SeqDex

1. Introduction

SeqDex is an automated model for the deconvolution of dataset of whole genome sequencing of symbiont(s) together with the hosts. It uses taxonomic affiliations obtained through comparison to reference databases, coupled with k-mers frequencies and rRNA 16S gene identification to isolate genomic contigs of the organism(s) of interest. The workflow has three main phases: the first, where data are prepared to subsequent analysis; the second, where taxonomic affiliation is used as category to train a supervised machine learning model on k-mers frequencies to predict the taxonomy of contigs with no homologies in the databases; the third, where unsupervised automated clustering is performed and contigs in the cluster(s) containing the rRNA 16S gene(s) of interest are extracted and assigned to different sources.

2. Dependencies
2.1. Packages

The dependencies that can be installed via terminal are:
- Samtools
- Bedtools
- NCBI-BLAST+
- Barrnap
- Prodigal
- Diamond
- Seqtk

Of these, only Prodigal and Diamond are optional (see above).

Most of SeqDex has been developed in R, so it needed some R package to run:
- Taxonomizr
- Seqinr
- randomForest
- e1071
- Uwot
- Dbscan
- Parallel
- doParallel
- Foreach
- Optparse
- Ggplot2
- igraph

Of these, only Taxonomizr needs some manual curation. As described elsewhere (https://cran.r-project.org/web/packages/taxonomizr/vignettes/usage.html), Taxonomizr is used to convert NCBI accession number in taxonomic information. To do so, it uses an sql file that stores the information for the conversion. This file (`accessionTaxa.sql`) cannot be downloaded but must be compiled following the instruction listed into the vignette linked below. Take care that: (1) following the instruction for 'preparation' and use `prepareDatabase('accessionTaxa.sql')` command, it will produce a file containing only nucleotidic NCBI accession number; if you wish to use SeqDex by providing also protein taxonomic affiliation, then the 'Manual preparation database' instruction must be followed; (2) this step needs to be done once, but be aware that once you have the `accessionTaxa.sql` file, any new NCBI nt or nr version will contain new accession number not present in the older sql file, so the Taxonomixr preparing commands may need to be rerun or these new taxonomic informations will be lost; (3) SeqDex internally searches for a file named exactly '`accessionTaxa.sql`' so please use this default name and do not create multiple versions, or it will return an internal error.

2.2. Databases

SeqDex models combine supervised and unsupervised step. To do so, we used taxonomic affiliations to be able to develop automated supervised prediction and cluster selection steps. Therefore, SeqDex needs databases to which compare sequences: three in total, of which one is optional.

To run SeqDex the mandatory databases are a nucleotidic reference database to which compare contigs, and the RDP rRNA database to which compare 16S genes found by `Barrnap`.
As for the nucleotidic database, you may choose to:

1) use a custom made reference database. In this case make sure that the sequences that compose it are called by their NCBI name. To obtain it download the sequences of interest form NCBI ftp, group them in a single file using `cat` command and then run `makeblastdb` to obtain a database readable to blastn;
2) use the entire NCBI nt database, downloaded form NCBI ftp site (ftp://ftp.ncbi.nlm.nih.gov/blast/db/) in database format, as it is already in a readable blastn format.

For the 16S database, we decided to use the RDP database, as it is the most complete one. You can download it form https://rdp.cme.msu.edu/misc/resources.jsp, in the unaligned format to avoid gaps. You may download the fasta file and then use `makeblastdb` to obtain blastn readable database. Do not trash the fasta file, it is used by SeqDex too to retrieve taxonomic informations, as RDP does not used NCBI accession numbers.

When target endosymbionts are particularly distant from relative genome published online, or when the sample under analysis is particularly complex (lots of organisms, either symbionts or contaminating) nucleotidic taxonomic affiliation might not be enough. In these cases, coupling nucleotidic with proteic taxonomic information may be useful, as proteins are more stable, conserved and less susceptible to little variations, and so could be possible to retrieve taxonomic informations of even these distant-near relatives.
To do so, you will need to install the two optional dependencies (`prodigal` and `diamond`) as well as download a reference database. As for the nucleotides, you may choose to:

1) use a custom made reference database. As before, make sure that the sequences are called by their NCBI name, merge them in a file using `cat` command and create a Diamond readable database by using `diamond makedb`.
2) use the NCBI nr database (ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/). Make sure to download from NCBI ftp site not the blast database files, but the fasta file, to then create a Diamond readable database file using `diamond makedb`.

We chose to implement this part by using Diamond instead of blastp to reduce computational time needed, as Diamond is faster (original fasta file not required).

You can choose to skip this step at the beginning to decide to fulfill this later, only whether you need it. Only take care that, if at the beginning you create the `accessionTaxa.sql` file only for nucleotidic taxonomic informations, then you must rerun the Taxonomizr preparation step to include also proteins (see above).

3. SeqDex workflow

SeqDex can be divided into 3 main phases, discussed in detail below. These comprehend (i) an initial manipulation of the data to obtain the information needed in the subsequent phases; (ii) an initial discrimination through machine learning, and (iii) a final clustering and prediction step.

