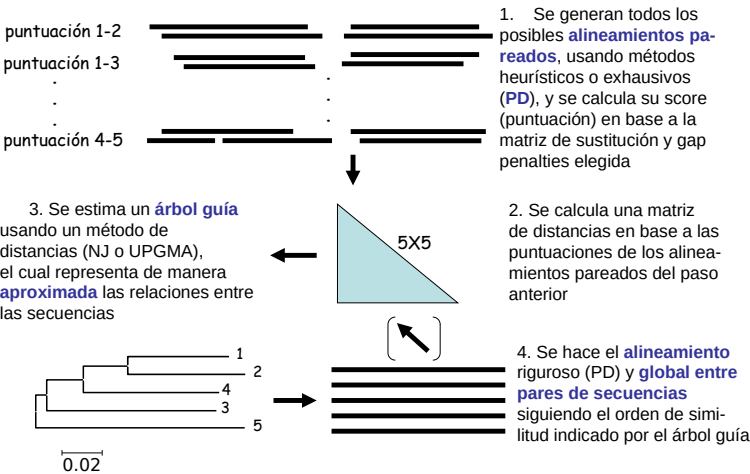
Existen diversas estrategias computacionales para obtener alineamientos múltiples de manera (semi)automática para conjuntos grandes (cientos - miles) de secuencias.

1.- Implementación de algoritmos de alineamiento progresivo.

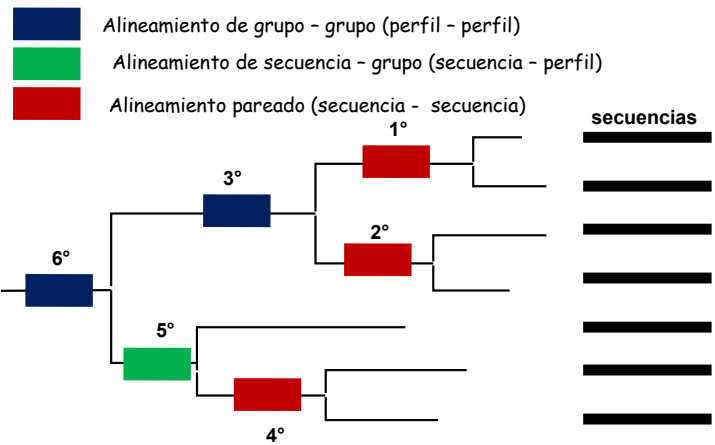
Así como los alns. múltiples son indispensables para reconstruir filogenias a partir de secs, un árbol de relaciones filogenéticas representa información muy valiosa para guiar la generación de un aln. múltiple.

La mayor parte de los alineadores automáticos modernos se basan en este tipo de algoritmos. Construyen un **árbol guía** aproximado a partir de distancias calculadas entre todos los pares posibles de secuencias. De la matriz de distancias resultantes se construye un árbol usando un método algorítmico (**NJ o UPGMA**). El árbol guía resultante se emplea para **construir el alineamiento de manera progresiva**. Las dos secuencias más similares se alinean primero usando PD y una matriz o esquema de ponderación particular. Una vez alineado el primer par, los gaps generados ya no se mueven. Este par es tratado como una sola secuencia y es alineada contra la siguiente secuencia o grupo de secuencias más próximas en el árbol. Se repite el proceso hasta que todas las secs. están alineadas. El proceso es suficientemente rápido como para alinear varios cientos de secuencias. Son menos precisos que los métodos basados en la WSPs, pero muchísimo más rápidos.

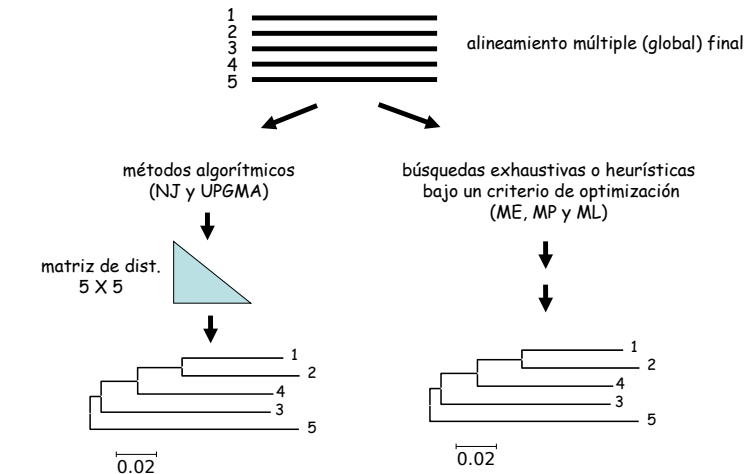
Pasos en la generación de un alineamiento múltiple siguiendo la estrategia de alineamiento progresivo



