Using a Novel Negative Selection Inspired Anomaly Detection Algorithm to Identify Corrupted Ribo-seq and RNA-seq Samples

Patrick Perkins
Bioinformatics Research Center
North Carolina State University
Raleigh, NC USA
pjperki2@ncsu.edu

Steffen Heber
Department of Computer Science
North Carolina State University
Raleigh, NC USA
sheber@ncsu.edu

ABSTRACT

RNA-seq and Ribo-seq are popular techniques for quantifying cellular transcription and translation. These experiments use next-generation sequencing to produce genome-wide high-resolution snapshots of the total populations of mRNAs and translating ribosomes within the investigated samples. When performed in concert, these experiments yield valuable information about protein synthesis rates and translational efficiency. Due to their intricate experimental protocols and demanding data processing requirements, quality control and analysis of such experiments are often challenging. Therefore, methods for accurately assessing data quality, and for identifying contaminated samples, are greatly needed. In the following we use a novel negative selection inspired algorithm called Boundary Detection Using Nearest Neighbors (BDUNN), for the identification of corrupted samples. Our algorithm constructs a detector set and reduced training set that defines the boundaries between normal data points and potential anomalies. Subsequently, a nearest neighbor algorithm is used to classify unseen observations. We compare the performance of BDUNN with other popular negative selection and one-class classification algorithms, and show that BDUNN is capable of accurately and efficiently detecting anomalies in standard anomaly detection datasets and simulated RNA-seq and Ribo-seq data sets. Furthermore, we have implemented our method within an existing R Shiny platform for analyzing RNA-seq an Ribo-seq datasets, which permits downstream analysis of anomalous samples.

CCS CONCEPTS

• **Computing methodologies~Machine learning algorithms** • **Applied computing~Bioinformatics**

KEYWORDS

Ribosome profiling, RNA-seq, anomaly detection, sample quality, machine learning, negative selection algorithm

ACM Reference format:

P. Perkins and S. Heber. 2019. Using a Novel Negative Selection Inspired Anomaly Detection Algorithm to Identify Corrupted Ribo-seq and RNA-seq Samples. In *10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB'19), September 7--10, 2019, Niagara Falls, NY, USA.* ACM, New York, NY, USA, 10 pages. DOI: https://doi.org/10.1145/3307339.3342169.

1 INTRODUCTION

In this paper, we describe a novel negative selection inspired algorithm, and evaluate its ability to identify contaminated RNA-seq and Ribo-seq datasets. RNA-seq and Ribo-seq are two of the most popular techniques for quantifying cellular translation and transcription [1]. By conducting paired RNA-seq and Ribo-seq experiments, researchers can quantify the number of ribosomes that are associated with individual transcripts, and estimate differential expression and translational efficiency. While these techniques are powerful methods for assessing dynamics of genes expression, they are often plagued by complications during sample preparation and data processing. These issues can make it difficult for researchers to interpret their data in biologically meaningful ways, and make data analysis and quality control challenging tasks that can require a considerable amount of expert knowledge. A valuable technique that can be used to aid researchers in identifying potentially impactful quality issues in these types of data before they perform downstream analyses is anomaly detection.

Anomaly detection is the process of identifying unusual patterns, events, or observations that deviate from expected behavior. It has applications across a variety of fields, including network intrusion, fraud detection, health monitoring, and

ACM-BCB '19, September 7–10, 2019, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-6666-3/19/09…$15.00
DOI: https://doi.org/10.1145/3307339.3342169

image processing [3]. Due to its potential for such widespread usage, many techniques have arisen to address the various forms of the problem as efficiently and accurately as possible. Examples include classification techniques, such as support vector machines and neural networks, distance-based methods like k-nearest-neighbor, clustering methods, and more traditional statistical techniques such as regression [3]. Among these techniques, a class of biological inspired algorithms, called artificial immune systems, have gained traction in various classification and anomaly detection applications. Artificial immune systems are soft-computing techniques which are inspired by the biological mechanisms of the vertebrate immune system [4]. One of the most prevalent artificial immune system techniques is the negative selection algorithm (NSA), first proposed by Forrest et al. in 1994, which mimics the negative selection processes that occur during T cell maturation [5]. NSAs identify anomalies, also called non-self data points, by establishing sets of detectors which do not match elements from a collection of self points used for training. During the process of detector generation, random detectors are generated, and those which match self points are eliminated from the detector set, similarly to how the thymus gland eliminates T cells which recognize self cells. Detectors are then used to classify a set of training samples as self or non-self using a designated matching rule. In traditional NSAs, observations were represented as binary strings, which facilitated simple matching rules for detector generation. While these methods commonly produce accurate classification results, they have several disadvantages. Algorithms which rely on random detector generation often suffer often from long and unpredictable runtimes, and difficulties in determining the amount of coverage of the problem space. NSA methods have also been shown to perform poorly in high-dimensional space due to the 'curse of high-dimensionality', which is a common characteristic of datasets in the current era of big-data, including problems in the field of bioinformatics [4].

