# MolD

**Version 1.4**

## Manual

21.10.2022

**Author:** Alexander Fedosov <fedosovalexander@gmail.com>

## Disclaimer

## Table of Content

## Citation

**Fedosov A.E.**, Achaz G., Gontchar A., Puillandre N. 2022. MOLD, a novel software to compile accurate and reliable DNA diagnoses for taxonomic descriptions. *Molecular Ecology Resources*, DOI: 10.1111/1755-0998.13590.

## Operating systems

Mac OSX, Windows, and Linux are supported

## System requirements

Python 3 installed (standard distribution) or Python 2.7 installed (original code – upon request)

## Input

### DATAFILE

The input file is in fasta format: each entry starts with the identifier line, and one or more lines of nucleotide sequence. Identifier line starts with '>' and must contain two parts, separated by a pipe ('|') symbol. The first part is a free-style **sequence identifier**, the second is the **taxon identifier** of the query level. The names of the taxa to be diagnosed correspond to the **second** element.

1. query ID example:

>GBXXXXXXX|query

2. reference ID example:

>GBXXXXXXX|ref1

### *EXAMPLE*

[Species of the cone-snail genus *Conasprella* (Gastropoda) – Puilllandre et al. 2014]

```
###############################################
>Conasprella_alisi1|alisi
TATAAGATTTTGGCTTTTACCTCCTGCCCTTCTTTTACTCCTTTCTTCAGCT
>Conasprella_alisi2|alisi
TATAAGATTTTGGCTTTTACCTCCTGCCCTTCTTTTACTCCTTTCTTCAGCT
>Conasprella_alisi3|alisi
TATAAGATTTTGGCTTTTACCTCCTGCTCTTCTTTTACTCCTTTCTTCAGCT
>Conasprella_baileyi1|baileyi
TATAAGATTTTGACTTTTGCCTCCGGCCCTTCTTTTACTTCTTTCTTCAGCC
>Conasprella_baileyi2|baileyi
TATAAGATTTTGACTTTTGCCCCCGGCCCTTCTTTTACTTCTTTCTTCAGCC
>Conasprella_boholensis|boholensis
TATAAGATTTTGACTTTTACCTCCTGCGCTTCTTTTACTTCTTTCTTCAGCT
>Conasprella_boucheti|boucheti
TATAAGATTTTGACTTTTACCTCCCGCACTTCTTTTACTTCTTTCTTCAGCT
>Conasprella_comatosa|comatosa
TATAAGATTTTGACTTTTACCTCCTGCGTTGCTTCTACTCTTATCTTCAGCT
>Conasprella_coriolisi|coriolisi
TATAAGATTTTGACTTTTACCCCCTGCGTTGCTTCTACTCCTATCTTCAGCT
###############################################
```

**Please, check that:**

1. Each identifier line has only one pipe symbol.
2. There are no spaces in the sequence and taxon identifiers.
3. All taxa identifiers are provided **and correct**.
4. Sequence lines only contain valid nucleotides ('A', 'C', 'G', 'T'), gaps ('-'), and ambiguous nucleotides ('N', 'K', 'M', 'R', 'S', 'W', 'Y')

**Please, note that:**

1. Any data file extension (.fas / .fa / .fasta / .txt etc.) will do the job.

## PARAMETERS INPUT

Either entered in the web-interface, or (if a command-line implementation is used) provided in the parameter file after '='.

### *1. INPUT / OUTPUT FILES*

-INPUT_FILE – input alignment file with complete path.
-OUTPUT_FILE – output file with complete path (only command-line version)

### *2. INPUT PARAMETERS* (NO DEFAULTS - no parameters entered will lead to an error).

qTAXA (**Query taxa**)

**ALL** -if all taxa in the dataset are to be diagnosed.

**>N** -if all taxa with more than N sequences available (where N is a natural number) to be diagnosed;

**Taxon1,Taxon2...** -a comma separated list of taxa to be diagnosed. **Please check** that all taxa identifiers are provided as in the input alignment;

**Taxon1+Taxon2...** -will combine all records of the taxa Taxon 1 and Taxon 2 (or more) in a combined taxon **Taxon1+Taxon2** and diagnose it. Any number of taxa, less than a total number of taxa in a dataset may be combined. Any number of combined taxa may be diagnosed in one run. Combined taxa can be diagnosed in one run alongside with not combined taxa;

**P:pattern** -if only taxa with certain pattern in the taxon name are to be diagnosed;

**P+:pattern** -merges all taxa with certain pattern in the taxon name in one combined taxon and diagnoses it;

**Taxon1VSTaxon2** -identifies diagnostic single nucleotide mDNCs that differentiate **Taxon1** from **Taxon2**; output is written to a separate file '*Basename_pairwise.out'* derived from the provided name of the output file.

