FASTCAR: Rapid alignment-free prediction of sequence alignment identity scores

Benjamin T. James[1,+], Brian B. Luczak[1,+], and Hani Z. Girgis[1,*]

[1]Bioinformatics Toolsmith Laboratory, Tandy School of Computer Science, The University of Tulsa, 800 South Tucker Drive, Tulsa, OK 74104, USA
[*]hani-girgis@utulsa.edu
[+]these authors contributed equally to this work

ABSTRACT

Pairwise alignment has been the predominant algorithm in the field of bioinformatics since its beginning. Several applications have been made in order to speed up this algorithm using heuristics, but almost all of these methods still depend on the slow quadratic alignment algorithm. Many applications utilize sequence identity scores without the corresponding alignments, e.g. scanning a database for similar sequences to a query sequence or sequence clustering. For these applications, we propose FASTCAR, which is the first machine-learning application that predicts alignment identity scores using completely alignment-free methods. Training data are produced from the input database by a generative method, mutating sequences to generate known alignment identity scores, thereby bypassing alignment algorithms. We evaluated FASTCAR, USEARCH, and BLAST by using them to scan three large-scale databases consisting of millions of sequences. FASTCAR is faster — up to 100 times — than USEARCH and BLAST. FASTCAR has reasonable sensitivity and accuracy while achieving the highest specificity, precision, and F-measure. Identity scores produced by FASTCAR are closer to the scores of the pure alignment algorithm than those produced by USEARCH and BLAST. This is the first time when the identity scores can be obtained in linear time and linear space.

Introduction

We live in an era when sequences are generated at an unprecedented rate. Analyzing these countless sequences requires efficient computational methods. Algorithms for comparing sequence similarity are among the most fundamental tools for analyzing DNA, RNA, and protein sequences.

Alignment algorithms[1,2] have been the standard methods for assessing sequence similarity over the past 40 years. Multiple software tools for alignment are available[3,4]. Applications include gene finding[5], genome assembly[6,7], function prediction[8,9], phylogenetic trees[10,11], and many other applications[12–14]. Many advancements have been made since the Needleman-Wunsch alignment algorithm was devised[2,15–17], but these new algorithms still depend on slow, quadratic, dynamic programming. This limitation is well manifested when comparing two very long sequences or when scanning a very large number of sequences. Almost all of the speed-ups are based on heuristics methods; a widely-used example of these tools is USEARCH[15].

This shortcoming of alignment algorithms has led the field to develop plenty of faster, alignment-free methods[18–38]. Multiple reviews of alignment-free methods have been published[39–46], indicating the importance and the abundance of such methods. One particular class of these methods depends on comparing two histograms of short words called k-mers, i.e. words of fixed length k. Building the histograms and comparing them can be done very efficiently. Although these alignment-free methods are very efficient, they have not been widely adapted by the field because their scores are not as intuitive or biologically relevant as the identity scores generated by alignment algorithms. However, k-mer statistics are often used in alignment tools as heuristics, such as in BLAST and USEARCH.

Often times, the identity score alone is enough; generating the alignment itself is not needed. For example, consider scanning GenBank for similar sequences to a particular gene. For another example, consider the task of clustering a large number of sequences. In these two applications — and many others — there is no need to generate the alignment; only the identity score is used.

We propose a Fast and Accurate Search Tool for Classification And Regression (FASTCAR) to predict global sequence similarity. FASTCAR can predict global identity scores in linear time and space — this is the first time that identity scores can be obtained in linear time. FASTCAR overcomes the weaknesses of alignment algorithms and those of alignment-free methods. The tool utilizes machine learning algorithms in predicting global identity scores using a small number of alignment-free, k-mer statistics. The new tool generates the identity score, which is intuitive, biologically relevant, and the standard metric in the field. Because calculating the k-mer statistics and predicting the identity score require linear time and space, FASTCAR is much

more efficient than alignment algorithms. Therefore, we expect that FASTCAR will save countless hours of computations.

