NSDPY: A python package to download DNA sequences from NCBI

Raphaël Hebert, Emese Meglécz

NSDPY: A python package to download DNA sequences from NCBI

Raphael Hebert and Emese Meglécz
raphaelhebert18@gmail.com, emese.meglecz@imbe.fr

Aix Marseille Univ, Avignon Univ, CNRS, IRD, IMBE, Marseille, France
Corresponding author:
Emese Meglécz
emese.meglecz@imbe.fr
Aix Marseille Univ, Avignon Univ, CNRS, IRD, IMBE
Chemin de la batterie des Lions
13007 Marseille FRANCE

Abstract.
Downloading large batches of DNA sequences can be useful to create custom databases containing for example sequences of a particular genomic region or a group of organisms. These sequences can be found on NCBI databases and accessed via a web browser (GUI) or directly via NCBI API. While the GUI is user-friendly, it lacks certain functionalities. On the other extreme, the use of the API is flexible but requires coding knowledge. NSDPY is a python package that combines flexibility and ease of use to download large amount of DNA sequences and includes several taxonomic or filtering options like batch downloading sequences for a list of taxa, downloading sequences including taxonomic lineage or filtering CDS sequences for a specific gene. NSDPY is available on PyPI, it is written to minimize dependencies on other packages and to be used directly from the terminal by simple command lines so that most users can use it without prior coding experience.

Keywords:
NCBI; Batch Download; CDS; Python; PyPI; Taxonomy

Table 1 Code Metadata

	Code metadata description	
C1	Current code version	*v.0.2.3*
C2	Permanent link to code/repository used of this code version	*https://github.com/RaphaelHebert/nsdpy*
C3	Code Ocean compute capsule	
C4	Legal Code License	*MIT*
C5	Code versioning system used	*Git*
C6	Software code languages, tools, and services used	*python*
C7	Compilation requirements, operating environments & dependencies	*Python 3.8+; Windows, Linux, Mac OS*
C8	If available Link to developer documentation/manual	*https://nsdpy.readthedocs.io/en/latest/*
C9	Support email for questions	*raphaelhebert18@gmail.com*

1. Motivation and Significance

Downloading large batches of DNA sequences is becoming a common routine for molecular biologists for a wide range of purposes such as creating custom databases for example for taxonomic assignments in DNA metabarcoding projects[1,2], accessing sequences for phylogenetic studies[3], in silico PCR primer design and testing[4].

The National Center for Biotechnology Information (NCBI) hosts a large number of databases. One of the most frequently used and largest is the NCBI nucleotide database[5]. It is composed of GenBank, which contains primary nucleotide sequences from a large range of organisms and RefSeq, which is a non-redundant, well-annotated set of reference sequences. There are different ways to download large quantities of sequences from NCBI databases. The simplest and most straightforward way is using the Entrez search engine[6] via the NCBI website (NCBI GUI). It is an easy-to-use and versatile tool, since complex queries can be easily constructed. Sequences can be downloaded in different formats (fasta, GenBank, INSDSeqXML, TinySeqXML) but only formats relatively complicated to parse (GenBank and INSDSeqXML) contain both the sequence and the taxonomic lineage, thus it is necessary to have coding knowledge to obtain the file format desired by the users. Furthermore, the NCBI GUI may timeout if many sequences should be downloaded or if the connection is unstable; thus it is not well adapted for mass downloading.

The use of the NCBI API [7] is a powerful and versatile way for downloading sequences and libraries like Biopython [8] or the rentrez package in R [9] are also available to ease its use. However, they require programming skills and thus their use is quite complex for inexperienced users. The NCBI Mass Sequence Downloader [10] is an easy-to-use option that can be used in graphical or command-line mode and it is specifically designed for mass downloading. However, it lacks some useful options such as retrieving sequences with their taxonomic lineages and sorting these sequences according to them.

The NSDPY (NCBI Sequence Downloader) python package aims to provide a solution to these problems while keeping the tool simple to use. NSDPY automatically downloads sequences based on an NCBI query sentence, eventually combined with a list of taxa of interest. It provides several options to retrieve taxonomic identifiers (TaxID) and taxonomic lineages and sorting sequences according to a given taxonomic level is also possible. Further options allow downloading Coding DNA Sequences (CDS) instead of the original fasta files and filter the results using regular expressions (a suite of characters that specifies a search pattern in text) to retain only relevant sequences if the original query cannot be sufficiently precise. The download is handled through the e-utilities to be able to handle lost connection and timeout. The script is written in python 3.8 and special care was taken to avoid unnecessary dependencies to limit maintenance issues and allow easy installation and use.

