FEB 14, 2023

Additional information for the manual annotation of the terpene synthase gene family in Melaleuca alternifolia (tea tree)

Julia Voelker[1]

[1]Faculty of Science and Engineering, Southern Cross University, Military Road, East Lismore NSW 2480, Australia

Julia Voelker

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.3byl4bjr2vo5/v1

Protocol Citation: Julia Voelker 2023. Additional information for the manual annotation of the terpene synthase gene family in Melaleuca alternifolia (tea tree). **protocols.io** https://dx.doi.org/10.17504/protocols.io.3byl4bjr2vo5/v1

MANUSCRIPT CITATION: The terpene synthase genes of *Melaleuca alternifolia* (tea tree) and comparative gene family analysis among Myrtaceae essential oil crops

Protocol status: Working
We use this protocol and it's working

Created: Feb 21, 2022

Last Modified: Feb 14, 2023

ABSTRACT

This protocol was created to be read in conjunction with the publication about the manual annotation of the terpene synthase (*TPS*) gene family in *Melaleuca alternifolia* (tea tree). It provides additional information regarding the computational methods mentioned in the manuscript, especially concerning specific commands for reproducibility of the methodology.

When using and citing this protocol, please also refer to the original article publication as mentioned in the metadata section of this protocol.

IMAGE ATTRIBUTION

M. alternifolia photograph by Mervyn Shepherd

PROTOCOL integer ID:
58489

Keywords: Melaleuca alternifolia, terpenoids, TPS annotation, Myrtaceae, tea tree

Manual terpene synthase (TPS) annotation

1 The genome of the tea tree reference genotype SCU01 (BioProject PRJNA702189) was used for the manual TPS annotation. Genes in the tea tree genome sequence were automatically predicted with Fgenesh++ v7.2.2 (RRID:SCR_018928) and GenemarkET+ v4.59 (RRID:SCR_011930). The methods for the Fgenesh++ prediction are explained in Voelker et al. (2021) and the accompanying protocol.For the Genemark prediction, the same NCBI nr-protein plant database and RNA-seq reads were used as evidence. The protein database was aligned to the genome assembly using ProtHint v2.5.0 (RRID:SCR_021167) and RNAseq spliced read alignments were created with STAR v2.7 (RRID:SCR_004463):

- Julia Voelker, Mervyn Shepherd, Ramil Mauleon, A high-quality draft genome for Melaleuca alternifolia (tea tree): a new platform for evolutionary genomics of myrtaceous terpene-rich species, *Gigabyte*, 2021 https://doi.org/10.46471/gigabyte.28

- Julia Voelker 2021. Additional information for the creation of a high-quality draft genome for Melaleuca alternifolia (tea tree). protocols.io https://dx.doi.org/10.17504/protocols.io.bwi2pcge

1.1 RNAseq read trimming

M. alternifolia RNAseq data from three other individual trees of the same chemotype (BioProject PRJNA388506; SRR5630605, SRR5630611, SRR5630622) were downloaded and converted to fastq format with the NCBI SRA toolkit. Quality controls were undertaken with FastQC (RRID:SCR_014583) and reads were trimmed with Flexbar (RRID:SCR_013001).

```
#Identify adapter sequences in each read library
bbmap/bbmerge.sh in1=RNAseq_1.fastq in2=RNAseq_2.fastq
outa=adapters.fasta

#Trim reads and remove identified adapters
flexbar -r RNAseq_1.fastq -p RNAseq_2.fastq -a adapters.fasta --
adapter-trim-end RIGHT -adapter-error-rate 0.1 --adapter-min-
overlap 6 -ap ON --min-read-length 40 --max-uncalled 1 -q TAIL -
qf i1.8 -qt 30 -pre-trim-left 10 -t RNAseq_trim
```

