HAPP: High-Accuracy Pipeline for Processing deep metabarcoding data

Authors: John Sundh[1,*], Emma Granqvist[2,*], Ela Iwaszkiewicz-Eggebrecht[2], Lokeshwaran Manoharan[3], Laura J. A. van Dijk[4], Robert Goodsell[5], Nerivania N. Godeiro[6], Bruno C. Bellini[7], Piotr Łukasik[8], Andreia Miraldo[2], Tomas Roslin[5], Ayco J. M. Tack[4], Anders F. Andersson[9,†], Fredrik Ronquist[2,†]

*These authors contributed equally

†These authors also contributed equally

†Corresponding authors: fredrik.ronquist@nrm.se and anders.andersson@scilifelab.se

Affiliations

[1]Department of Biochemistry and Biophysics, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Stockholm University, Box 1031, SE-17121 Solna, Sweden

[2]Department of Bioinformatics and Genetics, Swedish Museum of Natural History, Box 50007, SE-104 05 Stockholm, Sweden

[3]Dept of Laboratory Medicine, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Kansli ILM, BMC, I:12, Sölvegatan 19, SE-221 00 Lund, Sweden

[4]Department of Ecology, Environment and Plant Sciences, Stockholm University, SE-106 91 Stockholm, Sweden

[5]Department of Ecology, Swedish University of Agricultural Sciences (SLU), Ulls väg 18B, Uppsala 75651, Sweden

[6]Shanghai Natural History Museum, Shanghai Science & Technology Museum, Shanghai, People's Republic of China

[7]Department of Botany and Zoology, Biosciences Center, Federal University of Rio Grande do Norte, Natal, Brazil

[8]Institute of Environmental Sciences, Faculty of Biology, Jagiellonian University, ul. Gronostajowa 7, 30-387 Krakow, Poland

[9]Science for Life Laboratory, Department of Gene Technology, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

We introduce HAPP, a high-accuracy pipeline for processing deep metabarcoding data, leveraging data richness to enhance the signal-to-noise-ratio. Starting with denoised amplicon sequence variants, the pipeline consists of four steps: (1) additional chimera removal, using UCHIME and a strict sample-based approach; (2) taxonomic annotation, combining *k*-mer matching (SINTAX) to a reference library with phylogenetic placement (EPA-NG) on a reference tree; (3) OTU clustering using SWARM, an open-source algorithm with precision and recall comparable to RESL used in circumscribing BOLD BINs; and (4) noise filtering (NUMTs and sequencing errors), using a new algorithm introduced here, NEEAT, which combines “echo” signals across samples with detection of unusual evolutionary signatures among clusters with similar DNA sequences. HAPP computations are parallelized across taxa, making analyses tractable on very large datasets. The performance of HAPP was validated through extensive benchmarks, involving CO1 data from BOLD and Malaise trap data, demonstrating significant improvements over the state of the art.

Main

DNA metabarcoding—PCR amplification of a marker gene from an environmental sample followed by high-throughput sequencing[1]—has revolutionized the collection of biodiversity data thanks to its high efficiency and low cost. However, the data generated are noisy[2,3,4]. In addition to the authentic amplicon sequence variants (ASVs), data may include sequences containing chimeras (sequences of mixed origin, often formed during PCR amplification), remaining errors accumulated through the PCR and sequencing steps, and off-target sequences (in the case of mitochondrial marker genes, often in the form of nuclear DNA of mitochondrial origin, or NUMTs[5].

Interpreting the authentic ASV data presents additional challenges. Identifying the species in the sequenced sample typically involves the clustering of ASVs into operational taxonomic units (OTUs). Recognizing the conspecific nature of ASVs that differ by one or a few single-nucleotide polymorphisms (SNPs) may be straightforward but the OTU clustering is often complicated by conspecific ASVs separated by larger genetic distances. Finally, the short sequence length makes it difficult to annotate the ASVs taxonomically by matching them to reference libraries or placing them in phylogenetic trees. The last decade has seen considerable work towards addressing each of these challenges, and a plethora of tools are now available for cleaning, clustering and annotating metabarcoding data. Yet, how to select and combine these tools in generating optimal data on authentic taxa remains a daunting challenge.

At the same time, we are now seeing a significant expansion in the scope of DNA metabarcoding projects. The biomes investigated are sampled more intensely, the samples are sequenced more deeply, and the amplified sequences tend to be longer[6 7]. We refer to this as ‘deep metabarcoding’. The methods are also increasingly applied to macrobial diversity and to markers appropriate for these organisms, such as the mitochondrial gene cytochrome *c* oxidase 1 (CO1). The advent of deep metabarcoding justifies a reassessment of available tools and how they can be combined into an optimal workflow for the new data streams. The data richness and the application to macrobial diversity also provide new

opportunities for refining and ground truthing existing tools and developing new ones. In the present paper, we pursue these themes in developing a new high-accuracy pipeline for processing deep metabarcoding data, HAPP.

To properly benchmark any pipeline for deep metabarcoding data, realistic test data is needed. Ideally, the data should cover a range of taxa and biogeographic regions, as different faunas and floras pose different challenges. Here, we focus on a uniquely large data set generated in the Insect Biome Atlas (IBA) project and covering arthropod communities from Sweden and Madagascar[6]. The core data consists of 6,483 weekly Malaise trap samples (4,560 from Sweden and 1,923 from Madagascar)—estimated to contain 9 million specimens and tens of thousands of species—subjected to deep metabarcoding (ca. 1 M read-pairs per sample) of a 418 bp stretch of CO1. Importantly, the data cover a range of diverse taxa with dramatically different life histories, from small and ubiquitous decomposers like bristletails to large and rare predators like dragonflies, and two regions with dramatically different coverage in available reference databases. The power of the benchmarking is augmented by the fact that the knowledge of different taxa in the Swedish insect fauna is well characterized and partly virtually complete[8], and that a carefully curated and extensive CO1 reference library is available from the neighbouring country Finland[9].

In the evaluations of current and new tools underlying the design of HAPP, we complement benchmarking on these data with tests using the entire set of CO1 data from BOLD. This allows us to highlight specific properties through experimentation with reference databases, and to reveal the performance in a range of real settings, from analysis of well-known groups with nearly complete reference libraries to the other extreme. Specifically, we designed HAPP for post-processing of data from a standard pipeline producing denoised ASVs, such as DADA2, using four major steps: chimera removal, taxonomic annotation, clustering, and noise filtering.

It is well known that partial sequences of different origin may fuse together during PCR amplification to form hybrid ASVs, or chimeras. The chimeras typically do not match authentic sequences and can easily be mistaken for previously unknown OTUs. Standard metabarcoding pipelines include chimera removal but our initial benchmarking revealed that the default method in DADA2 was insufficient, suggesting that chimeras may be particularly problematic in deep metabarcoding. Therefore, we added a chimera removal step in HAPP. Several powerful tools exist, including ChimeraChecker[10], ChimeraSlayer[11] and Perseus[12]. Some of them rely on trusted reference databases, while others allow de novo detection of chimeras; the latter is more adequate for metabarcoding. We focused on UCHIME, one of the most accurate and fastest of the available tools[13]. It compares each ASV to other ASVs in decreasing order of abundance. Alignments between the query and candidate parents are constructed and scored according to a scheme designed to identify chimeras. We explored several different strategies in applying UCHIME to deep metabarcoding data.

Taxonomic annotation is an essential step in processing metabarcoding data. It can be achieved by matching ASVs directly to sequences in a reference library, or by placing them in a phylogenetic tree of reference sequences. In general, the former is expected to work well when reference libraries are complete or nearly so, while the latter should be more reliable when this is not the case[14]. We explored five different tools for taxonomic annotation in HAPP. For phylogenetic annotation, we chose the evolutionary placement algorithm (EPA), which places query sequences in a reference tree using maximum likelihood [14].

Specifically, we used the EPA-NG version[15] , followed by taxonomy assignment of query sequences using Gappa[16].

For matching against a reference library, we tested both *k*-mer based and alignment-based approaches. The RDP classifier[17], which is widely used in microbiome analyses, relies on a *k*-mer-based naive Bayesian approach, followed by bootstrapping to generate confidence estimates. Specifically, we used the DADA2[18] implementation of the RDP classifier. We also explored SINTAX[19] implemented in VSEARCH[20], in which a non-Bayesian *k*-mer algorithm finds the top hit and then applies bootstrapping to provide estimates of confidence. Finally, among the *k*-mer tools, we tested a multinomial naive Bayes classifier trained on the reference library, implemented in QIIME2[21]. For alignment-based similarity matching against the reference library, we focused on VSEARCH through the QIIME2 interface. It uses an alignment matching method similar to BLAST, and then computes a consensus taxonomy for the matches.

Clustering of ASVs into species is another essential step in processing metabarcoding data. For CO1 barcode sequences, often targeted in macrobial metabarcoding projects, the refined single linkage (RESL) algorithm has been particularly influential. It was developed to cluster the BOLD reference library[22] into putative species, called BINs (Barcode Index Numbers)[23]. The algorithm employs single-linkage clustering to identify putative clusters, followed by Markov clustering using simulated random walks[24] aimed at refining the resolution of cluster boundaries based on similarity and connectivity. Unfortunately, the RESL algorithm is not available as open-source code. Metabarcoding projects sometimes cluster ASVs indirectly according to RESL criteria by matching them to BOLD BINs. This approach is referred to as closed-reference clustering, or open-reference clustering if combined with de novo clustering of unmatched ASVs[25,26]. Whether RESL itself scales sufficiently to be feasible for de novo clustering of deep metabarcoding data remains an open question. Deep metabarcoding data differs from BOLD sequences in one important respect: the amount of sequence data per sampled species is much higher, potentially simplifying the task of recognizing species clusters. This suggests that other algorithms may be more appropriate for deep metabarcoding data.

In developing HAPP, we focused on three open-source algorithms designed for clustering metabarcoding data: SWARM, OptiClust and dbOTU3. SWARM shares many features with RESL. It is a two-phased, agglomerative, unsupervised single-linkage-clustering algorithm[27,28]. During the growth phase, SWARM delineates OTUs based on sequence differences between aligned pairs of amplicons, computed using *k*-mer comparisons and a custom global pairwise alignment algorithm. The maximum distance allowed between sequences in the same OTU is a tuning parameter set by the user. During the breaking phase, SWARM uses amplicon abundance information and the internal structure of OTUs to refine clustering results.