3.1. Input files

The inputs needed for SeqDex.sh are: (1) the basename of the mapping file (sam format); (2) the basename of the contig/scaffold assembly file (fasta format). So, prior to running SeqDex, you have to assembly the reads and then map the reads back to the contigs/scaffolds. The quality of the assembly can influence the capability of the model to discriminate organisms, so please take care in this step. We suggest: (1) evaluate sequencing quality by using FastQC; (2) remove adapters and low quality base with Trimmomatic or similar tools; (3) if sequencing has been performed in paired end mode, then merge the overlapping mates by using FLASh; (4) assemble

these reads (with SPADes or an equivalent assembler) by enabling reads correction and testing several k-mer lengths; (5) evaluate the best assembly using Quast (including both contigs and scaffolds in the analysis); (6) map the corrected reads to the best contigs/scaffolds file. You can choose to use whenever tool you prefer, as long as the assembly file is in fasta format and the mapping file in sam format.
We considered paired end sequencing as most whole genome sequencing are performed by using this modality. However, SeqDex can be used also on single end sequencing data without modifying the code.

3.2. Phase 1 - data preparation

3.2.1 Coverage Calculation

FLASh merges overlapping paired reads, when found. Once performed, you will obtain as output the paired non-overlapping reads and the extended reads resulting from merging the overlapping mates. As Bedtools coverage, the tool used to get coverage values for the contigs, do not differentiate among single reads from mates of the same pair, we calculate the coverage separately for single and paired reads and then sum them appropriately. The outputs are saved in the Coverage folder.

3.2.2 Taxonomic affiliations

As described before, SeqDex needs high quality taxonomic affiliations to train the machine learning model. `SeqDex.sh` by default uses only nucleotide-derived taxonomic affiliations. Insert the name of the nucleotide database (`NTI`) and the path to this file (`NT`) by editing the `SeqDex.sh` file and allowing the next call to blastn by using the contigs as queries. Taxonomy.R then reads the blast output to : (1) convert NCBI codes into taxonomy information; (2) filter hits by alignment length (`–aliLength`, default= 200 bp) and percentage of identity (`–Xid`, default= 70%); (3) for each taxonomic level (`–taxaLevels`, default= 1) the taxonomy Density (`–taxonDensity`, default= 0.75) is calculated and filtered at a defined threshold; (4) save the output in Taxonomy folder for subsequent analysis.

If you want to use protein-derived affiliations too, you must change in `SeqDex.sh TAX` to `NTNR`, provide the name of the database (`NRI`) and its path (`NTR`). Nucleotide workflow proceeds as explained above. The protein workflow implements Prodigal to obtain gene predictions and corresponding protein sequences, and then compare them to the `NR` database by using Diamond. Outputs are analyzed by Taxonomy.R to: 1) convert NCBI code into taxonomic informations; (2) filter hits by alignment length (`–aliLength; –aliLengthNR`, default= 70 aa) and percentage of identity (`–Xid; –XidNr`, default 90 %); (3) for each taxonomic level (`–taxaLevel`) the taxonomy Density (`–taxonDensity`) is calculated and filtered at a defined threshold; (4) the shared affiliations combined with the unique affiliations, relative to both databases, are merged and the output is saved in Taxonomy folder.

You may wish to repeat the machine learning classification step on more than one taxonomy level, i.e.: on Superkingdom level to select only bacterial contigs, to then predict bacterial class and select only the class of interest (e.g. *Alphaproteobacteria*, see below). To do this, you need to tell to the model to prepare the taxonomic informations for each level considered (Superkingdom and Class) by providing a comma separated list of numbers corresponding to the levels desired, where 1 means Superkingdom and 7 species (run `Rscipt Taxonomy.R –help` to see a full list of options; `–taxaLevels`). We suggest not to go too deepen in taxonomy levels: most of the times, when the sample under analysis shows low complexity (only one symbionts with high coverage), the classification only at Superkingdom level is enough to then clustering the contigs relative to the target organism.

As we build SeqDex to be highly flexible, you can choose to prepare all the file at first entrance, run SeqDex as desired, evaluate the final output and then consider rerunning only certain part by manually copying commands from `SeqDex.sh` to the terminal.