1.2 GenemarkET+ gene prediction

Trimmed RNAseq reads and sequences in the NCBI nr-protein database were aligned to the genome assembly before running GenemarkET+

```
#Run GenemarkES prediction without any evidence (required for
ProtHint)
gmes_petap.pl --ES --sequence assembly.fasta [--cores]

#Run ProtHint with NCBI nr-protein plant database
prothint.py assembly.fasta /path/to/database/nr_plants --
geneMarkGtf genemarkES_out.gtf [--threads]

#Create genome index for STAR
STAR [--runThreadN] --runMode genomeGenerate --
genomeSAindexNbases 13 --genomeDir /path/to/STAR_index/ --
genomeFastaFiles /path/to/assembly.fasta

#Align all RNAseq reads to genome
STAR [--runThreadN] --genomeDir /path/to/STAR_index/ --
readFilesIn
./SRR5630605_trim_1.fastq,./SRR5630611_trim_1.fastq,./SRR5630622
_trim_1.fastq
./SRR5630605_trim_2.fastq,./SRR5630611_trim_2.fastq,./SRR5630622
_trim_2.fastq --outSAMattrRGline ID:SRR5630605 , ID:SRR5630611 ,
ID:SRR5630622 --outFileNamePrefix ./RNAseq --outSAMtype BAM
SortedByCoordinate

#Convert STAR alignment output to intron file required by
Genemark, the sript is part of the Genemark package
~/programs/gmes_linux_64/star_to_gff.pl --star
./RNAseqSJ.out.tab  --gff RNAseq_introns.gff  --label
STAR_RNAseq

#Run GenemarkET+ gene prediction
gmes_petap.pl --ET /path/to/RNAseq_introns.gff --evidence
/path/to/ProtHint/evidence.gff --sequence assembly.fasta --v [--
cores]
```

1.3 Exonerate alignment of TPS sequences

As further evidence for exon-intron borders, TPS protein sequences were aligned to the genome assembly with Exonerate v2.4.0 (RRID:SCR_016088). First, protein sequences matching the ‘terpene synthase’ keyword search were retrieved from Phytozome v13 (accessed 20/05/2021) for *A. thaliana* ARAPORT11, *C. citriodora* v2.1, *E. grandis* v2.0, *P.*

trichocarpa v4.1 and *V. vinifera* v2.1. These sequences were then aligned to the genome using the protein2genome mode of Exonerate.

```
#Align reference sequences of TPS proteins to genome
exonerate -m protein2genome -t /path/to/assembly.fasta -q
/path/to/terpene_synthase_ref_protein_phytozome.fa --querytype
protein --targettype dna -V 1 --showalignment FALSE --showvulgar
TRUE --showtargetgff TRUE --fsmmemory 500 --percent 40 --
dpmemory 50 --maxintron 5000 > protein2genome_TPS.out

#Filter output for visualisation in genome viewer
grep -v 'vulgar' protein2genome_TPS.out | sed '1d; $d' >
protein2genome_TPS.gff

awk '($3 !~ /cds|similarity|intron/ ) {print}'
protein2genome_TPS.gff > protein2genome_TPS.exons_splice.gff
```

1.4 Manual adjustment of TPS gene structures

The GFF-file containing exons and splice sites from TPS alignments were visualised in Apollo (RRID:SCR_001936) together with RNAseq read alignments, the Fgenesh++ and the GenemarkET+ gene predictions and the genome sequence. Potential terpene synthases were identified based on TPS alignments and predicted protein sequences containing TPS Pfam domains. To identify the C-terminal (PF03936) and N-terminal (PF01397) TPS Pfam domains, the GenemarkET+ and Fgenesh++ protein sequence predictions were screened for PfamA domains (RRID:SCR_004726) using InterProScan (RRID:SCR_005829) in Galaxy Europe (RRID:SCR_006281).
All lines of evidence were taken into consideration for the manual annotation of the terpene synthase gene structures using the genome annotation editor Apollo.

1.5 Quality control and removal of duplicates

The manually annotated *TPS* genes were named consecutively (MaltTPS001-MaltTPS73) based on their location on a scaffold, from lowest to highest scaffold number, and their location within the scaffold. Some *TPS* sequences, however, seemed to be highly similar, so percent sequence identity matrixes were calculated via multiple sequence alignment in Clustal Omega v2.1 (RRID:SCR_001591) with default settings, using the predicted protein, CDS and genomic TPS sequences as input, respectively. Furthermore, the scaffolds containing *TPS* were aligned in all-vs-all manner using NUCmer v3.1 (RRID:SCR_018171) and Minimap2, respectively. Each scaffold was manually assessed for overlaps with other scaffolds and potential assembly errors or scaffold duplications by going through the alignment outputs and also visualising alignments with Alvis.

