DNA Read Feature Importance Using Machine Learning for Read Alignment Categories

Jacob S. Porter(✉)

Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA, USA
jsporter@virginia.edu

Abstract. An empirical understanding of how DNA read features affect read alignment quality categories is useful in designing better read mapping and alignment software, read trimmers, and sequence masks. Many programs appear to use arbitrarily chosen features that are putatively relevant to DNA alignment quality. Machine learning gives a ready way to empirically assess a variety of features and rank them according to their importance. Sequence complexity features such as run length distribution, DUST, and entropy, and quality measures from the DNA read data were used to predict read alignment quality categories on Ion Torrent and Illumina data sets using both bisulfite-treated and untreated short DNA reads. Run length mean and variance did as well or better than the DUST score and entropy, even though several programs use the DUST score and entropy. Sequence compression features performed poorly. Predictive accuracy of the models had F1-scores between 0.5–0.95 indicating that the feature set can fairly well predict alignment categories.

Keywords: DNA alignment · Machine learning · Sequence complexity

1 Introduction

A DNA read sequencer produces DNA fragments called reads. A DNA read is a string over the alphabet $\{A, C, T, G, N\}$ corresponding to the nucleotide bases and the N wildcard character. DNA sequence alignment programs map these DNA reads to a reference genome. This process can be error prone as the DNA fragments may not match a portion of the reference genome perfectly because of natural variation and mutation or because of sequencing error [24,30].

DNA sequence mapping software that is used for regular untreated reads includes Bowtie2 [9], BWA [11], and BFAST [6]. Mapping software for bisulfite-treated reads must adjust for the bisulfite treatment, and such software includes Bismark [8], BWA-Meth [17], and BisPin [22]. Bisulfite treatment is used to search for covalent modification of cytosine in DNA. There are many more examples of alignment and mapping software.

S. K. Jha et al. (Eds.): ICCABS 2020, LNBI 12686, pp. 3–14, 2021.
https://doi.org/10.1007/978-3-030-79290-9_1

Insight into which read features are important to alignment quality categories could lead to more effective alignment software, read trimmers, masking algorithms, and so on. I used machine learning to study which numerical features of short DNA reads are predictive of read alignment quality categories. These features include metrics of quality, sequence complexity, and sequence compressibility.

2 Related Work and Motivation

I used machine learning to predict up to four read alignment categories as discussed in Sect. 3.2. Four classifiers were trained for each data set for each mapping software.

My purpose wasn't to use machine learning to predict alignment categories since learning the categories can be done simply by running the alignment software. My purpose was to explore features relevant to read alignment quality. However, simple machine learning approaches could be used to efficiently filter out predicted low quality reads, and so forth. This is explored in Sect. 4.4.

Assessing feature relevance allows for good decisions to be made in their use in bioinformatics software. Trimming and masking software such as InfoTrim and Cookiecutter use sequence complexity [21,28]. The bisulfite software BatMeth has a low complexity filter using Shannon entropy [14], and BLAST uses the DUST score for complexity masking [1,15]. The DUST score measures trinucleotide frequency. The sequence complexity measures chosen for these programs appear to be arbitrarily chosen or chosen for convenience. Compression software has been used to determine sequence similarity [31]. A thorough evaluation of such measures with machine learning gives an empirical rationale for the choice of the sequence complexity measures.

Other work has used machine learning to predict DNA function from DNA sequence identity [13] and methylation loci from DNA reads [32]. My own study found that Shannon entropy corresponds to read alignment categories [20]. A study found that genome complexity relates to read mapping quality [19], but my study examines reads rather than genomes.

3 Methods

Reads were mapped using typical alignment programs, and standard machine learning approaches were used to predict alignment categories. Custom Python program were used for feature extraction.

3.1 Data Acquisition and Read Mapping

Six data sets of three million reads each were downloaded from the sequence read archive (SRA) [10] at https://www.ncbi.nlm.nih.gov/sra. This data represents a variety of bisulfite-treated and regular short DNA reads. Bisuflite- treated reads

are used to search for epigenetic cytosine covalent modifications, and these reads were included since aligning these reads can be challenging with low alignment quality [20,29]. The data includes quality information that gives the probability that the base was called correctly. No trimming was performed.

