Benchmarking of computational error-correction methods for next-generation sequencing via unique molecular identifiers

Keith Mitchell[1$,] Igor Mandric[1,3$], Lana Martin[2], Ekaterina Gerasimov[3], Russell Littman[1], Brian Hill[1], Nicholas C. Wu[4], Harry Yang[1], Kevin Hsieh[1], Linus Chen[1], Taylor Shabani[1], German Shabanets[1], Douglas Yao[8], Ren Sun[7], Jan Schroeder[8], Eleazar Eskin[1], Alex Zelikovsky[3,6], Pavel Skums[3], Mihai Pop[5], Serghei Mangul[1,2#]

[1] Department of Computer Science, University of California Los Angeles, 580 Portola Plaza, Los Angeles, CA 90095, USA

[2] Institute for Quantitative and Computational Biosciences, University of California Los Angeles, 611 Charles E. Young Drive East, Los Angeles, CA 90095, USA

[3]Department of Computer Science, Georgia State University 1 Park Place, Atlanta, GA, 30303

[4] Department of Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, CA 92037, USA

[5] Department of Computer Science and Center for Bioinformatics and Computational Biology, University of Maryland, College Park, MD 20742, USA

[6] The Laboratory of Bioinformatics, I.M. Sechenov First Moscow State Medical University, Moscow, 119991, Russia

[7] Department of Molecular and Medical Pharmacology, University of California Los Angeles, 650 Charles E. Young Drive South, Los Angeles, CA 90095, USA

[8] Department of Molecular, Cell, and Developmental Biology, University of California Los Angeles, 650 Charles E. Young Drive South, Los Angeles, CA 90095, USA

$ - These authors contributed equally to the paper

- Corresponding author

Abstract

Recent advancements in next-generation sequencing have rapidly improved our ability to study genomic material at an unprecedented scale. Despite substantial improvements in sequencing technologies, the errors present in the data still risk confounding downstream analysis and limiting the applicability of sequencing technologies in clinical tools. Computational error-correction promises to eliminate sequencing errors, but the relative performance of error correction algorithms remains unknown. In this paper, we evaluate the ability of error-correction algorithms to fix errors across different types of datasets that contain various level of heterogeneity. We highlight the advantages and limitations of computational error correction techniques across different domains of biology, including immunogenomics and virology. To demonstrate the efficacy of our technique, we applied the UMI-based high-fidelity sequencing protocol to eliminate sequencing errors from both simulated data and the raw

reads. We then performed a realistic evaluation of error correction methods. We find that method performance varies substantially across different types of datasets, with no single method performing best on all types of examined data. Finally, we also identified the techniques that offer a good balance between precision and sensitivity.

Introduction

Rapid advancements in next-generation sequencing have improved our ability to study genomic material of a biological sample at an unprecedented scale and promise to revolutionize our understanding of living systems[1]. Sequencing technologies are now the technique of choice for many research applications in human genetics, immunology, and virology[1,2]. Modern sequencing technologies dissect the input genomic DNA (or reverse transcribed RNA) into millions of nucleotide sequences, known as reads. Despite constant improvements in sequencing technologies, the data produced by these techniques remain affected by the introduction of random and systematic errors. The majority of errors are introduced at the sequencing step, and additional errors may be introduced at sample preparation or library preparation stages. Such errors present in the data limit the applicability of sequencing technologies in clinical settings. Further, the error rates vary across platforms[3], with the most popular Illumina-based protocols producing approximately one error in one thousand nucleotides[4]. Our paper conducts a comprehensive benchmarking of currently available error correction methods in order to better understand the nature of these errors and

effective correction methods. We have explored the effect of various sequencing setting and parameters of the error correction methods on the accuracy of error correction. We have also investigated the advantages and limitations of computational error correction techniques across different domains of biology, including immunogenomics and virology.

Eliminating errors in sequencing data is challenging. Currently available molecular-based techniques to correct errors in the sequencing data (e.g. ECC-Seq[5]) usually carry an increased cost that limits the scalability of such approaches across a large number of samples. Computational error-correction techniques promise to eliminate the sequencing errors and improve the results of downstream analyses (**Figure 1a**)[6]. Many computational error correction methods have been developed to meet the demand for accurate sequencing data in the biomedical community [7–9]. However, the lack of a systematic comparison of error correction methods impedes their integration into standardized next-generation sequencing data analysis pipelines. Previous benchmarking studies[10,11] lacked a comprehensive experimental gold standard, and instead relied on simulated reads and real reads which were uniquely aligned to the reference genome. Differences between the reads and the reference genome were consider as errors. Since this approach is unable to differentiate between the real errors and single nucleotide variants it oversimplifies the error correction problem and may bias benchmarking results[12]. In addition, error correction algorithms have undergone significant development since the prior benchmarking studies, and the performance of the newest methods has not yet been evaluated. Other studies[13] provide a detailed description of available error correction tools yet lack the benchmarking results. The efficiency of today's error

correction algorithms,when applied to the extremely heterogeneous populations composed of highly similar yet distinct genomic variants, is presently unknown. The human immune repertoire, a collection of diverse B and T cell receptor clonotypes, is an excellent example of a heterogeneous population with need for reliable error correction. The increased heterogeneity of such datasets and the presence of low-frequency variants further challenges the ability of error-correction methods to fix sequencing errors in the data.

In this paper, we evaluate the ability of error-correction algorithms to fix errors across different types of datasets with various levels of heterogeneity. In doing so, we produce a gold standard that provides an accurate baseline for performing a realistic evaluation of error correction methods. We highlight the advantages and limitations of computational error correction techniques across different domains of biology, including immunogenomics and virology. For example, we challenged the error correction methods with data derived from diverse populations of T and B cell receptor clonotypes and intra-host viral populations. To define a gold standard we applied the UMI-based high-fidelity sequencing protocol (also known as safe-SeqS) [14,15] to eliminate sequencing errors from raw reads.

Results

Gold standard datasets

We used both simulated and experimental gold standard datasets derived from human genomic DNA, human T and B cell receptor repertoires and intra-host viral populations. The

datasets we used correspond to different levels of heterogeneity; the difficulty of error correction increases as the dataset becomes more heterogeneous. The least heterogeneous datasets were derived from human genomic DNA (**Table 1**). We used TCR-Seq reads derived from patients diagnosed with Human Immunodeficiency Virus (HIV)[16]. In addition, we used reads derived from a population of HIV viral strains which were sampled from an infected individual. This type of population represents a complex community of closely related viral mutant variants (known as quasispecies)[17].

To generate error-free reads, we used a UMI-based high-fidelity sequencing protocol (also known as safe-SeqS) [14,15] capable of eliminating sequencing errors from the data. A high-fidelity sequencing protocol attaches the Unique Molecular Identifier (UMI) to the fragment prior to amplification of DNA fragments. After sequencing, the reads which originated from the same biological segment are grouped into clusters based on their UMI tags. Next, we applied an error-correction procedure inside each cluster. In case at least one nucleotide inside the UMI cluster lack the the support of 80% reads, we were not able to generate consensus error-free read and all the reads from the UMI cluster are disregarded (**Figure 1c**).

Availability of both error-free reads and the original raw reads carrying errors provide an accurate,robust baseline for performing a realistic evaluation of error correction methods. In addition to the experimental data, we used simulated data with various read length and throughput to estimate the effect of such sequencing parameters on the accuracy of error correction. A detailed description of the dataset used for this portion of our study is provided in

Supplementary Materials and is available at https://github.com/smangul1/error.correction.benchmarking .

Table 1. Overview of the gold standard datasets.

	D1	D2	D3	D4
Technology	Whole genome sequencing	T cell receptor sequencing	B and T cell receptor sequencing	Viral sequencing
Heterogeneity	Low	High	High	High
Technique	Simulated	UMI-based error-free reads	Simulated	UMI-based error-free reads
Number of samples	100	10	200	1

Choice of error correction methods

We chose the most commonly-used error correction tools to assess the ability of current methods to correct sequencing errors. The following algorithms were included in our

benchmarking: Coral[18], Bless[8], Fiona[8,19], Pollux[9], BFC[20], Lighter[21], Musket[7], Racer[22], RECKONER[21,23], and SGA[24]. We excluded HiTEC and replaced it with RACER, as was recommended by the developers of HiTEC. We also excluded tools solely designed for non-Illumina based technologies[25] and tools which are no longer supported. We summarize the details of each tool, including the underlying algorithms and the data structure (**Table 2)**. To assess the simplicity of the installation process for each of the methods, we describe the software dependencies in **Table 3**. Commands required to install and run each of the tools are available in Supplementary Materials and are freely available at https://github.com/smangul1/error.correction.benchmarking.

Table 2: Summary of error correction methods parameters and publication details. Error correction methods are sorted by the year of publication (indicated in column 'Published Year'). We documented underlying algorithm (indicated in column "Underlying algorithm"), version of the error correction tool used (indicated in column "Version"), name of the software tool (indicated in column 'Software tool').

Software tool	Version	Underlying algorithm	Published year	Programming language	Default k-mer size	Read trimming
Coral	1.4.1	MSA*	2011	C	N/A	YES

SGA	0.10.15	FM-index search	2012	C++	31	NO
Musket	1.1	k-mer spectrum	2012	C++	N/A	NO
Racer	1.0.1	k-mer spectrum	2013	C++	N/A	NO
BLESS	1.02	k -mer spectrum	2014	C++	N/A	YES
Lighter	1.1.1	k -mer spectrum	2014	C++	N/A	NO
Fiona	0.2.8	k -mer spectrum	2014	C++	N/A	YES
BFC	1	k -mer spectrum	2015	C	N/A	NO
Pollux	1.0.2	k -mer spectrum	2015	C	31	YES
RECKONER	0.2.1	k-mer spectrum	2017	C++	N/A	NO

Table 3. Summary of technical characteristics of error correction methods.

Software tool	Data structure	Types of reads accepted	Organism	Journal	In the publication compared to	Tools webpage
BLESS	Bloom filter and hash table	SE/PE	Human, *E. Coli*, S. aureus	Bioinformatics	SGA, QuorUM, Lighter, BFC, DecGPU, ECHO, HiTEC, Musket, Quake, Reptile	https://sourceforge.net/p/bless-ec/wiki/Home/
Fiona	partial suffix array	SE	human, drosophila, *E. Coli, C. elegans*	Bioinformatics	Allpaths-LG,Coral,H-Shrec,ECHO,HiTEC,Quake	https://github.com/seqan/seqan/tree/master/apps/fiona
Pollux	Hash table	SE/PE	Human, *E. Coli, S. aureus,* mixed genome data	BMC Bioinformatics	Quake, SGA, BLESS, Musket, RACER	https://github.com/emarinier/pollux
BFC	Bloom filter and hash table	SE/PE	Human, *C. Elegans*	Bioinformatics	BLESS, Bloocoo, fermi2, Lighter, Musket, and SGA	https://github.com/lh3/bfc
Lighter	Bloom filter	SE/PE	Human, *E. Coli, C. elegan*	Genome Biology	Quake, Musket, Bless, Soapec	https://github.com/mourisl/Lighter
Musket	Bloom filter and hash table	SE/PE	Human, *E. Coli, C. elegans*	Bioinformatics	SGA, Quake	http://musket.sourceforge.net/homepage.htm
Racer	Hash table	SE/PE	Human, *E. Coli, C. elegans* , Drosophila, other bacteria	Bioinformatics	Coral, HITEC, Quake, Reptile, SHREC	http://www.csd.uwo.ca/~ilie/RACER/

Coral	Hash table	SE/PE	Human, *E. coli, S. aureus*	Bioinformatics	COMPASS 3.0, HHalign 1.5.1.1 and PSI-BLAST	https://www.cs.helsinki.fi/u/lmsalmel/coral/
RECKONER	Hash table	SE	Human, *S. cerevisiae, C. elegans, M. acuminata*	Bioinformatics	Ace, BFC, BLESS, Blue, Karect, Lighter, Musket, Pollux, RACER, Trowel	https://github.com/refresh-bio/RECKONER
SGA	FM-index	SE/PE	Human, *C. elegans, E. coli,*	Genome Research	Velvet, ABySS, SOAPdenovo, Quake, HiTEC	https://github.com/jts/sga

Evaluation of the accuracy and performance of error correction methods

We evaluated the accuracy and performance of error correction methods using an extensive set of evaluation metrics. We define true positives (TP) as errors correctly fixed by the error correction tool, false positives (FP) as correct bases that were erroneously changed by the correction tool, false negatives (FN) as erroneous bases not fixed or incorrectly fixed, and true negatives (TN) as correct bases which remain unaffected by the error correction tool (**Figure 1b**) (**Figure S1**). Following the previous benchmarking? study[10], we used the **gain metric** defined as the percentage of sequencing errors errors correctly fixed by the error correction tool (**Table 3**). In cases where the number of correctly fixed bases is smaller when compared to incorrectly fixed bases, the gain becomes negative. The overall gain is also negative in scenarios where the majority of bases are incorrectly fixed. Gains of greater than 0% represent an overall positive effect of the error correction algorithm, whereas a value of less than 0% shows that the

tool performed more incorrect actions then correct actions. A gain of 100% means the tool made all possible TP corrections without any FP alterations; in other words all the correction performed by the error correction tools were correct and all the sequencing errors in the data were fixed.

We define precision as the proportion of proper corrections among the total number of performed corrections. Sensitivity evaluates the proportion of fixed errors among all existing errors in the data; in other words, sensitivity indicates which algorithms are correcting the highest majority of induced errors [26]. Finally, some methods remove the bases in the corrected reads, usually at the end of the reads. Removing the bases may correspond with an attempt to correct deletion (TP trimming) or may just remove a correct base (FP trimming) (**Figure S2**). We have also reported CPU time and the maximum amount of RAM used by each of the tools.

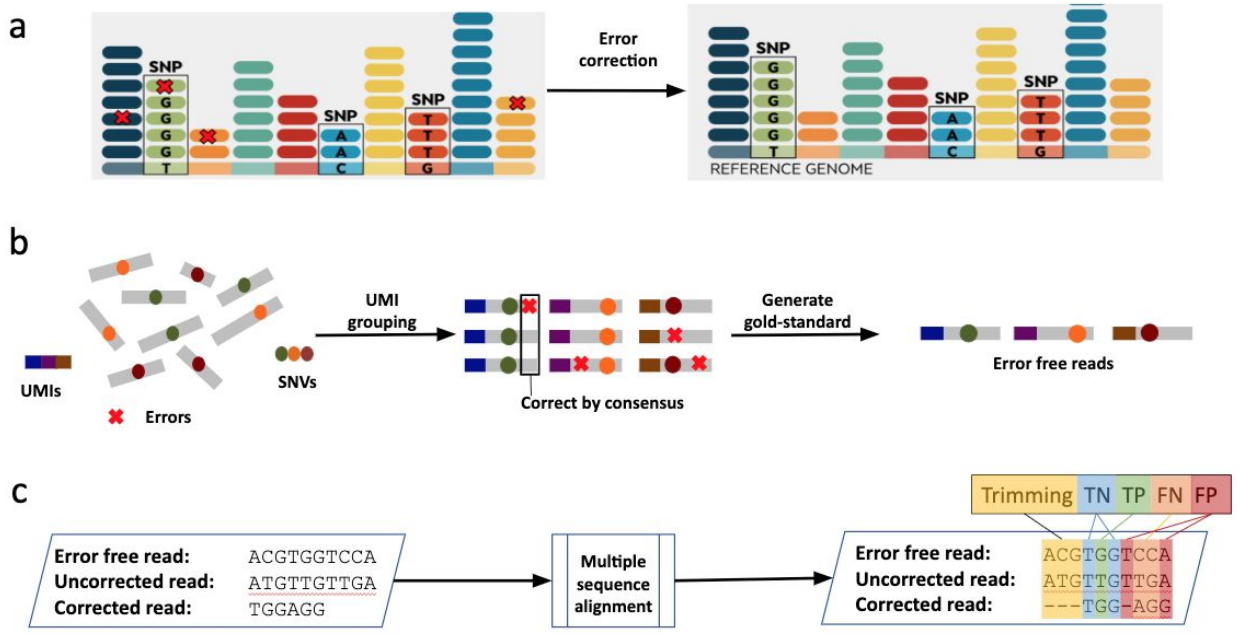


Figure 1. Study design to benchmark computational error correction methods. **(a)** Schematic representation of the general goal of error correction algorithms. Error correction aims to fix sequencing errors while maintaining the heterogeneity of the data. **(b)** Error-free reads for gold standard were generated using Unique Molecular Identifier (UMI)-based clustering. Reads were grouped based on matching UMIs and corrected by consensus, where 80% majority was required to correct sequencing errors without affecting naturally occurring SNVs (Single Nucleotide Variations). (**c)** The framework to evaluate the accuracy of error correction methods. Multiple sequence alignment between the error-free, uncorrected (original), and corrected reads is performed to classify bases in the corrected read. Bases fall into the category of Trimming, True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP).

Table 3. Evaluation of the accuracy of error correction methods. (a) The trim efficiency is the proportion of trimmed bases from the tool that were considered TP trimming. (**b)** The trim percent is the proportion of nucleotides trimmed out of all nucleotides analyzed. (**c)** Precision evaluates the proportion of proper corrections among the total number of performed corrections (**d)** Gain determines the relative number of TP corrections as opposed to FP assessments by the error correction tools. (**e)** Sensitivity evaluates the proportion of fixed errors among all existing errors in the data.

	Metric Name	Metric Formula
a.	Trim Efficiency	$\frac{TP(TRIM)}{TP(TRIM)+FP(TRIM)}$
b.	Trim Percent	$\frac{TP(TRIM)+FP(TRIM)}{TotalBases}$
c.	Precision	$\frac{TP}{(TP+FP+FP(INDEL))}$
d.	Gain	$\frac{TP-(FP+FP(INDEL))}{(TP+FN)}$
e.	Sensitivity	$\frac{TP}{(TP+FN)}$

Correcting errors in the whole genome sequencing data

We evaluated the efficacy of currently available error correction methods in fixing errors introduced to whole genome sequencing (WGS) reads using various coverage settings. First, we explored the effect of kmer size on the accuracy of error correction methods. An increase in kmer size typically offers increased accuracy of error correction. In some cases, increased kmer

size has no effect on the accuracy of error correction (**Figure S3**). Thus, we have used the largest kmer size of 32bp for all the methods except BFC(**Table S1**). The BFC method for WGS data with 32x coverage performs the best with kmer size of 23bp (**Figure S3f**). For other coverages, BFC performs the best with kmer size of 32bp, which was chosen in those cases. Shorter kmer size results in the error correction to perform fewer corrections. Pollux and Musket make the largest number of corrections across all coverage settings (**Figure S4**).

In general, higher coverage allows error correction methods to make more corrections and fix more errors in the data (**Figure S5a**). For all the methods except Pollux and Musket, higher coverage also results in decreasing the number of false corrections (**Figure S5b**). While an overall increase in the number of proper corrections with coverage increase across all the methods, the gain has different trends across the variety of tools. For approximately half of the tools (e.g., BFC, Coral, and SGA) gain constantly increases with coverage increase. For other tools, the gain has multiple trends (**Figure 2a**). For the vast majority of the error correction tools, the gain becomes positive only for coverage 32x. The only methods able to achieve positive gain for 16x coverage is SGA and Coral. For coverage of 8x only, Coral is able to obtain a positive gain. None of the surveyed methods are able to maintain positive gain for coverage lower than 8x (**Figure 2a**). Coverage level also has a strong impact on both precision and sensitivity (**Figure 2b-c**). For example, Bless and Musket are only able to accurately correct the errors in highly covered datasets (**Figure 1a**). In general, none of the methods are able to correct more than 60% of the data for datasets with coverage less than 16x (**Figure S6**). Coverage of 32x allows several methods to correct more than 95% of the errors with high

precision (**Figure 1f**). The error correction tools typically trim a minor portion of the reads. We have compared the trimming rates and trends across the error correction methods (**Figure S7**). Overall, the majority of trimmed bases were correct bases (**Figure S8**).

