Faculty of Engineering and Information Technology
University of Technology Sydney

Error Correction Algorithms for Genomic Sequencing Data

Thesis submitted in fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Xuan Zhang

September 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Xuan ZHANG, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE OF CANDIDATE: Production Note: Signature removed prior to publication.

[Xuan Zhang]

DATE: 5th July, 2021

PLACE: Sydney, Australia

Acknowledgments

First of all, I would like to express my sincere thankfulness to my principal supervisor Prof. Jinyan Li for his continuous support, assistance and advice during my Ph.D. studies. I would like to thank other members of Prof. Jinyan's research team for their kind helps on my research and daily life, they are Dr. Hui Peng, Dr. Yi Zheng, Dr. Chaowang Lan, Dr. Yuansheng Liu, Dr. Zhixun Zhao, Dr. Xiaocai Zhang, Tao Tang, Tian Lan and Pengyao Ping. I would further like to thank my parents for their love and continued support. Thanks for all the memorized scenes they brought to my life.

I am also grateful to my co-supervisor Prof. Michael Blumenstein and Prof. Gyorgy Hutvagner, for their kindly helps and suggestions during my Ph.D. career.

I sincerely acknowledge the organizations that provide financial supports for this research, including the Australia Research Council and the University of Technology Sydney.

I would like to thank all the anonymous peer reviewers for their insightful comments and suggestions to significantly improve the quality of this work.

Lastly, I would like to express my deepest gratitude to my family members and friends, who were always encouraging me when I was in difficult times.

Xuan Zhang @ Sydney
July 2021

Contents

List of Figures

List of Tables

List of Publications

Journal Papers :

J-1: **Zhang Xuan**, Liu Yuansheng, Yu Zuguo, Blumenstein Michael, Hutvagner Gyorgy, & Li Jinyan. (2021) 'Instance-based error correction for short reads of disease-associated genes', *BMC Bioinformatics* 22.6, 1-18.

J-2: **Zhang Xuan**, Ping Pengyao, Hutvagner Gyorgy, Blumenstein Michael, & Li Jinyan. (2021) 'Aberration-corrected ultrafine analysis of miRNA reads at single-base resolution: a k-mer lattice approach', Accepted by *Nucleic Acids Research*. .

J-3: **Zhang Xuan**, Blumenstein Michael, & Li Jinyan. (2021) 'Small-RNA sequencing reads restoration through accurate error rectification' plan to submit to *Genome Biology*.

Conference Papers :

C-1: Zhang Xiaocai, **Zhang Xuan**, Verma Sunny, Liu Yuansheng, Blumenstein Michael. & Li Jinyan. (2019), Detection of anomalous traffic patterns and insight analysis from bus trajectory data, *in* 'Pacific Rim International Conference on Artificial Intelligence'. Springer, pp. 307-321.

C-2: Chen Hongjie, Wang Xun, **Zhang Xuan**, Zeng Xiangxiang, Song

Tao & Rodríguez-Patón Alfonso. (2018), LncRNA-disease association prediction based on neighborhood information aggregation in neural network. *in* 'IEEE International Conference on Bioinformatics and Biomedicine'. IEEE, pp. 175-178.

Abstract

The rapid development of high-throughput next-generation sequencing (NGS) platforms has produced massive sets of genomic reads under low costs for a wide range of biomedical applications (*e.g.*, *de novo* genome assembly, read alignment, resequencing, and Single-nucleotide polymorphism discovery). A serious concern over these datasets is that machine-made sequencing data suffers from lots of random errors (such as substitutions, insertions and deletions). To the best of our knowledge, all the existing methods suffer limitations. This work aims to rectify as many errors as possible by designing strategies adapted to specific cases. Three novel error correction algorithms are designed to providing high-quality sequencing data.

The first method is to use an instance-based strategy to correct errors, as described in Chapter 3. This novel instance-based error correction method is able to provide high quality reads for any given instance case and implemented as a tool named InsEC. It is designed to correct errors in reads related to instance cases (*e.g.*, a set of genes or a part of the genome sequence. The nature of data characteristics and fine-grand features are considered to gain better correction performance. In our method, the instance-based strategy makes it possible to make use of data traits only related to an instance, which guarantees that we can approach the ground truth of the instance case and then achieve better error correction performance. In the instance extraction step, all reads related to a given instance are extracted by using read mapping strategies. In the correction step, we take advantage of alignment processes and correct errors according

to the alignment. Besides, statistical models are used to avoid induced errors as well. Intensive experiments are conducted with other state-of-the-art methods on both simulated and real datasets. The results demonstrate the superiority of our method, which achieves the best error correction performance (*e.g.*, precision, recall and gain rate in average) and further assembly results (*e.g.*, N50, the length of contig and contig quality).

Chapter 4 develops the first method for miRNA read error correction. Existing error correction methods do not work for miRNA sequencing data attributed to miRNAs' length and per-read-coverage properties distinct from DNA or mRNA sequencing reads. Although the error rate can be low at 0.1%, precise rectification of these errors is critically important because isoform variation analysis at single-base resolution such as novel isomiR discovery, editing events understanding, differential expression analysis, or tissue-specific isoform identification is very sensitive to base positions and copy counts of the reads. We present a novel lattice structure combining kmers, (k-1)mers and (k+1)mers to address this problem. Moreover, the method is particularly effective for correcting indel errors. Extensive tests on datasets having known ground truth of errors demonstrate that the method is able to remove almost all of the errors, without introducing any new error, to improve the data quality from every-50-reads containing one error to every-1300-reads containing one error. Studies on wet-lab miRNA sequencing datasets show that the errors are often rectified at the 5' ends and the seed regions of the reads. Note that there are remarkable changes after the correction in miRNA isoform abundance, the volume of singleton reads, overall entropy, isomiR families, tissue-specific miRNAs, and rare-miRNA quantities.

