Species delimitation

P. Škaloud, Faculty of Science, Charles University, Prague

version 2017-01-08

In recent years, the number of methods available for delimiting species based on sequence data has dramatically increased. Some methods, in particular those based on assessment of distances between sequences (e.g., ABGD) and on the coalescent theory (e.g., GMYC, Brownie, SpedeSTEM, BPP, bPTP) have gained great popularity and have been applied to many taxonomic groups. The goal of these methods is to provide a convenient and reliable tool for species delineation; they have been used in many studies as a base to justify taxonomic conclusions, particularly the split into numerous species of difficult taxa for which morphological discrimination is impossible.

We will start with the most widely used method, GMYC. First, we need to produce ultrametric tree using the BEAST software. Then, we will run GMYC delimitation using the R package "splits". Finally, we will delimit the species boundaries using another method, the bPTP.

1. Generating the ultrametric tree using BEAST

Open BEAUti,

- load alignment in NEXUS format (press + button and select "alignment.nex"
- Partitions – we can unlink here different partitions defined at the end of the NEXUS file. In this case, three partitions are defined – ITS1, RNA, ITS2. Select „Unlink Subst. Models" separately for every partition
- Sites – here you should set substitution models for each partition. For this testing run, we can leave predefined HKY models.
- Clocks – here we will set mutation rates. Under the "model" option, select "uncorelated relaxed clock" and "lognormal" relaxed distribution
- Trees – as Tree Prior, select Coalescent: Constant Size
- MCMC – here we set the number of MCMC generations (Length of chain). Usually Length of chain is set to 10,000,000, Echo state to every 10,000 and Log parameters to every 1,000. To speed up the process, set the Length of chain to 500,000 only.
- Finally, generate BEAST file by clicking on the button on the bottom-right, press "Continue" and save the file as "alignment.xml".
- You can save this BEAUti file for future work by File -> Save as...

Open BEAST,

- As BEAST xml file, select the saved "alignment.xml" by clicking on "Choose file..."
- Press "Run"

MCMC will run for a few minutes. Afterwards, open Tracer to check for MCMC convergence

- Open the "alignment.log" file of MCMC analysis by pressing + button on the left
- We can check the likelihood distribution during the MCMC run by displaying the "Trace" window. Since we ran the analysis for 500,000 generations only, we can clearly see the oscilation of the likelihood on the graph. Anyway, the likelihood does not increase obviously, so we can analyse the results. Optimally, several (5- 10) BEAST runs should be run and load to Tracer by clicking + button as long as ESS values are highlighted in red.
- We should set "Burn-In" to remove initial MCMC generations. Try to change Burn-In to 10000 to see the change in likelihood distribution.

Open TreeAnnotator,

- Specify “Burnin” based on the trace shown by Tracer (10000 in this case)
- Set “Posterior probability limit” to 0.5
- Set “Node heights” to Mean heights
- Select Input tree file (“alignent.trees”)
- Specify Output File (e.g., “ultrametric.tree”)
- Press Run to execute the analysis

After the analysis, the resulting tree can be displayed, e.g., in FigTree.

Note: If multiple BEAST runs are performed (which is strongly recommended), Tree files should be merged using LogCombiner prior generating the final ultrametric tree by TreeAnnotator.

2. Species delimitation by the GMYC method

Open R Studio,

- *File -> New File -> R Script* - save this new file to your working directory (the same with input files)
- *Session -> Set Working Directory -> To source file location*
- Now sequentially copy-paste following commands to the new R file and run each batch. To run the command, highlight the lines you just inserted by mouse and press Ctrl+R.

First install and load the required libraries for all the downstream analyses – i.e. add the following text to your script and Run

```
install.packages(c("ape", "paran"))
install.packages("splits", repos = "http://R-Forge.R-project.org")
library(paran)
library(ape)
library(splits)
```

Now load and visualize the ultrametric tree produced by BEAST analysis.

```
my_tree<-read.nexus('ultrametric.tree')
# we can remove outgroup taxa to delimit the species correctly
my_tree=drop.tip(my_tree, "SAG_245_80_HG972969_Elliptochloris_bilobata")
my_tree=drop.tip(my_tree, "SAG_62_90_HG972970_Elliptochloris_antarctica")
# display the tree
plot(my_tree)
```

Then we can run the GMYC analysis

```
gmyc.results<-gmyc(my_tree, method = "single", interval = c(0, 10), quiet = FALSE)
```

We can use the function summary() to see how many species are found (including the confidence interval) and to check the significance of species delimitation by LR test

```
summary(gmyc.results)
```

The first five lines of the summary list the likelihood score of the model that consider that all the sequences belong to the same species, and then the likelihood score of the model that splits the sequences into different species. Result of LR test is displayed below.