OptiClust is based on minimizing the discord between the desired maximum intra-cluster distance between sequence pairs (a tuning parameter) and the actual pairwise distances within and between clusters[29]. By default, the discord is measured using Matthew's correlation coefficient, balancing true positives (pairs below the threshold in the same OTU) and true negatives (pairs above the threshold in different OTUs) against false positives (pairs above the threshold in the same OTU) and false negatives (pairs below the threshold in different OTUs).

The dbOTU3 algorithm is unique in that it uses the distribution of ASVs across samples in addition to the genetic distances used by most other methods[30,31]. The sample distribution should be a particularly good source of information in clustering deep metabarcoding data. The algorithm has two tuning parameters: the genetic distance criterion and the distributional similarity threshold.

Finally, it has become increasingly obvious that off-target sequences represent a significant source of noise in metabarcoding; they may be particularly prevalent in deep metabarcoding because of the deep sequencing. In the case of mitochondrial markers, off-target sequences often represent NUMTs, which may be present in large numbers in the genome[32]. Several characteristics can be used to identify NUMTs. Protein-coding mitochondrial sequences typically lose their function when they are copied to the nucleus, resulting in an accumulation of mutations, insertions and deletions over time. One might also expect NUMTs to be present in lower numbers than the mitochondrial sequences they originated from, as the nuclear genome copy number is lower than that of the mitochondrial genome by a factor of 100–10,000[33,34]. However, many NUMT variants, represented by multiple copies each, can be present within a genome, and changes in primer binding sites can also bias PCR amplification rates toward generating more NUMT copies.

Initially, NUMTs are identical to their parent sequence, and cause no problem if they appear in the data. When sufficient time has passed, the NUMTs are easy to identify based on frame shifts or stop codons. It is the time in-between that is problematic, as the NUMTs can then be mistaken for authentic mitochondrial sequences representing distinct species.

One way of identifying NUMTs is to focus on evolutionary signatures, as NUMTs should differ from functional genes subject to strong constraining selection. Evolutionary signatures should also reveal amplification and sequencing errors, as well as off-target sequences that are not NUMTs. To utilize this signal, one can fit a profile hidden Markov model (HMM) to the protein coded by the target gene, and identify outliers in the metabarcoding data as likely NUMTs [35]. Another possibility, explored here, is to compare the evolutionary distance to the edit distance (number of changes needed to go from one sequence to another) of similar ASVs. At a given edit distance, pairs of authentic ASVs should have fewer non-synonymous substitutions than pairs including one NUMT.

A completely different approach is to look for “echoes” in the distribution among samples, as NUMTs should consistently co-occur with their parent sequences, but in lower copy numbers. The same may be true for many types of amplification or sequencing errors. The “echo” signal is explored by LULU[36], which looks for less abundant sequences matching the distribution across samples of more common ones that are within a predefined genetic distance (default is 84% similarity). As originally described, LULU groups the echo sequences with their parents, but the same principle can be used to filter them out as noise.

A third possibility is to filter out NUMTs and potential errors simply by applying an abundance threshold, as they are unlikely to generate ASVs with large read numbers[4]. The metaMATE tool[37] represents a particularly sophisticated use of abundance filters, optimizing them separately for different organism groups and datasets.

Here, we introduce a new noise filtering algorithm, NEEAT (noise reduction using echos, evolutionary signals and abundance thresholds), which combines these techniques, and we benchmark it against LULU, HMM filtering and abundance filtering.

The design of HAPP is based on the results of these efforts. It integrates all tools required for post-processing metabarcoding data into a highly configurable and scalable workflow.

Results

The HAPP pipeline reconstructs species-level OTUs from deep metabarcoding data. As input, it expects denoised ASVs and their counts across samples from a standard workflow, such as DADA2[18]. After an initial quality filtering that removes ASVs outside the expected size-range or with stop codons in the expected reading frame, post-processing is applied in the four steps described above: chimera removal, taxonomic annotation, OTU clustering, and noise filtering. We present the benchmarking results for each step first, and then discuss the design of HAPP.

Extra chimera filtration for improved data quality

We benchmarked four different strategies for removing chimeras from denoised ASV data. Specifically, we assessed whether chimera removal could be improved in deep metabarcoding through more reliable detection of chimeric parents across all samples (batchwise method), compared to relying entirely on finding parents in the same sample (samplewise method). For both methods, we tested lenient and strict requirements on chimera-parent co-occurrence (see Methods). We found that the two samplewise methods removed considerably more ASVs than the batchwise methods, while simultaneously improving the clustering results as indicated by the higher recall and smaller cluster:species ratios (both metrics reflecting the degree to which OTUs are matching and not over-splitting species; Fig. 1). Thus, the samplewise methods had a better signal to noise ratio. The strict samplewise method performed slightly better than the lenient alternative in terms of trusted ASVs (see Methods) or reads removed. Enforcing a strict chimera-parent matching criterion improved the false positive ratio considerably for the batchwise method (reflected by the smaller number of trusted ASVs and reads removed) but it did not improve the false negative ratio (reflected by the number of ASVs and proportion abundant ASVs removed), or the clustering results. In conclusion, the strict samplewise method performed best; it removed a large amount of chimeric sequences while retaining the vast majority of authentic ASVs.

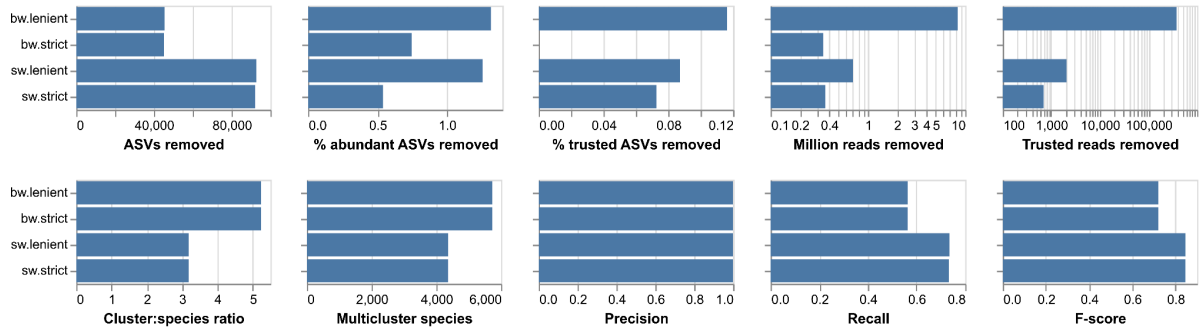

Fig. 1: Benchmarking of chimera removal methods. Each panel shows a quality metric for the four chimera removal methods tested (see Methods for detailed description of the metrics). The benchmarking was run on the early version of the Swedish IBA data (636,297 ASVs).

Improved taxonomic classification by combining *k*-mer and phylogenetic approaches

To benchmark the different taxonomic classification algorithms, we used a customized version of the BOLD database (see Methods) and generated different query sequences/reference database combinations, each with 1000 randomly sampled query sequences with a known taxonomy obtained from BOLD. These ranged from the easiest (case 1), where identical sequences to the queries remained in the database, to the hardest (case 5), where sequences of the same orders but not families remained in the database (Fig. 2a). For both query and database sequences, only the 418 bases of the CO1 gene corresponding to the barcode region sequenced in the IBA project were used.

We found that the *k*-mer based tools SINTAX[19] ('vsearch_sintax') and SKLEARN ('qiime2_sklearn') had the highest number of correctly assigned species when all sequences were kept in the reference database (case 1) and when sequences identical to the query sequence were removed but sequences belonging to the same species remained in the database (case 2; Fig. 2a). The third *k*-mer based tool, the RDP classifier ('dada2_rdp'), had significant problems even in these simple cases. The alignment-based VSEARCH tool ('qiime2_vsearch') showed low numbers of assigned species in both case 1 and 2 but results improved for higher taxonomic ranks. When the reference database was filtered to remove the sequences corresponding to either the species (case 3), genus (case 4), or family (case 5) of the query, SINTAX was more conservative than SKLEARN and made considerably fewer (incorrect) predictions at the ranks where reference sequences were missing. Interestingly, VSEARCH had the highest number of correctly assigned queries at the family level in case 3 (species missing) and case 4 (genus missing), indicating that this method works better than the *k*-mer based classifiers for higher taxonomic ranks when the reference database is incomplete.

Due to its overall good performance and robustness to parameter choice, SINTAX was chosen as the default option for taxonomic assignment in HAPP. However, it had a low classification rate at the order level when the family was missing in the database, which could be problematic for ecosystems with poor representation in the reference database. This was indeed observed when applying SINTAX on IBA data from Madagascar, but not when applied to the Swedish data (Fig. 2b), consistent with a more extensive and less cataloged arthropod diversity in Madagascar. Considering that phylogenetic approaches are potentially more efficient when close reference sequences are lacking, we included such a method (EPA-NG) in the benchmarking, utilizing a phylogenetic tree of 49,358 insects derived by extension and quality filtering of a published tree[38] (see Methods). Indeed, annotation rate was doubled compared to SINTAX and approached 100% at order level when the family was missing in the reference, and likewise drastically increased at family level when the genus was missing, while still giving low false annotation rates (Fig. 2a). Applying it to the IBA ASVs annotated as class Insecta or Collembola by SINTAX but unclassified at the order level more than doubled the annotation rate at the order level for the Madagascar data (from 28 to 58%), but only marginally increased the rate for the Swedish data (78 to 80%), consistent with the *in silico* test (Fig. 2b).