**Taxon1VSALL** -identifies diagnostic single nucleotide mDNCs that differentiate **Taxon1** from every other taxon in a dataset; output is written to a separate file '*Basename_pairwise.out*';

**ALLVSALL** -identifies diagnostic single nucleotide mDNCs for all possible taxa pairs in a dataset; output is written to a separate file '*Basename_pairwise.out*';

**Multiple** input patterns involving taxa names or 'ALL' can be run together:
**Taxon1, Taxon2, Taxon1+Taxon2, Taxon1VSTaxon2** – will work;
**All, AllVSALL** – will work too.

`Taxon_rank`

| | |
|---|---|
| **1** | for species |
| **2** | for supraspecific taxa |

`Code gaps as characters` **(whether alignment gaps are used as a character or not)**

| | |
|---|---|
| **Yes** | dashes ('-') in the alignment are transformed into 'D', which is treated as an independent characters |
| **No** | dashes are treated as missing data ('N') |

### *3. ADVANCED PARAMETERS FOR mDNC RECOVERY*

[For explanation see '*Review of MolD*' below or *Fedosov et al. 2019*. If you do not want to set themleave respective fields blank, and the defaults will be used.]

`Cutoff`

| | |
|---|---|
| -integer (default **100**) | denotes the **number of informative sites** to be considered for inclusion into a mDNC; |
| -integer prepended by ('>') | Informative sites are ranked based on how many sequences of reference taxa differ from the query in the nucleotide at each site (The **cut-off value**). If this option selected, all informative sites with cut-off value above specified after ('>') will be considered. If '**>0**' is used **all** informative sites will be considered for inclusion into mDNC. |

`NumberN` Number of ambiguously called nucleotides allowed, integer (default **5**).

**If value of NumberN would lead to the discard of more than half read-in sequences**, the alignment length will be adjusted to the median sequence length, and the alignment will be trimmed automatically to remove start and end columns mainly represented by 'N's. If this leads to the change in the base indexing, the site indexes in output DNCs will match the base indexing in the original (untrimmed) alignment.

`Number_of_iterations` Number of iterations of MolD, integer (default **10000**).

`MaxLen1` Maximum length of draft DNCs, integer (default **12**).

`MaxLen2` Maximum length of refined mDNCs, integer (default **7**).

`Iref` Indexing reference sequence (default **NO**) sets the 0 index, so that the sites indexes in DNCs correspond to the indexing of the reference. To set a sequence as reference, enter its identifier, and after a comma (,) 'incl' or 'excl', which will instruct MolD to either use the sequence as a part of the dataset, or solely as a indexing reference.

### *4. PARAMETERS OF ARTIFICIAL DATASETS (only rDNSs).*

`Pdiff` Percent difference between original and modified sequences, integer (default **1** for species-level taxa, **3** for for supraspecific taxa).

`NmaxSeq` Max number of sequences per taxon to modify, integer (default 10).

`Scoring` To score each candidate rDNC, 100 simulated test datasets are created. If rDNC remains valid in a test dataset, it adds 1 to the score, so lowest possible score is 0 and highest is 100. If two consecutive scores are above the threshold value defined by a **keyword argument** here (default is `moderate`) the rDNC is output. Arguments:

| | |
|---|---|
| `lousy` | 66 |
| `moderate` | 75 |
| `stringent` | 90 |
| `very_stringent` | 95 |

## Run

Run in terminal:

```
python /PATH_TO/MolD_v1.4.py –i /PATH_TO/MolD_parameter_file
```

or

```
python3 /PATH_TO/MolD_v1.4.py –i /PATH_TO/MolD_parameter_file
```

## Review of the MolD algorithm

For term definition and theoretical background see:

**Fedosov A.E.**, Achaz G., Gontchar A., Puillandre N. 2022. MOLD, a novel software to compile accurate and reliable DNA diagnoses for taxonomic descriptions. *Molecular Ecology Resources*, 22(5): 2038-2053. DOI: 10.1111/1755-0998.13590.