3.2.3 Mate network construction

Nucleotidic and/or proteic derived taxonomic affiliation can return too partial or scarce information to predict and classify contigs with low error. This is imputable to peculiarities of the organism under analysis: stable obligate endosymbionts show skewed reduced genome, with gene losses and high AT content. In these conditions, even if a related genome and its proteins are present in public databases (or, at least, in the databases used for inferring taxonomic affiliations), they could be too divergent to obtain good and valuable matches, especially if the near relative is a endosymbiont itself.
This will reduce the number of labelled contigs, affecting negatively the parametrization and the performance of the machine learning models. Including protein-derived taxonomical affiliations could improve the number of labels obtained. However, there is the risk to inflate computational time and resources for little improvement. Also, due to possible erroneous taxonomic affiliation given to deposited NCBI sequences, there is the possibility to obtain different affiliations for the same contigs coming form nt and nr databases. In this case, there will be no shared labels between the two databases and these will be discarded, reducing the number of labels retrieved.
To help improve the taxonomy coverage of the sample, we decided to use the information of paired end reads mapping in the assembly. In detail, SeqDex builds a graph based on mates mapping on different contigs. Here, if the edges (corresponding to the pairing information) that connect vertices (corresponding to contigs) are confirmed by `—Edges`, pairs (default= 10 pairs), we assume that the connected vertices (contigs) were parts of the same genome that have been put into different contigs due to lack of overlap. The complexity of such a graph can be high and the user can control it by setting the maximum degree of contigs (`—VerticesDegree`, default= 5). Highly connected contigs are mostly repeated regions, and their presence is basically at the basis of our difficulties in assembling highly contiguous genomes from short reads only. However, in this case the presence of intricate connected components (CC) provides a way to increase the taxonomy coverage of the dataset. Therefore, we set rules to exploit this graph to transfer taxonomy labels to contigs belonging to connected components where at least some of the contigs provide congruent taxonomical information derived from the homology search. The user can set `—componentSize`, which is the minimum size for a CC to be considered for trying label transfer (default=2).

Use with care!

1. if the CC has size 2 and only one vertex is labeled, then the label is transferred; if the two vertices have incongruent labels the labels are not considered for model training.
2. If the CC has size more than 2 vertices if there is only one type of label, then this will be extended to the unlabelled vertices; if there are two label types and the underrepresented have a frequency less than `—mixedComponents`, then these labels will be considered as ‘erroneous’ and corrected; if there are more than two labels, these will be discarded.

These parameters and rules will be applied to the taxonomy, but also to predicted affiliations to correct errors or give more support to predictions, and to final clustering to retrieve also the contigs shorter than a provided value (and so not included in the analysis; see below).

3.2.4 rDNA 16S

The 16S genes present into the sample need to be located and taxonomically characterized because the final clustering step searches for the cluster with the 16S gene with the Class target taxonomy category with higher coverage among all found. Fasta file is analyzed by using barrnap to locate putative 16S rRNA genes, and then compare to an RDP database by using Blastn. To do this you must provide in `SeqDex.sh` the name of the fasta RDP file (`RDPF`), the name of the database file (`RDPI`) and the path to the folder where these are located (`RDP`).

The output of blastn is then analyzed by `rRNA16S.R`, where only one among all the taxonomic affiliations obtained for each 16S contigs is selected. Only affiliations of contigs longer than `—minContigLen` (default= 1000) and with alignment length longer than 500 bp are saved in the output, in the Taxonomy folder. 16S genes can be fragmented in the assemblies, but we choose to consider only putative genes e of length at least 500 bp. However, the user can change it by directly open and change this value in `rRNA16S.R` script.

3.2.5 K-mers frequencies

K-mers frequencies are used both to machine learning classification and clustering steps.
In `GCKmersCov.R` script, fasta file is read, and the k-mers counts are calculated by taking into account reverse, complement and palindromes, as to be able to reduce redundant information, avoiding differential contribution to global information of palindromes and non-palindromes k-mers, and avoid inflate dimensionality, which will lead to huge improve of computational time needed. Frequencies are calculate by normalizing the counts by the length of their sequence, and then multiplied by 1000 to avoid calculation problems of the algorithms used by SeqDex.

SeqDex by default calculate 3-mers frequencies (`—Kmers`). We choose this length because the main purpose of this model is to discriminate endosymbionts form their hosts, which are mostly eukaryotic, and so trimers can implicitly consider differential codon usages and coding density, which should differ from Bacteria to Eukaryotes. Although you can choose whichever k-mer length you prefer.
In `GCKmersCov.R` also total coverage is calculated by summing single coverage value to half of the paired. For each sequence, these values are divided by the length to calculate the mean per nucleotide coverage, to be able to compare coverage of contigs of different length.
The output of `GCKmersCov.R` is saved in the Coverage folder and contain Kmers frequencies, GC content, nucleotide coverage and contigs length.

3.3. Phase 2 - machine learning classification

Once completed, the output files of the phase 1 became the input file for the phase 2, which couple Kmers frequencies with taxonomic affiliations to teach a machine learning model to then predict the taxonomically unknown contigs. So both `SVM.R` and `RF.R` scripts take as input the table with Kmers frequencies (`—gcCovKmersTable`), 16S rRNA gene table (`—rRNA16S`), the taxonomy table (`—taxonomy`) and the mate CC network (`—network`). Together with the network, also argument `—Edges`, `—VerticesDegree`, `—componentSize` and `—mixedComponents` are needed, as described before.
Both machine learning algorithms are implemented following the same workflow, so in the subsequent paragraph we will discuss it in detail, together with common parameters, to then focus on individual options.

