RESEARCH ARTICLE

Comprehensive Analysis of Non Redundant Protein Database

Hamid Bagheri[1*], Robert Dyer[2†], Andrew J Severin[3†], and Hridesh Rajan[1†]

Abstract

Background: Scientists around the world use NCBI's non-redundant (NR) database to identify the taxonomic origin and functional annotation of their favorite protein sequences using BLAST. Unfortunately, due to the exponential growth of this database, many scientists do not have a good understanding of the contents of the NR database. There is a need for tools to explore the contents of large biological datasets, such as NR, to better understand the assumptions and limitations of the data they contain.

Results: Protein sequence data, protein functional annotation, and taxonomic assignment from NCBI's NR database were placed into a BoaG database, a domain-specific language and shared data science infrastructure for genomics, along with a CD-HIT clustering of all these protein sequences at different sequence similarity levels. We show that BoaG can efficiently perform queries on this large dataset to determine the average length of protein sequences and identify the most common taxonomic assignments and functional annotations. Using the clustering information, we also show that the non-redundant (NR) database has a considerable amount of annotation redundancy at the 95% similarity level.

Conclusions: We implemented BoaG and provided a web-based interface to BoaG's infrastructure that will help researchers to explore the dataset further. Researchers can submit queries and download the results or share them with others.

Availability and implementation: The web-interface of the BoaG infrastructure can be accessed here: http://boa.cs.iastate.edu/boag. Please use **user = boag** and **password = boag** to login. Source code and other documentation are also provided as a GitHub repository: `https://github.com/boalang/NR_Dataset`.

Keywords: NR; Domain-Specific Language; Protein functions; Taxonomic assignments

1 Background

The amount of sequencing data generated every year continues to grow exponentially. GenBank [1], has more than doubled in the last three years from 317 million sequences with 1.3 trillion bases to over 773.7 million sequences with 3.6 trillion bases. A researcher can choose to deposit a sequence into one of several different databases and frequently deposit the same sequence into multiple databases. This results in the problem of redundant information inflating the size of all known sequences. To address the growing challenge of sequences redundancy in public databases, a Non-Redundant (NR) database was introduced by the National Center for Biotechnology Information (NCBI) [2]. NR is defined by NCBI as protein sequences that have 100% identity and are the same protein length. This means that sequences that are shorter but have 100% identity are retained in the database and may or may not be labeled as a partial sequence. There is still redundant information contained in NCBI's non-redundant database, the extent of which is not widely known. The NR database encompasses protein sequences from non-curated (low quality) and curated (high quality) databases:

- GenBank/GenPept:
  This is unreviewed and low-quality sequences due to submission from individuals and laboratories.

[*] Correspondence: hbagheri@iastate.edu
[1] Department of Computer Science, Iowa State University, 226 Atanasoff Hall, 50011 Ames, US
Full list of author information is available at the end of the article
[†] Email address: Robert Dyer: rdyer@unl.edu; Andrew Severin: severin@iastate.edu; Hridesh Rajan: hridesh@iastate.edu

- trEMBL: This is an unreviewed subset of UniProt [3]. These sequences are annotated with computational tools.
- SwissProt: This is a manually annotated protein sequences [4].
- RefSeq: This is the manually reviewed sequences from GenBank and is maintained by NCBI's staff [5].
- PIR: This is a non-redundant annotated protein sequence database [6].
- PDB: This database is annotated experimentally, and it also contains structures of proteins and nucleic acids [7].

Researchers use BLAST [8] to query the NR database to identify homologous sequences and use that information to try and make an informed decision on the taxonomic assignment and function of unknown protein sequences.

The main advantage of NR is that it is comprehensive and solves the redundancy at the identical sequence level; however, the amount of redundancy and ambiguity of annotations at the large scale remains largely unknown and problematic to the user. Each sequence can have multiple annotations (i.e., taxonomic assignments and protein functional) resulting from the merging of definition lines from an identical sequence found in multiple databases. This redundancy impacts the ability of researchers to use, curate, and explore theNR database. For example, it is difficult to assess the confidence of an annotation because it is hard to determine where (the provenance) and how many times (frequency) a given annotation was assigned to a known sequence from multiple databases. In addition, there are unseen biases to the sequences contained in the nr database with significantly more coverage for some species/clades in the tree of life and other species with little to no sequences. To fully leverage the sequence data contained in the NR database, the clustering of proteins based on sequence similarity would be greatly beneficial.

A robust, distributed infrastructure is needed to analyze and quantify the content of the NR database and its clustering information. To this end, we utilized BoaG to address these challenges at scale. BoaG belongs to the family of a domain-specific language and shared infrastructure, called Boa, that has been applied to address challenges in mining software repositories [9], genomics data [10], and big data transportation [11]. Boa can process and query terabytes of raw data and uses a backend based on map-reduce to effectively distribute computational analyses and querying tasks. MapReduce is a framework that has been used for scalable analysis in scientific data. Hadoop is an open-source implementation of MapReduce. BoaG has been shown to substantially reduce programming efforts, thus lowering the barrier to entry to analyze very large data sets and drastically improves scalability and reproducibility. BoaG has aggregators that are functions that run on the entire database or a large subset of the database and therefore takes advantage of the BoaG database designed for both the data and the computation to be distributed across the Hadoop cluster.