.

```
#Align scaffolds containing TPS in all-vs-all manner with NUCmer
and create coords file
nucmer --prefix=Malt_TPS_scf_nucmer Malt_TPS_scaffolds.fasta
Malt_TPS_scaffolds.fasta
show-coords -rcld -T Malt_TPS_scf_nucmer.delta >
Malt_TPS_scf_nucmer.coords.tab

#Remove self-alignments of scaffolds -> this file is for manual
assessment of alignments
awk '$14 != $15' Malt_TPS_scf_nucmer.coords.tab >
Malt_TPS_scf_nucmer.coords.filter.tab

#Create Minimap2 alignment of scaffolds in all-vs-all manner
minimap2 -x asm20 -DP [-t <threads>] -o Malt_TPS_scf_minimap.paf
Malt_TPS_scaffolds.fasta Malt_TPS_scaffolds.fasta

#Remove minimap self-alignments and filter for alignments with
>60% sequence similarities for manual assessment of alignments
awk '$1 != $6' Malt_TPS_scf_minimap.paf >
Malt_TPS_scf_minimap_filtered.paf

awk '{if ((P=$11/$2*100) >= 60) {print P"\t"$_}}'
Malt_TPS_scf_minimap_filtered.paf >
Malt_TPS_scf_minimap_60pc.tab

#Create Alvis input file for visualisation of nucmer alignments

show-coords -B -rcld Malt_TPS_scf_nucmer.delta >
Malt_TPS_scf_nucmer_btab.coords
awk '$1 != $8' Malt_TPS_scf_nucmer_btab.coords >
Malt_TPS_scf_nucmer_btab.filter.coords

#Run Alvis, use minimap or nucmer alignment as input and create
different visualisation types, filter out all alignments that
are less than 10% of target read and manually inspect outcomes
java -jar ~/programs/alvis-master/dist/Alvis.jar -type
{contigAlignment|alignment} -filter -minAlignmentPC 10 -inputfmt
{paf|coords} -outputfmt svg -in <paf or coords input> -outdir
<directory> -out <prefix>
```

1.6 Screen for conserved regions and organelle-targeting signal peptides

After the removal of scaffold duplications, the final set of manual annotations were screened

for conserved amino acid motifs using patmatdb on Galaxy Australia. The NSE/DTE motif sequences for TPS-a, TPS-b and TPS-g proteins were extracted from multiple-sequence alignments with MUSCLE (RRID:SCR_011812) in Unipro UGENE v39.0 (RRID:SCR_005579). The extracted sequences with a length of 11 amino acids were then visualised with WebLogo v3.7.4 (RRID:SCR_010236).
N-terminal transit peptides in the amino acid sequence of TPS were predicted using TargetP v2.0 (RRID:SCR_019022) and DeepLoc v1.0.

Phylogenetic tree creation

2 A phylogenetic trees was created for the predicted *M. alternifolia* TPS protein sequences. For comparison to other species, the TPS amino acid sequences for *A. thaliana*, *C. citriodora, E. grandis* and *E. globulus*, *P. trichocarpa* and *V. vinifera* were retrieved and used for the creation of an additional phylogenetic tree containing the TPS sequences of all mentioned species.

2.1 *M. alternifolia* tree and multi-species tree

The *M. alternifolia* TPS protein sequences were manually assessed in Unipro UGENE v39.0 (RRID:SCR_005579) and where possible, trimmed to the start of the RRx8W motif. Multiple-sequence alignment (MSA) was carried out with the inbuilt MUSCLE tool (RRID:SCR_011812), and columns with gaps >75% were removed from the alignment. The trimmed MSA output was converted into NEXUS format and used as input for phylogenetic tree construction using PhyML v3.3 (RRID:SCR_014629).

```
#PhyML command for M. alternifolia tree
mpirun -np 10 ~/programs/phyml-3.3.20200621/src/phyml-mpi -i
Malternifolia_TPS_muscle.nex -b 100 -d aa -m JTT -v e -a e -o tl
--print_trace --no_memory_check --leave_duplicates -c 4
```

Similarly, the sequences for all species were loaded into Unipro UGENE and where possible, trimmed to the start of the RRx8W motif, as already done for *M. alternifolia*. Next, MSA was carried out for all species, using MUSCLE and trimming the resulting alignment to remove all columns with >75% gap representation. Phylogenetic tree construction was carried out with 100 bootstraps and settings as already mentioned for the *M. alternifolia* tree. After the whole tree was visualised and annotated in iTOL v6, different clades were exported into separated trees for a clearer visualisation of each subfamily.

Synonymous substitution rates among orthogroups

3 In addition to investigating the similarity of *TPS* sequences among *M. alternifolia, E. grandis* and *C. citriodora* via phylogenetic comparison, the overall similarity of their protein sequences were also compared by investigating single-copy orthologous genes shared among species and

calculating their synonymous substitution rates.

