CoCoPyE: feature engineering for learning and prediction of genome quality indices

Niklas Birth* Nicolina Leppich* Julia Schirmacher Nina Andreae
Rasmus Steinkamp Matthias Blanke Peter Meinicke†

Institute of Microbiology and Genetics, University of Goettingen, Germany

Abstract

The exploration of the microbial world has been greatly advanced by the reconstruction of genomes from metagenomic sequence data. However, the rapidly increasing number of metagenome-assembled genomes has also resulted in a wide variation in data quality. It is therefore essential to quantify the achieved completeness and possible contamination of a reconstructed genome before it is used in subsequent analyses. The classical approach for the estimation of quality indices solely relies on a relatively small number of universal single copy genes. Recent tools try to extend the genomic coverage of estimates for an increased accuracy. CoCoPyE is a fast tool based on a novel two-stage feature extraction and transformation scheme. First it identifies genomic markers and then refines the marker-based estimates with a machine learning approach. In our simulation studies, CoCoPyE showed a more accurate prediction of quality indices than the existing tools.
Keywords: metagenomics, quality assessment, machine learning

1 Introduction

The number of genomes assembled from metagenome sequencing projects increases rapidly. However, the variation in the quality of the reconstructed genomes is considerable [1]. For all subsequent analyses that make use of these data, information about the completeness and purity of these genomes is essential. Therefore, an important aspect is the quality assessment of the assembled genomes in terms of completeness and contamination estimates: how much of the original genome is actually present and how much of the genomic material possibly stems from other organisms? It has been suggested that a high-quality genome assembly should be complete by more than 90 percent with a contamination below 5 percent [2]. According to the MIMAG standard high-quality genomes also must contain tRNA and rRNA (SSU/LSU) genes.

The standard for measuring the quality indices of bacterial and archaeal genomes is to count the occurrences of universal single copy marker genes in the query genomes. The evident problem of this classical approach is that these marker genes only represent a small part of the entire genome. Universal single copy gene (SCG) sets for archaea and bacteria typically contain less than 50 genes which can only cover a few percent of an average size microbial genome [3, 4]. As a consequence, estimation of genome quality indices can become highly unreliable. In particular, it is difficult if not impossible to distinguish between indicators for completeness and contamination. In fact, it is easy to mix up two different genomes to equal proportions in such a way that a high completeness with almost no contamination may be predicted as long as the few marker genes from the two organisms complement each other.

CheckM [1] tries to overcome these problems by using lineage-specific SCG sets. If a candidate genome can be assigned to one of the clades represented in CheckM, the tool uses a specific SCG set which may comprise more than a thousand genes, depending on the similarity and the number of reference genomes in the assigned clade. The assignment in CheckM is based on phylogenetic placement of universal SCGs. This considerably improves the completeness and contamination estimates in cases where closely related genomes are present in the CheckM reference tree. If an assignment to a more specific clade is not possible, CheckM automatically uses a universal SCG set for prediction. Although the identification of

*Authors equally contributed to this work.
†Corresponding Author (peter@gobics.de)

lineage-specific SCG sets can substantially enlarge the statistical basis for estimation of quality indices, the corresponding procedure to establish and maintain the SCG sets for all nodes of the reference tree is rather complex [1]. BUSCO [5] offers a similar approach that is also based on prior identification of lineage-specific markers. In contrast to these marker-based approaches, also novel methods have been introduced that utilize machine learning [6, 7, 8] to assess the quality of metagenome assembled genomes. In principle, these methods may also work for candidate genomes for which no closely related reference genomes exist in current data bases. One of these tools is the successor to CheckM, CheckM2 [6], which predicts genome quality indices directly from a high-dimensional genomic feature space.

With CoCoPyE we have developed a hybrid approach for the estimation of genome quality indices which combines the concept of marker genes with a machine learning approach. The corresponding tool is available for offline installation under the main operating systems and can be accessed online by means of the CoCoPyE web server.

2 Methods

2.1 Hybrid approach to quality assessment

Our quality assessment approach is based on the protein domain profile of a query genome. This profile contains the genome-specific frequencies of all protein domain families according to the Pfam database [9]. Based on these counts, the prediction of completeness and contamination is achieved in two stages: in stage I, suitable genomes for comparative analysis are identified in a reference database by profile similarity search. The dynamically extracted marker domains of these reference genomes serve as a basis to establish a first estimate of quality indices. If either the predicted completeness is below 60% or the predicted contamination is above 30% the estimates are directly reported as final predictions. Otherwise, the estimates are further refined in stage II. Thereby, the high-dimensional profile space is transformed to a low-dimensional feature space based on all Pfam count ratios between query genomes and reference profiles without restriction to a specific set of marker domains. The corresponding features are finally used for a machine learning-based prediction. An important difference between the two stages is that stage I can work with a wide range of genome quality because it is not trained for a particular range. In contrast, stage II is subject to specific training with data from a defined quality range and therefore requires stage I for filtering and prediction in cases where the stage I estimates indicate that the stage II quality range is not met. An overview of the complete prediction engine is shown in Figure 1.