This work is built on top of previous work that we introduced BoaG as a domain-specific language and shared data science Hadoop-based infrastructure for genomic data [12]. We demonstrated the computational power of BoaG on a proof of concept dataset, RefSeq, on a VirtualBox and Docker container. We also showed a use case of BoaG in detecting and correcting misclassified sequences in the NR database [12].

Here, we built the infrastructure and made it publicly available for researchers to test different hypotheses. We extended the BoaG infrastructure by integrating the sequencing data of the NR database, and its clustering information to illustrate the potential of BoaG to analyze the information contained in large public sequence databases. To that end, the BoaG database and schema were generated, and the compiler has been modified. We took the protein sequences in the NR database and clustered them using CD-Hit at several sequence similarity levels, then took this clustering information and combine it with the sequence metadata corresponding to protein function and taxonomic assignment. Using this information, we are able to better quantify the content, taxonomic distribution of proteins, and protein functions in the NR database. Specifically, we answer the following questions:

- What are the provenance and frequency of annotations in the NR database?
- What are the levels of ambiguity and redundancy in the taxonomic assignment and protein functions?
- How many conserved proteins are there in the NR database?
- What is the taxonomic distribution of protein across the tree of life?
- What are summary statistics for clustering information at different similarity levels?
- What is the distribution of proteins length in the NR database?

We found that BoaG can perform queries on this large dataset to quickly determine the average length of protein sequences, along with the most common taxonomic assignments and functional annotations, and the area of the tree of life that are less explored by researchers. For all the analyses, the BoaG infrastructure took fewer lines of code, reduced storage size, and provided automatic parallelization for these analyses. BoaG's web-interface is also implemented and made publicly available for researchers to test different hypotheses and share them among others.

The rest of the paper is organized as follows. In Section 2, we present methods and materials for dataset generation and the BoaG infrastructure. In Section 3, we present some interesting insights from the NR database and its clustering information by utilizing BoaG. Then, we discuss the performance and efficiency of the BoaG language and infrastructure and compare it with Python and MongoDB. In Section 5, we conclude with suggestions for the future.

2 Methods and Materials

In this section, we will describe the overview architecture of the publicly available BoaG language and infrastructure. Then, we discuss the BoaG language types to support NR and its clustering information. Next, we describe data generation steps. Finally, we explain how to write an arbitrary BoaG query and submit it to our infrastructure.

2.1 Overview architecture

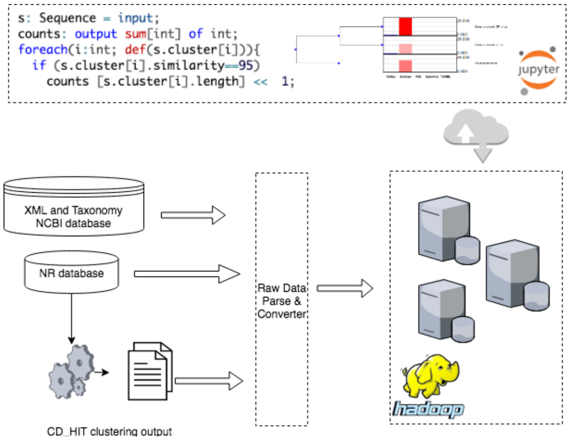
BoaG is a domain-specific language that uses a Hadoop based infrastructure for biological data [10]. A BoaG program is submitted to the infrastructure through the web-interface, as seen in Figure 1. It is compiled and executed on a distributed Hadoop cluster to query data in the BoaG formatted database 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 optimized for the 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.

2.2 BoaG domain-specific language

To utilize the potential of BoaG for our raw data, we created domain types, attributes, and functions specific to the non-redundant protein (NR) dataset and its clustering information. As shown in Table 1, Sequence, Cluster, and Annotation are types in our domain-specific language, and tax_id, tax_name, and definition line are attributes of the Annotation type. Sequence, Annotation types, and their attributes in BoaG language represent the NR database, and Cluster type with its attributes represents the CD-HIT clustering information. We created the data schema based on the Google protocol buffer, which is an efficient data representation of genomic data that provides both storage and computation efficiency on Hadoop. The raw data, i.e., the flat file of our raw data, was parsed into a Hadoop sequence file. When a BoaG program is executing in parallel, it emits values to the output aggregator that collects all data and provides the final output. Aggregators, for example, top, mean, maximum, and minimum, also can contain indices that would be a grouping operation similar to traditional query languages [9].