The output then lists how many clusters, and entities, are associated with the highest likelihood score. Finally, the last line indicates the threshold time, i.e. the time at which the model infers that the threshold transitioning from the speciation-level events to the coalescent-level events takes place. In our analysis, we did not calibrate our tree, and this value is not meaningful. However, if we use a calibrated ultrametric tree as an input, this value can be interpreted in a biological context.

Next, we can obtain a list of delimited species entities by typing

```
spec.list(gmyc.results)
```

This command returns a 2-column table. The first column lists the species number as inferred from GMYC, and the second column lists the sample identifier.

We can then generate three plots showing GMYC results by typing

```
plot(gmyc.results)
```

By hitting "Enter" on the keyboard, three plots are displayed: (1) the number of lineages through time, with a red vertical line showing the inferred position of the threshold; (2) the profile of the likelihood through time; (3) the tree with the individual clusters highlighted in red.

Finally, we can plot the significance or delimited species clusters by a couple of commands:

```
pdf("GMYC_support.pdf", width = 10, height=3)  # generate a PDF file
support <- gmyc.support(gmyc.results)          # estimate the supports
is.na(support[support == 0]) <- TRUE           # select the nodes to show
support values
plot(my_tree, cex=.6, no.margin=TRUE)          # plot the tree
nodelabels(round(support, 2), cex=.7)          # plot the support values
dev.off()
```

3. Species delimitation by the bPTP method

We will take advantage of the bPTP species delimitation server where we will run the analysis:

- Prior the analysis, re-save the ultrametric tree obtained by BEAST analysis in FigTree. First, open the tree by File -> Open -> select "ultrametric.tree". Then, re-save it by File -> Export Trees -> Tree file format: NEXUS.
- Open the web page http://species.h-its.org/ptp/
- Select the input tree in NEXUS, generated by FigTree
- Set "No. MCMC generations" to 200,000
- Specify outgroup taxa by typing "SAG_245_80_HG972969_Elliptochloris_bilobata SAG_62_90_HG972970_Elliptochloris_antarctica" and select "Remove outgroups"
- specify your e-mail address and press "Submit" button

After the analysis, several output files can be downloaded, including the ML and Bayesian species delimitation solutions with the individual clusters highlighted in red, and the support values of all species entities (delimitation results).

Trait evolution

Tomáš Fér, Faculty of Science, Charles University, Prague
version 2017-01-12

The phylogenetic tree describes a hypothesis about evolutionary relationship between individuals. Each trait of interest (e.g., flower colour, genome size, life form, growth rate etc.) can be mapped onto the phylogeny. We could test a hypothesis that particular trait value and its distribution on a phylogeny is random, i.e. that no single tip bearing a given character trait is any more likely to share that trait with adjoining taxa than we would expect due to chance. In other words, how strong is the phylogenetic signal for particular trait. In this tutorial we learn how to test the trait phylogenetic association and how to reconstruct the trait evolution along given phylogenetic tree (and display it graphically) for both continuous and discrete character.

Last, phylogenetic generalized least square (PGLS) comparative method for phylogenetically correct trait correlation and its comparison with ordinary least square methods is included.

1. Evolution of a continuous character

The evolution of continuous characters, e.g., genome size or body size, can be modelled using the function fastAnc from the R package phytools. We will work with genome size data (in the file cx.csv) in the genus Globba (phylogenetic tree in s.nex)

Open R Studio,

- *File -> New File -> R Script* - save this new file to your working directory (the same with input files)
- *Session -> Set Working Directory -> To source file location*

Now sequentially copy-paste following commands to the new R file and run each batch. To run the command, highlight the lines you just inserted by mouse and press Ctrl+R.

Load libraries

```
library (phytools)
library (caper)
```

Read the phylogenetic tree and visualize it:

```
tree = read.nexus("s.nex")    #read a tree
plot(tree)                    #plot a tree
```

Read the text file with trait values and assign specific trait and sample names to a vector. Input text file is a table with values separated by commas, first column includes taxon names (must match the names in the tree), the second trait values. In the first row there are headers (names of the columns, i.e. ‘taxon’, ‘cvalue’, ‘app’ and ‘max’):

```
data = read.csv("traits.csv")
data1 <- as.vector(data$cvalue) #save values (in a column 'cvalue') as a
vector
names(data1)<-data$taxon #assign taxon names (in a column 'taxon')to the
cvalue values in data1
```