In recent years, NSAs have been adapted for use in a much wider variety of applications. In 2003, Gonzalez et al. proposed the real-valued negative selection algorithm (RNSA), which applied the logic of traditional NSAs to problems in a real-value space [6]. Ji and Dasgupta presented V-detector in 2009, which was the first NSA to implement variable sized detectors to reduce the number of total detectors required to define the non-self space [7]. The demands of the big-data era have led to significant increases in the number of algorithms capable of efficiently analyzing large, multidimensional real-valued datasets [9]. The inefficiencies of early NSAs greatly limited their applications in problems such as these. To address these problems, numerous algorithms have been developed that attempt to increase the accuracy and efficiency of NSAs for more complex datasets, such as GB-RNSA, BIORV-NSA, and Vor-NSA [9-11]. GB-RNSA uses a grid mechanism to reduce the time cost of calculating distances and the overlapping coverage between detectors. BioRV-NSA implements variable self radii and a method for dynamically replacing ineffective detectors to decrease redundancy and the overall number of detectors. Vor-NSA uses Voronoi diagrams to calculate the optimal position for detectors. Their method achieves much faster computation by breaking away from the random detector generation steps of traditional NSAs to a method which places detectors in specific locations. NSAs such as these have proven to be powerful tools for performing anomaly detection in real valued datasets from various fields, and are therefore good candidates for identifying anomalies in RNA-seq and Ribo-seq samples [7-11].

Here we present Boundary Detection Using Nearest Neighbors, or BDUNN, an anomaly detection algorithm inspired by NSAs. BDUNN generates a set of detectors and a reduced set of self observations that define the boundary between self and non-self space, and employs a nearest neighbor algorithm to make data testing fast and accurate. We demonstrate that BDUNN can perform anomaly detection via one-class classification and matches the performance of comparable methods, and also show that BDUNN can be used to identify low quality RNA-seq and Ribo-seq samples using a set of informative data quality metrics. Additionally, we have implemented BDUNN within riboStreamR, a platform for quality control of RNA-seq and Ribo-seq data [12]. RiboStreamR uses BDUNN to highlight user-supplied samples which exhibit anomalous quality, and can be used to visualize characteristic features of the anomalous data.

2 Properties of BDUNN

2.1 Detector generation

In this section, we describe the process of detector generation in BDUNN, our one-class classification algorithm for anomaly detection. Detectors can be generalized as points which do not match (by some matching rule) the self training data and can be used during testing to identify non-self data. The algorithm is inspired by the detector generation process employed by NSAs, but instead of using detectors to define the entirety of the non-self space, it establishes a condensed set of detectors and training points that define the boundary of the self and non-self spaces in order to facilitate a nearest neighbor (NN) classification. A visual comparison between a tradition RNSA, V-detector, and BDUNN are shown in Figure 1.

The algorithm takes as input a set of training points which represent the self class, or normal observations, such as those in Figure 2A. A self space around each training point is determined based on a self radius r_s. The algorithm first establishes a designated number, n, of initial non-self detectors, D_p, by randomly generating detectors within the problem space, and removing points which are less than r_s away from any training point, as seen in Figure 2B and described in Algorithm 1a. Additional detectors are not added in subsequent steps of the algorithm, and therefore the initial number of randomly generated detectors, n, represents the maximum number of detectors that BDUNN will use during the testing phase. In general, it is safer to use a large value of n in order to ensure that sufficient coverage of the problem space is attained, although increasing the value of n leads to longer runtimes.