Only contigs longer than `minContigLength` will be considered, of which only the sequences which have a label in the taxonomy table, will be used to build the machine learning model. The dataset is so divided into labeled and unlabeled.
Labeled dataset is randomly divided into training set, which is composed by two thirds of the contigs and is used to teach the model; and the test set, i.e. the remaining contigs which is used to test the model and verify the error committed in the prediction: comparing the predicted and the inferred affiliations in this subset gives percentages of error committed by the models. This procedure is repeated 100 times, so that each time the training and test set are randomly sampled. The percentage of erroneous contigs/length is reported as a cumulative measure of all 100 permuted models. Also, the scripts calculate sensitivity, precision, accuracy and F1 score of both number of contigs and cumulative length on the comparison between empirically inferred taxonomy and the predicted (over the 100 models).

After this procedure, the unlabeled contigs are predicted by using all the 100 models and then the percentage of inclusion within a taxonomical category is calculated. As output, only the most represented taxonomical category is reported. However, if a contig shows more than an affiliation

and the 100 predictions are distributed among them such that are equally divided and the so percentages of belonging to each are equal (i.e.: two categories, 50% of presence each), the model keep them all.

The taxonomic affiliation, predicted or inferred, is integrated, and eventually corrected, by using CC, following the same rules described above. Here, uncertain predicted affiliations may be corrected or verified. If there is more taxonomical affiliations than allowed by CC parameters, then the contigs involved are considered '`misclassified`'. These contigs, as there is uncertainty about their origin, are included into the next iteration or in the output. Also, these scripts calculate a homogeneity (H) index, defined as the number of homogeneous CC divided by the number of heterogeneous CC in terms of taxonomic predictions. It goes from 0 (all heterogeneous) to 1 (all homogeneous) and give a magnitude of how much is supported the prediction. It is calculated by dividing the number of homogeneous by the total number of components.

The user can choose if print on screen these stats or only on the output stats file (`–verbose`, default `TRUE`).
The machine learning algorithms have been developed to be highly parallelizable, so both SVM and RF support `–threads` argument (default 8).

We developed our SVM and RF scripts to allow easy customization of critical parameters of both algorithms.
Support vector is sensible to `–cost`, `–gamma` and `–cross` values. Our implementation automatically select best cost and gamma from a provided interval of values. The user can choose to use default parameters (`–cost`: 1e-1,1e3; `–gamma`: 1e-5,1e-1; `–cross`: 5) or provide its own. Also e1071 automatically select the best kernel type for the data provided. To further informations see https://cran.r-project.org/web/packages/e1071/index.html.
The randomForest package used by SeqDex `RF.R` script chooses by default the number of variables randomly sampled as candidate at each split on the total number of variables given to the algorithm. Also, our implementation allows the user to choose the number of tree to be grown in the forest and if the sampling have to be done with or without replacement (default: `–ntree` 550, `–replace` TRUE).

You can decide whether to pass the output of the machine learning through clustering (Phase 3) by setting `CLUSTER=yes` (default) or not in the sh file (the scripts just recognize if this variable is equal or not to yes). In the latter case, the model gives as final output the fasta file of the contigs listed in the machine learning output.

3.4. Phase 3 - clustering

The clustering takes as input the taxonomy (`–taxonomy`), k-mers frequencies (`–gcCovKmersTable`), 16S genes (`–rRNA16S`), mate CC network (`–network`), as well as the phase 2 output files (`–modelOutput`). If the machine learning prediction is done with iteration, only one of the taxonomy files need to be provided, as also the last prediction output.
The `Cluster.R` script reduces the Kmers frequencies dimension to a user defined number of components (`–ncomponents`, default= 2). As the uwot packages perform a parallelized form of UMAP, then the `THREADS` value set in the sh file will be used here too (`–threads`, default= 8) to reduce execution time.
The user can choose to do the subsequent clustering only on these new components or also on GC content and/or coverage value (`input`, default= Kmers). Endosymbionts may have different coverage and GC content in respect to hosts or other associated organisms; when this is suspected, these two variables may bring a significant improvement in the performances.

After obtaining these new dimensions, DBscan is used to cluster the data. The script automatically calculates a minPts values, as equal to the logarithm of the number of contigs used in clustering, and on this calculate the best eps value. Usually, Eps is chosen by plotting the k Nearest Neighbors distances (kNN) and selecting the distance values where the curve changes slope. We avoid this completely user-dependent part by rounding the kNN distances to the first

digits, calculate the difference between consecutive and selecting the kNN value which has the greatest difference from its subsequent value.

After DBscan, each contig is assigned to a cluster and whether this assignation is supported or not is confirmed by using mate CC network, as previously done for taxonomy and machine learning prediction step. However, here also contig shorter than the provided length (`minContigLen`, default= 1000) will be included in the output by extending the cluster belonging through CC.
Finally, the cluster composed by the putative contigs of interest is obtained by searching the contigs with the 16S gene with Class taxonomical affiliation of interest (`TRG` in sh file, `targetName` in R script) and selecting the one with higher coverage value among them.

Finally, the fasta file of the contigs of interest is returned as output.

3.5. Output files

SeqDex produces various folders with output files, most of which are used as checkpoint to be able to then rerun the analysis starting form them, if needed.