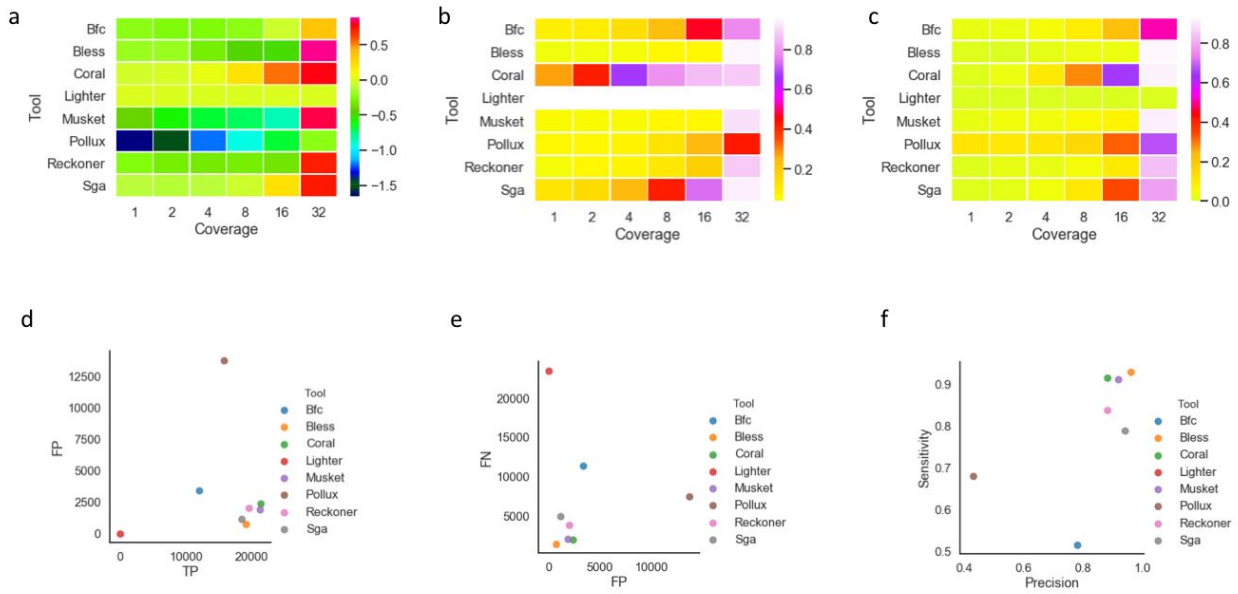


Figure 2. Correcting errors in whole genome sequencing data. **(a)** Heatmap depicting the gain across various coverage settings. Each row corresponds to an error correction tool each column corresponds to a dataset with a given coverage. **(b)** Heatmap depicting the precision across various coverage settings. Each row corresponds to an error correction tool each column corresponds to a dataset with a given coverage. **(c)** Heatmap depicting the sensitivity across various coverage settings. Each row corresponds to an error correction tool each column corresponds to a dataset with a given coverage. **(d)** Scatter plot depicting the number of TP corrections (x-axis) and FP corrections (y-axis) for datasets with 32x coverage **(e)** Scatter plot depicting the number of FP corrections (x-axis) and FN corrections (y-axis) for datasets with 32x coverage **(f)** Scatter plot depicting the sensitivity (x-axis) and precision (y-axis) for datasets with 32x coverage.

Correcting the errors in the immune repertoire sequencing data

We have evaluated the ability of error correction methods to fix errors in B cell receptor sequencing data (BCR-Seq) and T cell receptor sequencing data (TCR-Seq) reads with various coverage and read length settings. Similarly to the WGS data, we have explored the effect of kmer size on the accuracy of error correction methods for sequencing data. In contrast to WGS data, accuracy of error correction in BCR-Seq data does not increase with increasing the kmer size. Instead, each error correction method prefers various kmer sizes (**Figure S9**). For example, for coverage 16x Polux archives best gain with kmer of size 18bp and Mucket with kmer size 24bp. We identified the kmer size with the best performance for each tool and for each coverage (**Table S2**). Those kmer sizes were used when evaluating the performance of the error correction methods. In contrast with the WGS data, we find no observable general trend regarding the number of corrections affected by the kmer size (**Figure S10**).

Remarkably, the gain was positive across all the coverages and error correction tools (**Figure 3a**). All surveyed tools maintain high precision rates (above 60%) (**Figure 3b**). Sensitivity has larger variation across the tools and coverages. For several error correction methods, sensitivity drops when the coverage increases (**Figure 3c**). In particular, Coral and Musket achieve a good balance between sensitivity and precision (**Figure 3f**) (**Figure S11**). Other

methods are less balanced; for example, due to the decreased number of TP corrections, SGA has the lowest sensitivity across all coverages (**Figure S11**).

Similarly to WGS data, the error correction tools do not trim or only trim a minor portion of the reads. Percentage of trimmed bases is less than 30% (**Figure S17**). Typically, the majority of trimmed bases were correct bases **(Figure S17)**. We also noticed that read length has a positive effect on the ability of methods to correct errors in the BCR-Seq data. Typically, the increase in reading length improves the results of error correction. In several cases, shortened read length provides superior accuracy. For example, Pollux achieves gain of 40% for the read lengths of 50bp and 100bp, and negative gain for the read length of 75bp **(Figure S11)**.

We also compared the ability of error correction methods to fix the errors in reads derived from the T cell receptor (TCR) repertoire. Similarly, with BCR-Seq data we simulated the reads of various lengths from transcripts derived from T cell receptor alpha chain (TCRA). In contrast to BCR-Seq data, an increase of kmer length produces a positive effect on error correction accuracy across the error correction tools (**Figure S12**). Thus, we used the largest kmer size of 24bp for all surveyed methods (**Table S3**). We also investigated the effect of kmer size using real TCR-Seq data derived from 10 HIV positive patients. Error-free reads for a gold standard were generated by consensus using UMI-based clustering (**Methods Section**). For all the samples, kmer size has a positive effect on the ability of the methods to correct errors (**Figure S13**). The highest accuracy is achieved using Lighter and BFC methods, followed by SGA (**Figure 4a**). Both Lighter and BFC achieve a good balance between precision and sensitivity,

with the precision exceeding 99.9% (**Figure 4d**). Similar to BCR-Seq data, SGA shows the smallest sensitivity among all the methods we tested. In general, methods with positive gain are able to correct less than 60% of the error. Methods with high sensitivity typically have overall poor performance due to the increased number of inaccurate corrections (**Figure 4b-c**). Similarly to other datasets, only a small number of reads are trimmed (0-0.001%) (**Figure S14**).

In contrast with the immune repertoire sequencing data, none of the error correction tools are suitable to correct reads derived from the heterogeneous viral populations**.** Despite the ability of the methods to correct the majority of the sequencing errors, the high number of incorrect adjustments resulted in a decreased precision across all the methods **(Figure S15).**

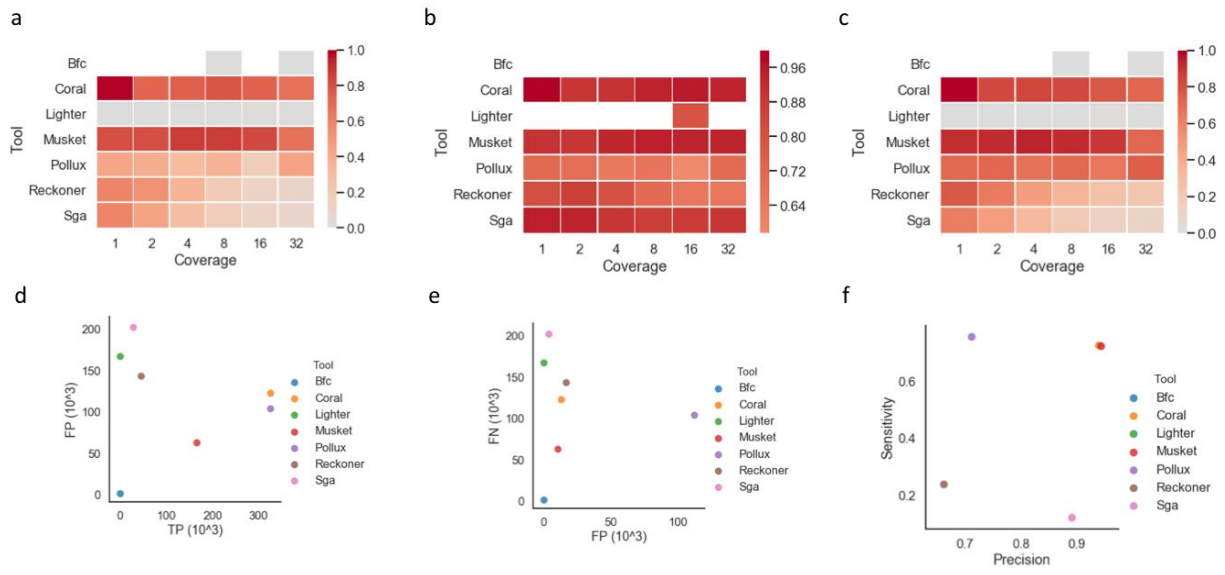


Figure 3. Correcting the errors in BCR-Seq data **(a)** Heatmap depicting the gain across various coverage settings. Each row corresponds to an error correction tool each column corresponds to a dataset with a given coverage. **(b)** Heatmap depicting the precision across various coverage settings. Each row corresponds to an error correction tool each column corresponds to a dataset with a given coverage. **(c)** Heatmap depicting the sensitivity across various coverage settings. Each row corresponds to a error correction tool each column corresponds to a dataset with a given coverage. **(d)** Scatter plot depicting the number of TP corrections (x-axis) and FP corrections (y-axis) **(e)** Scatter plot depicting the number of TP corrections (x-axis) and FP corrections (y-axis) **(f)** Scatter plot depicting the sensitivity (x-axis) and precision (y-axis).

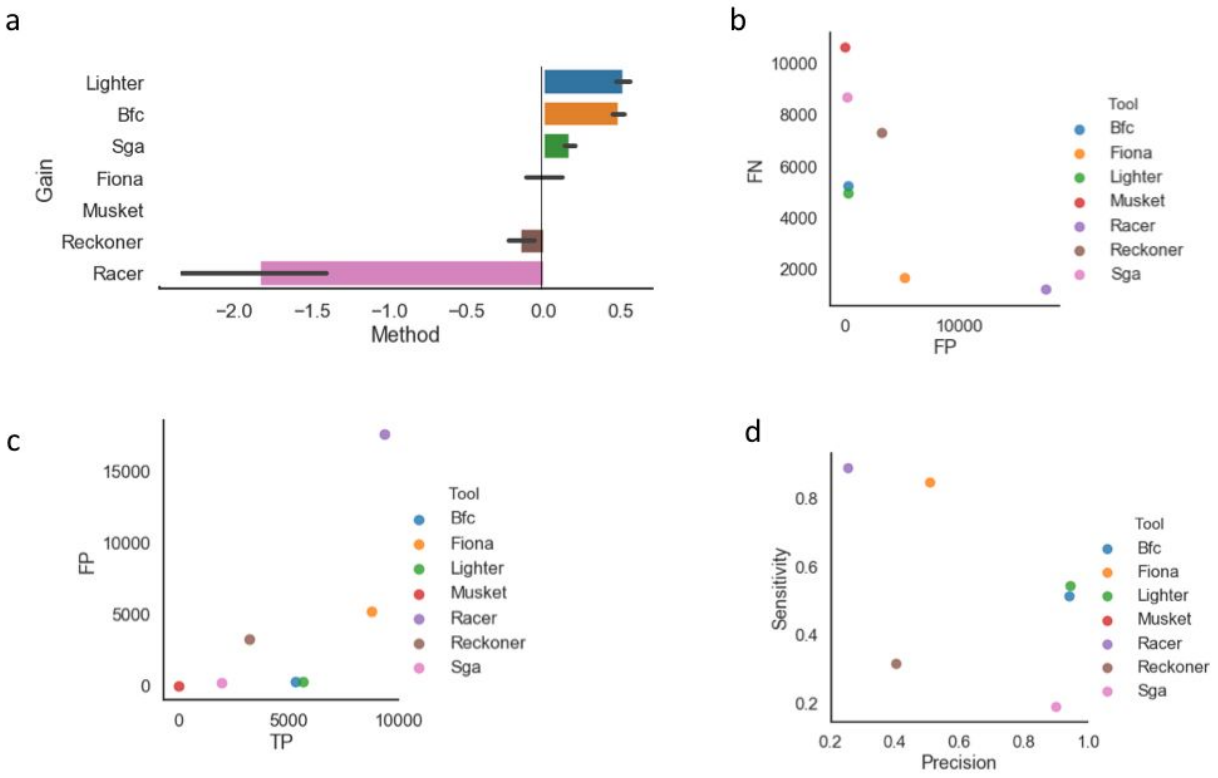


Figure 4. Correcting the errors in TCR-Seq data (a) Barplot depicting the gain across various error correction methods. Vertical bars depict the various across 10 TCR-Seq samples. **(b)** Scatter plot depicting the number of TP corrections (x-axis) and FP corrections (y-axis). Mean value across 10 samples is reported for each tool **(e)** Scatter plot depicting the number of TP corrections (x-axis) and FP corrections (y-axis). Mean value across 10 samples is reported for each tool **(f)** Scatter plot depicting the sensitivity (x-axis) and precision (y-axis). Mean value across 10 samples is reported for each tool.

Discussion

We evaluated the ability of error-correction algorithms to fix the errors across different types of datasets with various level of heterogeneity. In addition to the regular DNA sequencing, we evaluated the performance of error correction algorithms on highly heterogeneous data derived from human immune repertoires and intra-host viral populations. For whole genome sequencing error correction, methods deliver improved sequencing reads for datasets with coverage higher than 8x. The performance of error correction methods generally exhibits a large variability across different types of datasets, with no single method performing best on all types of data. We also identified methods with a good balance between precision and sensitivity. We determined that coverage larger than 8x allows the error correction tools to reduce the number of sequencing errors in the whole genome sequencing (WGS) data. Coverage of 32x allows several methods to correct more than 90% of the errors with high precision. For example, Bless was able to correct 93.0% of the errors with 96.0% precision.

In addition to the coverage we have also evaluated the effect of kmer size on the error correction accuracy. An increase of kmer size typically offers an increase in accuracy of error correction for both WGS and TCR-Seq data. Accuracy of error correction for BCR-Seq data does not increase with increasing the kmer size. Instead, each error correction method prefers various kmer sizes. We have also notice the differences in the ability of error correction tools to leverage the decrease coverage, with some tools able to produce accurate results only for high coverage datasets.

We find that method performance varies substantially across different types of datasets, with no single method performing best on all types of examined data. For example, Lighter has shown the best performance on the heterogeneous TCR-Seq data and poor performance on WGS data. We noticed that error correction methods are particularly useful in the field of immunogenomics, where multiple error correction methods provided significantly improved results compared to the baseline level even for extremely low coverages. This results suggest the benefits of integrating the error correction tools into the immune repertoires data analysis pipelines to increase the accuracy of T and B cell receptors repertoires profiling. In contrast, none of the methods were able to accurately correct errors in the data derived from the heterogeneous viral populations.

References

1. Schuster, S. C. Next-generation sequencing transforms today's biology. *Nature Methods* **5**, 16–18 (2008).
2. Scholz, M. B., Lo, C.-C. & Chain, P. S. G. Next generation sequencing and bioinformatic bottlenecks: the current state of metagenomic data analysis. *Curr. Opin. Biotechnol.* **23**, 9–15 (2012).
3. Robasky, K., Lewis, N. E. & Church, G. M. The role of replicates for error mitigation in next-generation sequencing. *Nat. Rev. Genet.* **15**, 56–62 (2014).
4. Ratan, A. *et al.* Comparison of Sequencing Platforms for Single Nucleotide Variant Calls in a Human Sample. *PLoS One* **8**, e55089 (2013).
5. Chen, Z. *et al.* Highly accurate fluorogenic DNA sequencing with information theory-based error correction. *Nat. Biotechnol.* **35**, 1170–1178 (2017).
6. Heydari, M., Miclotte, G., Demeester, P., Van de Peer, Y. & Fostier, J. Evaluation of the impact of Illumina error correction tools on de novo genome assembly. *BMC Bioinformatics* **18**, 374 (2017).
7. Liu, Y., Schröder, J. & Schmidt, B. Musket: a multistage k-mer spectrum-based error corrector for Illumina sequence data. *Bioinformatics* **29**, 308–315 (2013).
8. Heo, Y., Wu, X.-L., Chen, D., Ma, J. & Hwu, W.-M. BLESS: bloom filter-based error correction solution for high-throughput sequencing reads. *Bioinformatics* **30**, 1354–1362 (2014).

9. Marinier, E., Brown, D. G. & McConkey, B. J. Pollux: platform independent error correction of single and mixed genomes. *BMC Bioinformatics* **16**, 10 (2015).

10. Yang, X., Chockalingam, S. P. & Aluru, S. A survey of error-correction methods for next-generation sequencing. *Brief. Bioinform.* **14**, 56–66 (2013).

11. Molnar, M. & Ilie, L. Correcting Illumina data. *Brief. Bioinform.* **16**, 588–599 (2015).

12. Mangul, S. *et al.* Systematic benchmarking of omics computational tools. *Nat. Commun.* **10**, 1393 (2019).

13. Laehnemann, D., Borkhardt, A. & McHardy, A. C. Denoising DNA deep sequencing data—high-throughput sequencing errors and their correction. *Brief. Bioinform.* **17**, 154–179 (2015).

14. Zhang, T.-H., Wu, N. C. & Sun, R. A benchmark study on error-correction by read-pairing and tag-clustering in amplicon-based deep sequencing. *BMC Genomics* **17**, 108 (2016).

15. Kinde, I., Wu, J., Papadopoulos, N., Kinzler, K. W. & Vogelstein, B. Detection and quantification of rare mutations with massively parallel sequencing. *Proc. Natl. Acad. Sci. U. S. A.* **108**, 9530–9535 (2011).

16. Heather, J. M. *et al.* Dynamic Perturbations of the T-Cell Receptor Repertoire in Chronic HIV Infection and following Antiretroviral Therapy. *Front. Immunol.* **6**, 644 (2015).

17. Mangul, S. *et al.* Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics* **30**, i329–37 (2014).

18. Salmela, L. & Schröder, J. Correcting errors in short reads by multiple alignments. *Bioinformatics* **27**, 1455–1461 (2011).

19. Schulz, M. H. *et al.* Fiona: a parallel and automatic strategy for read error correction.

Bioinformatics **30**, i356–63 (2014).

20. Li, H. BFC: correcting Illumina sequencing errors. *Bioinformatics* **31**, 2885–2887 (2015).
21. Song, L., Florea, L. & Langmead, B. Lighter: fast and memory-efficient sequencing error correction without counting. *Genome Biol.* **15**, 509 (2014).
22. Ilie, L. & Molnar, M. RACER: Rapid and accurate correction of errors in reads. *Bioinformatics* **29**, 2490–2493 (2013).
23. Dlugosz, M. & Deorowicz, S. RECKONER: read error corrector based on KMC. *Bioinformatics* **33**, 1086–1089 (2017).
24. Simpson, J. T. & Durbin, R. Efficient de novo assembly of large genomes using compressed data structures. *Genome Res.* **22**, 549–556 (2012).
25. Wirawan, A., Harris, R. S., Liu, Y., Schmidt, B. & Schröder, J. HECTOR: a parallel multistage homopolymer spectrum based error corrector for 454 sequencing data. *BMC Bioinformatics* **15**, (2014).
26. Olson, D. L. & Delen, D. *Advanced Data Mining Techniques*. (Springer Science & Business Media, 2008).
27. lh. lh3/wgsim. *GitHub* Available at: https://github.com/lh3/wgsim. (Accessed: 9th May 2019)
28. Mangul, S. *et al.* Profiling immunoglobulin repertoires across multiple human tissues by RNA Sequencing. doi:10.1101/089235
29. Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**, 1792–1797 (2004).

Methods

Running error correction tools

Tools were installed using the directions provided with each of the respective tools (Table 2). Wrappers were then prepared in order to run each of the respective tools as well as create standardized log files. When running the tools, we choose the Illumina technology option and paired-end mode when possible. In case the paired-end option is not available (Table 1), we prepared single-end reads obtained from paired-end data by disregarding pairing information and treating each read from the pair as a single-end read. The computational pipeline to compare error correction methods is open source, free to use under the MIT license and available at https://github.com/smangul1/error.correction.benchmarking.

Generate error-free reads using UMI-based clustering

Error-free reads for gold standard were generated using UMIs(Unique Molecular Identifiers)-based clustering. Reads were grouped based on matching UMIs and corrected by consensus, where 80% majority was required to correct sequencing errors without affecting naturally occurring SNVs (Single Nucleotide Variations) (**Figure 1B**).

Simulated datasets generation.

We have generated simulation data mimicking the WGS data (D1). To generate the D1 dataset, we customized version of the tool WgSim[27] **(Figure S16).** Read coverage varied between 1 and 32. Briefly, this version along with generating the sequencing reads with errors also has the feature of reporting the error-free reads to the files provided as command line arguments. The WgSim fork is available at https://github.com/mandricigor/wgsim.

We have generated simulation data mimicking the BCR-Seq and TCR-Seq data (D3). To generate the BCR-Seq dataset, we have used Immunoglobulin heavy chain (IGH) transcripts. To generate the TCR-Seq dataset, we have used T cell receptor alpha chain (TCRA)[28]. For both BCR-Seq and TCR-Seq we generated samples with read length between 50 and 100bp. Read coverage varied between 1 and 32. For all the samples, the mean fragment length was set to 200 bp.

