SeqDex

1. Introduction

SeqDex is an automated approach for the extraction of contigs from target species in whole genome sequencing of symbiont(s) together with the host. It uses taxonomic affiliations obtained through comparison to reference databases, coupled with a composition based strategy to identify genomic contigs of the target organism(s). The workflow has four main phases: (i) data are prepared for subsequent analysis; (ii) taxonomic affiliation are associated to contigs and used to train supervised machine learning models (SVM, RF) on k-mer frequencies; (iii) the models are used to predict the taxonomy of unlabeled contigs; (iv - optional) unsupervised clustering (DBscan) can be used to refine the classification at lower taxonomic ranks.

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 requires the following R packages:

- 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 a sql file that stores the information for the conversion. This file (`accessionTaxa.sql`) have to be downloaded and compiled. We make the user note that following the instruction for 'preparation' and use `prepareDatabase('accessionTaxa.sql')` command, a file containing only nucleotidic NCBI

accession number will be generated; for using SeqDex with proteins, then the 'Manual preparation database' instruction must be followed. Moreover, while this step needs to be done only once it has to be rerun if the database change.
SeqDex uses the standard filename '`accessionTaxa.sql`' so 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 steps. In the supervised section, SeqDex needs BLAST databases for defining taxonomy labels of the contigs:

The minimal requirement databases are (i) a nucleotide database to which comparing the contigs, and (ii) the RDP rRNA database to which comparing the 16S genes found by `Barrnap (`download it form https://rdp.cme.msu.edu/misc/resources.jsp, in the unaligned format and then use `makeblastdb` from the BLAST plus suite to obtain the BLAST database. Do not remove the fasta file, as it is used by SeqDex to retrieve taxonomic information).

When the target endosymbionts are particularly distant from relative genome published online, or when the sample under analysis is particularly complex taxonomic affiliation from a nucleotide-nucleotide comparison can be insufficient to train the machine learning models. In these cases, coupling nucleotide and protein derived taxonomy may be useful. To do so, the user has to install two optional dependencies (`prodigal` and `diamond`) as well as download a protein database to be formatted using diamond (see instructions for building the database).

Please note that to use protein-derived taxonomy affiliations you must rerun the Taxonomizr preparation step to include proteins in the database (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) a classification/prediction through machine learning, and (iii) a final clustering and prediction step.

3.1. Input files

The inputs to SeqDex.sh are: (1) the basename of the contig/scaffold assembly file in fasta format, and (2) the basename of the sam file obtained when mapping reads to the assembly. The quality of the assembly can influence the capability of the model to discriminate organisms, so please take care in this step. We suggest to evaluate sequencing quality by using FastQC and to remove adapters and low quality bases with Trimmomatic or similar tools. If the sequencing has been performed in paired end mode, it may be important to check the overlapping and merge the overlapping mates by using FLASh. At this point assemble the reads with SPADes or an equivalent assembler at several k-mer lengths to identify the best assembly using Quast. At this point, reads can be mapped back on the assembly. 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 do not differentiate among single reads and "mates" (here used in the general sense of the two reads composing a pair), we calculate the coverage for single and paired reads separately and then sum them. The outputs are saved in the Coverage folder.

3.2.2 Taxonomic affiliations

SeqDex needs taxonomic affiliations to train the machine learning models. `SeqDex.sh` by default uses only nucleotide-derived taxonomic affiliations (`TAX=NT`). Insert the name of the nucleotide database (`NTI`) and the path to this file (`NT`) by editing the `SeqDex.sh` file. SeqDex will run blastn by using the contigs as queries. SeqDex `Taxonomy.R` R script then reads the blast output to : (1) convert the NCBI codes into taxonomy information; (2) filter hits by the alignment length (`--aliLength`, default= 200 bp) and the 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 taken by SeqDex `Taxonomy.R` R scrip to: (1) convert the NCBI code into taxonomic informations; (2) filter hits by the alignment length (`--aliLength; --aliLengthNR`, default= 70 aa) and the 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 the Superkingdom level to select only bacterial contigs, and then at the level of bacterial classes, to select the class of interest (e.g. *Alphaproteobacteria*, see below). To do this, you need to tell to the model to prepare the taxonomic information for each level considered (Superkingdom and Class) by providing a comma separated list taxonomy levels desired, from superkingdom to species (run `Rscript Taxonomy.R --help` to see a full list of options; `--taxaLevels`, default=superkingdom). We suggest not to go too deep in the taxonomy levels: most of the times, when the sample under analysis shows low complexity (only one symbionts with high coverage), the classification at Superkingdom level is enough to then clustering and obtain the contigs relative to the target organism (this can be guided by preliminary rtPCR on 16S, for instance).
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 parts by manually copying commands from `SeqDex.sh` to the terminal.

