NORTH: a highly accurate and scalable Naive Bayes based ORTHologous gene clustering algorithm

Nabil Ibtehaz[1], Shafayat Ahmed[1], Bishwajit Saha[1], M. Sohel Rahman[1], and
Md. Shamsuzzoha Bayzid[1,*]

[1]Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka-1205, Bangladesh
[*]Corresponding author

January 23, 2019

Abstract

Background: The principal objective of comparative genomics is inferring attributes of an unknown gene by comparing it with well-studied genes. In this regard, identifying orthologous genes plays a pivotal role as the orthologous genes remain less diverged in the course of evolution. However, identifying orthologous genes is often difficult, slow, and idiosyncratic, especially in the presence of multiplicity of domains in proteins, evolutionary dynamics (gene duplication, transfer, loss, introgression etc.), multiple paralogous genes, incomplete genome data, and for distantly related species where similarity is hard to recognize.

Motivation: Advances in identifying orthologs have mostly been constrained to developing databases of genes or methods which involve computationally expensive BLAST search or constructing phylogenetic trees to infer orthologous relationships. These methods do not generally scale well and cannot analyze large amount of data from diverse organisms with high accuracy. Moreover, most of these methods involve manual parameter tuning, and hence are neither fully automated nor free from human bias.

Results: We present NORTH, a novel, automated, highly accurate and scalable machine learning based orhtologous gene clustering method. We have utilized the biological basis and intuition of orthologous genes and made an effort to incorporate appropriate ideas from machine learning (ML) and natural language processing (NLP). We have discovered that the BLAST search based protocols deeply resemble a "text classification" problem. Thus, we employ the robust *bag-of-words model* accompanied by a Naive Bayes classifier to cluster the orthologous genes. We studied 1,255,877 genes in the largest 250 ortholog clusters from the KEGG database, across 3,880 organisms comprising the six major groups of life, namely, Archaea, Bacteria, Animals, Fungi, Plants and Protists. Despite having more than a million of genes on

distantly related species with acute data imbalance, NORTH is able to cluster them with 98.48% Precision, 98.43% Recall and 98.44% F_1 score, showing that automatic orthologous gene clustering can be both highly accurate and scalable. NORTH is available as a web interface with a server side application, along with cross-platform native applications (available at `https://nibtehaz.github.io/NORTH/`) – allowing queries based on individual genes.

Index terms— Orthologous genes, Comparative Genomics, Homology, Machine Learning, Bag of words, Naive Bayes.

1 Introduction

The concept of homology has been adopted as the basis of phylogenetics and comparative genomics. Although there have been many arguments about its interpretation [1], homology can be defined as a similarity relationship between features that is due to shared ancestry. Homologous sequences can be further divided into orthologs and paralogs, according to how they diverged from their common ancestor. As originally defined in [2], homologous sequences that are derived through speciation are called orthologs. On the contrary, paralogs are homologous sequences that are derived from a duplication event [2] [3]. Moreover, these orthologs and paralogs can further be categorized like in-paralogs issued from duplication post-speciation and out-paralogs corresponding to duplication prior to speciation [4].

Detection of orthologs is of utmost importance in many fields of biology, especially in the annotation of genomes, functional genomics, evolutionary and comparative genomics. The pattern of genetic divergence can be used to trace the relatedness of organisms. Since orthologs are originated from speciation, they tend to preserve similar molecular and biological functions [5]. On the other hand, paralogs are originated from gene duplications and tend to deviate from their ancestral behavior and functions [6, 7]. However, orthologs are not necessarily identical in nature, as some orthologous genes can substantially diverge even among closely related organisms [8]. The converse argument is also valid – identically functioning genes are not necessarily orthologs [9, 10].

To date, considerable attentions have been drawn on developing methods to identify orthologs and paralogs and experimented using complex datasets [11, 12, 13, 14, 15, 4]. These methods can be primarily categorized into phylogenetic tree based approaches and sequence similarity based methods (also known as "graph-based" methods). Tree-based methods are the classical approach for orthology inference that seek to reconcile gene and species trees. In most cases gene and species trees have different topologies due to evolutionary events acting specifically on genes, such as duplications, losses, lateral transfers, or incomplete lineage sorting [16]. Goodman et al. [17] resolved these incongruences by explaining them in terms of speciation, duplication, and loss events on gene trees with respect to species trees. Therefore, orthology/paralogy inference can be reduced to gene tree and species tree reconciliation problem.

Sequence based approaches exploit the relative similarity between the orthologous genes. These methods are primarily based on BLAST search for finding the pairs having the highest sequence identity, as with Inparanoid [18], OrthoMCL [19], bidirectional best hit (BBH) [20], reciprocal smallest distance (RSD) [21] OrthoFinder c[22], morFeus [23],

Ortholog-Finder [24], OrthoVenn [25], PorthoMCL [26] etc. These methods mainly rely on the detection of bi-directional best hits (BBH), also known as the best reciprocal hits (BRH), in a sequence search. They are also known as "graph-based" techniques as they create a graph with genes as the vertices and pairwise scores as the edge weights and use graph based clustering techniques for finding the orthologs. Most of these algorithms also include a post-processing step involving various heuristics, and these heuristics and selection of parameters differentiate one from the other.

In recent years, a number of machine learning based algorithms have been proposed for the inference of orthologous relationships among genes. Galpart et al. [27] studied three pairs of related yeast species, *S. Cerevisiae – S. Pombe*, *S. Cerevisiae – C. Glabrata* and *S. Cerevisiae – K. Lactis*. They extracted various features (e.g., alignment scores, sequence length, membership to locally collinear blocks and physio-chemical properties), and trained Support Vector Machine (SVM) and Random Forest models to predict pairwise orthology. Sutphin and Mahoney presented a meta-tool, WORMHOLE, to integrate 17 distinct algorithms for orthology detection to identify least diverged orthologs (LDOs) among 6 eukaryotic species, namely, Human, Mouse, Zebrafish, Fruit-fly, Nematode, and Yeast [28]. They took the binary outputs of the 17 traditional orthology algorithm and performed an ensemble using a SVM model. Towfic *et al.* incorporated Protein-Protein interaction network in the orthology prediction pipeline and subsequently performed machine learning [29]. They extracted the data from the KEGG database [30]; however, their study was limited only to Human, Mouse, Fly and Yeast. They applied various machine learning algorithms, and the best performance was obtained by an ensemble of models which resulted in accuracies between $88.79 \pm 7.82\%$ for different pairs of species.

These three types of methods circumvent some of the challenging issues in orthology inference, but have limitations of their own. One of the major issue with the existing method is that they are prone to *precision-recall trade-off* (i.e., they fail to simultaneously achieve high precision and recall) [12, 13, 31]. Existing methods are computationally too demanding to analyze the ever-increasing number of genomes and thus often requires access to supercomputers [32, 4]. Some of the existing methods are difficult to apply on Eukaryotes [19], while some are restricted only to pairwise orthology detection [27, 20]. Tree-based methods suffer less from differential gene-loss and varying rates of evolution than BBH methods [33, 34], but genome-wide reconstruction of gene trees and species trees are computationally very demanding. Moreover, this approach is very sensitive to gene tree estimation error [35], and often performs worse than the sequence similarity based methods; it also requires manual curation [12, 36]. Sequence similarity based methods can overcome many of these issues and can perform well on closely related organisms [37]. However, very often they do not take evolutionary ortholog divergence into consideration. Due to this lack of evolutionary information, they mistakenly detect homoplasious paralogs as orthologs [38]. These methods may achieve relatively high precision albeit at the cost of a relatively low recall [20, 39, 40]. Moreover, BBH based methods can only account for one-to-one orthologs and may potentially miss true orthologs where one-to-many and many-to-many relationships are required to properly describe the orthology relationships [37, 4]. On the other hand, machine learning techniques can potentially lead to automatic and accurate orthology inference by utilizing the vast amount of genome data. However, existing ML based techniques have not been able to achieve this goal yet, and have only been validated on a small numbers of species.

