Contents

Aligning coding sequences with MACSE

MACSE provides various subprograms to allow complex alignment strategies. The most direct approach is to simply use the alignSequences subprogram. Here we present some examples where this subrogram is used to solve frequent problems (e.g. alignment including pseudogenes, or NGS contigs)

Basic examples of coding sequence alignments

Example 1

The files and command lines for the following examples are available in the 01_basicAlignment directory of the demo archive.If you have a small set of coding sequences you can simply use MACSE with default options. The only required option is the name of the input alignment file.

```
java -jar macse.jar -prog alignSequences -seq tmem184a.fasta
```

Example 2

In the previous example the name of the output files are generated automatically based on the name of the input file. Alternatively you can provide the name of the two output files:

- the nucleotide alignment (-out_NT)
- the amino acid alignment (-out_AA)

java -jar macse.jar -prog alignSequences -seq tmem184a.fasta -out_NT tmem184a_nuc.aln -out_AA tmem184a_amino.aln

Example 3

If your sequences do not use the standard genetic code, you can indicate the default genetic code (def_gc) that MACSE should uses to align your sequences for (def_gc)For instance for chloroplatic MATK gene from Poacea you need to specify the genetic code #11.The genetic codes and associated numbers are those provided by the NCBI (http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi)

java -jar macse.jar -prog alignSequences -seq poacea_matk.fasta -def_gc 11

Note that the MACSE alignment is correct even though some sequences do not start at the first position of a codon since MACSE is able to handle those problematic cases by introducing a frameshift in the first and last codons (such frameshifts are not strongly penalized).

Aligning datasets including functional and non-functional sequences (pseudogenes).

MACSE can take two different data files as input "-seq" and "-seq_lr" containing sequences assigned respectively to "reliable" and "less reliable" sequences. You can then use different stop codon and frameshift costs for these two kinds of sequences using options "fs" and "stop" to define the cost of reliable protein-coding sequences and option "fs_lr" and "stop_lr" to define the cost of the so-called less reliable pseudogene sequences.

Example 1

The files and command lines for the following examples are available in the 02_aligningPseudogenes directory of the demo archive. If you have a small set of sequences and have no particular idea of which ones are protein-coding sequences and which ones are pseudogene sequences, you can simply use MACSE with default options (stop and frameshift have a penalty of 7 30).

java -jar macse.jar -prog alignSequences -seq AMELX.fasta

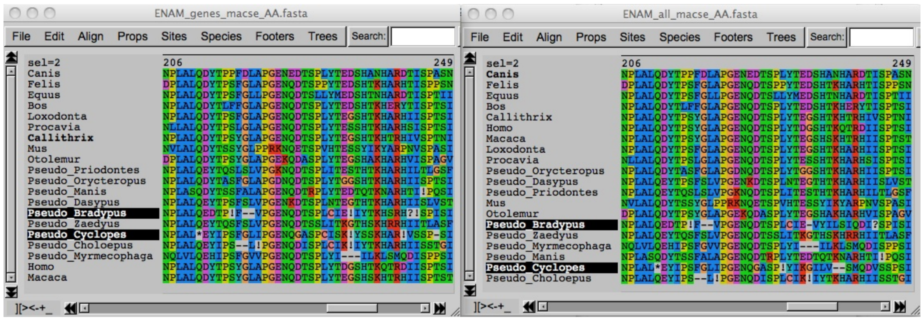
java -jar macse.jar -prog alignSequences -seq ENAM_all.fasta

Example 2

In the previous example, the same penalties are used for all sequences. If you a priori know which sequences are functional protein-coding sequences and which are non-functional pseudogenes, it is preferable to have them in different files. You can then let MACSE uses different costs for the sequences included in the two files assigning standard cost to one file (option "fs" and "stop") and a different cost for the other (option "fs_lr" and "stop_lr"). For pseudo-genes "-fs_lr 10 -stop_lr 10" is often a good initial parameter set.

java -jar macse.jar -prog alignSequences -seq ENAM_genes.fasta -seq_lr ENAM_pseudos.fasta -fs_lr 10 -stop_lr 10