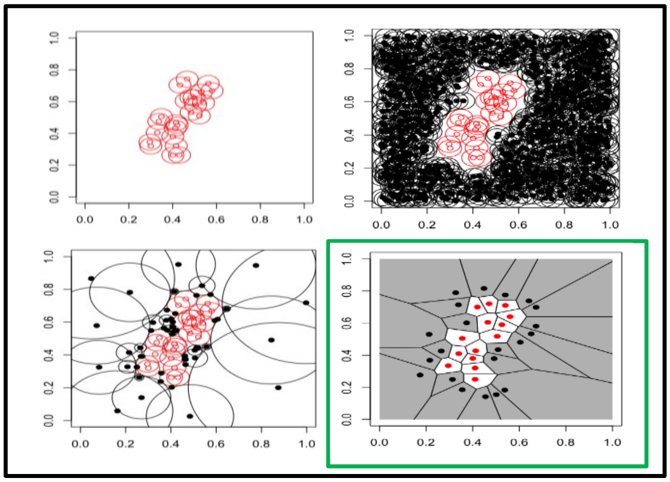

Figure 1. Two dimensional comparison of a traditional RNSA (top right), V-detector (bottom left), and BDUNN (bottom right) using a set of self training observations (top left). A self radius of 0.05 was used for all methods. RNSA and V-detector required 513 and 53 detectors, respectively, to reach 99% coverage. BDUNN generated a reduced set of 34 detectors.

```
Algorithm 1a. Initial Detector Generation:
Input: Set of all self points S, self radius $r_s$, detector sample size $n$;
Output: Initial detector set $D_p$;

While $length(D_p) < n$:
    Generate random detector $d$;
    If $d > r_s$ for all $s \in S$:
        Add $d$ to $D_p$;
Done;

Algorithm 1b. Boundary Set Determination:
Input: Set of all self points $S$, initial detector set $D_p$;
Output: Boundary detector set $D_b$, boundary self set $S_b$;

Compute Delaunay triangulation $T$ of $S \cup D_p$;
For all Delaunay simplices $t \in T$:
    If t contains points from both $S$ and $D_p$:
        Add points in $t$ from $S$ to $S_b$;
        Add points in $t$ from $D_p$ to $D_b$;
Done;

Algorithm 1c. Detector Reduction:
Input: Boundary detector set $D_b$, detector radius $r_d$;
Output: Reduced detector set $D_r$;

Calculate Euclidean distance matrix of $D_b$;
Repeat:
    Find the closest pair of detectors $A$ and $B$;
        $dAB = distance\{A,B\}$
        If $dAB < r_s$:
            Replace $A$ and $B$ in $D_b$ by their centroid;
            Update distance matrix;
Until: $dAB \geq r_s$;
Done;
```

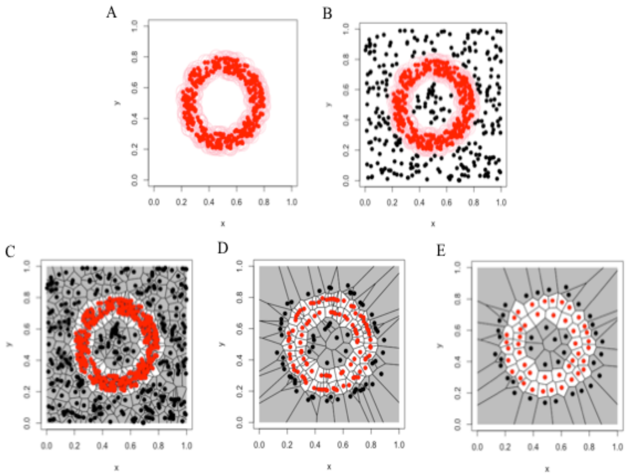

Figure 2. Phases of BDUNN. A) Training set with self radii. B) Random detector generation. C) Voronoi diagram of all training points and detectors. D) Training points and detectors which share an edge in the Voronoi diagram. E) Set reduction using detector-radii.

2.2 Boundary Set Determination

Given points from both classes, the training set S and initial detector set D_p, the algorithm attempts to determine a subset of the points in each class which adequately define the self and non-self spaces with respect to a one nearest neighbor classification, a task similar to the condensed nearest neighbor problem[13]. BDUNN uses Voronoi diagrams and Delaunay triangulations to condense the detectors and training sets [14]. The training set and detector set are reduced to only the points which share an edge in a Voronoi diagram with a point of the opposite class. A Delaunay triangulation, the dual of a Voronoi diagram, is used to compute the set of shared edges, as points which are connected by an edge in a Delaunay triangulation also share an edge in the Voronoi diagram. Therefore, any point in S or D_p which is connected in the Delaunay triangulation with a point of the other class is retained. The reduced set of detectors, D_b, and training points, S_b, form the minimum set which defines the boundary between the two classes in the Voronoi diagram. Points within these sets are guaranteed to not be redundant, as their removal from the set would effect on the boundary between the classes. These points are depicted in Figure 2d and described in Algorithm 1b. In k-dimensional problem spaces, the Delaunay triangulations are represented as simplices formed from k+1 points. In these cases, detectors and training points which occur in a simplex with a point from the other class are retained.

2.3 Boundary Set Reduction

Additionally, we establish a method for reducing the total number of detectors and training points without significantly impacting performance. We accomplish this by simply merging points of the same class which are within a certain distance of one another. Any points which are within a set detector radius r_d from one another are replaced by a single point, positioned at their centroid. The closest pair of points are merged first, and so forth, until there are no two points of the same class less than r_d apart. This process is described in Algorithm 1c, and the effect of reduction is shown in Figure 6. To avoid generating new detectors which lie within the self space, it is advised to use a detector radius no larger than the self radius.

