Supplementary online methods

1 Simulation

1.1 Inferring the reference tree topology

We used SATIVA (`epa_trainer.py` script) to infer the constrained ML tree on the full LTP123 alignment (11939 taxa). Internally, SATIVA builds a multifurcating constraint tree from the taxonomic annotations of the LTP123 sequences. Then, it calls RaxML (bundled version) as follows:

```
$ raxmlHPC-PTHREADS-AVX -T 16 -s ltp123_full.phy -g tax_constraint.tre
-m GTRCAT -n reftree --no-seq-check --no-bfgs -p 25204
```

to resolve the multifurcations and thus obtain the fully bifurcating tree which is congruent with the taxonomy.

1.2 Partitioning

It is known that 16S rRNA gene contains both highly variable and conserved regions, which correspond to loops and stems in the secondary structure of rRNA molecule. This results in heterogeneity of empirical 16S alignments, which comprise interleaving regions of high and low similarity. To account for this fact in our simulation, we partitioned the original LTP alignment according to 16S rRNA secondary structure, and estimated all model parameters for each partition individually. It allowed to build a simulated alignment which is structurally similar to the empirical LTP alignment.

Bacterial 16S rRNA contains nine variable regions (v1-v9) with the following *E.coli*-based coordinates :

```
v1=69-99
v2=137-242
v3=433-497
v4=576-682
v5=822-879
v6=986-1043
v7=1117-1173
v8=1243-1294
v9=1435-1465
```

We used a custom script (`trans_vx_borders.py`) to translate the above *E.coli* positions into LTP alignment positions, thereby obtaining the coordinates of variable regions in the alignment. We created a partition file by assigning each variable region (v1-v9) to an individual partition, whereas all conserved regions were merged into single *cons* partition. Furthermore, we added the *flanks* partition for 'flanking' alignment regions which have no correspondence in the reference *E.coli* sequence. The resulting partition file (`ltp123_part.txt`) is as follows:

```
DNA, cons=308-489,840-1044,1893-3055,3226-3370,3650-4005,4149-4404,4711-4874,5162-5287,5411-5751
DNA, flanks=1-307,6002-6880
DNA, v1=490-839
DNA, v2=1045-1892
DNA, v3=3056-3225
DNA, v4=3371-3649
DNA, v5=4006-4148
DNA, v6=4405-4710
DNA, v7=4875-5161
DNA, v8=5288-5410
DNA, v9=5752-6001
```

1.3 Estimating branch lengths and model parameters

Next, we used the partition file obtained above as well as the reference tree topology from step 1.1 to optimize the branch lengths and model parameters *for each partition individually*. For this, we ran RAxML v8.2.3 as follows:

```
$ raxmlHPC-PTHREADS-AVX -f e -M -T 16 -s ltp123_full.phy -t reftree.nw -q ltp123_part.txt -m GTRGAMMA -n eval
```

1.4 Generating the simulated alignment

First, we extracted *empirical* subalignments for individual partitions using the following command:

```
$ raxmlHPC-PTHREADS-AVX -f s -T 16 -s ltp123_full.phy -q ltp123_part.txt -m GTRGAMMA -n split
```

Then, we executed INDELible v1.03 to generate *simulated* subalignments for each partition.

Following simulation parameters were set to the values obtained previously in step 1.3:

- "true" tree (`[TREE]`)
- GTR substitution rates (`[submodel]`)
- Base frequencies (`[statefreq]`)
- Alpha parameter of GAMMA distribution (`[rates]`)

Furthermore, we manually tuned the insertion/deletion rates (`[indelmodel]`/`[indelrate]`) such that both *width* and *% gaps* in each *simulated* subalignment matches those values in the corresponding *empirical* subalignment.

Finally, we used a custom script (`merge.py`) to merge all simulated subalignments into a single PHYLIP alignment file.