Estimate Pagel's lambda (degree of phylogenetic signal), model likelihood (logL), likelihood for lambda=0 (logL0) and significance of the difference between these two models (P), i.e. a test between these two values (whether the lambda is significantly different from 0):

```
phylosig(tree, data1, method="lambda", test=TRUE)
```

Pagel's lambda, kappa and delta can be also estimated with functions from the package caper. First, the phylogenies are combined with trait values (function comparative.data) and second, the pgls function fits the linear model taking into account phylogenetic non-independence between data points.

```
sumdata <- comparative.data (phy = tree, data = data, names.col = taxon,
vcv = TRUE, na.omit = FALSE, warn.dropped = TRUE)
model.lambda <- pgls (cvalue~1, data = sumdata, lambda= "ML")
summary (model.lambda)
```

The model estimates lambda branch length scaling parameter and its confidence intervals. Now we can plot a likelihood surface for lambda and save it as a *.png:

```
mod.l <- pgls.profile (model.lambda, "lambda")
plot (mod.l)
dev.copy (png, "lambda.png")
dev.off()
```

Similarly we can also do estimates for parameter delta (here we manually set limits for the estimates of all scaling parameters) and save the picture with its likelihood surface:

```
sumdata <- comparative.data (phy = tree, data = data, names.col = taxon,
vcv = TRUE, na.omit = FALSE, warn.dropped = TRUE)
model.delta <- pgls (cvalue~1, data = sumdata, delta = "ML", bounds =
list(lambda=c(0.001,1), kappa=c(1e-6,3), delta=c(1e-6,7))) # set upper
limit for delta to 7 (default is 3)
summary (model.delta)
mod.d <- pgls.profile (model.delta, "delta")
plot (mod.d)
dev.copy (png, "delta.png")
dev.off()
```

And also estimate parameter kappa:

```
sumdata_3d <- comparative.data (phy = tree, data = data, names.col = taxon,
vcv.dim = 3, vcv = TRUE, na.omit = FALSE, warn.dropped = TRUE) # creates 3D
variance-covariance matrix
model.kappa <- pgls (cvalue~1, data = sumdata_3d, kappa= "ML")
summary (model.kappa)
mod.k <- pgls.profile (model.kappa, "kappa")
plot (mod.k)
dev.copy (png, "kappa.png")
dev.off()
```

The change of trait values can be also reconstructed throughout the phylogeny and plotted, the reconstruction is estimated by fastAnc function from phytools package:

```
contMap(tree, data1, res=100, fsize=NULL, ftype=NULL, lwd=4, legend=NULL,
lims=NULL, outline=FALSE, sig=3, type="phylogram", direction="rightwards",
plot=TRUE)
```

Make a ML reconstruction of trait ancestral values for each each node in a tree (a vector of values) and add these values as node labels. This tree can be opened in, e.g., FigTree:

```
anc <- fastAnc(tree, data1)
writeAncestors(tree, Anc=anc, file="treeanc.tre", format="nexus")
```

2. Evolution of a discrete character

The evolution of categorical characters, e.g., life forms, number of flowers etc. can be modelled using either continuous-time Markov chain model (so-called Mk model) or stochastic mapping. Both methods estimate ancestral character states and are implemented in the R package phytools.

First, read the phylogenetic tree and the file with trait values for all samples and match tree and trait samples:

```
tree = read.nexus("s.nex") #read a tree
data = read.csv("traits.csv") #read a table, values separate by commas,
first row includes names of the columns
data<-data[match(tree$tip.label, data$taxon),]  #sort the data according to
the order of tips in the tree
data1<-as.vector(data$app) #save values (in a column 'app') as a vector
names(data1)<-data$taxon #assign taxon names (in a column 'taxon') to the
trait values in data1
```

a) Mk model

Fit the model with equal rates of changes (ER) and print the results. The results include model likelihood, the transition matrix, and marginal ancestral likelihoods:

```
fitER <- rerootingMethod(tree, data1, model = "ER")
print(fitER) #print the results (likelihoods for each node)
```

Following command will set unique colour values to all trait values, plot the tree, add pies with marginal ancestral likelihoods to each node and add coloured points indicating state of each tip:

```
cols<-setNames(palette()[1:length(unique(data1))],sort(unique(data1))) #set
unique colour values
plotTree(tree,type="phylogram",fsize=0.8,ftype="i") #plot the tree
nodelabels(node = as.numeric(rownames(fitER$marginal.anc)), pie =
fitER$marginal.anc, piecol = cols, cex = 0.5) # add pies with marginal
likelihoods to each node
tiplabels(pie = to.matrix(data1, sort(unique(data1))), piecol = cols, cex =
0.2) #add tip states to the tree as colour dots
```

b) stochastic character mapping

Simulate single stochastic character map using empirical Bayes method:

```
mtree <- make.simmap(tree, data1, model = "ER")
```