3.2.3 Mate network construction

Nucleotide and/or protein derived taxonomic affiliation can be too scarce information to build satisfactory predictive models. This is imputable to the peculiarities of the organism under analysis: stable obligate endosymbionts often show reduced genomes, 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.
This will reduce the number of labeled 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 the computational time and resources for little improvement. Also, due to possible erroneous taxonomic affiliation of some 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 complete 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!
These graph, with the parameters and the rules described below, will be applied: (i) when parsing the taxonomy derived from nt and/or nt to transfer the affiliations to connected unlabeled contigs; (ii) after each machine learning prediction step, to correct errors or give additional support; (iii) after the clustering, to collect connected contigs which may be erroneously assigned to another group, but also to retrieve the contigs shorter than the threshold for length (see below).
Rules to transfer taxonomy information:

1. if the CC has size 2 and only one vertex is labeled, then the label is always transferred;
2. if the CC has size 2 and the vertices have incongruent labels then: during (i) the two are maintained as they are while in (ii) and (iii) all the contigs in the CC will be labeled as ‘misclassified’.
3. if the CC contains more than two vertices and there is only one type of label, then this is always extended;
4. if the CC contains more than two vertices, there are two labels and the underrepresented have a frequency less than `--mixedComponents`: during (i) the unlabeled contigs receive the most frequent label, whereas the second label is nonetheless maintained; however, in (ii) and (iii) the incongruent labels **are considered as ‘erroneous’ and corrected**;
5. if there are more than two labels: in (i) all the labels obtained through the BLAST search are kept, but no transfer occurs; in (ii) and (iii) all the contigs in the CC are labeled as ‘misclassified’.

In (i) the user can choose to use all contigs in each CC to transfer the taxonomy affiliations with the rules described above, or to limit the transferring to a certain distance from the taxonomy labeled contigs (`--taxTransfer`, default = all).

3.2.4 rDNA 16S

The 16S genes in the sample have to be located and their likely origin has to be identified at the end of the clustering step, when SeqDex looks for the cluster with the rRNA gene with the target taxonomy affiliation and the highest coverage among all the 16S rDNA genes present (if any other is present). `Barrnap` identifies putative 16S rRNA genes, that are then compared to the RDP database. 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`) has to be specified in `SeqDex.sh`.
The output of blastn is analyzed by `rRNA16S.R`, with the selection of only one among all the taxonomic affiliations obtained for each 16S contigs. 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 chose to consider only putative genes of at least 500 bp (change is possible, editing the `rRNA16S.R` script).

3.2.5 K-mer frequencies

K-mer frequencies are used both to perform the machine learning classification and the clustering. In the `GCKmersCov.R` script, k-mer frequencies are calculated by accounting for reverse, complement and palindromes. Frequencies are calculated by normalizing the counts by the 1000/(length of the sequence).

SeqDex calculates 3-mers frequencies by default (`--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 the trimers could implicitly summarize the information on codon usage and on coding density, which is very different in Bacteria and Eukaryotes. Moreover, the phylogenetic signal of k-mers increases with the length, but this introduces the necessity for setting a threshold on the length of contigs that becomes larger for longer k-mers. Using trimers reduces this problem, as there are only 64 combinations that reduces to XX variables thanks to reverse complementariety and palindrome compression.
Nonetheless, the user can choose whichever k-mer length but going above 5-6 can cause memory and computational time to increase a lot.
In `GCKmersCov.R` also calculates the total coverage per nucleotide by first summing for each contigs the single end reads coverage with half of the paired end reads coverage, to then divide it by the size.
The output of `GCKmersCov.R` is saved in the Coverage folder and it contains the k-mers frequencies, the GC content, the nucleotide coverage and the contigs length.