In this study, we propose a novel method for orthologous gene clustering which potentially circumvents most of the challenging issues and idiosyncrasies of the existing techniques. Instead of using the widely used BLAST search, we treat this as a "text classification problem" and show that a Multinomial Naive Bayes algorithm with bag-of-words model is able to predict orthology relationships to a near perfect accuracy even when we are dealing with extremely unbalanced data with more than a million of genes on a diverse set of organisms. In particular, this paper makes the following contributions.

- We present NORTH, a machine learning based highly accurate and scalable orthologous gene clustering algorithm which does not require sophisticated parameter tuning and complicated feature engineering.
- We leveraged the biological insights of orthology inference, and exploited appropriate machine learning and natural language processing paradigms.
- We present a scalable implementation of NORTH with a customized in-house implementation of Naive Bayes algorithm, so that it does not require super computers or cluster computing for genome-wide comparisons.
- We conduct extensive experiments on over a million of genes from six major groups of life (Archaea, Bacteria, Animals, Fungi, Plants and Protists). To the best of our knowledge, this is the first attempt to conduct an study on the orthologs using machine learning, on such a large scale and with high accuracy, as the previous studies experimented on only a few organisms.
- We make available a server side application along with standalone desktop applications, allowing easy queries on finding probable orthologous genes.

2 From BLAST Search to Text Classification

Sequence-based methods rely on the assumption that the protein sequences obtained from orthologous genes tend to be quite similar since they are more conserved in the course of evolution [41]. These pipelines revolve mostly around a BLAST search [42], and often some parameter tuning and post-processing are performed for further refinement [43].

In this section, we briefly describe the general protocol of the BLAST search in a simplified manner and present a parallel to the Text Classification problem. The latter will subsequently allow us to coalesce the biological insights with Natural Language Processing (NLP) paradigm to aid us in solving our problem.

2.1 Overview of the BLAST Search Algorithm

BLAST (Basic Local Alignment Search Tool) divides the entire sequence into small words or k-mers, and searches the database for finding sequences that resemble the query sequence in terms of the k-mer distribution [44]. At first, the algorithm filters out the regions composed of only a few types of elements. These regions are called "Low Complexity Regions" as they may give high but insignificant scores and hamper the prediction. Next, the query sequence is broken into k-mers ($k = 3$ for protein and $k = 11$ for DNA

sequences). Then, all k-mers are listed together and their scores are computed using a substitution matrix (for example BLOSUM62 [45]), and only the k-mers with a score higher than a predefined threshold, T, are kept and organized in an efficient search tree. After that, the algorithm scans the database sequences for the extracted high-scoring k-mers, and the exact matches are used to seed a probable un-gapped alignment. The alignments are extended to the left and right of the seed points until the accumulated total score declines below a threshold. Next, the high-scoring segments whose scores are greater than a cutoff score, S, are listed and their statistical significance are computed. Some of the segments are also merged together in this step. Finally, the gapped alignment of the query sequence with all the database sequences are determined and the matches are compared using a predefined threshold parameter, E.

2.2 The Text Classification Problem

The text classification problem, also known as the document classification problem, is a well-studied problem in the field of Natural Language Processing (NLP) [46]. A traditional text classification pipeline starts with breaking the sentences into words using a proper tokenization scheme [47]. Next, various features are extracted from the tokenized texts [48], and a classifier is trained using these features to categorize the documents or texts into one of the predefined classes. The complexity of the classifier may range from simple Naive Bayes adopting the popular bag-of-words model [49], to state of the art Recurrent Neural Network (RNN) [50] coupled with Word Embeddings [51].

2.3 Orthologs Clustering through Text Classification

Since orthologous genes are less divergent, when a BLAST search is performed on one sequence, high scores (in terms of k-mer distribution) are obtained for its orthologous counterparts. Similarly, for text classification, a document is broken into a bag-of words and the most similar category is selected based on the distribution of words. Therefore, if we treat k-mers as words, these two approaches are quite similar. This was the initial enthusiasm for this work – replacing the BLAST search procedure with a standard text classification protocol. This approach elegantly coalesces the biological knowledge of orthology inference with the concepts of NLP.

3 Methods

3.1 Algorithmic Pipeline

NORTH takes a protein sequence as input like most other algorithms, and puts it in one of the predefined ortholog clusters using the following steps (please see Fig. 1 for an overview of the pipeline).

3.1.1 k-mer Construction

We break the input sequence into k-mers. Although the common practice is to take $k = 3$ for doing a BLAST on Amino Acid sequence, in our algorithm k is a tunable parameter.

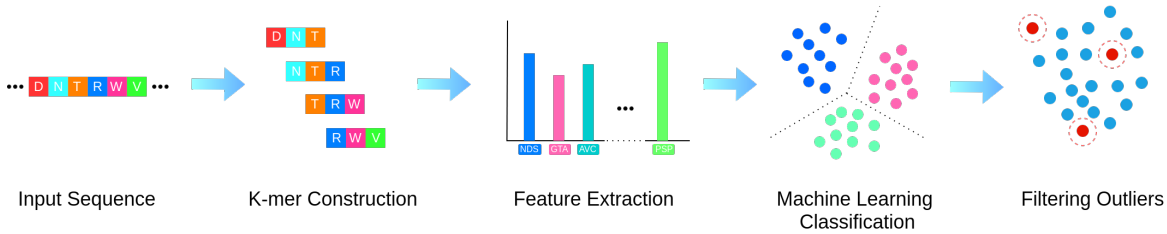


Figure 1: **Algorithmic pipeline of NORTH**. NORTH starts with breaking the input protein sequence of the gene into k-mers. It uses the k-mer frequencies as the features, and uses multinomial Naive Bayes classifier with the bag-of-words model to classify the input sequence into one of the predifined orthologous clusters. Finally, it reports whether the gene is a member of the predifined orthologous clusters or an outlier.

The intuition is that as we increase the value of k, the algorithm should capture more context. This will be beneficial if the data we are considering is diverse in nature. But at the same time, this may cause overfitting, compelling our algorithm to focus on some specific patterns. On the contrary, decreasing the value of k will dissipate significant details, increase bias and thus make our method more relaxed.

3.1.2 Feature Extraction

After breaking down the input protein sequence into k-mers, we compute the frequency distribution of the diversified k-mers. We use the k-mer frequencies as features for our algorithm, which is quite similar to the very popular bag-of-words model [52] widely used in Natural Language Processing (NLP). We do not resort to any other sophisticated feature engineering techniques.

Notably, although for the bag-of-words model in NLP *tf-idf* schemes are often used [53] to treat words based on their relative importance, such a scheme when applied in our context initially did not lead to good results (results not shown).

3.1.3 Machine Learning Classification

Every gene in an Ortholog Cluster is mutually orthologous to each other. Thus our objective is to map each sequence into one of the predefined Ortholog Clusters. As part of our attempt to coalesce the approaches of orthologous gene clustering and text classification, we use a Naive Bayes classifier to cluster the genes based on the computed features. In particular, we use the Multinomial Naive Bayes classifier [54] as it performed the best among all the different variants of Naive Bayes. However, it is worth to note that we experimented with other machine learning classifiers like Logistic Regression, Random Forest, Support Vector Machine (SVM) etc. But Naive Bayes outperformed all of them.

3.1.4 Filtering Outliers

NORTH, being a supervised classifier, is expected to infer a predefined number of orthologous clusters, leading to an erroneous decision for a gene outside these clusters. Hence, in

order to assert the reliability of the proposed algorithm, it is required to identify potential outlier genes, i.e., genes that are outside of the clusters under consideration.

Analysis of the pattern of the distribution of cluster probabilities obtained from the Naive Bayes algorithm reveals that in case of a valid clustering the probabilities follow a sharp, needle in a haystack pattern. On the contrary, while predicting the cluster of an outlier gene, the probability distribution is comparatively flat and scattered (please see Sec. 4.16). Therefore, comparing the pattern of cluster probabilities becomes a viable option for NORTH to identify outlier genes. In order to formulate the pattern of the cluster probabilities, we normalize all the probability values to the range $[0, 1]$. Then, various statistical attributes of the distribution, namely, mean, standard deviation, sum, skewness and kurtosis are extracted [55]. Using these as features, we train a Random Forest classifier, which is an ensemble of decision trees [56, 57], to infer whether a prediction is valid or represents an outlier gene.