Coverage: here can be found the two coverage files (paired and single: `sambasenamefile_PAIRED_end.bed` and `sambasenamefile_SINGLE_end.bed`), the file with kmers frequencies, GC content and total coverage informations (`gckCovTable.txt`), and the mate network file (`contigNet.txt`).

Taxonomy: in this folder there are blastn output (`ContigsvsNt.txt`), prodigal predicted proteins (`prodigalContigs.faa` and gff format `ProdContigs`) used then by diamond (`ContigsvsNr.txt`), if performed, and their elaboration made by Taxonomy.R, which produces one file per iteration performed (`1taxonomyIteration.txt`, the number indicate the iteration); the 16S genes predicted by barrnap (`barrnap16s_contigs.gff`, `16sContigs.fasta`), its alignment file to RDP (`16sContigsvsRDP.txt`), the RDP based taxonomy file (`RDP16s_taxa_mod.txt`) and the final elaboration performed by rRNA16S.R (`rRNA16sTaxonomy2.txt`).

SVMoutput and RFoutput: for each iteration performed, the SVM/RF script will produce a stats file (`1output_statsSVM.txt`, `1output_statsRF.txt`; the number indicate the iteration), a file with the Kmers frequencies file with only the target contigs (`1outputSVM.txt`, `1outputRF.txt`; chosen on the provided taxonomy level and selecting the target taxonomy label, both inferred and provided) and the environment (`1SVMModel.RData`, `1RFModel.RData`; so that an expert user may be able to use the so saved SVM/RF model).
If the clustering step is not performed, in these folders can be found also the fasta files of the contigs selected with the machine learning algorithms (`1outputSVM_contigs.fasta`, `1outputRF_contigs.fasta`).
In the stats file are reported performance statistics of the 100 models constructed in each machine learning R script. In detail, for each model, the 100 prediction performed on the test set and their relative empirically inferred taxonomy are used to calculate cumulative percentage of error, sensitivity, precision, accuracy and F1 score by considering both number of contigs and total length. At the end of these files is reported the H-index.
The output file with the target contigs contains:
- name of the contigs;
- GC content;
- k-mers frequencies;
- coverage by length;
- TaxonDensity value: if it is 'NoBlastHit' means that the contigs had not empirically inferred taxonomic affiliations, which have been predicted in this phase; either if it is '-1' means that this labels is derived form the mate CC network based extension;
- taxonomy labels converted into numbers (the order reported in stats file);

- percentage of belongs of a contig to its label, calculated over the predictions done with 100 model constructed. '-1' Could means either that the contigs label is empirically inferred or mate CC network extension derived.

ClusteringOutput: if SeqDex is run enabling both machine learning algorithms, then it will produce two clustering folders (`ClusteringOutputSVM` and `ClusteringOutputRF`) to avoid possible overwriting, or only `ClusteringOutput` if only one machine learning algorithm is performed. In the folder(s) there are different files: the list of the contigs in the target cluster, extended by using mate network to retrieve also contigs shorter than the minimum length considered (`OutputClustering.txt`); the fasta file of the target contigs (`OutputClusteringSVM.fasta`, `OutputClusteringRF.fasta`); the clustering stats, reassuming the number of clusters obtained and amount of contigs in each, and the homogeneity index (`output_statsClustering.txt`); the final clustering table validated with the mate network (`extendedClusteringCC.txt`); a plot file with the scatterplot of the contigs plotted by using first two umap dimensions, and the scatterplot of the contigs colored by clusters (the black points are the outliers, which are excluded by clustering; `Rplots.pdf`).
The extendedClusteringCC.txt contains:
- the names of the contigs;
- the new umap calculated dimensions;
- the percent of belonging of a contigs to its lables (calculated in SVM/RF prediction phase);
- the number of the component to which each contig belongs: if it is '-1' means that the contigs does not belong to a CC;
- The number of the cluster/group to which each contig belongs.

4. Running SeqDex

You can run SeqDex by using default parameters or by providing your own.
In both cases, there are variables that need to be provided. To do it, open the SeqDex.sh file with you preferred text editor and provide the `THREADS` (number of threads to be used), `NT` (path to nucleotidic dabases file), `NTI` (name of the nucleotidic index, basename if it is NCBI nt), `NR` (path to proteic database file), `NRI` (proteic database index), `RDP` (path to RDP file), `RDPF` (name of RDP fasta file), `RDPI` (name of RDP index), `SCRIPT` (path to the folder with SeqDex scripts), `TAX` (`NT` or `NTNR` to consider only nt or both nt and nr taxonomy), `MLALG` (`SVM`, `RF`, `BOTH`: which type of machine learning algorithm to be used), `TRG` (target category at Class level), `CLUSTER` (whether to perform final clustering or not), as described previously.