Chapter 5 introduces a novel method for small RNA error correction which supports substitution, insertion and deletion error rectification. Compared with the miRNA error correction method, this method is more robust by supporting all kinds of small RNA sequencing (read length from 20 to 200 nucleotides). Furthermore, we improve the three-layer lattice structure

and combine it by reads with the same length, length plus one and length minus one, which dramatically increases the method's efficiency. Finally, we consider RNA's isoform and propose to do correction proportionally to make a fine correction. Specifically, in the correction phase, we do not correct all potential erroneous copies to the top one candidate; Instead, we divide corrections into top 3 candidates proportionally to remain all possible recovery. With this improvement, the method achieves high error correction performance, and its precision, recall and gain rate are superior to all other existing error correction methods. Extensive experiments on simulation and raw sequencing data prove our method's ability. Thus, our error correction method does help improve data quality and is necessary for all downstream analyses.

Chapter 1

Introduction

This chapter describes the background, research motivations, research objectives, research contributions and structure of the thesis. In Section 1.1, the backgrounds of genomic sequencing data, error correction strategies as well as some significant applications are presented. Section 1.2 introduces the motivations in this research work. The corresponding research objectives and contributions of each motivation are specified in Section 1.3. Finally, the structure of this thesis is detailed in Section 1.4.

1.1 Background

Rapid development of highthroughput next-generation sequencing (NGS) platforms has produced massive sets of genomic reads under low costs for a wide range of biomedical applications (Salzberg, Phillippy, Zimin, Puiu, Magoc, Koren, Treangen, Schatz, Delcher, Roberts et al. 2012, Frazer 2012, Beerenwinkel & Zagordi 2011, Schirmer, Sloan & Quince 2012, Slatko, Gardner & Ausubel 2018, Chandran 2018). Accessibility of NGS data attracts amount of researches to explore secrets behind the genome, including *de novo* genome assembly (Salzberg et al. 2012), identifying functional elements in genomes (Frazer 2012), finding variations in populations (Beerenwinkel & Zagordi 2011) and genomics analysis (Schirmer et al. 2012).

A serious concern over these data sets is that there are lots of random errors (such as substitutions, insertions and deletions) existing in the sequences of these reads. To avoid possible negative effects on the downstream analysis caused by the sequencing errors, correction algorithms have been intensively studied and have become available to rectify the errors before the raw data is utilized for downstream analysis. Therefore, this thesis aims to propose high-efficient error correction methods to provide precise and convincing sequencing data for further genomic research.

1.1.1 Genomic Sequencing Data

Next-generation sequencing technology has seen significant improvements in data characteristics and costs. The length of sequence data, the number of data and the way data generated distinctly vary from the first-generation sequencing technology to the third-generation sequencing technology (Metzker 2010). Since the Sanger DNA sequencing technology (the first-generation sequencing technology) developed in 1997, the third-generation sequencing technology increases the length of reads from less then 100 nucleotide bases to more than 100,000 nucleotide bases, while the second-generation sequencing technology dramatically changes the number of data to millions in a single machine run. Comparisons of the first, second and third generation technology are shown in Table 1.1.1.

With the development of sequencing technology, more NGS sequencers have been launched into market. For examples, Illumina, Oxford Nanopores, PacBio and Roche sequencing platforms are widely used (Kamps, Brandão, Bosch, Paulussen, Xanthoulea, Blok & Romano 2017). Different platforms generate read with different length. According to the length of sequencing read, we divided sequencers into two catagories (short-read sequencers and long-read sequencers). Illumina, as representive of short-read sequencers, generates read usually shorter than 300bp with cheap price. Ion Torrent is another example of short-read sequencers. While costly long-read sequencers consist of PacBio and Roche, which use single-molecule real-time (SMRT)

technology and produce reads with longer length (e.g. 2.5 kb). Long-read sequencers go with a relatively higher error rate compared with short-read sequencers. Comparisons of the first, second and third generation technology are shown in Table 1.1.1. Among the existing sequencing platforms, the most ubiquitous Illumina platforms have advantages that high-throughput and low-cost. Thus, the error correction methods in this thesis are designed for short-read generated by Illumina platforms.

High-throughput sequencers generate a number of sequences per run. A sequence is also called a read that consists of nucleotides Adenine, Cytosine, Guanine and Thymine denoted by alphabet A, C, G, and T, respectively. Unfortunately, reads from the sequencing machine contain errors that may be produced by several factors. Errors can be induced in every phrase such as library preparation, sample preparation, genome content collection, experimental design, and sequencing machine running (Yamamoto 2021). There are three main types of errors from sequence data: substitutions errors (subs), insertions, and deletions errors (indels). Note that error rates of subs and indels vary from different sequencing platforms. For example, the Illumina, a widely used platform, generates high-throughput data with a relatively short read length from 100bp to 300bp and up to 2% error rates (Minoche, Dohm & Himmelbauer 2011). Most of the errors from the Illumina sequencer are subs, while from the Pacific Biosciences sequencer primary error type is indels. Moreover, longer reads from Pacific Biosciences sequencer have a higher error rate of more than 15% (Hackl, Hedrich, Schultz & Förster 2014). Poor quality of reads limits the performance of the downstream analysis. Therefore, high-quality sequencing reads with less errors are required and error correction is designed to improve the quality of the sequence data.