2.3 Cluster the NR database at different level of sequence similarity

We have utilized the CD-HIT program [13] to cluster protein sequences in NR using XSEDE computational resources [14]. CD-Hit provides protein clusters and a representative sequence for each cluster at the specified similarity level. CD-HIT [13] (version v4.6.8-2017-1208) was used using the following parameters (-n 5 -g 1 -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 shorter sequence be at least 80% of its length. The representative sequence, which is defined as the longest sequence in the cluster, was then clustered using the same parameters at 90% similarity. Clusters of lower similarity were generated using CD-Hit at 5% increments until 65% similarity, and until all of the following similarity, clusterings were obtained: 95%, 90%, 85%, 80%, 75%, 70%, and 65%. The database size for (entire NR) 95% similarity, was about 100GB. The CD-Hit computation required six days and 20 hours on a compute node with 2 CPU with 14 core each (Model: Intel(R) Xeon(R) CPU E5-2695 v3 @ 2.30GHz). The same analysis of representative sequences at 90%, 85%, 80%, 75%, 70% and 65% the database size were 40GB, 33GB, 28GB, 24GB, 21GB, 18GB, and 16GB respectively and the running times were three days, one day and 21 hours, one day and 12 hours, one day and two hours, 20 hours, and 16 hours respectively.



```
s: Sequence = input;
counts: output sum[int] of int;
foreach(i:int; def(s.cluster[i])){
  if (s.cluster[i].similarity==95)
    counts [s.cluster[i].length] << 1;
```

Figure 1: Users submit scripts on the web-interface to the BoaG infrastructure and results could be visualized by any general purpose languages such as R or Python.

2.4 Generate BoaG database from the raw dataset

The NCBI NR protein FASTA files were downloaded from ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/ on Oct 22, 2018. Taxonomic information was obtained from XML files downloaded from `https://ftp.ncbi.nlm.nih.gov/blast/temp/DB_XML/`. A flat file was generated by appending cluster information and taxonomic assignment to the annotation of each sequence in the NR database. A line of raw flatten data file that used for data generation is as follows: List of def-lines, as appeared in the original NR database, are separated by *ˆA* then it follows by σ character as a separator and starting of clustering information. Since one sequence might appear in

Table 1: Domain specific types for the NR database and its clustering information

Type	Attributes	Description
Sequence	seqid	sequence id
	Annotation	Annotation type
	Cluster	Cluster type
Annotation	keyID	protein sequence id
	defline	Definition line for proteins
	tax_id	taxonomic assignment
	tax_name	taxonomic name
Cluster	similarity	Similarity level from 65 to 95
	cid	cluster id
	representative	Cluster representative
	length	length of cluster
	seq_start	sequence starting point
	seq_stop	sequence stop
	rep_start	representative start
	rep_stop	representative stop
	match	percentage in sequence similarity

```
1 s: Sequence = input;
2 count : output sum [int][string][string] of int;
3 foreach(i:int; def(s.annotation[i]))
4   foreach(j:int; def(s.cluster[j]))
5    count[s.cluster[j].similarity][s.cluster[j].cid]
6   [s.annotation[i].tax_name]<<1;
```

Figure 2: Frequencies of taxonomic assignments for each cluster at different sequence similarity level. The variable count is the output aggregator that produces the sum of output indexed over similarity level, cluster id, and taxonomic name. The BoaG script and results are publicly available here:
http://boa.cs.iastate.edu/boag/?q=boa/job/public/31

different similarity levels we will have a list of clustering information separated by *ü* character, for example, be a representative or nr95, nr90, etc. or be a representative of a cluster at nr95 and a member of another cluster at nr90. As shown in Figure 1, a raw dataset has converted to a BoaG dataset based on the data schema shown in (Table 1). The converter program is written in Java, and it took about two hours. Data preprocessing, downloading, and clustering took about three days.

2.5 Submit queries on the BoaG infrastructure

We have implemented BoaG and provide a web-based interface to BoaG's infrastructure [15]. Researchers can go to the BoaG's web-interface and submit their query and download or share the results with others. For example, the program to determine the number of taxonomic assignments in each cluster at different similarity levels requires only five lines of BoaG code (Figure 2). In the first line, the variable s is defined as a sequence in NR, which is a top-level type in our language. In the second line, the variable *count* is an output aggregator that produces the sum of output indexed over cluster similarity level, cluster id, and taxonomic name. For each sequence, lines three and four iterates over all the annotations and clusters. Line five emits the value to the reducer for all the protein sequences in NR and provides the final results. The output can be downloaded and utilized for the post-processing tasks in the downstream analyses. For example, we used ETE3 (toolkit [16]) to generate the tree of life in Section 3.1. A compiler, data generation, and other documentation are provided on our GitHub repository.

3 Results

In this section, we present several interesting findings by utilizing BoaG language and its infrastructure to analyze 174M protein sequences and its 88M clusters. First, we discuss protein length in the NR database. Later, we will talk about the distribution of proteins across the tree of life. Then, we present clustering statistics from 95% down to 65% similarity. Other analyses are the frequency of taxonomic assignments in the proteins, the statistics about highly conserved proteins, the provenance of annotations, and the redundancy and ambiguity of annotations in the NR database.