3.2 Scalability

As we aim at analyzing a large number of orthologous clusters, we have made our pipeline scalable both in time and memory. The computation is divided automatically into a number of independent jobs. The subtasks are computationally less expensive and can also be performed in parallel. In a single computer environment, these jobs can be performed sequentially or in parallel on different threads. Moreover, our architecture is designed based on map-reduce framework, and thus it is possible to run NORTH in a distributed computing environment with much higher efficiency. More details of our scalable implementation can be found in the supplementary material SM1.

3.3 Implementation

We have implemented NORTH in Python [58]. We first used the Naive Bayes algorithm from the Scikit-Learn [59] library. But unfortunately it did not scale well, and as we increased the number of classes, the time and memory requirements became unmanageable. Therefore, we implemented our own scalable version of the Naive Bayes algorithm, which is highly optimized and scales with the number of classes and data, both in time and memory. All the codes are available at: `https://github.com/nibtehaz/NORTH`.

4 Results

The experiments were performed in a desktop computer with Ubuntu 16.04 operating system, Gigabyte B250M-Gaming 3 motherboard, Intel Core i7-7700 processor (3.6 GHz, 8MB cache), 8 GB 2400 rpm DDR4 RAM, 4 GB NVidia GTX 1050 Ti GPU, 6 Gbps SATA SSD.

4.1 Dataset

Among the ortholog databases, the KEGG [60, 30] database offers the most diverse collection of orthologous genes. These genes are arranged in a substantially large number of orthologous clusters. When most other databases available in the literature solely

depend on sequence similarity measures [61], KEGG database incorporates the molecular function of the genes as well. This manual curation makes the KEGG dataset superior over other databases obtained using a simple BLAST search based protocol [61]. However, a challenge in analyzing the KEGG dataset is that the clusters therein are quite sparse. In fact, these clusters are highly imbalanced, with some having a few thousands of genes while some with only a handful (see Fig. 2). To elaborate, we have a total of 7895 clusters among which, the biggest cluster contains 23,012 genes, whereas the smallest ones contain single gene only. The mean, median and standard deviation of the gene distributions in various clusters are 691, 220 and 1195 respectively indicating the presence of acute imbalance. Thus, we face a big challenge of training machine learning algorithm on imbalanced datasets [62].

The KEGG database is also conveniently divided into several partitions, namely, Plants, Fungi, Protists, Animals, Eukaryotes, Bacteria, Archaea, Prokaryotes and lastly a partition containing all the organisms (refered to as 'All'). This allowed us to test NORTH on different families of species and validate NORTH on diverse species families. We manually collected the ortholog tables from KEGG web browser [63], and then fetched the protein sequences from UniProt [64] using the UniProt REST API [65]. Unfortunately, a few genes listed in KEGG were not present in UniProt, which we had to exclude from our experiments. The overview of our dataset is depicted in Fig. 2a with the distributions of organisms and genes among various groups of life. In Fig. 2b, we have illustrated the data imbalance among the clusters. The gene population in the biggest clusters are scattered over large ranges of gene counts, and the majority of the remaining clusters contain only a handful of genes. This acute data imbalance prevented us from studying all the clusters and limited our investigation to only the biggest clusters.

4.2 Evaluation Metrics

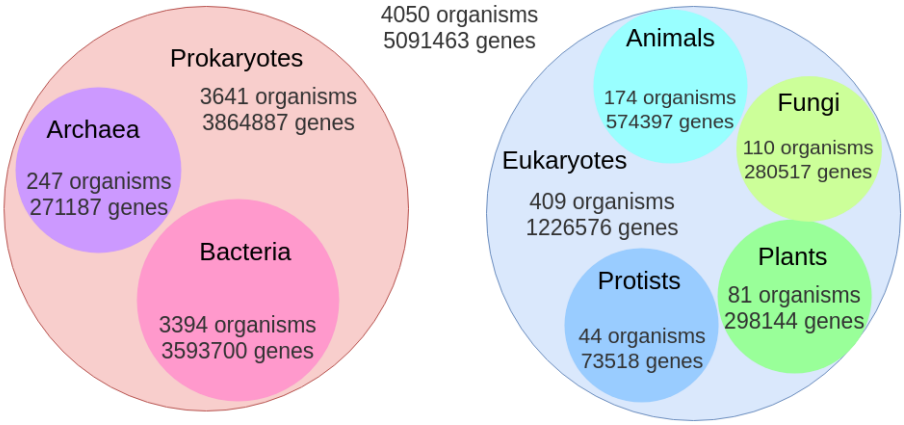
This problem is actually a multi-class classification problem with a large number of classes. By adopting the one-vs-all strategy for each of the classes, we can treat the classification output as a binary output. Thus, for each class C_i, we can define the following:

1. True Positive (TP) : If a sample of the class C_i is predicted to be of class C_i
2. False Positive (FP) : If a sample of class $C_j (i \neq j)$ is predicted to be of class C_i.
3. True Negative (TN) : If a sample of class $C_j (i \neq j)$ is predicted not to be of class C_i.
4. False Negative (FN) : If a sample of class C_i is predicted to be not of class C_i.

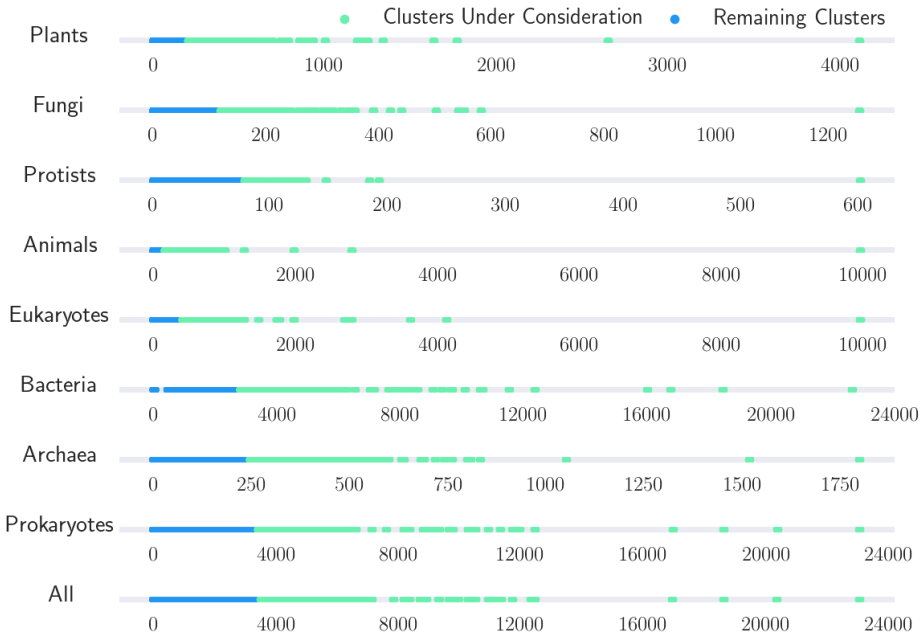
We have used the commonly used metrics namely, Precision, Recall and F_1 score in order to evaluate the performance of NORTH. These measures have been defined in the supplementary material SM1.

