Multiple Alignments and Weblogo

Nicolas Schmelling

Abstract

This protocol show you how to create a sequence alignment with multiple sequences using CLUSTAL Omega and to futher modify it with Jalview and visualize it with Weblogo.

Citation: Nicolas Schmelling Multiple Alignments and Weblogo. **protocols.io** https://www.protocols.io/view/multiple-alignments-and-weblogo-gscbwaw
Published: 19 Dec 2016

Protocol

Preparation

Step 1.

Before you can start with the creation of multiple sequence alignments you need to have a FASTA file containing multiple full length sequences. One way to create such a FASTA file is using BLAST. Attached is a protocol for a reciprocal best hit BLAST that creates the FASTA files you need. The Python script csv_*to_fasta.py* will create FASTA files from the CSV files that are normally produced by the attached protocol.

PROTOCOL

. Reciprocal Best Hit BLAST

CONTACT: Nicolas Schmelling
cmd COMMAND

```
python csv_to_fasta.py
```

Download sequences from NCBI

Step 1.1.

The first thing you need to do is to download the sequences of choice from the NCBI server. You need to decide whether you need the nucleotide or protein sequences. I recommend only downloading sequences from genomes that are labeled as *Complete* or *Chromosome* level to ensure a certain quality of the genome assembly.

The following commands will download all protein sequences from genome assemblies labeled as *Complete* or *Chromosome*, futher merge them into a single sequence FASTA file, and move them into the newly created *db* directory.

cmd COMMAND

```
mkdir PROJECT_NAME
cd PROJECT_NAME

wget ftp://ftp.ncbi.nih.gov/genomes/refseq/assembly_summary_refseq.txt

awk -
F '\t' '{if($12=="Complete Genome") print $20}' assembly_summary_refseq.txt > assembly_s
ummary_complete_genomes.txt
awk -
F '\t' '{if($12=="Chromosome") print $20}' assembly_summary_refseq.txt > assembly_summar
y_chromosome.txt

mkdir RefSeqCompleteGenomes
mkdir RefSeqCompleteReports
mkdir RefSeqChromosomeGenomes
mkdir RefSeqChromosomeReports
mkdir db

for next in $(cat assembly_summary_complete_genomes.txt);
do
wget -P RefSeqCompleteGenomes "$next"/*protein.faa.gz;
wget -P RefSeqCompleteReports "$next"/*assembly_report.txt;
done

for next in $(cat assembly_summary_chromosome.txt);
do
wget -P RefSeqChromosomeGenomes "$next"/*protein.faa.gz;
wget -P RefSeqChromosomeReports "$next"/*assembly_report.txt;
done

gunzip RefSeqCompleteGenomes/*.gz
gunzip RefSeqChromosomeGenomes/*.gz

python change_fasta_header.py RefSeqCompleteGenomes
python change_fasta_header.py RefSeqChromosomeGenomes

cat RefSeqCompleteGenomes/*.fasta > db/all_complete_genomes.fasta
cat RefSeqChromosomeGenomes/*.fasta > db/all_chromosomes.fasta

cat db/all_complete_genome.fasta db/all_chromosome.fasta > db/all_genomes.fasta
```

Create database from sequences

Step 1.2.

After downloading the sequences of choice from NCBI, you need to create a database. Therefore you change the directory into the *db* directory where all the sequences are stored. Next you run the second command. This will create a protein database with all the downloaded sequences. If you downloaded nucleotide sequence make sure you change the *dbtype* option to *nucleotide*.

Parameters

in: FASTA file including all sequences for you database

out: Database name

dbtype: Database type usually 'prot' for proteins or 'nucleotide' for nucleotides

cmd COMMAND

```
cd db
makeblastdb -in FILE_NAME.fasta -dbtype 'TYPE' -out DATABASE_NAME
```

Select sequences as queries to search for homologs in database

Step 1.3.

After downloading the sequences and creating the database you want to start with your first BLAST run. To do that you need to place your sequences of choice in the directory *seq*. Use a single FASTA file for each query sequence. The following commands will loop over the query sequences and serach for homologs in your database. The program used here is BLASTP, which is used for protein sequences. Other BLAST programs can be found in the documentation.

