Subject Section

Detecting and correcting misclassified sequences in the large-scale public databases

Hamid Bagheri¹.*, Andrew Severin² and Hridesh Rajan ¹

¹Dept. of Computer Science, Iowa State University, Ames, 50011, USA and
²Genome Informatics Facility, Iowa State University, Ames, 50011, USA.

* To whom correspondence should be addressed.
Associate Editor: XXXXXXX
Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: As the cost of sequencing decreases, the amount of data being deposited into public repositories is increasing rapidly. Public databases rely on the user to provide metadata for each submission that is prone to user error. Unfortunately, most public databases, such as non-redundant (NR), rely on user input and do not have methods for identifying errors in the provided metadata, leading to the potential for error propagation. Previous research on a small subset of the non-redundant (NR) database analyzed misclassification based on sequence similarity. To the best of our knowledge, the amount of misclassification in the entire database has not been quantified. We propose a heuristic method to detect potentially misclassified taxonomic assignments in the NR database. We applied a curation technique and quality control to find the most probable taxonomic assignment. Our method incorporates provenance and frequency of each annotation from manually and computationally created databases and clustering information at 95% similarity.

Results: We found more than 2 million potentially taxonomically misclassified proteins in the NR database. Using simulated data, we show a high precision of 97% and a recall of 87% for detecting taxonomically misclassified proteins. The proposed approach and findings could also be applied to other databases.

Availability: Source code, dataset, documentation, Jupyter notebooks, and Docker container are available at https://github.com/boalang/nr.
Contact: hbagheri@iastate.edu
Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Researchers use BLAST on the non-redundant (NR) database (NCBI (2018b)) on a daily basis to identify the source and function of a protein/DNA sequence. The non-redundant (NR) database encompasses protein sequences from non-curated (low quality) and curated (high quality) databases. It contains non-redundant sequences from GenBank translations (i.e. GenPept) together with sequences from other databases (Refseq (Pruitt et al. (2006)), PDB (Berman et al. (2003)), SwissProt (Boeckmann et al. (2003)), PIR (Wu et al. (2003)) and PRF). NR removes 100% identical sequences and merges the annotations and sequence IDs.

We have identified three root causes for annotation errors in the public databases: user metadata submission, contamination error in the biological samples, and computational methods. NCBI relies on the accuracy of the metadata provided by researchers that are depositing sequencing data into the database. Data are deposited into NCBI into Biosamples and Bioprojects as raw data, genome assemblies, and transcriptomes. Biosamples contain metadata describing the data type, scope, organism, publication, authors, and attributes, which include cultivar, biomaterial provider, collection date, tissue, developmental stage, geographical location, coordinates, and additional notes. This metadata is then propagated to the sequences that are deposited. For example, if data for DNA sequences were deposited by a plant researcher studying soybeans obtained from a soybean roots, then all sequences tied to that metadata will be labeled with the organism name Glycine max. If the researcher had in fact been working on Glycine soja then this would result in a misassignment of all Glycine max sequences.

The second key challenge that all large databases have is the issue of contamination (Schnoes et al. (2009)). For example, if the aforementioned...
hypotheoretical soybean research did not remove the soybean root nodules during sample processing, then the tissue sample could also contain DNA from Nitrogen fixing soil bacteria that infect nodules leading to contamination of the sequences and ultimately the sequence database. NCBI is aware of the potential for contamination in sequence databases and describes potential sources of contamination that include: DNA recombination techniques (vectors, adaptors, linkers and PCR primers, transposon, and insertion sequences) and sample impurities (organelle, DNA/RNA, multiple organisms). NCBI encourages the use of programs to try to reduce issues with contamination. Specifically, they recommend screening for contamination using VecScreen (vec (2019)) and BLAST for the sequences used in sequencing library preparation. More broadly, they recommend BLAST to screen out bacterial, yeast, and *Escherichia coli* sequences and BLASTing against the NR database to identify potential contaminating sequences. Unfortunately, despite efforts to reduce contamination, sequences still end up in the NR database that is incorrectly taxonomically classified. This can limit our ability to identify contamination of future sequence submissions, as BLASTing against the database could propagate these types of errors as the database grows in size (Schnoes et al. (2009)). The contamination problem is not unique to NCBI but can be found in all large databases. A large-scale study of complete and draft bacterial and archea genomes in the NCBI RefSeq database revealed that 2250 genomes are contaminated by human sequences (Breitwieser et al. (2019)). Breitwieser et al. reported 3437 spurious protein entries that are currently present in NR and TIGER protein databases.