Error correction methods designed for mixed genome

Among the error correction methods designed for single genome, Pollux is a unique method designed for metagenomics data composed of multiple microbial genomes. It also can work for sequencing data derived from a single genome. Pollux determines the number of occurrences of each observed k-mer in the data. The k-mer counts are used to determine k-mer depth profile for each read and localize sequencing errors.

Choosing k-mer size

We use k-mer sizes fanging from 18bp to 32bp for each of the datasets. In case error correction tool was equipped with an option for the genome size, we provide the length of corresponding

genome size. The genome size used for the T-Cell and B-Cell immune repertoire sequencing was 405,000 bp while the whole genome sequencing size used was 3,000,000 bp.

Evaluating error correction accuracy

The evaluation of error correction involves taking the error-free reads, the original raw reads, and the original reads corrected by computational error correction tools. Reads are then compared using multiple sequence alignment. We used MUSCLE[29] to perform multiple sequence alignment. Raw read represents the base before the error correction tool has been used. E.C. read represents the base after the error correction tool has been used. True read represents the correct base. True positive (TP) indicates a sequencing error was correctly changed. False negative (FN) indicates that either an error was ignored, or an error was incorrectly changed (**Figure S1**). False positive (FP) indicates a correct base was changed to an incorrect base. True negative (TN) indicates a correct base was left as is. Trimming was additionally evaluated as either TP or FP trimming. FN base calls were evaluated as either FN wrong if the base was changed incorrectly or just FN if the base was untouched and should have been corrected. (**Figure S2**)

Data compression format

Due to the quantity and size of the error corrected fastq files the evaluation of the reads were compressed. In order to summarize the errors that were not resolved by each of the various tools a method similar to the evaluation of error correction was utilized; although, in substitute for determining the number of TP, TN, FP from INDELs, FP from trimming, normal FP, and FN

bases the data compression will represent this data in the following reduced manner. The format is in the following order: read_name, length, TP, FN, FN WRONG, FP, FP INDEL, FP TRIM, TP TRIM. (Example: 1_22_238_1:0:0_3:0:0_0/1,100,3,0,0,0,0,0,0) By producing this data only when a TP, FP of an type, and FN are encountered this enables the extraction of the important features in the error correction of each read in a condensed form due to large volume of reads. This data is stored at the following link for future access:

https://drive.google.com/drive/u/0/folders/14ctrYlB5ldzwcYXG3MaoCmpqLK7SkIPZ

Performance estimation

We compared the performance of the error correction tools by reporting wall time, CPU time, and the maximum amount of RAM used by each of the tool. These performance metrics were obtained via -qsub option, with an additional -m bse option allowing automatically generated CPU and memory usage statistics. A typical node of the cluster used to benchmark the tools has dual twelve-core 2.2GHz Intel ES-2650v4 CPUs and an Intel 800GB DC S3510 Series MLC (6 Gb/s, 0.3 DWPD) 2.5" SATA SSD.

Supplementary Tables

Table S1. The size of kmers with the best performance for each tool for WGS data across various coverages ranging from 1x to 32x.

Method	**1x**	**2x**	**4x**	**8x**	**16x**	**32x**
BFC	32	32	32	32	23	23
Bless	32	32	32	32	32	32
Coral	32	32	32	32	32	32
Lighter	32	32	32	32	32	32
Musket	32	32	32	32	32	32
Pollux	32	32	32	32	32	32
Reckoner	32	32	32	32	32	32
SGA	32	32	32	32	32	32

Method	**1x**	**2x**	**4x**	**8x**	**16x**	**32x**
BFC	24	24	24	24	24	24
Bless	24	24	24	24	24	24
Coral	24	24	24	24	24	24
Lighter	24	24	24	24	24	24
Musket	24	19	24	24	24	24
Pollux	24	23	23	23	19	19
Reckoner	18	18	18	18	18	18
SGA	24	24	24	24	24	24

Table S2. The size of kmers with the best performance for each tool for BCR-Seq data across various coverages ranging from 1x to 128x.

Method	**1x**	**2x**	**4x**	**8x**	**16x**	**32x**
BFC	24	24	24	24	24	24
Bless	24	24	24	24	24	24
Coral	24	24	24	24	24	24
Lighter	24	24	24	24	24	24
Musket	24	24	24	24	24	24
Pollux	24	24	24	24	24	24
Reckoner	24	24	24	24	24	24
SGA	24	24	24	24	24	24

Table S3. The size of the kmers with the best performance for each tool for TCR-Seq data across various coverages ranging from 1x to 32x.

Supplementary Figures

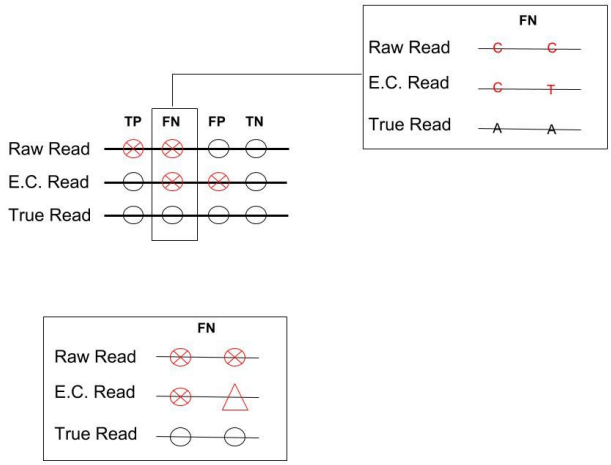


Figure S1. Various scenarios of error correction at the base level. Raw read represents the base before the error correction tool has been used. E.C. read represents the base after the error correction tool has been used. True read represents the correct base. True positive (TP) indicates a sequencing error was correctly changed. False negative (FN) indicates that either an error was ignored, or an error was incorrectly changed. False positive (FP) indicates a correct base was changed to an incorrect base. True negative (TN) indicates a correct base was left as is.

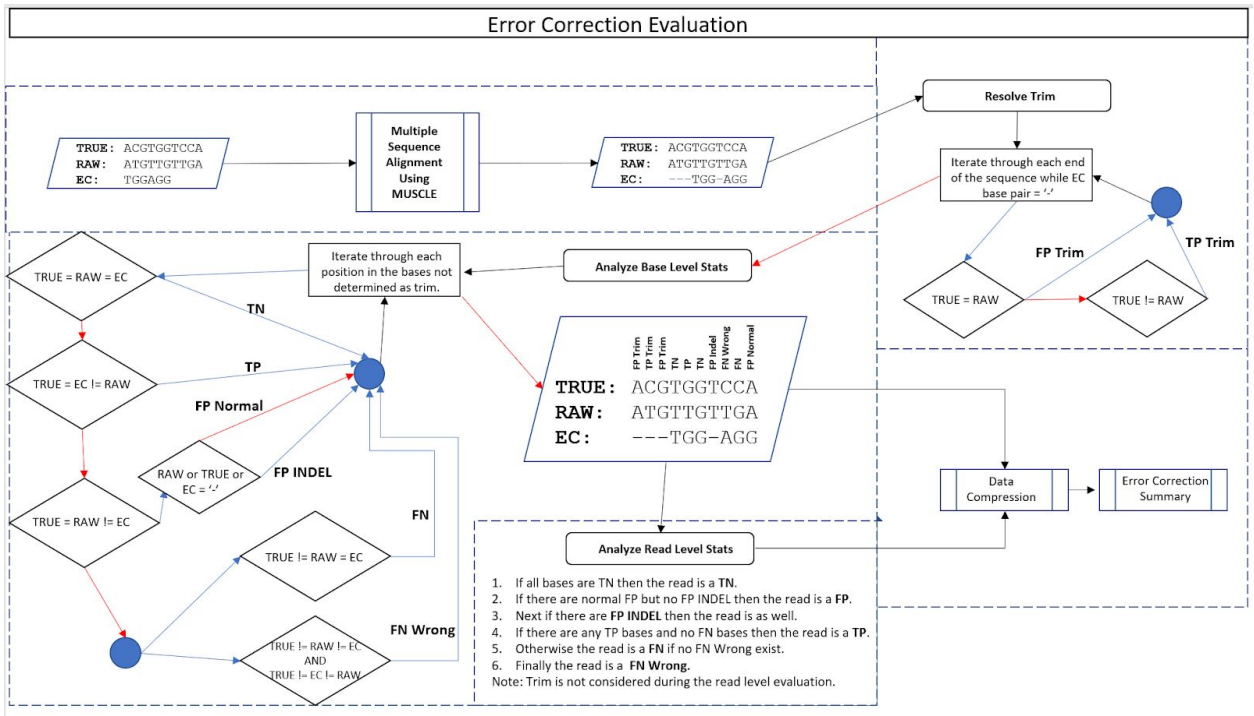
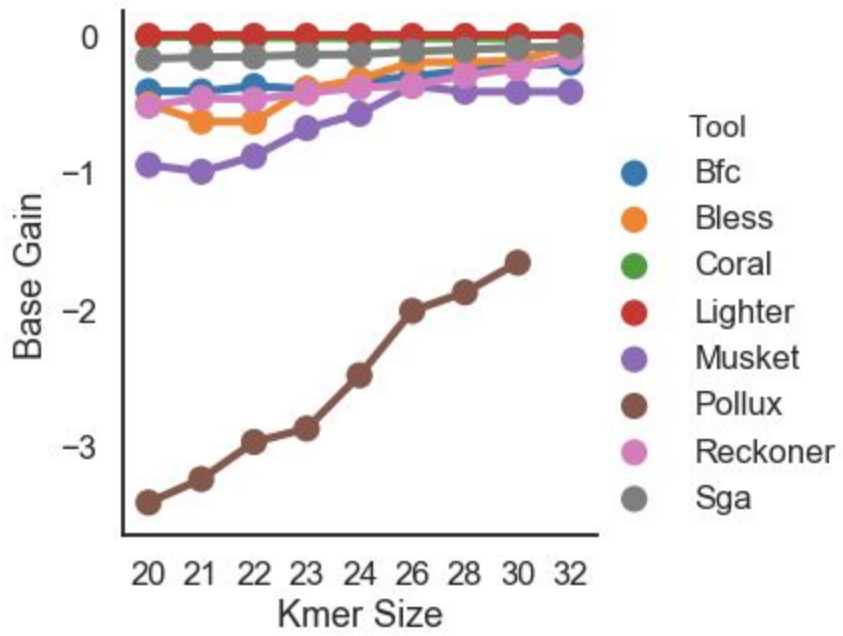
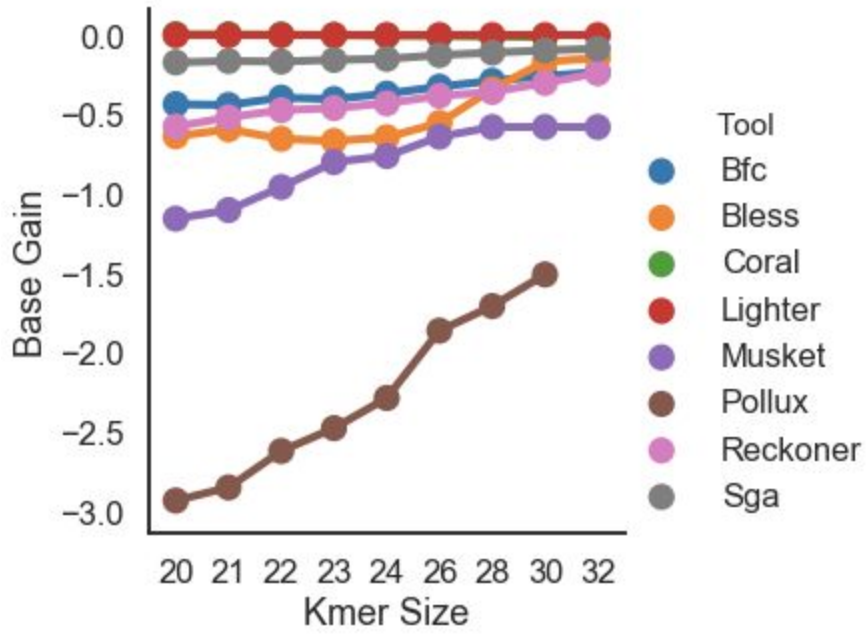


Figure S2: Methodology to evaluate the accuracy of error correction methods. The image above displays the series of events that take place in order to progress from the error-corrected read supplied from any one of the tools mentioned in Table 1**.** MSA is first performed to enable an analysis of each base in comparison to the "gold standard", or proposed true read, as well as the raw read prior to error correction. The progression of events following MSA is Resolve Trim, Analyze Base Level Stats, Analyze Read Level Stats, Data Compression, and finally the Error Correction Summary. Algorithmic details for Resolving Trim, Analyze Base Level Stats, and Analyze Read Level Stats are shown above within each of the relevant dashed boxes. Red or blue arrows represent whether a decision was false or true; in addition, a red or blue arrow represents whether a process iteration is over or still in progress.

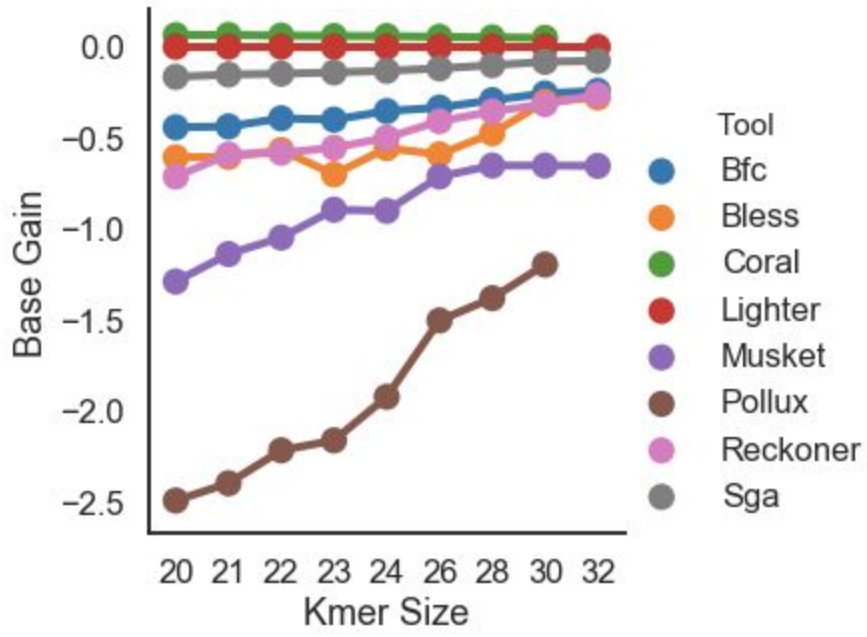
(a)


0
−1
−2
−3
Base Gain
20 21 22 23 24 26 28 30 32
Kmer Size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

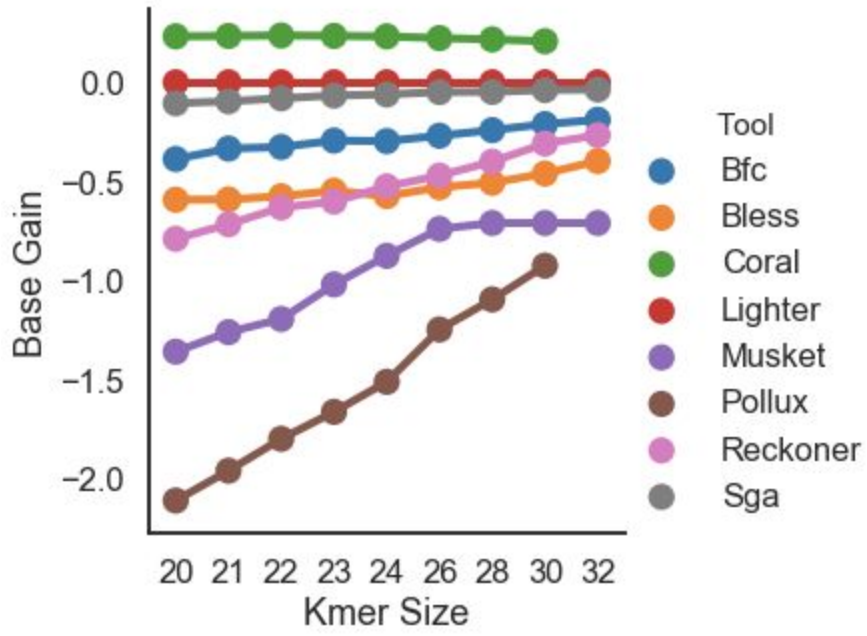
(b)


0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
Base Gain
20 21 22 23 24 26 28 30 32
Kmer Size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

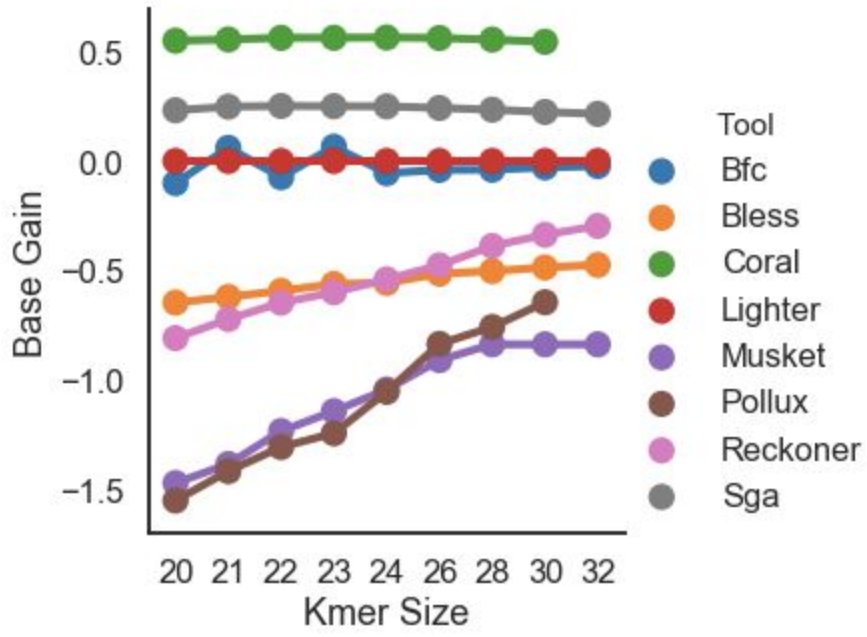
(c)


0.0
−0.5
−1.0
−1.5
−2.0
−2.5
Base Gain
20 21 22 23 24 26 28 30 32
Kmer Size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(d)


0.0
−0.5
−1.0
−1.5
−2.0
Base Gain
20 21 22 23 24 26 28 30 32
Kmer Size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(e)


0.5
0.0
−0.5
−1.0
−1.5
Base Gain
20 21 22 23 24 26 28 30 32
Kmer Size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(f)