Plot this simulation and add legend:

```
cols<-setNames(palette()[1:length(unique(data1))],sort(unique(data1)))
plotSimmap(mtree, cols, pts = FALSE, lwd = 3) #plot the stochastic
simulation
```

```
add.simmap.legend(colors = cols, vertical = FALSE, prompt = FALSE, x = 0, y
= 24) #add legend
```

Simulate 100 stochastic character maps and plot them:

```
mtrees <- make.simmap(tree, data1, model = "ER", nsim = 100)
par(mfrow=c(10,10)) #prepare 10x10 plot
null<-sapply(mtrees,plot,colors=cols,lwd=1,ftype="off") #plot all
stochastic maps
```

Summarize all the stochastic maps and plot the summary:

```
pd<-summary(mtrees,plot=FALSE) #summarize a set of stochastic maps
plot(pd,fsize=0.6,ftype="i") #plot the summary
```

Plot a random map and overlay with the posterior probabilities:

```
plot(mtrees[[1]],cols,fsize=0.8,ftype="i")
nodelabels(pie=pd$ace,piecol=cols,cex=0.5)
tiplabels(pie = to.matrix(data1, sort(unique(data1))), piecol = cols, cex =
0.2) #add tip states to the tree as colour dots
add.simmap.legend(colors=cols,prompt=FALSE, x=9, y=10, fsize=0.8)
```

3. Phylogenetic generalized least square method

This method uses the historical relationships among samples (phylogenies) to test the relationships between variables.

First, compute ordinary least square (OLS) regression to assess the relationship between two continuous traits. In this example we look for the correlation between ‘cvalue’ and ‘max’. Before computation we look at the distribution of the data (and of their logarithmic transformation):

```
data = read.csv("traits.csv") #read the data
par(mfrow = c(2,2)) # plot in a 2x2 window
hist(data$cvalue)
hist(data$max)
hist(log(data$cvalue))
hist(log(data$max))
par(mfrow = c(1,1)) # reset graphical parameters to a 1x1 plot window
```

Compute the ordinary least square regression:

```
model.ols = lm(log(cvalue) ~ log(max), data=data) #compute OLS
summary(model.ols) #print statistical summary
```

And plot it including the regression line:

```
plot(log(cvalue) ~ log(max), data=data) # plot the OLS regression
abline(model.ols) # add the regression line
```

Before we calculated PGLS we need to combine the phylogeny and data into a ‘comparative data’ object (vcv=TRUE stores a variance-covariance matrix of the tree):

```
mypgls = comparative.data(phy=tree, data=data, names.col=taxon, vcv=TRUE,
na.omit=FALSE, warn.dropped=TRUE)
```

Now we can fit a model with the maximum likelihood estimate of lambda and print the summary:

```
model.pgls = pgls(log(cvalue) ~ log(max), data=mypgls, lambda="ML")
```

```
summary(model.pgls)
```

As a result you can compare regression fit of the two variables by comparing R-squared and p-values for OLS and PGLS.

DNA sequences submission to GenBank

Eliška Záveská, Faculty of Science UK, Prague
version 2017-01-11

Submission of DNA Sanger various types of seqeunces into GenBank database using software Sequin.

I. Preparation of the sequence and protein alignments [Excel]

II. DNA sequence submission via Sequin [Sequin]

When you plan to present the results based on the sequence data in any peer-reviewed journal you will need to insert your sequences in some publicly available database of nucleotide sequences (and their protein translations) and refer the database accession number of your sequence in your paper. On of the commonly used publicly available and open access sequence database is GenBank produced and maintained by the National Center for Biotechnology Information (NCBI), accessible on https://www.ncbi.nlm.nih.gov/.

There are generally two options how to submit your sequence into the database. Either using online tool **BankIt** (which is currently unavailable) or the stand-alone submission program **Sequin**. Here we will use Sequin to prepare the file with all necessary informations about your sequences which can be then sent by email to the GenBank staff. Upon receipt of a sequence submission, the GenBank staff examines the originality of the data and assigns an accession number to the sequence and performs quality assurance checks. When everythink is going well, you will receive back your sequence accession numbers.

Depending on the type of the sequence and the amount of the samples there might be various ways how to use sequin most effectively. For submission of one simple sequence (e.g. without any coding region included) you can perhaps annotate the sequence manually, while for tens or hundreds of samples you will need some more advanced tools in Sequin. Generally recommendable is thorough preparation of the sequence alignment that might save some additional annotation using Sequin as will be shown.