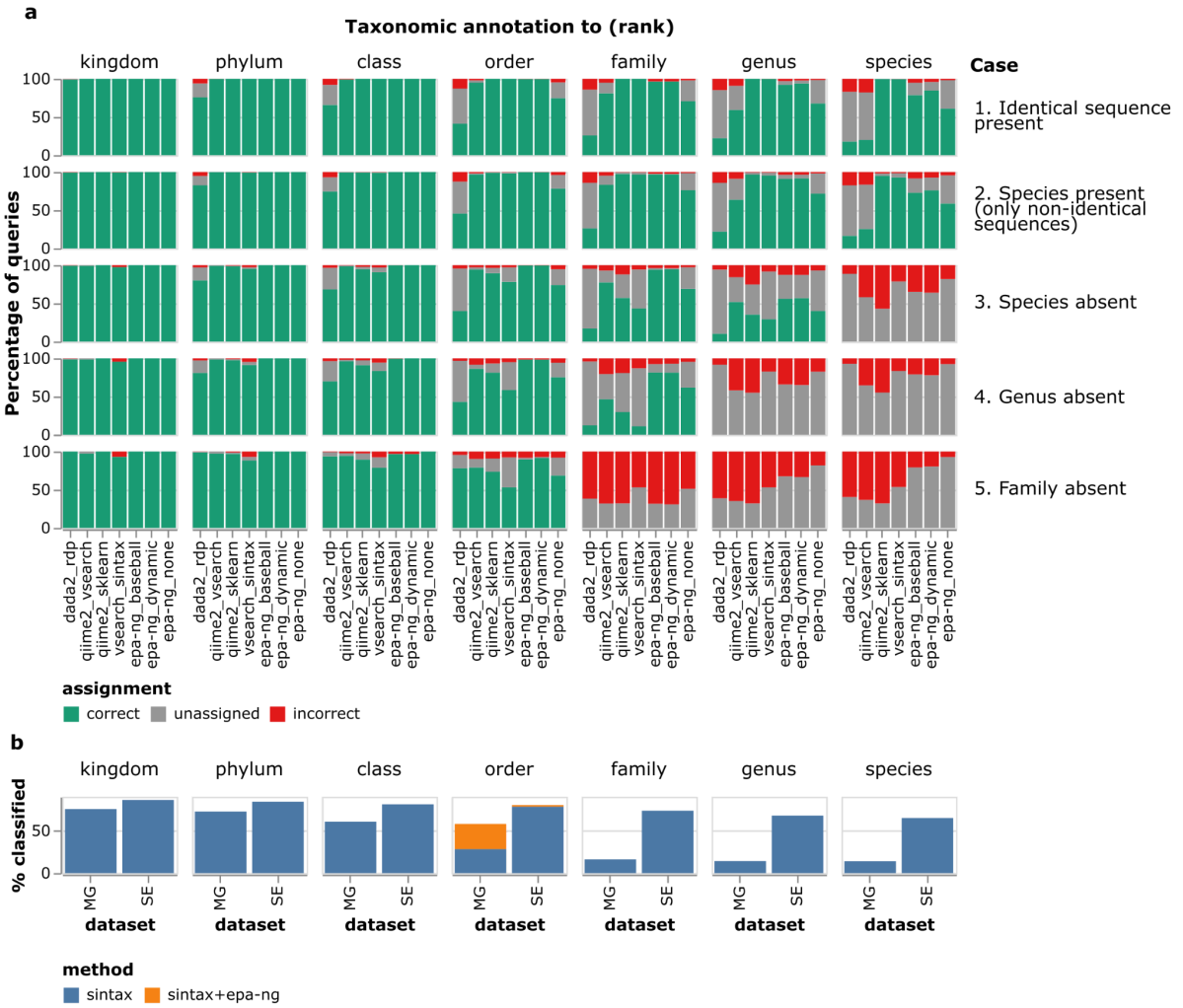

Fig. 2: Benchmarking of taxonomic assignment tools. a. Barplots of correct, unassigned and incorrect sequences for different ranks and test cases. 'dada2_rdp' = DADA2 assignTaxonomy (an RDP classifier), 'qiime2_vsearch' and 'qiime2_sklearn' = QIIME2 feature-classifier vsearch and sklearn, respectively, 'vsearch_sintax' = SINTAX algorithm implemented in VSEARCH, 'epa-ng.baseball', 'epa-ng.dynamic' and 'epa-ng.none' = EPA-NG phylogenetic placement with baseball heuristics, dynamic heuristics and no heuristics, respectively, followed by assignments with GAPPA. **b.** Percentages of sequences classified at each rank on two real-world datasets from Madagascar ('MG', 701,769 ASVs) and Sweden ('SE', 821,559 ASVs) using the SINTAX algorithm. For the taxonomic rank 'order' we also show the percentage of classified sequences obtained when running EPA-NG (dynamic heuristics) on sequences that were annotated as unassigned Insecta by SINTAX, then taking order-level assignments from the EPA-NG result.

Optimized sequence clustering to recover species-level OTUs

We benchmarked different ASV clustering methods on the Swedish IBA data, after chimeras had been removed, with performance being evaluated using the SINTAX assignments of ASVs to species. The precision (tendency of OTUs to consist of a single species) was high for all tools (in the interval 0.89-0.99; Fig 3a) but recall was lower, with mean values ranging from 0.09 to 0.82. This indicates that, under the parameters tested in our benchmark, the

tools were prone to oversplit ASVs, *i.e.* placing apparently conspecific ASVs into different clusters. This was especially evident for dbOTU3 which had the lowest recall value (0.025) of the tools tested.

Both SWARM and OptiClust use thresholds of sequence similarity to cluster ASVs, either in the form of maximum allowed 'differences' between sequences (SWARM, default=1) or a maximum ‘distance cutoff' (OptiClust, default=0.03). The default settings resulted in low recall values (0.80 for OptiClust and 0.31 for SWARM). Increasing the maximum allowed differences for SWARM and the distance cutoff for OptiClust notably improved recall, which reached ≥0.9 for SWARM at differences ≥9 and for OptiClust at distance cutoff ≥0.04, with only marginal decreases in precision. However, for OptiClust, cutoff values >0.07 resulted in a sharp decrease in precision with little gain in recall. Assessing the performance using the F-score, which is the harmonic mean of precision and recall, we found that OptiClust with a cutoff of 0.05 and SWARM with a difference of 15 gave the best, and highly similar, metrics (F-scores of 0.95 for both OptiClust and SWARM; Fig 3b).

To assess the performance of the open-source tools against the closed-source RESL algorithm[23], we downloaded the complete BOLD database (from 06-Jul-2022) and calculated precision and recall values (treating BIN assignments as ‘ASV clusters’) for 3,953,373 sequences with both BIN and species assignments (Fig. 3a, diamond; Fig. 3b, dotted line). The results show that RESL clustering of BOLD sequences into BINs performs on par with SWARM and OptiClust clustering of deep metabarcoding reads. Using closed-reference clustering of metabarcoding reads against BOLD gave similar results (Supplementary Fig. S1).

In terms of runtime, SWARM and OptiClust showed similar performance in our benchmark, with the former slightly ahead; both tools were considerably faster than dbOTU3 (Fig. 3c). Unlike the other tools, SWARM does not require a pairwise distance matrix to be computed beforehand, and it also has better multi-threading support than OptiClust. SWARM and OptiClust are still actively maintained, while dbOTU3 has been archived and has not been updated in the past 6 years. Because of its active developer community, its robust performance across settings, and its computational efficiency, we chose SWARM as the default clustering method in HAPP.

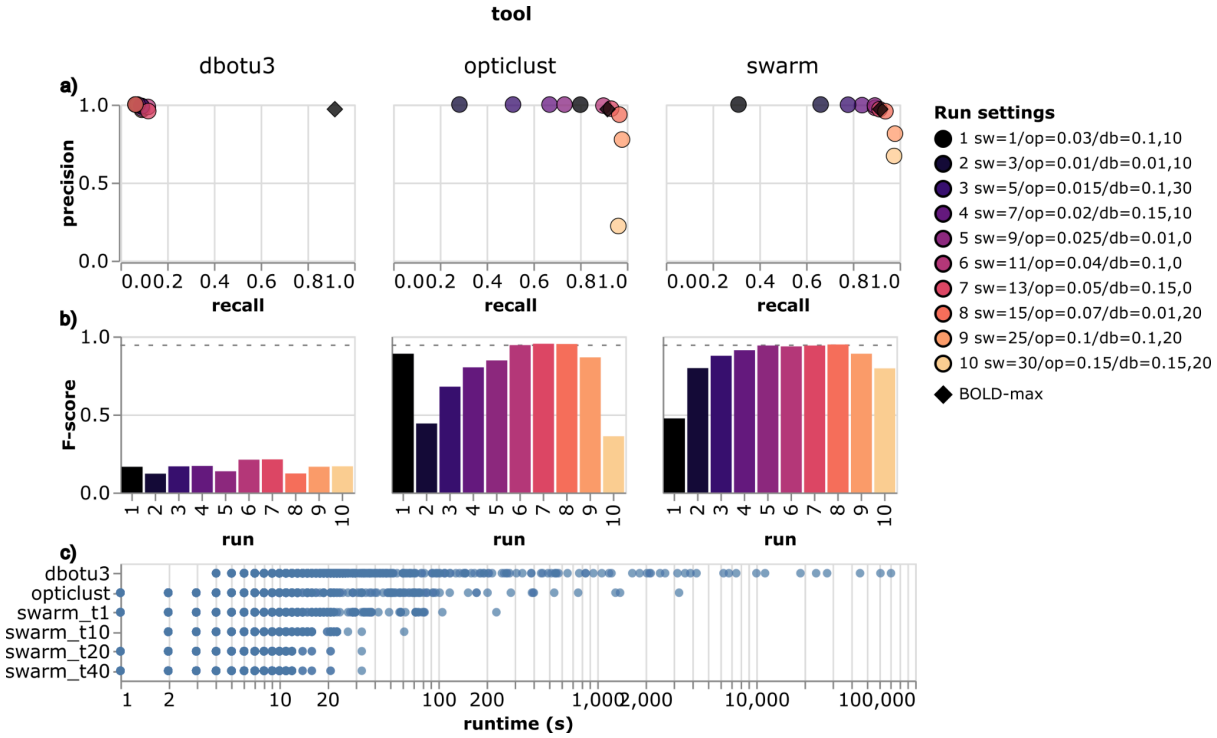

Fig. 3: Benchmarking of ASV clustering tools. a. Precision vs. recall for each run of the different tools. Colors correspond to the different runs and the legend shows the settings used: 'sw' is the difference cutoff in Swarm, 'op' the distance cutoff in OptiClust, and 'db' the distance and abundance cutoffs in dbOTU3. The diamond symbol shows the precision and recall of the RESL-generated BINs in the BOLD database ('BOLD-max'). **b.** F-score values (harmonic mean of precision and recall) of the different runs for each tool. The dashed line shows the F-score of the BINs in the BOLD database. **c.** Runtime statistics for the different tools obtained from the same run settings. Each point shows the time to cluster ASVs belonging to a single taxonomic family. For SWARM, the number of threads is indicated on the y-axis ('swarm_t10' = 10 threads).