For many life scientists, comparing DNA sequences that are too dissimilar will not lead to meaningful results. When scanning a sequence database, genes in animals that are highly dissimilar to humans are not useful when we are searching for potential functions to a new human gene. For this reason, it is unnecessary for our method to focus on these dissimilar sequences. Therefore, the core of FASTCAR is an adaptive, hierarchical, linear model for predicting the identity scores above a user provided threshold. This design was inspired by our earlier research. We have successfully implemented adaptive methods for ranking protein structures[47–49], locating cis-regulatory modules[50], and for identifying DNA repeats[51]. Hierarchical models were reported to perform very well in ranking the quality of predicted protein structures[47–49]. Multiple software tools we developed earlier utilize Generalized Linear Models (GLMs)[47–53]. An alignment-assisted methods that can classify similar and dissimilar sequences was developed[53]. Such a method uses alignment-free statistics to predict the alignment identity scores, on which it is trained — it is not alignment-free completely. These earlier tools justify our design choice of the adaptive, hierarchical, linear model as the core of FASTCAR.

Methods

Method overview

FASTCAR first selects randomly a small number (about 300) of the input sequences. The selected sequences are uniformly distributed with regard to length. A few semi-synthetic sequences are generated by mutating each of these sequences to generate identity scores to simulate the actual data. Since the mutated data has a known mutation rate, the identity score can be easily calculated — alignment algorithms are avoided. After that, a model can be trained to predict identity scores from few k-mer statistics. The advantage of this tool over the traditional alignment algorithm is that it uses a limited number of efficient, linear, k-mer statistics rather than the slow, quadratic dynamic programming utilized in alignment algorithms. Two components comprise this new predictive model. The first component is a classifier that recognizes whether the similarity between two sequences is above the desired threshold or below it. The second component is a regression model, which estimates the identity score of two sequences if they are above the threshold. In some applications, the user would be interested in finding similar sequences to a query sequence; however, the user is not interested in the identity scores themselves. For this reason, the user will also have the option to use the classifier only rather than the classifier followed by the regression model[53]. In other situations, the value of the threshold may not have a biological meaning; thus, the user may select to use regression without the preceding classification step.

Semi-synthetic data generation

Semi-synthetic data were generated by mutating real sequences taken from the input database to be searched. This data were then mutated using the following mutation types:

- Single point: A single nucleotide is mismatched, deleted, or inserted.
- Insertion: A block of random nucleotides with random length is inserted.
- Deletion: A block of random nucleotides with random length is deleted.
- Duplication: A copy of a substring at a random index and with random length is placed in tandem to the original substring.

This mutation process preserves the nucleotide composition of the original sequence, i.e. each nucleotide has the same percentage in the original sequence as in the mutated sequence. Because the mutation rate is known, the identity score is known and the alignment algorithm is no longer needed. This procedure is applied to generated two data sets for training and testing the predictive model.

Training and testing sets

When a database of DNA sequences is given, the first step is to sample 300 sequences to generate semi-synthetic data (3000 original-mutated sequence pairs). This set is balanced; the number of sequence pairs with identity scores above the threshold is equal to the number of pairs with identity scores below the threshold. Further, each 5% segment of the identity scores is equally represented. If there is no identity threshold, sequence pairs with balanced identity scores are generated randomly between 40% and 100%. Balancing the data set ensures that the model is not biased towards a particular segment in the identity scores. Ultimately, we want the distribution of the data set to include a wide a variety of the similar/dissimilar sequences that appear in the entire data set. Finally, the data set is divided into two mutually exclusive sets — the training set and the testing set. Next, we illustrate how these data sets are represented to the classifier and the regression model.

Calculating the k-mer statistics

Using our survey of alignment-free methods as a resource[45], we chose the following statistics: Earth Mover's Distance, Euclidean, Intersection, Kulczynski$_2$, Length Difference, Manhattan, Normalized Vectors, Pearson Correlation Coefficient, and Similarity Ratio. These statistics are chosen because they are much faster while still maintaining strong predictive power. We compute these 9 statistics then normalize each of them between 0 and 1. Some statistics represent distances and others represent similarities. We convert each distance to a similarity by subtracting the normalized distance from 1. We call these 9 statistics "single statistics". One of the primary results of our evaluation study was that squared versions or multiplicative combinations can often times outperform single statistics. For this reason, we square each of the single statistics to create 9 additional statistics. Finally, the paired statistics are generated by multiplying each unique combination of the 18 single and squared statistics.