4.1 Default

To run SeqDex by using default parameters, you need at least to provide the above mentioned variables. Then change permissions of the file, if needed, and run on the terminal
`./SeqDex.sh basename_mapping_file basename_contigs_fastafile`
Like this, `SeqDex.sh` produce a taxonomy file only at Superkingdom, predict affiliation with both SVM and RF selecting only Bacteria contigs (empirically inferred or predicted) together with the misclassified, to then perform the final clustering step and retrieving the cluster with *Alphaproteobacteria* 16S gene with higher coverage (if you do not change `TRG`). The list of the default parameters are reassumed in tables 1-6.

4.2 Taxonomy parameters

To change the target, the taxonomy level and the category on which performing the prediction and selection of the contigs:
(1) change `TRG` argument in `SeqDex.sh`, to your taxonomy category at Class level;
(2) change the `TaxaName` argument in `Taxonomy.R` and `SVM.R`/`RF.R` to obtain taxonomic affiliation and prediction on a different level. If `TAX=NT` you have to change this parameter in line 162, otherwise in line 164 of `SeqDex.sh`. In addition, If `MLALG=SVM` you have to change

this parameter in line 208, or in line 264 if `MLALG=RF`, wheter if `MLALG=BOTH` the interested lines are 314 and 365 of `SeqDex.sh` file;

(3) provide a different label to `targetName` in the same lines of (2) to obtain taxonomic affiliation on a different level than the default.

4.3 Iteration parameters

To perform the iteration:

(1) change default parameter in `SeqDex.sh` for `Taxonomy.R` and provide a comma separated list, without spaces, of levels on which retrieve taxonomic information (`taxaLevels`, i.e.: `1,3` for Superkingdom and Class) to obtain one taxonomy file for each level of interest (see above). These files will be named by number of repetitions pasted with the basename '`taxonomyIteration.txt`' (i.e.: `1taxonomyIteration.txt` for the first iteration for `taxaLevels` 1; `2taxonomyIteration.txt` for second iteration for `taxaLevels` 3). If `TAX=NT` you have to change these parameters in line 162, otherwise in line 164 of `SeqDex.sh`;

(2) provide to `SVM.R` and/or `RF.R` scripts a comma separated list, without spaces, of taxonomic levels (`–TaxaName`, i.e.: `superkingdom,class`), taxonomic category (`–targetName`, i.e.: `Bacteria,Alphaproteobacteria`), as also taxonomy files with their paths (`–taxonomy`, i.e: `../Taxonomy/1taxonomyIteration.txt,../Taxonomy/2taxonomyIteration.txt`). Here, the scripts will (i) use the labels of the first iteration (`1taxonomyIteration.txt`) to predict the unlabelled contigs; (ii) integrate the affiliations with the mate CC network to validate/correct labels; (iii) select contigs with the target label (`Bacteria`) and then (iv) use only these on the subsequent iteration. Step (i)-(iv) will be repeated for all taxonomic levels and target provided. If `MLALG=SVM` you have to change these parameters in line 208, If else `MLALG=RF` these changes have to be done in line 264, else if `MLALG=BOTH` the interested lines are 314 and 365 of `SeqDex.sh` file;

(3) If you disabled the final clustering phase (`CLUSTER=not`), the `seqtk subseq` command at the end of each machine learning algorithm in the `SeqDex.sh` needs to be changed in order to provide the last SVM/RF output to obtain the final fasta file (`MLALG=SVM` line 239; `MLALG=RF` line 292; `MLALG=BOTH` line 342 for SVM and 395 for RF). Else, if the final clustering step (`CLUSTER=yes`) is enabled, then you have to change the `modelOutput` argument to perform it on the last output file of SVM/RF (`MLALG=SVM` line 232; `MLALG=RF` line 286; `MLALG=BOTH` line 338 for SVM and 382 for RF).

Adding iterations may be meaningless if the percentages of error returned by the model are high. Most of the times, in our experience, you may need to adjust Taxonomy.R values to use only highly sure taxonomic affiliations and predict only at Superkingdom level. However, we suggest not to go too deepen in taxonomic levels: you will not be able to obtain target contigs performing prediction an all taxonomy levels from Superkingdom to Species, because at each iteration SeqDex uses only contigs with taxonomic affiliations the empirically derived, also only the ones selected in the previous iteration. Like this, the number of contigs on which tough the model will decrease each iteration, and the error committed in each iteration will influence the subsequent, making nearly impossible to go from Superkingdom to *Species* with affordable predictions. In our experience, we used at maximum only two iteration (Superkingdom and Class)

4.4 Multiple targets dataset

To deconvolve dataset with more than a target symbionts, the user can decide to

(1) run a SeqDex file for each symbiont: run SeqDex for the first time on one of the target organisms, and then take advantage of coverage, taxonomy, rRNA16S, Kmers frequencies files already prepared to speed up subsequent run. `SeqDex.sh` will detect them and skip the command lines needed to produce them, which are highly time consuming. Take care to rename the file of the prediction/clustering of the first organisms, or move them, to avoid overwriting;

(2) modify the SeqDex.sh file to make SeqDex able to run all the targets deconvolving in sequences: (i) add a new `TRG` argument (which will substitute the first provided) at the end of

the first organisms deconvolving; (ii) add command `mkdir name_new_folder; cd name_new_folder`; (iii) copy and paste the prediction and clustering, if performed, part (lines 183 to 392 of the file); (iv) add command `cd ..`; (v) repeat point (i)-(iv) for each subsequent target. In this case, take care to change the path of each file that have to be provided to each new prediction-clustering R scripts.