The data includes DNA reads generated from the Illumina platform and the Ion Torrent platform. Ion Torrent sequencers create variable length reads from 100–300 base pairs with greater error in homopolymer runs [23]. Illumina technology creates reads of uniform length that can be a bit shorter than Ion Torrent reads. Illumina technology is much more common, and it can generate 'paired-end' reads. Table 1 shows a summary of the data used. This data set represents a variety of sequencing technologies and platforms, so it useful for generalizing the results.

Table 1. Summary of the DNA read data.

SRA #	Type	Platform	Len	Species	Mappers
ERR2562409	BS	Illumina	90	Mouse	BisPin, Bismark
SRR1104850	BS	Illumina	200	Human	BisPin
SRR5144899	BS	Illumina	101	Human	BisPin, Bismark
SRR1534392	BS	Ion Torrent	Varies	Mouse	BisPin, Tabsat
SRR2172246	Reg	Illumina	76	Human	BFAST, Bowtie2
ERR699568	Reg	Ion Torrent	Varies	Mouse	BFAST-Gap, TMAP

One or two read mapping and alignment programs were used to map and align each data set to the reference genome. The GRCh38.p9 human reference genome was used, and the GRCm38.p5 mouse reference genome was used. These genomes can be downloaded from the NCBI (National Center for Biotechnology Information) data store at https://www.ncbi.nlm.nih.gov/genome. Table 1 indicates which read mapping programs were used with which data set. Thus, eleven alignment files were created to do machine learning.

For bisulfite-treated Illumina reads, BisPin [22] and Bismark [8] were used on their default settings. A primary and secondary index was used with BisPin with rescoring turned off. Bismark is a popular read mapper for bisulfite-treated reads, and it uses Bowtie2 [9] to do alignments. BisPin is a versatile read mapper that has good accuracy with a variety of data [22]. Bismark did not return any mapped reads for data set SRR1104850, so only BisPin was used there. For Illumina regular untreated reads, BFAST (BLAT-like Fast Accurate Search Tool) [6] and Bowtie2 [9] were used.

For bisulfite-treated Ion Torrent reads, BisPin and Tabsat were used. BisPin was used with default settings appropriate to Ion Torrent reads as found in [22]. Tabsat [16] uses Bismark's Perl code and the Ion Torrent read mapper TMAP (Torrent Mapping Alignment Program https://github.com/iontorrent/TMAP). For regular untreated Ion Torrent reads, BFAST-Gap [22] and TMAP were used. TMAP was used with the map4 algorithm.

3.2 Feature and Class Extraction

Feature Extraction. For each DNA read, 67 numerical features were created that comprised sequence complexity, read content, compressibility, and quality. Reads with N's in them were excluded from the analysis as their presence interferes with the sequence complexity measures; however, N's are highly relevant to read mapper performance as an N means an ambiguous nucleotide base that can match to any nucleotide base in the reference genome.

The sequence complexity features included run length metrics, the DUST score, entropy, $D_k(a)$, $R_k(a)$, Bzip2 compressibility, and LZMA compressibility. Compressibility is related to sequence complexity [12], and it has been used to measure DNA sequence similarity [31].

The run length distribution was computed. A run is a substring of the DNA string comprised of the same base. The length of the run is the number of bases in that run. For example, "AATCCC" has a length 2 run of A's, a length 1 run of a T, and a length 3 run of C's. The mean, variance, and maximum of this distribution were used as features.

The DUST score is a sequence complexity metric based on tri-nucleotide frequency [15]. A search of the literature did not reveal why this metric is called DUST. Given that a is a sequence of n characters from $\mathcal{A} = \{A, C, T, G\}$, a *triplet* is a substring of length 3, and there are 64 possible triplets. The space of triplets is $\mathcal{R}$. There are $n-2$ non-unique triplets in a for $n > 2$. If $c_t(a)$ is the number of times triplet t occurs in a, then the DUST score is

$$\frac{\sum_{t \in \mathcal{R}} c_t(a)(c_t(a) - 1)/2}{n - 3}.$$

The DUST score was normalized to be between 0 and 1 by dividing it by $\frac{(n-2)(n-3)/2}{n-3}$, the maximum DUST score.