3.1 NR Proteins are not evenly distributed across tree of life

We performed an analysis to understand how researchers have explored known phylums, described by Ruggiero *et al.* [17], across the tree of life. The distribution of the protein sequences at the 95% sequence similarity level among all known phyla in the tree of life is shown in Figure 3. The majority (74%) of the protein sequences are in Bacteria (74%), followed by Eukaryota (23%) and finally Archaea (2.21%). The phyla with the most sequenced proteins include Actinobacteria (14%), Proteobacteria (31%) and Firmicutes (12%) for Bacteria. In Eukaryota, the phyla with the most abundant sequenced proteins are Ascomycota (4.65%), Chordata (4.366%), Arthropoda (2.44%), and Basidiomycota (2.09%). In Archaea, only the Euryarchaeota phyla (1.68%) had sequenced proteins above 1% of the total. In the opposite extreme, there are several phyla that had little to no protein sequences. Specifically, the phyla in Eukaryota, Nematomorpha (44), Loricifera (0), Kinorhyncha (41), Gastrotricha (78), Cycliophora (3), Gnathostomulida (36), and Rhombozoa (83) each have fewer than 100 sequenced proteins after a 95% CD-HIT clustering. While the phyla with the lowest number of sequenced proteins in Bacterial and Archea had more than 5000 protein sequences, specifically, Dictyoglomi (5503) and Chrysiogenetes in Bacteria and Crenarchaeota (237862) in Archeal. The BoaG query is shown in supplemental file Figure 1, and the results generated in 52 minutes, and we used ETE3 toolkit [16] to generate the tree.

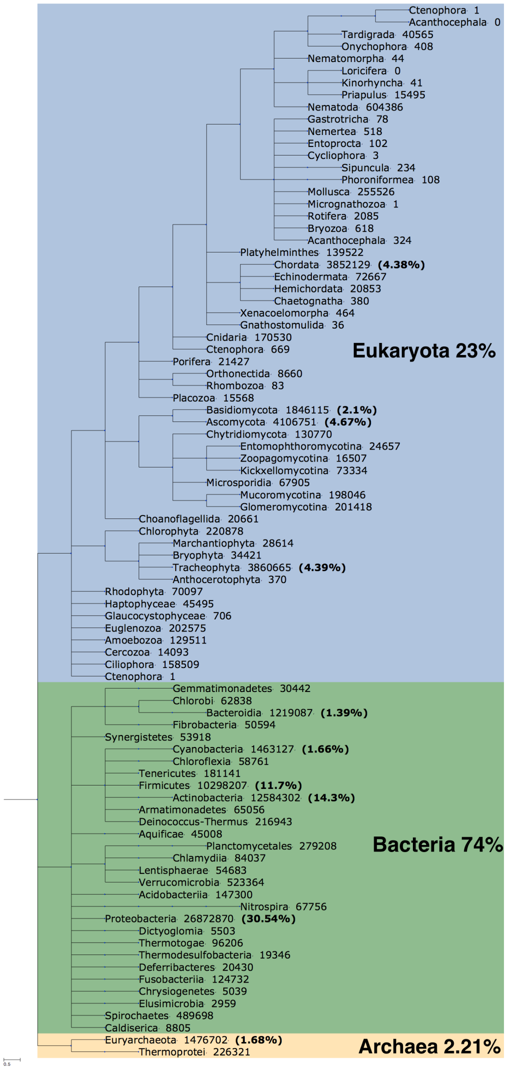


Figure 3: Distributions of proteins in the tree of life. Number in each node represents all proteins rooted with that node. Percentages less than 1 are not shown.

3.2 Proteins in NR vary greatly in length

The length of protein sequences in the NR database appears to be normally distributed with a mean of 365 amino acids, a standard deviation of 353 amino acids, and a long tail to the right (Figure 4). The smallest proteins are 11 amino acid peptides. These peptides are in the NR database from the PDB database, where small peptides are commonly part of a larger protein-peptide structure. These peptides can be synthetic constructs of viruses or fragments of larger proteins involved in protein-peptide or protein binding. The 100 longest proteins in NR have one of three protein functions: hypothetical protein (DBY08_01055) found in Clostridiales bacterium with unknown function (PWL95011), Titin found in multiple species involved in passive elasticity of muscle or LEPR-XLL domain-containing protein found in Chlorobium chlorochromatii with sizes of 74,488, 38,105 and 36,805 amino acids in length, respectively. The hypothetical protein (DBY08_01055) was submitted in March of 2018 and is now the longest known protein sequence superseding the previous holder of the longest sequence, Titin, by more than two-fold. Figure 4 shows the protein frequency by log(2) of protein length, which is normally distributed around a median length of 256 amino acids (2^8). Researchers may also explore and analyze the length of proteins in a different subset of NR. This query required five lines of code in BoaG (see supplemental file Figure 2, and it provided the results in about two minutes for the entire NR database using BoaG infrastructure.

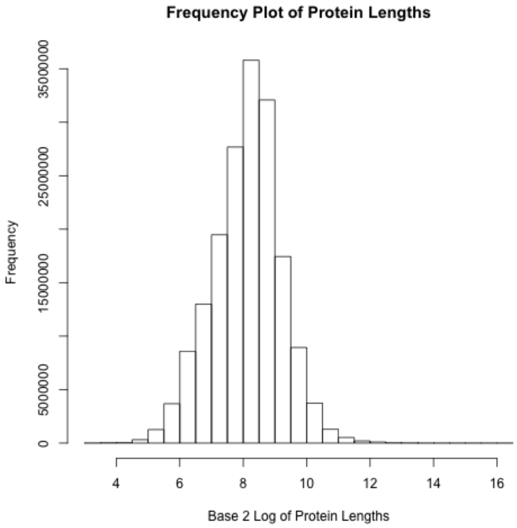


Figure 4: The protein frequency by log(2) of protein length.