3.3. Phase 2 - machine learning classification

Once completed, the output files of the phase 1 became the input file for phase 2, which couples K-mer frequencies with taxonomic affiliations to train machine learning models that are at the basis of the prediction of taxonomy for all contigs. The `SVM.R` and `RF.R` scripts both take as input the table with K-mer frequencies (`--gcCovKmersTable`), 16S rRNA gene table (`--rRNA16S`), the taxonomy table (`--taxonomy`) and the mate CC network (`--network`). Together with the network, arguments `--Edges`, `--VerticesDegree`, `--componentSize` and `--mixedComponents` are needed, as described before.

Only contigs longer than `--minContigLength` and having a label assigned in the taxonomy table will be used to build the machine learning models (labeled contigs viz unlabeled). The labeled dataset is randomly split into training and test set, where the former represents two thirds of the total and is used to train the models while the latter is used to test the models and to calculate error rates. **This procedure is repeated, with a novel split of the dataset into train and test each**

time. The number of model build is user defined by setting `-nmodel` (default = 100). The percentage of erroneous contigs/length is reported as a cumulative measure over all 100 times permuted models and therefore represents a measure that is largely independent on the identity of the contigs in the train/test datasets. The scripts calculate sensitivity, precision, accuracy and F1 score based on both the number of contigs and the cumulative length on the comparison between empirically inferred taxonomy and the predicted.

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 affiliations 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 obtained through blast, is integrated and eventually corrected using the mate CC network discussed before and following the same rules described above. During this step uncertain predictions can be corrected. If there are more taxonomical affiliations than allowed by the mate CC network 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 to avoid thrashing sequences which may turn out to be informative. Also, these scripts calculate a homogeneity (H) index, defined as the number of homogeneous CC divided by the total number of CC (both homogenous and heterogenous in terms of taxonomic predictions). It goes from 0 (all heterogeneous) to 1 (all homogeneous) to give an idea of how much the CCs are likely representing contigs from the same genome. When this value is low, we suggest not using the network at all.

The user can choose to 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). Nonetheless, the entire strategy, mainly for the 100 iterations, requires several hours. To reduce execution time, the user can control the number of iterations.

We developed our SVM and RF scripts to allow easy customization of critical parameters for both. Support vector is sensible to `--cost`, `--gamma` and `--cross` values. Our implementation automatically selects best cost and gamma from a provided interval of values. The user can choose to use default intervals (`--cost`: 1e-1,1e3; `--gamma`: 1e-5,1e-1; `--cross`: 5) or provide its own. Also, the SVM implementation in e1071 automatically selects the best kernel type for the input data. For further information see https://cran.r-project.org/web/packages/e1071/index.html.
The randomForest package used by SeqDex (in `RF.R R` script) automatically chooses one of the major influent parameters of the random forest algorithm (the number of variables tried at each split, see https://CRAN.R-project.org/package=randomFores). However, our implementation allows the user to choose the number of trees 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 to the clustering (Phase 3) by setting `CLUSTER=yes` (default). When the user avoids running the clustering, 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-mer frequencies (`--gcCovKmersTable`), 16S genes (`--rRNA16S`), mate network (`--network`), and the Phase 2

output files (`--modelOutput`). If the machine learning prediction is done at several taxonomy ranks, only one of the taxonomy files need to be provided, as also the last prediction output.
The `Cluster.R` script reduces the k-mer frequencies to a user defined number of components (`--ncomponents`, default= 2). As the uwot packages performs 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 clustering only on these new components or also on GC content and/or coverage value (`input`, default= Kmers). In this way the user can combine the *Blobology* and the higher order composition analysis spirits in the clustering step. Combining these variables at this step allows to give some more weight to k-mers (2 variables) than to coverage or GC (1 variable each).

After obtaining these new dimensions, DBscan clusters the data. The script calculates a `minPts` value, as equal to the logarithm of the number of contigs used in clustering, and on this calculates 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 user-dependent part by rounding the kNN distances to the first digit, calculating 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 support to the assignments is evaluated by using the mate network, as previously done for the taxonomy and the machine learning prediction steps. However, here also contigs shorter than the provided length (`minContigLen`, default= 1000) will be included in the output by extending the cluster belonging to all members of the CCs.
At this point SeqDex searches for the cluster containing the target 16S gene that has also the higher coverage among all genes with the same label.
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 checkpoints to be able to rerun the analysis, if needed.