Shannon entropy [26] is a sequence complexity measure common in machine learning. If $f_b(a)$ is the frequency of character b in sequence a, then entropy is given by

$$-\sum_{b \in \mathcal{A}} f_b(a) \log_2(f_b(a)).$$

For each $b \in \mathcal{A}$, the base frequency $f_b(a)$ was included as a feature. This captures sequence content related features.

The metrics $D_k(a)$ and $R_k(a)$ are found in [19]. The function $g(x)$ gives the number of times that the substring x occurs in a. $D_k(a)$ measures the rate of distinct substrings. Given a number k for the substring length, $D_k(a)$ is defined as

$$D_k(a) = \frac{|\{x : g(x) > 0 \,|\, |x| = k, x \in a\}|}{|a| - k + 1}.$$

$R_k(a)$ measures the rate of repeats, and it is

$$R_k(a) = \frac{\sum_{g(x)>1, |x|=k} g(x)}{|a| - k + 1}.$$

$R_k(a)$ and $D_k(a)$ for $k = 2, 3, 4, 5$ were used. These metrics can be computed in linear time and space using suffix arrays [19].

The Bzip2 and LZMA implementations in Python3 were used to measure the compressibility of the DNA sequence. The number of bytes returned by the compression algorithms was divided by the length of the uncompressed sequence to get a compressibility metric.

Quality related features were computed from the probability measures given with the DNA reads. This included the mean, variance, skewness, maximum, and minimum. Since the probabilities are arranged in a sequence, the difference between each probability was computed, and these values were averaged and included as a feature.

The preceding features were computed for the whole read. For each third of the DNA sequence, each of the preceding features except for $D_k(a)$, $R_k(a)$ and the run length metrics, were computed and included in the feature set as well.

Label Extraction. This problem was modeled as a classification problem since every read mapping program gives some indication of read alignment uniqueness. For each read in an alignment file, the FLAG field of the SAM alignment record was inspected to assign the read into one of four classes: uniquely mapped, ambiguously mapped, unmapped, and filtered.

A read is uniquely mapped if the read mapping software reports that there is a unique best scoring alignment for that read. A read is ambiguously mapped if there are multiple best scoring locations. An unmapped read maps to no location, and a filtered read has an alignment score below some program specific threshold. Not every read mapper reports every class, so some classes were excluded for some read mappers. One of these classes is predicted for each read.

3.3 Machine Learning Methods

Python3 with scikit-learn 0.19.1 [18] was used to do machine learning. Four machine learning classifiers were used to assess predictive accuracy: random assignment (Rand), random forest (RF), multi-layer perceptron neural network (MLP), and logistic regression (LR). All features were centered and scaled using the StandardScaler in scikit-learn for each classifier for each data set. Because there were eleven alignment results, eleven machine learning models were created for each classifier type and for each software for a total of 44 trained classifiers.

A random classifier (Rand) was trained. This classifier learns the proportion of classes in the training data and simply guesses a class with probability equal to the proportion that it learned for that class. This classifier was used to determine if the other three classifiers were better than random guessing.

A random forest is an ensemble of decision trees. At each level in the tree, a value for a feature is used to split the level. The leaves are labeled with classes. An MLP is a neural network with hidden layers that linearly combine previous layers and apply an activation function. The ReLU activation function was used. The output of the network is a vector of probabilities for each class. Logistic regression is a binary statistical model that uses a log-odds ratio. It was used

with the l2 norm. A binary problem was used for each class, and the class with the maximum probability was reported as the predicted class [5].

Bayesian optimization with scikit-optimize was used to do hyperparameter tuning with three-fold cross-validation. Bayesian optimization strategically selects a point in the hyper-parameter space based on the performance of previously selected hyperparameters [27]. The GP-hedge acquisition function was used, and twenty-five iterations were performed.

Random forest hyperparameters max depth and max features were optimized. After some experiments, a MLP architecture with four hidden layers of size 30, 20, 15, and 10 was chosen, and the regularization parameter alpha was optimized. Logistic regression uses a regularization parameter that was optimized.