The third key challenge is that there are errors in the annotations due to the computational error in tools that are based on homology to existing sequences to predict the annotations (Schnoes et al. (2009)). These errors have caused annotation accuracy and database quality issues over the years. Annotation errors are not limited to contamination or computationally predicted one. For instance, there exists evidence that suggests some of the experimentally derived annotations may be incorrect (Schnoes et al. (2009)). Therefore, it will be beneficial for researchers to utilize a quality control method to detect misclassified sequences and propose the most probable taxonomic assignment.

To address these well-known problems, there are two approaches in the literature: phylogenetic-based approach and functional approach. For the first approach, Koziot (Koziot et al. (2016)) has proposed a phylogeny-aware method to detect and correct misclassified sequences in public databases. They utilized the Evolutionary Placement Algorithm (EPA) to identify mislabeled taxonomic annotation. Edgar (Edgar (2018)) has studied taxonomy annotation error in rRNA databases. They showed that the annotation error rate in SILVA and Greengenes databases is about 17%. They also used the phylogenetic-based approach.

In the second approach, it is a common technique for quality control and data cleaning to utilize domain knowledge in the form of ontologies (Chu et al. (2015)). Gene Ontology (GO) (Ashburner et al. (2000)) has been suggested to infer aspects of protein function based on sequence similarity (Holliday et al. (2017)). The MisPred Nagy and Pathy (2013) and FixPred (Nagy and Pathy (2013)) programs are used to address the identification and correction of misclassified sequences in the public databases. The FixPred and MisPred methods are based on the principle that an annotation is likely to be erroneous if its feature violates our knowledge about proteins (Nagy et al. (2008)). MisPred (Nagy and Pathy (2013)) is a tool developed to detect incomplete, abnormal, or mispredicted protein annotations. There is a web interface to check the protein sequence online. MisPred uses protein-coding genes and protein knowledge to detect erroneous annotations at the protein function level. For example, they have found for a subset of protein databases that violation of domain integrity accounts for the majority of mispredictions. Modha et al. have proposed a pipeline to pinpoint taxonomic error as well as identifying novel viral species (Modha et al. (2018)). There is another web-server for exploratory analysis and quality control of proteome-wide sequence search (Medlar et al. (2018)) that requires a protein sequence in a FASTA format. European Bioinformatics Institute (EMBL-EBI) developed InterPro (InterPro (2019)) to classify protein sequences at the superfAMILY, family and subfamily levels. UniProt has also developed two prediction systems, UniRule and the Statistical Automatic Annotation System (SAAS) (SAAS (2019)), to annotate UniProtKB/TIGER, protein database automatically. CDD is a Conserved Domain Database for the functional annotation of proteins (Marchler-Bauer et al. (2010)).

Exploring public sequence databases and curating annotations at large-scale is challenging. Previous research on the NR database focused on a small subset of the NR database and analyzed annotation error due to the computational requirements. There has been a study (Schnoes et al. (2009)) on misclassification levels for molecular function for a model set of 37 enzyme families. To the best of our knowledge, the amount of misclassification in the entire database has not been well quantified. Here, we attempt to address these limitations in detecting and correcting annotations at large-scale and make the following contributions:

- We utilize a genomics-specific language, BoaG, that uses the Hadoop genomics libraries to detect misclassified sequences and BLASTing against the NR database that is not available in other works.
- We also present a heuristic-based method to detect misclassified taxonomic assignments in the NR database that is low-cost and easy to use. We automatically generate a phylogenetic tree from a list of taxonomic assignments and use the tree, along with frequency, the provenance (database of origin) of each taxonomic annotation, and clustering information from NR at 95% similarity to identify potential misclassification and propose the most probable taxonomic assignment.
- The technique proposed in this work could be generalized to apply to other public databases and different kinds of annotations like protein functions. In this work, we address the taxonomic annotation error in protein databases. We also tested our approach on the RNA dataset introduced in the literature.

We have identified “29,175,336” proteins in the NR database that have more than one distinct taxonomic assignments, among which “2,238,230” (7.6%) are potentially taxonomically misclassified. We also found that the total number of potential misclassifications in clusters at 95% similarity, above the genus level, is “3,689,089” out of 88M clusters, which are 4% of the total clusters. This percentage of misclassifications in NR has a significant impact due to the potential for error propagation in the downstream analysis (Mukherjee et al. (2015)).