4.3 Experimental Setup

We have performed an extensive experimental study on the nine subgroups in KEGG database separately as well as on the combined dataset. The genes are distributed in



(a)



(b)

Figure 2: Overview of the dataset. (a) Distribution of organisms and genes in the nine sub-groups of life in KEGG database. (b) Distribution of gene population among different ortholog clusters, showing the number of genes in each of the clusters. We considered the biggest 250 clusters from each of the sub-groups (except for 'Protists' where we took only the biggest 40).

a large number of ortholog clusters, and these clusters are extremely imbalanced. Since the genes are distributed across a large number of imbalanced clusters, we started with the biggest 10 clusters, where the data imbalance is not too high. Next, we gradually increased the data imbalance by studying the biggest 50, biggest 100, biggest 150 and the biggest 250 clusters. After the biggest 250 clusters, the samples in the clusters reduces too drastically to conduct 10-fold cross validation properly. Hence, we were compelled to

stop adding any more clusters. However, for the Protists dataset, the class imbalance was the worst and we had to follow a different scheme as described later. Moreover, we varied the values of k (3 to 6) to evaluate how sensitive our approach is to various values of k. The results for all the experiments are explained in subsequent sections and congregated in Fig. 3.

Note that, we have chosen 10-Fold Cross Validation as experiments on real datasets justify 10-Fold to be the best Cross Validation scheme [66]. Since our dataset is adversely unbalanced, we followed the stratified variation of cross-validation, where all the labeled classes are uniformly balanced in all the splits [66]. Otherwise, due to random shuffling, some undersized classes could be absent from the training set.

4.4 Results on Plants

The Plants dataset comprises 81 organisms with 298144 genes. While considering only the biggest 10 clusters, NORTH was able to cluster the orthologs with nearly 100% accuracy, since the data is nearly balanced. The best performance is observed for $k = 4$ with Precision, Recall and F_1 score all being 99.75%. As we increase the number of clusters (which in turn introduces greater data imbalance), NORTH is still able to achieve very high accuracy (see Fig. 3A, and Table S1 in the supplementary material SM1). For the largest 250 clusters, NORTH obtained a Precision, Recall and F_1 score of 99.39%, for $k = 6$.

4.5 Results on Fungi

The Fungi dataset contains 280,517 genes across 110 organisms. Regardless of the number of clusters ($10 \sim 250$), NORTH consistently have achieved around 99% Precision, Recall and F_1 score. For most of the cases, $k = 5$ gives the best results, but other values of k do not affect the performance much. These results are reported in Fig. 3B (also see Table S2 in the supplementary material SM1).

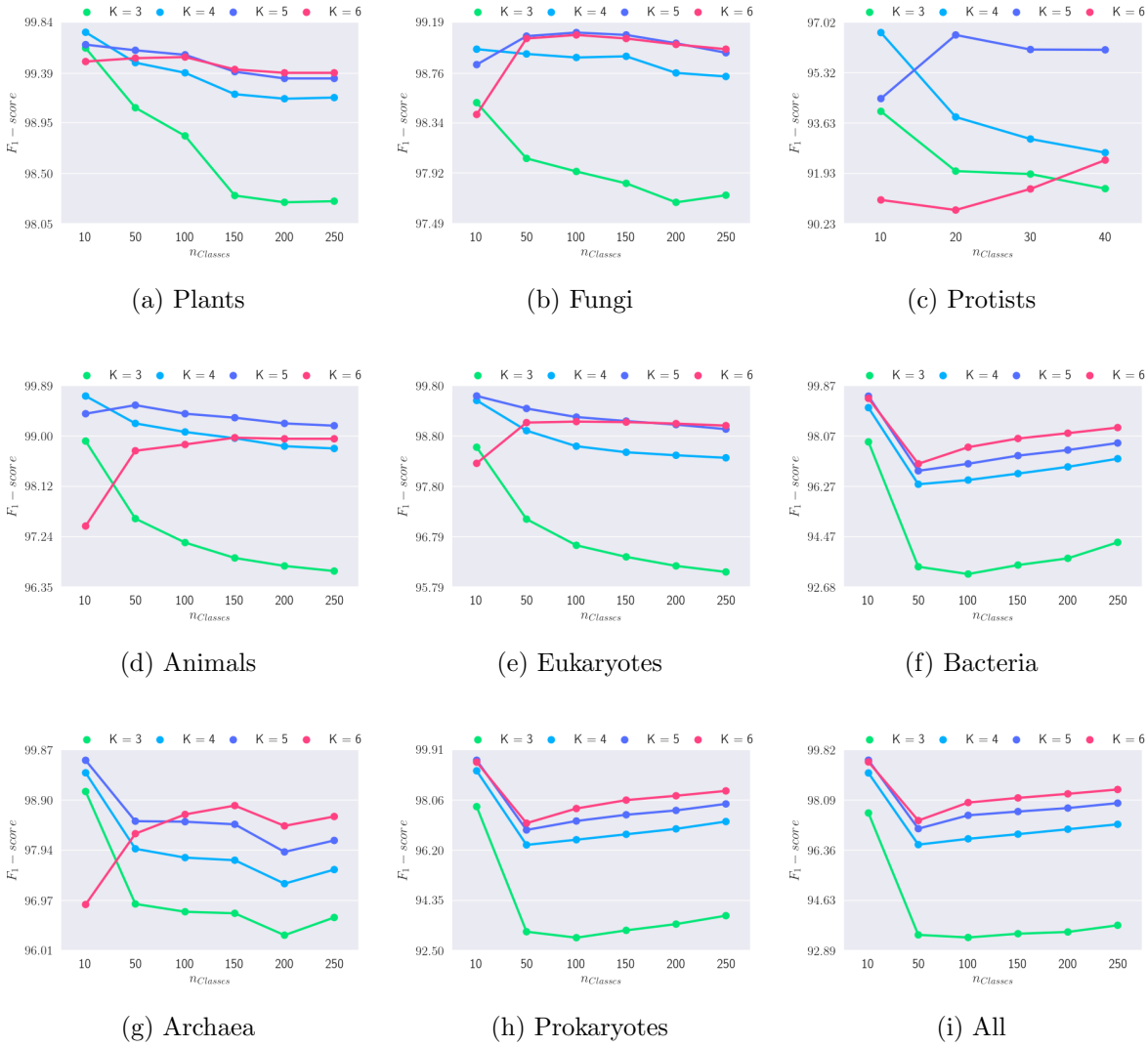
4.6 Results on Protists

The Protists dataset contains relatively small number of genes (73,518 genes) from 44 organisms. Largest 10 clusters contain only 1,876 genes and the largest 250 clusters contain only 15,364 genes. Since this small number of genes are distributed in a broad number of ortholog clusters, the number of genes in the individual clusters are too small (see Fig. 2). They are indeed so small that we could not consider more than 40 clusters.The best results have been obtained for $k = 5$, as illustrated in Fig. 3C and reported in Table S3 in SM1. Despite the lack of sufficient data, the Precision, Recall and F_1 scores obtained by NORTH remarkably lie within 96% - 97%.

4.7 Results on Animals

Animal dataset contains 574,397 genes across 174 organisms. NORTH works very well on this dataset regardless of the number of clusters, and consistently have obtained more than 99% Precision, Recall and F_1 score. Here, $k = 5$ gives the best performance in most

of the cases, and the performance metrics for the best performing models range from 99.18% to 99.70%. Fig. 3D and Table S4 in SM1 present these results.



(a) Plants (b) Fungi (c) Protists

(d) Animals (e) Eukaryotes (f) Bacteria

(g) Archaea (h) Prokaryotes (i) All

Figure 3: Performance of NORTH on various datasets in the KEGG database. We show results on the biggest 10, 50, 100, 150, 200 and 250 clusters (except for Protists where the number of genes become very small as we increase the number of clusters). We show only the F_1 score for varying numbers of ortholog clusters with different values of k (3 $\sim$ 6) as it is a balanced representation of both Precision and Recall (please refer to the supplementary material SM3 for an elaborated plot with the Precision and Recall values).

4.8 Results on Eukaryotes

The Eukaryotes dataset is a combination of the Plants, Fungi, Protists and Animals datasets, and many existing methods have been shown to have difficulty in analyzing Eukaryotes [19]. However, NORTH has been able to cluster the orthologous genes with outstanding accuracy. Notably, while NORTH's performance on Protists dataset was

slightly lower (i.e., in the range of $96\% - 97\%$), when we combine all the Eukaryotic genes together the overall accuracy improves. This apparent anomaly could be attributed to the fact that unlike the Protists dataset, where we had inadequate training data and that too with huge imbalance, in Eukaryotes dataset, the other sub-species datasets helped NORTH with large ortholog clusters as well as adequate training data. Here we can observe from Fig. 3E and Table S5 in SM1 that most of the best results are obtained for $k = 5$, and all the performance metrics (Precision, Recall and F_1 score) lie between 99.0% to 99.6%.