1.5 Generating the simulated taxonomies

Because of the way we generated the simulated alignment (see above), it ought to be absolutely consistent with the original LTP taxonomy. In other words, we assume LTP taxonomy to be mislabels-free on the simulated alignment. Therefore, we can deliberately introduce taxonomic mislabels by changing the original sequence annotations at random. We used this approach to generate six test taxonomies: three replicates with 1% mislabels and three replicates with 5% mislabels. The distribution of mislabels among the taxonomic rank levels was as follows:

```
Phylum  5%
Class  10%
Order  15%
Family 35%
Genus  35%
```

For instance, given that the overall number of sequences is 11939 and the mislabel rate of 5%, *approximately* 597 sequences will be mislabeled. From those, $\lfloor 597 * 0.05 \rfloor = 29$ will have an incorrect Phylum, $\lfloor 597 * 0.05 \rfloor + \lfloor 597 * 0.10 \rfloor = 88$ – an incorrect Class, $\lfloor 597 * 0.05 \rfloor + \lfloor 597 * 0.10 \rfloor + \lfloor 597 * 0.15 \rfloor = 177$ – an incorrect Order, and so on. This distribution approximately represents the proportion of mislabels at each taxonomic level that we identified in the empirical LTP123 dataset.

Finally, we constructed six test datasets by combining the aforementioned six taxonomies with the simulated alignment.

1.6 Running mislabels identification algorithms

- SATIVA was executed as follows:

  ```
  $ sativa.py -t sim<T>.tax -s sim_full.phy -x BAC -n sativa<T> -m
  thorough -N 1 -v -T 16
  ```

  where <T> = (1..6) is the number of test dataset (input/output paths and other non-essential details were trimmed from the command line).

- UCLUST was executed via a custom script (`mis_tests.py`) which implements a leave-

one-test with UCLUST. More specifically, for each of the N sequences in the dataset, it generates the `QUERY.FA` file containing this sequence, the `REF.FA` file containing the remaining N-1 sequences as well as the `REF.TAX` file containing the taxonomic annotations for the sequences in `REF.FA`. Then, it calls QIIME v1.8 taxonomy assignment script as follows:

```
$ assign_taxonomy.py –i QUERY.FA –r REF.FA –t REF.TAX –m uclust
```

Finally, the script compares the suggested taxonomic annotation to the original annotation of the query sequence. In case of disagreement, the query sequence is considered to be mislabeled; the corresponding mislabel record is printed to the output file in a SATIVA-like format. Mislabel confidence value was set equal to the assignment confidence provided by UCLUST.

- RDP Classifier was executed via a similar custom script (`mis_tests2.py`) . This script, however, allows to specify a range of sequences to test. This became necessary because acceptable running time could be only achieved by parallelization over several cluster nodes. The QIIME taxonomy assignment script was called as follows:

```
$ assign_taxonomy.py –i QUERY.FA –r REF.FA –t REF.TAX –m rdp
--rdp_max_memory 7500
```

As above, RDP assignment confidence was used as mislabel confidence, where applicable.

1.7 Evaluating the results

In this step, custom scripts (`calc_stats_mis.py` & `calc_stats.py`) were used to compare mislabels identified by each method to the 'ground truth', i.e., to the list of deliberately mislabeled sequences for each particular dataset (see Section 1.5). Only the mislabels with the confidence above the respective method-specific threshold were considered (see main text).

Finally, `summarize_stats.py` script was used to average over all replicates and generate the final statistics for each accuracy metric (identification/correction), mislabel level (1% / 5%) and method (SATIVA / UCLUST / RDP). These final statistics are shown in Table 3 in the main text.

2 Real-world datasets

2.1 LTP123_T

We downloaded an ARB database file for LTP release 123 (September 2015) from

http://www.arb-silva.de/fileadmin/silva_databases/living_tree/LTP_release_123/LTPs123_SSU.arb

Then, we used ARB software to export both alignment and taxonomic annotations for all 11939 sequences included in LTP123.

To avoid spurious mislabels, we consistently removed the *"Unclassified"* prefix in taxonomic rank names (e.g., *"Unclassified Alphaproteobacteria"* → *"Alphaproteobacteria"*).

2.2 SLV123_T

This dataset includes only type strains sequences from SILVA release 123 (more specifically, the same sequence set as in LTP123_T). Since the aforementioned ARB database for LTP123 provides alternative taxonomic classifications for each sequence it contains, we used ARB export functionality to obtain SILVA classification from this database (stored in “tax_slv” field).

2.3 GG13_T

As above, we exported Greengenes (version 13.8) taxonomic classification for type strain sequences from the LTP123 database (“tax_greengenes” field).

Sequences labeled as “Unclassified” were excluded, which resulted in a dataset comprising 10635 sequences.

2.4 RDP11_T

Although LTP123 database provides RDP classification, it is not in the most recent version (release 10). Therefore, we extracted taxonomic annotations anew from the files provided by RDP.