Coverage: coverage files (paired and single: `sambasenamefile_PAIRED_end.bed` and `sambasenamefile_SINGLE_end.bed`), the file with k-mer frequencies, GC content and total coverage informations (`gckCovTable.txt`), and the mate network file (`contigNet.txt`).

Taxonomy: (i) blastn output (`ContigsvsNt.txt`); (ii) if performed, prodigal predicted proteins (`prodigalContigs.faa` and gff format `ProdContigs`) and diamond output (`ContigsvsNr.txt`); (iii) the elaboration made by `Taxonomy.R`, which produces one file per taxonomy rank (`superkingdomTaxonomyIteration.txt`); (iv) 16S genes predicted by barrnap (`barrnap16s_contigs.gff`, `16sContigs.fasta`), the alignment on RDP (`16sContigsvsRDP.txt`), the RDP based taxonomy file (`RDP16s_taxa_mod.txt`); (v) the final elaboration performed by `rRNA16S.R` (`rRNA16sTaxonomy2.txt`).

SVMoutput and RFoutput: for each iteration performed, the SVM/RF script produces a file containing statistics (`superkingdomOutput_statsSVM.txt`, `superkingdomOutput_statsRF.txt`; the number indicate the iteration therefore the taxonomy rank), a file with k-mer frequencies of the target contigs selected at the end of the iteration (`superkingdomOutputSVM.txt`, `superkingdomOutputRF.txt`) and the environment (`superkingdomSVMModel.RData`, `superkingdomRFModel.RData`; so that an expert user can use the SVM/RF models outside SeqDex). In the stats file is reported performance statistics of the models constructed in each machine learning R script. In detail, for each model, the prediction performed on the test set is compared to the relative empirically inferred taxonomy; the cumulative

comparison of all test set are used to calculate cumulative percentage of error, sensitivity, precision, accuracy and F1 score by considering both the number of contigs and the total length. At the end of this file is reported the H-index.

When the clustering step is not performed, these folders also contain the fasta files of the contigs selected as belonging to the target organism by the machine learning algorithms (`superkingdomOutputSVM_contigs.fasta`, `superkingdomOutputRF_contigs.fasta`).
The output file with the target contigs contains:

- name of the contigs;
- GC content;
- k-mer frequencies;
- Coverage;
- TaxonDensity value: if it is 'NoBlastHit' means that the contig affiliations is not empirically inferred but has been predicted in this phase; if it is '-1' means that the label is derived from the mate CC network based extension;
- taxonomy labels converted into numbers (the order reported in stats file);
- percentage of times a contig has been classified in a certain taxonomical category, calculated over the predictions done over the 100 models performed. '-1' Could mean either that the contigs label is empirically inferred or that it comes from network based extension.

ClusteringOutput: if SeqDex is run enabling both SVM and RF, it will produce two clustering folders (`ClusteringOutputSVM` and `ClusteringOutputRF`) but only a folder named `ClusteringOutput` if only one machine learning algorithm was performed. The `Clustering.R` script produces: (i) the list of contigs in the target cluster, extended by using the mate network to (`OutputClustering.txt`); (ii) their sequences (`OutputClusteringSVM.fasta`, `OutputClusteringRF.fasta`); (iii) the clustering stats, reassuming the number of clusters obtained, the amount of contigs in each cluster, and the homogeneity index (`output_statsClustering.txt`); (iv) the final clustering table validated with the mate network (`extendedClusteringCC.txt`); (v) a file with the scatterplot of the contigs plotted by using two umap dimensions, and the scatterplot of the contigs colored by clusters (`Rplots.pdf`; cluster 0 is used to indicate outliers contigs, so the belonging to this cluster is not considered).
The `extendedClusteringCC.txt` contains:

- the names of the contigs;
- the new umap dimensions;
- the percent of assignment of a contigs to a label (calculated during the SVM/RF prediction phase);
- the identifier of the CC where each contig is found: if it is '-1' means that the contigs does not belong to a CC (it is an isolated vertex);
- The identifier of the cluster/group to which each contig belongs.