4.9 Results on Bacteria

The Bacteria dataset consists of 3,593,700 genes from 3394 organisms. Similar to other datasets, NORTH obtains Precision, Recall and F_1 score values of around 99.48% while considering only the biggest 10 clusters. Even for 250 clusters, NORTH was able to obtain over 98% precision, recall and F_1 score. The best results are obtained for $k = 6$, as opposed to $k = 5$ for most of the other sub-groups. This may indicate that there are certain potential longer patterns in the sequence in this dataset that help in distinguishing the orthologs better. The findings are illustrated in Fig. 3F and reported in Table S6 in SM1.

4.10 Results on Archaea

Archaea contains 271,187 genes from 247 organisms. With varying number of clusters the performance metrics lie within 98.40% to 99.66%. Similar to the Bacteria dataset, the best results are obtained for $k = 6$. Fig. 3G illustrate these results (please refer to Table S7 in SM1).

4.11 Results on Prokaryotes

The Prokaryotes dataset is the combination of the Bacteria and the Archaea dataset, and NORTH achieves similar accuracies (around $97\% \sim 99.5\%$) as it obtains on Bacteria and Archaea individually. The best results are obtained for $k = 6$ (as illustrated in Fig. 3H and supplementary Table S8). Unlike Eukaryotes dataset where NORTH consistently obtains more than 99% accuracy for various model conditions, combining Bacteria and Archaea into Prokaryotes do not lead to notable improvement. This may be attributed to the fact that horizontal gene transfer plays a vital role in the diversification of Bacterial and other Prokaryotic genomes [67], where the heredity materials come laterally (not vertically). Thus, not only the decision process becomes harder (by analyzing only the gene sequences), we may also need to study more patterns (as we have seen that, in most of the cases, the best performance for Bacteria, Archaea and Prokaryotes are obtained by using $k = 6$). However further biological study is needed to validate this hypothesis.

4.12 Results on All

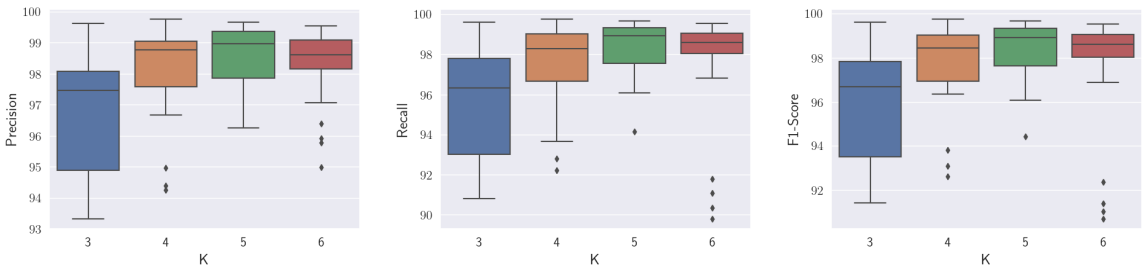
Finally, we evaluated NORTH on all the organisms together. It contains 1,255,877 genes from 3,880 organisms. The values of Precision, Recall and F_1 score are found to be

99.46%, 99.45%, and 99.45% respectively for the largest 10 clusters, and 98.48%, 98.43% and 98.44%, for the largest 250 clusters, showing that NORTH is able to cluster with very high accuracy even when we have more than a million of genes over thousands of distantly related organisms with a large number of imbalanced ortholog clusters (please refer to Fig. 3I and Supplementary Table S8). To the best of our knowledge, no other work (other than database compilation) has considered such a high number of orthologous genes from a wide range of diverse organisms.

4.13 Selection of K

In our proposed pipeline, we considered k to be a tunable parameter, which can be selected by analyzing the effect of changing the value of k. We varied the values of k, and observed that analyzing the data becomes computationally too expensive for $k > 6$ since we have to consider around 20^k k-mers (for instance, $20^7 = 1,280,000,000$ k-mers for $k = 7$). Although NORTH was able to handle it, the data preprocessing became exponentially complicated. Thus, we kept our experiments limited up to $k \leq 6$. It's worth noting that we conducted some experiments with $k = 7$, but those results closely resembled the results obtained for $k = 6$.

For most of the cases, the best results are obtained for $k = 5$ as illustrated in Fig. 4. The only exception is the presence of Prokaryotic genes where $k = 6$ gives us the best results as described earlier. However, $k = 5$ still holds up quite well in those cases, with a slight fall of accuracy (less than 0.5%). Thus, NORTH is robust and not much sensitive to the values of k, and considering the trade-off between computational requirement and accuracy, $k = 5$ appears to be a suitable option.



(a) Precision (b) Recall (c) F_1 score

Figure 4: Effect of varying values of k on various performance metrics. Here we have presented all the experimental results in box plots. By observing the values of Precision, Recall, F_1 score for different values of k, it can be noted that the best results are obtained for $k = 5$ in most of the cases. As we increase k from $k = 3$ the performance improves till $k = 5$, and then it starts to deteriorate slightly.

4.14 Class Imbalance

One of the difficulties in analyzing biological datasets is data imbalance [68]. Although machine learning based techniques may fail to give robust predictions in the presence of

data imbalance [62], our extensive experimental study establishes that NORTH is robust enough to handle data imbalance. When we consider the biggest 10 clusters, the data is reasonably balanced and NORTH is able to consistently classify the genes in the correct orthologous groups. The confusion matrix in Figure 5 shows that NORTH successfully clusters the genes with $99.39 \pm 0.52\%$ accuracy.

As we consider more ortholog clusters, due to the introduction of class imbalance, the accuracy declines slightly (less than 1%). For the biggest 250 clusters, NORTH obtains Precision = 98.48%, Recall = 98.43% and F_1 score=98.44%, which is quite remarkable considering the intensity of data imbalance. Apparently, it may be perceived that NORTH overfits the biggest clusters and thus obtains a high average accuracy. However, it is not the case as we have presented how the metrics are distributed over the individual clusters in Fig. 6. Note that, only a few clusters have around 8,000-24,000 genes, whereas the rest have less than $4,000$ genes each (see the distribution of genes across various clusters in Fig. 6(D)). It can be observed that for all but a few clusters the performance metrics are above 90%. Therefore, despite this significant data imbalance, NORTH is able to predict genes from even the smallest clusters accurately, exhibiting a balanced Precision and Recall.

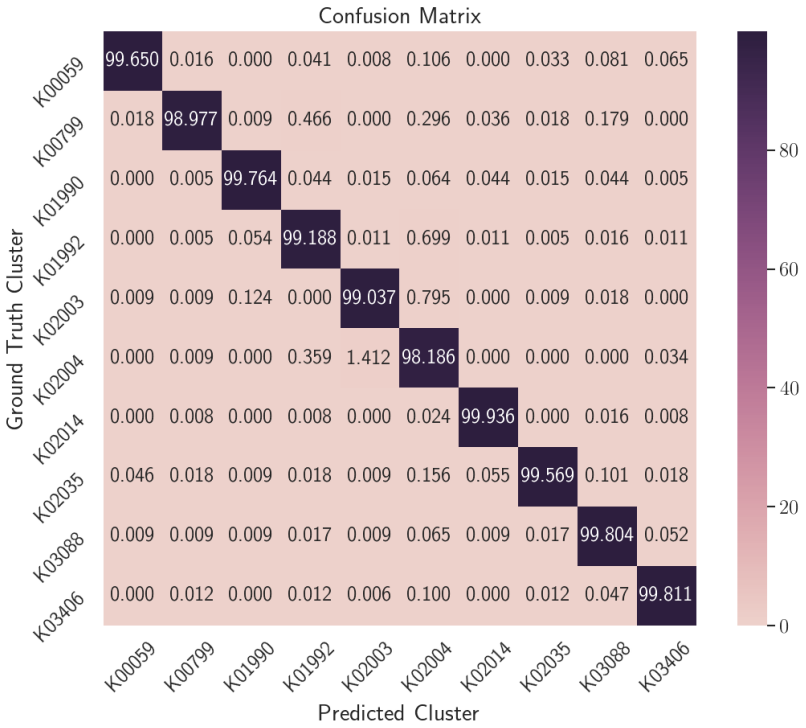


Figure 5: Confusion Matrix for a nearly balanced portion of the Dataset. Here, K00059, ..., K03406 denotes the KEGG cluster ids.