Deconvolving multiple target dataset can be tricky. Just take care of: percentage of error committed by the machine learning models, as low error may involve the possibility to go deeper in the taxonomy levels considered; use the rRNA16sTaxonomy2.txt to find on which contigs the targets 16S genes are, and then check if they are correctly deconvolved in different contigs in the final clustering step (if performed). In the latter case, if in the cluster of the searched target there is the 16S contigs of another target, then this means that the two have not been correctly deconvolved and so you may wish to add an iteration to go deepen in the taxonomy levels to be able to correctly separate them.

4.5 Rerunning

Similarly, once completed the analysis, you may want to rerun the entire model or only part of it to see if the performance or the outputs will change significantly by changing some parameters.
The user can then choose to:

(1) re-run the entire `SeqDex.sh` by changing parameter of interest and taking care to move or rename old file or either they will be overwritten;

(2) run only scripts of interest, as each of our R scripts can be run independently by typing `Rscrip namescript.R` in the terminal. Similarly, to see all the option available for the script, type `Rscript namescrpit.R —help` or `Rscript namescrpit.R -h` . As before, old file must be renamed or moved, or they will be overwritten.

When machine learning predictions of SeqDex return high percentage of errors, we suggest to try to change `aliLength` and `Xid` of `Taxonomy.R` (even the proteic parameters, if used). This because our model needs a highly sure taxonomic information to correctly deconvolve dataset. Samples with low complexity may not be highly influenced, so using default values will lead to good percentage of error; Samples with high complexity (more than a target, contaminating sequences of other bacteria, …) might need more affordable taxonomic affiliations, so using stricter threshold for these values may be the right choice. To test the influence of these on your dataset, move or rename files to avoid overwriting (point (1)).

5. Default values of R scripts

Table 1 - Taxonomy.R default parameters with indication of the `SeqDex.sh` lines in which are used.

	Line 162 (NT)	Line 164 (NTNR)	Meaning
—blast	ContigsvsNt.txt	ContigsvsNt.txt	Contigs vs nucleotidic database file. The path is needed if the R script is run in a different folder than this file (default: same folder)
—taxaLevels	1	1	Taxonomy level to consider. 1 means Superkingdom
—aliLength	200	200	Minimum alignment length to nucleotidic hit
—Xid	70	70	Minimum percentage of identity between contigs and nucleotidic hit
—TaxonDensity	0.75	0.75	Minimum considerable percentage (in 0-1 scale) of belonging to a taxonomic affiliation of a contigs

	Line 162 (NT)	Line 164 (NTNR)	Meaning
—diamond	-	ContigsvsNr_mod.txt	Contigs vs proteic database file. The path is needed if the R script is run in a different folder than this file (default: same folder)
—aliLengthNr	-	70	Minimum alignment length to proteic hit
—XidNr		80	Minimum percentage of identity between contigs and proteic hit
—network	../Coverage/contigNet.txt	../Coverage/contigNet.txt	Mate CC network file with path
—Edges	10	10	Minimum value for an edge to be considered
—VerticesDegree	5	5	Maximum vertices degree to be considered
—componentSize	2	2	Minimum size of the component to be considered

Table2 - rRNA16S.R default parameters with indication of the `SeqDex.sh` lines in which are used.

	Line 165	Meaning
—blastRDP	16sContigsvsRDP.txt	Output file of 16S genes vs RDP. The path is needed if the R script is run in a different folder than this file (default: same folder)
—taxaRDP	RDP16s_taxa_mod.txt	File with RDP taxonomy
—minContigLen	1000	Minimum contigs legth to be considered by the model

Table 3 - GCKmersCov.R default parameters with indication of the `SeqDex.sh` lines in which are used.

	Line 179	Meaning
—contigs	../"${2}".fasta	Contigs fasta file with its path. SeqDex uses the second argument to automatically find it
—Kmers	3	K-mers length to be used
—covSingle	onlymapping_sorted_SINGLE.bed	Extended reads coverage file. If the R script is run in a different folder than this file, the path is aslo needed
—covPaired	onlymapping_sorted_PAIRED.bed	Paired end reads coverage file. If the R script is run in a different folder than this file, the path is aslo needed
—threads	"$THREADS"	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7

Table 4 - SVM.R default parameters with indication of the `SeqDex.sh` lines in which are used.