Pasos en la generación de un alineamiento múltiple siguiendo la estrategia de alineamiento progresivo con un árbol guía



Pasos en la generación de un alineamiento múltiple siguiendo la estrategia de alineamiento progresivo - y su uso para estimar una filogenia





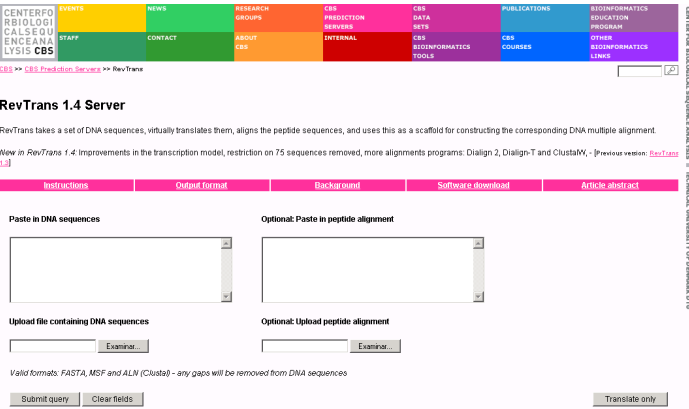


Alineamientos múltiples progresivos usando Clustal  
-un ejemplo: alineamiento de GDPs dependientes de NAD



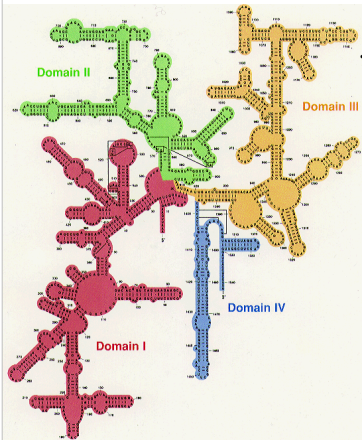
Servidores para alinear nts. en base a un alineamiento de proteínas

<http://www.cbs.dtu.dk/services/RevTrans/>



Servidores para alinear secuencias de rRNAs o rDNAs

- Los genes ribosomales representan un problema muy particular en el contexto de alineamientos múltiples. Deben de guiarse usando máscaras de información estructural.



- Servidores como [GreenGenes](#) y [RDP-II](#) proveen herramientas muy útiles en este contexto. Si quieres ver unos tutoriales sobre el uso de estos servidores, visita mi sitio web y busca bajo phylogeny tutorials:  
[http://www.ccg.unam.mx/~vinuesa/Using\\_the\\_GreenGenes\\_and\\_RDPII\\_servers.html](http://www.ccg.unam.mx/~vinuesa/Using_the_GreenGenes_and_RDPII_servers.html)

Formatos de secuencias  
1) FASTA

- Existen una gran cantidad de estilos o formatos de presentación de secuencias. Muchos programas de análisis filogenético usan su propio formato (Phylip, Nexus, Mega ...)
- El formato más sencillo es el **FASTA**, en el que cada secuencia se identifica mediante un renglón descriptor que comienza con > en el siguiente renglón comienza la secuencia

```
>lcl|1 R._galegae
CCGCTGGTCACCTCCGGCAAGCGGCCATCCACCAGGAAGCGCCTTCCTA
CGTCGATCAGTCGACCGAAGGCCAGATCCTGGTCACCGGCATCAAGGTCG

> lcl|2 M._plurifarum
CCGGTCGACGCCGTCGAGCTGCCGTGCCATCCACCAGCGGCTCCGGCCTA
TGTCGACCAGTCGACGGAAGCGCAGATCCTGGTTACCGGCATCAAGGTTCT

> lcl|3 B._japonicum
CCGGTCAAGTCGGAAGGCCTGCCGCGCCATCCACCAGGAAGCGCCGACCTA
CACCAGCAGTCCACCGAAGCTGAAATTCTCGTCACCGGCATCAAGGTCG
```