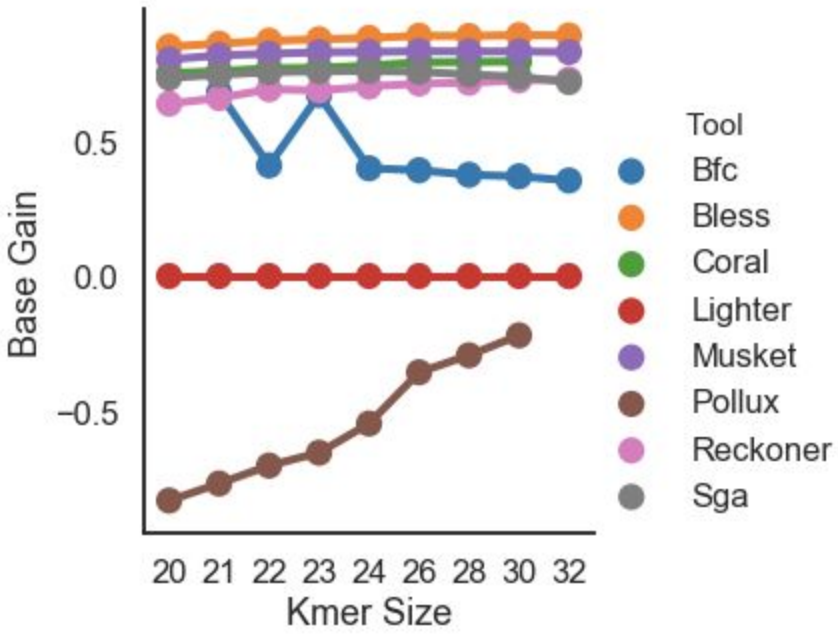
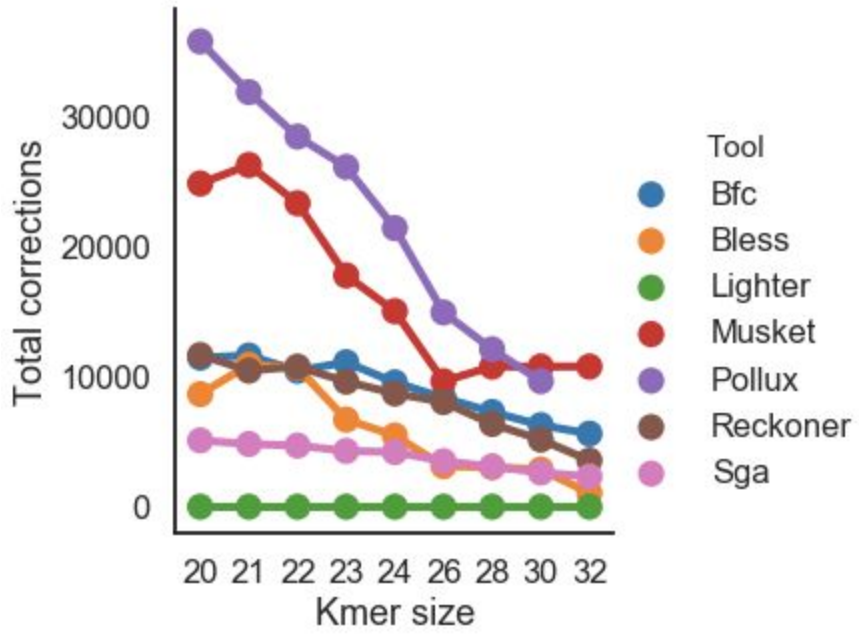
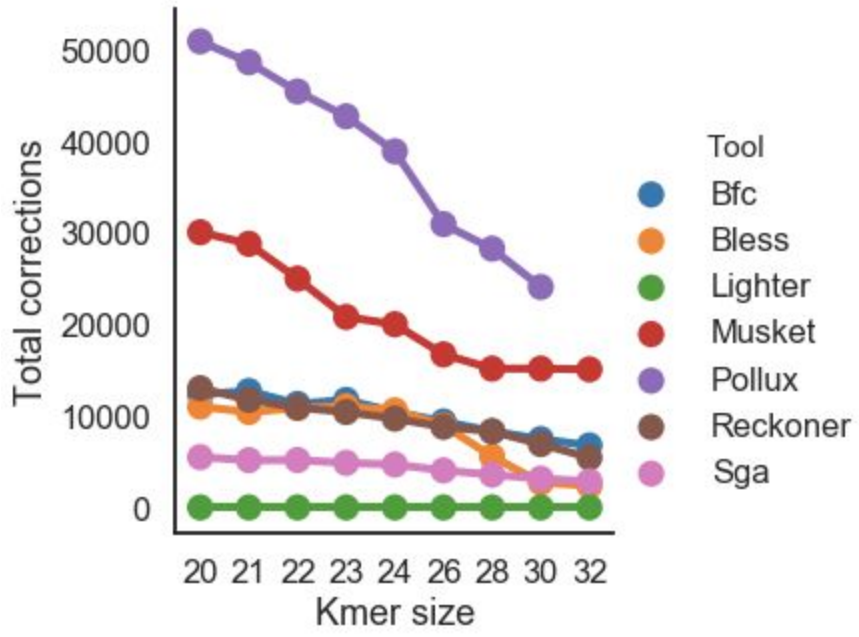


Figure S3. The effect of the kmer size on the accuracy of the error correction across various coverages settings for WGS data. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

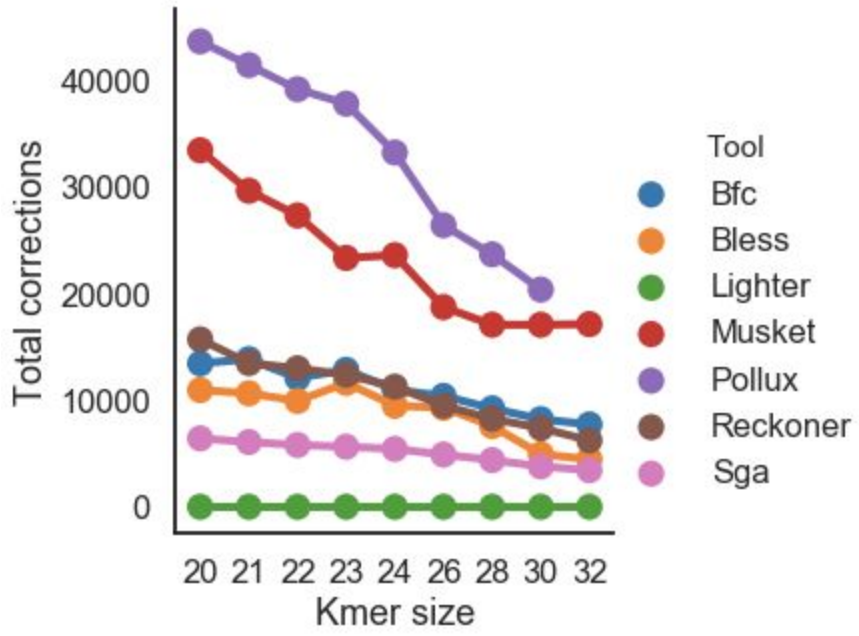
(a)


Total corrections
30000
20000
10000
0
20 21 22 23 24 26 28 30 32
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

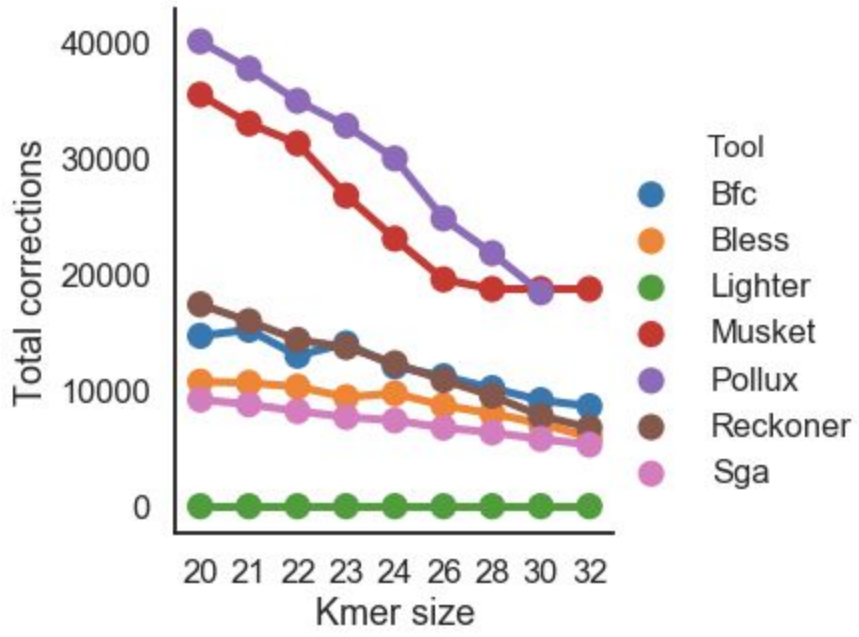
(b)


Total corrections
50000
40000
30000
20000
10000
0
20 21 22 23 24 26 28 30 32
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

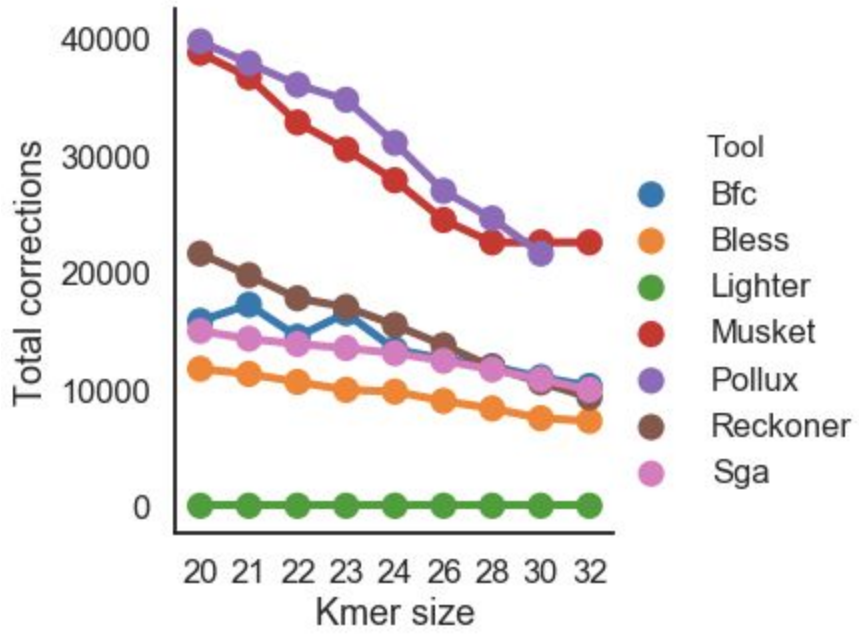
(c)


Total corrections
40000
30000
20000
10000
0
20 21 22 23 24 26 28 30 32
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

(d)


Total corrections
40000
30000
20000
10000
0
20 21 22 23 24 26 28 30 32
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

(e)


40000
30000
20000
10000
0
Total corrections
20 21 22 23 24 26 28 30 32
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

(f)

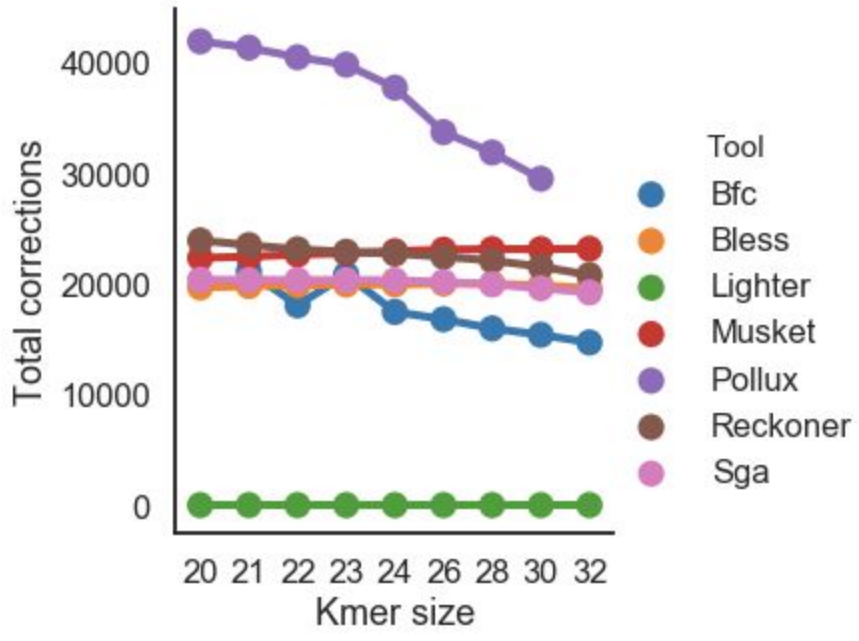
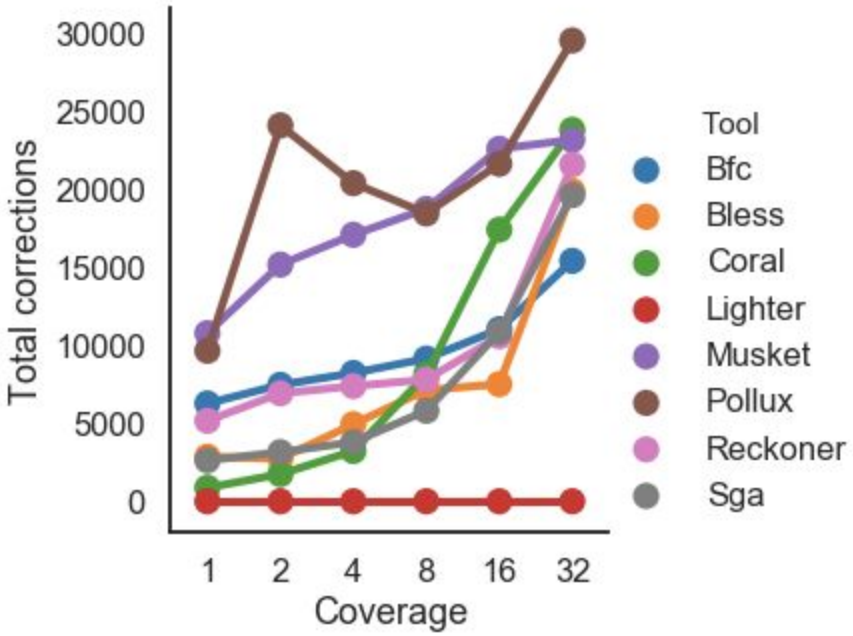
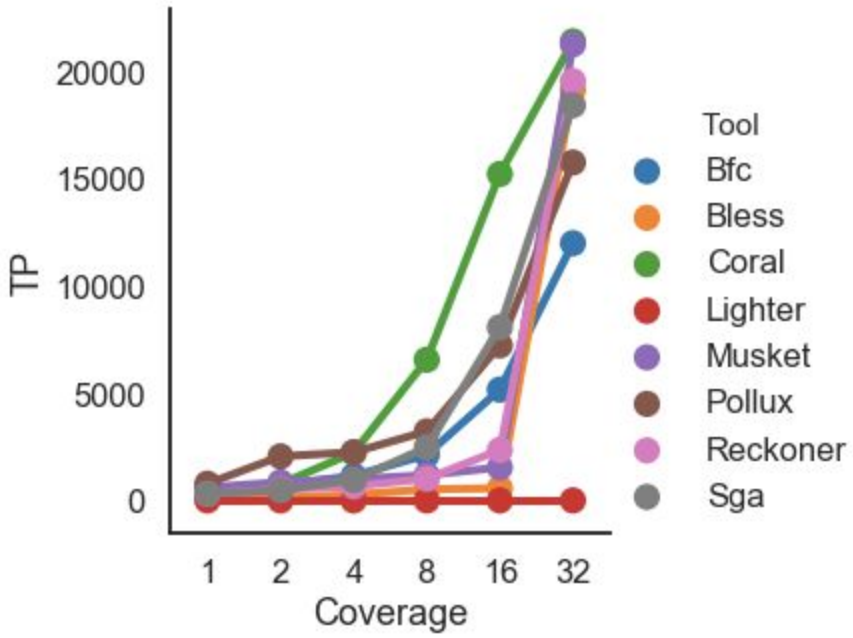

40000
30000
20000
10000
0
Total corrections
20 21 22 23 24 26 28 30 32
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

Figure S4. The effect of kmer size on the total number of corrections performed by error correction methods across various coverages settings of WGS datasets. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

(a)



(b)


20000
15000
10000
5000
0
TP
1
2
4
8
16
32
Coverage
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(c)

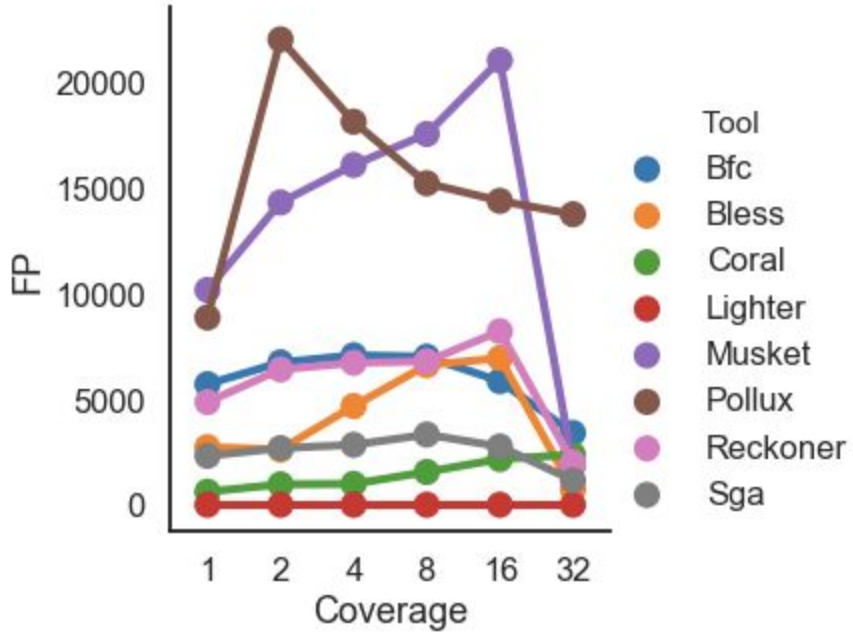
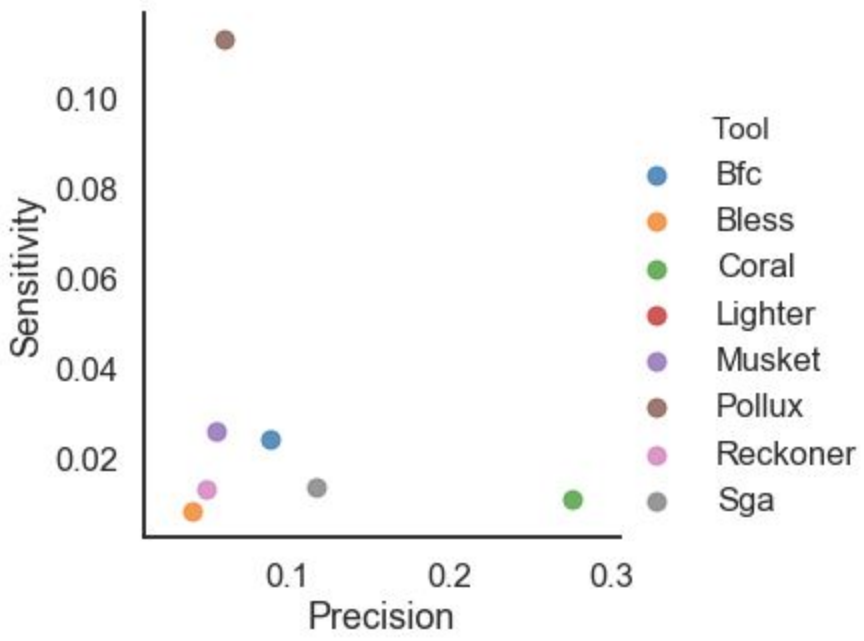
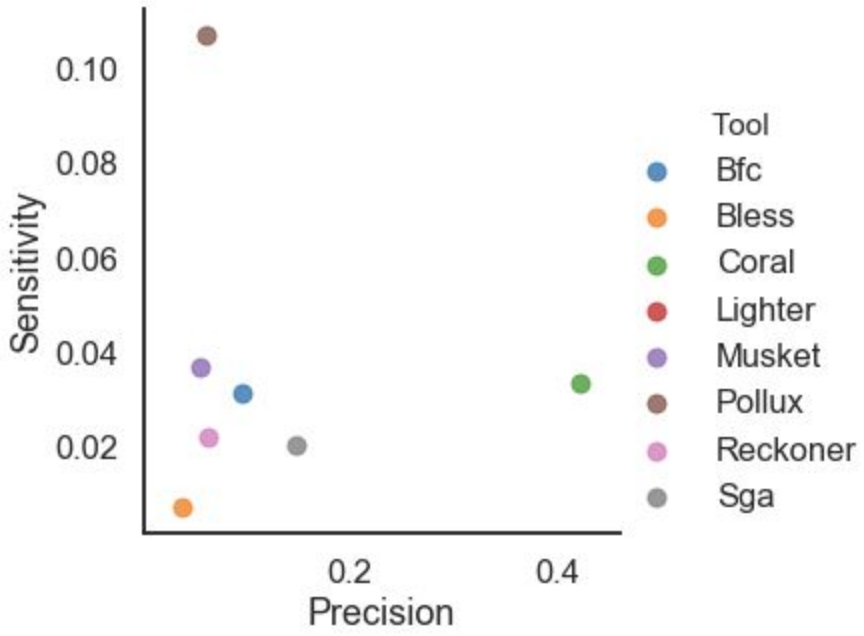

20000
15000
10000
5000
0
FP
1
2
4
8
16
32
Coverage
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

Figure S5. The effect of coverage on the accuracy of the error correction tools across various coverages settings for WGS datasets. (a) Total number of corrections (b) Number of TP corrections (c) Number of FP corrections

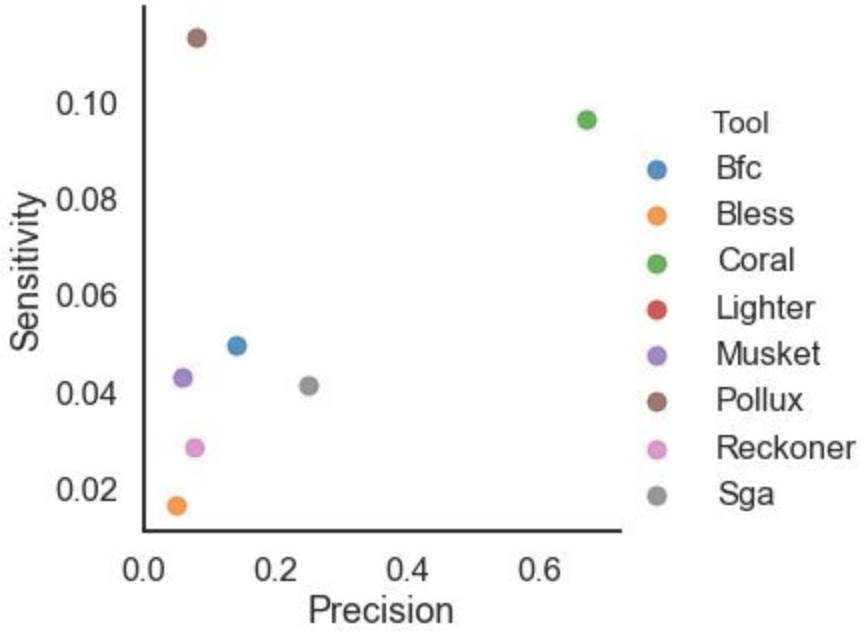
(a)