These statistics are called features in the language of machine learning. Using a small number of features is necessary to the success of a predictive model. Therefore, we utilize a greedy feature-selection procedure, which depends on GLMs.

GLMs

The general form of the linear model is $y = F * w$ where y is the target we wish to predict. For classification, y is a vector of 1 (the sequence pair has similarity above the threshold) and -1 (the sequence pair has similarity below the threshold). For regression, y represents the identity scores. F is the feature matrix; each of its columns represents a particular statistic except the first column is all ones. The coefficients in the w vector are found using the pseudoinverse solution (Equation 1).

$$w = (F^T F)^{-1} F^T y \tag{1}$$

Now that we have the coefficients of the GLM, Equation 2 is used for making predictions.

$$\hat{y} = w_0 f_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 + w_5 f_5 \tag{2}$$

Here, $\hat{y}$ represents the predicted label, above or below the identity threshold, or the estimated identity score value for a given sequence pair. Recall that the first column of the feature matrix, f_0, is a vector of ones; therefore, f_0 is always 1, and w_0 represents the bias or the constant term. The other weights, w_1–w_5, are the learned weights associated with the five features. In the next step, we discuss how to select the best five features, i.e. statistics (f_1–f_5).

Greedy feature selection

Features are selected automatically on each input data set. A greedy algorithm is used for selecting a strong group of features without trying every possible combination. FASTCAR utilizes a GLM in selecting features that maximize the training accuracy for classification or minimize the mean error for regression.

First, we will discuss how to select features for the classifier. In each iteration of the algorithm, the classifier is trained to recognize if a pair of sequences have an identity score above or below a given threshold. The accuracy is the number of correct classifications divided by the number of sequence pairs in a set. For example, if the classifier decides that two sequences have an identity score above the threshold and the identity score generated is also above the threshold, that classification counts as a success. The first step is to calculate the accuracy of the classifier using each feature to find the best performing feature. Once found, it will be added to the best-features set and be excluded in the next iterations of the algorithm. After that, the tool will go through the remaining features one by one, attempting to combine each of these features with the best performing feature(s) found in the previous iteration(s). This algorithm is based on forward stepwise selection which is often used in statistical learning. Once this step is finished, the feature that resulted in the best training accuracy is added to the best-features set and is excluded in the subsequent iterations. When five feature pairs are selected or no further improvement is attained, the algorithm stops. Our research has shown that adding an additional feature beyond this point does not increase the accuracy enough to warrant a speed reduction[53]. The end result is a set of at most five features that have a very high training accuracy on the given data set. We also assess the accuracy on the testing set in addition to the training set to guard against over-fitting.

A similar process is used for selecting the best five features for the regression model. The performance is measured according to the mean error, i.e. the average difference between the true identity score and the predicted one — the lower the mean error, the better.

Up to this point, we discussed how the training and the testing data sets are generated and how the features are extracted and selected. Next, we discuss three modes, in which the trained models can be applied.

Prediction modes

Now that we have a strong set of features and a trained classifier and a trained regression model, the next step is to classify the remaining sequence pairs. The classifier determines whether a sequence pair falls above or below the identity score threshold, which is provided by the user. Pairs classified below the threshold are removed; the remaining sequence pairs are sent to

the regression model to predict their identity scores. Alternatively, there are cases where classification only is desired[53]; it is possible to disable regression. On the other hand, if identity scores throughout the entire range of sequences are desired, regression alone can be performed, not excluding any sequence pairs. At this point, the description of FASTCAR's method is complete.