4. Running SeqDex

SeqDex can be run by using default parameters or by customizing one or more parameters.
In both cases, there are options that must be specified by the user. To do this, open the SeqDex.sh file as text and at least provide mandatory variables: the `THREADS` (the number of threads to be used), `NT` (the path to the nucleotidic dabases file), `NTI` (the name of the nucleotide database – without extension), `RDP` (the path to the RDP files – database and fasta file must be in the same folder), `RDPF` (the name of the RDP fasta file), `RDPI` (the name of the RDP database – without extension), `SCRIPT` (the path to the folder containing the SeqDex folder), `TAX` (specify `NT` or `NTNR`

to consider nt or both nt and nr-derived taxonomy labels), MLALG (SVM, RF, BOTH: self-explanatory), TRG (the target category at Class level – required only when CLUSTERING=YES), CLUSTER (YES/NO, whether to perform the DBscan clustering), ITER (taxonomy level on which perform SeqDex) and ITERTRG (target taxonomy category for ITER). If also protein-derived taxonomy affiliations are needed then set also the optional variables NR (the path to the protein database file) and NRI (the protein Diamond database),

4.1 Default

To run SeqDex by using default parameters, you only need to specify the above options. Then change permissions of the file, if needed, and run on the terminal

```
./SeqDex.sh basename_mapping_file basename_contigs_fastafile
```

Doing this, SeqDex.sh produces a taxonomy file at only the Superkingdom level, it predicts affiliation with both SVM and RF selecting only contigs from Bacteria (both obtained through BLAST or predicted by SeqDex) together with the misclassified ones, and then it performs the clustering step to retrieve the cluster with *TRG* 16S gene with higher coverage. The list of the default parameters is 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 ITER (taxonomy level) and ITERTRG (taxonomy category for ITER level) variables to obtain taxonomic affiliation and prediction on a different level than the default.

4.3 Iteration parameters

To perform the iterations (that is, to perform the training of the machine learning methods on multiple taxonomical categories, in series):

(1) change the default parameter in SeqDex.sh and provide a comma separated list of taxonomy levels (ITER), taxonomy target category for each level (ITERTRG) and the final target category at class level (TRG). By doing this, SeqDex will produce one taxonomy file for each level of interest (see above). These files will be named by the code corresponding to the taxonomy level pasted in front of the basename 'taxonomyIteration.txt' (i.e.: ITER=superkingdom,class, superkingdomTaxonomyIteration.txt for the first iteration; classTaxonomyIteration.txt for second iteration). These variables will be used also by both SVM.R and RF.R.
(2) provide to SVM.R and/or RF.R scripts a comma separated list, without spaces, of the taxonomy files produced by Taxonomy.R. By default, SeqDex will use ITER to reconstruct the name of these files, but if different filenames are needed, change the TAXFILE variable providing a comma separated list (without spaces) of the files (with also their path). Here, the scripts will (i) use the labels of the first iteration (superkingdom or superkingdomTaxonomyIteration.txt) to predict the unlabeled contigs; (ii) integrate the affiliations with the mate 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 targets provided;
(3) If the final clustering phase is not performed (CLUSTER=not), SeqDex will automatically recognize the last prediction file and return the FASTA file of the contigs in it. Else, if the final clustering step (CLUSTER=yes) is enabled, then the R script will use ITER together with MLALG to retrieve the last output prediction file and perform the clustering on it. However, if a different

file has to be used, than simply change `MODELOUT` variable providing the file together with its path.

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. We suggest not to go too deep in taxonomic levels: SeqDex at each iteration uses the contigs with the taxonomic affiliations obtained with `Taxonomy.R` which have been selected in the previous iteration; like this the number of contigs on which the model is applied will decrease at each iteration, and the error committed in each iteration will affect the next ones, making nearly impossible to go from Superkingdom to *Species* with affordable predictions. This is indeed also true for metagenomics tools, that not necessarily are able to provide species assignments with low error rate. Consider also that endosymbionts, especially the obligate ones, may not show high percentage of identity at species level even if close relatives genomes are available. 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 symbiont, the user can decide to:

(1) run a SeqDex run specifying a single target organism: just run SeqDex for the first time on one of the target organisms, and then take advantage of coverage, taxonomy, rRNA16S, k-mers frequencies files already prepared to speed up subsequent run on a different target organism. `SeqDex.sh` will detect them automatically and skip the command lines needed to produce them, which are highly time consuming. Take care to rename the files 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 deconvolution in sequences: (i) copy and paste all variable fields (line 7-143) and change them, in particular take care to change TRG, ITERTRG and the path of the input variable files to make sure it will find the file written in the first run and used by the subsequent other run (coverage, taxonomy, rRNA16S, k-mers frequencies); (ii) add command `mkdir name_new_folder; cd name_new_folder`; (iii) copy and paste the prediction and clustering, if performed, part (lines 301 to 517 of the file); (iv) add command `cd ..`; (v) repeat point (i)-(iv) for each subsequent target. Take care to change the path of each file that have to be provided to each new prediction-clustering R scripts. See example file.