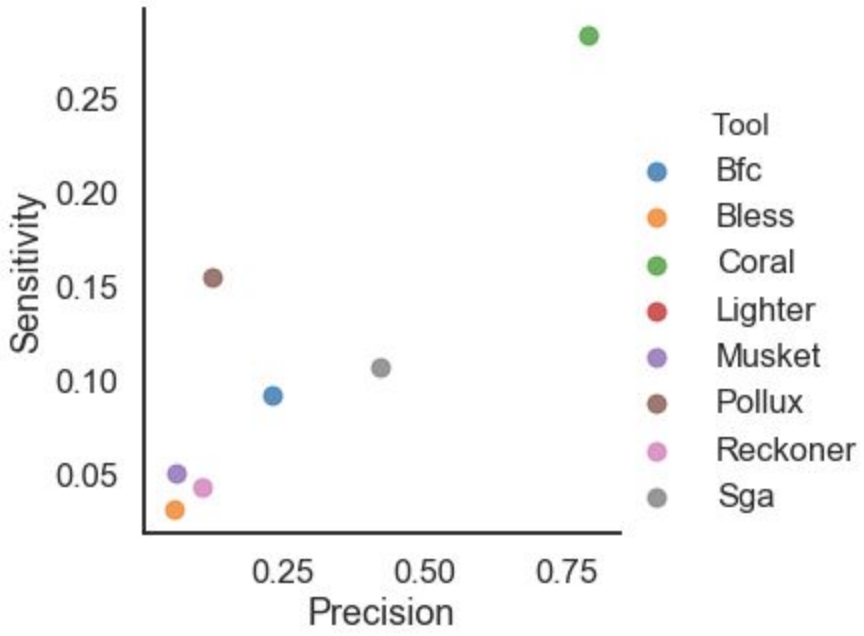
(b)


0.10
0.08
0.06
0.04
0.02
Sensitivity
0.2
0.4
Precision
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

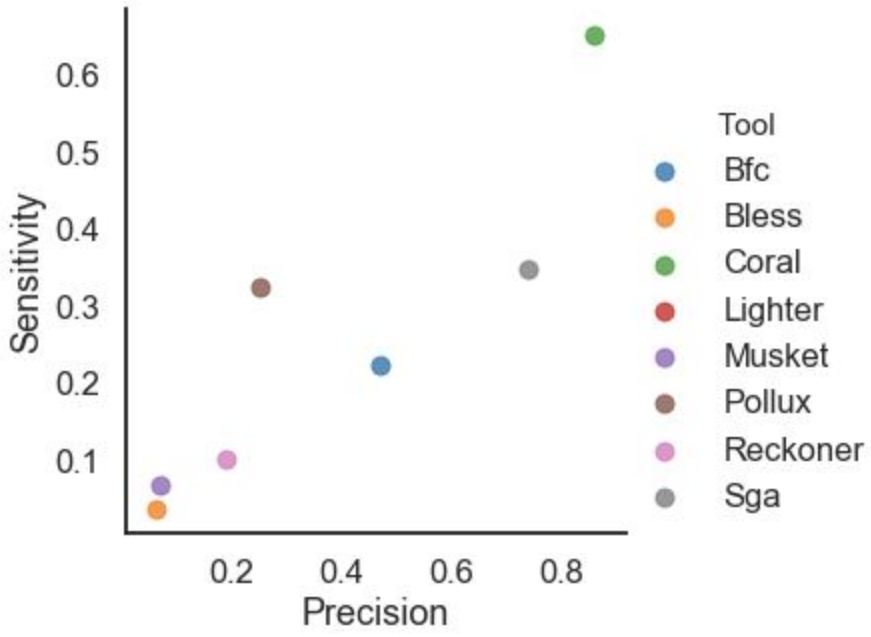
(c)


0.10
0.08
0.06
0.04
0.02
Sensitivity
0.0
0.2
0.4
0.6
Precision
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(d)


Sensitivity
0.25
0.20
0.15
0.10
0.05
0.25
0.50
0.75
Precision
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(e)


Sensitivity
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.4
0.6
0.8
Precision
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(f)

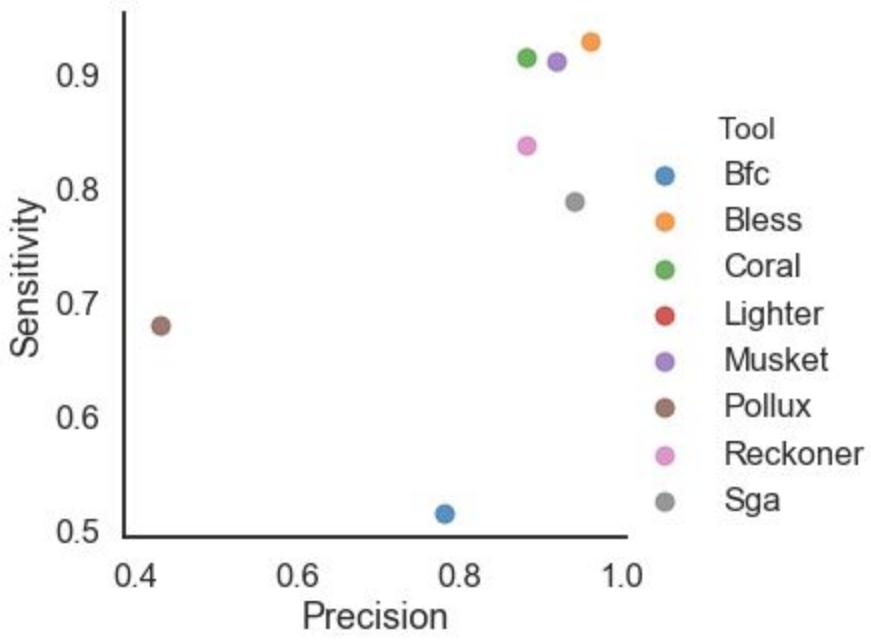


Figure S6. The effect of coverage on the accuracy of the error correction tools across various coverages settings for WGS datasets. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

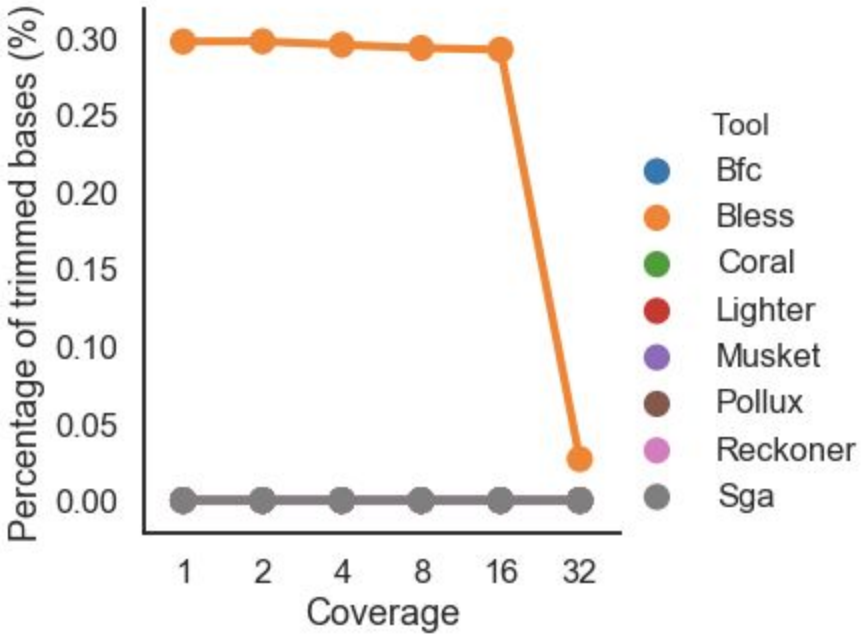


Figure S7. The number of trimmed across various coverages settings.

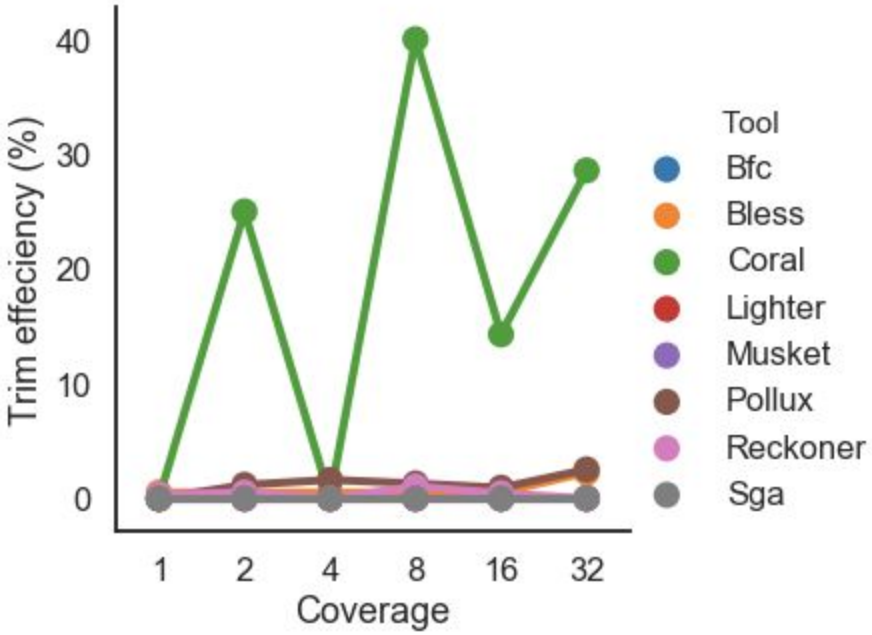
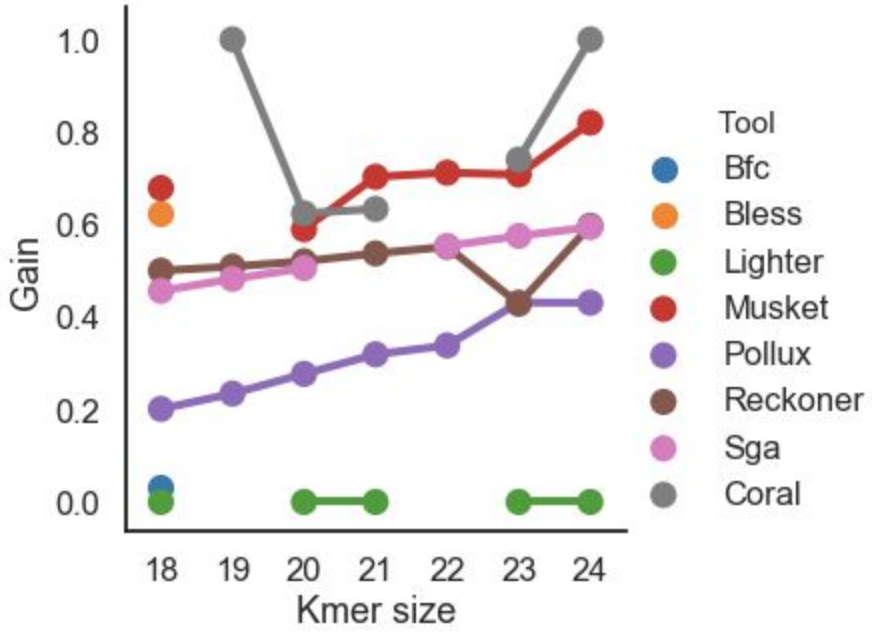
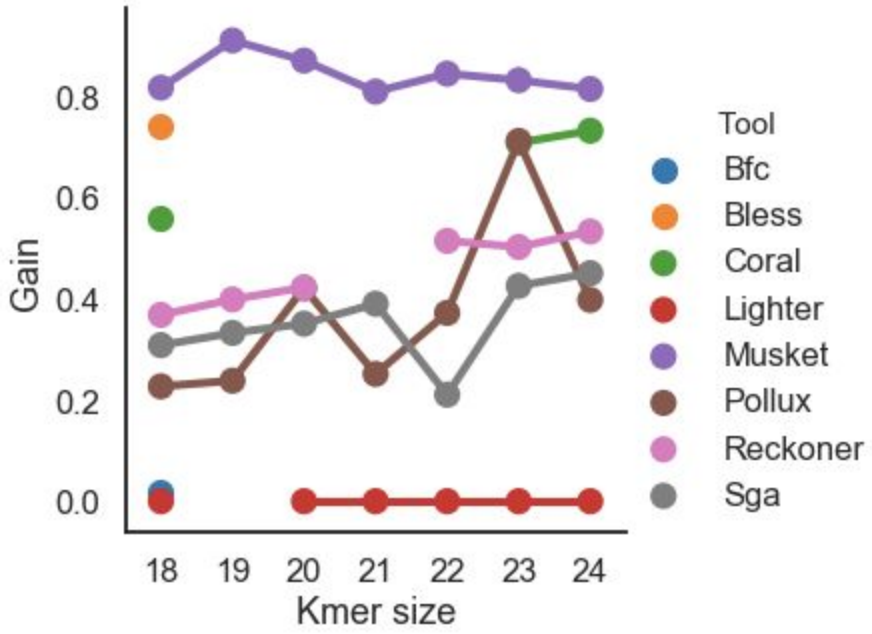


Figure S8. The efficiency of trimming across various coverages settings.

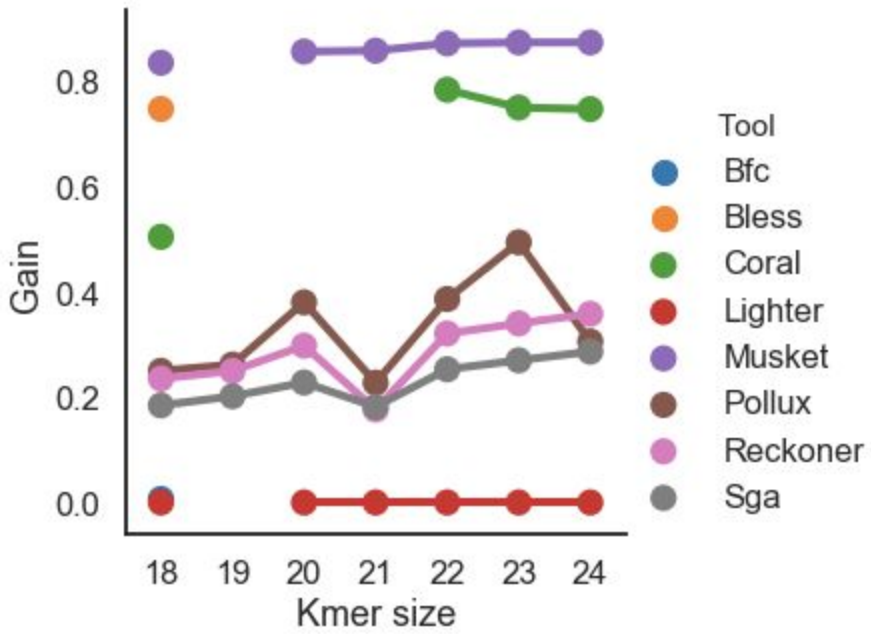
(a)


1.0
0.8
0.6
0.4
0.2
0.0
Gain
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga
Coral

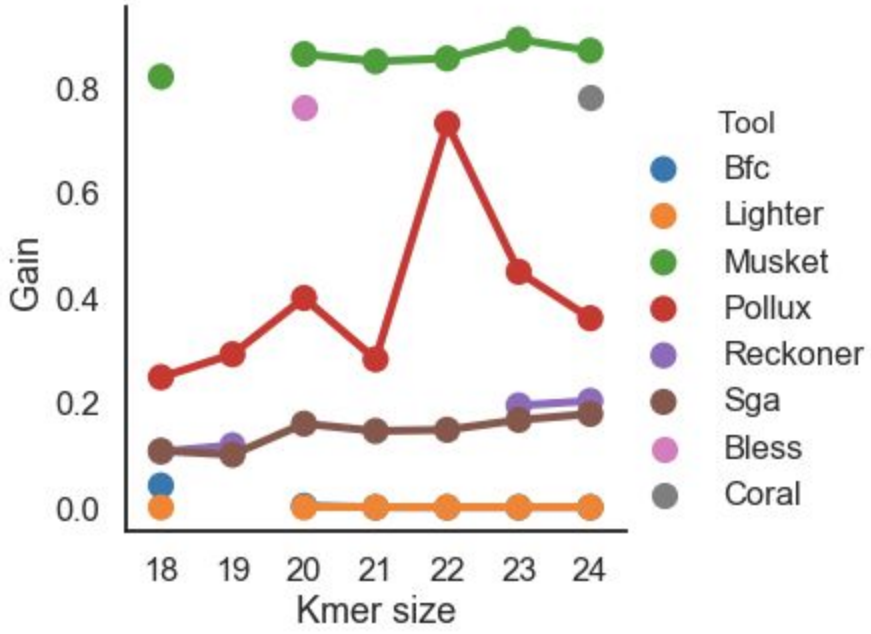
(b)


0.8
0.6
0.4
0.2
0.0
Gain
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

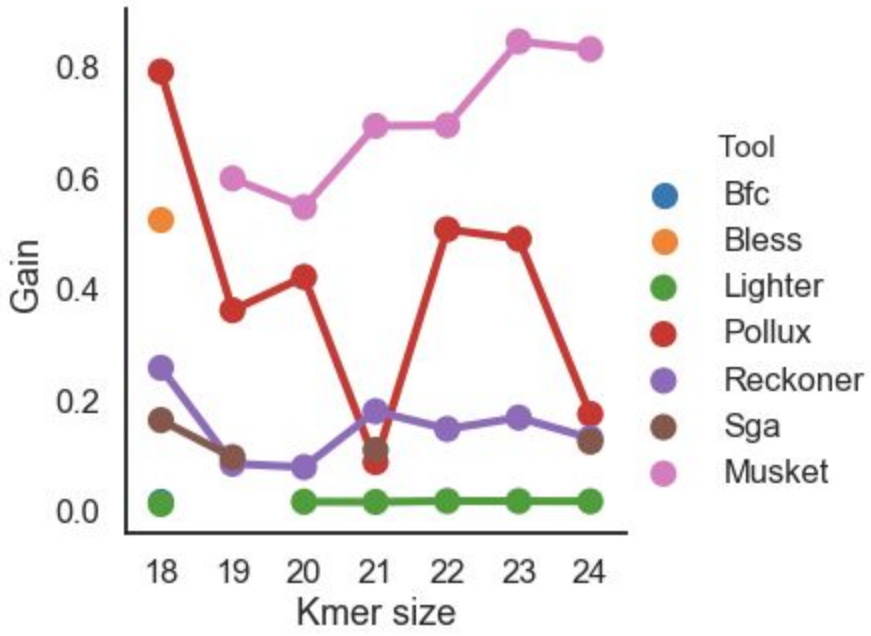
(c)


Gain
0.0
0.2
0.4
0.6
0.8
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(d)


Gain
0.0
0.2
0.4
0.6
0.8
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Lighter
Musket
Pollux
Reckoner
Sga
Bless
Coral

(e)


0.8
0.6
0.4
0.2
0.0
Gain
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Lighter
Pollux
Reckoner
Sga
Musket

(f)