2. Software Description

2.1. Software specifications

NSDPY is a python package written in python 3.8. It uses the e-utilities tool [7] to access information in NCBI databases. NSDPY can be easily installed using PyPI (https://pypi.org/project/nsdpy/) or can be cloned directly from GitHub (https://github.com/RaphaelHebert/nsdpy) and run as a Python script. It can be freely modified since it is available under the MIT licence.

The package can be installed and run directly from a terminal so it can be directly included in a pipeline and does not require any prior knowledge of Python. Alternatively, it can be run from a Google Colab notebook to avoid using the user's own computer and connection resources and have the results available directly on Google Drive. This approach is not suitable for very large downloads as the Google Colab resources are available for a limited time and Google Drive storage limits the amounts of data that can be stored. However, it is very convenient to test, modify and run the code without having to install any third-party software.
The main functionality of NSDPY is the mass download of sequences from NCBI. To handle eventual timeouts or lost connection, NSDPY keeps track of the downloaded sequences and the sequences that are still to be downloaded. If the connection is lost, the process will wait for the connection to be re-established and resumes the download from the last correctly downloaded accession.

2.2. Software Functionalities

NSDPY requires a query sentence just as the NCBI GUI searches[11]. The users can provide complex queries that combine several different search terms (e.g. MatK[Title], chloroplast[Filter], Viridiplantae [Organism]) linked by Boolean operators (and, or, not) and structured by parentheses. The basic usage of NSDPY is to download all relevant fasta files corresponding to the query and write them in one fasta file as a result. This basic usage can be completed by several options.

The default output format is a fasta file containing accession number and the definition field of the accession. However, the tsv option (-t, --tsv) produces a text file with tab-separated columns containing accession numbers (SequenceID), NCBI's numerical taxonomic identifiers (TaxID), sequence lengths and the sequences themselves. This file is particularly easy to filter for users without programming skills.

There are several taxonomy related options. The taxids option (-T, --taxids) prompts NSDPY to write a text file with SequenceIDs and TaxIDs. The information option (-i, --information) modifies the fasta definition line to write taxon name, SequenceID, TaxID and taxonomic lineage to the output fasta definition lines. It also completes the tsv file with the taxonomic lineage when the tsv option is evoked. At last, the downloaded sequences can be sorted to different fasta files according to a selected taxonomic level using the following mutually exclusive options: -k (kingdom), -p (phylum), -s (the lowest taxonomic level in the lineage) or preparing one file for each of the taxa specified by the user with the -l option (e.g. -l Crocodylia Lepidosauria Testudines).

When a large list of taxa should be included in the input query sentence, the user can provide an input file containing the list of taxa using the -L option. The query will return all sequences that belong to any of the taxa listed in the input file and satisfy the criteria of the query sentence.

In case of coding sequences, the users can choose to download the Coding DNA Sequences (CDS) fasta files (-c option), instead of the complete fasta file. This helps to eliminate non-coding sequences and obtain the coding sequences in the correct orientation. Furthermore, the -c option can be followed by a list of regular expressions, to retain only coding sequences with at least one of the regular expressions in the definition line. This feature is particularly helpful to isolate coding sequences from long sequences

such as mitogenomes. Occasionally, due to incomplete annotations, the CDS fasta file might not contain any of the regular expressions. In these cases, NSDPY downloads the accessions in GenBank format, and tries to identify the desired coding sequences based on other fields such as note, gene synonym or product.

The workflow is illustrated in Fig. 1. The input is an NCBI Entrez query sentence, or optionally a text file with the list of taxa of interest. The main output is a fasta file with a downloaded and eventually filtered sequences with original or modified identification lines with taxonomic information included (-i option). Additionally, sequences can also be written into a tsv file together with their metadata (SeqId, TaxID, taxonomic lineage, sequence length, sequence). A summary file (report.tsv) contains the query sentence, runtime, and information on the run.