4.15 Independent Testing

In order to perform an independent testing (performance of NORTH on the unseen data from the predefined set of clusters), we considered the largest 250 clusters and from each

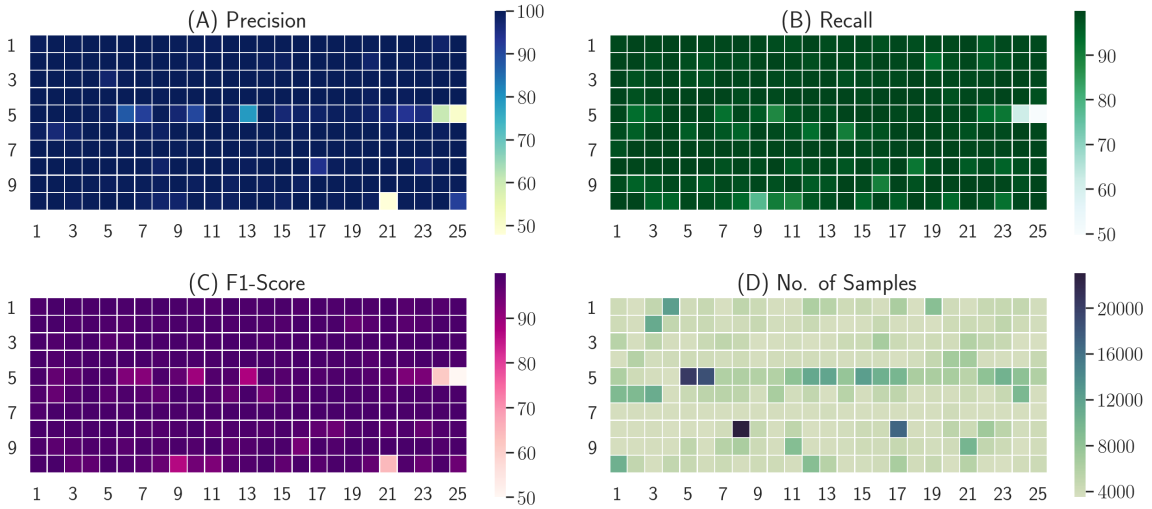


Figure 6: The performance of NORTH on highly imbalanced data. We show results on the largest 250 clusters comprising genes from all the six major groups of life. We report the Precision, Recall, and F_1 score. We also show the distribution of genes across different clusters to indicate the imbalance in the data. Here, the individual squares in the heatmaps correspond to different ortholog clusters (250 clusters are arranged in a two dimensional grid with 10 rows and 25 columns).

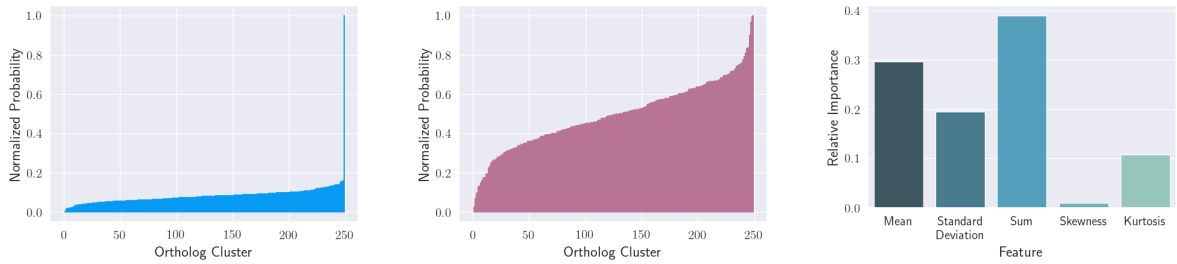
of the clusters, we took 10% of the total samples at random to form the independent testing dataset. This ensured adequate presence of genes from all the clusters in the test set. Using the rest of the data (i.e., training data in this context), we performed a stratified 10-fold cross validation again. Then, we used the best performing model (based on the F_1 score) to predict the clusters of the genes from the test set. We also performed an ensemble (majority voting) of all the ten models, that resulted from the stratified 10-fold cross validation, and observed the performance on the test set. The results are presented in Table 1. These results indicate the generality of NORTH, as the performance of the best model is similar to the ensemble of all the trained models, and the performance metrics are around 98%.

Table 1: Performance evaluation using independent testing. We considered the biggest 250 clusters and randomly selected 10% genes from each of the clusters as the independent testing dataset. The remaining data is used for training, and the best model (among 10 trained ones) is selected using 10-fold cross validation. The best model is then evaluated on the test data, and the Precision, Recall and F_1 score are reported. An ensemble of all the models is also performed.

Model	Precision	Recall	F_1 score
Best Model	98.15	97.81	97.90
Ensemble (majority voting)	98.14	97.82	97.91

4.16 Filtering Outliers

To assess the effectiveness of NORTH in identifying genes that do not belong to the predefined set of clusters (discussed in Sec. 3.1.4), we consider the biggest 250 clusters on all the organisms and train the Naive Bayes model. To introduce outliers, we consider all the remaining clusters with at least 3,000 genes. In this way, we obtain 1,255,877 member genes in the 250 clusters under consideration, and 648,938 genes from cluster not within the 250 predefined ones. As we can see from Fig. 7, the distribution of the cluster probabilities, predicted by the Naive Bayes model, is quite sparse for a predefined cluster (Fig. 7a) but dispersed for an outlier gene (Fig. 7b).



(a) (b) (c)

Figure 7: Filtering the outliers. (a) The cluster probability distribution, generated by the Naive Bayes model, for predefined clusters (a), (b) cluster probability distribution of the outlier genes. Figure 7c denotes the relative importance of the features. Despite 'sum' being the most prominent one, all the features are required for accurate and robust classification.

Next, we extract various features (statistical attributes of the distribution: mean, standard deviation, sum, skewness and kurtosis [55]), and train a Random Forest classifier comprising 10 trees. We perform a stratified 10-fold cross validation using the compiled genes. The task therefore reduces to a binary classification problem, and the results are presented in Table 2. It can be observed that the Precision and Recall values are quite balanced, and an overall F_1 score of 98.67% is obtained.

Table 2: Performance Evaluation of the Outlier Detection. Here we effectively classify whether the clustering predicted by the Naive Bayes model is a correct one, or represents an outlier. Therefore, the Precision, Recall and F_1 score of identifying an outlier or a valid clustering are reported.

Class	Precision (%)	Recall (%)	F_1 score (%)	Gene Count
Outlier	98.818	98.5365	98.6713	648938
Valid Clustering	99.250	99.390	99.319	1255877
Average	99.103	99.099	98.6713	1904815

Along with performing the classification, Random Forest also allows us to estimate the relative significance of various features [69, 70]. The relative importance of these features

is illustrated in Fig. 7c. It is apparent that the "sum" is the most prominent feature, as it directly harmonizes with the sparsity or dispersal of the probability distribution.

5 NORTH Web Application

A server-side application of NORTH has been developed to aid the researchers for discovering the probable orthologs of a newly sequenced gene. Since the entire algorithmic pipeline of NORTH is implemented in python, we developed the back-end of our application using Flask [71], a microframework for python, running on a lightweight guinicorn server [72]. The templates are rendered using Jinja 2 engine [73] and the responsive pages are designed using Materialize [74].The models are stored as pickle files and cluster information are saved in an SQL database, which are retrieved using AJAX calls. We have also developed cross-platform, native desktop applications using Electron JS framework [75].