The rest of the paper is organized as follows. In Section 2, we present methods and materials for dataset generation and our approach. In section 3, we discuss the results of taxonomically misclassified proteins within sequences and in NR 95%. We also present the correcting approach for detected sequences. In Section 4, we conclude with suggestions for the future.

### 2 Materials and methods

In this section, we will describe the overview architecture of our detection and correction approach. Then, we describe the dataset generation and how we generate a phylogenetic tree from taxonomic assignments. Next, we discuss our detection algorithm to find misclassified sequences. Then, we describe our approach to propose taxonomic assignments for the sequences identified as misclassified. Finally, we will describe the sensitivity analysis on changing the different parameters to propose the taxonomic assignments.
To describe our dataset, let

\[ D = \{ P, C, \tau, f \}, \quad \text{where} \quad P = \{ P_1, P_2, \ldots, P_n \}, \quad C = \{ C_1, C_2, \ldots, C_k \}, \quad \tau \text{ is a set of functions in the NR database.} \]

\[ \tau(P_i) = \{ A_{i,1}, A_{i,2}, \ldots, A_{i,\tilde{r}} \}, \quad j \in [1, \tilde{r}] \]

2.1 An overview of the method

Figure 1 shows an overview of our approach. The NCBI NR database files were downloaded from NCBI on Oct 20, 2018. The raw data was obtained from XML files downloaded from the NCBI ftp server (NCBI (2018b)). CD-HIT (Fu et al. (2012)) (version v4.6.8-2017-1208) was used to cluster NR protein sequences into clusters at 95% similarity using the following parameters (-n 5 -g 0 -aS 0.8 -d 0 -p 1 -T 28 -M 0). These parameters use a word length of 5 and require that the alignment of the short sequences is at least 80% of its length. The data acquisition, preprocessing, and clustering took about 3 days. The detection and correction part took about 8 hours.

We took the NR protein FASTA files that have the definition lines containing annotations from different databases and generate the BoaG format that took about 2 hours. Each definition line in the raw data includes protein ID, protein name followed by an organism name in square brackets, e.g., ">AAB18559 unnamed protein product [Escherichia coli sub. K-12 substr. MG1655]". BoaG is a domain-specific language that uses a Hadoop based infrastructure for biological data (Bagheri et al. (2019)). A BoaG program is submitted to the BoaG infrastructure. It is compiled and executed on a distributed Hadoop cluster to execute a query on the BoaG-formatted database of the raw data. BoaG has aggregators that can be run on the entire database or a subset of the database taking advantage of protobuf-based schema design optimized for a Hadoop cluster for both the data and the computation. These aggregators are similar to but not limited to aggregators traditionally found in SQL databases and NoSQL databases like MongoDB. A BoaG script requires fewer lines of code, provides storage efficiency, and automatically parallelized large-scale analysis.

2.1.1 Dataset generation

To describe our dataset, let \( D \) denotes the protein and clustering dataset in our study: \( D = \{ P, C, \tau, f \} \). Here, \( P = \{ P_1, P_2, \ldots, P_n \} \) is a set of all the proteins in the NR database. \( C = \{ C_1, C_2, \ldots, C_k \} \) represents a set of all clusters at 95% similarity. \( |P| \) and \( |C| \) in our dataset are about 174M and 88M respectively. \( \tau \) is a set of taxonomic assignment for proteins, and \( f \) is a set of functions in the NR database. In this work, we focus on exploring taxonomic assignments.

**Definition 1. Cluster.** We define cluster as a set of protein sequences such that their sequence are 95% similar and their sequence length is 80% similar. Every particular cluster, \( C_j \), has \( k \) members:

\[ C_j = \{ P_1, P_2, \ldots, P_k \}, \quad \text{and} \quad k \in [1, |P|] \]  

(1)

In Definition 1, each protein sequence belongs to exactly one cluster at 95% similarity, and each cluster has one representative sequence. If a protein is not identical in sequence and length, it will fall into a cluster with no other member.

2.1.2 Generating phylogenetic tree from taxonomic assignments

We get the list of taxonomic assignments that originate from different databases (manually reviewed and computationally created) and build a phylogenetic tree by utilizing the ETE3 library (Huerta-Cepas et al. (2016)). This library utilizes the NCBI taxonomy database that is updated frequently.

**Definition 2. Annotation List.** Each phylogeny tree is associated with one particular protein, \( P_i \), and has the set of taxonomic assignments that originate from different databases. Here, \( A_{i,j} \) denotes annotation number \( j \) for protein \( P_i \):

\[ r(P_i) = \{ A_{i,1}, A_{i,2}, \ldots, A_{i,\tilde{r}} \}, \quad j \in [1, \tilde{r}] \]  

(2)

For example, the protein sequence AAB18559 has taxonomic assignments of "511145" and "723603" that each appeared once.