1.1.2 Error Correction Strategy

To avoid possible negative effects on the downstream analysis caused by the sequencing errors, correction algorithms have been intensively studied and

Sequencing Technology	Platform Name	Max length per read	Run-time	Output(Gb) per run
The first-generation	**Sanger**	500 - 750 bp	-	-
The second-generation	**Illumina**-HiSeq	2*150 bp	10h-11d	15-500
	Illumina-MiSeq	2*300 bp	5h-3d	0.3-13
	Illumina-NextSeq	2*150 bp	11-30h	19-120
	Ion Torrent-S5	400 bp	2.4-4h	2-16
	Ion Torrent-PGM	400 bp	3-7h	0.09-1.9
	Ion Torrent-Proton	400 bp	4-6h	12-88
	Roche-GS-FLX	400bp	10h	0.5-1
The third-generation	**PacBio**-RSII	3,000 bp	2h	0.09
	PacBio-Sequel	20,000 bp	0.5h-6h	0.08-1.25
	Oxford Nanopore-MinION	10,000 bp	1min-48h	44

Table 1.1: Comparison with different sequencing techniques

have become available to rectify the errors before the raw data is utilized for downstream analysis. The existing error correction methods can be divided into three categories according to their correction strategies.

- **Kmer based error correction strategy**

  Kmer is defined as a sequence string that is composed of k continue bases from a read. For a read, the number of kmer differs from the value of k. For example, there is string ATCGAT, and k is set as 4. Then 4-mers here is ATCG, TCGA, and CGAT. To collected all kmer from reads dataset, a set of sequence string would be obtained, which is defined as a kmer spectrum. Specifically, if we only have one read (TAGCTA) in a dataset, the 4-mer spectrum is a string set including three different kmer (TAGC, AGCT and GCTA). The key idea of these methods is to use the frequencies of all kmer strings and a global frequency threshold to define solid and weak kmers. The error correction process is to transform each weak kmer into a solid kmer according to some heuristics (e.g., the minimum edit distance between a weak and a sold kmer), thereby correcting erroneous reads.

- **Multialignment-based error correction strategy**

  Compared to the kmer spectrum methods, the idea of multiple alignment methods is more intuitive. Firstly, reads are grouped based on whether they share some kmers. Reads in each group are then concatenated to form a long consensus contig, which is assumed error free. Then, these consensuses are used as references to correct the mismatches in every read. More specificly, multiple alignment-based methods have three steps: read indexing, multiple alignment, and read correction. In read indexing step, all kmers from sequencing reads are indexed through construct a hash table. And then, for each read, kmer sets include reads which share at least one kmer with the given read. Then the Needleman-Wunsch algorithm (Needleman & Wunsch 1970) is used for multiple alignments in each kmer set.

After the alignment, every base in reads obtain a quality score, and nucleotides with the minimum score are selected as the consensus bases. Meanwhile each kmer sets obtain a consensus sequence for the following correction. Finally, the consensus is regarded as a reference to guide read correction.

1.1.3 Applications of Sequencing Data

Thanks to the rapid development of Next-Generation Sequencing (NGS) technology, we are able to collect massive read data with lower cost. Accessibility of NGS data attracts amount of researches to explore secrets behind genome and all of the researches are demanding high-quality reads with less errors, including *de novo* genome assembly (Salzberg et al. 2012), identifying functional elements in genomes (Frazer 2012), finding variations in populations (Beerenwinkel & Zagordi 2011) and genomics analysis (Schirmer et al. 2012)). Among these applications, the most fundamental and important one is read assembly and read alignment.

The most famous read assembly project is the human genomic project, which approach the whole order of human genomic sequencing from raw sequencing data, also called *de novo* assembly. Another important application is read alignment research is used widely in genomic analysis. Read alignment is the fundational of read comparation, which plays important roles in genetic disease research. Studies of genes related to the diseases are used in many domains (e.g. SNP calling (Li, Li, Fang, Yang, Wang, Kristiansen & Wang 2009), genotyping (DePristo, Banks, Poplin, Garimella, Maguire, Hartl, Philippakis, Del Angel, Rivas, Hanna et al. 2011), cancer treatment (Cainap, Balacescu, Cainap & Pop 2021) and taxonomic assignation). For instance, in breast cancer studies (Findlay, Daza, Martin, Zhang, Leith, Gasperini, Janizek, Huang, Starita & Shendure 2018), tumour suppressor gene, BRCA1, attracts lots of interests. Single-nucleotide variants (SNVs) in exons are the encode functionally critical domains of BRCA1. Detecting precise SNVs is fundamental of discovering significance of variants

as well clinically actionable genes (Millot, Carvalho, Caputo, Vreeswijk, Brown, Webb, Rouleau, Neuhausen, Hansen, Galli et al. 2012). Sequencing read data are used in SNVs detection and SNPs discovery (Chopra, Burow, Farmer, Mudge, Simpson, Wilkins, Baring, Puppala, Chamberlin & Burow 2015) as well. Except that, in mutation and protein study, sequencing data plays important role.

1.2 Research Motivations

The rapidly increasing number of genomic reads generated by the whole genome sequencing (WGS) platforms contain random errors (such as substitutions, insertions and deletions). Error correction algorithms have been developed, aiming to tick off these widely distributed errors to ensure the downstream analysis with quality enhanced raw data. Almost all WGS analysis will benefits from high quality sequencing reads. All of these applications need high-quality read data to more convincing conclusion. So far, an increasing number of researches already have been working on error correction on read data. However there are still some challenge to overcome. Firstly, due to the nature of the existing sequencing technology, it generates sequencing data non-uniformly. Sequencing coverage varies a lot from different parts of genome. For low-coverage region, there is no enough knowledge to correct read errors. Secondly, reads are generated randomly and then errors are also distributed non-uniformly. There are many factors (e.g. experiment environment, temperature and CG content) affecting appearance of errors in reads. Error ratio varies from a part of genome sequence to another parts as well. Some reads may suffer high error rates, which make it harder to detect errors in these kind of reads. Thirdly, repetitive regions exist in genome sequences. Reads from repetitive regions are likely to share the same nucleotide sequence, thus tending to be similar with each other. The similarity of reads from different repetitive regions causes difficulties in error correction. These kind of reads tend to be corrected falsely and cause

more induced errors. All of the above challenge make it still hard to provide high-quality reads when we do error correction on whole sequencing data.