We downloaded the RDP release 11.4 (May 2015) from

http://rdp.cme.msu.edu/misc/resources.jsp#aligns

Then, we used custom scripts (available at https://github.com/amkozlov/mislabels16-data) to extract taxonomic annotations from the FASTA files and save them in the text format as required by SATIVA.

RDP11_T dataset comprises 11868 sequences (71 LTP123 sequences are missing in RDP).

2.5 SLV123_NR99

SILVA SSU Ref NR database (release 123, July 2015) was downloaded from:

http://www.arb-silva.de/fileadmin/silva_databases/release_123/ARB_files/SSURef_NR99_123_SILVA_12_07_15_opt.arb.tgz

We used ARB software to export both alignment and taxonomic annotations for 536224 prokaryotic SSU rRNA sequences (*Bacteria* and *Archaea* clades).

2.6 NR13_NR99

Taxonomic annotations of 99% identity OTUs and MSA of the representative sequence set were downloaded from

ftp://greengenes.microbio.me/greengenes_release/gg_13_8_otus/

Placeholder ranks ('s__', 'g__' etc.) were trimmed.

All 203452 sequences were included into analysis.

2.7 SATIVA results post-processing

Before computing mislabels statistics for real-world datasets, we manually checked SATIVA outputs and corrected following errors:

- Re-classifications to/from a “placeholder” taxon, e.g.,

  “Gammaproteobacteria Incertae Sedis” → “Chromatiales”

  were discarded. Following taxon names were considered placeholders: “*uncultured **”, “*Unclassified **”, “*Unknown **”, “**Incertae Sedis*” (unless numbered, s. below). Rationale: placeholders represent 'empty' ranks (lack of classification at this level). According to the SATIVA algorithm, re-classifications to/from an 'empty' rank are not considered as mislabels (see main text).

- Re-classifications to an obviously synonymous taxon, e.g.,

  “Clostridiales; Clostridiales_Incertae_Sedis_XI” → “Clostridiales;Incertae_Sedis_XI”

  were discarded.

- Rank levels which were incorrectly identified by SATIVA or missing from the standard 7-level taxonomy (e.g. Suborder) were corrected.

Please note, that Supplementary File 1 contains the *full* list of mislabels as identified by SATIVA. The aforementioned manual changes are marked in yellow.

3 Hardware configuration details

Hardware configuration and running time for each dataset are summarized in the Table S1.

Table S1.

Dataset	Pipeline step	Hardware configuration	Running time		RAM usage (peak, per node)
			Wall time (hh:mm:ss)	CPU time (hours)	
cyanoEMBL	ALL	Intel Xeon E5-2650 v2 2.6 GHz 8 cores 64 GB RAM	00:59:14	8	< 350 MB
cyanoRDP			01:23:32	11	
cyanoSILVA			01:01:13	8	
cyanoCTU			00:54:22	7	
cyanoGG			00:59:56	8	
GG13_T	ALL	2x Intel Xeon E5-2697 v3 2.6 Ghz 28 cores 64 GB RAM	34:15:27	959	~ 10,000 MB
RDP11_T			41:27:26	1,161	
LTP123_T			47:04:28	1,318	
SLV123_T			44:08:53	1,236	

Dataset	Pipeline step	Hardware configuration	Running time		RAM usage (peak, per node)
			Wall time (hh:mm:ss)	CPU time (hours)	
GG13_NR99	Reference tree inference	Intel Xeon E5-2650 v2 2.6 GHz 16 cores 128 GB RAM	93:41:52	1,499	~ 18,000 MB
	Leave-one-out test	**24 nodes:** 2x Intel Xeon E5-2697 v3 2.6 Ghz 28 cores 64 GB RAM	07:20:08	4,929	~ 13,000 MB
	Final EPA test	4x Intel Xeon E5-4620 2.2 GHz 32 cores 512 GB RAM	01:44:55	56	~ 60,000 MB
		Total:	102:46:55	6,485	
SLV123_NR99	Reference tree inference	4x Intel Xeon E5-4620 v2 2.60 GHz 32 cores 1024 GB RAM	264:25:00	8,461	~ 200,000 MB[1]
	Leave-one-out test	**104 nodes:** 4x Intel Xeon E7-4870 2.4 GHz 40 cores 256 GB RAM	~ 156:00:00[2]	616,728	~ 100,000 MB[1]
	Final EPA test	**4 nodes:** 4x Intel Xeon E5-4620 2.2 GHz 32 cores 512 GB RAM	~ 54:00:00[2]	6,929	~ 350,000 MB[1]
		Total:	~ 475:00:00	632,118	

Notes:

[1]With “memory saving” option enabled ('-S' command line option in SATIVA which correspond to the '-U' option in RAxML).