The MolD algorithm is divided into five consecutive steps. At **first** step sequences are sorted by taxon (as defined by the taxon identifier of the input) and the sites conserved within each taxon are identified.

At the **second** step, each of the sites shared by all query taxon sequences is assigned a ***cut-off*** value, which corresponds to the number of reference taxa sequences in the alignment with different nucleotide at this site. The sites that are conserved across the entire data set have a minimum cut-off value of 0 (i.e. non-informative). The sites that correspond to Type 1 characters (see Fedosov et al. 2019, Fig. 1) immediately differentiate the query, and have a maximum cut-off value.In this case the cut-off value equals to the total number of reference taxa sequences in the data set. Either the desired size of this subset (parameter **cutoff,** by default set to 100), or the threshold cut-off value (>N) can be set by user.

The **third** step contains main functionality of the MolD algorithm implemented in two piped core functions. The `step_reduction_complist` function initiates a draft DNC, and extends it by picking up informative sites one-by-one in random order and appending to the draft DNC. For each picked site the reference taxa sequences that differ at this site from the focus taxon sequences are identified and excluded from further comparisons. The list of reference taxa sequences that share a draft DNC with the query is thus reduced step-by-step until its length equals zero. This is a condition at which the function terminates, and the draft DNC is output, if it comprises no more than a predefined number of sites (parameter ***Maxlen1***, default 12). The draft DNC allows unambiguous differentiation of the query taxon members in the analyzed data set, but it usually includes more sites than necessary. So, the draft DNC, is refined by the `RemoveRedundantPositions` function. This function removes redundant sites from the draft DNCs by picking and discarding sites successively one-by-one, and each time checking whether the thus shortened combination remains diagnostic for the query or not. Once the draft DNC cannot be further refined, it constitutes an mDNC, and is sent to output, if its length is equal to or less than a pre-defined (parameter ***Maxlen2***, default 7). Each of the mDNCs defines a minimal and sufficient condition for a nucleotide sequence (and a corresponding specimen) to belong to the query taxon. Single execution of the two core functions is termed a **search iteration**; each search iteration supplies one mDNC in the case that length criteria are met. By default, MolD run runs 10,000 search iterations, but their number can be set by a user (parameter ***Number_of_iterations***) to generate a pool of mDNCs. The list of non-identical mDNCs sorted by length constitutes the output of the third step.

Two mDNCs may overlap by one or several sites, or share no sites; in the latter case the two mDNCs are termed independent mDNCs (see Fedosov et al. 2019). In the case that all identified mDNCs share one or more sites (i.e. no independent combinations are identified), such site(s) present in all mDNCs are termed **key positions**. The key position(s) are crucial for diagnosing a taxon, because a substitution at this site even in one sequence attributed to a query would immediately make the query-taxon impossible to diagnose with the selected genetic marker. On the contrary, when *n* independent mDNCs are recovered, *n* substitutions would be needed to make the query taxon undiagnosable; the likelihood of the latter scenario is obviously much lower. At the **fourth** step the set of mDNCs is analyzed to identify independent mDNCs, or (if present), key position(s). In the case that no mDNCs were recovered for a pre-defined set of DNA sequences, an exception is raised.

At the **fifth** step the set of mDNCs is converted into rDNC that fulfills pre-defined requirements of robustness. An rDNC is constructed from the list of mDNCs produced by MOLD in the step 3. First, mDNCs are sorted by increasing lengths, and mDNCs of the same length are 'binned'. In each bin, a given site can be shared by several mDNCs. MOLD computes for each site in each bin, its frequency of occurrence. Sites with frequency 1 are present in all mDNCs of the bin. Sites are thus double sorted, first by the mDNC length and then by frequencies. The top sites of this ranking have the highest frequency among the shortest mDNCs. If Type 1 characters exist for a query, they make the top of ranking, as they are considered as the DNCs of the length 1.