Thus, we aim to develp novel methods to improve correction performance for specific application scenarios. For example, most of them take a global approach and only make use of generic genome-wide patterns to rectify errors in read. In some particular cases, global correction is unnecessary sometimes, especially, when single nucleotide polymorphism (SNP) studies are focus on specific disease genes or pathways, where the paramount requirement is to ensure the relevant reads, instead of the whole genome, are error-free.

Meanwhile, most of error correction methods are designed for DNA sequences and a few designed for RNA sequences. Recently, small RNA research gets more and more attention and the accuracy of small RNA sequences is vital for related analysis. For instance, sequencing of miRNAs (a special type of small RNA molecules containing about 22 nucleotide bases) has been widely used to examine tissue-specific expression patterns, to identify isomiRs (mature miRNA variants) and to discover previously uncharacterized miRNAs (Yeung, Co, Tsuruga, Yeung, Kwan, Leung, Li, Lu, Kwan, Wong et al. 2016, Xiao & MacRae 2019, Giraldez, Spengler, Etheridge, Godoy, Barczak, Srinivasan, De Hoff, Tanriverdi, Courtright, Lu et al. 2018, Tan, Chan, Molnar, Sarkar, Alexieva, Isa, Robinson, Zhang, Ellis, Langford et al. 2014, Trontti, Väänänen, Sipilä, Greco & Hovatta 2018, Fernandez-Valverde, Taft & Mattick 2010). As key regulators in various biological processes, miRNA dysregulation is implicated in many diseases for example cancer and autoimmune disorders (Meng, Liu, Lü, Zhao, Deng, Wang, Qiao, Zhang, Zhen, Lu et al. 2017, Liu, Lei, He, Pan, Xiao & Tao 2018, Telonis, Magee, Loher, Chervoneva, Londin & Rigoutsos 2017, Dutta, Chinnapaiyan & Unwalla 2019, Dai, Li, Fan, Tan, Wang & Jin 2019). Numerous studies also reaffirm that miRNA regulatory functions are involved in post-transcriptional gene silencing (PTGS), transcriptional gene silencing (TGS), and transcriptional gene activation (TGA) (Pisignano, Napoli, Magistri, Mapelli, Pastori, Di Marco, Civenni, Albino, Enriquez,

Allegrini et al. 2017, Yang, Shao, Bofill-De Ros, Lian, Villanueva, Dai & Gu 2020), in which miRNAs bind to nascent RNA transcripts, gene promoter regions or enhancer regions and exert further effects via epigenetic pathways (Liu et al. 2018, Liu, Wang, Huang, Sun & Chen 2019). Thus, single-base errors are very sensitive for uncovering miRNA'isoforms (isomiRs) and alternative splicing, which is one of the most frontier research areas in this field (Liao, Li, Wang, Li & Zou 2018, Tan et al. 2014, Liu, Lai & Guo 2020, Bilanges, Posor & Vanhaesebroeck 2019, Sänger, Bender, Rostowski, Golbik, Lilie, Schmidt, Behrens & Friedrich 2020, Pillman, Goodall, Bracken & Gantier 2019, Hoefer 2020). Specific error correction methods designed for this kind of small RNA are required.

1.3 Research Objectives and Contributions

To address above research motivations, this thesis focuses on 3 research problems: 1) error correction for instance cases (e.g. reads of disease-associated genes), 2) error correction for microRNA sequencing reads, and 3) error correction for total small RNA sequencing reads. The **research objective** of this thesis is to contribute to high-quality sequencing data by approving novel error correction approaches that could generate accurate and reliable sequencing data for convincing downstream analysis . The specific objectives of above research problems are summarized as follows (O1 to O3).

O1: To develop a novel instance-based error correction method for short reads of disease-associated genes, to provide accurate sequencing reads for fine downstream analysis.

O2: To develop the first error correction method designed for microRNA reads through considering microRNA's characteristics and to provide single-base resolution for ultrafine microRNA research.

O3: To develop the first error correction method for small RNA sequencing read supporting both subs (substitution errors) and indels (insertiong

and deletion errors) corretion.

To complete these objectives, we have proposed 3 novel methods as presented in Chapter 3, Chapter 4 and Chapter 5, respectively. Our **contributions** are elaborated as follows (C1 to C3).

C1: Error correction for short reads using a novel instance-based method

The contributions include :

1) We proposed a novel instance-based approach for highly accurate error correction, which focuses on short reads especially related to disease genes.

2) Our strategies exploit local sequence features and statistics directly related to those genes, thereby being capable of high-performance correction related to any given instance.

3) We conducted extensive experiments in comparison with state-of-the-art methods on both simulated and real datasets of lung cancer associated genes (including single-end and paired-end reads). The results demonstrated the superiority of our method with the best performance on precision, recall and gain rate, as well as on sequence assembly results (e.g., N50, the length of contig and contig quality).

C2: Error correction method designed for microRNA sequencing reads

Our contributions in this research are summarized :

1) We presented an error rectification method for miRNA sequencing reads (named miREC), which is the first tool to address the problem of miRNA sequencing errors.

2) The novel strategy of our method is the use of a 3-layer (k-1)mer-kmer-(k+1)mer lattice structure to maintain the frequency differences of the kmers. These superset-subset frequency differences are very effective to detect the errors especially the indel errors. The lattice structure is also a moving structure where k is set continuously from a small number

to a big number 23 or 25 for a full coverage of error correction. 3) Results on datasets having known ground truth of errors demonstrated that the method is able to remove almost all of the errors, without introducing any new error, to improve the data quality from every-50-reads containing one error to every-1300-reads containing one error.
4) Extensive tests on both simulated and wet-lab (experimentally catalogued) miRNA sequencing datasets showed that miREC can excel performance in all of precision, recall and gain.