Parameters

query: Input sequence

db: BLAST database

out: Name of the output file

outfmt: Output format

evalue: Arbitrary cut-off of sequence similarity. The e-value depends on your database size, so the large your database the smaller your e-value can be. I recommend something between 10^{-5} and 10^{-20}.

word_size: Number of nucleotides/amino acids, which resembles the smallest unit of your query

num_alignments: Maximum number of alignment partner in the database for a single query sequence

cmd COMMAND

```
mkdir seq
# Add sequence FASTA files in seq directory

for f in seq/*.fasta
do
blastp -query "$f" -db DATABASE_NAME -out "${f%.fasta}_blast.xml" -outfmt 5 -
evalue NUMBER -word_size NUMBER -num_alignments NUMBER
done
```

Align hits against original genome

Step 1.4.

Now that you have FASTA files for the hits you can start the second BLAST run and align the hits back to the original genome from the query sequence. You need to create a database from the

genomic sequences first and then start your BLAST run. This time you set the *num_alignments* parameter to 1, because you only want to record the best alignment partner in the genome. You don't have to worry about the *evalue* since the best hit will most likely be 0 or in close proximity.

cmd COMMAND

```
makeblastdb -in GENOME_FILE.fasta -dbtype 'TYPE' -out DATABASE_NAME

for f in seq/*_matches.fasta
do
blastp -query "$f" -db DATABASE_NAME -out "${f%_matches.fasta}_back_blast.xml" -
outfmt 5 -word_size NUMBER  -num_alignments 1
done
```

Create new FASTA files for BLAST hits

Step 1.5.

You now created your first BLAST results. For a reciprocal BLAST you need to align these hits back to the original genome from your query sequence. For this you need to create FASTA files from your BLAST results. The Python script *parse_hits.py* will do this for you. You just need to run the following command.

cmd COMMAND

```
python parse_hits.py
```

Filter hits and create CSV files for each query sequence

Step 1.6.

In the last step you need to filter those hits that aligned back to the original query sequence. These are the only valid hits following the standard procedure for a reciprocal best hit BLAST. To filter these hits you need to run the Python script *hits_to_csv.py.*

cmd COMMAND

```
python hits_to_csv.py
```

Sequence alignment with CLUSTAL

Step 2.

Once you have the FASTA file with multiple sequences you can start aligning them. The following command will create such an alignment. You need to specify the *input* and *output* file name, the *seqtype* (whether it is protein or nucleotide), and the number of iterations. The output FASTA file can directly be used for the creation for Weblogos or the optional modification with Jalview.

Parameter

in: Input file name.

infmt: Input format. Recommended: FASTA

seqtype: Type of sequences in your input file (protein or nucleotide).

out: Output file name

outfmt: Output format. Recommended: FASTA

dealign: Dealignes the sequences before the actual alignment.

iter: Number of iterations, i.e. number of sequence alignments that are produced to calculate the average alignment

max-guidetree-iteration: Number of iterations to create the guidetree.

cmd COMMAND

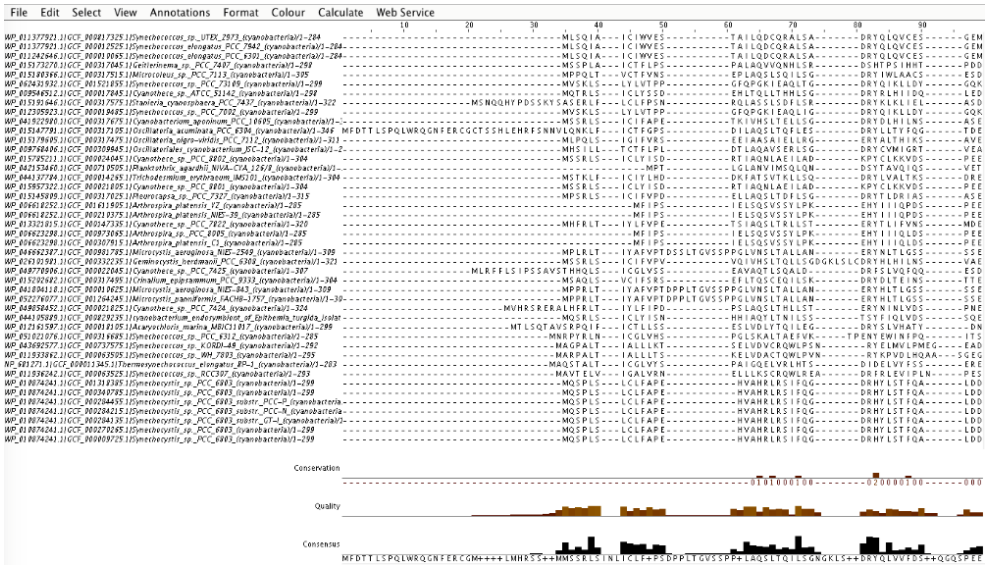
```
clustalo --in INPUT_FILE --infmt fasta --seqtype TYPE --dealign --out OUTPUT_FILE --
outfmt fasta --iter NUMBER --max-guidetree-iterations NUMBER --force
```