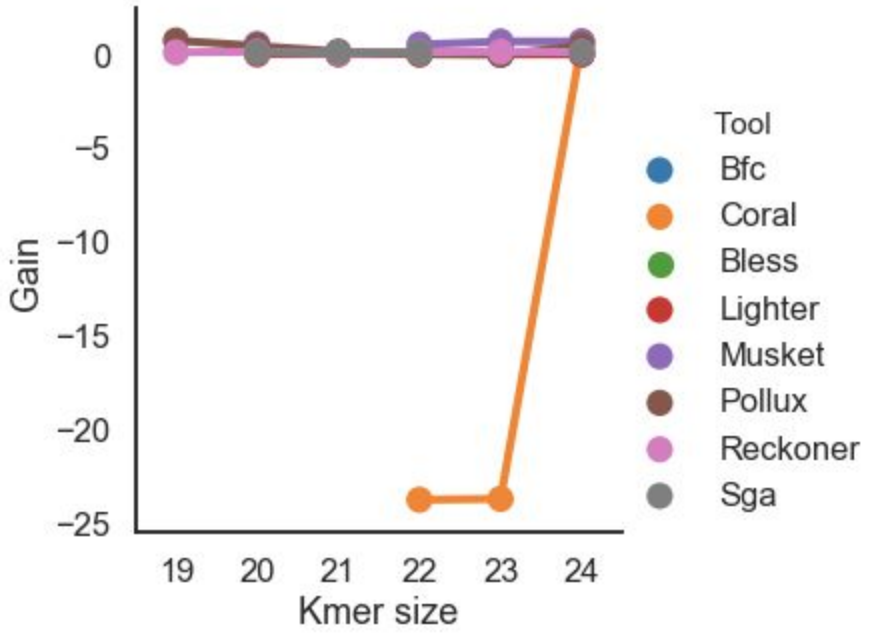
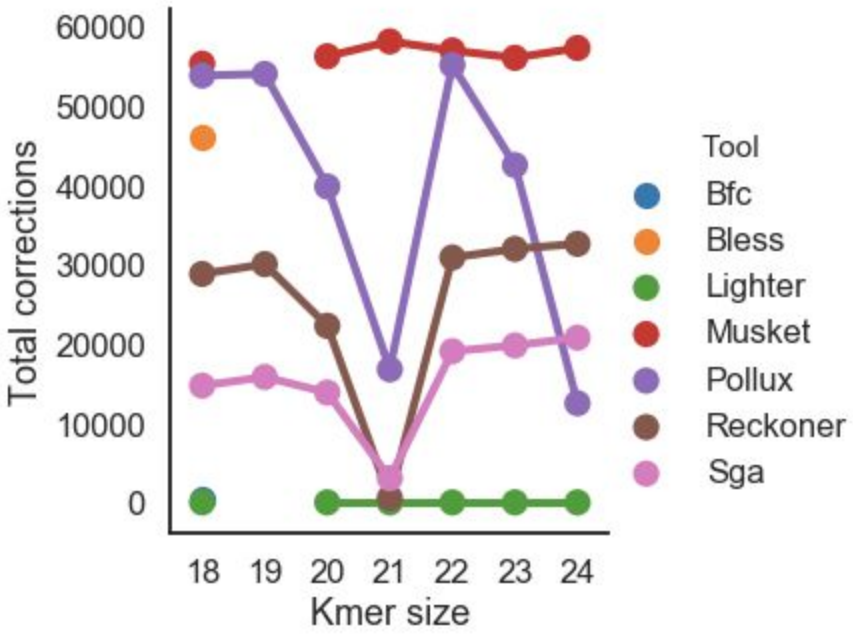
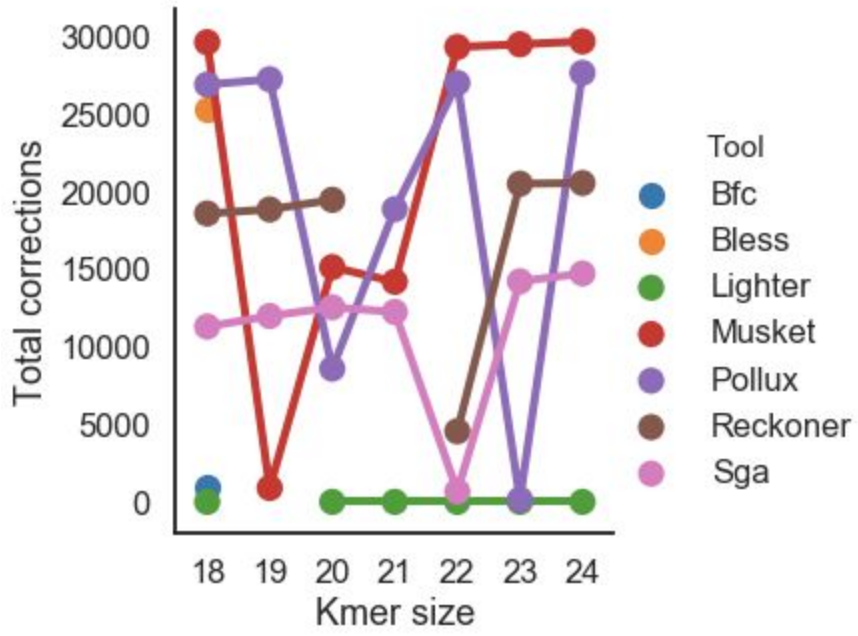

0
−5
−10
−15
−20
−25
Gain
19
20
21
22
23
24
Kmer size
Tool
Bfc
Coral
Bless
Lighter
Musket
Pollux
Reckoner
Sga

Figure S9. The effect of kmer size on the accuracy of the error correction tools across various coverages settings for BCR-Seq data. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

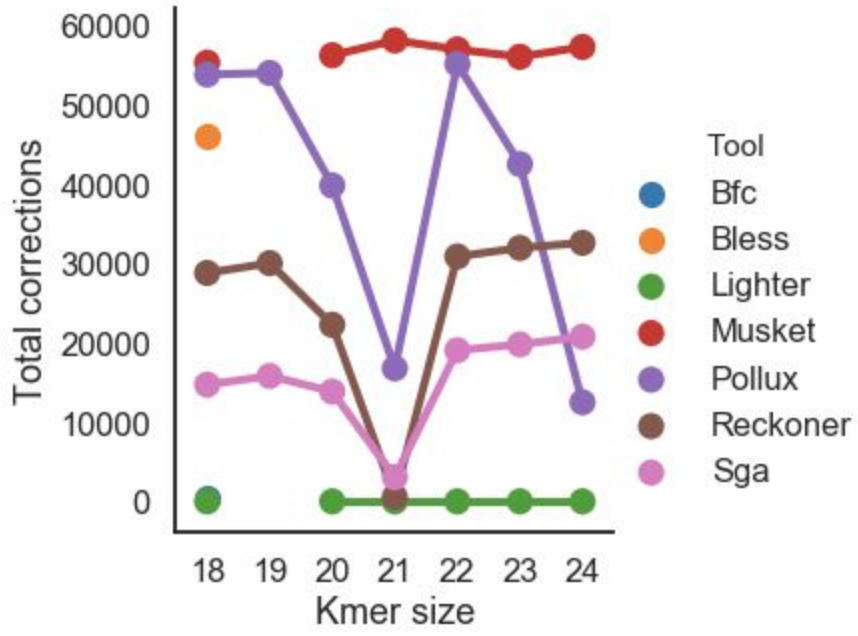
(a)



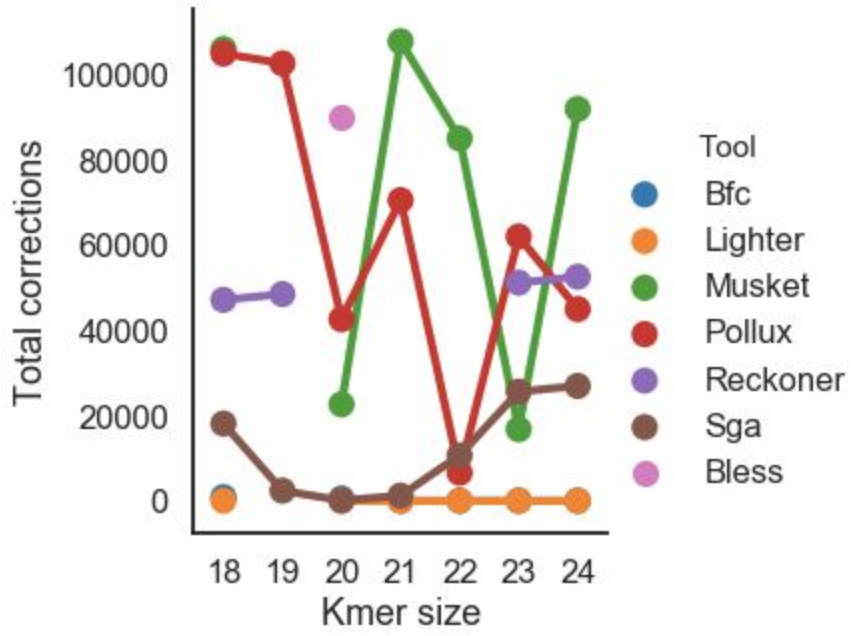
(b)


30000
25000
20000
15000
10000
5000
0
Total corrections
18 19 20 21 22 23 24
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

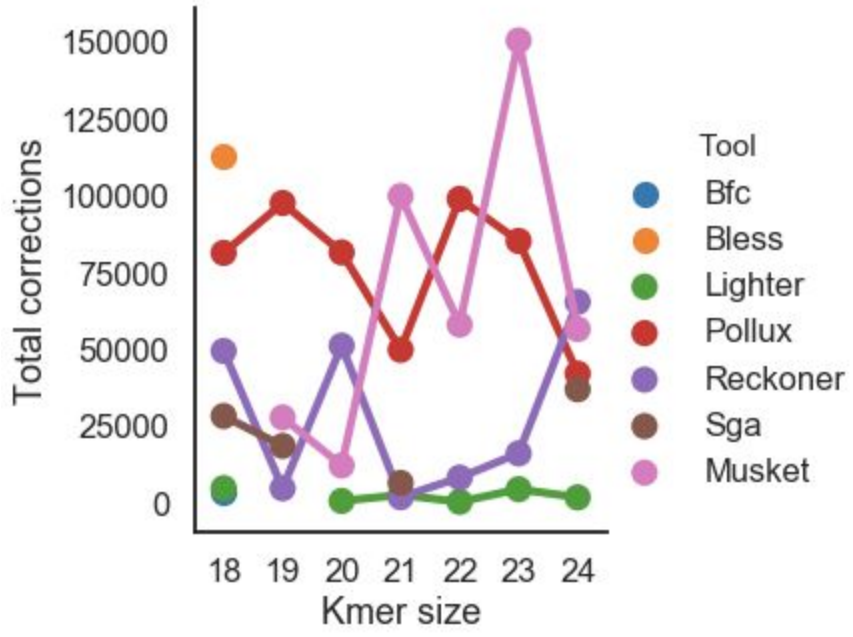
(c)


60000
50000
40000
30000
20000
10000
0
Total corrections
18 19 20 21 22 23 24
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

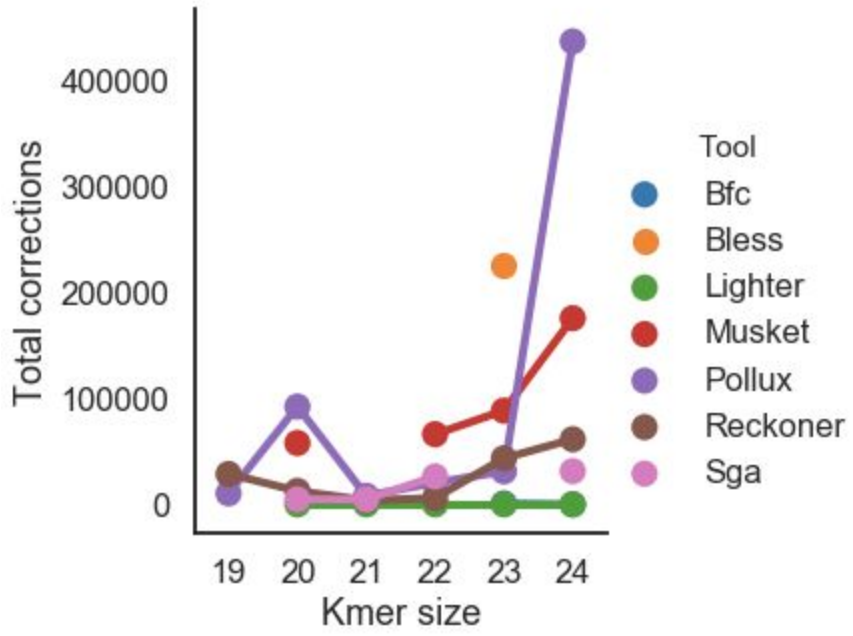
(d)


Total corrections
100000
80000
60000
40000
20000
0
18 19 20 21 22 23 24
Kmer size
Tool
Bfc
Lighter
Musket
Pollux
Reckoner
Sga
Bless

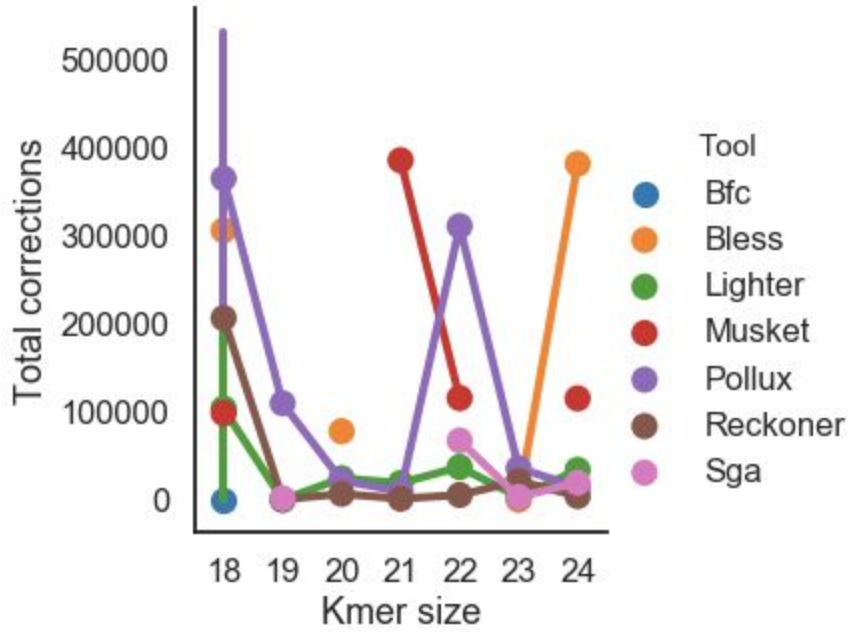
(e)


Total corrections
150000
125000
100000
75000
50000
25000
0
18 19 20 21 22 23 24
Kmer size
Tool
Bfc
Bless
Lighter
Pollux
Reckoner
Sga
Musket

(f)


Total corrections
400000
300000
200000
100000
0
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

(g)


Total corrections
500000
400000
300000
200000
100000
0
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Lighter
Musket
Pollux
Reckoner
Sga

(h)

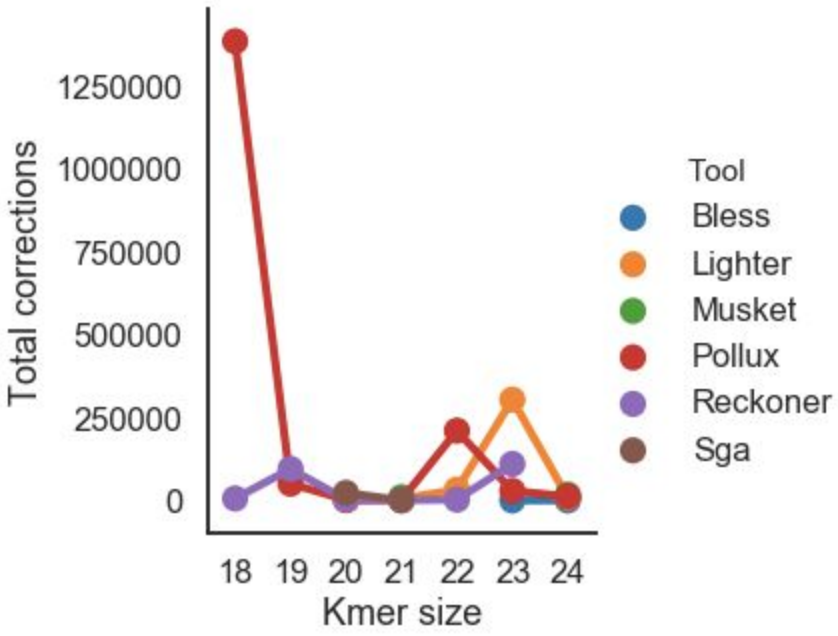
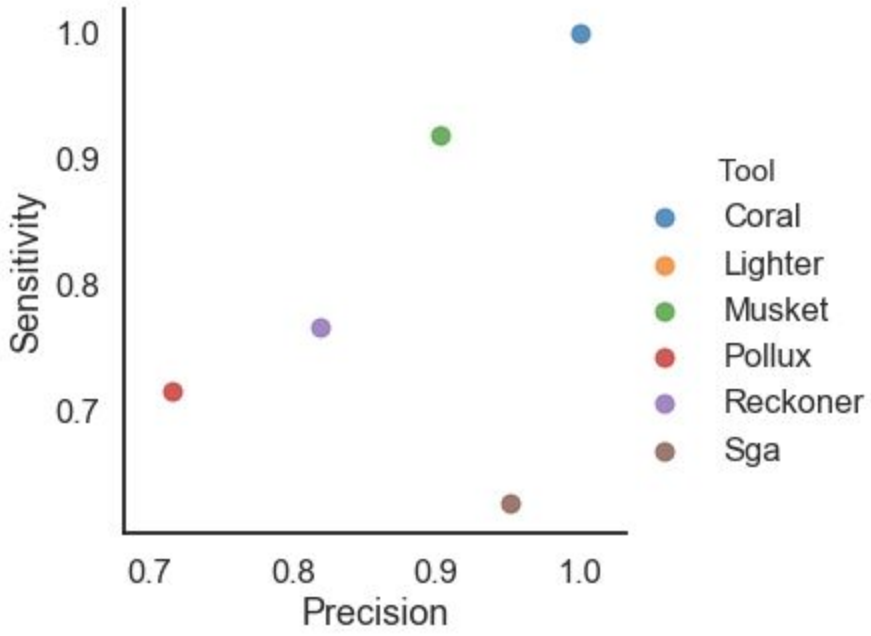
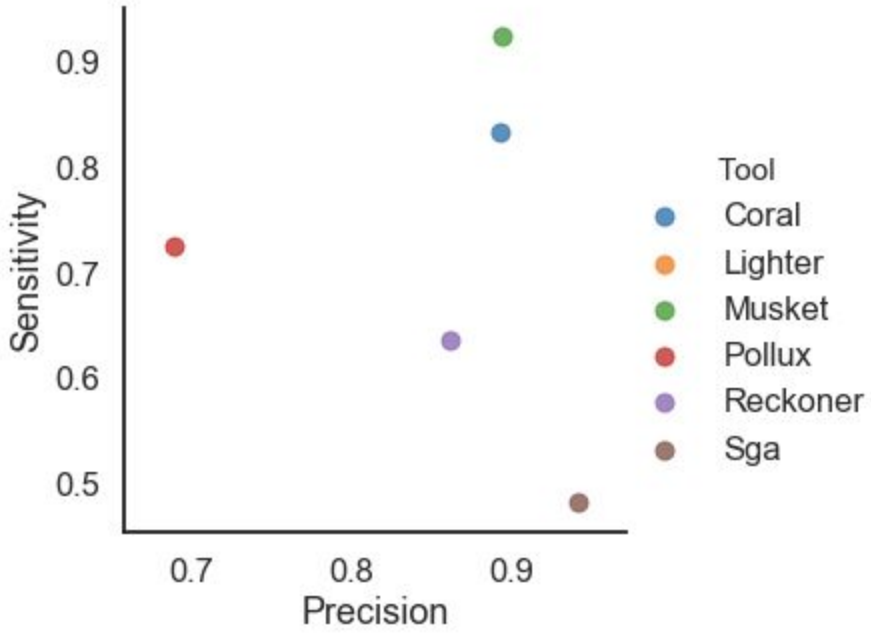


Figure S10. The effect of kmer size on the total number of corrections performed by error correction methods across various coverages settings for BCR-Seq data. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x (g) 64x (h) 128x

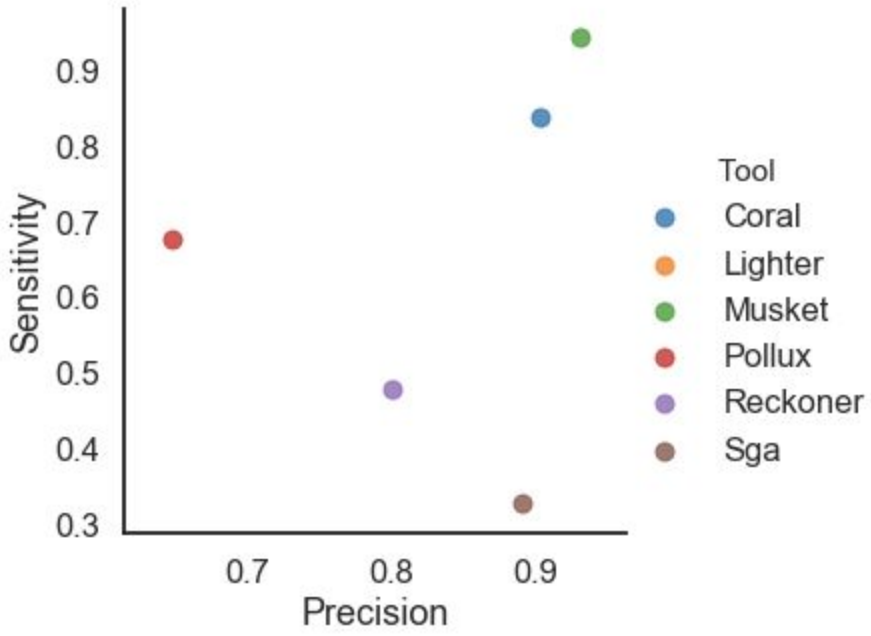
(a)