3.3 Clustering of similar protein sequences indicate a much lower number of unique proteins in NR

We clustered NR at 95% sequence similarity and then at lower similarities with 5% intervals using the longest sequence in each cluster until we reach 65% similarity. As we would expect, the number of clusters, proteins, amino acid content, and taxa decreases as we form clusters at lower similarity using only representative sequences from the previous clusters. Approximately half of the protein sequences fall into clusters at 95% sequence similarity and requiring over 80% the length of the shorter sequence. However, 64 of the 174 million proteins at 95% sequence similarity remain unclustered. At 65% similarity, the NR database can be clustered into 34.4 million clusters, containing 23% of the original unclustered proteins and 21.5% of the original amino acid content. However, of the 40 million proteins at this similarity, 30.6 million (76.5%) are unclustered and of the 11.9% of the original unclustered

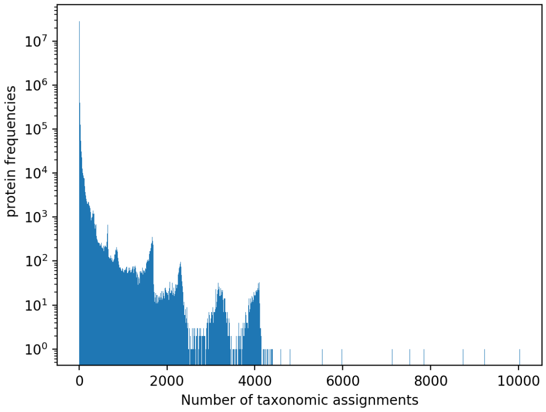


Figure 5: Frequency of protein sequences with different taxonomic assignments. The x-axis shows the number of taxonomic assignments and the y-axis the frequencies of protein sequences.

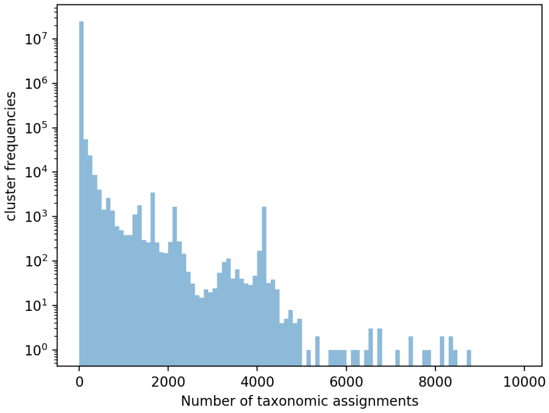


Figure 6: Frequency of clusters at 95% similarity. The X axis shows number of taxonomic assignments and the y axis the frequencies of clusters. The output of BoaG query in Figure 2 utilized to generate this chart.

taxa. The amount of similar data in the NR database has important consequences, assumptions, and limitations. The presence of 159 million taxa for 174 million sequences suggests that the naming of taxa almost has as unique as the sequence ids themselves. In reality, additional information is often added to the taxonomy to add specificity about the line, cultivar, or sample.

3.4 Almost as many Taxa as proteins

Since the taxonomic assignment for the protein sequences in NR were merged from several databases, there can be anywhere from one to 10034 taxonomic assignment for a given protein sequence. We analyzed the frequencies of taxonomic assignment in the NR database and identified sequences with a large number of taxa. The Protein sequence with the highest number of assigned taxonomic classifications included *Influenza A virus* with more than 10k assignments. Table 2 shows a few examples of the proteins that have a large number of taxonomic assignments. The protein sequences with the highest level of taxonomic assignments are most likely due to the fact that viruses and bacteria are given a strain identifier appended to their taxonomy name resulting in many taxonomic assignments for the same sequence. Figure 5 shows the frequency of proteins that have a certain number of taxonomic assignments. For example, 17,496,167 protein sequences have two annotations, and 5,921,066 proteins have three annotations. This implies that annotations have a large number of redundancy that impacts exploring and analyzing of the NR database. More details are discussed in the Section 3.7.

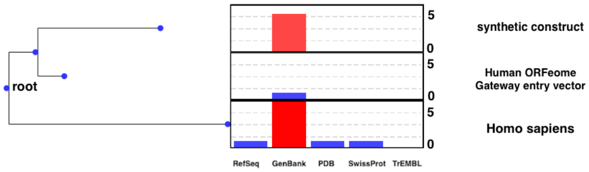


Figure 7: Provenance and frequency of annotations from each database for sequence with primary id of NP_000311.2. Most annotations originate from GenBank.

Similarly, Figure 6 shows the frequency of clusters at 95% similarity that has a certain number of taxonomic assignments. For example, 12,960,476 clusters sequences have two taxonomic assignments, and 4,683,663 clusters have three taxonomic assignments. To generate this output, a BoaG script, shown in Figure 2, needs five lines of code and takes about seven minutes on the BoaG infrastructure.