Three-fold cross validation was used to train on 2.5 million training examples. Approximately 500,000 reads were held-out as test data to assess model predictive performance. Reads with N's were excluded from the analysis. Cohen's kappa metric was used for model selection since it is supposed to perform better than accuracy with rare classes [3]. Precision, recall, and the F1-score (the harmonic mean of precision and recall) were computed for each class for each data set. These were used to assess predictive performance on the held-out test data.

The source code and a results spreadsheet can be found at:
https://github.com/JacobPorter/AlignmentML.

4 Results

Models' F1-scores ranged from 0.5–0.95. The most important features were sequence complexity features. Quality and compression features were less important. A read filter based on trained machine learning models found improvements in some data.

4.1 Model Accuracy

The F1-score was computed for each class, and then each class's F1-score was averaged to assess model predictive performance. These results are presented in Table 2. The mapping classes are represented as letters (U = Unique, A = Ambig, N = Unmapped, F = Filtered). All models performed better than random guessing. Random forest models always had the highest F1-score, and logistic regression was generally the worst with the slowest training time. The MLP had the fastest training time of the three.

Predictive accuracy was generally good for uniquely mapped reads and poor for ambiguously mapped reads. Predictive accuracy for unmapped and filtered reads ranged from poor to fair. The number of uniquely mapped reads could be as high as approximately 90% of the data, and other classes could only be a few percent of the data. This makes non-unique classes rare and prediction difficult.

Table 2. Average class F1-score for each data set.

Data	Software	Classes	Rand	RF	MLP	LR
ERR2562409	Bismark	UAN	0.40	0.94	0.84	0.80
ERR2562409	BisPin	UANF	0.41	0.95	0.85	0.81
ERR699568	BFAST-Gap	UANF	0.86	0.91	0.90	0.90
ERR699568	TMAP	UA	0.87	0.92	0.91	0.91
SRR1104850	BisPin	UANF	0.52	0.77	0.77	0.74
SRR1534392	BisPin	UANF	0.59	0.82	0.73	0.72
SRR1534392	Tabsat	UAN	0.68	0.88	0.84	0.80
SRR2172246	BFAST	UANF	0.34	0.53	0.51	0.49
SRR2172246	Bowite2	UA	0.84	0.92	0.90	0.90
SRR5144899	Bismark	UAN	0.65	0.81	0.80	0.79
SRR5144899	BisPin	UANF	0.72	0.85	0.82	0.81

An example of precision, recall, and F1-score by class is shown in Table 3. The 'Read amount' column gives the number of reads in the class. Throughout this project, precision was generally better than recall, and Ambig was the class that was generally the hardest to predict. This may be because the ambiguously mapped class may have sequence complexity intermediate between uniquely mapped and unmapped reads [20] making the difference more difficult to distinguish. Ambiguously mapped reads may be a result of repetition in the genome [4,25] that can't be detected from examining the read alone.

Table 3. Precision, recall, and F1-Score by class for SRR5144899 Bismark.

Class	Precision	Recall	F1-Score	Read amount
Unique	0.851	0.974	0.909	393343
Ambig	0.657	0.133	0.221	36771
Unmap	0.775	0.473	0.587	69094

4.2 Feature Importance

Random forest feature importance was used to rank the features since the random forest models had the best predictive performance. This gives a ranking of features from most important to least important according to the model. This ranking was computed for each of the eleven data sets, and the distribution of ranks for each feature was computed. Figure 1 gives a notched box plot of these distributions for all of the features that used the entire read. Qual features are quality features. LZMA and bz2 are compression features, and all other features are related to sequence complexity.