Deconvolving multiple targets 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 groups 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 deeper in the taxonomy and arrive to a more correct separation.

4.5 Rerunning

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

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

(2) run only steps of interest, as each of our R scripts can be run independently by typing `Rscript namescript.R` in the terminal. Similarly, to see all the options available for the script, type

`Rscript namescrpit.R --help` or `Rscript namescrpit.R -h` . As before, old files 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` (and the corresponding parameters for the protein searches, if used). Samples with low complexity may not be highly influenced by these thresholds, while samples with high complexity (multiple targets, contaminating sequences of other bacteria, etc.) might need more affordable taxonomic affiliations; 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.

SeqDex.sh variables	Taxonomy.R	Default value	Meaning
BLAST	**--blast**	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)
ITER	**--taxaLevels**	1	Taxonomy level to consider. 1 means Superkingdom
ALILENGTH	**--aliLength**	200	Minimum alignment length to nucleotidic hit
XID	**--Xid**	70	Minimum percentage of identity between contigs and nucleotidic hit
TAXONDENSITY	**--TaxonDensity**	0.75	Minimum considerable percentage (in 0-1 scale) of belonging to a taxonomic affiliation of a contig
DIAMONDD	**--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	**--aliLengthNr**	70	Minimum alignment length to proteic hit
XIDNR	**--XidNr**	80	Minimum percentage of identity between contigs and proteic hit
NETWORK	**--network**	../Coverage/contigNet.txt	Mate CC network file with path
EDGES	**--Edges**	10	Minimum value for an edge to be considered
VERTICES	**--VerticesDegree**	5	Maximum vertex degree allowed
COMPONENTSIZE	**--componentSize**	2	Minimum size of the component CC
VERTEXDIST	**--taxTransfer**	all	If 'all' then all contigs in each CC is considered to transfer the taxonomy affiliations, whereas only the contigs distant no more than the provided value will be considered

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

SeqDex.sh variables	rRNA16S.R	Default	Meaning

BLASTRDP	**--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)
RDPTAXA	**--taxaRDP**	RDP16s_taxa_mod.txt	File with RDP taxonomy
MINLENGTH	**--minContigLen**	1000	Contigs length threshold: only contigs longer than this value will be considered

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

SeqDex.sh variables	GCKmersCov.R	Default	Meaning
(input)	**--contigs**	../"${2}".fasta	Contigs fasta file with its path. SeqDex uses the second argument to automatically find it
KMERS	**--Kmers**	3	K-mers length to be used
SINGLE	**--covSingle**	onlymapping_sorted_SINGLE.bed	Extended reads coverage file. If the R script is run in a different folder than this file, the path is aslo needed
PAIRED	**--covPaired**	onlymapping_sorted_PAIRED.bed	Paired end reads coverage file. If the R script is run in a different folder than this file, the path is aslo needed
THREADS	**--threads**	"$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.

SeqDex.sh variables	SVM.R	Default	Meaning
KMERSFREQ	**--gcCovKmersTable**	../Coverage/gckCovTable.txt	Output of GCKmersCov.R (with path)
RNA	**--rRNA16S**	../Taxonomy/rRNA16sTaxonomy2.txt	Output of rRNA16S.R (with path)
TAXFILE	**--taxonomy**	"${ITER}"	Output of Taxonomy.R (with path)
NETWORK	**--network**	../Coverage/contigNet.txt	Mate CC network file (with path)
CROSS	**--cross**	5	Maximum number of allowed erroneous contigs in SVM model construction
COST	**--cost**	1e-1,1e3	Range of cost to be tested to tune SVM model
GAMMA	**--gamma**	1e-5,1e-1	Range of gamma values to be tested to tune SVM model
SCALE	**--scale**	F	Boolean value meaning whether to

			scale the data or not (usually not needed)
MINLENGTH	**--minContigLen**	1000	Contigs length threshold: only contigs longer than this value will be considered
NMODEL	**--nmodel**	100	Number of models constructed
ITERTRG	**--targetName**	Bacteria	Name of the taxonomical label to be selected after prediction
ITER	**--TaxaName**	Superkingdom	Taxonomical level on which perform the prediction
THREADS	**--threads**	"$THREADS"	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7
EDGES	**--Edges**	10	Minimum value for an edge to be considered
VERTICES	**--VerticesDegree**	5	Maximum vertex degree allowed
COMPONENTSIZE	**--componentSize**	2	Minimum size of the component CC
MIXEDCOMP	**--mixedComponents**	0.2	Maximum proportion of alternative label in the component to consider it as erroneous and correct
VERBOSE	**--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.