C3: Error correction method for small RNA supporting substitutions and indels correction
The contributions in this research include :
1) We proposed the first algorithm to solve the problem of correcting errors in all types of small RNA sequencing reads, supporting substitutions and indels correction. The novel idea of the algorithm is a 3-layer read lattice structure and proportional correction strategy.
2)The algorithm achieved outstanding and robust correction performance; It detected and corrected almost all of the errors including the indel errors; More specifically, the average recall rate is 99.86%; the average precision is 99.9% and the average gain rate is 99.81%.
3) Identified significant changes in small RNA abundance and whole set entropy after error correction on wet-lab small RNA sequencing reads.

1.4 Thesis Structure

The structure of this thesis is illustrated in Figure 1.1 and briefly introduced as follows:

Chapter 1 introduces the background of this thesis, the research motivations and the corresponding research objectives as well as contributions. **Chapter 2** presents the related work of this research,

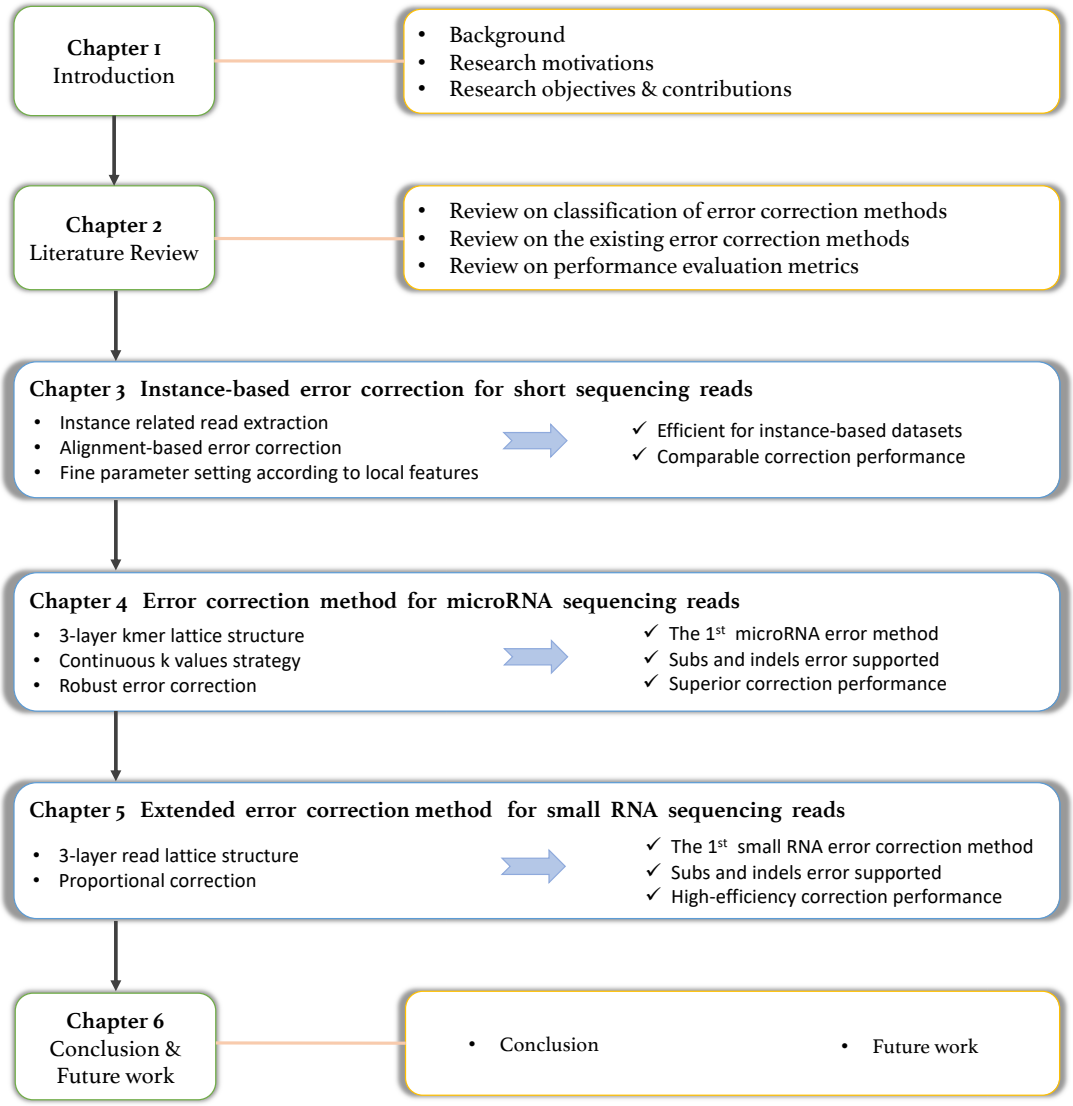

Figure 1.1: Thesis structure.

including the classification of the existing error correction methods, related evaluation metrics and applications. **Chapter 3** to **Chapter 5** detail the proposed methods for instance-based error correction, miRNA and small RNA error correction, respectively. Details of experimental evaluation, comparison and analysis are also included. **Chapter 6** concludes this thesis and provides discussions of future work.

Chapter 2

Related Work and Literature Review

This chapter describes the related work and literature review of the work in this thesis. Section 2.1 reviews the relevant work on sequencing read error correction and its classification. Then, the state-of-the-art methods for error corretion methods are presented in Section 2.2. Following this, the error correction evaluation metrics are introduced in Section 2.3. Finally, we briefly summarize the contents in this chapter.