Since frameshift and stop-codon costs are lower for a priori defined pseudogene sequences, MACSE is able to detect more events of this kind, hence producing a better alignment (see Figure Below).



```
sel=2                  206                                      249
Canis                  NPLALQDYTPPFDLAPGENEDTSPLYTEDSHANHARDTISPASN
Felis                  DPLALQDYTPSFGLAPGENQDTSPPYTEDSHTKHARHTISPPSN
Equus                  NPLALQDYTPSFGLLPGENQDTSLLYMEDSHTNHARDTISPTII
Bos                    NPLALQDYTLFFGLAPGENQDTSPLYTEDSHTKHERYTISPTSI
Loxodonta              NPLALQDYTPSFGLAPGENQDTSPLYTEGSHTKHARHIISPTSI
Procavia               NLLALQDYTPSLGLAPGENQDTSPLYTESSHTKHARHSISPTSI
Callithrix             NPLALQDYTPSYGLAPGENQDTSPLYTEGSHTKHTRHIVSPTNI
Mus                    NVLALQDYTSSYGLPPRKNQETSPVHTESSYIKYARPNVSPASI
Otolemur               DPLALQDYTPSYGLAPGEKQDASPLYTEGSHAKHARHVISPAGV
Pseudo_Priodontes      NPLALQEYTQSLSLVPGKNQDTSPLITESTHTKHARHILTLGSF
Pseudo_Orycteropus     NPLALQDYTASFGLAPGDNQDTSPLYTGGSHTKHARHIISPTSI
Pseudo_Manis           NPLASQDYTSSFALAPGENQDTRPLYTEDTQTKNARHTI!PQSI
Pseudo_Dasypus         NPLALQEYTPSFSLVPGENKDTSPLNTEGTHTKHARHIISLVST
Pseudo_Bradypus        NPLALQEDTP!F--VPGENQDTSPLCIE!IYTKHSRH?!SPISI
Pseudo_Zaedyus         NPLALQEYTQSFSLVPGENQDTSSLITKGTHSKHRRHIITLASF
Pseudo_Cyclopes        NPLAL*EYIPSFGLIPGENQGASPCISK!YSSKHAR!VSSP-SI
Pseudo_Choloepus       NPLALQEYIPS--L!PGENQDISPLCIK!IYTKHARHIISSTGI
Pseudo_Myrmecophaga    NQLVLQEHIPSFGVVPGENQDTSPLYI---ILKLSMQDISPPSI
Homo                   NPLALQDYTPSYGLAPGENQDTSPLYTDGSHTKQTRDIISPTSI
Macaca                 NPLALQDYTPSYGLAPGENQDTSPLYTEGSHSKHTRHIISPTST
```

```
sel=2                  206                                      249
Canis                  NPLALQDYTPPFDLAPGENEDTSPLYTEDSHANHARDTISPASN
Felis                  DPLALQDYTPSFGLAPGENQDTSPPYTEDSHTKHARHTISPPSN
Equus                  NPLALQDYTPSFGLLPGENQDTSLLYMEDSHTNHARDTISPTII
Bos                    NPLALQDYTLFFGLAPGENQDTSPLYTEDSHTKHERYTISPTSI
Callithrix             NPLALQDYTPSYGLAPGENQDTSPLYTEGSHTKHTRHIVSPTNI
Homo                   NPLALQDYTPSYGLAPGENQDTSPLYTDGSHTKQTRDIISPTSI
Macaca                 NPLALQDYTPSYGLAPGENQDTSPLYTEGSHSKHTRHIISPTST
Loxodonta              NPLALQDYTPSFGLAPGENQDTSPLYTEGSHTKHARHIISPTSI
Procavia               NLLALQDYTPSLGLAPGENQDTSPLYTESSHTKHARHSISPTSI
Pseudo_Orycteropus     NPLALQDYTASFGLAPGDNQDTSPLYTGGSHTKHARHIISPTSI
Pseudo_Dasypus         NPLALQEYTPSFSLVPGENKDTSPLNTEGTHTKHARHIISLVST
Pseudo_Priodontes      NPLALQEYTQSLSLVPGKNQDTSPLITESTHTKHARHILTLGSF
Mus                    NVLALQDYTSSYGLPPRKNQETSPVHTESSYIKYARPNVSPASI
Otolemur               DPLALQDYTPSYGLAPGEKQDASPLYTEGSHAKHARHVISPAGV
Pseudo_Bradypus        NPLALQEDTP!F--VPGENQDTSPLCIE-VYILSIQDI?SPISI
Pseudo_Zaedyus         NPLALQEYTQSFSLVPGENQDTSSLITKGTHSKHRRHIITLASF
Pseudo_Myrmecophaga    NQLVLQEHIPSFGVVPGENQDTSPLYI---ILKLSMQDISPPSI
Pseudo_Manis           NPLASQDYTSSFALAPGENQDTRPLYTEDTQTKNARHTI!PQSI
Pseudo_Cyclopes        NPLAL*EYIPSFGLIPGENQGASP!YIKGILV--SMQDVSSPSI
Pseudo_Choloepus       NPLALQEYIPS--L!PGENQDISPLCIK!IYTKHARHIISSTGI
```