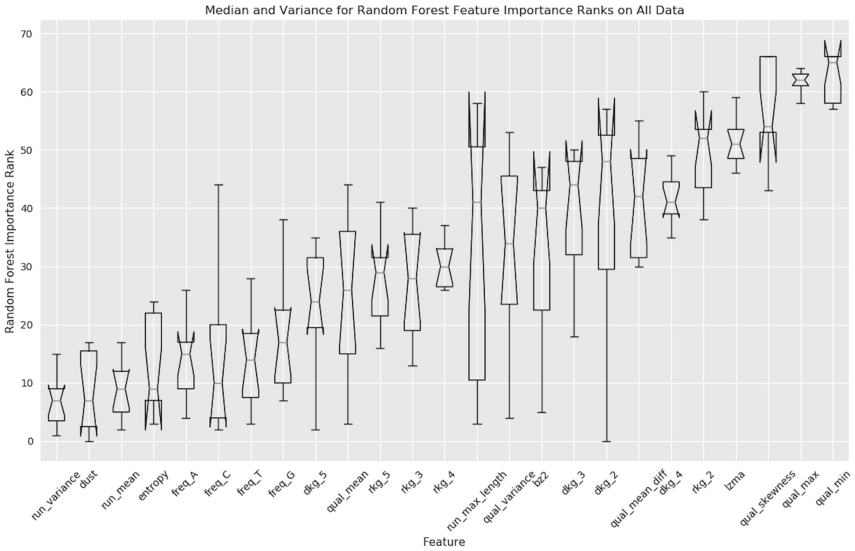

Fig. 1. Feature importances for all of the data. For each data set and each read mapper, random forest feature rank importances were calculated, and the distribution of rank for each feature was used to make the box plot. $D_k(a)$ is referred to as dkg, and $R_k(a)$ is referred to as rkg.

Run length variance and run length mean were among the most important and performed a bit better than entropy and the DUST score in some cases. This is interesting since several programs use the DUST score, such as BLAST [1,15], and entropy [14,21]. Run length metrics could be as good or better if they replaced the DUST score and entropy. Character frequency features were of good importance but not as important as the DUST score and entropy.

$D_k(a)$ and $R_k(a)$ performed more poorly; however, $D_2(a)$ was very important for the data ERR2562409 as it was ranked the most important with an average importance confidence 0.251, which was larger by 0.174 on average than the next best feature, the largest difference of its kind. Perhaps $D_k(a)$ is more useful for some data sets.

Compressibility measures were the worst average performing sequence complexity metrics. LZMA was the worst on average with a mean rank of 51.45. However, the Bzip2 feature from the first third of the sequence had the highest rank on the SRR1534392 data with BisPin, and LZMA in the second third of the sequence had the highest rank for the SRR1534392 data with Tabsat.

Quality metrics were generally not as important as sequence complexity metrics. The quality mean was the most important of these, and quality skewness, maximum, and minimum had the lowest importance of all features.

Since four of the six data sets were for bisulfite-sequencing reads, there could be a bias favoring bisulfite read mapping. Thus, the same feature rank analysis was performed with only the regular untreated data. The feature rank notched

box plots for this data can be found in Fig. 2. The order of features is very similar, but the DUST score does a little better, outperforming the run length metrics. The quality mean is a bit lower in the rankings.

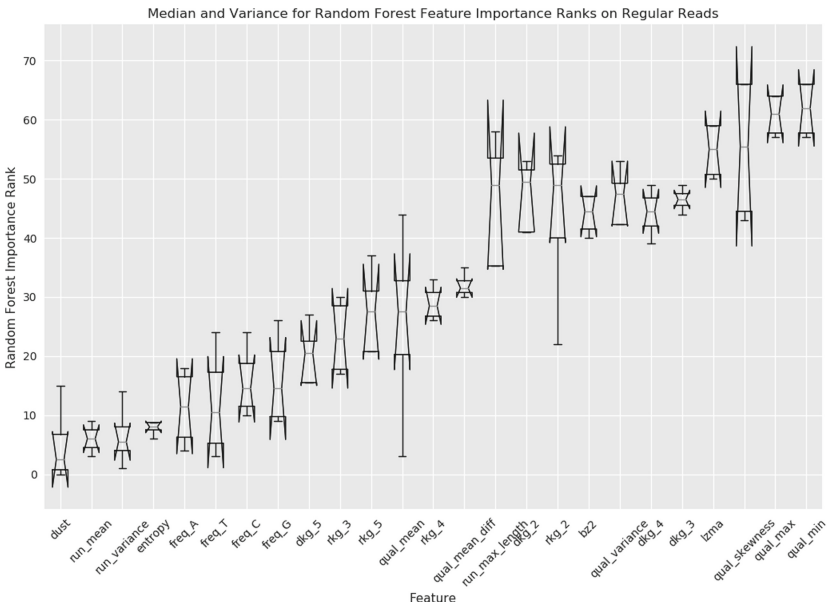

Fig. 2. Feature importances for the regular untreated data. $D_k(a)$ is referred to as dkg, and $R_k(a)$ is referred to as rkg.