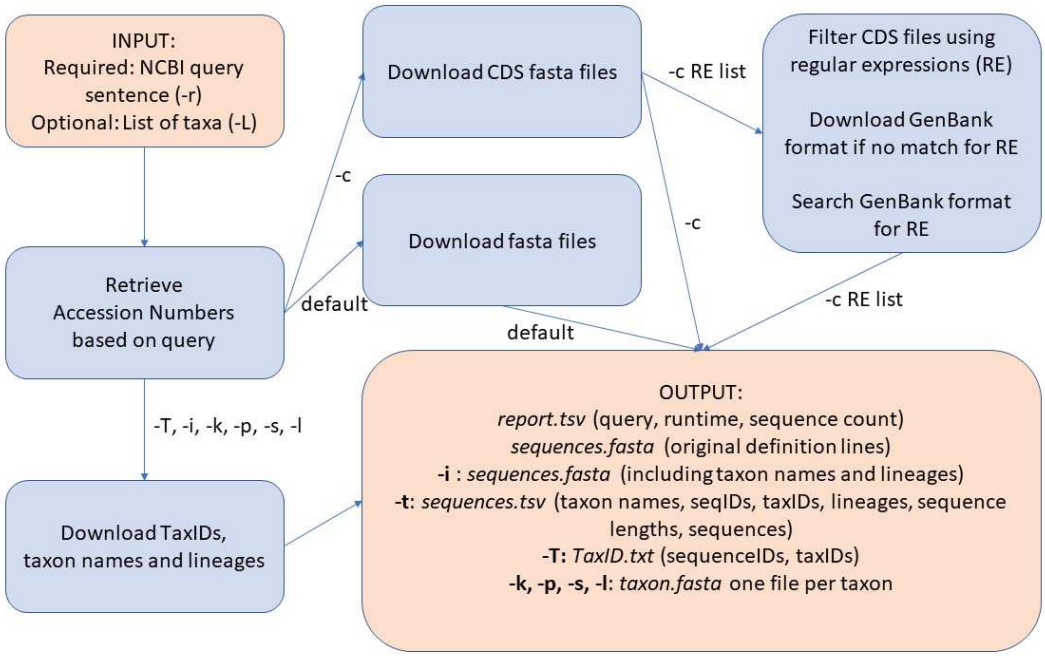

Fig. 1. Workflow of NSDPY. RE: Regular Expression.

3. *Illustrative Examples*

To illustrate the pros and cons of the script we ran it with three queries that illustrate typical uses of the script and use the main functionalities of NSDPY. We used queries that may be used to retrieve the cytochrome oxydase I (COI) sequences to make a custom database. For these examples, we installed NSDPY as a package from PyPI and ran the commands directly from the terminal.

3.1. First example

We used the following command to isolate the COI sequences from complete mitochondrial sequences:

```
nsdpy –r "(mitochondrion[Title] AND complete[Title])" –c 'cox[1i]' 'co[i1]' –v
```

In this query, the request (*-r*) helped to identify complete mitochondrial sequences. Since we are interested in extracting COI sequences from these mitogenomes, we used a *-c* (*--cds*) option to download the fasta files containing the CDS of the mitochondria, instead of the fasta files containing the complete records. Furthermore, a list of regular expressions was given after the *-c* option to filter the CDS file and to extract sequences with annotation corresponding to the provided regular expressions. When no CDS fasta file was available, or no COI gene found in it, the script downloaded and searched the accession in GenBank format (.gb). The annotation in GenBank files may be organized differently depending on the authors: in most cases, COI is annotated in the gene field (*e.g.* KY052129.1) while in others in the note (*e.g.* NC_046833.1, AY191994.1, EU273708.1), gene_synonym (*e.g.* AB475096.1,) or product (*e.g.* CP014249.1) fields.
Note that the *-v* option stands for verbose. It just allows the user to see how many percent of the sequences have been downloaded.
In short, in this query, the request helped to download complete mitochondrial sequences which would not have been found by a keyword search for COI gene and adding the regular expression helped to filter the sequences to isolate the desired gene.
The results are as follows:

- 98 398 coding sequences including 97 149 sequences identified from the fasta files containing the CDS and 1 249 sequences extracted from the GenBank files, where the COI gene was not available in the fasta file.
- 6 301 accession numbers in the notfound.txt file, where the COI gene could not be identified (*e.g*. MH500775.1).
- 252 accession numbers in the duplicates.txt file with accession numbers that returned more than one coding sequence for the COI. For example, the accession MK562756.1 has two genes annotated as COX1 in its GenBank file.