**Definition 3. Provenance.** For the particular protein \( P_i \), we define \( prov(A_{i,a}) \) the provenance of annotation \( A_{i,a} \) as a set of databases that the annotation \( A_{i,a} \) originates from:

\[ prov(A_{i,a}) \in \{ \text{GenBank, trEMBL, PDB, RefSeq, SwissProt} \} \]

In Definition 3, annotations from GenBank, trEMBL, and PDB are calculated computationally, while annotations from RefSeq and SwissProt are manually reviewed. For example, \( prov(511145) = \text{GenBank} \) meaning that the tax id "511145" for the sequence AAB18559 originates from the GenBank database.

**Definition 4. Annotation Probability.** We define probability for each taxonomic assignment based on the frequency of each annotation divided by total taxonomic assignments from different databases as follows:

\[ \text{prob}(A_{i,a}) = \frac{freq(A_{i,a} \in \text{Comp}) + w \times freq(A_{i,a} \in \text{Rev})}{\sum_{j \in \text{Comp}} freq(A_{i,j}) + \sum_{j \in \text{Rev}} (w \times freq(A_{i,j}))} \]  

(4)

In Definition 4, \( A_{i,a} \in \text{Comp} \) represents the annotation that calculated computationally (Comp) from databases i.e., GenBank, trEMBL, PDB, and \( A_{i,a} \in \text{Rev} \) denotes the reviewed (Rev) one from RefSeq, SwissProt. One annotation might originate from both reviewed and computational created databases. We use a conservative weighting factor, \( w \), to denote the importance of the experimental annotation (manually reviewed) in which \( w \) is an integer number and \( w \geq 1 \).

The upper bound for total proteins, i.e., \( |P| \), is 174M at the time we downloaded NR. Each leaf node, \( V_a \), in the phylogenetic tree is annotated with the information described in the Definitions 2, 3, and 4. There are list of frequencies and provenances, shows as top bar, since one particular taxonomic annotation might originate from different databases.

\[ V_a = \{ \text{prob}(A_{i,a}), freq(A_{i,a}), prov(A_{i,a}) \} \]  

(5)

For particular protein \( P_i \), we define most probable annotation (MPA) as \( MPA(P_i) = A_{i,\tilde{r}} \) as an annotation with the highest probability among
2.2 Approach to detect taxonomic misclassification

Our approach is as follows: first, we run a BoaG query on the NR database. This query runs on the full NR database in the Hadoop cluster. BoaG query is shown in the supplemental file. The algorithm 1 describes the detection approach for misclassified sequences. It iterates over the entire NR database. In line 4, it takes a protein $P_i$ and generates a phylogeny tree from the set of taxonomic assignments for $P_i$. Then, in line 5, it checks if it has a misclassification. If the lowest common ancestor (LCA) is the root level, it means there is a considerable distance between taxonomic assignments for that particular protein sequence. Therefore, there is a potential misassignment among the list of the taxonomic assignments due to the contamination in the sample, error in the computational method, or data entry by the researchers who deposited the sequence. We call this a root violation or any other violations. In addition, we looked at the highly conserved proteins in the order, and family violations. In line 6, it checks if the LCA of the generated tree shows a root violation or any other violations. The conserved($P_i$) expression checks if the protein sequence is a conserved one (See Eq. 6). This requires $O(1)$ time because this is a straight-forward fetch, and we have the pointer to the root of the tree to check the LCA. In line 5, to check that a protein is not in a conserved list, Definition 5, it requires a membership test and takes $O(1)$ time. This conserved list is a precomputed list from our dataset that is shown in our repository. We wrote a multi-threaded Python code, and the total run time for the algorithm was seven hours for the entire NR database on an iMac (Retina 5K, 27-inch, Late 2015) with core i7 and 32 GB RAM. For the second procedure, in line 11, the algorithm requires $O(|τ|)$ to calculate the probability of each leaf in the generated phylogenetic tree.