A new rDNC is seeded using one random mDNC among the shortest ones (i.e. Type 1 characters when they exist in the list). Then, extra sites are picked from the top of the double-sorted list of sites and are added to the rDNC one-by-one. After each addition of a site, the rDNC is scored for reliability (see below), and the score is recorded. The rDNC extension process stops, either when two successive scores exceed the user-defined reliability threshold (parameter **Scoring**) - then the best-scoring rDNC is sent to output, or when the rDNC reaches 10 nucleotide sites. In the latter case, if at any step an rDNC has scored above the reliability threshold, it is output with a warning. If the scores remain consistently below the reliability threshold, a message is output to inform the user that no sufficiently reliable rDNC could be constructed.

To evaluate an rDNC after each step of elongation, MOLD uses simulated datasets that are generated by altering the original alignment with artificial mutations. MOLD repeatedly creates **test datasets** with artificial sequences derived from the real ones. Each artificial sequence is generated by introducing *p* nucleotide substitutions into an existing sequence, where *p* is a random uniform integer in [1, x*L*/100], where x is the parameter **Pdiff** (default **1**), and *L* is the alignment length. Mutations are introduced only at polymorphic sites by substituting the original nucleotide into one of the three others, selected randomly with respect to their observed frequencies at this site in the original alignment. For each species of the original alignment, k randomly sampled sequences (parameter **NmaxSeq**, default **10**) are artificially created by mutation. For species with more than 10 sequences in the original alignment, randomly sampled sequences with no mutation are added to the test dataset to match the original number of sequences for this species. Therefore, a test datasets includes at least 10 sequences per species.

For each rDNC evaluation step, MOLD generates 100 new test datasets. For each of them, the rDNC under evaluation scores 1 if it unambiguously delimits the query taxon (unique combination defining the query taxon) or 0 otherwise. So, DNC score ranges from 0 (if rDNC failed in all 100 test datasets) and 100. Importantly, MOLD tolerates one discordant site when evaluating whether the query taxon is correctly diagnosed in a test dataset - if all but one sites delineate the query unambiguously, it scores 1.

Thus the rDNC is output as a final DNA diagnosis if:

- rDNC scores **66+** in two consecutive runs, and the Scoring is set as **lousy**, or
- rDNC scores **75+** in two consecutive runs, and the Scoring is set as **moderate**, or
- rDNC scores **90+** in two consecutive runs, and the Scoring is set as **stringent**, or
- rDNC scores **95+** in two consecutive runs, and the Scoring is set as **very stringent.**

## Quick how to...

[some advices to help setting the MOLD run]

**First** it makes sense to run MOLD with all default settings and check whether all queries were successfully diagnosed or not. If not, one of the following issues might happen:

| **Issue** | **Reason /** Troubleshooting |
|---|---|
| No mDNCs identified for a query<br>**or**<br>Number of identified mDNC is too small (< 10) | **There is a problem with sequences attribution to taxa/** Please, check carefully taxon identifiers in query and reference records. It is **strongly recommended** to make sure that taxon identifiers correspond to clades in the phylogenetic tree.<br>**Lack or paucity of signature DNA characters/**More thorough search for mDNCs may help. Set up ‘Cutoff’ as ‘>0’ to include all informative sites in the mDNCs. If it doesn’t help, try excluding sequences containing ‘N’s at the alignment polymorphic sites.<br>If it doesn’t help, **at last resort:**<br>-If the **query is a superspecific taxon**, try splitting it into distinctive phylogenetic clusters, and providing a separate diagnosis to each of them.<br>-If the **query is a species**, it looks like you may need to look for an alternative locus, or consider deeper genomic sampling. |
| No sufficiently reliable rDNC could be identified for a query (while multiple mDNC are recovered) | **There is a problem with sequences attribution to taxa/** Please, check carefully taxon identifiers in query and reference records. It is **strongly recommended** to make sure that taxon identifiers correspond to clades in the phylogenetic tree.<br>**Too strict parameters for scoring rDNC/** Try relaxing (using lower values) each of the following parameters* in the following order :<br>**NMaxSeq** – setting it at 5 is acceptable, below is not recommen.<br>**Pdiff** – for species-level it should be either 1 or 2.<br>**Scoring** – ‘lousy’ should only be used at last resort<br>*note that by relaxing each or all parameters you compromise reliability of the resulting diagnosis. |