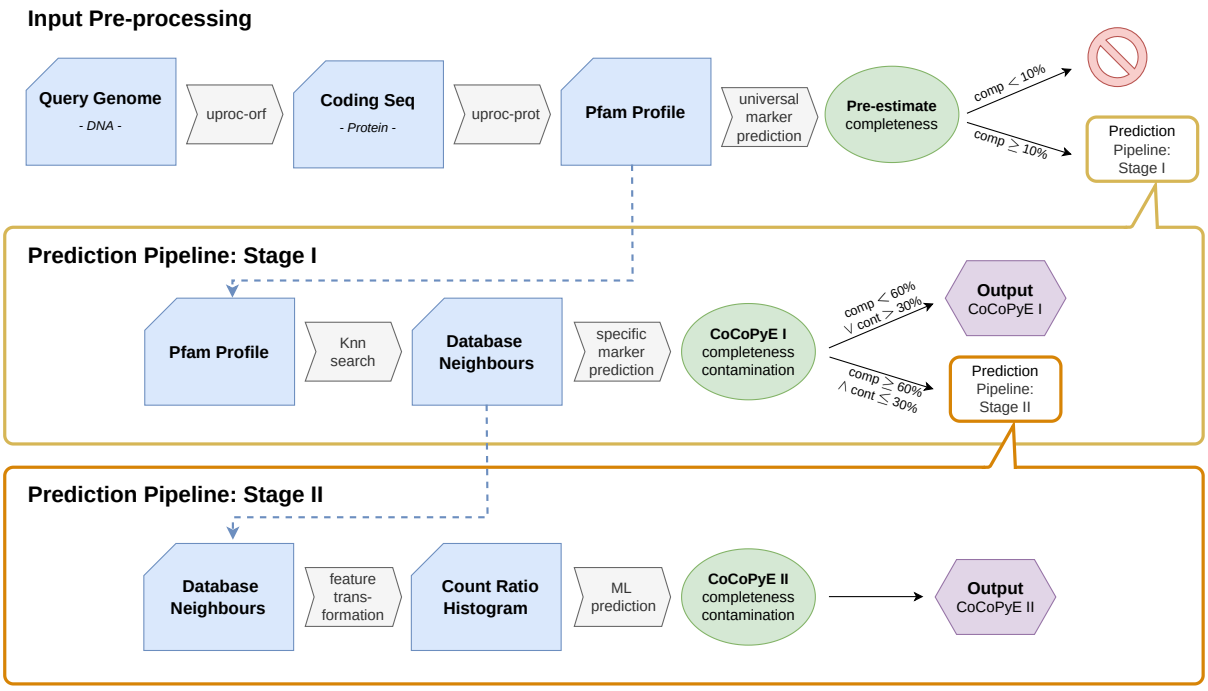

Figure 1: Schematic overview of the prediction pipeline.

2.1.1 Protein domain features

The analysis of a query genome starts with a protein domain search with UProC [10] using default parameter values and counting the occurrences of protein sequence families within potential coding regions as obtained from all translated open reading frames with a minimum length of 20 amino acids found in the genomic sequences. This results in a high-dimensional profile of protein domain counts. As a protein database we offer two pre-processed versions of Pfam [9] which in case of versions 24 and 28 result in 11 912 and 16 230 features, respectively.

2.1.2 Pre-filtering

A central step in our method is the search for nearest neighbours of a query genome in the reference database. This step becomes unreliable if the completeness of the input is too low, i.e., if the input lacks a sufficient number of potential protein domain markers. For this reason we require an initial completeness estimate as obtained from two superkingdom-specific marker sets, according to all bacterial and archaeal reference genomes. For each set we applied a 95% coverage criterion on single copy domains to define the initial markers. If the completeness estimates with regard to both marker sets are below 10%, we reject the query.

2.1.3 Nearest neighbour search

Otherwise, the high-dimensional profile vector of a query genome is compared with the pre-computed profiles in a reference genome database. For this, similar reference profiles are identified by K-nearest neighbour search: for protein family indices i, j, k and protein family counts $C_q^{(i)}$ (query) and $C_r^{(i)}$ (reference), the similarity measure

$$\mathrm{sim}\left(\vec{C}_q, \vec{C}_r\right) = \frac{|\{i \mid (C_q^{(i)} = C_r^{(i)}) \wedge (C_q^{(i)} > 0)\}|}{\sqrt{|\{j \mid C_r^{(j)} > 0\}| \cdot |\{k \mid C_q^{(k)} > 0\}|}} \tag{1}$$

counts the number of coinciding non-zero counts in corresponding profile entries. The required equality implies that mainly small counts contribute to the similarity estimate. In prior studies we found that larger counts usually introduce too much variation and unfavourably increase the impact of possible contaminants on the similarity measure.

2.1.4 Marker-based estimate with nearest reference profiles

From the K nearest neighbours according to the above profile similarity measure we compute an initial marker-based estimate of the completeness and contamination indices. We do not use pre-defined static marker sets, but instead obtain a set of specific markers $\mathbb{M}_q$ for each query that arises from feature dimensions with equal non-zero counts in all K nearest neighbours. With index set $\mathbb{I}_K$ containing all reference indices of the neighbours we obtain the marker set

$$\mathbb{M} = \{m \mid \forall i, j \in \mathbb{I}_K : (C_i^{(m)} = C_j^{(m)}) \wedge (C_i^{(m)} > 0)\}\,. \tag{2}$$

According to this definition, markers are not restricted to single copy protein domains. With the query-specific markers we then apply the standard estimation scheme. From M specific markers in $\mathbb{M}$ with reference counts $C_r^{(m)}$ we get the stage I contamination (cont) and completeness (comp) estimates

$$\mathrm{cont} = \frac{1}{M} \sum_{m \in \mathbb{M}} \left[\frac{C_q^{(m)}}{C_r^{(m)}} - 1\right]_+, \tag{3}$$

$$\mathrm{comp} = \frac{1}{M} \sum_{m \in \mathbb{M}} \frac{C_q^{(m)}}{C_r^{(m)}} - \mathrm{cont}, \tag{4}$$

where $[z]_+ = \max(z, 0)$. If these estimates are within the above-mentioned range (contamination below 30% and completeness above 60%), stage II is utilized to refine these estimates. Otherwise, the predictions of stage I are reported as the final result.

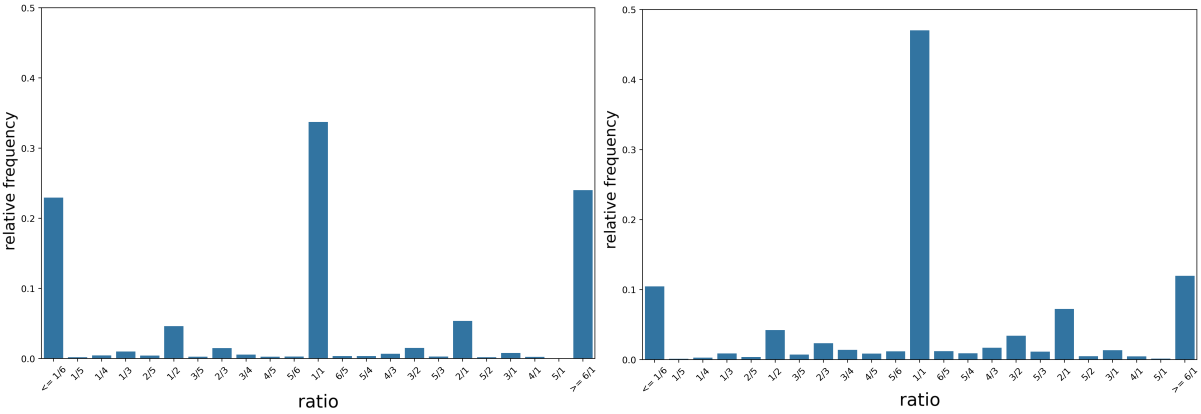

Figure 2: Count Ratio Histograms for two simulated example genomes with $c_{max} = 6$. The left histogram shows a bin with 61% completeness and 30% contamination. The right one for a bin with 90% completeness and 10% contamination.