1.0
0.9
0.8
0.7
Sensitivity
0.7
0.8
0.9
1.0
Precision
Tool
Coral
Lighter
Musket
Pollux
Reckoner
Sga

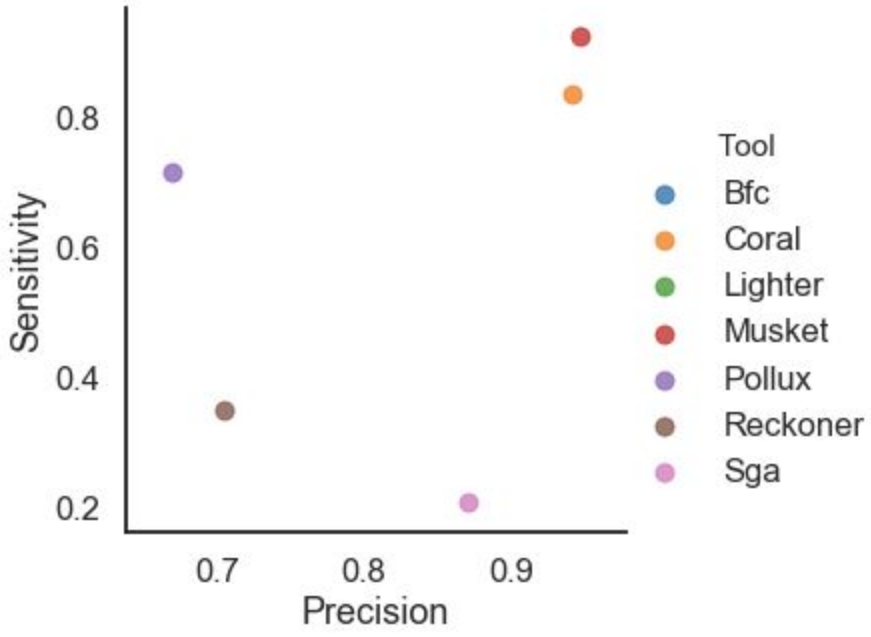
(b)


0.9
0.8
0.7
0.6
0.5
Sensitivity
0.7
0.8
0.9
Precision
Tool
Coral
Lighter
Musket
Pollux
Reckoner
Sga

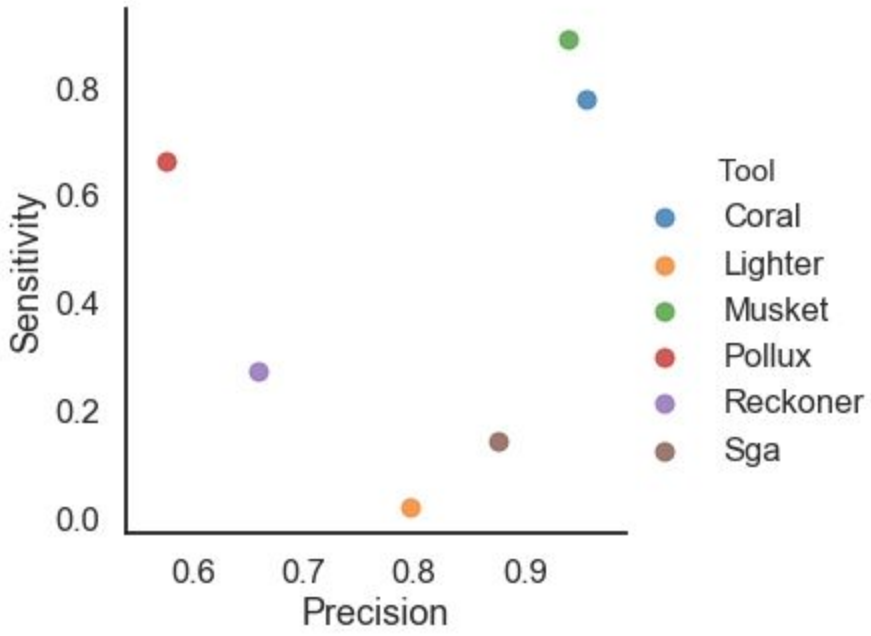
(c)


Sensitivity
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.7
0.8
0.9
Precision
Tool
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(d)


Sensitivity
0.8
0.6
0.4
0.2
0.7
0.8
0.9
Precision
Tool
Bfc
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(e)


Sensitivity
0.0
0.2
0.4
0.6
0.8
0.6
0.7
0.8
0.9
Precision
Tool
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(f)

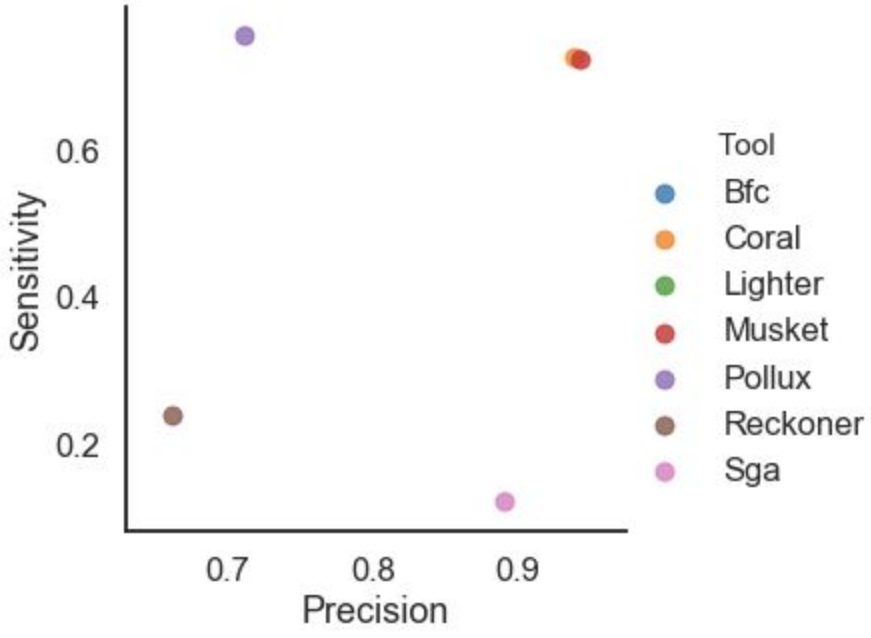
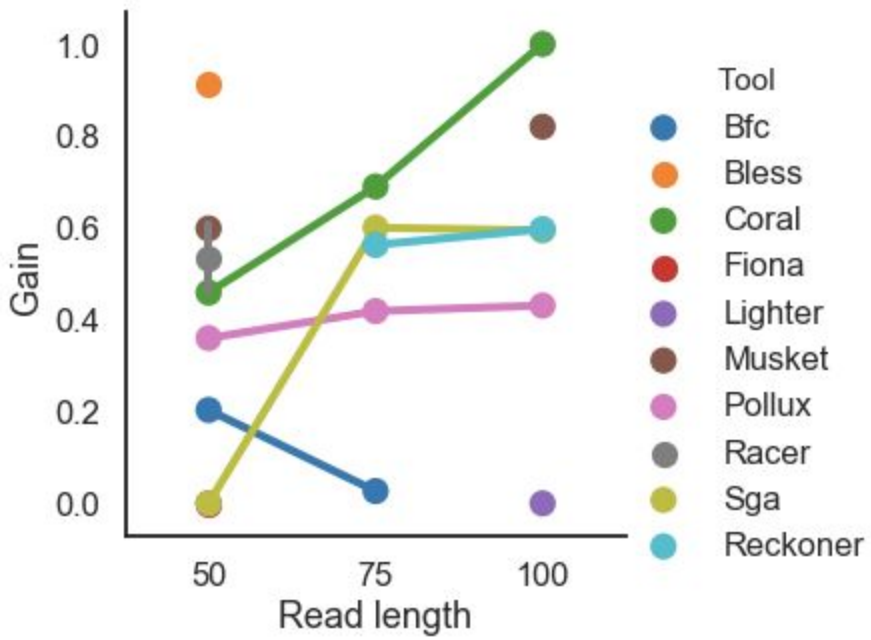
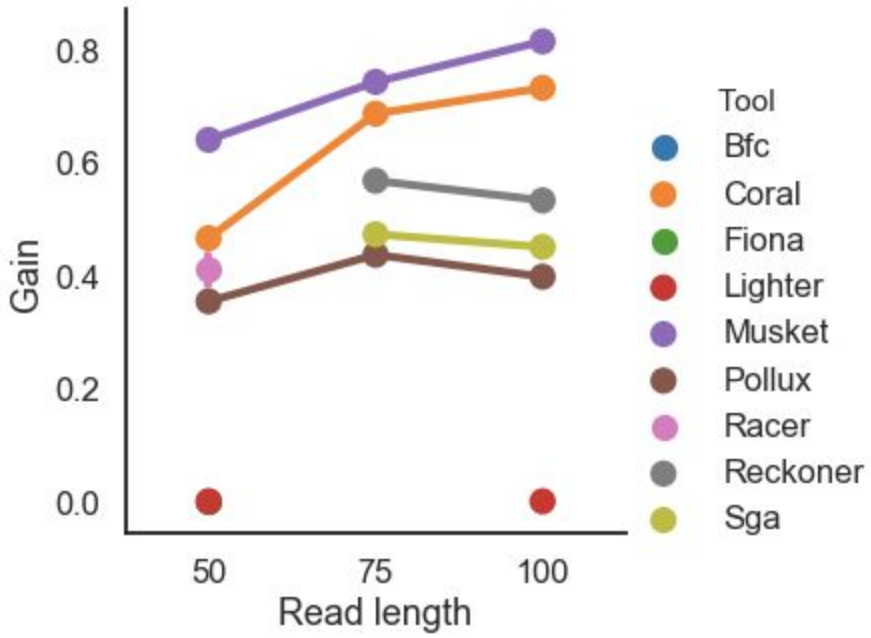

Sensitivity
0.2
0.4
0.6
0.7
0.8
0.9
Precision
Tool
Bfc
Coral
Lighter
Musket
Pollux
Reckoner
Sga

Figure S11. The effect of coverage on the accuracy of the error correction tools across various coverages settings for WGS datasets. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

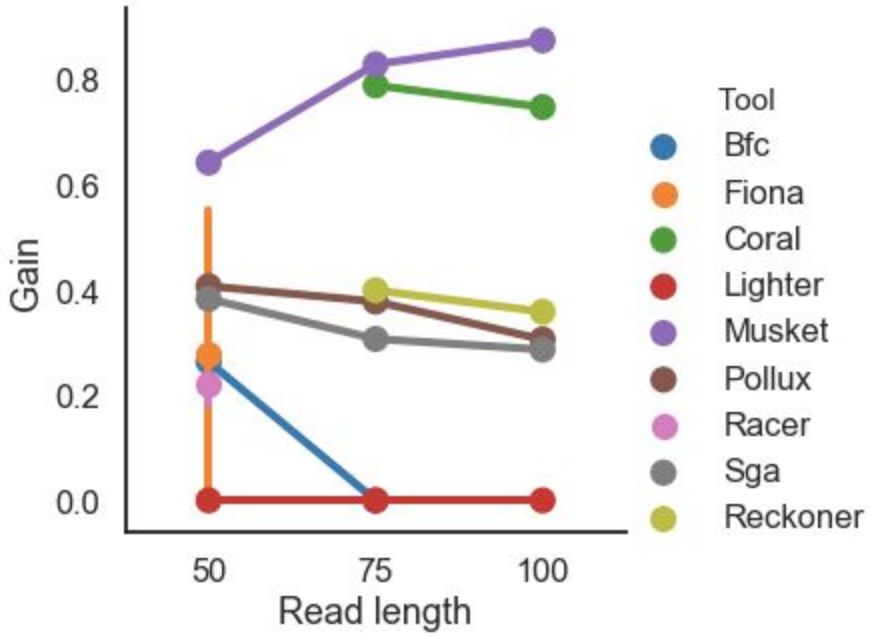
(a)



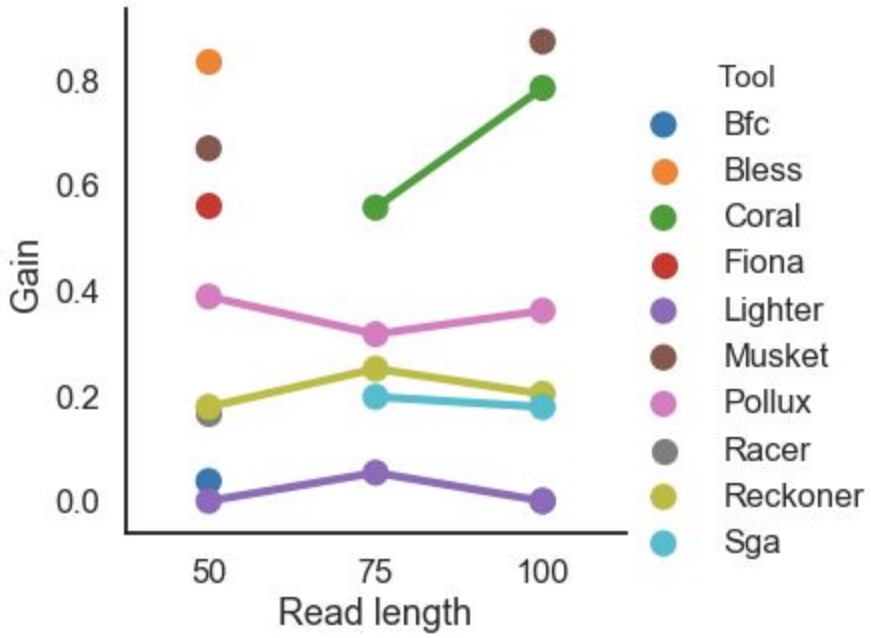
(b)


0.8
0.6
0.4
0.2
0.0
Gain
50
75
100
Read length
Tool
Bfc
Coral
Fiona
Lighter
Musket
Pollux
Racer
Reckoner
Sga

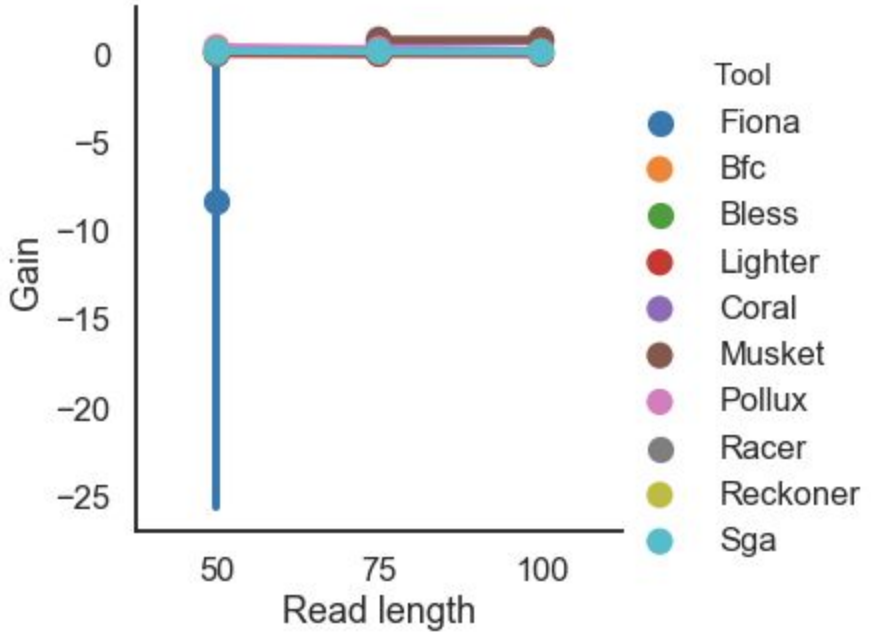
(c)


0.8
0.6
0.4
0.2
0.0
Gain
50
75
100
Read length
Tool
Bfc
Fiona
Coral
Lighter
Musket
Pollux
Racer
Sga
Reckoner

(d)


Gain
0.8
0.6
0.4
0.2
0.0
50
75
100
Read length
Tool
Bfc
Bless
Coral
Fiona
Lighter
Musket
Pollux
Racer
Reckoner
Sga

(e)


Gain
0
−5
−10
−15
−20
−25
50
75
100
Read length
Tool
Fiona
Bfc
Bless
Lighter
Coral
Musket
Pollux
Racer
Reckoner
Sga

(f)

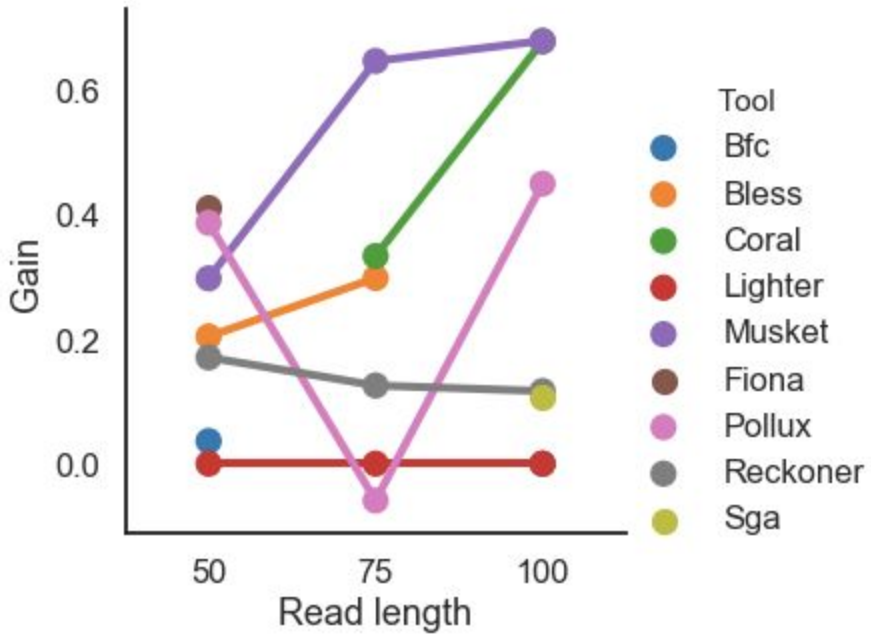
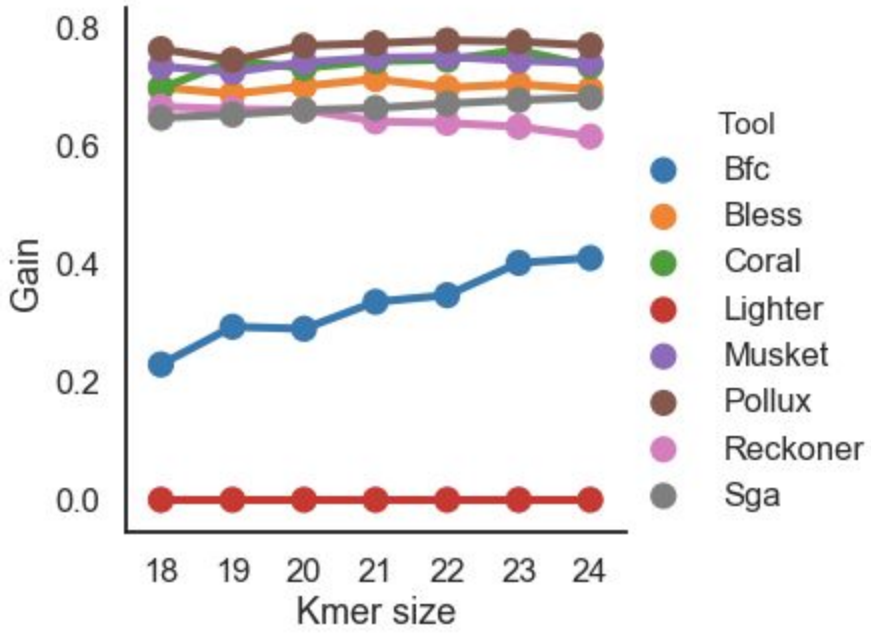
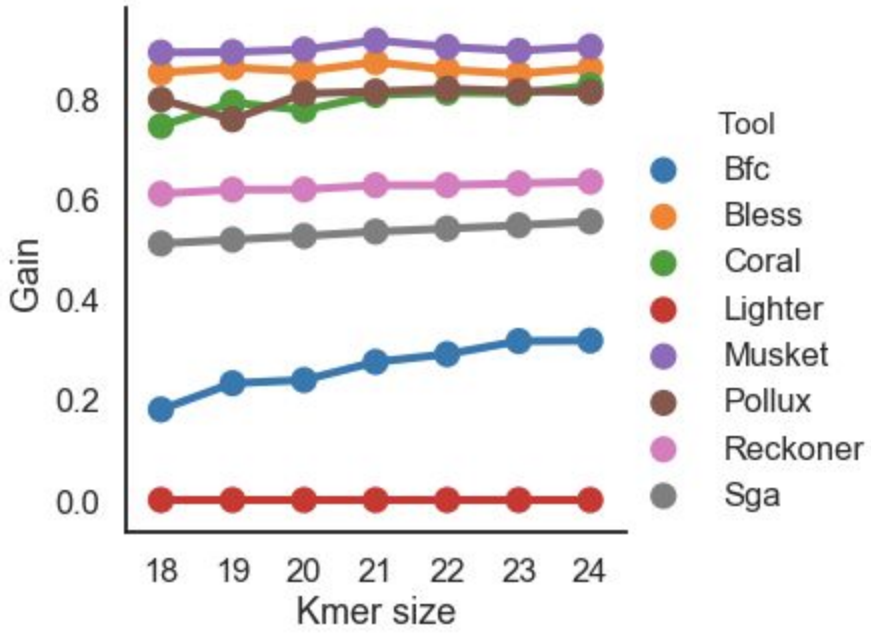