In Illumina data sets, features from the last third of the read generally had a higher importance than features in the first or second thirds of the read sequence. Features from the second third were generally more important than features from the first third. This may be because there is often lower quality in the last third of a read since Illumina sequencing technology can make more errors in later cycles [2]. In Ion Torrent data, features from each third were generally more evenly distributed in the top 15 most important features.

4.3 Feature Ranking Similarity Across Different Data

There is weak evidence that the feature importance ranking depends more on the read mapper than the data set. This conclusion was drawn by looking at Kendall's tau coefficient for feature rankings across different data. Kendall's tau coefficient is used to measure how similar two ordered sequences are [7]. It ranges from 1.0 to −1.0. A 1.0 means the sequences are identical, and a −1.0 means that the sequences are the reverse of each other.

Kendall's tau coefficient and p-value were computed using scipy. The feature importance ranking for both read mappers for the same SRA number was used to

calculate Kendall's tau. Only ERR2562409 and ERR699568 had p-values below 0.1. All tau's were positive. The highest was for ERR699568 at 0.308, and the lowest was for SRR5144899 at 0.0276. Both data sets come from bisulfite-treated Illumina reads.

The feature importance ranking for all data mapped with BisPin was compared with SRR1104850 since it was mapped only with BisPin. In all cases, tau was larger than in the previous analysis. This suggests that read mapper feature rankings correlate better than feature rankings based on the same data set but mapped by different programs. This suggests that there is some program-specific qualities of feature performance, and data set specific qualities are less important.

4.4 Machine Learning Filter Proof-of-Concept

The random forest machine learning model was used as a read filter to test the idea that these features could lead to more effective read trimmers, masking algorithms, and so on. First, the average alignment score and the average edit distance were calculated on additional 300k–500k reads after alignment. The alignment score and edit distance are reported by the alignment program. Then, reads that were marked as unmapped or filtered by the RF model were excluded, and the averages were calculated. Table 4 summarizes the results. A positive number represents an improvement while a negative number represents a loss. The 200bp data set SRR1104850 had slightly worse alignments on average, but the other data sets showed a bigger improvement. This validates that these methods can be used as a low complexity filter to improve alignments.

Table 4. Differences in alignment score and edit distance for filtered reads.

Data	Mapper	Alignment diff	Edit diff
SRR2172246	BFAST	626.47	5.06
SRR5144899	BisPin	2283.69	6.86
SRR1104850	BisPin	−110.59	−2.03

5 Conclusions

My study showed that sequence complexity measures are important in predicting the read mapping quality of short DNA reads. Read quality metrics were less important. Run length mean and variance, the DUST score, and entropy were the best performing sequence complexity measures. Bioinformatics programs may consider using run length statistics because they were among the best features.

Without knowledge of the genome, and only knowledge of the DNA read, machine learning models, especially random forests, were able to predict alignment quality with surprisingly good accuracy approaching F1-scores of 0.95. The

features that work well on regular untreated reads tended to work well on bisulfite reads. This suggests that sequence complexity measures that work well in one application will probably work well in other applications.