2.1 Error Correction Classification

According to the development of sequencing technology, we catalogue error correction methods based on data characteristics. Also we can catalogue methods according to their error correction strategies. A summarized table is shown in Table 2.2.3.

2.1.1 Classification According to Data characteristics

The sequencing technology innovated two times from the Sanger sequencing technology (the first-generation sequencing), the high-throughput sequencing technology (the second-generation sequencing) and the third sequencing

technology. Thanks to the development of sequencing technology, amount of genome sequence data is easily to collected. All these sequencing data contains errors and error correction methods are needed for providing high-quality sequencing data. Different sequencing technology generates reads with different length. Specificly, the second-generation sequencing technology (High-throughput sequencing technology) generates short sequencing reads, while the third-generation sequencing technology generates long sequenging reads.

Thus, according to lenght of read, we can divided error correction methods into two catalogues (error correction methods for short reads and error correction methods for long reads). Short-read error correction methods includes three types according to its strategies and more details are shown in section 2.1.2. Long-read error correction methods are two types. One is self correction method and the other is hybird error correction method. Self correction means only long reads are used when correction, while hybird correction means short reads and long reads are used together for error correction.

2.1.2 Classification According to Correction Strategies

According to the correction strategies, the existing error correction methods are divided into three catagories. One is the kmer based error correction methods including BFC (Li 2015), BLESS (Heo, Wu, Chen, Ma & Hwu 2014), Lighter (Song, Florea & Langmead 2014), Blue (Greenfield, Duesing, Papanicolaou & Bauer 2014), ACE (Sheikhizadeh & de Ridder 2015), Reptile (Yang, Dorman & Aluru 2010), Musket (Liu, Schröder & Schmidt 2012), RACER (Ilie & Molnar 2013). The other is the multi-alignment based error correction methods including Coral (Salmela & Schröder 2011), ECHO (Kao, Chan & Song 2011), MEC (Zhao, Chen, Li, Jiang, Wong & Li 2017) and Karect (Allam, Kalnis & Solovyev 2015). Last is graph-based correction method.

2.2 The Existing Error Correction Methods

In this section, we will introduce three types of the existing error correction methods according to their strategies.

2.2.1 The Kmer Based Methods

In kmer based methods, kmer is defined as a string composed of k continue bases from a read. A representative kmer-based method is BFC (Li 2015). Firstly, methods extract kmers from read datasets and then build index information for each kmer. Secondly, a threshold is set to divide all kmers into two sets, according to prior knowledge (like the frequency of kmers). kmer in a set whose value is lower than the threshold are defined as weak kmer; while kmer in the other set whose value is higher than the threshold are recognized as solid kmer. Finally each weak kmer will be transformed to solid kmers by referring to the solid one with pre-designed rules (e.g., the minimum edit distance between a weak and a sold kmer).

For example, Reptile (Yang et al. 2010), proposed in 2010, use Hamming distance to infer that whether a kmer contains errors. Hamming distance is the number of difference between two strings. When rectifing errors Reptile also take contextual information from neighboring kmers into account. Only if hamming distance of kmers are less than a fixed threshold, they are identified as weak kmers and needed to be corrected. Reptile is implemented in c plus plus program language. Another method named Musket (Liu et al. 2012) use three strategies to rectify errors. Firstly, all kmers are collected from read data and then the number of each kmer is counted by using a hash structure with a master-slave model. Secondly, the voting strategies are used to choose true base information and weak kmers are transformed into solid kmers. Compared with the above methods, RACER (Ilie & Molnar 2013) focuse more on methods' efficiency (such as faster speed and less memory). Every read are encoded by 2-bit way storing with a hash table structure to save memory and increase speed of algorithm. From

both end of reads, eight counters are used to count the number of different nucleotides. There is a threshold fixing to decide whether a base is a correct or not. According to pre-results, Racer corrects errors in reads. Lighter (Song et al. 2014) is different from previous methods which need to count kmers. In Lighter, counting kmers process are replaced by randomly sampling kmers. After that, samples are stored in a bloom filter for subsequent tests. All positions of reads are tested to find a subset of kmers which are stored in the second bloom filter. Compared with the solid kmers, Lighter turns weak kmers into solid kmer and the error-free reads are finally obtained. Searching for an efficient solution, Lim et al. propose the Trowel method based on a data structure which can access easily and support massively parallelization (Lim, Müller, Hagmann, Henz, Kim & Weigel 2014). Different from some kmer spectrum methods which use kmer's frequencies to identify solid parts and weak parts, Trowl detects solid kmer only by the quality value of bases. The selected kmers are called bricks which indexes as keys. BLESS (Heo et al. 2014) corrects reads error by counting frequencies of each kmer. It applies DSK(Disk Streaming of kmer) and KMC(kmer counter) to do error correction. Through a Bloom filter, weak kmer are converted to solid kmer. A threshold should be set to recognized weak kmer from the solid one. That means the value of the threshold can affect the result of error correction.

In kmer based methods, the value of K plays an important role for error correction performance, which limits the robustness of the methods. So how to set k value and how to design rules for dividing solid and weak kmers are vital for kmer based error correction methods (Akogwu, Wang, Zhang & Gong 2016).

2.2.2 The Alignment Based Methods

Compared to kmer based methods, multiple alignment methods do not rely too much on the selection of k-value. The idea behind multiple alignment methods is more intuitional. Firstly, reads are grouped based on whether they share the same segment of reads. The aim of this step is to obtain

a longer read by concatenating reads which share overlap parts. Through concatenating overlap parts, a convincing longer read, called consensus, is obtained as error-free reference. Finally, these consensuses are used as reference sequences to correct remnant reads.