2.1.5 Count ratio histograms

The original feature space as described above comprises more than 10 000 dimensions which correspond to different protein domain families. Large-scale machine learning within such a high-dimensional space is burdensome. While neural networks, in principle, are well-suited for training with large data sets, a high-dimensional input space slows down iterative training and increases the risk of overfitting. Therefore, we mapped the original profile space to a lower dimensional histogram space. A count ratio histogram (CRH) arises from the comparison of a candidate profile with a reference profile in terms of the observed ratios between the corresponding protein domain counts. More specifically we consider all ratios $C_q^{(i)}/C_r^{(i)}$ between counts from the query (nominator) and reference (denominator) genomes, where i is the Pfam domain index. Note that all Pfam domain families with non-zero counts are included in the CRH computation, not just the markers that have been used for the initial prediction in stage I.

As bin centers for the CRH we use the set of possible ratios between counts in a range $1 \ldots c_{\max}$. Thus, the integer variable $c_{\max}$ specifies the resolution of the histogram. For example, with $c_{\max} = 4$ we obtain the bin centers $C = \{1/4, 1/3, 1/2, 2/3, 3/4, 1, 4/3, 3/2, 2, 3, 4\}$. In addition, the left-most bin of the histogram also represents the domains of the reference genomes that have not been observed in the query genome. Similarly, on the other extreme of the histogram we use the last right-most bin to also count all the protein families with a count ratio larger than $c_{\max}$ and we also include those entries that occur in the query genome, but not in the reference in this bin.

The histogram is normalized to relative frequencies. Therefore, a CRH with a single central peak at 1 would indicate a highly complete query genome without contamination. With a decreasing completeness the variance of the count ratio distribution increases and more probability mass will be observed in non-central bins on the left hand side. In contrast, an increase of the right hand side bins indicates a growing contamination. Two examples of CRHs for query genomes with different completeness and contamination values are shown in Figure 2.

2.1.6 Machine learning methods

With the CRH feature vectors we improved the marker-based estimates of the stage I prediction with different machine learning approaches in stage II. The query input of stage II was obtained from the average over all CRH vectors that result from comparison of the query with the K nearest references in stage I. As additional features we used the estimates of completeness and contamination as predicted in stage I. To identify suitable machine learning methods, we compared several linear and non-linear regression techniques as implemented in `sklearn` Vers. 1.3.1 [11]. For linear prediction we evaluated SVM and elastic net regression, and as non-linear approaches we tested nearest neighbour, Neural Network and Random Forest regression. For each method we identified suitable values for the hyperparameters by performing a grid search for the stage I neighbourhood size K and for the CRH resolution $c_{\max}$. Additionally, method-specific hyperparameters were included in the grid search where necessary. In particular, this included the SVM regularization parameter, the weight decay in the feed-forward Neural

Networks with one hidden layer, and the smoothing parameter for the nearest neighbour regression. For the elastic net we used the built-in hyperparameter optimization and for Random Forests we used the default parameter values in the `sklearn` implementation.

2.2 Training and test data

2.2.1 Reference database

Our method requires a database of reference genomes for comparative analysis. As a basis we use all genomes from the RefSeq [12] database with a *complete* or *chromosome* status annotation[1]. We aim to provide references with a high quality that cover a wide range of different species without redundant protein profiles. We achieved this with a multi-stage filtering process: at first, we used UProC to determine the Pfam frequencies of all downloaded RefSeq genomes. We then applied an agglomerative clustering algorithm to the obtained frequencies in order to filter out closely related genomes, keeping only one representative for each cluster. Based on these representatives we determined a set of single-count Pfam markers, separately for bacteria and archaea. We consider a protein family to provide such a general marker if and only if it occurs exactly once in at least 95% of our cluster representatives. This results in a set 107 markers for bacteria and 128 for archaea that we use to calculate a first completeness estimate of all downloaded genomes and removed those that had an estimated completeness of less than 95%. We applied this step to reduce the risk of including RefSeq entries with an erroneous completion state annotation. In a final step, all genomes that fulfilled the completeness criterion were clustered again, using the same method as before. The resulting cluster representatives constitute the genomes of our actual reference data set.

2.2.2 Simulation

For our evaluation we generated input query genomes with known ground truth according to specified values for completeness and contamination. In principle, we tried to simulate the results of a metagenome binning process that shows a broad spectrum of genome quality indices. First, all input genomes were fragmented and fragments from different genomes were randomly combined to simulate variable degrees of completeness and contamination. For training we used a 20kb fragment length for simulated contigs. For the evaluation on separate test data we also used other fragment lengths to study the impact of a differing contig length during prediction.

The simulation of a query genome was performed in the following way: first, random values for completeness and contamination were independently drawn from the ranges 60% to 100% and 0% to 30%, respectively. Then, a query genome was randomly selected from the set of candidate genomes and fragments were randomly drawn to match the given completeness. Contamination refers to the original complete length of the query genome and a corresponding number of fragments was drawn from a randomly selected contaminant genome. However, random selection is restricted to genomes with a similar genomic signature in terms of the similarity between tetramer frequency profiles. By this, we avoid mixing genomes with very different genomic signatures, which is usually not met in metagenomic binning results. If the selected contaminant genome is too short to provide the required contamination, more genomes are drawn from the similarity range until the specified contamination can be realized. For the range selection we considered genomes according to tetramer profiles with a Bray-Curtis similarity ≥ 80 %.

2.2.3 Training and validation

To obtain separate data sets for training of the machine learning methods and validation of the hyperparameters we performed a data splitting scheme based on the genome release date. The underlying idea is to simulate the novelty of genome data under realistic conditions, which imply a mixture of all kinds of evolutionary distances between the more recent genomes and previously published data. We divided the set of RefSeq cluster representatives into three subsets according to two split dates[2]: the "oldest" and largest part 1 contains 6 036 genomes that served exclusively as reference genomes for the database nearest neighbour search of our approach. The middle part 2 and the most recent part 3 were interchangeably used for training and validation, containing 1 503 genomes and 1 515 genomes, respectively. For each of the training/validation genomes we simulated 20 bins with randomly chosen completeness and contamination (see above). The contaminant genomes were always drawn from the same part, to