Future work could include training a regressor to predict the alignment score rather than alignment categories; however not all programs (such as Bismark) report such a score. A model with very few features that predicts the alignment score could make a fast read filter. The effect of read trimming can be explored.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
2. Buermans, H., Den Dunnen, J.: Next generation sequencing technology: advances and applications. Biochimica et Biophysica Acta (BBA)-Mol. Basis Dis. **1842**(10), 1932–1941 (2014)
3. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. **20**(1), 37–46 (1960)
4. Deininger, P.: Alu elements: know the sines. Genome Biol. **12**(12), 236 (2011)
5. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. SSS, vol. 12. Springer, New York (2001). https://doi.org/10.1007/978-0-387-21606-5
6. Homer, N., Merriman, B., Nelson, S.F.: BFAST: an alignment tool for large scale genome resequencing. PLOS One **4**(11), e7767 (2009)
7. Kendall, M.G.: A new measure of rank correlation. Biometrika **30**(1/2), 81–93 (1938)
8. Krueger, F., Andrews, S.R.: Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. Bioinformatics **27**(11), 1571–1572 (2011)
9. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. Nat. Methods **9**(4), 357 (2012)
10. Leinonen, R., Sugawara, H., Shumway, M., International Nucleotide Sequence Database Collaboration: The sequence read archive. Nucleic Acids Res. **39**(1), D19–D21 (2010)
11. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics **25**(14), 1754–1760 (2009)
12. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications. TCS, Springer, New York (2008). https://doi.org/10.1007/978-0-387-49820-1
13. Libbrecht, M.W., Noble, W.S.: Machine learning applications in genetics and genomics. Nat. Rev. Genetics **16**(6), 321 (2015)
14. Lim, J.Q., et al.: BatMeth: improved mapper for bisulfite sequencing reads on DNA methylation. Genome Biol. **13**(10), R82 (2012)
15. Morgulis, A., Gertz, E.M., Schäffer, A.A., Agarwala, R.: A fast and symmetric DUST implementation to mask low-complexity DNA sequences. J. Comput. Biol. **13**(5), 1028–1040 (2006)
16. Pabinger, S., et al.: Analysis and visualization tool for targeted amplicon bisulfite sequencing on Ion Torrent sequencers. PloS One **11**(7), e0160227 (2016)
17. Pedersen, B.S., Eyring, K., De, S., Yang, I.V., Schwartz, D.A.: Fast and accurate alignment of long bisulfite-seq reads. arXiv preprint arXiv:1401.1129 (2014)
18. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)

19. Phan, V., Gao, S., Tran, Q., Vo, N.S.: How genome complexity can explain the difficulty of aligning reads to genomes. BMC Bioinform. **16**(17), S3 (2015)
20. Porter, J., Sun, M.a., Xie, H., Zhang, L.: Investigating bisulfite short-read mapping failure with hairpin bisulfite sequencing data. BMC Genomics **16**(11), S2 (2015)
21. Porter, J., Zhang, L.: InfoTrim: A DNA read quality trimmer using entropy. In: 2017 IEEE 7th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS), pp. 1–2. IEEE (2017)
22. Porter, J., Zhang, L.: BisPin and BFAST-Gap: Mapping bisulfite-treated reads, p. 26. bioRxiv (2018). https://doi.org/10.1101/284596, https://www.biorxiv.org/content/early/2018/06/16/284596
23. Quail, M.A., et al.: A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics **13**(1), 1–13 (2012)
24. Rougemont, J., Amzallag, A., Iseli, C., Farinelli, L., Xenarios, I., Naef, F.: Probabilistic base calling of Solexa sequencing data. BMC Bioinform. **9**(1), 1–12 (2008)
25. Schmid, C.W., Deininger, P.L.: Sequence organization of the human genome. Cell **6**(3), 345–358 (1975)
26. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press (1949)
27. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Advances in Neural Information Processing Systems, pp. 2951–2959 (2012)
28. Starostina, E., Tamazian, G., Dobrynin, P., O'Brien, S., Komissarov, A.: Cookiecutter: a tool for KMER-based read filtering and extraction, p. 024679. bioRxiv (2015)
29. Tran, H., Porter, J., Sun, M.a., Xie, H., Zhang, L.: Objective and comprehensive evaluation of bisulfite short read mapping tools. In: Advances in Bioinformatics, vol. 2014, p. 11 (2014)
30. Wang, X.V., Blades, N., Ding, J., Sultana, R., Parmigiani, G.: Estimation of sequencing error rates in short reads. BMC Bioinform. **13**(1), 185 (2012)
31. Zielezinski, A., Vinga, S., Almeida, J., Karlowski, W.M.: Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol. **18**(1), 186 (2017)
32. Zou, L.S., et al.: BoostMe accurately predicts DNA methylation values in whole-genome bisulfite sequencing of multiple human tissues, p. 207506. bioRxiv (2018)