The multiple sequence alignment strategy is first used to correct reads errors in Coral (Salmela & Schröder 2011). Unlike the existing EC (Error Correction) tools, the Coral computes bases distant for errors by alignment. There are three steps in Coral methods: indexing the reads, forming multiple alignments and correction of reads. In the first step, kmers from all reads are indexed through construct a hash table. In the second step, for each read, kmer sets defined as reads which share at least one kmer with the given read. Then the Needleman-Wunsch algorithm (Needleman & Wunsch 1970) are used for multiple alignments in each kmer sets. After alignment, every base in reads has a quality score and nucleotides with the minimum score are selected as the consensus bases. Finishing the alignment step, each kmer sets obtain a consensus sequence for the following correction. In the final step, the consensus is regarded as a reference to guide read correction. Coral is able to handle a wide range of sequencing data and its performance is superior to previous error correction methods.

The Karect (Allam et al. 2015) is also based on multi-alignment idea, increasing performance of the error correction a lot. Karect take advantage of heuristic strategy to extract a set of reads. multiple sequence alignment is conducted on every read set and the results of multiple sequence alignment are stored in a graph structure, known as POG (partial order graph). According to the graph structure, the erroneous bases are easy to be identified. Besides that the insertion and deletion errors also can be rectified by Karect. Another MEC (Zhao et al. 2017) method uses the MapReduce technique to correct error correction, which is time-saving and high-efficiency. MEC selects reads sharing with the same kmers in the same group. And then it computes a log likelihood of every base in every read of each group. Accorfing to the likelihood score, MEC corrects errors in reads.

In multiple alignment methods, the influence of parameter decreases and k parameter is almost free from the consideration, but high time and memory consuming limits its application especially in large datasets.

2.2.3 The suffix array Based Methods

The suffix array/tree based methods are aimed to explore the beneficial nature of the suffix tree structures. This type of methods use reads data to generalize a suffix tree and then traverses the suffix tree structure. Finally, through analyzing nodes of the tree to detect errors in reads. Compared with multiple sequence alignment based methods, the suffix array based methods avoid the computation of the alignment process and only traverse a space-efficient tree structure to complete error correction.

SHREC (Schröder, Schröder, Puglisi, Sinha & Schmidt 2009) first uses a generalized suffix tree to store reads data and corrects erroneous reads by traversing the suffix tree structure. In the beginning, SHREC extracts all reads information to build the suffix tree. all suffix strings of each read are represented by a path and each leaves nodes in the suffix tree means termination of a suffix string. According to the number of edges followed by a node, weight of the node is computed for constructed a weighted suffix tree. Finally, SHREC selects an imbalanced node to locate the error and for further error correction.

Following the SHREC method, the author proposes Hybird-SHREC method based on SHREC method (Salmela 2010). By this methods, all kinds of error types can be identified and corrected. It also supports different kinds of reads data, including a mixed set of reads which is produced by several sequencing platforms. Compared with the SHREC method, the hybrid-SHREC show a better performance with high sensitivity and high specificity.

Ilie et.al proposes a High Throughput Error correction algorithm named HiTEX which uses a thorough statistical analysis of the suffix array built on the string of all reads and their reverse complemets (Ilie et al. 2010). To avoid the limations in SHREC methods that it requires several parameter

Method Catagory	Method Name	Error Type	Year	Referencen
The K-mer based method	Reptile	subs	2010	(Yang et al. 2010)
	Musket	subs	2012	(Liu et al. 2012)
	RACER	subs	2013	(Ilie & Molnar 2013)
	Lighter	subs	2014	(Song et al. 2014)
	Trowel	subs	2014	(Lim et al. 2014)
	BLESS	subs	2014	(Heo et al. 2014)
	BFC	subs	2015	(Li 2015)
The Alignment-based method	Coral	subs	2011	(Salmela & Schröder 2011)
	Karect	subs and indels	2015	(Allam et al. 2015)
	MEC	subs	2017	(Zhao et al. 2017)
The Suffix array-based method	SHREC	subs	2009	(Schröder et al. 2009)
	Hybird-SHREC	subs and indels	2010	(Salmela 2010)
	HiTEX	subs and indels	2010	(Ilie, Fazayeli & Ilie 2010)

Table 2.1: Comparison with the existing error correction methods

sets to find the optimizal value of the accuracy, HiTEC are more robust. Due to the main idea of HiTEC which is based on statistical analysis, it can be suitable for many situations no matter what the read length is or what the coverage level is.

2.3 Error Correction Evaluation metrics

To comprehensively evaluate our error correction methods, we not only focus on error correction performance but also examine performance of further assembly.

2.3.1 Statistical Error Correction Performance

To access the accuracy of error correction methods, we deduct the following three statistics. There are some pre-definitions as follows: true positives (TP) correspond to corrected errors; true negatives (TN) correspond to initially correct bases left untouched; false positives (FP) correspond to newly introduced errors; false negatives (FN) cor respond to unidentified errors.

- Precision: *TP/(TP+FP)*, shows the fraction of truly corrected bases among all changed bases.
- Recall: *TP/(TP+FN)*, shows the fraction of truly corrected bases among all bases which are supposed to be corrected.
- Gain: *(TP-FP)/(TP+FN)*, shows the fraction of removing errors without inducing additional errors.

2.3.2 Statistical Assembly Performance

To access the impact of error correction methods on further assembly, an assembler (e.g. SPAdes (Bankevich, Nurk, Antipov, Gurevich, Dvorkin, Kulikov, Lesin, Nikolenko, Pham, Prjibelski et al. 2012)) is used to assemble reads before and after error correction. And then the assemble results are

compared directly. Note that other assembly tools are also available to use and need to run without built-in error correction.

For simulated datasets, the ground truth read data are assembled by SPAdes as well for evaluation. The assembly results are evaluated by QUAST (Gurevich, Saveliev, Vyahhi & Tesler 2013), a quality assessent tools for genome assemblies. Detailed reports include the number of contigs, largest contigs, total length, CG percentage and N50. Contigs is continuous nucleotide sequences obtained from assembly process. N50 is defined as the minimum contig length needed to cover 50% of genome.