3.5 Highly conserved protein functions

We used BoaG aggregators to query the NR database and identify highly conserved protein sequences. We defined highly conserved as the protein sequences with at least 10 distinct taxonomic assignments. Some examples of the top protein functions and their frequencies are shown in Table 3. For example, as we would expect, we see the highly conserved rRNA protein function among the list. In addition, we see an abundance of uninformative/generic functions like *unknown function*, *membrane protein*, and *transcriptional regulator*. The BoaG query required 11 minutes on the infrastructure to finish (supplemental Figure 3).

3.6 Provenance of annotations

We refer to annotation provenance as a database of origin for the annotations, i.e., taxonomic assignments and protein functions. Protein annotations come from

Table 2: proteins that have the large numbers of taxonomic assignments

Sequence ID	Protein Name	#of taxa
AAX11496	Influenza A virus (A/New York/32/2003(H3N2))	10,034
Q76V02	RecName: Full=Matrix protein 1; Short=M1	9,227
AAD31614	histone H3, partial [Euperipatoides leuckartii]	8,735
AAZ38596	Influenza A virus (A/New York/391/2005(H3N2))	7,854
YP_009118623	Influenza A virus (A/California/07/2009(H1N1))	7,536

Table 3: Examples of protein functions and their appearances in sequences that have more than 10 distinct taxa.

Category	Protein function	#of functions
Unknown	hypothetical/unknown/unnamed	27,649,805
Highly conserved	conserved hypothetical protein	96,348
Highly conserved	membrane protein	204,891
Highly generic	transcriptional regulator	192,757
Highly conserved	rRNA	21,836

different databases that are curated manually or calculated computationally. Therefore, in terms of quality of metadata, it would be beneficial for researchers to know about the origin of each taxonomic assignment as they explore their protein of interest. For each protein, users can create a phylogenetic tree from the list of taxonomic assignments. Figure 7 provides an example of the provenance and frequency of each taxonomic assignment for the protein sequence with id NP_000311.2. Leaves are annotated with a frequency of each taxonomic assignment as a bar chart from all reviewed and unreviewed databases, i.e., RefSeq, GenBank, PDB UniProt\SwissProt, and UniProt\TrEMBL respectively. Details on generating the tree are on the GitHub repository.

As shown in the previous work [12], the provenance information could be utilized to clean the NR database by assigning more weight to the manually reviewed annotations.

3.7 Redundancy and ambiguity of annotations

There is significant redundancy in the protein annotations of the NR database for the taxonomic assignment and protein function due to the integration from different databases. Using BoaG, we generated a non-redundant version of annotations, collapsing all identical annotations and providing the number of times that annotation was present in the original ID description. As it can be seen in Table 2, some proteins have thousands of taxonomic assignments. We previously explored the NR database for the taxonomic misclassified sequences [12]. The non-redundant version of annotations in the NR database improves the usage and querying of the NR database. The running time for this query was 19 minutes, as the output size was 54 GB that needed a longer time to write on the disk.

In addition, researchers independently use different words to refer to the same biological concept; for example, *unnamed protein*, *hypothetical protein*, and *unknown protein* have been used to describe an unknown protein function in different public databases. Another example is rRNA that appears in 21,836 functions with a different combination of other words. The protein function analysis needs a huge effort in natural language processing. This ambiguity in annotations negatively impacts the usage of the NR database. This implies that we need a better annotation methodology to improve the quality of metadata and answer different biological questions. One way of improving the annotation in public databases would be to utilize the ontologies, for example, GO [18] and PRO [19]. NCBI provides tools to limit the effects of redundancy. For example, the Conserved Domain Database (CDD) maintained by NCBI is a resource for proteins that clusters redundant homologous families to reduce redundancy [20]. However, this is at the level of sequences, and it does not address annotations and metadata.

4 Discussion

In this work, we implemented the BoaG infrastructure and made it publicly available. We used BoaG to explore the NR database along with the clustering information. We discussed the average length of proteins, distributions of proteins in the tree of life, top taxonomic assignments, and top protein functions. We also showed the annotation redundancy and ambiguity that affect the quality of metadata. Here, we utilized BoaG's aggregators to generate a summarized and non-redundant version of annotations. Summarizing these annotations will help researchers to utilize the wealth of public databases on protein annotations for different areas of biological research that include but are not limited to phylogenetics, taxonomy, and medical research to identify the causes of genetic diseases.

4.1 Storage and computational efficiency in BoaG

Exploring the entire non-redundant (NR) database is computationally expensive. Most researchers use subsets of the NR database to test their hypothesis, while BoaG provides a facility to explore the NR database in its entirety. There have been works on deduplication and reducing the NR size. Yu *et.al* [21] developed a pipeline to construct a subset of NCBI-NR database for the quick similarity search and annotation of huge metagenomic datasets based on BLAST-MEGAN. There is another approach based on MD5 checksum to provide a non-redundant protein database [22] by splitting sequence data in a single FASTA file and metadata in a SQL database.