3.1 OrthoFinder analysis

Protein sequences for primary transcripts of E*. grandis* v2*, C. citriodora* v2.1*, Populus trichocarpa* v4*, Salix purpurea* v1 and *Vitis vinifera* v2.1 were obtained from Phytozome v13. For *M. alternifolia,* protein sequences predicted by fgenesh++ and the manually predicted TPS sequences were used.
The sequences of all species were input for OrthoFinder v2.5.2 to identify single-copy orthologues, i.e. orthogroups containing one orthologous gene per species, with all species present in the group.

```
#Orthofinder command, protein sequences were in separate fasta
files per species
orthofinder -t 15 -f ./sequence_dir/
```

3.2 Synonymous substitution rates

Orthofinder identified single-copy orthogroups and created fasta files containing one amino acid sequence per species for each orthogroup. The sequences withing each orthogroup were used for mutiple sequence alignment (MSA) in MUSCLE v5.1. A short script was created to take the sequences within each orthogoup fasta file for MSA, and output the alignments into a newly created alignment directory

```
for ORTHO in ../Single_Copy_Orthologue_Sequences/OG*
        do
                NAME=`basename ${ORTHO} .fa`
                muscle5.1 -align ${ORTHO} -output
${NAME}_muscle.fa -threads 14
        done
```

3.3 Translation of amino acid alignments into nucleotide sequences

To calculate synonymous substitution rates from the MSA, the protein sequence alignment had to be converted into coding sequence (CDS) alignments for each orthogroup with pal2nal v14. All coding sequences for the above mentioned species were downloaded from Phytozome v13.
 In order to run pal2nal, the number of input sequences needed to be exactly the same as the number of output sequences. Therefore, the CDSs from all species had to be placed into separate fasta files corresponding to the orthogroups.

To simplify further processing, the MSA files were also sorted alphabetically, so that sequences for each species were at the same position in all files.

```
#Sort MSA output files by species name, and place into new
directory
mkdir sortbyname
for MSA in OG*
    do
        NAME=`basename ${MSA} .fa`
        bbmap/sortbyname.sh in=${MSA}
out=./sortbyname/${NAME}_sort.fa ignorejunk=t
    done

#Create fasta file containing CDS of all species
cat
/path_to/Malt_fgenesh_final_annot_incl_58_manual_TPS.cds.fasta
/path_to/Ccitriodora_507_v2.1.cds_primaryTranscriptOnly.fa
/path_to/Egrandis_297_v2.0.cds_primaryTranscriptOnly.fa
/path_to/Spurpurea_289_v1.0.cds_primaryTranscriptOnly.fa
/path_to/Vvinifera_457_v2.1.cds_primaryTranscriptOnly.fa
/path_to/Ptrichocarpa_533_v4.1.cds_primaryTranscriptOnly.fa >
CDS_sequences_all_species.fasta

#Extract CDSs into separate files per orthogroup
for PEP in ../Alignment/sortbyname/OG*
        do
                NAME=`basename ${PEP} _muscle.fa`
                bbmap/filterbyname.sh
in=CDS_sequences_all_species.fasta
                out=./CDS_sequences_per_orthogroup/$NAME.cds.fa
names=${PEP} include=t
                ignorejunk=t
        done

#Run pal2nal to translate protein alignment into codon alignment
for PEP in ../Alignment/sortbyname/OG*
        do
                NAME=`basename ${PEP} _muscle_sort.fa`

CDS=./CDS_sequences_per_orthogroup/${NAME}.cds.fa
                echo "start ${PEP} CDS alignment"
                pal2nal.pl ${PEP} ${CDS} ;
                -output paml >
./CDS_alignment_seq/${NAME}_cds_align.nuc
                echo "end ${PEP} CDS alignment"
        done
```