Algorithm 2 Annotation correction: The Most Probable Annotation (MPA) for the misclassified sequences. Input from the BoaG query (See supplemental)

```
1: procedure mostProbable($P_i$, p, c)
2:   top_ann ← max(prob($P_i$))
3:   if prob(top_ann) ≥ p then
4:     return top_ann.
5:   else
6:     cluster ← $C_j$ in which $P_i ∈ C_j$
7:     top_ann ← ClusterMostProbable(cluster, p, c).
8:     return top_ann.
9: procedure ClusterMostProbable(cluster, p, c)
10:   if size(cluster) ≥ c then
11:      for $A_{i,a}$ in $τ(cluster)$ do
12:        $V_a ← prob(A_{i,a}).list(freq(A_{i,a}), prov(A_{i,a}))$
13:      top_ann ← max($prob(τ(cluster))$) ⇒ Most probable taxa
14:     if prob(top_ann) ≥ p then
15:        return top_ann.
16:   else
17:      return null ⇒ Cannot fix misclassification
```

2.3 The most probable taxonomic assignment for detected misclassifications

For the detected misclassified sequences, we defined criteria to propose the most probable taxonomic assignment (MPA). First, we ran a BoaG query to retrieve the annotations and clustering information at 95% similarity. The BoaG script is shown in the supplemental file. As shown in Definition 4, we considered provenance or database of origin, frequency of annotations to calculate the probable taxonomic assignment (MPA), which is the highest probability. Let’s assume that $P_i$ belongs to cluster $C_j$. If the algorithm does not find the MPA within a certain threshold, probability $p$, then we look at the cluster of 95% similarity that the sequence belongs to. Second, we found the most probable taxonomic assignment in $C_j$. If a particular taxonomic assignment was the most frequent one in $C_j$, then we return that annotation as the MPA for the protein sequence $P_i$. For example, in cluster $C_j$, seven sequences out of 10 sequences have a specific annotation. Then, we consider this annotation to be the most probable annotation protein sequence $P_i$ with 70% confidence.

Details are shown in the algorithm 2. In line 2, for a particular protein $P_i$, it returns the most frequent taxonomic assignment within a certain threshold $p$. Let’s assume we want a taxonomic assignment that appears more than 70% of the time. If the algorithm does not find the MPA, it checks the cluster $C_j$ with 95% similarity that this sequence belongs to and finds the one with a certain probability, $p$, and a cluster size, $c$ (line 7). In line 9, ClusterMostProbable takes the cluster id and finds the most probable taxonomic assignment in the cluster (line 13).

The algorithm 2 requires $O(|τ(P_i)|)$ time, Definition 2, to find the top(1) or maximum probability of an annotation in the list of annotations.

2.4 Simulated and literature dataset

To evaluate the performance of our taxonomic misclassification approach, we generated a simulated dataset. We took a subset of one million proteins

...
of the reviewed dataset, i.e. RefSeq database, and randomly misclassified 50% of the proteins in the sample by adding a taxonomic assignment from another phylum or kingdom. Then, we tested if the approach can detect these sequences. We also tested our approach for detecting misclassified sequences and correcting them on the real-world data, presented in the literature (Kozlov et al. (2016); Edgar (2018)). These works have focused on the RNA dataset, and they quantified misclassified RNA sequences. We also used CD-HIT to cluster RNA databases based on 95% sequence similarity. Further details on the simulated dataset, scripts, and data files can be accessed from https://github.com/boalang/nr.

2.5 Sensitivity analysis

We define sensitivity analysis as a way that an input parameter affects the output of the proposed approach. Here, probability based on annotation frequencies and the cluster size are the two input parameters that affect what percentages of detected misclassified sequences that we can fix, i.e., MPA, as shown in Algorithm 2 on the NR dataset. The algorithm will not give the same suggestion for changes in parameters. For example, if we change the cluster size, no. of proteins in the cluster, it may or may not find correct taxa. We conducted a sensitivity analysis based on the probability of each annotation that we defined in Definition 4 and the size of the cluster corrected taxa. We conducted a sensitivity analysis based on the probability of each annotation that we defined in Definition 4 and the size of the cluster corrected taxa. We conducted a sensitivity analysis based on the probability of each annotation that we defined in Definition 4 and the size of the cluster corrected taxa. We conducted a sensitivity analysis based on the probability of each annotation that we defined in Definition 4 and the size of the cluster corrected taxa.

3 Results

In this section, we present the number of proteins that are misclassified taxonomically. We also present the performance of our work on the simulated dataset and the datasets presented in the literature. Then, we describe our findings on misassignments in the clusters. Next, we present correcting taxonomic misclassification. Finally, we discuss a case study that we explored deeply to identify a subset of clusters that contain sequences with a taxonomic misclassification.

Table 1. Detected misclassified taxonomic proteins in the NR database.