In this work, we integrated all 174 million protein sequences and 159 million annotations. The BoaG data schema is based on the protocol buffer and stores in a binary file format, and hence it will significantly reduce the storage size. The translated dataset is much smaller even though in the Hadoop file system by default, we may have replication of factor two in order to provide the reliability in data storage across machines in a Hadoop cluster [23]. In this translation, no data loss happens since the BoaG database is binary. Figure 8re shows the storage efficiency of BoaG. The file size in the BoaG database is much smaller than the JSON file used in MongoDB. More details about the comparison between BoaG and the MongoDB and original raw data is shown in the GitHub repository.

Figure 9 describes the decrease in the required computation time with a corresponding increase in the number of Hadoop mappers. Query 1 is the analysis of protein length frequencies that described in Section 3.2. Query 2 is the analysis of protein distribution across the tree of life that described in Section 3.1. Query 3 is the analysis of Highly Conserved Proteins that we discussed in Section 3.5. Query 4 is the analysis of clusters in NR that described in Section 3.4. For these queries, we varied the number of mappers in the 5-node shared Hadoop clusters to evaluate the speedup results by adding additional mappers to an analysis. As we added more mappers, the running time decreases significantly.

4.2 Programming efficiency in BoaG

These analyses we discussed in this work required fewer lines of code in BoaG language and automatically translated to a larger parallel code in Java and run on Hadoop, reducing programming efforts. For example, the analysis of protein length, as discussed in Section 3.2, required BoaG only 2 minutes on a 5-nodes shared Hadoop, while a more complex one required 42 minutes and produced 40 GB output. The analyses that were performed here could have been performed using MongoDB and post-processing in Python or directly on the raw flat file using Bash. However, it would have taken more time and lines of code (See Supplemental Figure 4). Utilizing a genomics specific language like BoaG that provides a data preprocessing and curation at the data generation phase, makes downstream analysis based on the generated dataset more reliable. For example, while generating the BoaG dataset from the raw data, we found anomalies in the NR database. We detected several sequences that had no information and contained only X (unknown amino acid). These sequence IDs were reported to NCBI. A list of detected anomalies is given in the supplemental folder on our GitHub repository.

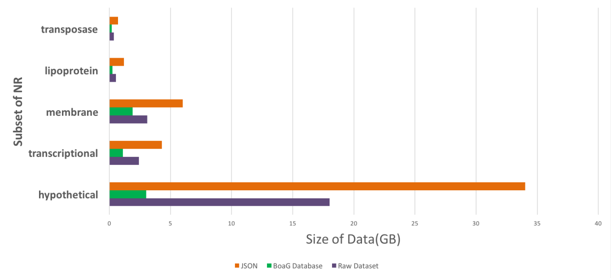


Figure 8: The NR storage efficiency in *Boa*. The *Boa* dataset is compared with the raw data and the equivalent of MongoDB.

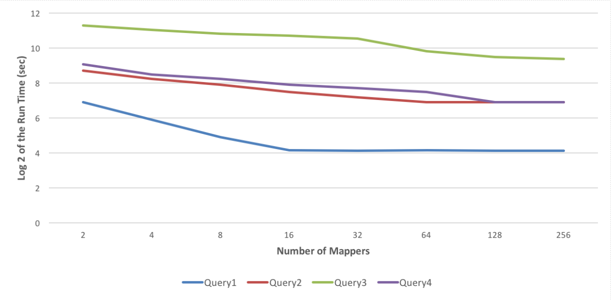


Figure 9: Scalability of *Boa* programs (time in log 2 seconds). Four queries are analysis of protein length frequencies, distribution in the tree of life, highly conserved proteins, and cluster analysis.

To summarize, BoaG provides automatic parallelization on top of Hadoop, reduces programming errors by abstracting details in few lines of code, and the curated dataset that is smaller than the original raw data. We anticipate that the strategy of BoaG might facilitate the exploration of other biological databases. We will also provide facilities for researches to clean NR database.

5 Conclusion

In this work, we explored the NR database and clustering information of the NR database at different similarity levels and showed the computational power of the BoaG language and infrastructure. We showed the storage efficiency, automatic parallelism with less effort compared to the general-purpose languages. We described the average length of protein sequences found in NR, most common taxonomic assignments, top protein functions, redundancy and ambiguity of annotations in NR. The redundancy and ambiguity at the annotation level impacts the usage, curation, and exploration of the NR database. BoaG infrastructure will greatly improve the usage and exploration of NR.

Declarations

Abbreviations
BoaG: Boa for Genomics; DSL: Domain-Specific Language; NR: Non-Redundant Protein Database

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Availability of data and material
The web-interface of the BoaG infrastructure can be accessed here: http://boa.cs.iastate.edu/boag. Please use **user = boag** and **password = boag** to login. Source code and other documentation are also provided as a GitHub repository: `https://github.com/boalang/NR_Dataset`.

Competing interests
The authors declare that they have no competing interests.

Funding
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.

Author's contributions
AJS conceived of the application of BoaG to the NR database and contributed to exploring the dataset from a biological perspective. HB wrote the codes, implemented the genomics specific types and customized the compiler. He ran analysis and prepared figures. HB provided a first manuscript. HR and RD contributed to the design of the BoaG domain-specific language for computing over data. All the authors read and approved the final manuscript.

Acknowledgments
We would like to thank the Bridges super computing center for providing support with Hadoop.