3.2. Second example

We used the following command to download all COI sequences using keywords present in the title fields and sort the sequences by kingdom:

```
nsdpy –r "(COX1[Title] OR COXI[Title] OR CO1[Title] OR COI[Title])" –k –T
```

The request (*-r*) combines a series of abbreviations frequently used to refer to the cytochrome oxidase subunit1 gene. The search is limited to the title field of the accessions, to avoid false positives. Automatically sorting the sequences according to given taxonomic levels can help users in post-processing the sequences. The kingdom option (*-k*) sorts the sequences according to the kingdoms that are pre-recorded in the script. The *-T* (or *--taxids*) option produces a text file with all the accession numbers returned by NCBI for a user's query and the corresponding NCBI taxonomic IDs.
The results are as follows:

- METAZOA.fasta file containing 3 637 305 sequences

- PLANTAE.fasta file containing 25 987 sequences
- FUNGI.fasta file containing 2 089 sequences
- OTHERS.fasta file containing 30 371 sequences with lineage that does not correspond to any of the above three kingdoms (*e.g.* MZ152022.1 which is a heterokont).
- The no_taxid.fasta file contains 31 581 sequences for which no taxonomic information could be found. In some cases, the taxonomic identifiers available in the downloaded files are out of date, and not easily accessible from the taxonomy database.
- The TaxIDs.txt file contains all accession numbers and their TaxIDs in two, tabulation separated columns.

3.3. Third example

We used the following command to download mitochondrial (mitochondrion[Filter]) sequences of four Lepidopteran genera (Parnassius[ORGN] OR Pieris[ORGN] OR Melithaea[ORGN] OR Iphiclides[ORGN]) published in NCBI nucleotide database since 2010 ("2010"[Publication Date] : "3000"[Publication Date]), with a barcode present in the keyword field (barcode[Keyword]):

```
nsdpy –r "(Parnassius[ORGN] OR Pieris[ORGN] OR Melithaea[ORGN] OR
Iphiclides[ORGN]) AND COI[Title] AND (2010[Publication Date] :
3000[Publication Date]) AND barcode[Keyword] AND mitochondrion[Filter]" –i –t
```

The request (*-r*) combines a series of search terms, each of them limited to a particular field (e.g. Organism, Publication Date) of the GenBank record. Note the use of OR to link the different names of the genera, to obtain sequences from all of them, and the use of parentheses to correctly structure the query. The -i option replaces the original definition lines to give taxonomically informative format more adapted for phylogenetic studies: taxon_name-SeqID | TaxID | Taxonomic lineage with comma-separated values. The -t option produces a tsv file with the following fields: Taxon name, SequenceID, TaxID, Taxonomic lineage, Sequence length, Sequence.
Note, the exact same result can be obtained by the following command:

```
nsdpy –r "COI[Title] AND (2010[Publication Date] : 3000[Publication Date])
AND barcode[Keyword] AND mitochondrion[Filter]" –i –t –L genus_list.txt
```

The genus_list.txt is a text file with the four Lepidopteran genera of interest (one genus name in each line). We used just a short list, in this example, but it can contain thousands of taxa of any taxonomic level.

The command has downloaded 1447 sequences form 55 taxa (genus, species, or subspecies).

4. ***Impact***

High throughput sequencing (HTS) has gone through a spectacular development in the last 20 years. Sequencing throughput increased, while sequencing cost decreased exponentially[12]. This has

revolutionized molecular biology and now HTS is routinely used in very diverse fields of biology such as medical research[13–15], transcriptome analyses[16], genome structure[17], genomic variation and biodiversity studies [18–20]. Many of these fields require the use of custom databases, but metabarcoding is one of the fields where the need of custom databases is most pressing[21]. Metabarcoding has become a well-established method for biodiversity surveys[22]. The number of metabarcoding studies has dramatically increased in the past ten years. Google Scholar returned 124 studies with metabarcoding as a keyword published in 2011, while this number is about 5500 for 2021. One of the key aspects of metabarcoding is the taxonomic assignation of the sequences, which relies on a reference database as complete as possible. The NCBI nucleotide database contains nearly 500 million sequence records[5] and this number is steadily increasing. This provides an excellent resource for taxonomic assignment due to its large taxonomic coverage. However, due to its size, direct comparison of sequences to this database is impractical. Therefore, there is a need to establish custom databases of molecular markers and/or taxonomic groups and to maintain them up to date[21,23–25].