Modify with Jalview (Optional)

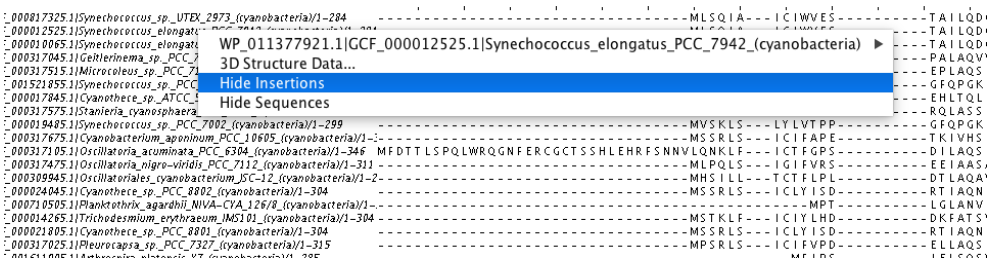
Step 3.

You can use Jalview to analyze your alignment with graphical visualization. Jalview provides also a convenient way to slice alignments and remove gaps for a single reference sequence. The following instructions describes how to do it.

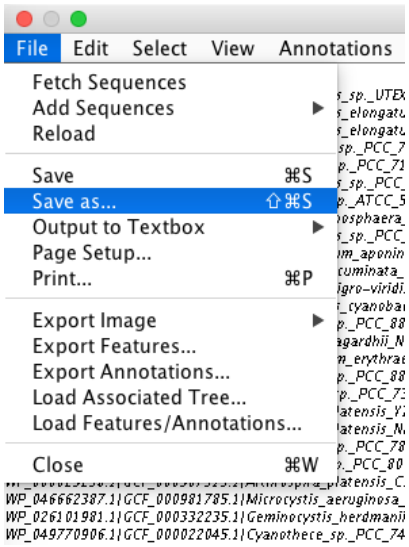
1. Open the alignment FASTA file in Jalview.