<table>
<thead>
<tr>
<th>taxa</th>
<th>total</th>
<th>root</th>
<th>Kingdom</th>
<th>Phylum</th>
<th>Class</th>
<th>Order</th>
<th>Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>17,496,167</td>
<td>39,237</td>
<td>47,271</td>
<td>59,606</td>
<td>147,132</td>
<td>256,065</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5,921,066</td>
<td>14,376</td>
<td>19,666</td>
<td>58,755</td>
<td>104,709</td>
<td>236,515</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2,132,971</td>
<td>4,673</td>
<td>21,587</td>
<td>64,801</td>
<td>17,662</td>
<td>49,054</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1,022,482</td>
<td>3,143</td>
<td>9,469</td>
<td>34,322</td>
<td>10,050</td>
<td>33,276</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>342,760</td>
<td>2,509</td>
<td>5,662</td>
<td>24,136</td>
<td>7,333</td>
<td>37,998</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>388,794</td>
<td>1,572</td>
<td>3,959</td>
<td>12,972</td>
<td>5,905</td>
<td>13,488</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>262,682</td>
<td>1,121</td>
<td>2,803</td>
<td>5,988</td>
<td>3,757</td>
<td>10,073</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>190,756</td>
<td>783</td>
<td>2,647</td>
<td>3,825</td>
<td>5,173</td>
<td>7,557</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>156,767</td>
<td>667</td>
<td>1,843</td>
<td>3,805</td>
<td>2,451</td>
<td>6,413</td>
<td></td>
</tr>
<tr>
<td>&gt;10</td>
<td>960,991</td>
<td>10,940</td>
<td>23,232</td>
<td>30,048</td>
<td>38,679</td>
<td>46,391</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.1 Detected taxonomically misclassified proteins

We found '29,175,336' proteins in the NR database that have more than one distinct taxonomic assignments. The rest of the proteins have identical taxonomic assignments, even though they originate from different databases. The total number of potential taxonomically misclassified sequences is '2,238,230' out of '29,175,336' (7.6%) at the time of download. This percentage of NR is significant because of the error propagation in the downstream analysis (Mukherjee et al. (2015)). Table 1 shows the number of violations in the protein sequences in NR at the superkingdom to the family level that have been detected by applying distance in the phylogenetic tree. The second column shows the number of total proteins that have a certain number of taxonomic assignments. For example, there are '17,496,167' protein sequences in NR that have two taxonomic assignments in which '30,237' of them have potential root violations and '47,271', '202,205', '59,606', '147,132', '256,065' have kingdom, phylum, class, order, and family violations respectively. For the NR dataset, we did a sample study of 1000 samples, and manually found 5.5% misassignment. The potentially misclassified sequences detected by the approach was around 7.6% that is consistent with the total number that was manually found, i.e. 5.5%.

Table 1 shows proteins that have less than 10 taxonomic assignments. The last row shows all other proteins with more than 10 assignments. The first two bold rows show the highest potential misassignments because if a protein has two or three taxonomic assignments and shows a root or kingdom violation, it is more likely to be misclassified. The full list of detected misclassified proteins, and detailed analysis are shown in our GitHub repository. We did not report the genus conflict since the probability of a false-positive misclassification is much higher compared to other taxonomic levels of conflict, such as root and superkingdom.

Figure 3 shows one example of a detected misclassified protein, with an id of NP_001026909. Since the lowest common ancestor in this tree is the root, it means those taxonomic assignments belong to a different kingdom, phylum, class, order, and family respectively. For example, there are '29,175,336' proteins in the NR database that have more than one distinct taxonomic assignments.

In this section, we present the number of proteins that are misclassified taxonomically. We also present the performance of our work on the simulated dataset and the datasets presented in the literature. Then, we describe our findings on misassignments in the clusters. Next, we present correcting taxonomic misclassification. Finally, we discuss a case study that we explored deeply to identify a subset of clusters that contain sequences with a taxonomic misclassification.
and UniProt (TrEMBL, Consortium (2014)) respectively. As it is shown in the annotations, there are potential misassignments even though the key IDs originate from the reviewed databases, i.e., RefSeq and SwissProt. In this example, synthetic construct is the misassignment, and the MPA for this protein is <i>homo sapiens</i>.

We also explored some clusters in depth as a case study and identified proteins that are taxonomically misclassified as Glycine, which are in fact contamination in the sample (see supplementary materials).