	Line 208 (SVM) or 314 (BOTH)	Meaning
—gcCovKmersTable	../Coverage/gckCovTable.txt	Output of GCKmersCov.R (with path)

	Line 208 (SVM) or 314 (BOTH)	Meaning
—rRNA16S	../Taxonomy/ rRNA16sTaxonomy2.txt	Output of rRNA16S.R (with path)
—taxonomy	../Taxonomy/ 1taxonomyIteration.txt	Output of Taxonomy.R (with path)
—network	../Coverage/contigNet.txt	Mate CC network file with path
—cross	5	Maximum number of allowed erroneous contigs in SVM model construction
—cost	1e-1,1e3	Range of cost to be tested to tune SVM model
—gamma	1e-5,1e-1	Range of gamma values to be tested to tune SVM model
—scale	F	Boolean value meaning whether to scale the data or not (usually not needed)
—minContigLen	1000	Minimum contigs legth to be considered by the model
—targetName	Bacteria	Name of the taxonomical label to be selected after prediction
—TaxaName	Superkingdom	Taxonomical level on which perform the prediction
—threads	“$THREADS”	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7
—Edges	10	Minimum value for a edge to be considered
—VerticesDegree	5	Maximum vertices degree to be considered
—componentSize	2	Minimum size of the component to be considered
—mixedComponents	0.2	Maximum proportion of alternative label in the component to consider it as erroneous and correct
—verbose	T	Boolean value meaning if to print output stats of the model in screen or not (won’t disable stats file writing)

Table 5 - RF.R default parameters with indication of the `SeqDex.sh` lines in which are used.

	Line 264 (RF) or 365 (BOTH)	Meaning
—gcCovKmersTable	../Coverage/ gckCovTable.txt	Output of GCKmersCov.R (with path)
—rRNA16S	../Taxonomy/ rRNA16sTaxonomy2.txt	Output of rRNA16S.R (with path)
—taxonomy	../Taxonomy/ 1taxonomyIteration.txt	Output of Taxonomy.R (with path)
—network	../Coverage/contigNet.txt	Mate CC network file with path
—minContigLen	1000	Minimum contigs legth to be considered by the model

	Line 264 (RF) or 365 (BOTH)	Meaning
—targetName	Bacteria	Name of the taxonomical label to be selected after prediction
—TaxaName	Superkingdom	Taxonomical level on which perform the prediction
—replace	T	Boolean value meaning whether the RF sampling have to be done with or without replacement
—ntree	500	Number of tree to be constructed by RF
—threads	“$THREADS”	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7
—Edges	10	Minimum value for a edge to be considered
—VerticesDegree	5	Maximum vertices degree to be considered
—componentSize	2	Minimum size of the component to be considered
—mixedComponents	0.2	Maximum proportion of alternative label in the component to consider it as erroneous and correct
—verbose	T	Boolean value meaning if to print output stats of the model in screen or not (won’t disable stats file writing)

Table 6 - Clustering.R default parameters with indication of the `SeqDex.sh` lines in which are used.

	Line 232 (SVM), 286 (RF), 336 and 388 (BOTH)	Meaning
—modelOutput	../SVMoutput/1outputSVM.txt (SVM, BOTH) ../RFoutput/1outputRF.txt (RF, BOTH)	Output file of SVM.R or RF.R
—gcCovKmersTable	../Coverage/gckCovTable.txt	Output of GCKmersCov.R (with path)
—rRNA16S	../Taxonomy/rRNA16sTaxonomy2.txt	Output of rRNA16S.R (with path)
—taxonomy	../Taxonomy/1taxonomyIteration.txt	Output of Taxonomy.R (with path)
—network	../Coverage/contigNet.txt	Mate CC network file with path
—minContigLen	1000	Minimum contigs legth to be considered by the model
—targetName	“$TRG”	Name of the target Class searched. SeqDex.sh uses TRG variable assigned in line 32
—threads	“$THREADS”	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7
—Edges	10	Minimum value for a edge to be considered

	Line 232 (SVM), 286 (RF), 336 and 388 (BOTH)	Meaning
—VerticesDegree	5	Maximum vertices degree to be considered
—componentSize	2	Minimum size of the component to be considered
—mixedComponents	0.2	Maximum proportion of alternative label in the component to consider it as erroneous and correct

6. Example files

In SeqDex there is the `Example` folder, which contain files to run the analysis on simulated dataset (see Chiodi et al, in preparation). By running SeqDex on this example dataset you will be able to check if R dependencies have been installed successfully. To do so, enter in this folder in the terminal, change permissions to `SeqDex_example.sh` file (if needed), and then run `./SeqDex_example.sh remap contigs`

This command will automatically detect the files in `Taxonomy` and `Coverage` folders to pass them to SVM.R and RF.R, without final clustering step. We decided to provide these folders to allow an easy and speed check of the mandatory dependencies without the need to have also reference databases for nucleotidic and 16S genes taxonomic affiliations. Also, as we compared the contigs against a database composed by only the two genomes used to create this simulated dataset, the scripts provided an affordable taxonomy affiliation for all the contigs, making impossible to complete the predictive step. We then selected the `blastn` hit for 2/3 of the contigs (randomly sampled) to then perform the prediction on the other 1/3.

In addition, in `Example` folder we added the file used to perform SeqDex on the *P. penetran*s dataset (seqDex_Ppenetrans.sh), which contain at least two endosymbionts. We added it as an example of how the sh file can be modified to perform our model on more than a symbiont dataset.