2. Choose a reference sequence, right click on it and select *Hide Insertion*.



3. Save alignment under a new name.

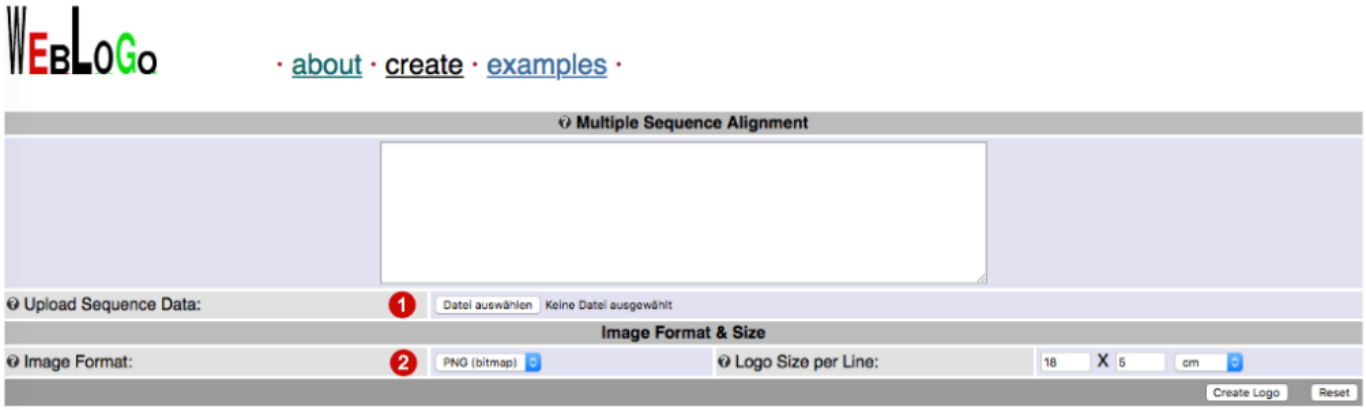


LINK: http://www.jalview.org

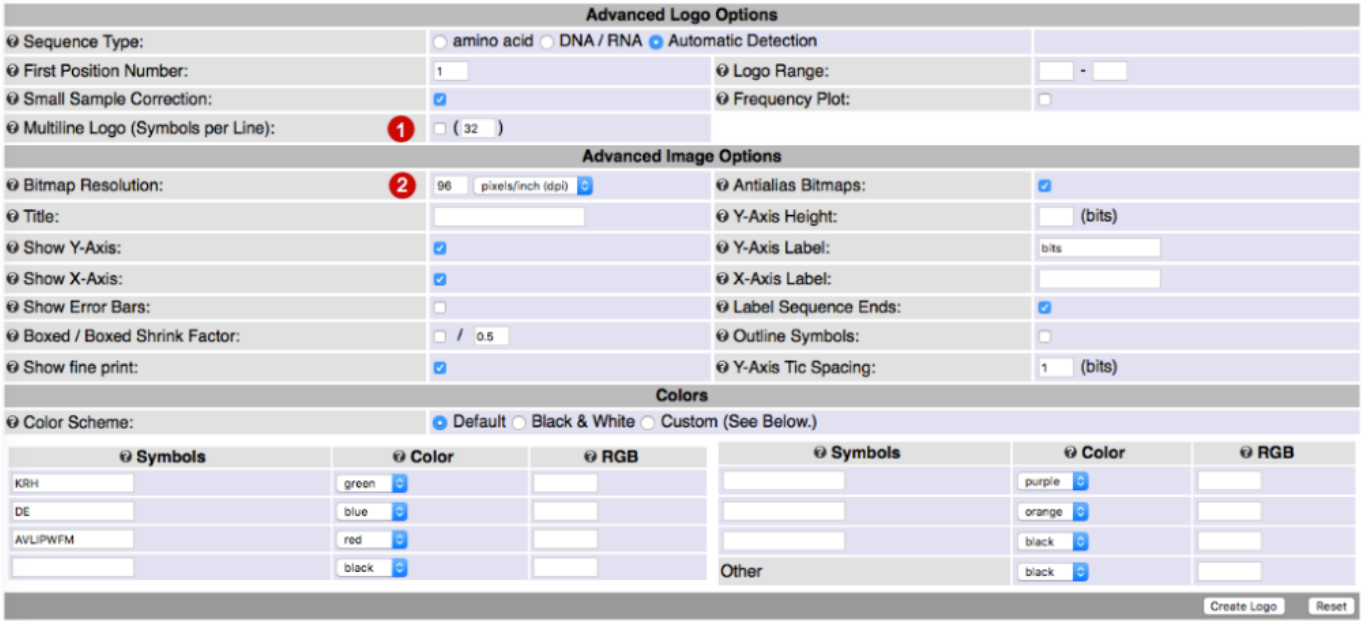
Visualize with Weblogo

Step 4.

In the last step you want to create a summary visualization of your alignment. Weblogos are a great way to do so, because they create a conservation score for each position. You can use your original alignment or use the modified alignment that you created with Jalview



1. Upload FASTA file from computer.
2. Choose output format. Vector graphic format is recommended.



1. If you produce multiple alignments, it is recommended to set a fixed line length.
2. If you choose a non-vector graphic format, it is recommended to set a high resolution, i.e. 300 dpi.

LINK:
http://weblogo.berkeley.edu/logo.cgi