Figure S11. The effect of the read length on the accuracy of the error correction tools across various coverages settings for BCR-Seq data (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

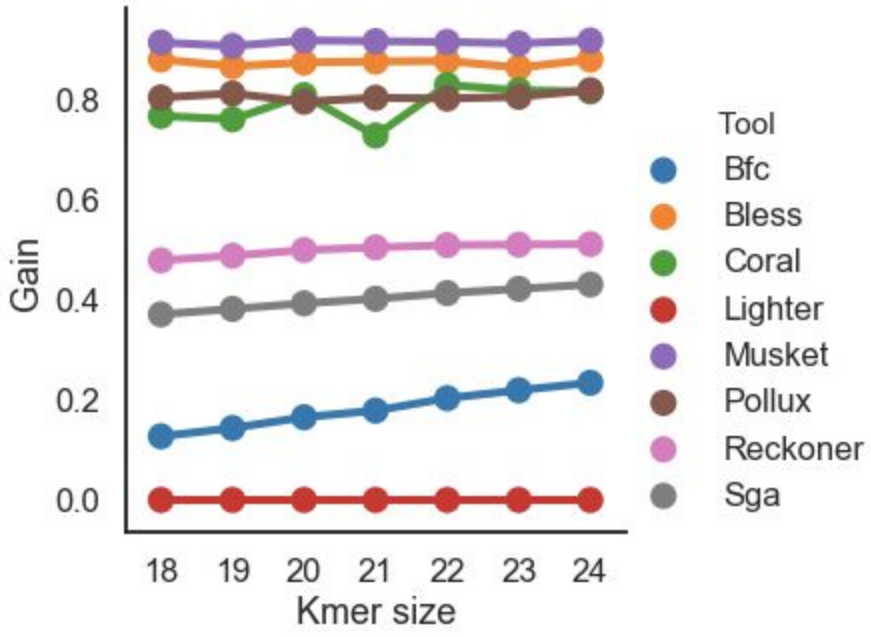
(a)



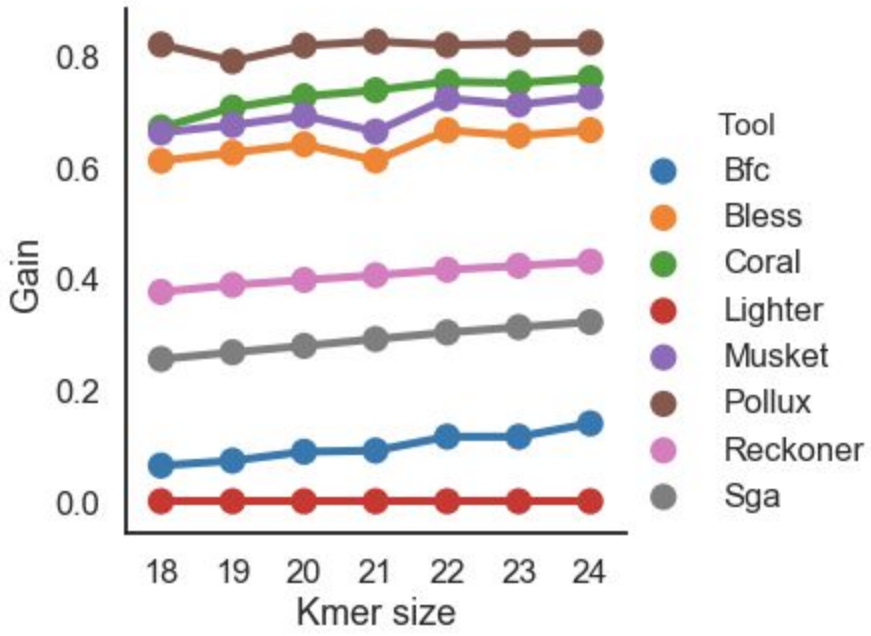
(b)


Gain
0.0
0.2
0.4
0.6
0.8
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

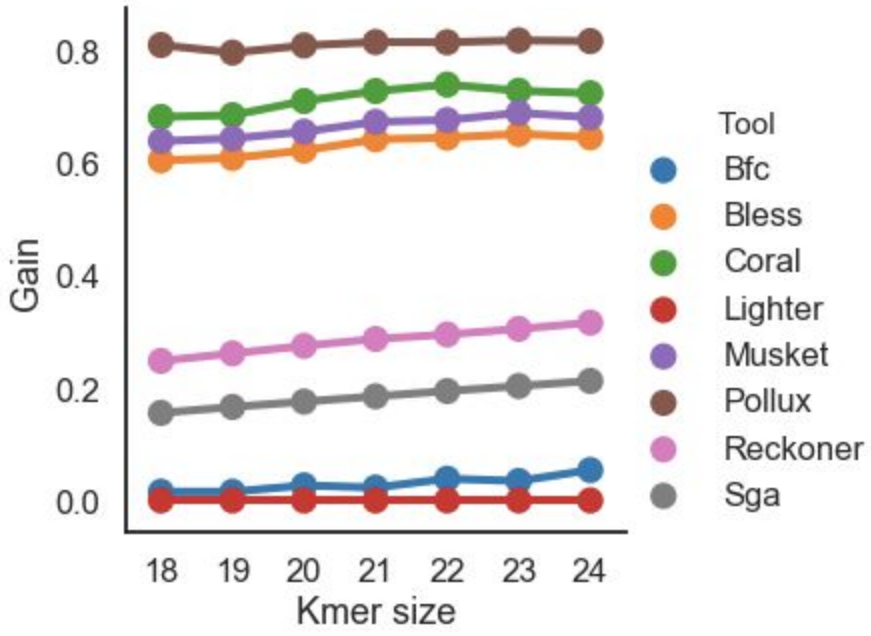
(c)


Gain
0.0
0.2
0.4
0.6
0.8
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(d)


0.8
0.6
0.4
0.2
0.0
Gain
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(e)


0.8
0.6
0.4
0.2
0.0
Gain
18
19
20
21
22
23
24
Kmer size
Tool
Bfc
Bless
Coral
Lighter
Musket
Pollux
Reckoner
Sga

(f)

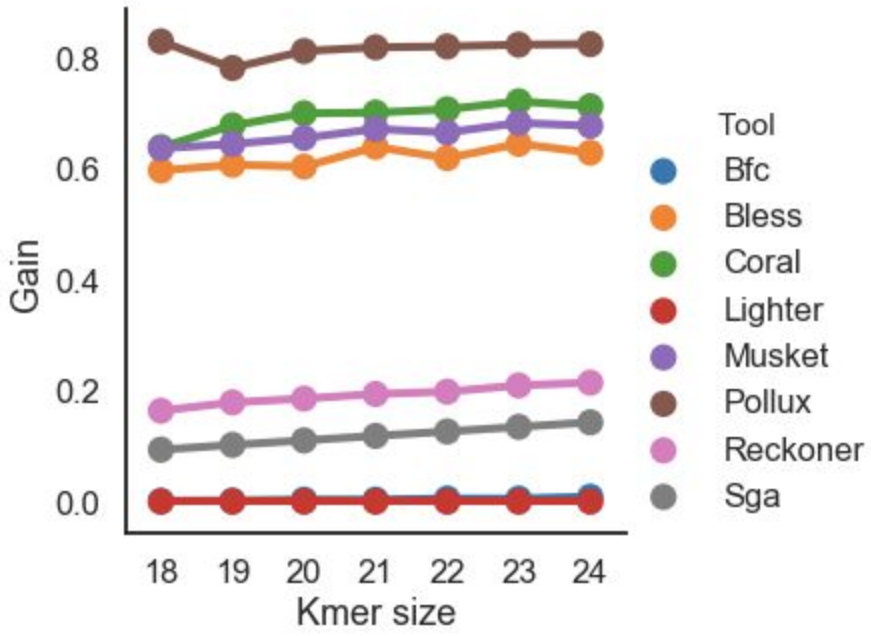


Figure S12. The effect of kmer size on the accuracy of the error correction tools across various coverages settings for TCR-Seq data. The reads were simulated from TCRA transcripts. (a) Coverage of 1x (b) Coverage of 2x (c) Coverage of 4x (d) Coverage of 8x (e) Coverage of 16x (f) Coverage of 32x

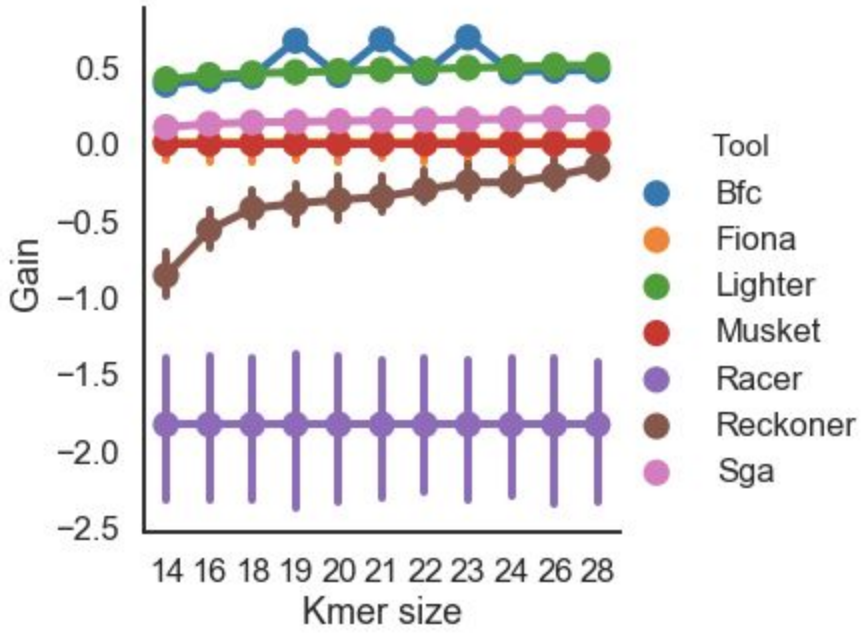


Figure S13. The effect of kmer size on the accuracy of the error correction tools across various coverages settings for TCR-Seq data. TCR-Seq reads were obtained from HIV patients. Error-free reads for gold standard were obtained using UMI-based protocol.

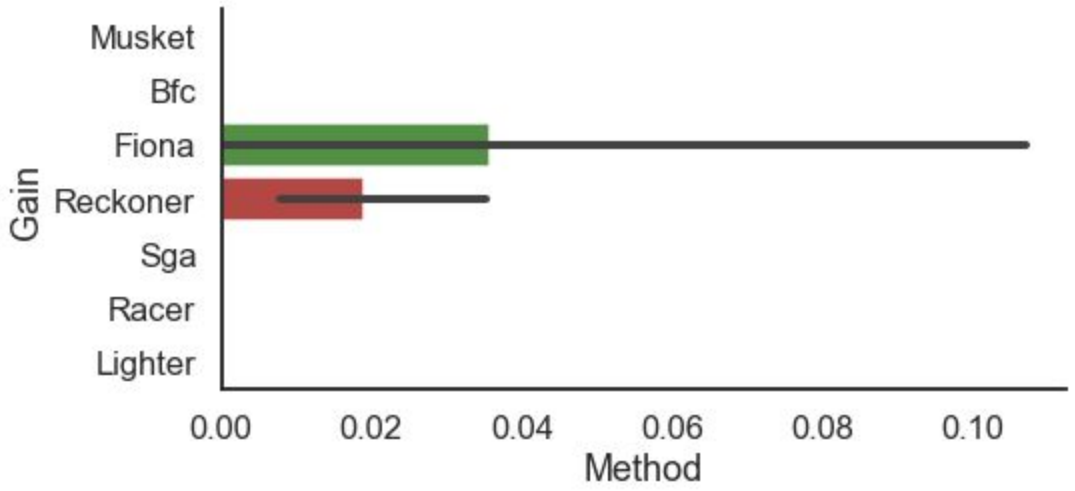


Figure S14. The efficiency of trimming across various coverages settings for TCR-Seq data.

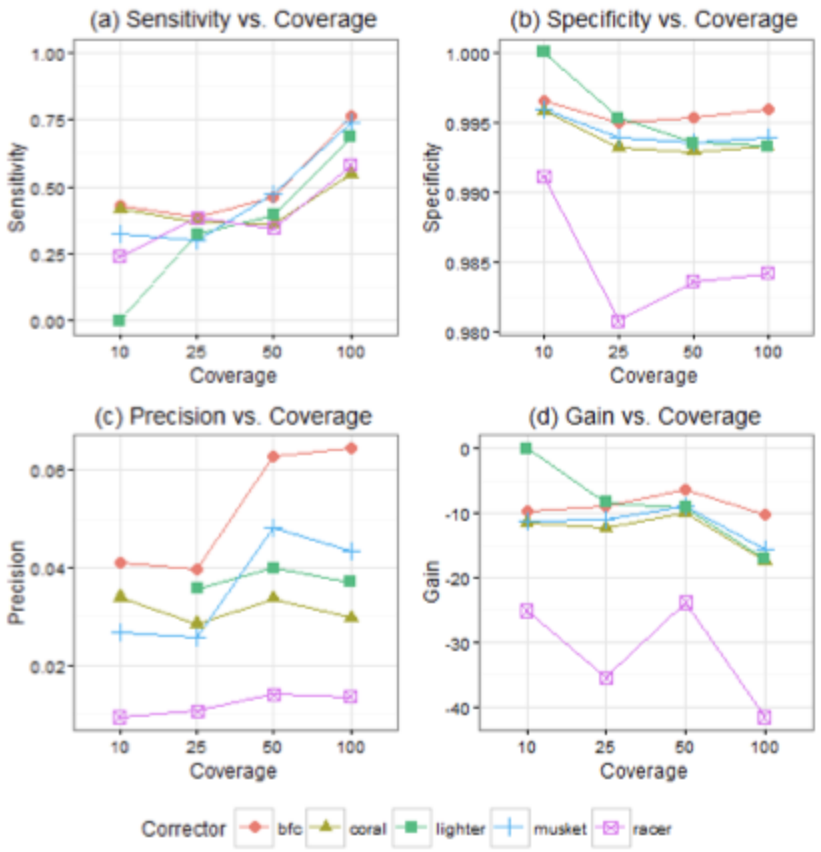


Figure S15. Error correction method evaluation based on viral sequencing data. Error-free reads for gold standard were obtained using UMI-based protocol. Datasets with various coverage were obtained by downsampling the original data.

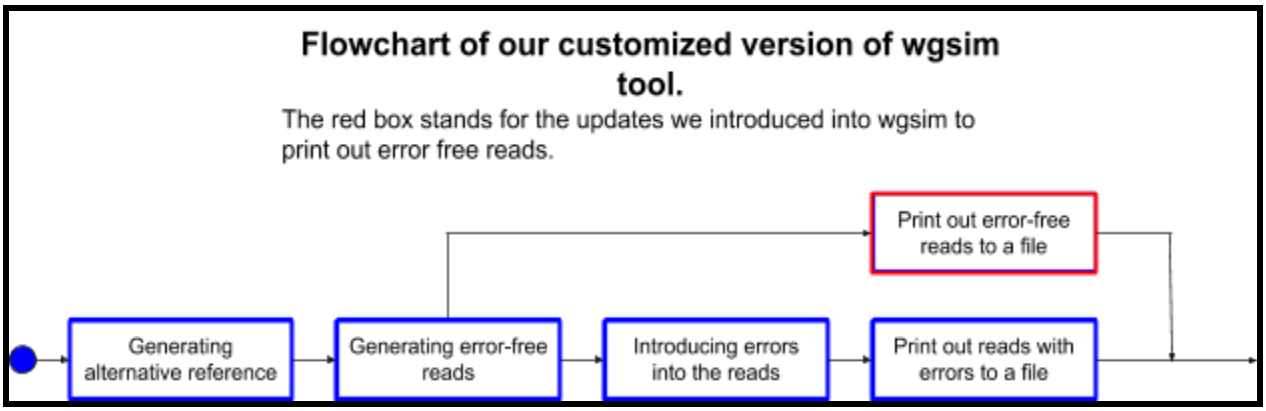


Figure S16. Flowchart displaying the alterations made to wgsim. Along with producing error prone reads using the tools, error-free reads are additionally produced in order to have.

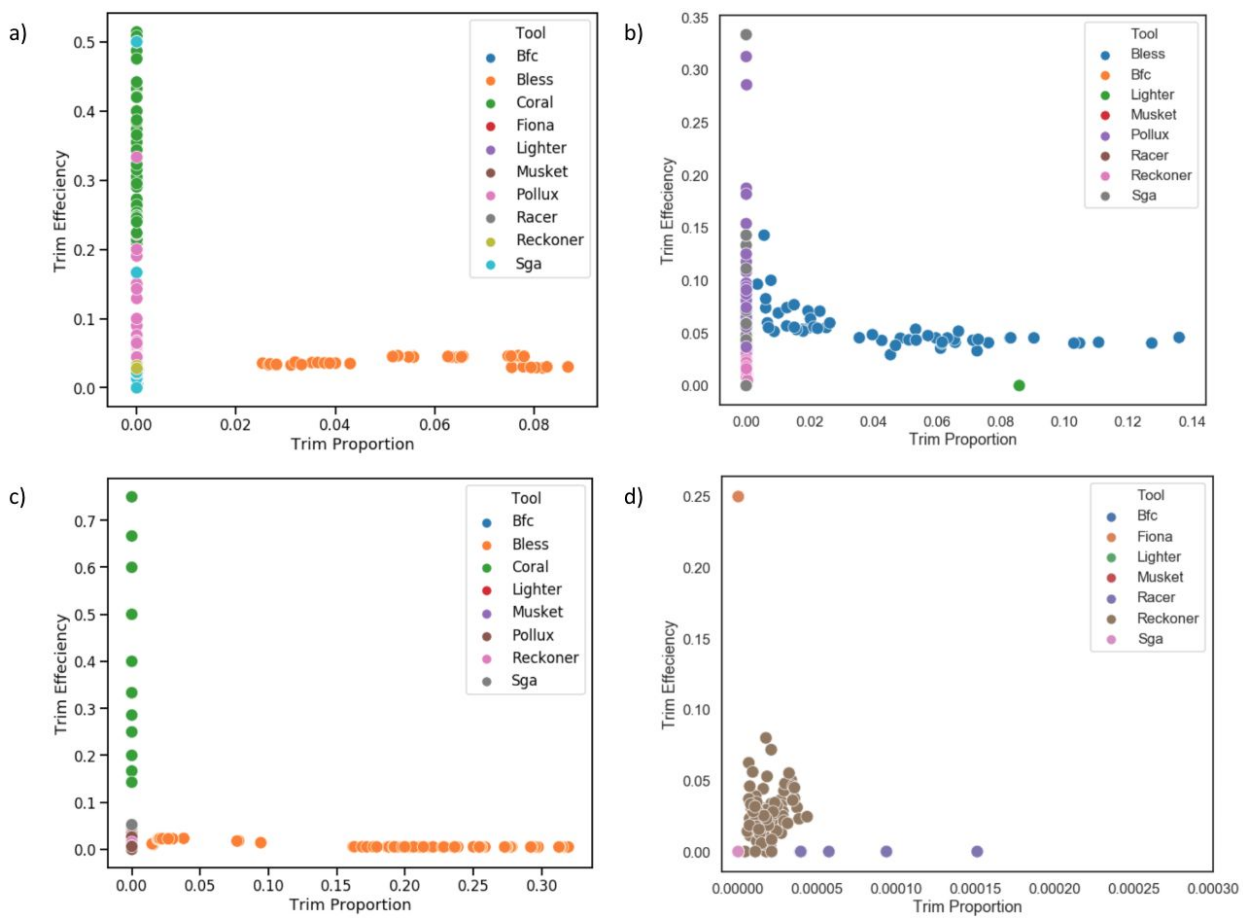


Figure S17. Trimming Efficiency vs. Trim Proportion **a)** BCR-Seq **b)** TCR-Seq **c)** WGS **d)** TCR-Seq real