[2]Estimated wall time. These analysis steps were divided into subtasks and executed as multiple job submissions. Thus the actual parallelism level was varying as it depended on the job scheduler. It makes direct runtime measurement problematic. Therefore, we estimated the corresponding runtimes based on the average number of nodes allocated at once (specified in the 3rd column).

4 An example of annotation error propagation

Here, we present an example of what we think was a case of error propagation. In SILVA release 119, a new sequence with the NCBI accession KF053060 was added to the database. It was initially annotated as *Alcaligenes faecalis (*order *Burkholderiales*, phylum *Proteobacteria)* in NCBI GenBank, and SILVA adopted this classification as well. Later on, sequence record in GenBank was amended. First, organism name was changed to *Bacillus sp.*, and taxonomic path was adjusted accordingly (order *Bacillales*, phylum *Firmicutes*). Then, the sequence itself was also corrected, and new revision obtained accession number KF053060.2 (revision history: https://www.ncbi.nlm.nih.gov/nuccore/KF053060.2?report=girevhist*)*.

In the subsequent release of SILVA (r123), sequence data for this accession was updated to the latest version (KF053060.2), but the original taxonomic path was preserved. Furthermore, seven sequences highly similar to KF053060 were added, and they were also classified as members of *Alcaligenes* genus (see Table S2). This was in disagreement with GenBank annotation, where they were classified as *Bacillus*. Presumably, the erroneous taxonomy was derived from an existing misclassified sequence (KF053060), which was highly similar to the new sequences.

SATIVA suggested to re-classify all eight aforementioned sequences into *Bacillus* genus. This is consistent with the current taxonomic annotation in NCBI GenBank and RDP-II.

Table S2.

NCBI accession	NCBI organism name[1]	NCBI taxonomy[1]	SILVA taxonomy[2]
Added in SILVA 119			
KF053060	Bacillus sp. BAB-2669	Bacteria; Firmicutes; Bacilli; Bacillales; Bacillaceae; Bacillus	Bacteria; Proteobacteria; Betaproteobacteria; Burkholderiales; Alcaligenaceae; Alcaligenes
Added in SILVA 123			
JX093131	uncultured Bacillus sp.	Bacteria; Firmicutes; Bacilli; Bacillales; Bacillaceae; Bacillus	Bacteria; Proteobacteria; Betaproteobacteria; Burkholderiales; Alcaligenaceae; Alcaligenes
JX093151	uncultured Bacillus sp.		
KC442332	Bacillus sp. CH6		
KF721697	uncultured Bacillus sp.		
KF722496	uncultured Bacillus sp.		
KF740382	Bacillus sp. ZYJ-39		
GAXY01000345[3]	Leptopilina clavipes	(Eukaryota)[3]	

Notes:

[1]As of April 11th, 2016.

[2]As of SILVA release 119/123.

[3] This sequence most probably represents a bacterial contamination of the original insect sample, and as of now it was removed from GenBank.

5 Data availability

Scripts and datasets that were used to produce the results described in this paper are available at https://github.com/amkozlov/mislabels16-data. In particular, this repository includes:

- Simulated alignment and *Indelible* control files used to generate it
- Scripts used to run UCLUST, RDP Classifier (both via QIIME) and SATIVA on simulated test datasets
- Scripts used to evaluate UCLUST/RDP/SATIVA results and to compute accuracy statistics
- Taxonomic classification files for real-world datasets and scripts used to obtain them
- SATIVA results for real-world datasets (mislabel lists)

6 Supplementary Figure Legend

Figure S1. A dendrogram showing all genera of the novel Cyanobacteria taxonomic framework (CyanoCTU), superimposed with order-level taxa groups from four existing taxonomies. The tree was obtained as described in the Materials and methods section of the main text. Ten Actinobacteria type strain sequences serve as the out-group. Bootstrap values are shown only when they are above 50%. The bar indicates 0.1 substitutions per nucleotide position. Sub=Subsection, Fam=Family, Chr=Chroococcales, Nos=Nostocales, Osc=Oscillatoriales, Ple=Pleurocapsales, Pro=Prochlorales, Pse=Pseudanabaenales, Sti=Stigonematales, Syn=Synechococcales.