3.4 Calculation of synonymous substitution rates

Once all orthogroup nucleotide sequences were aligned in separate files, PAML codeml v4.10.3 was run for each orthogroup, with pairwise substituion rate calculation (mode -2)

```
#General codeml control file
      seqfile = /path_to/CDS_alignment/OGinput_cds_align.nuc *
sequence data filename
     treefile = /path_to/Gene_Trees/OGinput_tree.txt      * tree
structure file name
      outfile = OGinput_codeml_pairwise.txt           * main
result file name

        noisy = 9  * 0,1,2,3,9: how much rubbish on the screen
      verbose = 1  * 0: concise; 1: detailed, 2: too much
      runmode = -2  * 0: user tree;  1: semi-automatic;  2:
automatic
                   * 3: StepwiseAddition; (4,5):PerturbationNNI;
-2: pairwise

      seqtype = 1  * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

*        ndata = 10
        clock = 0  * 0:no clock, 1:clock; 2:local clock;
3:CombinedAnalysis
       aaDist = 0  * 0:equal, +:geometric; -:linear, 1-
6:G1974,Miyata,c,p,v,a
   aaRatefile = ../dat/jones.dat  * only used for aa seqs with
model=empirical(_F)
                   * dayhoff.dat, jones.dat, wag.dat, mtmam.dat,
or your own

        model = 0
                   * models for codons:
                       * 0:one, 1:b, 2:2 or more dN/dS ratios
for branches
                   * models for AAs or codon-translated AAs:
                       * 0:poisson, 1:proportional, 2:Empirical,
3:Empirical+F
                       * 6:FromCodon, 7:AAClasses, 8:REVaa_0,
9:REVaa(nr=189)

      NSsites = 0  * 0:one w;1:neutral;2:selection;
3:discrete;4:freqs;
                   * 5:gamma;6:2gamma;7:beta;8:beta&w;9:betaγ
                   * 10:beta&gamma+1; 11:beta&normal>1;
12:0&2normal>1;
                   * 13:3normal>0

        icode = 0  * 0:universal code; 1:mammalian mt; 2-10:see
below
        Mgene = 0
                   * codon: 0:rates, 1:separate; 2:diff pi,
3:diff kapa, 4:all diff
                   * AA: 0:rates, 1:separate

    fix_kappa = 0  * 1: kappa fixed, 0: kappa to be estimated
        kappa = 2  * initial or fixed kappa
    fix_omega = 0  * 1: omega or omega_1 fixed, 0: estimate
        omega = .4 * initial or fixed omega, for codons or
codon-based AAs

    fix_alpha = 1  * 0: estimate gamma shape parameter; 1: fix
it at alpha
        alpha = 0. * initial or fixed alpha, 0:infinity
(constant rate)
       Malpha = 0  * different alphas for genes
        ncatG = 8  * # of categories in dG of NSsites models

        getSE = 0  * 0: don't want them, 1: want S.E.s of
estimates
 RateAncestor = 1  * (0,1,2): rates (alpha>0) or ancestral
states (1 or 2)

   Small_Diff = .5e-6
    cleandata = 0  * remove sites with ambiguity data (1:yes,
0:no)?
*  fix_blength = 1  * 0: ignore, -1: random, 1: initial, 2:
fixed, 3: proportional
       method = 0  * Optimization method 0: simultaneous; 1: one
branch a time

* Genetic codes: 0:universal, 1:mammalian mt., 2:yeast mt.,
3:mold mt.,
* 4: invertebrate mt., 5: ciliate nuclear, 6: echinoderm mt.,
* 7: euplotid mt., 8: alternative yeast nu. 9: ascidian mt.,
* 10: blepharisma nu.
* These codes correspond to transl_table 1 to 11 of GENEBANK.
```

After the creation of the general codeml control file according to the manual, the file had to be edited to contain the correct file names for each orthogroup, which was then used to run

PAML codeml on each orthogroup CDS alignment:

```
#Edit the codeml control file for each input CDS alignment and
corresponding phylogenetic tree file (tree is part of output
from orthofinder)
#Create separate output directory for each orthogroup

for i in ../CDS_alignment/OG*_cds_align.nuc
        do
                NAME=`basename ${i} _cds_align.nuc`
                mkdir ${NAME}
                cp codeml_ortho.ctl ./${NAME}/codeml_${NAME}.ctl
                sed -i "s/OGinput/${NAME}/g"
./${NAME}/codeml_${NAME}.ctl
                cd ./${NAME}
                echo "start PAML codeml for ${NAME}"
                codeml ./codeml_${NAME}.ctl
                cd ../
        done
```

The pairwise values for synonymous substitution rates (Ks) between species were extracted from the output for each orthogroup. Ks values greater than 2 were filtered out to remove ortholog groups whose synonymous substitutions may be oversaturated. A density plot showing the pairwise Ks distribution was created using ggplot2 in R.

The Ks mutatation rate/site/year (R) between species was calculated with the formula R = Ks/(2*divergence age), using the Ks value from the peak maximum of the Ks distribution.