Figure 1: Using lower frameshift and stop codon penalties for identified pseudogenes (left) allows to produce better alignment than if all sequences are considered equivalent (right). This can be seen, for instance, in this ENAM alignment for the Bradypus and Cyclopes sequences, which have an additional frameshift in the alignment on the left.

Aligning NGS data (reads or contigs).

MACSE can take two different data files as input “-seq” and “-seq_lr” containing sequences assigned respectively to “reliable” and “less reliable” sequences. You can then use different stop codon and frameshift costs for these two kinds of sequences using options “fs” and “stop” to define the cost of reliable reference protein-coding sequences and option “fs_lr” and “stop_lr” to define the cost of the so-called less reliable read or contig sequences.

Example 1

The files and command lines for the following examples are available in the 03_aligningNGSContigs directory of the demo archive. When aligning a set of sequences containing NGS data it is preferable, as for pseudogenes, to split the sequences in two files and to use different costs for frameshifts and stop codons in order to account for potential sequencing errors in NGS reads or contigs. For NGS raw sequences (reads or contigs) “-fs_lr 10 -stop_lr 15” is often a good initial parameter set. You can of course adapt these costs depending on the sequencing technology used since some induce more frequent errors in homopolymers (e.g. 454) seen as frameshifts by MACSE, while others rather induce nucleotide calling errors leading to potential stop codons. The ideal case is to have a set or reliable sequences (i.e. from public databases or manually annotated) that you can provide as reference to MACSE in order to guide the alignment process. In this case, we identify by BLAST that the reads are similar to Ensembl sequences ENSG00000119777 (TMEM214) and use orthologous CDS sequences to guide the alignments of 454 contigs while accounting for probable sequencing errors:

java -jar macse.jar -prog alignSequences -seq TMEM214.fasta -seq_lr TMEM214.fasta -fs_lr 10 -stop_lr 15”

Note that the more reliable sequences the better. If you include numerous reliable sequences or have a clean alignment corresponding to your current contigs, you can use the enrichAlignment to rapidly add new contigs to your initial alignment (see corresponding sections below for detailed examples).

Transforming coding sequence alignments with MACSE

MACSE provides subprograms, which allows splitting or merging alignments and removing frameshift nucleotides or codons.

Extracting a fraction of a coding sequence alignments

Example 1

The files and command lines for the following examples are available in the 01_splitAlignment directory of the demo archive.If you have an alignment, you can restrict it to a subset of sequences the names of which are listed in a separated file. For instance, considering tmem184a_NT.aln, which contains an alignment for 20 mammals you can restrict it to the 5 primate sequences specified in primate_seqId.list:

```
java -jar macse.jar -prog splitAlignment -align tmem184a_NT.aln -subset primate_seqId.list
```