[1] downloaded on the 2nd September 2023
[2] 5 May 2021 and 26 July 2022

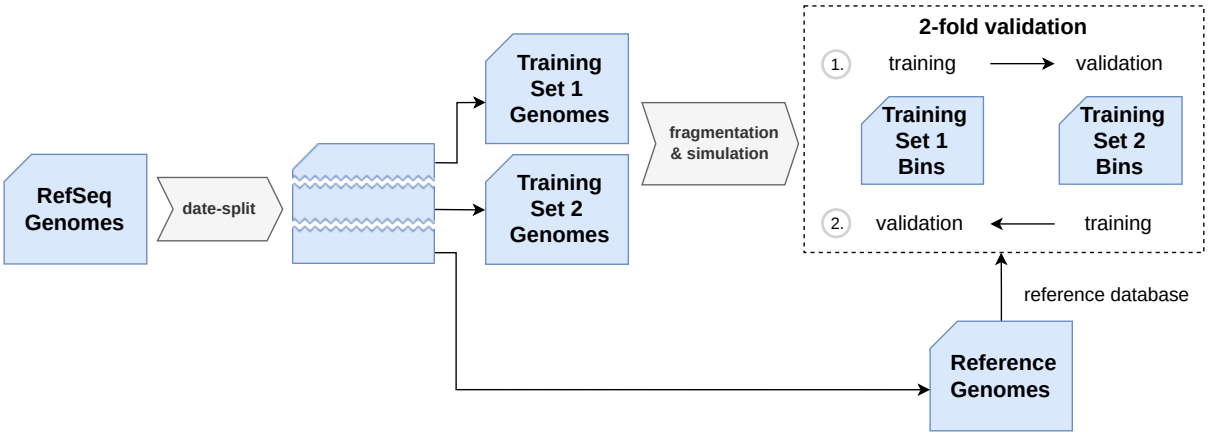

Figure 3: Schematic overview of the two-fold training and validation process with two subsets (Set 1, Set 2). If the first subset is used for training, the second is used for validation and vice versa.

method	hyperparameter	value ranges
all	Pfam version	24, 28
all	K (ref. neighbours)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
all	c_{max}	4, 5, 6, 7, 8, 10, 12
Linear SVR	C	1, 0.9, 0.8, 0.5, 0.3, 0.1, 0.01, 2, 3, 5, 10, 20, 40, 100, 125, 150, 175, 200
Knn Regression	knn	5, 10, 15, 20, 25, 30, 40 50, 60, 70, 80, 90, 100
MLP	alpha	0.0001, 0.00001, 0.000001, 0.0000001

Table 1: Table of all possible hyperparameter ranges for the grid search.

avoid overlap between train and validation data. For part 2 and part 3 we realized a simple 2-fold cross-validation. In a single fold, one part was used for training and one for validation. The overall validation performance in terms of the mean absolute error (MAE) in percentage points was averaged over the two validation folds. Figure 3 shows an overview of the training/validation process. For training and validation, feature vectors were computed for all query bins generated from the corresponding parts. The training feature vectors were used to train the machine learning regression models for predicting completeness and contamination. The trained methods were then evaluated using the validation feature vectors.

According to the lowest validation error, we identified optimal hyperparameter values for the different machine learning approaches by grid search over a defined set of values. The ranges of the parameters can be found in Table 1. While some hyperparameters are specific to particular machine learning methods, the Pfam version, K neighbours, and c_{max} apply to all methods because they affect the input data for all learners.

2.2.4 Test data and setup

After the selection of hyperparameters based on the 2-fold cross-validation, we chose the method with the best validation performance for the final prediction engine of CoCoPyE. In that way, we included two neural networks, one for the completeness and one for the contamination prediction. For the final version, the two networks were trained with both training sets (part 2 and 3) combined, since there was no need for a separate validation fold anymore. Again, the first subset (part 1) was used to provide the reference database. After the training, the final reference database for CoCoPyE was built from all representatives of the RefSeq dataset, i.e., we extended the reference set by adding the genomes from both training sets.

We compared CoCoPyE with the existing tools CheckM (referred to as CheckM1 from now on) and CheckM2 in a comprehensive evaluation. We prepared two new genome sets A and B to ensure that

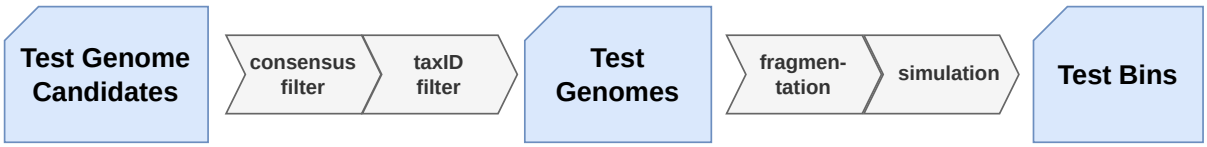

Figure 4: Schematic overview of the test data generation.

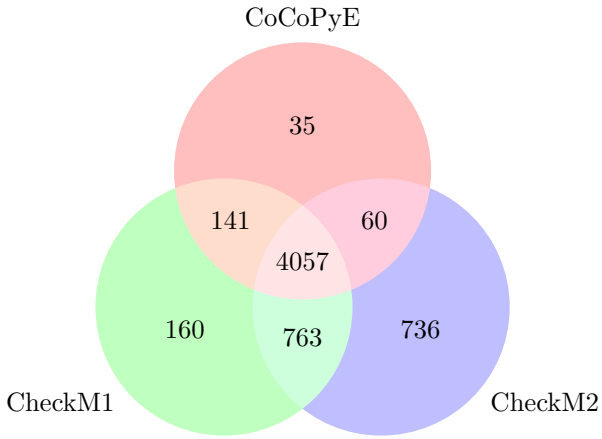

Figure 5: Number of test genome candidates that have at least a completeness of 95% and a contamination of 5% or less for CheckM1, CheckM2 and CoCoPyE.

the basis for the test data is distinct from that already used for the setup and training of the prediction engine.

Set A From a phylogenomics study [13] that included a broad range of bacterial and archaeal genomes, we selected a subset of genomes according to assembly levels *contig* and *scaffold*, contrary to the selection criterion for our RefSeq data set. With the resulting 8309 test genome candidates we generated the test data. An overview of this process is shown in Figure 4.

First, the genomes were analyzed by CheckM1, CheckM2 and CoCoPyE to provide a subset of test genomes where all three tools predict at least a completeness of 95% and a contamination of no more than 5%. The reason for this selection was two-fold: first, we are mainly interested in the ability of the prediction engines to cope with varying degrees of incompleteness and contamination. In particular, the case where incomplete genomes are mixed with a significant contamination is an algorithmic and statistical challenge even if a method can recognize the uncontaminated complete versions as high quality genomes. Furthermore, to account for the differing training and reference data used for implementation of the tools, it is fair to choose only test genomes for which all tools agree in the prediction of a high quality. Finally, we think this consensus analysis is the best way to cope with a missing ground truth, because for most of the reconstructed genomes in current databases the actual quality of the reconstruction is not entirely clear. From the consensus analysis we obtained 4 057 genomes and Figure 5 shows the agreement of the different tools. To avoid any direct overlap with our previous RefSeq-based data, we only considered genomes with a taxID not included in our training and reference sets. After consensus analysis and taxID filtering, we finally obtained a test set of 3 540 genomes.

With these test genomes, we generated three separate sets of query bins. We applied the simulation scheme as described above to provide test sets with 20kb, 50kb, and 100kb fragment length. For each set we generated 10 bins per test genome, again with randomly selected completeness and contamination values, so each of the three final test sets contained 35 400 bins. The three tools were then run on all test data to evaluate their prediction performance.