2.4 Testing

The testing phase of the BDUNN algorithm consists of using a nearest neighbor algorithm to classify test samples as either self or non-self. This process is shown in Algorithm 2. Here we employ a variation of the NN algorithm which utilizes KD trees to improve the runtime of the algorithm [15]. A KD tree is a space-partitioning data structure in which the terminal, or leaf, nodes, correspond to individual k-dimensional points, and every non-leaf node corresponds to a splits via a separating hyperplane. This method reduces the number of points which need to be considered when finding a nearest neighbor, and in turn can speed-up the computation of nearest neighbors considerably.

```
Algorithm 2. Sample testing:
Input: Detector set $D_b$, reduced self set $S_r$, test samples $Test$;
Output: Self samples $Test_{self}$, non-self samples $Test_{NS}$;
For all $t_i \in Test$:
    If $NearestNeighbor(t_i) \in D_b$, add $t_i$ to $Test_{NS}$;
    If $NearestNeighbor(t_i) \in S_r$, add $t_i$ to $Test_{self}$;
Done;
```

3 Experiments and Results

3.1 Synthetic Datasets

3.1.1 Shapes. Synthetic 2D datasets of points distributed in various shapes were generated in order to test the effectiveness of BDUNN, and to display the effects of the different parameters. The shapes chosen for this analysis were a bar, a square, a frame, and an hourglass. Each problem space is a 2D square $[0,1]^2$, where we assume the points are randomly distributed throughout the boundaries of the shape. Figure 4 displays the results of BDUNN for each shape. A set of 300 self points randomly distributed within a defined self space were used for training. The red points represent the reduced self points, while the black points represent the detectors. In the testing phase, new data which occurs in the gray regions would be classified as anomalies, while those in the white regions would be classified as normal.

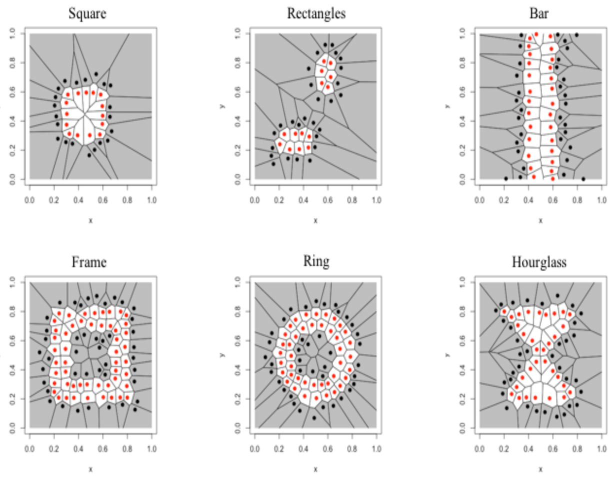

Figure 4. The different shape distributions for evaluating BDUNN.

3.1.2 Effect of Parameters. Using these 2D shapes, we can evaluate the effect of varying the different parameters of BDUNN, including the self radius, detector radius, and detector sample size. The top row of Figure 5 shows how the detection rate (DR) and false alarm rate (FAR) for each shape are effected by varying the parameters, while he bottom row of Figure 5 shows the effect these parameters have on the overall detector number (DN). The definitions of the detection rate and false alarm rate are shown in equations 1 and 2. The results of this analysis show that increasing the self radius generally decreases the DR, FAR, and number of detectors. For the investigated shapes, a self radius of 0.05 seems optimal. From these results, increasing the detector radius seems to have little effect on the detection rate, but increases the false alarm rate. Additionally, an increase in the initial detector set size increases the detector rate, slightly increases the DN, and has little effect on the DR.

$$DR = \frac{TP}{TP+FN} \quad (1)$$

$$FAR = \frac{FP}{FP+TN} \quad (2)$$

3.1.3 Detector Optimization and Reduction. The results of detector reduction for the 2D shape datasets are shown in Figure 6. The initial detector sets were reduced using a range of detector radii in order to assess the parameter's effect on the DN, DR, and FAR. These results show that this technique can be used to reduce the number of detectors by 10 to 15 percent while having a minimal effect on the DR and reducing the FAR by 1 to 2 percent.