The download of large amounts of DNA sequences is inevitable when constructing custom databases. NSDPY is a practical tool to automate the download, and to sort and filter the sequences. Accessing taxonomic information is particularly important for creating custom databases for biodiversity studies and particularly for metabarcoding projects. Precise taxonomic information (TaxID, Lineage) is seldom included in standard downloading formats, or if present, further processing is necessary that requires some programming knowledge. Thus, the ability to download the taxonomic information and sort the sequences to different files according to a given taxonomic level can save a considerable amount of time and helps to avoid sorting errors. Another time-saving option is the possibility to download sequences from a long list of taxa with a single command.

The possibility of filtering the sequences by a gene name (provided as a regular expression) is a useful feature when downloading whole organelle sequences such as mitochondria or selecting different genes from long sequences. This feature is very useful, since it allows users to obtain a set of high-quality sequences where the gene is complete and correctly oriented. When searching for genes of organelles, these high-quality sequences cannot be easily accessed by classic keyword searches. Thus, NSDPY fills a gap and helps users to easily obtain complete sequences of a gene from organelle genomes.

With NSDPY, if a disconnection occurs while downloading, the script will wait for the connection to be re-established to resume the download thus avoids restarting the procedure and losing time.

This software package is meant to be used to download large amount of DNA sequences from NCBI databases to generate custom reference databases. The software can be included in bioinformatic pipelines. The automation of gathering the sequences with their taxonomic information, sorting, and filtering the sequences without having to learn how to code can drastically improve the workflow of users. For the users with coding experience, the script can be easily modified to be adapted to other specific needs.

5. Conclusions

In this paper, we described the implementation of NSDPY and its built-in features and provided examples of possible usages. As it is written under the MIT licence and can be easily customized, it is expected that NSDPY will be developed to adapt to more specific work. It meant to support the work of researchers who need to work with custom databases of DNA sequences.

Conflict of Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

IMBE has covered the Article Processing Charges. We thank two anonymous reviewers for their constructive comments to improve NSDPY and the manuscript and Judith Nève for proofreading the manuscript.

Author contributions

Raphael Hebert: Conceptualization, Methodology, Software, Validation, Writing - Original Draft

Emese Meglécz: Supervision, Conceptualization, Validation, Writing- Reviewing and Editing

References

[1] V. Arranz, W.S. Pearman, J.D. Aguirre, L. Liggins, Scientific Data 7 (2020) 209.
[2] J.-N. Macher, T.-H. Macher, F. Leese, Metabarcoding and Metagenomics 1 (2017) e22262.
[3] M. Gouy, S. Guindon, O. Gascuel, Molecular Biology and Evolution 27 (2010) 221–224.
[4] V. Elbrecht, F. Leese, Methods in Ecology and Evolution 8 (2017) 622–626.
[5] E.W. Sayers, E.E. Bolton, J.R. Brister, K. Canese, J. Chan, D.C. Comeau, R. Connor, K. Funk, C. Kelly, S. Kim, T. Madej, A. Marchler-Bauer, C. Lanczycki, S. Lathrop, Z. Lu, F. Thibaud-Nissen, T. Murphy, L. Phan, Y. Skripchenko, T. Tse, J. Wang, R. Williams, B.W. Trawick, K.D. Pruitt, S.T. Sherry, Nucleic Acids Research 50 (2022) D20–D26.
[6] E.W. Sayers, T. Barrett, D.A. Benson, E. Bolton, S.H. Bryant, K. Canese, V. Chetvernin, D.M. Church, M. Dicuccio, S. Federhen, M. Feolo, I.M. Fingerman, L.Y. Geer, W. Helmberg, Y. Kapustin, S. Krasnov, D. Landsman, D.J. Lipman, Z. Lu, T.L. Madden, T. Madej, D.R. Maglott, A. Marchler-Bauer, V. Miller, I. Karsch-Mizrachi, J. Ostell, A. Panchenko, L. Phan, K.D. Pruitt, G.D. Schuler, E. Sequeira, S.T. Sherry, M. Shumway, K. Sirotkin, D. Slotta, A. Souvorov, G. Starchenko, T.A. Tatusova, L. Wagner, Y. Wang, W.J. Wilbur, E. Yaschenko, J. Ye, Nucleic Acids Res. 40 (2012) D13-25.
[7] J. Kans, Entrez Direct: E-Utilities on the Unix Command Line, National Center for Biotechnology Information (US), 2021.
[8] B. Chapman, J. Chang, SIGBIO Newsl. 20 (2000) 15–19.
[9] D.J. Winter, Rentrez: An R Package for the NCBI EUtils API, PeerJ Inc., 2017.
[10] F. Pina-Martins, O.S. Paulo, SoftwareX 5 (2016) 80–83.
[11] M.D. Bethesda, Entrez Help, National Center for Biotechnology Information (US), 2016.
[12] J.A. Reuter, D. Spacek, M.P. Snyder, Mol Cell 58 (2015) 586–597.
[13] W.W. Soon, M. Hariharann, M.P. Snyder, Molecular Systems Biology 9 (2013) 640.
[14] G. Lightbody, V. Haberland, F. Browne, L. Taggart, H. Zheng, E. Parkes, J.K. Blayney, Briefings in Bioinformatics 20 (2019) 1795–1811.
[15] C. Manzoni, D.A. Kia, J. Vandrovcova, J. Hardy, N.W. Wood, P.A. Lewis, R. Ferrari, Brief Bioinform 19 (2018) 286–302.
[16] Z. Jiang, X. Zhou, R. Li, J.J. Michal, S. Zhang, M.V. Dodson, Z. Zhang, R.M. Harland, Cell Mol Life Sci 72 (2015) 3425–3439.
[17] A.C. Brant, W. Tian, V. Majerciak, W. Yang, Z.-M. Zheng, Cell & Bioscience 11 (2021) 136.
[18] K.R. Andrews, J.M. Good, M.R. Miller, G. Luikart, P.A. Hohenlohe, Nat Rev Genet 17 (2016) 81–92.
[19] K.M. Ruppert, R.J. Kline, M.S. Rahman, Global Ecology and Conservation 17 (2019) e00547.
[20] P. Arribas, C. Andújar, M.I. Bidartondo, K. Bohmann, É. Coissac, S. Creer, J.R. deWaard, V. Elbrecht, G.F. Ficetola, M. Goberna, S. Kennedy, H. Krehenwinkel, F. Leese, V. Novotny, F. Ronquist, D.W. Yu,