I. Data preparation

Before you start with preparation of your sequences you have to know following characteristics of your organism and your sequences:

- Where did you get your sequence from?
    - What organism did you sequence?
    - Optionally, where, when, and how did you obtain your organism?

- Is it a genomic sequence or obtained from mRNA?
- Is it from the genome or a plasmid, plastid, or mitochondrion?
- What gene(s) does it represent? Did you sequence the whole gene or just a part?
- Did you sequence gene coding region or non-coding part of the gene?

Most probably you will know information about your studied organism, but if you are not sure about characteristics of the sequence region you analyzed the easiest way how to get to know it is to use BLAST on https://blast.ncbi.nlm.nih.gov/Blast.cgi.

We will use as an exemplar data multiple sequence alignment of low-copy gene region in fasta format. It is placed in

/lesson_6/sequin/Exemplar_data/GLO3_full_Final_s_H_aln.fas

To explore it you can open it in any sequence editor and it will be good to open the dataset also in the text editor.

- Copy the first sequence from the file and find the most similar sequence that is already present in GenBank using online tool BLAST, **https://blast.ncbi.nlm.nih.gov/Blast.cgi**). Open the record with most similar sequence and explore it properly. It should look like example below. Try to answer the question mentioned above (i.e. what organism have been sequenced? Is it a genomic sequence or obtained from mRNA? Is it from the genome or a plasmid, plastid, or mitochondrion? What gene(s) does it represent? Did you sequence the whole gene or just a part? Is it gene coding sequence or non-coding part of the gene?

```
LOCUS       KU896003                578 bp    DNA     linear   PLN 15-MAY-2016
DEFINITION  Curcuma bhatii voucher JLS 73446 GLOBOSA-like MADS box
            transcription factor (GLO3) gene, partial cds.
ACCESSION   KU896003
VERSION     KU896003.1
KEYWORDS    .
SOURCE      Curcuma bhatii
  ORGANISM  Curcuma bhatii
            Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
            Spermatophyta; Magnoliophyta; Liliopsida; Zingiberales;
            Zingiberaceae; Curcuma.
REFERENCE   1  (bases 1 to 578)
```

AUTHORS Zaveska,E., Fer,T., Sida,O., Marhold,K. and Leong-Skornickova,J.

TITLE Hybridization among distantly related species: Examples from the polyploid genus Curcuma (Zingiberaceae)

JOURNAL Mol. Phylogenet. Evol. (2016) In press

PUBMED 27090448

REMARK Publication Status: Available-Online prior to print

REFERENCE 2 (bases 1 to 578)

AUTHORS Zaveska,E., Fer,T., Sida,O., Marhold,K. and Leong-Skornickova,J.

TITLE Direct Submission

JOURNAL Submitted (09-MAR-2016) Department of Botany, University of Innsbruck, Sternwartestr. 15, Innsbruck 6020, Austria

COMMENT ##Assembly-Data-START##

Assembly Method :: MAFFT v. v.6

Sequencing Technology :: Sanger dideoxy sequencing

##Assembly-Data-END##

FEATURES Location/Qualifiers

source 1..578

/organism="Curcuma bhatii"

/mol_type="genomic DNA"

/specimen_voucher="JLS 73446"

/db_xref="taxon:251768"

/note="authority: Curcuma bhatii (R.M. Sm.) Skornick. & M. Sabu."

gene <1..>578

/gene="GLO3"

mRNA join(<1..21,106..150,563..>578)

/gene="GLO3"

/product="GLOBOSA-like MADS box transcription factor"

CDS join(<1..21,106..150,563..>578)

/gene="GLO3"

/codon_start=1

/product="GLOBOSA-like MADS box transcription factor"

/protein_id="AND67348.1"

```
                     /translation="WKMHKKNENFLEEENKQLTYMLHHHQL"
ORIGIN
        1 tggaagatgc acaagaagaa tgttagcata catattccct cgagctttaa tttgttgaat
       61 gcttaatttt gtttctgttc ctaatggttt tcatctctcg aataggaaaa ttttctagag
      121 gaggagaaca agcaactgac ttacatgctg gtaatcttct tagcaattga tgtctcatag
      181 tttgggtgtg tttcacagct taccgatgtt tatacctgtt tcattattca caattttcag
      241 taaccagtag ataatcctag tccaaatgtt ttagcacttg tggaacacat tattttataa
      301 ttctcctaga ttcatagatt ttatagaatt acacatttta gagaactatt ttagcacctc
      361 tagcaagttt tctatcaaag gaagccaact tgtgtaccct aattgctctt gtagttaacc
      421 taatggaact tttagaagct agaatgggca tctctagcta tgcataatca ggcaattctt
      481 gctctaatat gtaattctag agcacttcca ttcaatctcc ctgtttattg atgcgtgtta
      541 gaaaatttct ttgtaaattt agcaccatca tcaactgg
//
```