Related tools

We chose 2 widely-used tools, USEARCH[15] and BLAST[16], to compare to FASTCAR. Additionally, we needed a ground truth, on which the three tools can be evaluated. For this purpose, we chose needleall from EMBOSS[3] because of its ability to do an "all versus all" global alignment. The program needleall does not use heuristics like the ones used in the other two tools; since it is much slower than USEARCH and BLAST, our analysis does not include needleall. While BLAST is designed for local alignment, it also generates global alignments, as they are a special case of local alignment. It is possible to get global alignment scores by manipulating parameters and filtering out BLAST results. We did consider using another tool known as CABLAST[17]. Taking over a week to run on a data set, we deemed this "proof of principle" impractical to run due to its time requirement.

We have just finished discussing our choice of the related tools. Next, we discuss how we constructed 3 large-scale data sets, on which FASTCAR, USEARCH, and BLAST are evaluated.

Data Sets

We surveyed an array of nucleotide databases in search for large-scale data sets, on which the three tools are evaluated. These data sets include a microbial data set, we call Costello[11], which contains 1,071,335 sequences of length 200 base pairs (bp) to 400 bp. Another data set, deemed the Keratin data set, includes 5,220,536 sequences with sequence lengths ranging between 1700 bp and 3300 bp. The third data set — the P27 set — consists of 7,990,947 sequences between 1500 bp and 4000 bp.

Query sequences were selected for each data set. For the Costello set, 1189 query sequences were selected with varying lengths between 200 bp and 400 bp. To ensure that the data was evenly spread, a limit was placed on how many sequences could be chosen per 10 nucleotide lengths (10 from 200–209, 10 from 210–219, etc). The keratin query sequence is the NM_002283.3 — *Homo sapiens* keratin 85 (KRT85), transcript variant 1, mRNA sequence. The p27 query sequence is the NM_004064.4 — *Homo sapiens* cyclin dependant kinase inhibitor 1B.

The ground truth, i.e. similar sequences to the query sequence(s), of the Costello data set consists of all sequences that at least 97% identical to any query sequence. This ground truth set consists of 11,587,830 sequences. Both the p27 and keratin ground truth were found by searching the NCBI database[54]. The search parameters for keratin were: "srcdb_refseq[PROP] AND Keratin NOT Homo Sapien". The range was restricted to animal sequences between 1700 and 3250 nucleotides in length. At the time of writing this paper, this query resulted in 6669 sequences. To find similar sequences to the p27 query, we searched: "srcdb_refseq[PROP] AND cyclin dependant kinase inhibitor 1B NOT Homo Sapien". The range was restricted to animal sequences between 2000 and 3000 nucleotides in length. This query resulted in 132 sequences at the time of this writing. After that, sequences that have less than 70% identity with the query sequence was removed from the ground truth, resulting in 56 and 67 sequences similar to the keratin and the p27 query sequences.

Up to here, we described the computational principles of FASTCAR. Then the details of the related tools and the evaluation data sets were illustrated. Next, we specify how the tests were conducting to guarantee fair comparisons.

Testing Conditions

All tests were run on the same computer, Dell Precision Tower 5810, 10-core Xeon E5-2630 CPU, and 32 GB RAM. The database files were broken into 20 equally-sized parts to be processed by the tools that do not support multithreads. Next, we ran needleall, which is the global alignment tool, in parallel on the twenty files. It took 44 hours on the Costello data set, 72 hours on the Keratin data set, and 41 hours on the P27 data set. These times were the total times of the 20 concurrent needleall processes. We also had to divide the databases because the public version of USEARCH has a file size limit. For BLAST, the database was generated using one thread — the program that generates the database does not support multithreads. Then BLAST was executed on the query sequences using 20 threads. FASTCAR was run in its default mode, using classification followed by regression, on each data set using 20 threads. This generated an alignment table of pairs above the cutoff value with the corresponding predicted identity value.

At this point, we described the methods and the data used in our study. Next, the evaluation criteria are specified. After that, the performances of the 3 tools on the 3 data sets are reported and discussed.

Results

The main contribution of this research is the FASTCAR software tool (Supplementary data set 1).