L. Zinger, T.J. Creedy, E. Meramveliotakis, V. Noguerales, I. Overcast, H. Morlon, A.P. Vogler, A. Papadopoulou, B.C. Emerson, Mol Ecol 30 (2021) 1120–1135.
[21] H. Weigand, A.J. Beermann, F. Čiampor, F.O. Costa, Z. Csabai, S. Duarte, M.F. Geiger, M. Grabowski, F. Rimet, B. Rulik, M. Strand, N. Szucsich, A.M. Weigand, E. Willassen, S.A. Wyler, A. Bouchez, A. Borja, Z. Čiamporová-Zaťovičová, S. Ferreira, K.-D.B. Dijkstra, U. Eisendle, J. Freyhof, P. Gadawski, W. Graf, A. Haegerbaeumer, B.B. van der Hoorn, B. Japoshvili, L. Keresztes, E. Keskin, F. Leese, J.N. Macher, T. Mamos, G. Paz, V. Pešić, D.M. Pfannkuchen, M.A. Pfannkuchen, B.W. Price, B. Rinkevich, M.A.L. Teixeira, G. Várbíró, T. Ekrem, Science of The Total Environment 678 (2019) 499–524.
[22] K. Deiner, H.M. Bik, E. Mächler, M. Seymour, A. Lacoursière-Roussel, F. Altermatt, S. Creer, I. Bista, D.M. Lodge, N. de Vere, M.E. Pfrender, L. Bernatchez, Molecular Ecology 26 (2017) 5872–5895.
[23] R.T. Richardson, J. Bengtsson-Palme, M.M. Gardiner, R.M. Johnson, PeerJ 6 (2018) e5126.
[24] J. Decelle, S. Romac, R.F. Stern, E.M. Bendif, A. Zingone, S. Audic, M.D. Guiry, L. Guillou, D. Tessier, F. Le Gall, P. Gourvil, A.L. Dos Santos, I. Probert, D. Vaulot, C. de Vargas, R. Christen, Molecular Ecology Resources 15 (2015) 1435–1445.
[25] R.H. Nilsson, K.-H. Larsson, A.F.S. Taylor, J. Bengtsson-Palme, T.S. Jeppesen, D. Schigel, P. Kennedy, K. Picard, F.O. Glöckner, L. Tedersoo, I. Saar, U. Kõljalg, K. Abarenkov, Nucleic Acids Research 47 (2019) D259–D264.