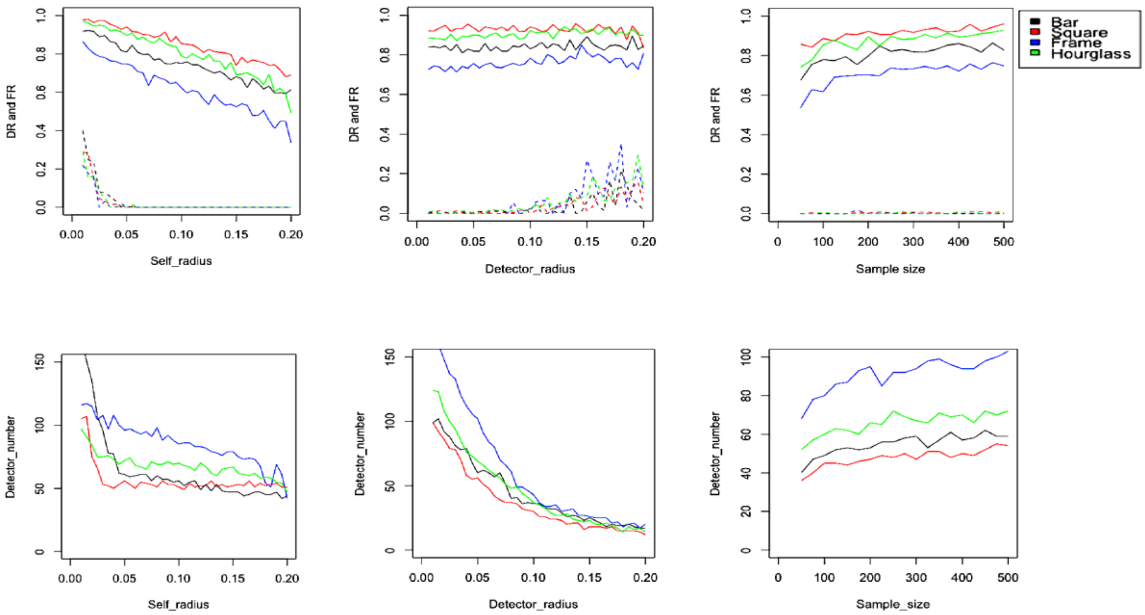

Figure 5. Top row: Effect of the self radius parameter, detector radius parameter, and detector set size on detection rate and false alarm rate. Solid lines represent DR, dotted lines represent FAR. Bottom row: Effect of the self radius parameter, centroid radius parameter, and training set size on the size of the detector set. The Bar, Square, Frame, and Hourglass shapes are adopted.

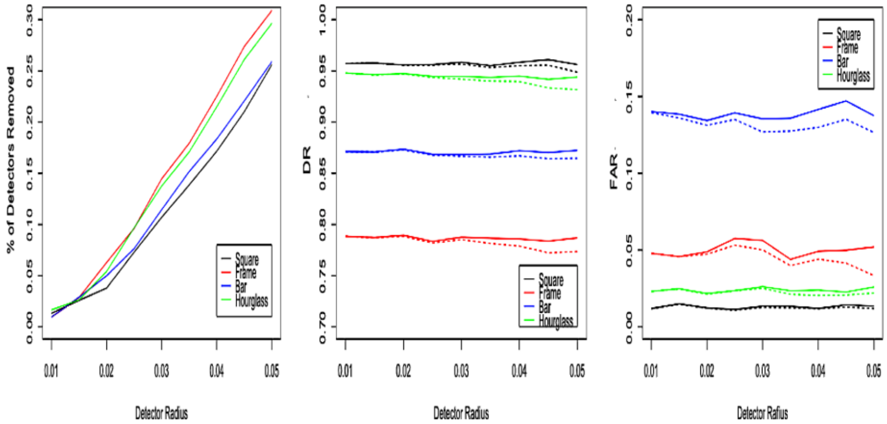

Figure 6. Effects of detector reduction in 2D shape datasets. Solid lines do not use reduction, dotted lines use reduction.

3.2 Outlier Detection Datasets

We evaluate the performance of BDUNN using four different standard datasets for anomaly detection and classification: Skin Pigmentation, Fisher's Iris data, Glass, and Haberman's survival. [16,17]. The experimental parameters of these datasets can be found in Table 1. The chosen performance metrics are detection rate (DR), false alarm rate (FAR), detector number (DN), data training time (DT), and data testing time (DTT). All data points are normalized to [0,1] using min-max normalization.

3.2.1 Iris, Glass, and Haberman's survival. We compare the performance of BDUNN to a traditional RNSA, V-detector, and a OC-SVM, using the Iris, Glass, and Haberman's Survival datasets. The OC-SVM uses RBF kernel functions, a nu of 0.5 and a gamma of 0.33. The self radius of RNSA, V-detector, and BDUNN are set to 0.1. An estimated coverage of 99% was used for RNSA and V-detector. Boundary set reduction in BDUNN was performed using a detector radius of 0.05 and an *n* of 1000.

The results of these experiments, depicted in Table 2, show that BDUNN performs similarly to V-detector, and better than RNSA, in terms of DR, FAR, and DT, but with significant improvements in DN and DTT. While the OC-SVM performed well in terms of DR, DT, and DTT, it consistently exhibited very large FARs.

3.2.2 Skin Pigmentation. We further the compare performance of BDUNN against GB-RNSA, BIORV-NSA, Vor-NSA, and a one-class support vector machine (OC-SVM) using the UCI skin segmentation dataset. Each sample reports three features, which represent the R, G and B values of the skin texture. As before, the OC-SVM uses RBF kernel functions, a nu of 0.5 and a gamma is 0.33. A self radius of 0.1 was used for RNSA, V-detector, VorNSA, and BDUNN. An estimated coverage of 99% was used for all NSAs.