Formatos de secuencias  
II) PHYLIP

- **Phylip (interleaved):** no. seqs, no. caracteres  
nombre secuencias (máx 10 caracteres) espacio, secuencia ...

```
3      100
R._galegae CCGCUGGUCA CCUCCGGCAA GCGCGCCAUC CACCAGGAAG CGCCUUCUA
M._plurifa ..G.C.A.G ..GU..AGCU ...U..... ..CCG. .U..GG...
B._japonic ...G.CAAGU .GGAA...CU ..... ..GA....

      CGUCGAUCAG UCGACCGAAG GCCAGAUCCU GGUCACCGGC AUCAAGGUCC
U.....C... ..G.... CG..... ..U..... ..UC
.AC...C... ..C..... CUG.A..U.. C.....
```

- **Phylip (sequential or non-interleaved)**

```
3 100
R._galegae CCGCTGGTCA CCTCCGGCAA GCGCGCCATC CACCAGGAAG CGCCTTCCTA
CGTCGATCAG TCGACCGAAG GCCAGATCCT GGTACCCGGC ATCAAGGTCG
M._plurifa CCGGTGACAG CCGTCGAGCT GCGTGCCATC CACCAGCCGG CTCCGGCCTA
TGTCGACCAAG TCGACGGGAAG GCGAGATCCT GGTACCAGGC ATCAAGGTTT
B._japonic CCGGTCAAGT CGGAAGGCCT GCGCGCCATC CACCAGGAAG CGCCGACCTA
CACCGACCAG TCCACCGAAG CTGAAATCTT CGTCACCGGC ATCAAGGTCG
```

Formatos de secuencias:  
su interconversión

- Cuando preparamos un fichero con nuestras propias secuencias generalmente lo más adecuado es hacerlo en formato FASTA
- Si necesitamos pasarlo a otro formato, una buena posibilidad es hacerlo con **ReadSeq**

<http://iubio.bio.indiana.edu/cgi-bin/readseq.cgi>

ReadSeq reconoce automáticamente el formato de entrada y si se trata de aas o nts

- Otra alternativa es escribir un sencillo *script* de **Perl** que haga uso de los objetos y métodos del módulo Bio::AlignIO de **BioPerl** (<http://www.bioperl.org>) para interconvertir Formatos ... veremos un ejemplo más adelante.

Formatos de secuencias  
III) NEXUS

```
#NEXUS
[OJO!!!, no usar guiones- (reservado para gaps!), sólo guiones bajos_]

BEGIN TAXA;                [taxa block]
DIMENSIONS NTAX=3;
TAXLABELS
R._galegae;
M._plurifarum;
B._japonicum;
END;

BEGIN CHARACTERS;          [character block]
DIMENSIONS NCHAR=100;
FORMAT DATATYPE=DNA MISSING=? GAP=- MATCHCHAR=. INTERLEAVE=yes ;
MATRIX
[
      10      20      30      40      50]
[
R._galegae      CCGCTGGTCACCTCCGGCAAGCGCGCCATCCACCAGGAAGCGCCTTCCTA
M._plurifarum   ...G.C.A.G..GT..AGCT...T.....CCG..T..GG...
B._japonicum    ...G.CAAGT.GGAA...CT.....GA....

      60      70      80      90      100]
[
R._galegae      CGTCGATCAGTCGACCGAAGGCCAGATCCTGGTCACCGGCATCAAGGTCG
M._plurifarum   T....C.....G....CG.....T.....TC
B._japonicum    .AC...C.....C....CTG.A..T..C.....TC
;
END;
```



Clustal Omega

"The last alignment program you'll ever need"

[Home](#) [Webservers](#) [Download](#) [Documentation](#) [Contact](#) [News](#)

Introduction

Clustal Omega is the latest addition to the Clustal family. It offers a significant increase in scalability over previous versions, allowing hundreds of thousands of sequences to be aligned in only a few hours. It will also make use of multiple processors, where present. In addition, the quality of alignments is superior to previous versions, as measured by a range of popular benchmarks.