### 3.2 Performance on simulated and real-world dataset

Our approach to detecting taxonomically misclassified proteins on the simulated dataset showed 87% recall and 97% precision. We define <i>true positive</i> (TP) as sequences that misclassified in the sample, and our approach retrieves those sequences. <i>False positives</i> (FP) are sequences that do not have misassignments, but our approach classified them as misclassified sequences. <i>False negative</i> (FN) is a reviewed sequence which the algorithm incorrectly classifies as correct (not misclassified), while it is misclassified. Some of these false negatives are due to changes in the taxonomies over time. Some taxonomic IDs might be obsolete, deleted, or get merged into other tax ids. We also found that some of the trees generated by NCBI API have the root named "Cellular Organisms" with rank equal to "no rank", that did not fall in any of the taxonomic ranking. We use the following formula to calculate precision and recall:

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

We extended our experiment and added more than two random assignments to the proteins and the precision increased. The reason is that adding more random assignments increases the distance among tax IDs in the phylogeny tree and hence increases the chance of detection by the approach. We also tested our approach on the dataset presented by (Edgar (2018)) in which they explored the Greengenes and the SILVA database for taxonomic error. Our methods reproduced their finding on annotation conflicts among SILVA and Greengenes (McDonal et al. (2021)) database. We did not run their approach on the simulated dataset since it was designed to detect misclassifications in RNA sequences, not proteins. For evaluating our work, we looked for similar works that focused on detecting taxonomic misassignments. However, their approach was hard-coded for RNA sequences. Therefore, we modified our approach to test on their dataset. The proposed work focuses on inconsistencies among the list of taxonomies, and it can be applied to the RNA sequences as well.

We clustered their dataset at 95% similarity and used the same consensus-based technique to detect conflicts between sequences and clusters. The phylogeny-aware technique proposed by Kozlov et al., called SATIVA, identifies and corrects misclassified sequences for RNA databases (Kozlov et al. 2016). They utilized the Evolutionary Placement Algorithm (EPA) to detect misclassified sequences. In their approach, a reference tree is created. Then, to estimate the most likely placements of the query sequence in the reference tree, they use EPA. They took their RNA dataset and cluster the sequences at 95% similarity, then utilized their technique to check if the annotation of each sequence has a conflict with a cluster that the sequence belongs to. There is a difference between the NR dataset and the RNA database presented by Kozlov et al. in terms of the number of taxonomic annotation. In their experiment, they have one taxonomic label for each sequence; however, in the NR database, there are several annotations for each protein sequence. Therefore, their approach is not designed to detect misclassification in a set of given annotations. In terms of running time, the clustering at 95% is less expensive than running sequence alignment and generating phylogeny-tree and verifying each query sequence. Therefore, our approach is scalable for large-scale sequence databases. In general,

### 3.3 Detected misassignments in clusters

Table 3 shows the number of potential misclassified sequences in the clusters at 95% similarity at the superkingdom level to the family level that has been detected by applying distance in the phylogenetic tree. The second column shows the number of entire clusters that have a certain number of taxonomic assignments. For example, there are "12,960,476" clusters at 95% similarity that have two taxonomic assignments in which "17,099" of them have potential root violations and "92,526", "263,844", "100,560", "267,251", "461,795" have kingdom, phylum, class, order, and family violations respectively. The number of root violations for two tax assignments in clusters is less than sequences because there are protein sequences that do not belong to any clusters at 95% similarity. 64m out of 174M proteins (36%) in the NR-database are unclustered (see supplemental files). The total number of potential misclassifications for clusters at 95% similarity, without genus level, is "3,689,089" out of "25,159,866" clusters that have more than one taxa, which are 15% of total clusters.

Figure 4 shows an example of detected taxonomically assigned annotations in the cluster-id 490503. Leaves are annotated with databases of origins and frequency of taxonomic assignments as a bar chart from all reviewed and unreviewed databases, i.e., RefSeq, GenBank, PDB, UniProt, SwissProt, and UniProt (TrEMBL) respectively. For this cluster, there is no annotation from PDB, SwissProt, and TrEMBL databases.
3.4 Correcting Taxonomic Misclassification

Each protein sequence belongs to one and only one cluster. We analyzed the top three taxonomic annotations of each sequence and compared them with the top three taxonomic annotations of the cluster the sequence belongs to. For example, top three taxonomic assignment for sequence with id AAA32344 is '10743', '656390', '562'. This sequence falls in the cluster id 8461728, and the top tax ids in this cluster are '562', '83334', '621'. We consider this as a conflict between sequence AAA32344 and cluster 8461728. All three annotations are different; therefore, we consider this case as three conflicts. If two annotations out of three are different, we classify this as two conflicts. If one taxonomic annotation is different from the two sets, we classify it as one conflict. Finally, if the three annotations are identical, there is no conflict. Figure 5 shows different percentages of conflicts from the subset of one million sequences in the NR database.