- By finishing the exploration you could figure out that the exemplar sequence comes from plant genus *Curcuma*, it is from nuclear (i.e. genomic) DNA and it is part of the gene similar to GLOBOSA MADS box transcription factor. The sequence includes coding as well as non-coding part.
- While we can have also simplier cases where sequence is just a part of non-coding sequence and we do not have to know anything about it's coding part (e.g. some intergenic regions of plastid DNA), on this example we will show the way how to rather easily anotate non-coding as well as coding part of the sequence which is compulsory prior the sequence submisison.

- By comparison of your sequence and the most similar sequence in the gene bank you infer what part of your alignment is the coding region and which part is non-coding. In the exemplar file and exemplar GenBank record it should be easy as the GenBank record refer to the exactly same region as in the exemplar alignment. **Be aware** that in the GenBank record the coordinates that indicate where coding region starts en ends (i.e. 'CDS join(<1..21,106..150,563..>578)') refering to the sequence without gaps (but in the exemplar alignmnet file you have some gaps)! I recommend to reconstruct based on GenBank record how the nucleotide sequence of the coding region should look like. In our case it is
  'TGGAAGATGCACAAGAAGAATGAAAATTTTCTAGAGGAGGAGAACAAGCAACTGACTTACATGCTGCACCATCATCAACTGG'
  When you translate it to the protein you should get 'WKMHKKNENFLEEENKQLTYMLHHHQL' which is the protein sequence that is also part of the GenBank record for the exemplar sequence.
- You should be able to find the borders of the coding the non-coding region in the exemplar alignment with these informations. However, sometimes you can have

troubles find closely related sequences to your seqeunce in GenBank and then the situation becoming more challenging.

- When you find out what part of your alignmemts is the coding part **prepare a new alignment which will include only the coding sequence**. You can try to translate the entire alignment into the sequence in order to check if there are no stop codons or some other unexpected characteristics of the coding region.
- Once you have one alignment that includes your entire sequence and the alignment that includes only coding part of the sequence we can proceed to the preparation of the sequence headers that have to include '**unique SeqID'** and several informations about the studied organism and type of DNA. These additional informations can be added via so called **'modifiers'** . The list of all modifiers that you can specify is here: https://www.ncbi.nlm.nih.gov/Sequin/modifiers.html
- **At minimum, the scientific name of the organism should be included as a modifier.** For your sequence description use only modifiers for which you have really reliable information.
- For the exemplar data we will use four modifiers for nucleotide sequences
  - [**organism**=Curcuma bhatii]
  - [**mol_type**=genomic DNA]
  - [**specimen_voucher**=JLS 73446]
  - [**authority**=(R.M. Sm.) Skornick. & M. Sabu.]

and two modifiers for protein sequences

  - [**gene**=GLO3]
  - [**protein**=GLOBOSA-like MADS box transcription factor]
- For the preparation of the headers for all individuals in the alignment you can use for instance Excel. The exemplar excel sheet with the seqeunces and their new headers is in 'Exemplar_data' folder.

- TO SEEK FOR MORE INFORMATIONS ABOUT HOW YOUR INPUT DATASET SHOULD LOOK LIKE CHECK THE Sequin HELP PAGE HERE: **https://www.ncbi.nlm.nih.gov/Sequin/sequin.hlp.html#SequenceFormatForm**
- As it states in the Sequin help: In FASTA format the line before the nucleotide sequence, called the **FASTA definition line**, must begin with a carat (">"), followed by a **unique SeqID** (sequence identifier, which migh be e.g. your original collection number). The SeqID must be unique for each nucleotide sequence and should not contain any spaces. Please limit the SeqID to 25 characters or less. Use of brackets ("[]") in the SeqID is also prohibited. The identifier will be replaced with an Accession number by the database staff when your submission is processed.
- **Information about the source organism** from which the sequence was obtained **follows the SeqID** and must be in the format **[modifier=text]**. Do not put spaces around the "=".