References

4 **References for tools mentioned in the sections above:**

Fgenesh++:
Solovyev V, Kosarev P, Seledsov I, Vorobyev D (2006) Automatic annotation of eukaryotic genes, pseudogenes and promoters. Genome Biol 7 Suppl 1 (Suppl 1):S10.11-12. doi:https://doi.org/10.1186/gb-2006-7-s1-s10
GenemarkET+:
Lomsadze A, Burns PD, Borodovsky M (2014) Integration of mapped RNA-Seq reads into automatic training of eukaryotic gene finding algorithm. Nucleic acids research 42 (15):e119-e119. doi:https://doi.org/10.1093/nar/gku557
ProtHint:
Brůna T, Lomsadze A, Borodovsky M (2020) GeneMark-EP+: eukaryotic gene prediction with self-training in the space of genes and proteins. NAR Genomics and Bioinformatics 2 (2).

doi:https://doi.org/10.1093/nargab/lqaa026
STAR:
Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR (2012) STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29 (1):15-21. doi:https://doi.org/10.1093/bioinformatics/bts635
Flexbar:
Dodt M, Roehr JT, Ahmed R, Dieterich C (2012) FLEXBAR-Flexible Barcode and Adapter Processing for Next-Generation Sequencing Platforms. Biology (Basel) 1 (3):895-905. doi:https://doi.org/10.3390/biology1030895
BBtools:
Bushnell B BBMap. sourceforge.net/projects/bbmap/.
Exonerate:
Slater GSC, Birney E (2005) Automated generation of heuristics for biological sequence comparison. BMC bioinformatics 6:31. doi:https://doi.org/10.1186/1471-2105-6-31
Apollo:
Dunn NA, Unni DR, Diesh C, Munoz-Torres M, Harris NL, Yao E, Rasche H, Holmes IH, Elsik CG, Lewis SE (2019) Apollo: Democratizing genome annotation. PLOS Computational Biology 15 (2):e1006790. doi:https://doi.org/10.1371/journal.pcbi.1006790
Clustal Omega:
Madeira F, Pearce M, Tivey ARN, Basutkar P, Lee J, Edbali O, Madhusoodanan N, Kolesnikov A, Lopez R (2022) Search and sequence analysis tools services from EMBL-EBI in 2022. Nucleic acids research. doi:https://doi.org/10.1093/nar/gkac240
NUCmer:
Marçais G, Delcher AL, Phillippy AM, Coston R, Salzberg SL, Zimin A (2018) MUMmer4: A fast and versatile genome alignment system. PLOS Computational Biology 14 (1):e1005944. doi:https://doi.org/10.1371/journal.pcbi.1005944
Minimap2:
Li H (2018) Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 34 (18):3094-3100. doi:https://doi.org/10.1093/bioinformatics/bty191
Alvis:
Martin S, Leggett RM (2021) Alvis: a tool for contig and read ALignment VISualisation and chimera detection. BMC Bioinformatics 22 (1):124. doi:https://doi.org/10.1186/s12859-021-04056-0
Unipro UGENE:
Okonechnikov K, Golosova O, Fursov M, team tU (2012) Unipro UGENE: a unified bioinformatics toolkit. Bioinformatics 28 (8):1166-1167. doi:https://doi.org/10.1093/bioinformatics/bts091
TargetP:
Almagro Armenteros JJ, Salvatore M, Emanuelsson O, Winther O, von Heijne G, Elofsson A, Nielsen H (2019) Detecting sequence signals in targeting peptides using deep learning. Life Science Alliance 2 (5):e201900429. doi:https://doi.org/10.26508/lsa.201900429
DeepLoc:
Almagro Armenteros JJ, Sønderby CK, Sønderby SK, Nielsen H, Winther O (2017) DeepLoc: prediction of protein subcellular localization using deep learning. Bioinformatics 33 (21):3387-3395. doi:https://doi.org/10.1093/bioinformatics/btx431

PhyML:
Guindon S, Dufayard J-F, Lefort V, Anisimova M, Hordijk W, Gascuel O (2010) New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. Systematic Biology 59 (3):307-321. doi:https://doi.org/10.1093/sysbio/syq010
Orthofinder:
Emms DM, Kelly S (2019) OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biology 20 (1):238. doi:https://doi.org/10.1186/s13059-019-1832-y
pal2nal:
Suyama M, Torrents D, Bork P (2006) PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. Nucleic acids research 34 (suppl_2):W609-W612. doi:https://doi.org/10.1093/nar/gkl315
PAML:
Yang Z (2007) PAML 4: Phylogenetic Analysis by Maximum Likelihood. Molecular Biology and Evolution 24 (8):1586-1591. doi:https://doi.org/10.1093/molbev/msm088