A new algorithm for effectively removing NUMTs and other noise

Despite the extra chimera filtering step, we noted that the IBA data included a significant amount of noise, which was particularly obvious for well known groups in the Swedish fauna like Lepidoptera (butterflies and moths). However, existing noise filtering algorithms use overly simple criteria, focus on a single criterion, or are too computationally complex to be applied to deep metabarcoding data. Therefore, we designed a new, scalable algorithm based on a combination of criteria, NEEAT.

To address scalability, we apply NEEAT to clustered data rather than single ASVs, which has often been the case in previous algorithms. Also, we apply NEEAT to taxonomically partitioned datasets, allowing it to be run in parallel using the same strategy adopted by HAPP for other steps in the pipeline.

Taxonomy. In a detailed analysis of Swedish IBA data for Lepidoptera[7], we noted that taxonomic annotation failures at higher ranks are often associated with non-authentic CO1 sequences. Therefore, NEEAT includes a filter that removes clusters with annotation failures at higher taxonomic ranks. The rank at which failures are considered to signal noise is a

user-provided setting (default order level). NEEAT uses three additional criteria to filter out noise clusters: echoes, evolutionary signatures, and read abundance.

Echoes. Like LULU, NEEAT searches across samples for unusual co-occurrence patterns among genetically similar clusters. Noise OTUs—both NUMTs and recurring sequencing errors—may be expected to consistently occur in samples together with their parent OTUs (containing the corresponding authentic sequences) but in lower read numbers. Tuning parameters of this filter include the maximum distance between echo and parent cluster, the minimum fraction of echo samples with parent present, the max read ratio threshold and type ('max' or 'mean' across samples), and whether a significant correlation between echo and parent read numbers is required.

Evolutionary signatures. For protein-coding sequences, NEEAT compares evolutionary and edit distances among sequences to find clusters likely to represent NUMTs, off-target sequences, or chimeras missed in previous filtering steps. Specifically, we use an evolutionary distance taking chemical similarities between amino acids into account, and a metric that is sensitive to unexpectedly large evolutionary distances at small edit distances (see Methods). To improve performance, searches are limited to the *N* nearest neighbors (in terms of edit distances), with *N* being a tuning parameter. NEEAT uses two versions of this filter, a local filter requiring a minimum sample overlap between the clusters, and a global filter without this requirement.

Abundance threshold. Finally, NEEAT also implements an abundance threshold for the mean or max (default 'max') number of cluster reads.

We benchmarked NEEAT against abundance filtering, as in metaMATE, profile HMM filtering, and LULU. The tests used the Hexapoda (classes Insecta and Collembola) subset of the final Swedish IBA dataset, after chimera filtering, annotated by SINTAX and clustered by SWARM (d=15) as described above. We measured false positives (authentic OTUs removed) as a weighted average of the number of unique BOLD BINs and trusted ASVs (see Methods) removed, and false negatives as a weighted average of the number of remaining duplicated BOLD BINs and spurious OTUs—OTUs in well-known insect families (>99% of expected species recorded[8]) not matching species in BOLD.

NEEAT significantly outperformed abundance filtering, HMM filtering and LULU abundance filtering in this benchmark (Fig. 4a). In tests of individual filters, the NEEAT echo algorithm did slightly better than LULU (Supplementary Fig. S2), and the NEEAT evolutionary algorithm (both local and global) outperformed HMM filtering by a large margin (Supplementary Fig. S3). The performance of the abundance filter was insensitive to the type of reads used (raw, calibrated, proportional to sample reads, or proportional to non-spike-in reads); however, filtering on the max read number was better than filtering on the mean (Supplementary Fig. S4). The false positive rate increased fairly rapidly with increasing thresholds for all major orders except the Diptera.

To assess the effectiveness of the taxonomic annotation filter, we compared the proportion of reads that would be removed by applying the criterion at different ranks for the IBA data from Sweden and Madagascar. The barcode reference library and reference tree include virtually all Swedish Hexapoda families, suggesting that most of the annotation failures at this level or above are likely to represent noise. As the noise level should be similar, a significant drop in the proportion of retained reads for Madagascar—with incomplete reference library and tree—would suggest that annotation failures include many false positives, that is, authentic

OTUs. The results (Fig. 4b) show that the phylogenetic placement method (EPA-NG) is considerably more robust to gaps in the reference tree than alignment-based (VSEARCH) or *k*-mer-based methods (SINTAX) are to gaps in the barcode library, despite the latter including two orders of magnitude more sequences. The drop is larger for reads than for unique ASVs or clusters, suggesting that annotation failures mainly affect the dominant taxa in Madagascar samples (Supplementary Fig. S5). The lower annotation success of EPA-NG at the family level and below is likely due to poor coverage of the most species-rich insect families in the reference tree (Fig. 4b).

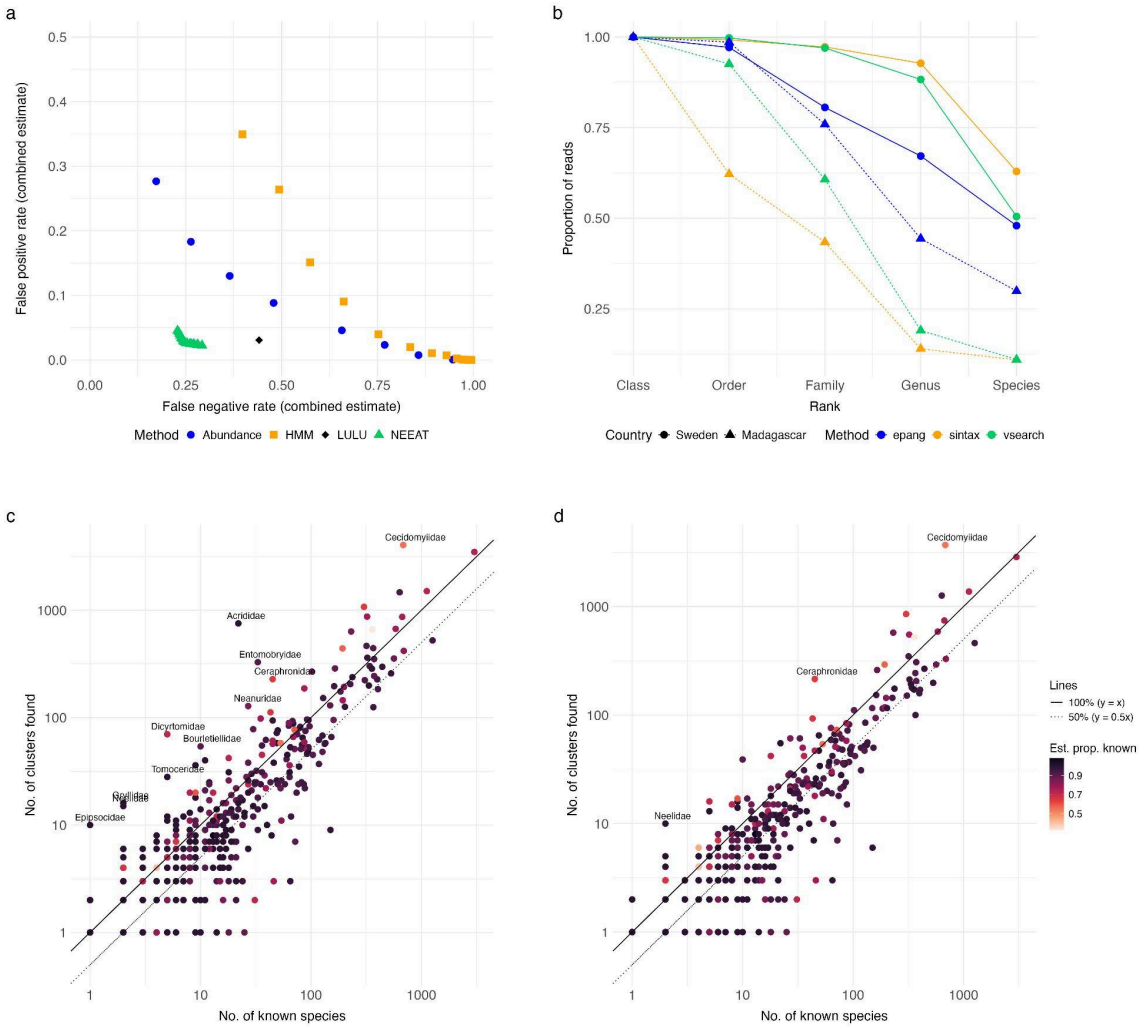

Fig. 4: Benchmarking of noise filtering tools. **a.** Performance of noise filtering on the Swedish IBA data on hexapods using an abundance threshold (raw counts, with varying cutoff in terms of max number of reads), an HMM profile model (with varying bitscore cutoff), the LULU algorithm (default settings) and the NEEAT algorithm (Pareto front of settings explored here). **b.** Annotation success of different taxonomic annotation algorithms (sintax, a k-mer-based tool; vsearch, an alignment-based tool; and epang, a phylogenetic placement tool) on Swedish and Malagasy IBA data. **c-d.** The number of OTUs found in the Swedish IBA data before (**c**) and after (**d**) noise filtering with NEEAT, plotted against the number of species recorded from Sweden. Each point represents one family; the colors represent the estimated proportion of the total Swedish species diversity that is currently known[8]. Analysis

of the Lepidoptera[7] suggests that the IBA data typically contains slightly more than 50% (dotted line) of the known species (full line) in well-known insect groups (dark purple). Poorly known families (lighter colors) may be represented by more species than currently known.

Overall, the NEEAT algorithm is quite effective in reducing noise in the Swedish IBA data (Figs. 4c, d). Detailed analysis of the Lepidoptera suggests that the data include roughly half the Swedish fauna of most families[7]. After NEEAT filtering, only a handful of well-known hexapod families fall significantly outside this range (Fig. 4d).

The HAPP pipeline provides a flexible and performant workflow

We designed HAPP as an open-source Snakemake[39] workflow that performs pre-filtering (ASV length and stop codons), chimera removal, taxonomic annotation, ASV clustering of non-chimeric sequences, and noise removal by NEEAT (Fig. 5). The default choice of tools and settings reflects the benchmarking results described above but HAPP supports the full range of clustering and annotation tools included in the benchmarking, and can easily be extended to accommodate additional tools. The workflow supports splitting the input ASVs by taxonomy (e.g., by class or order) and running the clustering software on each split separately, with the assumption that ASVs assigned to different taxa at this level should not cluster together. This can be useful for large datasets because it lowers the number of pairwise comparisons and allows for parallelization of several steps in the workflow. HAPP is available at https://github.com/insect-biome-atlas/happ.