Set B From the "Genomes from Earth's Microbiomes" (GEM) catalog [14], a collection of metagenome assembled genomes (MAGs), an additional test data set was built to study the prediction performance on simulated bins with real contig length distributions. Thus, we did not apply any fragmentation and used

Tool	Completeness Error		Contamination Error	
	Mean	Median	Mean	Median
CoCoPyE Stage I	3.74 (±3.33)	2.83	5.32 (±4.64)	4.08
ElasticNetCV	2.97 (±2.64)	2.28	4.24 (±3.24)	3.55
LinearSVR	2.94 (±2.59)	2.26	4.09 (±3.19)	3.38
Knn Regression	2.73 (±2.45)	2.07	3.80 (±3.05)	3.06
Random Forest	2.63 (±2.41)	1.98	3.68 (±3.01)	2.95
MLP	2.58 (±2.38)	1.95	3.50 (±2.94)	2.77

Table 2: Mean and median absolute errors in percentage points of machine learning methods and the stage I prediction of CoCoPyE. Results on validation data are shown for the best-performing choice of hyperparameters for each method.

the sequences in terms of contigs as obtained from GEM. GEM contains 52 515 MAGs, of which we used 9 143 high-quality MAGs according to the MIMAG standard. Besides covering a wide range of microbial genomic diversity, the dataset provides information about habitats of the genetic material. This enabled us to conduct a more detailed performance analysis on environmental (further subdivided into aquatic and terrestrial), host-associated and engineered MAGs. For the selection of suitable MAGs sufficiently distant from the references we only considered test genomes with a minimum dissimilarity (Bray-Curtis) of 10% of the corresponding protein domain profile to the closest reference profile. In the same way as for the selection of our first test set, all compared tools must agree on the high quality of the full MAG sequence. Thus, we only chose those MAGs for testing for which all tools predicted a completeness $\geq 95\%$ and a contamination $\leq 5\%$. Finally, to ensure variety in the test set, we clustered the remaining MAG protein domain profiles retaining one representative per cluster, similar to the reference set clustering before. This resulted in 1 210 test MAGS (421 environmental (thereof 297 aquatic, 124 terrestrial), 274 engineered, 515 host-associated) containing contigs, which were directly used for simulating test bins with different completeness and contamination values. The test bins were simulated in the same way as for test set A, except for the exclusion of the fragmentation step. For data set B we finally obtained 12004 test bins.

3 Results

3.1 Validation and model selection

As described in our training and validation setup we compared several machine learning models for stage II of our prediction engine and selected suitable values for the hyperparameters. Table 2 shows the mean and median absolute error in percentage points for the prediction of completeness and contamination on validation data. For comparison we also included the results of the marker-based prediction of stage I. While the stage I prediction already works quite well, all methods in stage II could improve the initial results. Here, the nonlinear methods (Knn, RF, MLP) show a slightly better prediction than the linear approaches (Elastic Net, SVR). The best result was achieved by the Neural Network with a MAE of 2.58%*pt* and 3.50%*pt* for completeness and contamination. In this case, we used networks with 100 neurons and feature vectors based on Pfam version 28 and $K = 9$ neighbours. The network for the completeness prediction had the parameters $\alpha = 0.0001$, $c_{\max} = 6$, and $\alpha = 0.0000001$ and $c_{\max} = 12$ for the CRH and contamination. Both neural networks were included in the final version of the CoCoPyE prediction engine. The best result was achieved with the larger UProC database (Pfam 28) for protein domain detection. With the smaller database (Pfam 24) the MLP produced the best results but the performance decreased slightly to 2.76%*pt* and 3.77%*pt*. This could justify the use of the smaller version in computers with limited RAM.

Comparing the different machine learning approaches, the performance differences between different methods were relatively small. Remarkably, Random Forest regression with built-in default settings was very close to the best MLP results, although we did not perform any hyperparameter optimization.

Tool	Completeness Error		Contamination Error	
	Mean	Median	Mean	Median
CoCoPyE	3.09 (±3.05)	2.26	4.49 (±4.52)	3.29
CheckM1	4.51 (±4.89)	2.96	7.78 (±8.33)	5.39
CheckM2	6.13 (±5.67)	4.45	7.39 (±6.20)	5.78

Table 3: Mean and median absolute error in percentage points for the test set A with 20kb fragments.

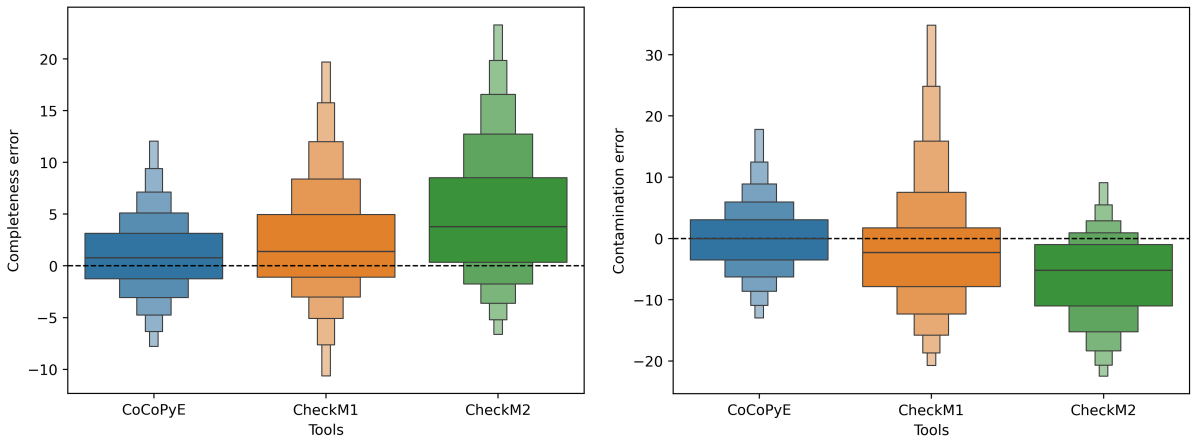

Figure 6: Signed error distributions of completeness (left) and contamination (right) for the compared tools on test set A with errors measured in percentage points. Note that the scale of the y-axis differs between the two plots.

3.2 Performance comparison

In our evaluation of the final tool on test data we compared CoCoPyE with CheckM1 [1] and CheckM2 [6]. For test data set A the results in Table 3 indicate that CoCoPyE yields the lowest prediction error for both completeness and contamination, with an MAE of 3.09%*pt* and 4.49%*pt*. While CheckM1 outperforms CheckM2 in terms of a lower completeness error (4.51%*pt* vs. 6.13%*pt*), for the prediction of contamination the MAE for CheckM1 is higher than for CheckM2 (7.78%*pt* vs. 7.39%*pt*).