SeqDex.sh variables	RF.R	Default	Meaning
KMERSFREQ	**--gcCovKmersTable**	../Coverage/gckCovTable.txt	Output of GCKmersCov.R (with path)
RNA	**--rRNA16S**	../Taxonomy/rRNA16sTaxonomy2.txt	Output of rRNA16S.R (with path)
TAXFILE	**--taxonomy**	"${ITER}"	Output of Taxonomy.R (with path)
NETWORK	**--network**	../Coverage/contigNet.txt	Mate CC network file with path

MINLENGTH	**--minContigLen**	1000	Contigs length threshold: only contigs longer than this value will be considered
ITERTRG	**--targetName**	Bacteria	Name of the taxonomical label to be selected after prediction
NMODEL	**--nmodel**	100	Number of models constructed
ITER	**--TaxaName**	Superkingdom	Taxonomical level on which perform the prediction
REPLACE	**--replace**	T	Boolean value meaning whether the RF sampling have to be done with or without replacement
NTREE	**--ntree**	500	Number of tree to be constructed by RF
THREADS	**--threads**	"$THREADS"	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7
EDGES	**--Edges**	10	Minimum value for an edge to be considered
VERTICES	**--VerticesDegree**	5	Maximum vertex degree allowed
COMPONENTSIZE	**--componentSize**	2	Minimum size of the component CC
MIXEDCOMP	**--mixedComponents**	0.2	Maximum proportion of alternative label in the component to consider it as erroneous and correct
VERBOSE	**--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.

SeqDex.sh variables	Clustering.R	Default	Meaning
MODELOUTSVM	**--modelOutput**	"${ITER}"	Output file of SVM.R to be used for clustering
MODELOUTRF	**--modelOutput**	"${ITER}"	Output file of RF.R to be used for clustering

KMERSFREQ	**--gcCovKmersTable**	../Coverage/gckCovTable.txt	Output of GCKmersCov.R (with path)
RNA	**--rRNA16S**	../Taxonomy/rRNA16sTaxonomy2.txt	Output of rRNA16S.R (with path)
TAXFILE	**--taxonomy**	"${ITER}"	Output of Taxonomy.R (with path)
NETWORK	**--network**	../Coverage/contigNet.txt	Mate CC network file with path
TYPE	**--input**	Kmers	Which type of input variable to be used for clustering (Kmers=only k-mers frequencies reduced by umap)
MINLENGTH	**--minContigLen**	1000	Contigs length threshold: only contigs longer than this value will be considered
TRG	**--targetName**	"$TRG"	Name of the target Class searched. SeqDex.sh uses TRG variable assigned in line 32
THREADS	**--threads**	"$THREADS"	Number of threads to be used. SeqDex.sh uses the variable THREADS defined in line 7
EDGES	**--Edges**	10	Minimum value for a edge to be considered
VERTICES	**--VerticesDegree**	5	Maximum vertices degree to be considered
COMPONENTSIZE	**--componentSize**	2	Minimum size of the component to be considered
MIXEDCOMP	**--mixedComponents**	0.2	Maximum proportion of alternative label in the component to consider it as erroneous and correct
MDL	**--modelType**	"$MDL"	Variable used to reconstruct input filename in case modelOutput=ITER

6. Example files

Download the `Example` folder from figshare.com/articles/SeqDex_example_data_zip/8845841. This folder contains files to run the analysis on the simulated dataset. 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 fast check of the mandatory dependencies without the need to have also reference databases for nucleotide and 16S genes. Also, we compared the contigs against a database composed by only the two genomes used to create this simulated dataset which indeed 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, also the file used to perform SeqDex on the *P. penetran*s dataset (`seqDex_Ppenetrans.sh`) can be found, 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.