The server application can be accessed from the project website (`https://nibtehaz.github.io/NORTH/`). Moreover, we provide the APIs to use our system programmatically. The desktop softwares can also be downloaded from the same link.

6 Conclusions

In this work, we have made an attempt at developing a highly accurate and automatic machine learning based pipeline to cluster orthologous genes together. Orthology identification is sufficiently complex and hence existing methods cannot fully comprehend or predict them. Thus the goal here should be the creation of an appropriate model itself; the model should account for as much genome data as possible (based on current availability), and continue improving by modifying itself to incorporate new data and scientific findings. This view emphasizes the importance of machine learning (ML) for orthology identification. ML based algorithms allow us to analyze a plethora of data, which is not possible manually otherwise. Moreover, these algorithms try to adaptively select the best set of parameters, and thus remove human biases and the necessity of manual intervention while setting various parameters. Therefore, machine learning algorithms are more promising for developing algorithms in comparative genomics due to the availability of a vast repertoire of gene sequences. However, the existing ML-based techniques mostly depend on sophisticated feature engineering, and are prone to precision-recall tradeoff and constrained to small numbers of organisms – making them difficult for handling large amount of complex biological data.

NORTH involves a novel combination of tools from natural language processing and machine learning with the biological basis of orthologs, and offers not only high accuracy which is otherwise quite hard to achieve without sophisticated manual intervention, but scalability that enables successful analyses of millions of genes on a wide range of distantly related organisms without having to rely on supercomputers. NORTH works satisfactorily even in a core i3 CPU with 4 GB of RAM. BLAST algorithm demands high processing memory, restricting the methods to limited numbers of sequences. Scalability and reduced running time of NORTH can be attributed to replacing the expensive BLAST search and

subsequent graph based algorithms, and our highly efficient and customized in-house implementation of the multinomial Naive Bayes algorithm.

One of the limitations of our system is that, due to the lack of sufficient data in individual clusters, we considered the largest 250 ortholog clusters from the KEGG database. However, as we have shown in Sec. 3.1.4, NORTH can identify genes that do not belong to the 250 predefined clusters with high precision and recall. We will continue to evolve and improve our system as more genes become available. Another limitation of NORTH is that it is currently not suitable for pairwise orthology prediction. Although a clustering algorithm can also infer the pairwise relationship, the one-to-one, one-to-many, and many-to-many varients of orthologs need to be considered carefully as they may raise false negatives or false positives. We left this as a future work.

We conducted an extensive simulation study using data from KEGG databse. However, we could not compare our method to others for various reasons as discussed in [12, 15, 14]. Conceptual differences among various orthology inference methods and databases make the issue of verifying and benchmarking orthology predictions very difficult if not impossible. Comparison becomes even more difficult due to the lack of benchmark datasets. Fortunately, a community effort (Quest for Orthologs consortium) has been started to provide a mean for benchmarking orthology prediction algorithms [40]. However, this system is developed to evaluate the pairwise orthology prediction algorithms, making it unsuitable for a comparison with NORTH, which is a ortholog clustering algorithm. Moreover, a benchmarking system would actually reflect upon the effectiveness of the KEGG database, instead of our algorithm since NORTH is trained on KEGG database. This is a common problem affliated with supervised machine learning, where we try to emulate the labelling of the training data [76].

On an ending note, we have demonstrated that an appropriate bag-of-words model with a Naive Bayes classifier can successfully cluster the orthologous genes together. We report on an extensive experimental study and demonstrated the high accuracy and scalability of NORTH. Thus, NORTH can be considered as a potential alternative to the typical phylogenetic tree based or sequence based methods. We believe NORTH will evolve with the availability of new data, and in response to scientific findings and systamatists' feedback – laying a firm, broad foundation for fully automated, highly accurate and scalable orthology identification.

Supplementary information

Supplementary Material SM1: overview of the scalable implementation of multinomial Naive Bayes algorithm in NORTH, and summary of the experimental results.
Supplementary Material SM2: detailed results showing the performance metrics for each of the individual ortholog clusters under various model conditions.
Supplementary Material SM3: an illustration of the experimental results including precision, recall and F_1-score.

Author contributions

MSB conceived the study and NI helped design the study; NI, SA, BS, MSB and MSR developed the methods; NI implemented the methods; NI, SA, and BS conducted the experiments; NI, MSB and MSR analyzed and interpreted the results; MSB and MSR supervised the study; NI and MSB wrote the first draft and all the authors took part in finalizing the manuscript.

Competing interests

The authors declare that they have no competing interests.

Data availability

Data used in this paper are available at KEGG (https://www.genome.jp/kegg/), and UniProt (https://www.uniprot.org/) databases.

Code availability

The software is freely available as open source code at `https://github.com/nibtehaz/NORTH`.

References

[1] Walter M Fitch. Homology: a personal view on some of the problems. *Trends in genetics*, 16(5):227–231, 2000.

[2] Walter M Fitch. Distinguishing homologous from analogous proteins. *Systematic zoology*, 19(2):99–113, 1970.

[3] DP Wall, HB Fraser, and AE Hirsh. Detecting putative orthologs. *Bioinformatics*, 19(13):1710–1711, 2003.

[4] Fredj Tekaia. Inferring orthologs: open questions and perspectives. *Genomics Insights*, 9:GEI–S37925, 2016.

[5] Mark E Peterson, Feng Chen, Jeffery G Saven, David S Roos, Patricia C Babbitt, and Andrej Sali. Evolutionary constraints on structural similarity in orthologs and paralogs. *Protein Science*, 18(6):1306–1315, 2009.

[6] Michael Lynch and Vaishali Katju. The altered evolutionary trajectories of gene duplicates. *TRENDS in Genetics*, 20(11):544–549, 2004.

[7] Susumu Ohno, Ulrich Wolf, and Niels B Atkin. Evolution from fish to mammals by gene duplication. *Hereditas*, 59(1):169–187, 1968.

[8] Rafael Díaz, Carmen Vargas-Lagunas, Miguel Angel Villalobos, Humberto Peralta, Yolanda Mora, Sergio Encarnación, Lourdes Girard, and Jaime Mora. argc orthologs from rhizobiales show diverse profiles of transcriptional efficiency and functionality in sinorhizobium meliloti. *Journal of bacteriology*, 193(2):460–472, 2011.

[9] Marina V Omelchenko, Michael Y Galperin, Yuri I Wolf, and Eugene V Koonin. Non-homologous isofunctional enzymes: a systematic analysis of alternative solutions in enzyme evolution. *Biology direct*, 5(1):31, 2010.

[10] Andreas Henschel, Wan Kyu Kim, and Michael Schroeder. Equivalent binding sites reveal convergently evolved interaction motifs. *Bioinformatics*, 22(5):550–555, 2005.

[11] Eugene V Koonin. Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, 39:309–338, 2005.

[12] Tim Hulsen, Martijn A Huynen, Jacob de Vlieg, and Peter MA Groenen. Benchmarking ortholog identification methods using functional genomics data. *Genome biology*, 7(4):R31, 2006.

[13] Feng Chen, Aaron J Mackey, Jeroen K Vermunt, and David S Roos. Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PloS one*, 2(4):e383, 2007.

[14] Kalliopi Trachana, Tomas A Larsson, Sean Powell, Wei-Hua Chen, Tobias Doerks, Jean Muller, and Peer Bork. Orthology prediction methods: a quality assessment using curated protein families. *Bioessays*, 33(10):769–780, 2011.

[15] Adrian M Altenhoff and Christophe Dessimoz. Inferring orthology and paralogy. In *Evolutionary genomics*, pages 259–279. Springer, 2012.

[16] M Nei. Molecular evolutionary genetics columbia university press new york google scholar. 1987.

[17] Morris Goodman, John Czelusniak, G William Moore, Alejo E Romero-Herrera, and Genji Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Biology*, 28(2):132–163, 1979.

[18] Kevin P O'brien, Maido Remm, and Erik LL Sonnhammer. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic acids research*, 33(suppl_1):D476–D480, 2005.