Evaluation measures

We evaluated FASTCAR, USEARCH, and BLAST using the following 8 evaluation measures:

- **Sensitivity:** Sensitivity is the rate of true positive to the combined true positive and false negative values. It measures the ability of a tool to identify true positives from the data set — the more positive points found, the better.
- **Specificity:** Specificity is the rate of true negative values found, that is, correctly predicted negatives, to the actual number of negative values found in the data set.
- **Accuracy:** Accuracy is the percentage of correct predictions, giving a measure that captures both the negative predictive power and positive predictive power. In many cases, accuracy is the main statistic to be measured. However, since search is the application of interest, accuracy is not paramount to success because of the large size of negative values.
- **Precision:** Precision is the ratio of true positives to true positives and false positives. Precision measures the relevancy of returned results, since it rates the true positives to the total positive labels. This measure is very important when validating the results experimentally is considered.
- **F-measure:** F-measure combines sensitivity and precision by taking the harmonic mean between them. Combining some elements of both, it allows for one value to capture the values from both precision and sensitivity, the two metrics which are most important for this research. This allows for a single measure that can capture multiple aspects of classification.
- **Mean error:** Mean error is used in regression analysis to measure how close, on average, the predicted value is to the actual value. The mean error is the average absolute difference between the predicted value and the actual value. The usage of the mean error also allows a comparable benchmark for the error. Additionally, it can be used as an expected margin of error.
- **Time:** Time reported is the wall clock time, as multi-threaded applications are best estimated using real time.
- **Memory:** Max memory used by a tool is measured, as the memory requirement is set by the maximum amount.

Next, we report the performance of the 3 tools as measured according to these 8 criteria.

Evaluations on large-scale data sets

Using the 3 data sets described earlier, FASTCAR, BLAST, and USEARCH were evaluated by querying some sequence(s) against a database of sequences. Tool performances were measured using the 8 criteria. Table 1 shows these results.

Across the different data sets, FASTCAR was the most precise and had the highest F-measure. On the Keratin data set, FASTCAR's F-measure was comparable to BLAST (0.87 versus 0.85), whereas every other F-measure was much lower than that achieved by FASTCAR — USEARCH resulted in 2 to 50 times worse F-measure. In terms of specificity, FASTCAR achieved the highest specificity, which is quite important due to the composition of these tests. Since these query sequences are

Data Set	Tool	Sensitivity	Specificity	Accuracy	Precision	F-measure	Mean Error (%)	Time (s)	Memory (KB)
Costello	USEARCH	0.949420	0.980754	0.965087	0.311714	0.469335	11.235243	4568	2678412
	BLAST	0.106299	0.998922	0.552611	0.475165	0.173734	7.043712	5907	225436
	FASTCAR	0.945091	0.999118	0.972105	0.907759	0.926049	0.963210	79	242408
Keratin	USEARCH	1.000000	0.999756	0.999878	0.043511	0.083393	59.510276	9083	687084
	BLAST	0.928571	0.998034	0.963302	0.800000	0.859504	9.146092	279	3144564
	FASTCAR	0.803571	0.999697	0.901634	0.957446	0.873786	4.129757	189	1516724
P27	USEARCH	1.000000	0.999169	0.999584	0.010136	0.020068	62.934551	21490	1083524
	BLAST	0.910447	0.738461	0.824454	0.782051	0.841379	9.808807	452	4975904
	FASTCAR	0.925373	1.000000	0.962686	1.000000	0.961240	4.472948	285	1556204

Table 1. Evaluations of USEARCH, BLAST, and FASTCAR on the ground truth. Alignment scores were generated with needleall. The mean error is displayed as a percentage, and all other values are displayed on a scale of 0 to 1. All values have been rounded to 6 decimal places. These evaluations were conducted on three data sets: Costello, P27, and Keratin. The Costello set includes 1,071,335 sequences of length 200 bp to 400 bp. We searched for sequences similar to the query sequences (at least 97% identity). The Keratin data set has 5,220,536 sequences, ranging between 1700 bp and 3300 bp. There is only one query sequence for the Keratin data set; we searched for sequences that at least 70% identical to the query sequence. The P27 set consists of 7,990,947 sequences between 1500 bp and 4000 bp. Similar to the Keratin set, there is one query sequence. Sequences with at least 70% identity to the query sequence are considered true positives.

selective, there will be many negatives in the data. In fact, on the microbial (Costello) data set, there were over 1 billion true negative values. Therefore, since most negatives would be correctly predicted, small changes in specificity matter much more than changes in sensitivity. FASTCAR sensitivity was reasonable; it was comparable to the best performing tool on the Costello set (0.95), the lowest on the Keratin set (0.80), and the second best on the P27 set (0.93). The overall accuracy of FASTCAR was also reasonable (0.90–0.97).