For further analysis of the prediction error we inspected the distributions of signed deviations as shown in Figure 6. While the completeness error of CoCoPyE shows a roughly symmetric distribution, CheckM2 shows a skewed distribution indicating a clear tendency for overprediction of the quality index. A slight overprediction of completeness is also visible for CheckM1 and CoCoPyE, where CoCoPyE shows the lowest bias of all tools. The prediction of contamination also shows clear differences between the tools: while CoCoPyE yields an almost unbiased distribution, CheckM2 has a clear tendency to underestimate contamination. This is also visible in a weaker form for CheckM1, which however shows a highly asymmetric distribution with a heavy tail for the positive error.

We also investigated the performance for different fragment lengths of the simulated test data. Here, we observed a slight decrease in the prediction performance for all tools as shown in Table 5 and Table 6 for 50kb and 100kb fragment lengths. Because CoCoPyE was trained with 20kb fragments this result was not entirely surprising. Therefore it could be beneficial to provide different neural networks trained with different fragment lengths. In this case, for the prediction at stage II, the network that best matches the average contig length of the input would be selected automatically.

To investigate if the CoCoPyE prediction error depends on the phylogenetic distance to the reference data we analyzed the relationship of test and reference genomes in terms of the GTDB taxonomy [15]. In this analysis (A.2) we found that the error slightly increases with the deviation between taxonomic labels. As expected, the MAE for test data was minimal when a reference genome with the same species label exists. We observed a maximum increase of 1.01%*pt* for completeness when the taxonomic labels of test and reference data at most agreed up to order level (see Table 7). For contamination we measured a maximum increase of 1.28%*pt*. Note that this analysis does not reflect, how CoCoPyE actually uses

the reference data because the prediction always depends on multiple references with possibly varying taxonomy.

Unlike test set A, test set B is not based on a fixed fragment length because it uses the original MAG contigs. In comparison to set A, the results for set B show a slight degradation of the CoCoPyE prediction for completeness with an overall MAE of 4.07%*pt* (see Table 4). Meanwhile, the contamination error is slightly below the set A results with 3.75%*pt*. The CheckM1 results moved in the same directions, with overall higher values. In contrast, both the completeness error and contamination error slightly decreased for CheckM2 on test set B. However, CoCoPyE still has the best overall prediction performance, yielding the lowest error for completeness as well as for contamination.

The analysis of the signed error distributions shows a similar result like the analysis for set A. Again, CheckM2 errors are clearly biased towards overestimation for completeness and underestimation for contamination (see Figure 7). Here, CheckM1 shows a more balanced distribution than for set A but for contamination the underestimation bias is still visible. Also for set B CoCoPyE shows the most balanced distribution for both, completeness and contamination error.

For set B we were able to assign all MAG-based test bins to ecosystem categories based on the main genome component in simulated bins. Therefore, we could analyze the variation of the prediction performance across different habitat types. The results (see Table 8 and Table 9) show that CoCoPyE has a relatively stable predictive power among all the examined habitat categories with completeness MAE values ranging from 3.89%*pt* (engineered) up to 4.65%*pt* (terrestrial) according to a maximum difference of 0.76%*pt*. The contamination MAE values show a maximum difference of 0.45%*pt* ranging from 3.56%*pt* (engineered) up to 4.01%*pt* (aquatic). CheckM1 provides slightly more widespread prediction errors amongst the habitats with maximum MAE differences of 1.12%*pt* in completeness and 0.92%*pt* in contamination. CheckM2 revealed the highest variation between ecosystem categories with a completeness spread of 1.5%*pt* and a contamination spread of 1.17%*pt*.

	Completeness Error		**Contamination Error**	
Tool	Mean	Median	Mean	Median
CoCoPyE	4.07 (±3.47)	3.22	3.75 (±3.52)	2.81
CheckM1	5.21 (±5.17)	3.68	5.85 (±6.18)	3.95
CheckM2	5.91 (±5.38)	4.35	6.27 (±5.74)	4.79

Table 4: Mean and median absolute error in percentage points for test set B.

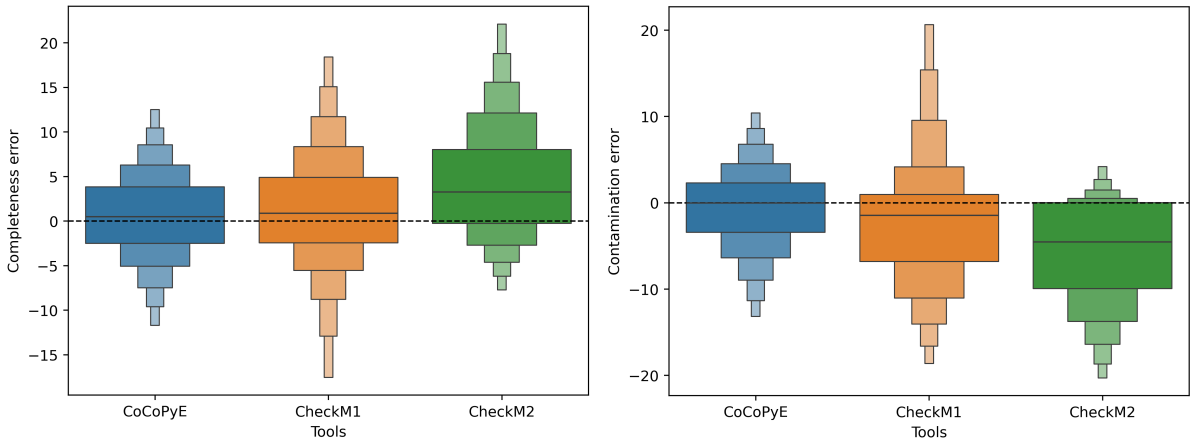

Figure 7: Signed error distributions of completeness (left) and contamination (right) for the compared tools on test data set B with errors measured in percentage points. Note that the scale of the y-axis differs between the two plots.

Runtime In addition to the prediction performance, we also evaluated the runtime and memory usage of the three tools. For the evaluation we used a workstation computer with an Intel Core i9-13900

processor and 64GB of memory. The tests were run with CoCoPyE 0.2.1 on Debian 12.4. For measuring runtime and peak memory usage we used the GNU `time` tool[3]. As test sets for runtime measurement we used 10 mutually exclusive random subsets from our original set of test bins, each containing sequences from 1 000 simulated bins with an average size of 3.8Mb per bin. While CheckM1 showed the longest average runtime of 94min 21s, CheckM2 required 63min 45s and CoCoPyE only 17min 36s. The runtime variation across different runs was small for all tools. The maximum deviation between two different runs we measured for CheckM1 (4min 30s), the minimum for CoCoPyE (51s). On average, CoCoPyE required 16.6GB of memory, CheckM1 36.8GB and CheckM2 17.7GB.