[19] Li Li, Christian J Stoeckert, and David S Roos. Orthomcl: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–2189, 2003.

[20] Yuri I Wolf and Eugene V Koonin. A tight link between orthologs and bidirectional best hits in bacterial and archaeal genomes. *Genome biology and evolution*, 4(12):1286–1294, 2012.

[21] Dennis P Wall and Todd DeLuca. Ortholog detection using the reciprocal smallest distance algorithm. In *Comparative genomics*, pages 95–110. Springer, 2007.

[22] David M Emms and Steven Kelly. Orthofinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome biology*, 16(1):157, 2015.

[23] Ines Wagner, Michael Volkmer, Malvika Sharan, Jose M Villaveces, Felix Oswald, Vineeth Surendranath, and Bianca H Habermann. morfeus: a web-based program to detect remotely conserved orthologs using symmetrical best hits and orthology network scoring. *BMC bioinformatics*, 15(1):263, 2014.

[24] Tokumasa Horiike, Ryoichi Minai, Daisuke Miyata, Yoji Nakamura, and Yoshio Tateno. Ortholog-finder: a tool for constructing an ortholog data set. *Genome biology and evolution*, 8(2):446–457, 2016.

[25] Yi Wang, Devin Coleman-Derr, Guoping Chen, and Yong Q Gu. Orthovenn: a web server for genome wide comparison and annotation of orthologous clusters across multiple species. *Nucleic acids research*, 43(W1):W78–W84, 2015.

[26] Ehsan Tabari and Zhengchang Su. Porthomcl: parallel orthology prediction using mcl for the realm of massive genome availability. *Big Data Analytics*, 2(1):4, 2017.

[27] Deborah Galpert, Sara del Río, Francisco Herrera, Evys Ancede-Gallardo, Agostinho Antunes, and Guillermin Agüero-Chapin. An effective big data supervised imbalanced classification approach for ortholog detection in related yeast species. *BioMed research international*, 2015, 2015.

[28] George L Sutphin, J Matthew Mahoney, Keith Sheppard, David O Walton, and Ron Korstanje. Wormhole: novel least diverged ortholog prediction through machine learning. *PLoS computational biology*, 12(11):e1005182, 2016.

[29] Fadi Towfic, Susan VanderPIas, Casey A OIiver, OIiver Couture, Christopher K TuggIe, M Heather West GreenIee, and Vasant Honavar. Detection of gene orthology from gene co-expression and protein interaction networks. In *BMC bioinformatics*, volume 11, page S7. BioMed Central, 2010.

[30] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. Kegg as a reference resource for gene and protein annotation. *Nucleic acids research*, 44(D1):D457–D462, 2015.

[31] Kyung Mo Kim, Samsun Sung, Gustavo Caetano-Anolles, Jae Yong Han, and Heebal Kim. An approach of orthology detection from homologous sequences under minimum evolution. *Nucleic acids research*, 36(17):e110–e110, 2008.

[32] Marcus Lechner, Sven Findeiß, Lydia Steiner, Manja Marz, Peter F Stadler, and Sonja J Prohaska. Proteinortho: detection of (co-) orthologs in large-scale analysis. *BMC bioinformatics*, 12(1):124, 2011.

[33] Joseph W Thornton and Rob DeSalle. Gene family evolution and homology: genomics meets phylogenetics. *Annual review of genomics and human genetics*, 1(1):41–73, 2000.

[34] Joanna C Chiu, Ernest K Lee, Mary G Egan, Indra Neil Sarkar, Gloria M Coruzzi, and Rob DeSalle. Orthologid: automation of genome-scale ortholog identification within a parsimony framework. *Bioinformatics*, 22(6):699–707, 2006.

[35] Matthew D Rasmussen and Manolis Kellis. Accurate gene-tree reconstruction by learning gene-and species-specific substitution rates across multiple complete genomes. *Genome research*, 17(12):000–000, 2007.

[36] Heng Li, Avril Coghlan, Jue Ruan, Lachlan James Coin, Jean-Karim Heriche, Lara Osmotherly, Ruiqiang Li, Tao Liu, Zhang Zhang, Lars Bolund, et al. Treefam: a curated database of phylogenetic trees of animal gene families. *Nucleic acids research*, 34(suppl_1):D572–D580, 2006.

[37] Toni Gabaldón. Large-scale assignment of orthology: back to phylogenetics? *Genome biology*, 9(10):235, 2008.

[38] Liisa B Koski and G Brian Golding. The closest blast hit is often not the nearest neighbor. *Journal of molecular evolution*, 52(6):540–542, 2001.

[39] Daniel A Dalquen and Christophe Dessimoz. Bidirectional best hits miss many orthologs in duplication-rich clades such as plants and animals. *Genome biology and evolution*, 5(10):1800–1806, 2013.

[40] Adrian M Altenhoff, Brigitte Boeckmann, Salvador Capella-Gutierrez, Daniel A Dalquen, Todd DeLuca, Kristoffer Forslund, Jaime Huerta-Cepas, Benjamin Linard, Cécile Pereira, Leszek P Pryszcz, et al. Standardized benchmarking in the quest for orthologs. *Nature methods*, 13(5):425, 2016.

[41] David M Kristensen, Yuri I Wolf, Arcady R Mushegian, and Eugene V Koonin. Computational methods for gene orthology inference. *Briefings in bioinformatics*, 12(5):379–391, 2011.

[42] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[43] Alexander CJ Roth, Gaston H Gonnet, and Christophe Dessimoz. Algorithm of oma for large-scale orthology inference. *BMC bioinformatics*, 9(1):518, 2008.

[44] W Mount David. Bioinformatics–sequence and genome analysis. *CSHL, New York*, pages 75–85, 2001.

[45] Mark P Styczynski, Kyle L Jensen, Isidore Rigoutsos, and Gregory Stephanopoulos. Blosum62 miscalculations improve search performance. *Nature biotechnology*, 26(3):274, 2008.

[46] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[47] Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112, 2014.

[48] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

[49] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[50] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[51] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[52] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[53] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142, 2003.

[54] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer, 2004.

[55] D Middleton. Mathematical statistics and data analysis, by john a. rice. pp 595.1988. isbn 0-534-08247-5 (wadsworth & brooks/cole). *The Mathematical Gazette*, 72(462):330–331, 1988.

[56] Tin Kam Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[57] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[58] Guido Van Rossum et al. Python programming language. In *USENIX Annual Technical Conference*, volume 41, page 36, 2007.

[59] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[60] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[61] Andrey Alexeyenko, Julia Lindberg, Åsa Pérez-Bercoff, and Erik LL Sonnhammer. Overview and comparison of ortholog databases. *Drug Discovery Today: Technologies*, 3(2):137–143, 2006.

[62] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[63] Kegg genome browser. `https://www.genome.jp/kegg/ko.html`, (Last accessed on December 20, 2018).

[64] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl_1):D115–D119, 2004.

[65] Uniprot. `https://www.uniprot.org/help/api`, (Last accessed on December 20, 2018).

[66] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[67] Howard Ochman, Jeffrey G Lawrence, and Eduardo A Groisman. Lateral gene transfer and the nature of bacterial innovation. *nature*, 405(6784):299, 2000.

[68] Xing-Ming Zhao, Xin Li, Luonan Chen, and Kazuyuki Aihara. Protein classification with imbalanced data. *Proteins: Structure, function, and bioinformatics*, 70(4):1125–1132, 2008.

[69] Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[70] Bjoern H Menze, B Michael Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich, and Fred A Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):213, 2009.

[71] Flask. `http://flask.pocoo.org/`, (Last accessed on December 20, 2018).

[72] Gunicorn. `https://gunicorn.org/`, (Last accessed on December 20, 2018).

[73] Jinja. `http://jinja.pocoo.org/`, (Last accessed on December 20, 2018).

[74] Materialize. `https://materializecss.com/`, (Last accessed on December 20, 2018).

[75] Electron js. `https://electronjs.org/`, (Last accessed on December 20, 2018).

[76] Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.