The RNSA and BioRV-NSA algorithms were allowed a maximum of 3000 and 1000 detectors, respectively. BDUNN was run with a detector radius of 0.05, and an *n* of 1000. Each algorithm was run 20 times, and the means and standard deviations are reported in Table 3. The results in Table 3 for all algorithms other than BDUNN come from Zhu et al. [11].

From the results, it can be seen that BDUNN performs favorably to the other methods in terms of DR and DN. BDUNN has a slightly worse FAR than RNSA, V-detector, and VorNSA, but the differences appear minor. Although the DT and DTT results for BDUNN seem very promising, valid comparisons between the other methods cannot be made, as BDUNN and the other algorithms were run on different machines.

3.5 Analysis of RNA-seq and Ribo-seq Quality

Furthermore, we evaluate the efficacy of BDUNN for identifying contaminated RNA-seq and Ribo-seq samples by leveraging information from samples which we know are of normal quality. To assess whether BDUNN is capable of this task, we require a dataset which contains reliable features from both normal and low-quality RNA-seq and Ribo-seq samples. As acquisition, mapping, and processing of a sufficiently large number of real datasets would take considerable time, we look to use artificial datasets for training.

A set of descriptive quality features for RNA-seq and Ribo-seq samples were chosen based on the experience of data quality control experts [18]. For RNA-seq, the features are as follows:

- Mean of sample read lengths
- Standard deviation of sample read lengths
- Mean of read GC%
- Standard deviation of read GC%
- Mean coverage, calculated as the mean number of reads per base.
- Sample complexity, calculated as the number of unique alignment coordinates divided by the number of total reads
- Percentage of reads aligned in:
    - an exon
    - an rRNA region
    - an intragenic region
    - any region not mentioned above

The following quality metrics were collected for the Ribo-seq datasets:

- Mean of sample read lengths
- Standard deviation of sample read lengths
- Mean of read GC%
- Standard deviation of read GC%
- Sample periodicity, calculated as the percentage of reads in a coding sequence that align to the major frame
- Sample complexity, calculated as the number of unique alignment coordinates divided by the number of total reads
- Percentage of reads aligned in:
    - a coding sequence
    - an rRNA region
    - a tRNA region
    - any region not mentioned above

Using eighteen publication quality RNA-seq and Ribo-seq Arabidopsis Thaliana samples from Merchante et al., Hsu et al., and Liu et al., we performed bootstrap simulation to generate 2000 artificial samples consisting of the aforementioned quality metrics [2,19,20].

Table 1. Properties of the datasets used for performance evaluation.

Dataset	Record Number	Dimensions	Self sets	Non-self sets	Training set	Testing set
Skin pigmentation	245,057	3	Skin: 50,859	Non-skin: 194,198	Skin: 50	Skin: 50,809 Non-skin: 194,198
Iris	150	4	Setosa: 50	Versicolour: 50 Virginica: 50	Setosa: 25	Setosa: 25 Versicolour: 25 Virginica: 25
Glass	214	7	Normal: 206	Abnormal: 9	Normal:25	Normal: 181 Abnormal: 9
Habermans survival	306	3	Survived: 225	Died:81	Survived: 150	Survived: 50 Died: 50
Artificial RNA-Seq	2000	10	Normal Quality:1600	Low Quality: 400	Normal Quality: 50	Normal: 1550 Low: 400
Artificial Ribo-Seq	2000	10	Normal Quality:1600	Low Quality: 400	Normal Quality: 50	Normal: 1550 Low: 400

Table 2. Results for Iris, Glass, and Haberman’s survival datasets

Dataset	Algorithm	DR%		FAR%		DN		DT (s)		DTT (s)	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Iris	OC-SVM	100	0	52.05	0.11	-	-	0.007	0.002	0.0002	0.0001
	RNSA	98.74	.63	2.66	1.18	415.47	99.01	3.78	0.63	25.60	400.49
	V-detector	99.94	.25	1.31	0.82	209.91	45.83	3.12	0.85	5.56	325.50
	BDUNN	100	0	1.36	1.43	69.5	5.21	1.12	0.023	0.0003	0.0001
Glass	OC-SVM	77.83	13.30	55.04	9.24	-	-	0.02	0.004	0.008	0.001
	RNSA	85.32	.63	7.67	1.48	8414.70	906.55	38.73	13.24	3586.2	595.61
	V-detector	91.06	.27	10.98	3.22	2569.54	476.15	82.71	23.86	984.55	325.50
	BDUNN	89.63	9.13	9.19	3.56	1031.57	40.02	15.27	0.45	0.002	0.005
Haberman’s Survival	OC-SVM	77.21	7.07	61.66	6.67	-	-	0.02	0.005	0.01	0.007
	RNSA	79.42	8.33	55.4	7.04	2141.1	152.8	6.54	0.88	1481.94	302.94
	V-detector	84.90	3.23	27.41	5.25	804.07	174.66	10.19	23.86	482.09	95.45
	BDUNN	80.88	3.78	23.12	7.32	515.91	51.56	11.70	2.38	0.0044	0.003

Table 3. Results for Skin Pigmentation dataset.