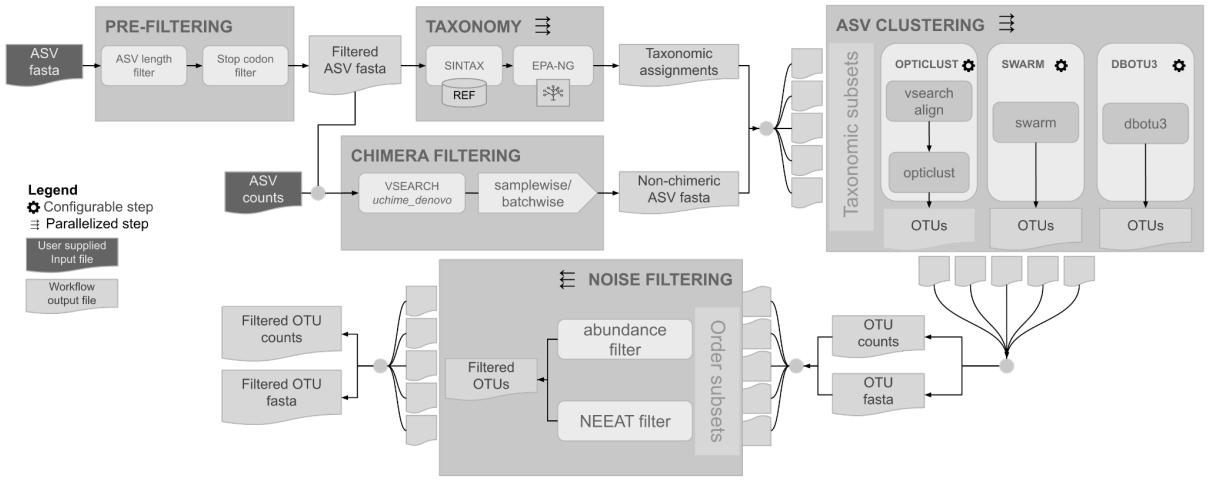

Fig. 5. Design of the HAPP pipeline. After prefiltering to remove sequences of unexpected length or with in-frame stop codons, the pipeline involves four steps: (i) chimera filtering, (ii) taxonomic annotation, (iii) clustering and (iv) noise filtering. Parallelization is implemented by splitting the input into equally sized chunks in the taxonomic assignment step, and into taxonomic groups in the clustering and noise filtering steps.

Discussion

Deep metabarcoding projects come with unique challenges and opportunities. The deep sequencing and the large number of samples can reveal species that would otherwise have been missed. At the same time, the data comes with significant numbers of chimeras,

NUMTs and other types of noise. Thus, utilizing the full potential of deep metabarcoding requires powerful bioinformatic tools.

In this paper, we presented HAPP, a pipeline designed specifically for processing deep metabarcoding data. The pipeline includes optimised procedures for chimera removal, taxonomic annotation, OTU clustering and removal of noise OTUs. For the former steps, HAPP is based on existing tools, while a novel algorithm (NEEAT) was developed for the removal of noise OTUs. Each of the steps was optimized using a combination of *in silico* tests and benchmarking with real metabarcoding data.

The IBA data[6] used for benchmarking HAPP allowed us to perform tests across a range of settings, from small or moderately diverse groups with virtually complete reference libraries (many Swedish insects) to megadiverse taxa that are largely missing in current databases (some Swedish arthropods, most Malagasy taxa). Taken together, the benchmark results suggest that HAPP will be useful for many types of deep metabarcoding projects.

Our results emphasize the importance of filtering out chimeras from deep metabarcoding data. We found that the standard chimera filtering ('removeBimeraDenovo') in a commonly used ASV denoising pipeline (DADA2[18]) was inadequate. Applying a second filtering pass in HAPP identified an additional 15% of ASVs in the Swedish IBA data as chimeras. Tweaking the DADA2 settings might bring down the frequency of chimeras, but the extra chimera removal in HAPP has a negligible rate of false positives, so there is little gain in bypassing it.

Among the taxonomic annotation tools we tested, SINTAX[19] performed best in our *in silico* tests using CO1 sequences from BOLD, in particular at lower taxonomic ranks. However, both our *in silico* tests and IBA data benchmarks revealed that SINTAX was overly conservative in assigning queries at higher taxonomic ranks. This was particularly obvious for the Madagascar data (Fig. 4b; Supplementary Fig. S5) despite new taxa at this level being unlikely to occur in the data. This profile contrasts starkly with that of phylogenetic placement using EPA-NG and Gappa[14–16], which performed well at higher taxonomic ranks while being unreliable at lower ranks (undoubtedly due to the poor coverage in the reference tree). This suggests that combining SINTAX and EPA-NG may be advantageous. We found that EPA-NG doubled the annotation rate at the order level for the Madagascar IBA data, while having only marginal effect on the Swedish data, consistent with the expected differences in the completeness of the reference libraries. As the available reference trees increase in coverage, we expect phylogenetic methods to gain ground. Given the current state, where no method can handle all contexts, we designed HAPP to support a range of taxonomic annotation tools and combinations of them.

BOLD and similar reference databases are invaluable resources for taxonomic annotation. However, relying on closed-reference clustering, where metabarcoding reads are assigned to clusters by matching them to existing records, is not adequate for the megadiverse and poorly covered organism groups often targeted in metabarcoding. Deep metabarcoding is likely to yield substantially more data per target OTU than contained in the reference databases, allowing OTUs to be circumscribed more effectively through de novo clustering. We think this explains why OptiClust[29] and SWARM[28] perform so well in our benchmarks, better than closed-reference clustering using BOLD BINs (Supplementary Fig. 1). Clearly, the computational efficiency of OptiClust and SWARM is sufficient for de novo clustering of

deep metabarcoding data. Unlike the RESL algorithm used by BOLD[23], both of these algorithms are open source, supporting reproducible science and community-based refinement.

Filtering out data based on read numbers is often used as the main or sole cleaning step for metabarcoding data[37,40]. However, our results suggest that such abundance filtering is ineffective because deep metabarcoding data contain many authentic OTUs with few reads: the false positives rise quickly as the abundance threshold is increased (Supplementary Fig. S4). However, because most groups seem to be associated with many low-read but authentic OTUs in our tests, we find it difficult to justify the use of taxon-specific thresholds as recommended by Andújar et al. (2021)[37].

The widely used LULU tool[36] can be understood as a special case of our echo algorithm, but we succeeded in improving performance slightly by ordering clusters first in terms of sample numbers and then read numbers, instead of just read numbers, allowing more echoes to be correctly identified when passing down the list. We also found that relaxing the read ratio threshold, targeting the mean rather than the max across samples, usually leads to performance gains, in contrast with LULU recommendations.

The work on profile HMMs[35] represents one of the most sophisticated attempts yet to use evolutionary signatures to separate noise from signal in metabarcoding data. However, our benchmarking indicates that it is quite ineffective compared to our approach, using pairwise comparisons of the ratio between edit distance and evolutionary distance. We think the reason for this is that the CO1 protein varies across the insect phylogeny, such that a single HMM profile is insufficient to accurately capture small deviations. This problem may be aggravated by taxon biases in the training data, resulting in poor fit of the model to the dominant groups in the metabarcoding data.

The optimal NEEAT performance was obtained with stringent settings for each of the component filters. The Pareto front was dominated by combinations using low thresholds for the read ratio in the echo filter (much lower than LULU defaults), high thresholds for the distance ratio in the evolutionary signature filter, and a low maximal read number in the abundance filter. Our results suggest that annotation failures can also be used for noise filtering but that the taxonomic rank at which it is applied needs to be adjusted according to the completeness of the reference database. For instance, detailed analysis of the Swedish IBA data for Lepidoptera suggests that SINTAX annotation failures at the family level are often associated with noise ASVs in this dataset[7].

Even though HAPP makes substantial improvements over the state of the art, there is clearly room for further improvement. Both the echo and evolutionary signature algorithms are heuristic in that they start from the most abundant cluster, assume it represents an authentic marker sequence, and then sequentially test and add sequences to the trusted set. An obvious improvement would be to revisit and continuously update the circumscription of the trusted set, even though it would increase the computational complexity. The success of the phylogenetic approach in improving the taxonomic annotation at higher ranks—despite the limited size of the reference tree—suggests that noise filtering could also be improved by considering phylogenetic information. The fact that our pairwise evolutionary signature algorithm outperformed the global HMM profile approach points in the same direction.

Finally, switching to a proper probabilistic framework, and explicitly considering error models in that context, would also seem worthwhile exploring.

Despite the potential for further improvement, we think our benchmarks show that HAPP is sufficiently mature to be trusted for standard monitoring of well-known faunas, while at the same time being powerful enough to be used in exploring the diversity and composition of species groups or regional faunas that are largely unknown or poorly represented in reference databases.

Methods

Benchmarking datasets

Deep metabarcoding data. The deep metabarcoding data we used for benchmarking originate from the Insect Biome Atlas (IBA) project[6]. The project sampled the terrestrial arthropod faunas of Sweden and Madagascar using Malaise traps, soil samples and litter samples. The samples were metabarcoded using several different protocols, targeting a 418 bp region of CO1, were sequenced on the NovaSeq6000 platform. The raw sequences were preprocessed as described in Miraldo et al. (2024). Briefly, read trimming and filtering was applied using a Snakemake workflow available at https://github.com/insect-biome-atlas/amplicon-multi-cutadapt, followed by filtering to remove sequences containing stop codons in the expected read-frame. Finally, the reads were denoised using the nf-core/ampliseq Nextflow workflow (v2.4.0)[41], which uses the DADA2 algorithm[18] to infer amplicon sequence variants (ASVs) from the preprocessed reads. This workflow also uses the DADA2 removeBimeraDenovo function for chimera removal using the following settings: 'method="consensus", minSampleFraction = 0.9, ignoreNNegatives = 1, minFoldParentOverAbundance = 2, minParentAbundance = 8, allowOneOff = FALSE, minOneOffParentDistance = 4, maxShift = 16'. We used two versions of the data for benchmarking: (1) an early version, restricted to mild lysis of the Malaise trap samples (4,560 samples from Sweden and 1,923 from Madagascar), containing 636,297 and 559,023 ASVs, respectively; and (2) a final version, containing CO1 data from all sample types (mild lysis and homogenates of Malaise trap samples, as well as soil and litter samples) (total of 5,804 and 2,111 samples containing 821,559 and 701,769 ASVs, for Sweden and Madagascar respectively). The data are available from https://doi.org/10.17044/scilifelab.25480681.v1 and https://doi.org/10.17044/scilifelab.25480681.v6.