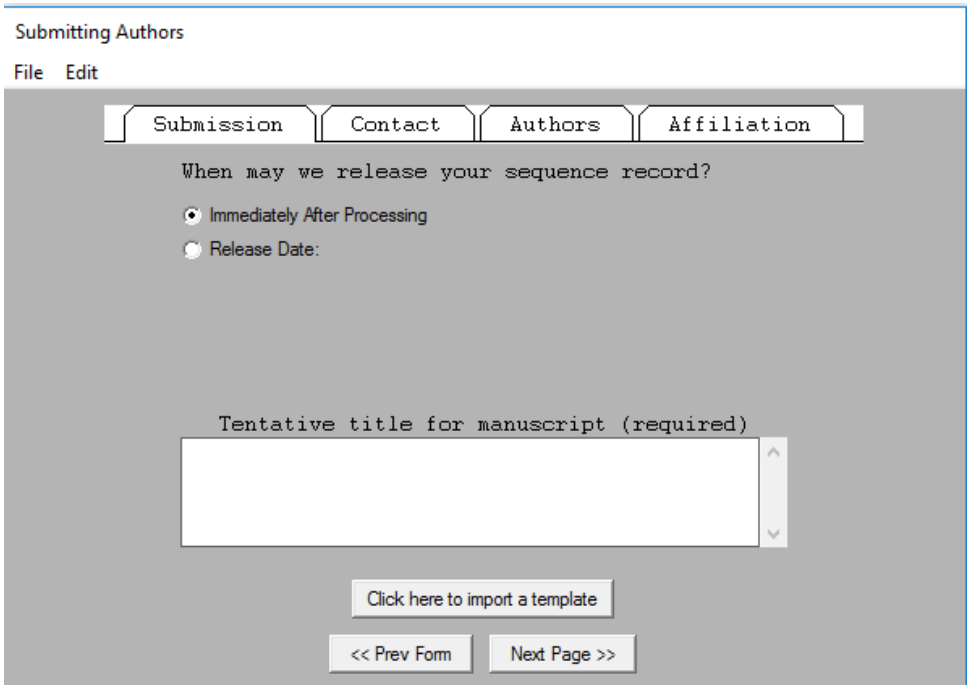
At minimum, the scientific name of the organism should be included. Optional modifiers can be added to provide additional information.

II. DNA sequence submission via Sequin

As mentioned above for the DNA sequence submisison we will use only reliable stand-alone submission program **Sequin**.

- programs:
  - Sequin, https://www.ncbi.nlm.nih.gov/projects/Sequin/download/seq_win_download.html
- test datasets
  - GLO3_full_Final_s_H_aln_with_modifiers_Edit.fas
  - GLO3_full_Final_s_H_protein_with_modifiers_Edit.fas
  - placed in folder /lesson_6/sequin/Exemplar_data/

  - Open **Sequin**
  - Check that ‚Database for submission' is set up to ‚GenBank'
  - Click on ‚Start New Submission' and new window will be opened with several tabs. You will have to go through each of them to introduce details about when you want your data to be open tu public (tab ‚Submission'), your contact information (tab ‚Contact'), details about all the all the authors of the study where your data are used/presented (tab ‚Authors') and your affiliation (tab ‚Affiliation'). All those informations are compulsory. In tab ‚Submission' you ususally want to set up the date when your sequences will be released to be after your paper is accepted. You can do so by inserting your desired date. When you fill in all your details it is good to export the ‚Template' that you just created using button ‚**Click here to export a template**' in the ‚Affiliation' tab. It can happend that you failed to finish the submission process and this template will be helpful when you have to start submission process denovo. When you finish with the ‚Affiliation' tab, click ‚Next form'. When some error or info appears try to fullfill its request.



- Exemplar templates are given in folder /lesson_6/sequin/Exemplar_data/submission_details/

- In the newly opened window ‚Preparing the sequences' select the option ‚Use the normal submission dialog' and continue with ‚Next'.
- In the next window select in the Submission Type section ‚Phylogenetic study' which enables you to select option ‚Alignment' in Sequence data format section. You might try to explore also other options (e.g. ‚Population study'), but if you have a data for phylogenetic study and data prepared in form of alignment (e.g. with gaps) in FASTA format this will be good choice. You can also consult the Sequin help

https://www.ncbi.nlm.nih.gov/Sequin/sequin.hlp.html#SequenceFormatForm

As a Submission cathegory select ‚Original submission'.