Algorithm	DR%		FAR%		DN	
	Mean	SD	Mean	SD	Mean	SD
OC-SVM	99.09	.7	51.2	6.67	-	-
RNSA	98.42	.63	0.66	1.48	3000	0
V-detector	99.05	.27	1.31	1.22	469.85	174.66
BIORV -NSA	99.42	.34	3.29	2.72	1000	0
VorNSA	99.20	.16	1.48	1.49	172.25	11.06
VorNSA /MR	99.43	.24	1.56	1.37	176.90	11.96
BDUNN	99.63	.20	1.16	1.65	151.83	8.98

The samples were aligned to the Arabidopsis Thaliana genome using Tophat with default parameters [21]. Read positions were adjusted to their approximate p-site. The p-site of the ribosome holds the tRNA that is linked to the growing polypeptide chain, and plays a vital role in translation initiation, elongation, and termination [22].

In the bootstrap simulation, for each of the ten data features, 10,000 samples of size 18 are sampled with replacement from the original features/data sets. These samples are used to estimate the mean and standard deviation of the distribution of each feature, and artificial samples are subsequently generated by randomly drawing values from each of these ten feature distributions.

To create a set of anomalous low-quality samples, we synthetically adjust the feature values of 400 of the high-quality samples. Between 1 and 10 features in these samples are adjusted randomly between 10 and 40 percent. Feature values are adjusted in a realistic manner, i.e. features that would appear lower in a real contaminated dataset are adjusted in the same manner.

Tables 4 and 5 show the results of running BDUNN, RNSA, V-detector, and a OC-SVM on these artificial datasets. The parameters for each algorithm are the same as those used previously, except that the self radius used for BDUNN, RNSA, and V-detector are increased to 0.2 to account for the higher dimension and larger problem space in this experiment. The initial number of detectors for BDUNN was set to 2000.

Table 4. Results for Artificial RNA-seq dataset

Algorithm	DR		FAR		DN		DT		DTT	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
OC-SVM	99.08	0.14	70.45	5.01	-	-	0.002	0.0003	0.002	0.0003
RNSA	88.91	1.12	9.77	3.09	5188.91	8857.23	209.97	74.06	1675.33	651.04
V-Detector	91.14	1.87	6.93	2.75	1755.32	401.82	51.98	15.15	698.97	190.25
BDUNN	92.03	1.15	3.38	0.69	1345.90	200.50	39.01	4.30	0.048	0.017

Table 5. Results for Artificial Ribo-seq dataset

Algorithm	DR		FAR		DN		DT		DTT	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
OC-SVM	99.60	0.17	72.21	4.85	-	-	0.002	0.0004	0.004	0.0003
RNSA	89.94	0.99	9.15	2.67	5570.09	904.11	220.57	80.03	1756.98	655.94
V-Detector	92.11	1.50	5.05	2.03	1980.83	302.07	45.34	15.77	614.40	181.12
BDUNN	92.84	1.02	2.66	.62	1486.15	228.67	33.77	4.99	0.051	0.019

From these results, we can see that BDUNN performs favorably compared to RNSA and V-detector in all metrics, including large improvements in DN, DT, and DTT. The OC-SVM attains the highest DR but has an average FAR of over 70%.

4 IMPLEMENTATION

We have implemented BDUNN within riboStreamR, a platform for quality control, visualization, and analysis of RNA-seq and Ribo-seq data [12]. Within this platform, users can upload their own data in the form of BAM alignment files.

RiboStreamR calculates each of the previously mentioned quality metrics for the user's data, and uses BDUNN to scan for anomalous samples. BDUNN tests the user's data's quality against the previously established artificial RNA-seq and Ribo-seq datasets for Arabidopsis. The Summary Table tool within riboStreamR displays the results of performing anomaly detection within the user's datasets using BDUNN. This environment contains 9 additional customizable tools which facilitate downstream inspection of different quality metrics of the user's data. Researchers can use this information to inform further decisions on data processing and analysis steps, and to make improvements to subsequent experiments. Future work will go into expanding this platform to include artificial datasets from more species. We also plan to develop a system which automatically highlights features which are irregular within anomalous samples, and identifies reads which are potentially contaminates.