Taxonomic reference databases based on BOLD. We generated a reference database for benchmarking taxonomic assignment tools from sequences in the BOLD database[23]. Firstly sequence and taxonomic information for records in BOLD were downloaded from the GBIF Hosted Datasets (https://hosted-datasets.gbif.org/ibol/). This data was then filtered to only keep records annotated as ‘COI-5P’ and assigned to a BOLD BIN ID. The taxonomic information, as processed by GBIF, was parsed in order to assign species names and resolve higher level ranks for each BIN ID. Sequences were processed to remove gap characters and leading and trailing Ns. After this, any sequences with remaining non-standard characters were removed. Sequences were then clustered at 100% identity using VSEARCH[20]. This clustering was done separately for sequences assigned to each BIN ID. The processed database is available at the SciLifeLab Figshare repository[42]; we used version 4 of the database (https://doi.org/10.17044/scilifelab.20514192.v4). For the purposes of benchmarking, sequences were trimmed to the 418 bp region matching the barcode region used in this study. We refer to this in the following as the ‘custom BOLD database’.

The above database adopts GBIF’s name resolution algorithm for BINs with several taxonomic name annotations, using an 80% threshold for accepting a name as the consensus annotation. For gauging the performance of clustering tools against the RESL algorithm used to generate BOLD BINs, we also downloaded the complete BOLD database (from 06-Jul-2022). From this database, information on 3,953,373 sequences with both

BOLD BIN and species assignments were extracted and used to compute precision and recall values, treating BIN assignments (the output of RESL) as 'ASV clusters'. No cleaning of taxonomic annotations was applied in this case.

For some benchmarks, we also used a carefully curated and almost complete barcode reference library for the Finnish fauna of arthropods[9], which is a small subset of BOLD (FinBOL). We expected this library to contain less noise than BOLD in general, and good coverage of the authentic CO1 ASVs likely to be encountered in Swedish arthropods. We refer to IBA ASVs with identical matches over the full ASV sequence to FinBOL reference sequences as 'trusted ASVs'.

Chimera removal

We explored postprocessing chimera removal strategies for deep metabarcoding data based on the UCHIME de novo algorithm[13]. Specifically, we applied UCHIME with default settings in either a 'batchwise' or 'samplewise' mode. In the 'batchwise' mode, chimera detection is run directly on the full ASV dataset while in 'samplewise' mode, the ASV dataset is split into sample-specific fasta files containing sequences with a count > 0 in the sample. This means that the candidate parents for each ASV are restricted to the sequences present in the same sample. In addition, we implemented cutoffs on either the fraction of samples in which chimeric ASVs had to be present with their parents (for 'batchwise') or the fraction of samples in which ASVs had to be identified as chimeric (for 'samplewise'), testing both 'lenient' and 'strict' settings (Table 1).

Table 1. Chimera detection settings. Four different strategies were tested. We used either a batchwise or samplewise approach, and then enforced a lenient or strict requirement on the parent-chimera co-occurrence pattern.

Name[1]	Mode	Settings
batchwise.strict	batchwise	ASVs had to share samples with both parents in ≧ 50% of samples in which they were present
batchwise.lenient	batchwise	ASVs had to share samples with both parents in ≧ 1 sample
samplewise.strict	samplewise	ASVs had to be identified as chimeric in all samples where they were present in order to be removed
samplewise.lenient	samplewise	ASVs had to be identified as chimeric in ≧ 1 sample in order to be removed

Notes: [1]Method name used in Fig. 1.

We benchmarked the performance of the chimera detection on the initial version of the Swedish IBA data (636,297 ASVs), using the following evaluation metrics (with corresponding labels in Fig. 1 in parenthesis): (1) total number of ASVs removed ('ASVs removed'); (2) proportion of the removed ASVs that were abundant, that is, found in more than one sample ('% abundant ASVs removed'); (3) proportion of ASVs identical to sequences in FinBOL removed ('% trusted ASVs removed'); (4) total number of reads removed ('Million reads removed'); (5) Number of reads of ASVs identical to FinBOL removed ('Trusted reads removed'); (6) generated clusters to the number of unique species annotations in the cleaned data ('Cluster:species ratio'); (7) number of species split across multiple clusters in the cleaned data ('Multicluster species''); and (8) precision, (9) recall and (10) F-score (harmonic mean of precision and recall) for the clustering ('Precision', 'Recall' and 'F-score', respectively), assessed against the SINTAX assignment of ASVs to species (see below).

Precision and recall were calculated as

$$precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}$$

where TP = true positives, FP = false positives and FN = false negatives. TP was calculated by counting the number of times two ASVs belonging to the same ASV cluster also belonged to the same species, while FN was calculated by counting the number of times two ASVs that belonged to the same species were placed into different ASV clusters. FP was calculated as

$$FP = NP - TP$$

where NP is the total number of within-cluster pairs, that is the total number of possible ASV pairs within each cluster, summed over all clusters. For the evaluation metrics based on clustering, we used OptiClust[29] as implemented in mothur v1.44.11[43] with a cutoff of 0.025.

Taxonomic assignment

We benchmarked the performance of four different tools for taxonomic assignment of ASVs against a reference database: SINTAX[19] implemented in VSEARCH[20], DADA2 assignTaxonomy[18] (an RDP-classifier type of tool), and SKLEARN and VSEARCH implemented in Qiime2 via the 'classify-sklearn' and 'classify-consensus-vsearch' pipelines in the 'feature-classifier' plugin[44]. We also tested a phylogenetic placement tool (EPA-NG combined with GAPPA for taxonomic annotation) with three different settings for the heuristic placement algorithm used (Table 2). The reference database was the custom BOLD database described above. For the phylogenetic placement, we used a phylogenetic tree of 49,331 Hexapoda sequences with representative outgroups from taxa likely to be present in the IBA data. As a core, we used the tree of[38]. From this tree, we removed all sequences with frame shifts or stop codons, as well as all outgroup sequences (they were sprinkled across the tree); this gave us the ingroup tree. We then assembled CO1 data for relevant outgroups, missing Hexapoda classes (Collembola, Protura and Diplura), and a couple of ingroup representatives from GenBank and Bellini et al. (2023)[45], preferably complete CO1 sequences when available. The DNA sequences were converted to amino-acid sequences and aligned using MAFFT[46], and then converted back to a nucleotide alignment using

PAL2NAL[47]. Relationships among the taxa in this dataset were inferred using MrBayes 3.2.7a[48] and a constrained backbone tree representing well established relationships and highly supported groups in the analysis of Bellini et al. (2023)[45]. Finally, the pruned Chesters tree was grafted onto a tree sampled from the posterior estimate from this analysis, replacing the ingroup representatives. All CO1 sequences in the final tree were realigned using MAFFT and PAL2NAL as described above, followed by removal of all sites present in less than 90% of taxa; this resulted in a final alignment of the same length as the original Chesters alignment. The reference tree and all data and scripts used in obtaining it are available from https://github.com/orgs/insect-biome-atlas/paper-bioinformatic-methods.

Table 2. Taxonomic annotation tools and settings. We tested four different tools for annotations based on matching to a reference database (SINTAX, DADA2, SKLEARN and VSEARCH), and one tool (EPA-NG) with three different heuristics for phylogenetic placement in a reference tree.

Name[1]	Tool	Type	Version	Settings (if different from default)	Reference
sintax	SINTAX	kmer	2.21.2 (vsearch)	–sintax_cutoff 0.8 –randseed 15 –threads 1	Edgar (2016), Rognes et al. (2016)
dada2_rdp	DADA2	kmer	1.30.0	minBoot=80	Callahan et al. (2016)
qiime_sklearn	SKLEARN	kmer	2023.9 (QIIME2)		Bolyen et al. (2019)
qiime_vsearch	VSEARCH	alignment	2023.9 (QIIME2)		Bolyen et al. (2019)
epa-ng_baseball	EPA-NG	phylogenetic		--baseball-heur	Barbera et al. (2019); Czech et al. (2020)
epa-ng_dynamic	EPA-NG	phylogenetic			Barbera et al. (2019); Czech et al. (2020)
epa-ng_none	EPA-NG	phylogenetic		--no-heur	Barbera et al. (2019); Czech et al. (2020)

Notes:
[1]Method name used in Fig. 2.

For benchmarking, we generated five combinations of test/train datasets using the custom BOLD CO1 reference database described above. For each combination, 1000 species were randomly sampled and one sequence was selected from each species and added to the test dataset. Specifically, the examined cases (matching the cases in Fig. 2) were (from simple to more challenging):

1. Keep all sequences in the reference database or tree, including the test sequences.
2. Remove all sequences identical to sequences in the test dataset from the reference database, but keep at least one sequence for each test species. For phylogenetic placement algorithms, use Insecta sequences from the reference database for which the species is represented in the tree but the specific sequence is missing.
3. Remove all species in the test dataset from the reference database, but keep at least one sequence for each test genus. For phylogenetic placement algorithms, use Insecta sequences from the reference database for which the species is missing but the corresponding genus is present in the reference tree.
4. Remove all genera in the test dataset from the reference database, but keep at least one sequence for each test family. For phylogenetic placement algorithms, use Insecta sequences from the reference database for which the genus is missing but the corresponding family is present in the reference tree.
5. Remove all families in the test dataset from the reference database, but keep at least one sequence for each test order. For phylogenetic placement algorithms, use Insecta sequences from the reference database for which the family is missing but the corresponding order is present in the reference tree.

For case 5 (family absent) only 490 species (instead of 1000) could be randomly sampled from the phylogenetic reference and placed into the test dataset.

In terms of benchmark metrics, we looked at the number of correctly assigned, incorrectly assigned and unassigned sequences for each of the five test cases. To further elucidate the performance “in the wild”, with various completeness of the reference databases and in the presence of noise among query sequences, we compared the performance of the selected tools in taxonomic annotation of the final IBA datasets for Sweden and Madagascar.

ASV clustering

We benchmarked the performance of three open-source tools for clustering metabarcoding ASVs into OTUs: (1) SWARM v3.1.0[28]; (2) OptiClust[29] as implemented in mothur v1.44.11[43]; and (3) dbOTU3 v1.5.3[31].