Please note that Clustal Omega is currently a command line-only tool.

A full description of the algorithms used by Clustal Omega is available in the Molecular Systems Biology paper [Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega](#). Latest additions to Clustal Omega are described in [Clustal Omega for making accurate alignments of many protein sciences](#)

Webservers

Clustal Omega can be run online at the following websites (this list will expand in the coming months).

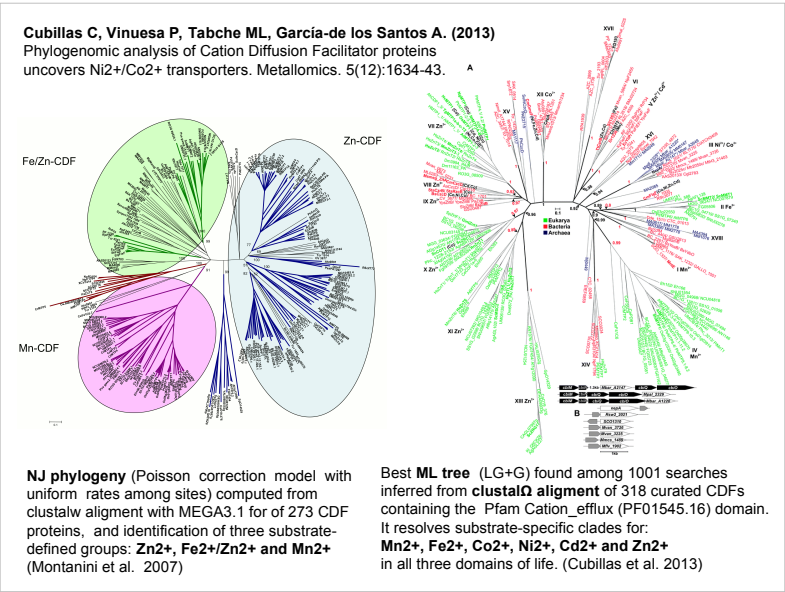
Download Clustal Omega


Source code

- [Source code .tar.gz \(1.2.4\)](#)


Citing Clustal

1. Sievers F, Wilm A, Dineen DG, Gibson TJ, Karplus K, Li W, Lopez R, McWilliam H, Remmert M, Söding J, Thompson JD, Higgins DG (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Molecular Systems Biology 7:539 doi:10.1038/msb.2011.75
2. Sievers F, Higgins DG (2018) Clustal Omega for making accurate alignments of many protein sciences. Protein Sci 27:135-145





**Clustal Omega**  
"The last alignment program you'll ever need"



**clustalo --help**  
Clustal Omega - 1.2.4 (AndreaGiacomo)

Check <http://www.clustal.org> for more information and updates.

- Usage:  
clustalo [-hv] [-i <file>.-] [-hmm-in=<file>]... [-hmm-batch=<file>] [-dealgn] [-profile1=<file>] [-profile2=<file>] [-is-profile] [-t (Protein, RNA, DNA)] [-infmt={a2m=fa[sta],clu[stal],msf,phy[lip],selex,st[ockholm],vie[nna]}] [-distmat-in=<file>] [-distmat-out=<file>] [-guidetree-in=<file>] [-guidetree-out=<file>] [-pileup] [-full] [-full-iter] [-cluster-size=<n>] [-clustering-out=<file>] [-trans=<n>] [-posterior-out=<file>] [-use-kimura] [-percent-id]

[-o <file>.-] [-outfmt={a2m=fa[sta],clu[stal],msf,phy[lip],selex,st[ockholm],vie[nna]}] [-residuenumber] [-wrap=<n>] [-output-order={input-order,tree-order}]

[-iterations=<n>] [-max-guidetree-iterations=<n>] [-max-hmm-iterations=<n>]

[-maxnumseq=<n>] [-maxseqlen=<|>] [-auto]

[-threads=<n>] [-pseudo=<file>] [-l <file>]

[-version] [-long-version] [-force] [-MAC-RAM=<n>]

- A typical invocation would be:

```
* clustalo -i my-in-seqs.fa -o my-out-seqs.fa
* for f in *.fa; do clustalo -i $f -o ${f%.*}_cluoAln.fa; done
```