FASTCAR had the lowest mean error on any of the data sets, having less than 5% mean error on all data sets, further showing the ability of this new method. In fact, BLAST had 9% mean error, on average. USEARCH had 44% mean error on average; this is not surprising considering how many false positives were found, as evidenced by its precision. These results demonstrate that the identity scores produced by FASTCAR are better — relative to the identity scores produced by the pure alignment algorithm — than those produced by BLAST and USEARCH.

Regarding memory, the max amount of memory used, 1.48 GB on the P27 data set, is readily available on modern hardware. Moreover, once the classifier and the regression model are trained, memory is kept low as sequences are processed as they are read in. FASTCAR has the lowest time requirement; much of the time is taken to train the models. This method is scalable and will take even less time using more threads. Specifically, on the Costello data set, FASTCAR was 87 times faster than USEARCH and 113 times faster than BLAST. The other data sets — only 1 query sequence versus over 1000 query sequences on the Costello set — yielded smaller (1.1–75 times faster) differences in performance.

Discussion

A number of machine learning algorithms for classification and regression are available. These algorithms include Support Vector Machines (SVM) and Artificial Neural Networks (ANN). We were attracted to the GLM because it is a parameter-free model, which is well suited to the idea of adaptive training. We chose the GLM primarily because of its time efficiency, which is due to the absence of parameters to be optimized. Both the SVM and the ANN can be highly accurate if given enough time to train. However, they require several variables that need to be optimized which does not fit well into our adaptive training idea. On the other hand, the GLM only requires calculating the pseudo-inverse solution to find the linear coefficients. This operation is much cheaper than searching for optimal parameters required by the other algorithms. Our experiments show that the hierarchical GLM can obtain comparable results to SVM and ANN — without parameter optimization however.

Conclusion

A very important algorithm in bioinformatics, pairwise alignment, is slow. Fast alternatives such as k-mer distances produce scores that do not have relevant biological meanings as the identity scores produced by alignment algorithms. We developed a novel software tools, FASTCAR, for estimating identity scores by training a classifier and regression model to predict identity scores using few, efficient, k-mer statistics. Training these models is done with a novel method of generating sequences with known identity scores, allowing for alignment-free prediction of alignment identity scores. This is the first time an identity score is obtained in linear time and space.

Data availability

The C++ source code of FASTCAR is available at https://github.com/TulsaBioinformaticsToolsmith/FASTCAR and as the Supplementary Data Set 1.

References

1. Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* **48**, 443–453 (1970).
2. Gotoh, O. An improved algorithm for matching biological sequences. *J Mol Biol* **162**, 705–708 (1982).
3. Rice, P., Longden, I. & Bleasby, A. EMBOSS: the european molecular biology open software suite. *Trends Genet.* **16**, 276–277 (2000).
4. Rizk, G. & Lavenier, D. GASSST: global alignment short sequence search tool. *Bioinformatics* **26**, 2534–2540 (2010).
5. Wang, X. *et al.* Emergence of a novel mobile colistin resistance gene, mcr-8, in NDM-producing klebsiella pneumoniae. *Emerg Microbes Infect* **1** (2018).
6. Butler, J. *et al.* ALLPATHS: de novo assembly of whole-genome shotgun microreads. *Genome Res* **18**, 810–820 (2008).
7. Luo, R. *et al.* SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience* **1**, 1–6 (2012).

8. Peled, S. *et al.* De-novo protein function prediction using dna binding and rna binding proteins as a test case. *Nat Commun* **7** (2016).