- In next window, in the tab ‚Nucleotide', click on ‚Import nucleotide alignment' and browse to find your alignment, e.g. exemplar fasta format with sequences of GLOBOSA3 gene

GLO3_full_Final_s_H_aln_with_modifiers_Edit.fas

If you will use the exemplar file warning window will pop up to suggest you that you should trimm some of your data. If it happend with your sequences, you should check your alignment and decide how seriously gappy your data are. If you check the exemplar data, there are some missing data, but you can proceed without trimming the sequences. Generally, I would sugget to make all the alignment changes before you start with Sequin

- You can check newly pop-uped window ‚Alignment reading summary' that is listing all headers of sequences that you want to submit and close it.
- You can specify characters for missing data and gaps by clicking on the button ‚Optional alignment settings'

- As you have most of the necessary information about the sequence in the sequence header you do not have to specify other characteristics in this window.
- Switch to the tab ‚Sequencing Method' and tick ‚Sanger dideoxy sequencing'. If your sequences are assembled (e.g. from forward and reverse primer) indicate this in the question below and enter also the name of the assembler you used.
- Click ‚Next Page' taht will brings you to window with three tabs ('Organism', 'Proteins' and 'Annotation'). In the tab 'Organism' you can see what modifiers you specified for your sequences. Perhaps you could add some more modifiers using the buttons in the bottom part of the tab, but I recommend to use as less tools of Sequin as possible, because at least for me the behavior of the program is sometimes hard to understand and it is not always possible undo your changes.
- If you studied region includes coding sequence switch to tab 'Protein' and using the button 'Import Protein FASTA' browse to your pre-prepared protein alignment (see the necessary characteristics of this file in the section 'Data preparation'). If the coding region(s) is (are) placed within your studied region you do not have to take care anything else in this tab. If the coding sequence is incomplete at the beginning or at the end of your sequence then you should specify this by ticking 'Incomplete at NH2 end' and/or 'Incomplete at CO2H end'.
- Switch to tab 'Annotation' and check the options what you have here. If you prepared you protein file properly, you actually do not have to add any additional information here and click 'Next Form'.
- In newly opened window you will see how the GenBank record will look like for all your samples. You can switch between the samples in the upper most window with scroll bar (section 'Target Sequence'). In our exemplar data we can see that in the all records in the section 'Organism' there is still note 'Unclassified'. This we can change by clicking on the tab 'Annotate '→ 'Batch Feature apply' → 'Add Source Qual'. In newly opened window go to the tick the button 'Taxonomy', select from scroll menu 'Lineage' (or anything what you want to specify) and add the text in the field 'Text'. In our example the appropriate description of the lineage where the organism analyzed would be:

  ```
  Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta; Spermatophyta; Magnoliophyta; Liliopsida; Zingiberales; Zingiberaceae; Curcuma.
  ```

- In the tab 'Annotate' you can find many other useful tools using which you can change information either in a batch manner or for particular individuals. Explore it individually.
- When you are satisfied with the record preview click the button 'Done'. Usually you will get some error or warning messages, but clicking 'Review errors' you will get more information about severity and what really need to be changed. In the upper part of the newly opened window you can filter type of the message. Keep type of message 'Normal; and in Severity filter select 'ERROR'. These types of errors are rather lethal and you would probably have problems during the further submission process. Therefore you will need to solve them. In our example there is only one error message that is pointing out the difference in length of the given protein

sequence and its expected length based on the nucleotide alignment. In such a cases you have to check both of your alignments and figure out, what is causing the problem. As there might be tons of different types of error messages it is impossible to cover them here. Only recommendation is to understand the error message and make the correction either in your source alignment(s), modifiers or annotations. If you cannot find the solution by yourself, you might contact GenBank staff got help or just to submit your sequences with errors and wait if GenBank staff will write you back with suggestions what should be changed.

- When you filter only for 'WARN', i.e. Warning messages, you probably get much more records but you can proceed without making any changes. When the GenBank staff will found the cause for the warning message serious they will contact you after you finally submit the data (you send your output of the Sequin by email to the GenBank staff that is taking care about sequence data). In our exemplar data the source of warning is pointing out that our 3' end of the coding region doesn't have a proper features of the endings of the coding regions (i.e. stop codon is missing, etc.). Again we can recheck the nucleotide and protein alignments if there was not some batch error in sequence reading but if we are sure that our sequence are of good quality it might be better to leave warning message without taking care about it. But take this recommendation with some reserve.
- To solve the error messages you can either use the tool from the menu bar, e.g. 'Edit' → 'Alignment assistant' or 'Edit' → 'Edit sequence' or you can make a changes in your original alignment (but then you will have to repeat entire submission process).
- When you think the Errors (and possibly also Warnings) are solved you can press the button 'Done' again and instead of checking error messages click 'Continue'. You will be asked if you are ready to save the data. Click 'yes' and save the Sequin output with the suffix *.sqn.
- Check file *.sqn using some text editor (e.g. notepad) and make sure that it includes information about all your samples. It should looks like the exemplar file 'Globosa_final_1_error_71484_C_montana.sqn' given in the folder /lesson_6/sequin/Exemplar_data/.
- This file you can send to the GenBank team on the suggested email address: gb-sub@ncbi.nlm.nih.gov
- You will be informed by GenBank staff about further steps.