3.3 Tool

Our feature-based prediction pipeline is included in CoCoPyE, which is available on PyPI and conda-forge and can be installed with the respective package managers, pip and conda. The source code is available on GitHub[4] under the GNU General Public License, version 3. The tool supports Windows, Mac OS and Linux. We also offer a web server[5] to try the tool without installation. In addition to the prediction of genome completeness and contamination, CoCoPyE also provides a lowest common ancestor taxonomic classification for the genome, based on the NCBI taxonomy annotation of the nearest neighbours in the reference database. Furthermore, neighbourhood similarities are provided for further analysis of the prediction confidence.

Currently, the user can choose between versions 24 and 28 of the Pfam database as used for the UProC-based feature extraction stage. We do not offer a more recent Pfam database because our results so far indicate that a growing sequence database with an increased number of protein families only has a minor impact on the final prediction performance. In contrast, the impact on RAM storage requirements and startup runtime of UProC is considerable. While UProC with Pfam version 24 runs on 16 GB machines without problems, we recommend 32 GB for version 28 while the actual version 36 would at least require 64 GB of memory. However, with the most recent Pfam version, structure predictions based on AlphaFold [16] have been used to improve many Pfam definitions in terms of more accurate domain boundaries. Therefore, we are currently working on an accessory tool that allows an automatic update to the latest Pfam version if sufficient memory is available.

4 Discussion

For several years CheckM1 has represented the state-of-the-art for predicting genome quality indices. With CoCoPyE we now provide a further development of marker-based estimation. The main differences to CheckM1 are the query-specific generation of suitable marker sets and the inclusion of machine learning methods for a refinement of marker-based estimates. In contrast to CheckM1, with CoCoPyE potential markers are not restricted to single copy domains. Furthermore, the dynamic marker extraction facilitates the setup and maintenance of the tool because it overcomes the requirement of a prior definition of lineage-specific marker sets. In CoCoPyE we use pre-defined markers only for prior filtering of insufficient input. All markers that are used for prediction are determined at runtime for a particular query genome. Because we do not rely on phylogenetic placement for marker set selection or Hidden Markov Models for protein domain detection, our tool is also considerably faster than CheckM1. Finally, our results indicate a clear improvement of the prediction accuracy. This can partly be attributed to our two-stage hybrid architecture. The advantage over purely marker-based tools like CheckM1 or BUSCO is that the second stage in CoCoPyE can learn how to compensate for errors that may result from the restriction to a limited set of markers, in particular with regard to contamination estimates [17].

A direct conceptual comparison with CheckM2 is more difficult, because it is based on a completely different prediction approach and does not use any kind of marker identification. Instead, functional profiles, in terms of frequencies of KEGG orthologs, together with a few other genome content features, are directly used for training and prediction with a machine learning approach. Although the extraction of the relevant information can in principle be learned from such high-dimensional input, the approach requires a large amount of training examples and particular care to avoid overfitting.

Because CheckM2 does not depend on the identification of suitable reference genomes and marker sets

[3] https://www.gnu.org/software/time/
[4] https://github.com/gobics/cocopye
[5] https://cocopye.uni-goettingen.de

it has the potential to provide reasonable results in cases where CheckM1 does not yield a meaningful prediction [6]. In contrast to CheckM1, CoCoPyE provides a more flexible scheme to evaluate the reference data, but finally it also depends on the existence of suitable reference genomes. Therefore, CoCoPyE is not intended to replace CheckM2 and possibly a combination of both tools can be beneficial. In particular, if suitable reference genomes for the neighbourhood-based analysis exist, CoCoPyE provides a more accurate prediction than CheckM2. Besides the estimation of completeness and contamination, CoCoPyE provides additional information which shows that it is more than just a black box: the number of markers in the stage I prediction, together with the neighbourhood similarity scores and the taxonomic classification, may provide an indicator for the confidence of the prediction and may also be helpful to decide in which cases CheckM2 should possibly be preferred for prediction.

Acknowledgements P. Meinicke was partly supported by Deutsche Forschungsgemeinschaft (DFG).

Data Availability
The test data that was used for comparative evaluation of the prediction performance is available at:

https://data.goettingen-research-online.de/dataset.xhtml?persistentId=doi:10.25625/H7QRXS

Availability of supporting source code and requirements
Project name: CoCoPyE
Project home page: https://github.com/gobics/cocopye
Operating system(s): Windows, MacOS, Linux
Programming language: Python
Other requirements: Python 3.8 or higher, UProC 1.2.0 or higher
License: GNU GPL v3 or higher

References

[1] Donovan H. Parks, Michael Imelfort, Connor T. Skennerton, Philip Hugenholtz, and Gene W. Tyson. "CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes". In: *Genome Research* 25.7 (2015), pp. 1043–1055. DOI: 10.1101/gr.186072.114.

[2] Robert M. Bowers et al. "Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea". In: *Nature Biotechnology* 35.8 (2017), pp. 725–731. DOI: 10.1038/nbt.3893.

[3] Francesca D. Ciccarelli et al. "Toward Automatic Reconstruction of a Highly Resolved Tree of Life". In: *Science* 311.5765 (2006), pp. 1283–1287. DOI: 10.1126/science.1123061.

[4] Dongying Wu, Guillaume Jospin, and Jonathan A. Eisen. "Systematic Identification of Gene Families for Use as "Markers" for Phylogenetic and Phylogeny-Driven Ecological Studies of Bacteria and Archaea and Their Major Subgroups". In: *PLOS ONE* 8.10 (2013), e77033. DOI: 10.1371/journal.pone.0077033.

[5] Mosè Manni, Matthew R. Berkeley, Mathieu Seppey, and Evgeny M. Zdobnov. "BUSCO: Assessing Genomic Data Quality and Beyond". In: *Current Protocols* 1.12 (2021), e323. DOI: 10.1002/cpz1.323.

[6] Alex Chklovski, Donovan H. Parks, Ben J. Woodcroft, and Gene W. Tyson. "CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning". In: *Nature Methods* 20.8 (2023), pp. 1203–1212. DOI: 10.1038/s41592-023-01940-w.

[7] Gleb Goussarov et al. "Accurate prediction of metagenome-assembled genome completeness by MAGISTA, a random forest model built on alignment-free intra-bin statistics". In: *Environmental Microbiome* 17.1 (2022), p. 9. DOI: 10.1186/s40793-022-00403-7.