9. Carradec, Q. *et al.* A global ocean atlas of eukaryotic genes. *Nat Commun* **9**, 373 (2018).

10. Naushad, S. *et al.* Comprehensive phylogenetic analysis of bovine non-aureus staphylococci species based on whole-genome sequencing. *Front Microbiol* **7** (2016).

11. Costello, E. K. *et al.* Bacterial community variation in human body habitats across space and time. *Science* **326**, 1694–1697 (2009).

12. Cabili, M. N. *et al.* Integrative annotation of human large intergenic noncoding rnas reveals global properties and specific subclasses. *Genes dev* **25**, 1915–1927 (2011).

13. Eaton, D. A. R. Pyrad: assembly of de novo radseq loci for phylogenetic analyses. *Bioinformatics* **30**, 1844–1849 (2014).

14. Dong, X. *et al.* Fast and simple analysis of MiSeq amplicon sequencing data with MetaAmp. *Front Microbiol* **8** (2017).

15. Edgar, R. C. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* **26**, 2460–2461 (2010).

16. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J Mol Biol* **215**, 403–410 (1990).

17. Loh, P.-R., Baym, M. & Berger, B. Compressive genomics. *Nat Biotechnol* **30**, 627 (2012).

18. Blaisdell, B. E. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proc Natl Acad Sci U S A* **83**, 5155–5159 (1986).

19. Zharkikh, A. A. & Rzhetsky, A. Y. Quick assessment of similarity of two sequences by comparison of their l-tuple frequencies. *BioSystems* **30**, 93–111 (1993).

20. Wu, T.-J., Hsieh, Y.-C. & Li, L.-A. Statistical measures of dna sequence dissimilarity under markov chain models of base composition. *Biometrics* **57**, 441–448 (2001).

21. Almeida, J. S. & Vinga, S. Universal sequence map (USM) of arbitrary discrete sequences. *BMC Bioinforma.* **3**, 6 (2002).

22. Lippert, R. A., Huang, H. & Waterman, M. S. Distributional regimes for the number of k-word matches between two random sequences. *Proc Natl Acad Sci U S A* **99**, 13980–13989 (2002).

23. Pham, T. D. & Zuegg, J. A probabilistic measure for alignment-free sequence comparison. *Bioinformatics* **20**, 3455 (2004).

24. Kantorovitz, M. R., Robinson, G. E. & Sinha, S. A statistical method for alignment-free comparison of regulatory sequences. *Bioinformatics* **23**, i249 (2007).

25. Dai, Q., Yang, Y. & Wang, T. Markov model plus k-word distributions: a synergy that produces novel statistical measures for sequence comparison. *Bioinformatics* **24**, 2296 (2008).

26. Reinert, G., Chew, D., Sun, F. & Waterman, M. S. Alignment-free sequence comparison (i): Statistics and power. *J Comput. Biol* **16**, 1615–1634 (2010).

27. Sims, G. E., Jun, S.-R., Wu, G. A. & Kim, S.-H. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proc Natl Acad Sci U S A* **106**, 2677–2682 (2009).

28. Costa, A. M., Machado, J. T. & Quelhas, M. D. Histogram-based DNA analysis for the visualization of chromosome, genome and species information. *Bioinformatics* **27**, 1207 (2011).

29. Liu, X. *et al.* New powerful statistics for alignment-free sequence comparison under a pattern transfer model. *J Theor Biol* **284**, 106–116 (2011).

30. Zhang, Y. & Chen, W. A new measure for similarity searching in dna sequences. *MATCH: Commun. Math. Comput. Chem.* **65**, 477–488 (2011).

31. Göke, J., Schulz, M. H., Lasserre, J. & Vingron, M. Estimation of pairwise sequence similarity of mammalian enhancers with word neighbourhood counts. *Bioinformatics* **28**, 656 (2012).

32. Ren, J., Song, K., Sun, F., Deng, M. & Reinert, G. Multiple alignment-free sequence comparison. *Bioinformatics* **29**, 2690 (2013).

33. Ghandi, M., Lee, D., Mohammad-Noori, M. & Beer, M. A. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput. Biol* **10**, e1004035 (2014).