Table 4 shows several examples of the protein sequences that we have found to be misclassified in the NR database. The first column represents the sequence id, and the second column is the cluster id corresponds to the sequence. The third column shows the top three taxonomic assignments. In our method, we proposed the most probable taxonomic assignment for each protein sequence. We applied this method to the dataset by SATIVA, and could reproduce the proposed taxa assignments. In our method, we proposed the most probable taxonomic assignment for each protein sequence. We applied this method to the dataset by SATIVA, and could reproduce the proposed taxa assignments.

3.5 Running time

We conducted an analysis of the RNA dataset presented by SATIVA (Kozlov et al. (2016)) with different samples of sequences. Firstly, we took 100 sequences and ran SATIVA in the sample. Next, we took 500 sequences. In two other experiments, we took 1000 and 2000 additional sequences and recorded the running time. Figure 6 shows the comparison in terms of running time (min) between proposed work and the SATIVA method. The most time-consuming part of our approach is the clustering time (run by CD-HIT software). By adding more sequences, the runtime slightly increased. In contrast, for the SATIVA method, as we increase the number of sequences, the running time increases significantly. The computational expensive part of the SATIVA approach is the phylogenetic methods (the Evolutionary Placement Algorithm) it employs. The comparison between the proposed approach and SATIVA method has been made on the local system iMac (Retina 5K, 27-inch, Late 2015) with core i7 and 32 GB RAM.

4 Discussion and conclusion

In this work, we addressed taxonomically misclassified sequences in the large publicly available databases by utilizing our domain-specific language and Hadoop-based infrastructure. We focused on the misassignments at the taxonomic level, and similar to MisPred (Nagy and Patthy (2013)), we utilize the current knowledge of organismal classification, to detect annotation errors. Similar to (Holiday et al. (2017)), we utilized this form of knowledge-based reasoning for quality control and detect annotation errors. Compared to other works, our work differs in that we do not need to run sequence similarity to explore annotations and find taxonomic inconsistency for each query sequence in the NR database. Instead, first, we clustered the NR proteins at the data generation phase, and this is a one-time cost and used the clustering information later to detect annotation error and propose the most probable annotations. In this work, we proposed a heuristic method to find inconsistencies in the metadata, i.e., taxonomic assignments. In our method, we proposed the most probable taxonomic assignment for each protein sequence. We applied this method to the entire dataset. We also provided a Python implementation in that a could be used for analyzing a list of annotations for any protein of interest and find the misclassification. The violations reported in this paper in Table 1 are the upper bound of the misassignments. The more stringent filter includes hypothetical protein and membrane protein functions in the list of conserved protein, which will lower the number of identified misclassification.
We use open-source CD-HIT clustering software only at the data generation phase, and we could utilize any other clustering software. Steiniger and colleagues have built a novel clustering tool that clusters a huge protein database in linear time (Steiniger and Söding 2018). Since this one-time clustering costs only in the data generation phase, our approach to detect misassignments and propose the most probable taxonomic assignment is scalable.

4.1 Applications and limitations
At 95% similarity, 64M sequences in the NR remain unclustered. Therefore, if a particular protein remains unclustered, there is not enough consensus information to correct annotation for that protein. A solution for this might be to take the evolutionary Placement Algorithm (EPA) approach (Kozlov et al. 2016) for these sequences that remains as future work. The proposed technique to detect misassignments may fail with recent horizontal gene transfer (HGT) events since HGT is not transferred from parent to offspring. However, the consensus information from the clusters might reveal annotation errors. The proposed heuristic technique and findings could also be applied to other databases. Current work focuses on detecting and correcting misassignments at the level of taxonomic assignments, and we do not address protein function annotations.

4.2 Conclusion
Misclassification can lead to significant error propagation in the downstream analysis. In this work, we proposed a heuristic approach to detect misclassified taxonomic assignments and find the most probable annotations for misclassified sequences. This method will be a valuable tool in cleaning up large-scale public databases. The technique we proposed could be extended in the form of ontologies to address other annotation errors like protein functions.

5 Acknowledgments
This study was supported by the National Science Foundation under Grant CCF-15-18897, CNS-15-13263, and CCF-19-34884 and the VPR office at Iowa State University. The listed funders played no role in the design of the study, data generation, implementation, or in writing the manuscript. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation grant number ACI-1548562.

References