Author details
[1]Department of Computer Science, Iowa State University, 226 Atanasoff Hall, 50011 Ames, US. [2]Dept. of Computer Science & Engineering, University of Nebraska – Lincoln, , 68588 Lincoln, US. [3]Genome Informatics Facility , Iowa State University, 206 Science I, 50011 Ames, US.

References

1. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Sayers, E.W.: Genbank. Nucleic acids research **37**(suppl_1), 26–31 (2008)
2. Non-Redundant database (NR). `https://www.ncbi.nlm.nih.gov/refseq/about/nonredundantproteins/`. Accessed: 2019-06-10 (2019)
3. Consortium, U.: Uniprot: a hub for protein information. Nucleic acids research **43**(D1), 204–212 (2014)
4. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'donovan, C., Phan, I., *et al.*: The swiss-prot protein knowledgebase and its supplement trembl in 2003. Nucleic acids research **31**(1), 365–370 (2003)
5. Pruitt, K.D., Tatusova, T., Maglott, D.R.: Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic acids research **35**(suppl_1), 61–65 (2006)
6. Wu, C.H., Yeh, L.-S.L., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z., Kourtesis, P., Ledley, R.S., Suzek, B.E., *et al.*: The protein information resource. Nucleic acids research **31**(1), 345–347 (2003)
7. Berman, H.M., Bourne, P.E., Westbrook, J., Zardecki, C.: The protein data bank. In: Protein Structure, pp. 394–410. CRC Press, ??? (2003)
8. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of molecular biology **215**(3), 403–410 (1990)
9. Dyer, R., Nguyen, H.A., Rajan, H., Nguyen, T.N.: Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In: Proceedings of the 2013 International Conference on Software Engineering, pp. 422–431 (2013). IEEE Press
10. Bagheri, H., Muppirala, U., Masonbrink, R., Severin, A.J., Rajan, H.: Shared data science infrastructure for genomics data, doi: https://doi.org/10.21203/rs.2.4295/v3. BMC Bioinformatics (2019). doi:10.21203/rs.2.4295/v3
11. Islam, M.J., Sharma, A., Rajan, H.: A cyberinfrastructure for big data transportation engineering. Journal of Big Data Analytics in Transportation **1**(1), 83–94 (2019)
12. Bagheri, H., Severin, A., Rajan, H.: Detecting and correcting misclassified sequences in the large-scale public databases. Bioinformatics (2020). doi:10.1093/bioinformatics/btaa586. btaa586
13. Fu, L., Niu, B., Zhu, Z., Wu, S., Li, W.: Cd-hit: accelerated for clustering the next-generation sequencing data. Bioinformatics **28**(23), 3150–3152 (2012)
14. Nystrom, N.A., Levine, M.J., Roskies, R.Z., Scott, J.R.: Bridges: A uniquely flexible hpc resource for new communities and data analytics. In: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure. XSEDE '15, pp. 30–1308. ACM, New York, NY, USA (2015). doi:10.1145/2792745.2792775. http://doi.acm.org/10.1145/2792745.2792775
15. BoaG web-interface for Genomics data. `http://boa.cs.iastate.edu/boag/`. Accessed: 2020-05-10 (2020)
16. Huerta-Cepas, J., Serra, F., Bork, P.: Ete 3: reconstruction, analysis, and visualization of phylogenomic data. Molecular biology and evolution **33**(6), 1635–1638 (2016)
17. Ruggiero, M.A., Gordon, D.P., Orrell, T.M., Bailly, N., Bourgoin, T., Brusca, R.C., Cavalier-Smith, T., Guiry, M.D., Kirk, P.M.: A higher level classification of all living organisms. PloS one **10**(4) (2015)
18. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., *et al.*: Gene ontology: tool for the unification of biology. Nature genetics **25**(1), 25 (2000)
19. Natale, D.A., Arighi, C.N., Blake, J.A., Bona, J., Chen, C., Chen, S.-C., Christie, K.R., Cowart, J., D'Eustachio, P., Diehl, A.D., *et al.*: Protein ontology (pro): enhancing and scaling up the representation of protein entities. Nucleic acids research **45**(D1), 339–346 (2016)
20. Marchler-Bauer, A., Lu, S., Anderson, J.B., Chitsaz, F., Derbyshire, M.K., DeWeese-Scott, C., Fong, J.H., Geer, L.Y., Geer, R.C., Gonzales, N.R., *et al.*: Cdd: a conserved domain database for the functional annotation of proteins. Nucleic acids research **39**(suppl_1), 225–229 (2010)
21. Yu, K., Zhang, T.: Construction of customized sub-databases from ncbi-nr database for rapid annotation of huge metagenomic datasets using a combined blast and megan approach. PLoS One **8**(4), 59831 (2013)
22. Wilke, A., Harrison, T., Wilkening, J., Field, D., Glass, E.M., Kyrpides, N., Mavrommatis, K., Meyer, F.: The m5nr: a novel non-redundant database containing protein sequences and annotations from multiple sources and associated tools. BMC bioinformatics **13**(1), 141 (2012)
23. Borthakur, D., *et al.*: Hdfs architecture guide. Hadoop Apache Project **53**, 1–13 (2008)