5 CONCLUSION

In this paper, we use the novel negative selection inspired one-class classifier for anomaly detection to identify contaminated RNA-seq and Ribo-seq datasets. Our algorithm, BDUNN, establishes a minimal set of detectors and a reduced training point set which define the boundaries between the self and non-self spaces, and subsequently uses a nearest neighbor algorithm to classify test observations. Our algorithm avoids many of the pitfalls of traditional negative selection methods, such as random detector generation, estimated coverage calculations, and slow testing phases. Using an artificial dataset, we show that BDUNN can identify low-quality RNA-seq and Ribo-seq samples after training with a set of normal samples. Additionally, we established BDUNN within the riboStreamR framework, which facilitates a more thorough inspection of metrics of anomalous datasets. This allows users to classify their own samples and generate a downstream visualizations of their quality metrics. While there are analysis tools which measure the quality of RNA-seq and Ribo-seq alignments [21], BDUNN, in concert with riboStreamR, employs a more informative, downstream set of quality metrics, and assesses the quality of user's data based on its similarity to known high-quality samples. In the future, we aim to further evaluate the efficacy of BDUNN on more real experimental datasets by testing its ability to identify known contaminated samples from various species.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation grant IOS1444561. The authors would like to thank Jose Alonso and Anna Stepanova for extremely valuable discussions about Ribo-seq analysis and quality control.

REFERENCES

[1] Ingolia, N. T., Ghaemmaghami, S., Newman, J. R., & Weissman, J. S. (2009). Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. science, 324(5924), 218-223.

[2] Merchante, C., Brumos, J., Yun, J., Hu, Q., Spencer, K. R., Enríquez, P., & Alonso, J. M. (2015). Gene-specific translation regulation mediated by the hormone-signaling molecule EIN2. Cell, 163(3), 684-697.

[3] Kim, J., Bentley, P. J., Aickelin, U., Greensmith, J., Tedesco, G., & Twycross, J. (2007). Immune system approaches to intrusion detection–a review. Natural computing, 6(4), 413-466.

[4] Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. Evolutionary computation, 8(4), 443-473.

[5] Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994, May). Self-nonself discrimination in a computer. In Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on (pp. 202-212). Ieee.

[6] González, F. A., & Dasgupta, D. (2003). Anomaly detection using real-valued negative selection. Genetic Programming and Evolvable Machines, 4(4), 383-403.

[7] Ji, Z., & Dasgupta, D. (2004, June). Real-valued negative selection algorithm with variable-sized detectors. In Genetic and Evolutionary Computation Conference (pp. 287-298). Springer, Berlin,

Heidelberg.

[8] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing, 2(3), 267-279.

[9] Cui, L., Pi, D., & Chen, C. (2015). BIORV-NSA: Bidirectional inhibition optimization r-variable negative selection algorithm and its application. Applied Soft Computing, 32, 544-552.

[10] Zhang, R., Li, T., & Xiao, X. (2013). A real-valued negative selection algorithm based on grid for anomaly detection. In Abstract and Applied Analysis (Vol. 2013). Hindawi.

[11] Zhu, F., Chen, W., Yang, H., Li, T., Yang, T., & Zhang, F. (2017). A Quick Negative Selection Algorithm for One-Class Classification in Big Data Era. Mathematical Problems in Engineering, 2017.

[12] Perkins, P., Mazzoni-Putman, S., Stepanova, A., Alonso, J., & Heber, S. (2019). RiboStreamR: a web application for quality control, analysis, and visualization of Ribo-seq data. BMC genomics, 20(5), 422.

[13] Hart, P. (1968). The condensed nearest neighbor rule (Corresp.). IEEE transactions on information theory, 14(3), 515-516.

[14] Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2009). Spatial tessellations: concepts and applications of Voronoi diagrams (Vol. 501). John Wiley & Sons.

[15] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509-517.

[16] Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., & Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining and Knowledge Discovery, 30(4), 891-927.

[17] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2), 179-188.

[18] DeLuca, D. S., Levin, J. Z., Sivachenko, A., Fennell, T., Nazaire, M. D., Williams, C., & Getz, G. (2012). RNA-SeQC: RNA-seq metrics for quality control and process optimization. Bioinformatics, 28(11), 1530-1532.

[19] Liu, M. J., Wu, S. H., Wu, J. F., Lin, W. D., Wu, Y. C., Tsai, T. Y., & Wu, S. H. (2013). Translational landscape of photomorphogenic Arabidopsis. The Plant Cell, 25(10), 3699-3710.

[20] Hsu, P. Y., Calviello, L., Wu, H. Y. L., Li, F. W., Rothfels, C. J., Ohler, U., & Benfey, P. N. (2016). Super-resolution ribosome profiling reveals unannotated translation events in Arabidopsis. Proceedings of the National Academy of Sciences, 113(45), E7126-E7135.

[21] Trapnell, C., Pachter, L., & Salzberg, S. L. (2009). TopHat: discovering splice junctions with RNA-Seq. Bioinformatics, 25(9), 1105-1111.

[22] Schäfer, M. A., Tastan, A. Ö., Patzke, S., Blaha, G., Spahn, C. M., Wilson, D. N., & Nierhaus, K. H. (2002). Codon-anticodon interaction at the P site is a prerequisite for tRNA interaction with the small ribosomal subunit. Journal of Biological Chemistry.