For benchmarking, we used the initial IBA dataset for Sweden after removal of chimeras using the strict samplewise method described above. We ran each tool with a range of parameters (Table 3), and evaluated results by calculating precision and recall values given the SINTAX assignment of ASVs to species using the custom BOLD database, as described above.

We also wanted to compare the performance of these open-source tools to the closed-source RESL algorithm used to compute OTU clusters (BINs) for the BOLD database[23]. There are two ways in which RESL clustering performance can be compared to

open-source tools for clustering of metabarcoding data: (1) open-source tools can be applied to the BOLD database, for which the RESL results are known (the BIN clusters); or (2) RESL results for a metabarcoding dataset can be inferred by closed-reference matching to the BOLD database, and interpreting the assignments of BINs to ASVs as the likely result of RESL clustering. Each of these methods has its advantages and disadvantages: The first approach compares the performance on a dataset of mostly authentic full-length CO1 barcode sequences, that is, slightly longer sequences than a deep metabarcoding dataset, but also a database with a sparse and uneven coverage of the diversity of most biomes studied by metabarcoding. Thus, the results may not be relevant for deep metabarcoding data. The second approach potentially yields results that are more appropriate for clustering of metabarcoding data, but the RESL results are only indirectly inferred in ways that can be beneficial for RESL (its clusters are based on longer sequences, helping it interpret the cluster structure of the OTUs) or problematic for RESL (it does not have access to the better coverage of the studied biome represented by the deep metabarcoding data). We opted for using the latter method here (Supplementary Fig. S1).

Table 3. Settings used in benchmarking ASV clustering tools. In all cases, 'eval1' represents default settings, which are shown in full. For other runs, only settings that differ from default values are shown

Name[1]	Swarm	Opticlust	dbOTU3
eval1 (default)	d=1, fastidious=True, boundary=3	cutoff=0.03	dist=0.1, abund=10
eval2	d=3	cutoff=0.01	dist=0.01, abund=10
eval3	d=5	cutoff=0.015	dist=0.1, abund=30
eval4	d=7	cutoff=0.02	dist=0.15, abund=10
eval5	d=9	cutoff=0.025	dist=0.01, abund=0
eval6	d=11	cutoff=0.04	dist=0.1, abund=0
eval7	d=13	cutoff=0.05	dist=0.15, abund=0
eval8	d=15	cutoff=0.07	dist=0.01, abund=20

eval9	d=25	cutoff=0.1	dist=0.1, abund=20
eval10	d=30	cutoff=0.15	dist=0.15, abund=20

Notes:
[1]Method name used in Fig. 3.

Design of the NEEAT algorithm

Each NEEAT filter was coded in a separate R script. The filters were designed as follows.

Echo filter. The echo filter takes the OTU table ('counts') and the pairwise distance values from vsearch ('matchlist') as input, in addition to parameter settings. First, the OTUs are ordered by number of samples (primary criterion) and number of reads (secondary criterion), from largest to smallest. The top OTU is regarded as authentic. Each of the following OTUs is then compared to the *n* closest previously identified authentic clusters ('n_closest', default 10) within the desired distance ('min_match', default 84% identity). For each potential parent cluster, the algorithm checks whether it co-occurs with the potential echo in the required proportion of samples ('min_overlap', default 0.95), and whether the read number criteria are fulfilled. If 'require_corr' is TRUE and the parent and echo co-occur in more than three samples, then a linear model is fit to the read numbers of parent and echo, and it is checked whether the regression is significant (*p* value smaller than 'max_p_val', default 0.05) and the coefficient is smaller than 'max_read_ratio' (default 1.0). In all other cases, it is simply checked whether the read ratio is smaller than 'max_read_ratio'. The read ratio type ('read_ratio_type') can be set to either 'max' or 'mean' of reads across samples where both parent and echo occur.

Evolutionary signature filter ('Evo' filter). This filter compares edit distances (computed with vsearch) to evolutionary distances between sequence pairs. To compute the latter, we first aligned ASV sequences by translating them to amino-acid sequences using the R package APE [49], aligning them with MAFFT[46] using the '--auto' option, and converting them back to nucleotide alignments using PAL2NAL[47]. We then computed two different metrics, 'dadn' and 'wdadn'. The former is simply the ratio between nonsynonymous (amino-acid) and synonymous differences. The latter weighted the amino-acid difference according to a biochemical similarity metric[50].

The evo filter orders OTUs in the same way as the echo filter. The top OTU is regarded as authentic, and the other OTUs are compared to authentic neighbors using 'min_match' and 'n_closest' criteria, as for the echo filter. In the local version of the evo filter ('require_overlap', default 'TRUE'), a minimum sample overlap ('min_overlap', default 0.95) is also required. If the ratio between the desired type of evolutionary distance ('dist_type', default 'wdadn') and the edit distance is larger than the threshold ('dist_threshold', default 1.0), the OTU is flagged as non-authentic.

Abundance filter. This filter simply flags OTUs with read numbers below the desired 'cutoff' as noise. The cutoff type ('cutoff_type') can be set to 'sum', in which case the sum of reads across samples is compared to the threshold, or to 'max', in which case the max reads in any sample is used instead.

Taxonomic annotation filter. This filter simply removes OTUs with uncertain annotation at the desired taxonomic rank ('assignment_rank', default 'Order'), relying on HAPP annotation conventions (see below).

NEEAT. In NEEAT, we apply the filters in sequence, using the retained OTUs from each step as input to the next. NEEAT first applies the echo filter and then the evo filter in local mode, the evo filter in global mode, the abundance filter, and finally the taxonomic annotation filter.

Benchmarking of noise filtering algorithms

To benchmark noise filtering algorithms, we used the Swedish IBA data for orders of Hexapoda. The taxonomic assignment of each OTU was based on the assignment of the representative ASV, that is, the ASV with the highest median read number across samples of all ASVs included in the OTU. To measure false positives (authentic OTUs filtered out), we used a weighted average of the number of unique BOLD BINs and the number of trusted OTUs (the OTUs containing trusted ASVs from FinBOL) removed. To measure false negatives (non-authentic OTUs remaining) we used a weighted average of the number of duplicated BOLD BINs and the number of 'spurious' OTUs remaining. An OTU was considered spurious if it belonged to a well-known family (>99% of species known according to Ronquist et al. 2020) but none of the included ASVs matched a species in the reference library.

We first benchmarked the component filters individually, exploring a range of settings and using LULU[36] and an HMM profile model[35] as references for the echo and evo filter, respectively (Supplementary Figs. 2-4). We then computed the Pareto front of the NEEAT filter by combining individual filters under parameter settings with a low level of false positives (as false positives add up in NEEAT), and then removing those combinations that were dominated in the two performance dimensions by other combinations.

These tests filtered out all OTUs with uncertain annotations at the order level. To assess the performance of the annotation filter, we compared the annotation success for a fauna that is well covered in the reference library and phylogeny (IBA data from Sweden) with one that is not (IBA data from Madagascar). The proportion of noise OTUs should be similar for both faunas; thus, differences in annotation rate should be due to the effect of coverage in reference databases.

Design of the HAPP pipeline

HAPP was designed as a Snakemake workflow. The user supplies two input files to the pipeline: a FASTA file of ASVs and a tab-separated file with counts of each ASV in all samples. The pipeline first pre-filters the ASVs, removing sequences that fall outside a user defined length interval (default: 403 - 418 bp) as well as those with in-frame stop codons. Taxonomy is assigned to the filtered ASVs using a two-step approach combining SINTAX and EPA-NG. Chimera filtering is applied to the filtered ASVs using the uchime_denovo algorithm and a user configurable filtering criteria (default: samplewise). The taxonomic assignments are then used to split the filtered, non-chimeric sequences into taxonomic subsets at a user configurable rank (default: Family). Each subset is then clustered into OTUs in parallel using all clustering software configured by the user (default: SWARM). The OTUs for all subsets are gathered per clustering software and are then split into order-level

subsets. Each of these subsets are then filtered for noise in parallel, taking the abundance of OTUs into account as well as applying the NEEAT algorithm to OTUs.

Additional filtering steps in HAPP. HAPP includes several additional filtering steps not discussed above. First, it allows filtering out of clusters occurring in more than a specified fraction of negative controls, and extra abundance filtering (if the NEEAT filtering is deemed insufficient). Both of these filters are applied per dataset. The user controls the filters by supplying a metadata file specifying which samples are blanks/negative controls as well as which samples belong to what datasets.

HAPP also allows the user to filter out spike-in clusters based on user-provided species annotations or BOLD BIN annotations. As an alternative, an occurrence threshold (clusters of specified higher taxa occurring in more than x% of samples are deemed as spike-in clusters) can be used to identify and remove spike-in clusters.

Handling of ambiguous taxonomic annotations. HAPP uses several approaches to handle and resolve ambiguities in taxonomic annotations. An ASV is labeled as ‘unassigned’ if the taxonomic annotation algorithm failed to assign the ASV to a taxon at this rank. If the ASV is assigned to a taxon named ‘Taxon_name’ at the next higher rank, it is labeled as ‘unassigned.Taxon_name’. If the ASV is unassigned at the next higher rank, the annotation at that level is simply copied to the lower rank. Ambiguities may also arise from the reference database itself. For the reference database we used, the taxon annotation of the sequences in each BOLD BIN was resolved by GBIF, if all sequences did not have the same taxon annotation in BOLD. Specifically, GBIF used an 80% cut-off to call a taxon annotation; BOLD BINs with more conflict among sequence annotations were not resolved to a name at that rank. In the case ASVs match such unresolved annotations in the reference database, HAPP labels the ASVs with the lowest rank that was fully resolved, followed by an underscore and one ‘X’ for each lower rank that was unresolved. For instance, a BIN in the family Hepialidae unresolved at the genus and species levels would be leveled at the species level as ‘Hepialidae_XX’.

Further ambiguities could result in the clustering of ASVs into OTUs. HAPP uses a method similar to the GBIF name resolution mechanism in obtaining a consensus assignment. First, HAPP takes all taxonomic assignments from cleaned ASVs for each cluster, and then iterates from the lowest (most resolved) taxonomic rank to the highest. At each rank, taxonomic assignments for all ASVs in the cluster are gathered and weighted by their total read counts. If the weighted assignment make up 80% or more of the assignments at that rank, that taxonomy is propagated to the ASV cluster, including parent rank assignments. If no taxonomic assignment is above the 80% threshold, the algorithm continues to the parent rank. Taxonomic assignments at any available child ranks are set to the consensus assignment and prefixed with ‘unresolved.’.