[8] Bruce Parrello et al. "A machine learning-based service for estimating quality of genomes using PATRIC". In: *BMC Bioinformatics* 20.1 (2019), p. 486. DOI: 10.1186/s12859-019-3068-y.

[9] Jaina Mistry et al. "Pfam: The protein families database in 2021". In: *Nucleic Acids Research* 49 (2021), pp. D412–D419. DOI: 10.1093/nar/gkaa913.

[10] Peter Meinicke. "UProC: tools for ultra-fast protein domain classification". In: *Bioinformatics* 31.9 (2015), pp. 1382–1388. DOI: 10.1093/bioinformatics/btu843.

[11] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830.

[12] Nuala A. O'Leary et al. "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation". In: *Nucleic Acids Research* 44 (2016), pp. D733–745. DOI: 10.1093/nar/gkv1189.

[13] Qiyun Zhu et al. "Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea". In: *Nature Communications* 10.1 (2019), p. 5477. DOI: 10.1038/s41467-019-13443-4.

[14] Stephen Nayfach et al. "A genomic catalog of Earth's microbiomes". In: *Nature Biotechnology* 39.4 (2021), pp. 499–509. DOI: 10.1038/s41587-020-0718-6.

[15] Donovan H. Parks et al. "A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life". In: *Nature Biotechnology* 36.10 (2018), pp. 996–1004. DOI: 10.1038/nbt.4229.

[16] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[17] Askarbek Orakov et al. "GUNC: detection of chimerism and contamination in prokaryotic genomes". In: *Genome Biology* 22.1 (2021), p. 178. DOI: 10.1186/s13059-021-02393-0.

A Additional material

A.1 Fragment length-specific performance

	Completeness Error		Contamination Error	
Tool	Mean	Median	Mean	Median
CoCoPyE	3.45 (±3.30)	2.53	4.79 (±4.70)	3.54
CheckM1	4.95 (±5.25)	3.26	8.24 (±8.57)	5.89
CheckM2	6.49 (±5.88)	4.78	7.83 (±6.42)	6.28

Table 5: Mean and median absolute error in percentage points for the test set with 50kb fragments.

	Completeness Error		Contamination Error	
Tool	Mean	Median	Mean	Median
CoCoPyE	3.77 (±3.62)	2.77	5.18 (±4.88)	3.94
CheckM1	5.40 (±5.58)	3.64	8.77 (±8.83)	6.38
CheckM2	6.81 (±6.09)	5.10	8.31 (±6.69)	6.77

Table 6: Mean and median absolute error in percentage points for the test set with 100kb fragments.

A.2 Taxonomic analysis of test data

We assessed the taxonomic relationship between set A test genomes and our reference genomes as follows: Taxonomic annotations for genomes were retrieved from the latest release[6] of GTDB [15]. We added annotations to our reference and test genomes based on the retrieved GTDB taxonomy where possible. By this, we were able to annotate 9 045 out of all 9 054 reference genomes (99.9%) and 3 028 out of all 3 540 test genomes (85.5%). We then created a taxonomic tree based on all labeled reference genomes. Subsequently, for each labeled test genome we obtained the reference genome with the closest taxonomic label and assigned the most specific taxonomic category that is shared between both. In total, we found 759 assignments on species level, 1 667 on genus level and 532, 51, 18 on family, order and class level. With these assignments we analyzed the category-specific prediction performance. For phylum level and above we obtained too few assignments for a statistical analysis.

From the prediction results on set A test bins (see Table 7) we observe a slight increase in mean absolute error (for completeness and contamination) with increasing taxonomic distance between a test genome and its next closest reference genome. Thereby, the total increase from species to order level is 1.01 percentage points for completeness. The mean absolute error then slightly decreases (0.25 percentage points) from order to class level, which may be attributed to the smaller number of class category assignments. For contamination, we observed a continuous increase with in total 1.28 percentage points from species to class level.

	Completeness Error		Contamination Error	
tax cat	Mean	Median	Mean	Median
species	2.72 (±2.55)	2.03	3.97 (±3.72)	2.95
genus	3.08 (±3.08)	2.23	4.44 (±4.52)	3.24
family	3.38 (±3.30)	2.52	4.87 (±4.99)	3.52
order	3.73 (±3.56)	2.72	5.10 (±4.78)	4.05
class	3.48 (±3.34)	2.64	5.25 (±5.03)	4.40

Table 7: Mean and median absolute error in percentage points depending on taxonomic closeness of next reference genome.

Note that the taxonomically closest reference genome does not reflect which reference genomes are actually used for the CoCoPyE predictions. Because CoCoPyE always requires K-nearest neighbours in protein profile space the taxonomic categories may vary among neighbours and not necessarily correspond to the closest category.

[6] by 8th of May, 2024 from https://data.ace.uq.edu.au/public/gtdb/data/releases/latest/

A.3 Ecosystem category-specific performance

	Completeness Error					
	Mean			Median		
Tool	Host	Engineered	Environmental	Host	Engineered	Environmental
CoCoPyE	4.06 (±3.37)	3.89 (±3.38)	4.20 (±3.64)	3.27	3.08	3.28
CheckM1	4.98 (±4.82)	5.23 (±5.12)	5.47 (±5.59)	3.62	3.70	3.75
CheckM2	5.46 (±4.96)	6.10 (±5.44)	6.33 (±5.80)	4.08	4.65	4.64

	Contamination Error					
	Mean			Median		
Tool	Host	Engineered	Environmental	Host	Engineered	Environmental
CoCoPyE	3.67 (±3.41)	3.56 (±3.37)	3.97 (±3.72)	2.76	2.61	2.96
CheckM1	5.74 (±5.69)	5.59 (±5.84)	6.17 (±6.93)	4.01	3.71	4.02
CheckM2	6.66 (±5.86)	5.78 (±5.44)	6.13 (±5.75)	5.28	4.22	4.52

Table 8: Mean and median absolute error in percentage points for set B test bins specific to ecosystem categories.

	Completeness Error			
	Mean		Median	
Tool	Aquatic	Terrestrial	Aquatic	Terrestrial
CoCoPyE	4.01 (±3.43)	4.65 (±4.06)	3.17	3.48
CheckM1	5.02 (±5.25)	6.10 (±6.26)	3.61	4.16
CheckM2	6.06 (±5.44)	6.96 (±6.51)	4.46	4.99

	Contamination Error			
	Mean		Median	
Tool	Aquatic	Terrestrial	Aquatic	Terrestrial
CoCoPyE	4.01 (±3.77)	3.86 (±3.61)	2.94	2.99
CheckM1	6.02 (±6.45)	6.51 (±7.91)	4.07	3.95
CheckM2	6.40 (±5.82)	5.49 (±5.52)	4.97	3.67

Table 9: Mean and median absolute error in percentage points for set B test bins assigned to environmental categories.