34. Haubold, B. Alignment-free phylogenetics and population genetics. *Brief Bioinform* **15**, 407 (2014).

35. Leimeister, C.-A., Boden, M., Horwege, S., Lindner, S. & Morgenstern, B. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics* **30**, 1991 (2014).

36. Pinello, L., Lo Bosco, G. & Yuan, G.-C. Applications of alignment-free methods in epigenomics. *Brief Bioinform* **15**, 419 (2014).

37. Borozan, I., Watt, S. & Ferretti, V. Integrating alignment-based and alignment-free sequence similarity measures for biological sequence classification. *Bioinformatics* **31**, 1396–1404 (2015).

38. Liao, W. *et al.* Alignment-free transcriptomic and metatranscriptomic comparison using sequencing signatures with variable length markov chains. *Sci Rep* **6** (2016).

39. Vinga, S. & Almeida, J. Alignment-free sequence comparison–a review. *Bioinformatics* **19**, 513 (2003).

40. Vinga, S., Carvalho, A. M., Francisco, A. P., Russo, L. M. & Almeida, J. S. Pattern matching through Chaos Game Representation: bridging numerical and discrete data structures for biological sequence analysis. *Algorithms Mol Biol* **7**, 10 (2012).

41. Bonham-Carter, O., Steele, J. & Bastola, D. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. *Brief Bioinform* **15**, 890 (2014).

42. Song, K. *et al.* New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Brief Bioinform* **15**, 343 (2014).

43. Vinga, S. Editorial: Alignment-free methods in computational biology. *Brief Bioinform* **15**, 341–342 (2014).

44. Chattopadhyay, A. K., Nasiev, D. & Flower, D. R. A statistical physics perspective on alignment- independent protein sequence comparison. *Bioinformatics* **31**, 2469–2474 (2015).

45. Luczak, B. B., James, B. T. & Girgis, H. Z. A survey and evaluations of histogram-based statistics in alignment-free sequence comparison. *Brief Bioinform* bbx161 (2017).

46. Zielezinski, A., Vinga, S., Almeida, J. & Karlowski, W. M. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biol* **18**, 186 (2017).

47. Girgis, H. Z. & Corso, J. J. Stp: the sample-train-predict algorithm and its application to protein structure meta-selection. Tech. Rep. 16, The State University of New York at Buffalo (2008).

48. Girgis, H. Z. *Machine-learning-based meta approaches to protein structure prediction*. Ph.D. thesis, The State University of New York at Buffalo (2008).

49. Girgis, H. Z., Corso, J. J. & Fischer, D. On-line hierarchy of general linear models for selecting and ranking the best predicted protein structures. In *Conf Proc IEEE Eng Med Biol Soc*, 4949–4953 (2009).

50. Girgis, H. Z. & Ovcharenko, I. Predicting tissue specific cis-regulatory modules in the human genome using pairs of co-occurring motifs. *BMC Bioinforma.* **13**, 25 (2012).

51. Girgis, H. Z. Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. *BMC Bioinforma.* **16** (2015).

52. Girgis, H. Z. & Sheetlin, S. L. MsDetector: toward a standard computational tool for DNA microsatellites detection. *Nucleic Acids Res* **41**, e22 (2013).

53. James, B. T., Luczak, B. B. & Girgis, H. Z. MeShClust: an intelligent tool for clustering DNA sequences. *Nucleic Acids Res* gky315 (2018).

54. O'Leary, N. A. *et al.* Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res* **44**, D733–D745 (2016).

Acknowledgements

The authors would like to thank Alexander Baumgartner for his help with data processing and coding the module for generating mutated sequences. This research was supported by internal funds provided by the College of Engineering and Natural Sciences and the Faculty Research Grant Program at the University of Tulsa. The research results discussed in this publication were made possible in part by funding through the award for project number PS17-015 from the Oklahoma Center for the Advancement of Science and Technology.

Author contributions statement

H.Z.G. conceived the idea and designed the experiments. B.B.L. developed the prototype and designed the experiments. B.T.J. developed the distribution version, conducted the experiments, and produced the results. All authors wrote and reviewed the manuscript.

Competing interests

None declared.