To illustrate the taxonomic annotation resolution algorithm in HAPP, consider the following example of an ASV cluster with five ASVs:

ASV_ID	Family	Genus	Species	Reads
asv1	Family1	Genus1	Species1	10

asv2	Family1	Genus1	Species1	10
asv3	Family1	Genus2	Species2	50
asv4	Family1	Genus2	Species2	10
asv5	Family1	Genus2	Species3	20

At rank=Species the percentage of assignments would be Species1=20% ((10+10)/100), Species2=60% ((50+10)/100), Species3=20% (20/100) meaning no assignment reached above the 80% threshold. Moving up to rank=Genus the values would be Genus1=20%, Genus2=80% which would assign Genus2 as the consensus. The ASV cluster would then be given the taxonomic assignment 'Family1, Genus2, unresolved.Genus2'.

HAPP also computes the 'representative' ASV of each cluster as the ASV with the highest mean read number across samples of all ASVs belonging to the cluster. As an alternative to the consensus algorithm described above, it is also possible to use the assignment of the representative ASV for the entire OTU.

Acknowledgements

This work was supported by the SciLifeLab & Wallenberg Data Driven Life Science Program, Knut and Alice Wallenberg Foundation (grants: KAW 2020.0239 and KAW 2017.0003), and by the National Bioinformatics Infrastructure Sweden (NBIS) at SciLifeLab. EG acknowledges funding from the Finance to Revive Biodiversity (FinBio) program, financed by Mistra – the Swedish Foundation for Strategic Environmental Research (DIA 2020/10). AFA is supported by a grant from the Swedish Research Council VR (2021-05563).

References

1. Taberlet, P., Coissac, E., Pompanon, F., Brochmann, C. & Willerslev, E. Towards next-generation biodiversity assessment using DNA metabarcoding. *Mol. Ecol.* **21**, 2045–2050 (2012).
2. Wang, W. Y., Srivathsan, A., Foo, M., Yamane, S. K. & Meier, R. Sorting specimen-rich invertebrate samples with cost-effective NGS barcodes: Validating a reverse workflow for specimen processing. *Mol. Ecol. Resour.* **18**, 490–501 (2018).
3. Creedy, T. J. *et al.* A validated workflow for rapid taxonomic assignment and monitoring of a national fauna of bees (Apiformes) using high throughput DNA barcoding. *Mol. Ecol. Resour.* **20**, 40–53 (2020).
4. Elbrecht, V., Vamos, E. E., Steinke, D. & Leese, F. Estimating intraspecific genetic diversity from community DNA metabarcoding data. *PeerJ* **6**, e4644 (2018).

5. Lopez, J. V., Yuhki, N., Masuda, R., Modi, W. & O'Brien, S. J. Numt, a recent transfer and tandem amplification of mitochondrial DNA to the nuclear genome of the domestic cat. *Journal of Molecular Evolution* **39**, 174–190 (1994).
6. Miraldo, A. *et al.* Data of the Insect Biome Atlas: a metabarcoding survey of the terrestrial arthropods of Sweden and Madagascar. *bioRxiv* (2024) doi:10.1101/2024.10.24.619818.
7. Iwaszkiewicz-Eggebrecht, E. *et al.* High-throughput biodiversity surveying sheds new light on the brightest of insect taxa. *bioRxiv* (2024) doi:10.1101/2024.10.25.620209.
8. Ronquist, F. *et al.* Completing Linnaeus's inventory of the Swedish insect fauna: Only 5,000 species left? *PLoS One* **15**, e0228561 (2020).
9. Roslin, T. *et al.* A molecular-based identification resource for the arthropods of Finland. *Mol Ecol Resour* **22**, 803–822 (2022).
10. Nilsson, R. H. *et al.* An open source chimera checker for the fungal ITS region. *Mol. Ecol. Resour.* **10**, 1076–1081 (2010).
11. Haas, B. J. *et al.* Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Res.* **21**, 494–504 (2011).
12. Quince, C., Lanzen, A., Davenport, R. J. & Turnbaugh, P. J. Removing noise from pyrosequenced amplicons. *BMC Bioinformatics* **12**, 38 (2011).
13. Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C. & Knight, R. UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics* **27**, 2194–2200 (2011).
14. Berger, S. A., Krompass, D. & Stamatakis, A. Performance, accuracy, and Web server for evolutionary placement of short sequence reads under maximum likelihood. *Syst Biol* **60**, 291–302 (2011).
15. Barbera, P. *et al.* EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences. *Syst Biol* **68**, 365–369 (2019).
16. Czech, L., Barbera, P. & Stamatakis, A. Genesis and Gappa: processing, analyzing and visualizing phylogenetic (placement) data. *Bioinformatics* **36**, 3263–3265 (2020).
17. Wang, Q., Garrity, G. M., Tiedje, J. M. & Cole, J. R. Naive Bayesian classifier for rapid

assignment of rRNA sequences into the new bacterial taxonomy. *Appl Environ Microbiol* **73**, 5261–5267 (2007).

18. Callahan, B. J. *et al.* DADA2: High-resolution sample inference from Illumina amplicon data. *Nat. Methods* **13**, 581–583 (2016).
19. Edgar, R. C. SINTAX: a simple non-Bayesian taxonomy classifier for 16S and ITS sequences. *bioRxiv* (2016) doi:10.1101/074161.
20. Rognes, T., Flouri, T., Nichols, B., Quince, C. & Mahé, F. VSEARCH: a versatile open source tool for metagenomics. *PeerJ* **4**, e2584 (2016).
21. Bolyen, E. *et al.* Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nat Biotechnol* **37**, 852–857 (2019).
22. Ratnasingham, S. & Hebert, P. D. N. bold: The Barcode of Life Data System (http://www.barcodinglife.org). *Mol Ecol Notes* **7**, 355–364 (2007).
23. Ratnasingham, S. & Hebert, P. D. N. A DNA-based registry for all animal species: the barcode index number (BIN) system. *PLoS One* **8**, e66213 (2013).
24. Van Dongen, S. Graph clustering via a discrete uncoupling process. *SIAM J. Matrix Anal. Appl.* **30**, 121–141 (2008).
25. Schloss, P. D. & Westcott, S. L. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Appl. Environ. Microbiol.* **77**, 3219–3226 (2011).
26. Navas-Molina, J. A. *et al.* Advancing our understanding of the human microbiome using QIIME. *Methods Enzymol.* **531**, 371–444 (2013).
27. Mahé, F., Rognes, T., Quince, C., de Vargas, C. & Dunthorn, M. Swarm: robust and fast clustering method for amplicon-based studies. *PeerJ* **2**, e593 (2014).
28. Mahé, F. *et al.* Swarm v3: towards tera-scale amplicon clustering. *Bioinformatics* **38**, 267–269 (2021).
29. Westcott, S. L. & Schloss, P. D. OptiClust, an Improved Method for Assigning Amplicon-Based Sequence Data to Operational Taxonomic Units. *mSphere* **2**, (2017).
30. Preheim, S. P., Perrotta, A. R., Martin-Platero, A. M., Gupta, A. & Alm, E. J.

Distribution-based clustering: using ecology to refine the operational taxonomic unit. *Appl Environ Microbiol* **79**, 6593–6603 (2013).

31. Olesen, S. W., Duvallet, C. & Alm, E. J. dbOTU3: A new implementation of distribution-based OTU calling. *PLoS One* **12**, e0176335 (2017).
32. Liu, X. *et al.* Genomic and transcriptomic perspectives on the origin and evolution of NUMTs in Orthoptera. *Mol Phylogenet Evol* **201**, 108221 (2024).
33. Bogenhagen, D. F. Mitochondrial DNA nucleoid structure. *Biochim Biophys Acta* **1819**, 914–920 (2012).
34. Quiros, P. M., Goyal, A., Jha, P. & Auwerx, J. Analysis of mtDNA/nDNA Ratio in Mice. *Curr Protoc Mouse Biol* **7**, 47–54 (2017).
35. Porter, T. M. & Hajibabaei, M. Profile hidden Markov model sequence analysis can help remove putative pseudogenes from DNA barcoding and metabarcoding datasets. *BMC Bioinformatics* **22**, 256 (2021).
36. Frøslev, T. G. *et al.* Algorithm for post-clustering curation of DNA amplicon data yields reliable biodiversity estimates. *Nat Commun* **8**, 1188 (2017).
37. Andújar, C. *et al.* Validated removal of nuclear pseudogenes and sequencing artefacts from mitochondrial metabarcode data. *Mol Ecol Resour* **21**, 1772–1787 (2021).
38. Chesters, D. Construction of a Species-Level Tree of Life for the Insects and Utility in Taxonomic Profiling. *Syst Biol* **66**, 426–439 (2017).
39. Mölder, F. *et al.* Sustainable data analysis with Snakemake. *F1000Res* **10**, 33 (2021).
40. Buchner, D. *et al.* Upscaling biodiversity monitoring: Metabarcoding estimates 31,846 insect species from Malaise traps across Germany. *Mol. Ecol. Resour.* **25**, e14023 (2025).
41. Straub, D. *et al.* Interpretations of Environmental Microbial Community Studies Are Biased by the Selected 16S rRNA (Gene) Amplicon Sequencing Pipeline. *Front Microbiol* **11**, 550420 (2020).
42. Sundh, J. COI reference sequences from BOLD DB. http://dx.doi.org/10.17044/scilifelab.20514192 (2022) doi:10.17044/scilifelab.20514192.

43. Schloss, P. D. *et al.* Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* **75**, 7537–7541 (2009).
44. Bokulich, N. A. *et al.* Optimizing taxonomic classification of marker-gene amplicon sequences with QIIME 2's q2-feature-classifier plugin. *Microbiome* **6**, 90 (2018).
45. Bellini, B. C. *et al.* The evolution of Collembola higher taxa (Arthropoda, Hexapoda) based on mitogenome data. *Diversity (Basel)* **15**, 7 (2022).
46. Katoh, K. & Standley, D. M. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.* **30**, 772–780 (2013).
47. Suyama, M., Torrents, D. & Bork, P. PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. *Nucleic Acids Res* **34**, W609–12 (2006).
48. Ronquist, F. *et al.* MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Syst Biol* **61**, 539–542 (2012).
49. Paradis, E. & Schliep, K. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* **35**, 526–528 (2019).
50. Grantham, R. Amino acid difference formula to help explain protein evolution. *Science* **185**, 862–864 (1974).