2.4 Summary

In this chapter, the literature is reviewed with respect to the research motivations of error correction methods. More precisely, relevant studies are divided into three types according to their strategies. Most of all the existing methods are described, followed by evaluation metrics.

Chapter 3

Instance-based Error Correction for Short Sequencing reads

3.1 Introduction

The rapid development of high-throughput next-generation sequencing (NGS) platforms has produced massive sets of genomic reads under low costs for a wide range of biomedical applications (Salzberg et al. 2012, Frazer 2012, Beerenwinkel & Zagordi 2011, Schirmer et al. 2012). Serious concern over these datasets is that there are lots of random errors (such as substitutions, insertions and deletions) existing in these reads. The most popular Illumina platforms generate sequencing data with 0.5-2.5% error rates (Laehnemann, Borkhardt & McHardy 2015). Substitutions are the major error type in the short sequencing reads, while insertions and deletions are the major error types in the long sequencing reads.

To avoid possible negative effects on the downstream analysis caused by the sequencing errors, correction algorithms have been previously studied and many tools (Limasset, Flot & Peterlongo 2020, Sheikhizadeh & de Ridder 2015, Song et al. 2014, Li 2015, Allam et al. 2015, Heo et al. 2014, Greenfield

et al. 2014, Salmela & Schröder 2011, Kao et al. 2011) have become available to rectify errors in the raw data. These methods take a global approach to rectify all possible errors using genome-wide patterns and statistics. Because the correction is operated on the whole set of reads (usually millions or billions in number), the algorithm complexity is high and the correction performance is not perfect; sometimes even a lot of new errors are introduced into the reads by these global approaches. These challenges are attributed to several reasons. Firstly, the sequencing depth is non-uniform — the sequencing coverage varies remarkably from one part to another in the genome. The resulting conflicts between the $k$mer statistics from the low-coverage regions and those from the high-coverage regions have significantly hindered the global approach to conduct effective error removal — Some genes may get under-corrected while some other genes get over-corrected. Secondly, genome fragmentation for read generation is random and the errors are distributed non-uniformly. Thirdly, repetitive regions exist in the genome sequences. Reads from the repetitive regions are likely to share the same nucleotide sequence, or highly similar to each other (Liu, Zhang, Zou & Zeng 2020). Errors in these reads tend to be corrected falsely by the global approaches and many new errors are introduced.

It is sometimes unnecessary to conduct global correction. Instead, highly-accurate instance-based error correction for short reads of specific genes is more important. For example when SNP (Li et al. 2009) or genotyping properties (DePristo et al. 2011) are of great importance, then only specific genes or pathways involved in the disease mechanism or a special segment of loci in the genome would be focused on. In these important situations, the paramount requirement is to ensure the relevant reads, instead of the whole genome, are error free after the correction step. As in a recent breast cancer study (Findlay et al. 2018), the tumour suppressor gene BRCA1 and particularly the single-nucleotide variants (SNVs) in this gene's exons are focused on understanding the functionally critical domains of BRCA1 and the related clinically actionable genes (Millot et al. 2012). It is vital to

provide error-free reads related to these specific genes (Chopra et al. 2015) for the precise detection of SNVs and accurate discovery of SNPs. As another example in the mutation and protein research area, error correction is important because one or two DNA base mutations in the coding region of a gene may lead to functionally different amino acids (Bashir, Ragab, Khabour, Khassawneh, Alfaqih & Momani 2018, Wang, Freedman, Liu, Moorman, Hyslop, George, Lee, Patierno & Wei 2017, Fung, Zhou, Joyce, Trent, Yuan, Grandis, Weissfeld, Romkes, Weeks & Egloff 2015), and more likely when the open reading frame mechanism is considered. These mutations are called *point* mutations, and more than 31,000 such mutations in the human genome are associated with genetic diseases (Ravindran 2019). The reads related to such a gene without error correction or with under-correction may mislead the conclusion about the functional properties of the proteins. The existing global error correction is not the best choice for this.

In this work, we propose to use an instance-based approach to make error correction for the reads of a disease-associated gene. The method is also applicable to the reads of multiple disease genes, or a set of genes related to a phenotype, or an unknown-function region in the genome, or even any nucleotide sequence of interests. The method, named InsEC, aims to rectify the errors in the instance reads with a very high accuracy and to reduce the number of introduced new errors to a minimum. The global approaches suffer from the issue of non-uniform sequencing depths occurred in error correction. However, when the instance-based approach is taken for the error correction in a subset of reads, this issue can be significantly moderated. Comparing with the global approaches which may have neglected the local features of the instance reads, our instance-based approach has the advantage that the patterns and statistics can be exhaustively explored to rectify the errors, and can be conservatively combined to reduce the number of introduced errors. InsEC has two steps. The first step is for read extraction, which collects all reads relevant to a given gene. The second step is for correction, which exploits the local sequence features in the extracted read sets It uses local

alignments to quantify erroneous probability of each base in the reads for an accurate correction.

In fact, global approaches can be turned into instance-based approaches if the whole set of reads is narrowed down to the subset of reads of a specific gene as input data. These global approaches include $k$mer based error correction methods such as BFC (Li 2015), BLESS (Heo et al. 2014), Lighter (Song et al. 2014), Blue (Greenfield et al. 2014), and ACE (Sheikhizadeh & de Ridder 2015). The key idea of these methods is to use the frequencies of all $k$mer strings and a global frequency threshold to define solid and weak $k$mers. The error correction process is to transform each weak $k$mer into a solid $k$mer according to some heuristics (e.g., the minimum edit distance between a weak and a sold kmer). Because the sequencing depths are non-uniform across the genome, some globally weak $k$mers are actually solid $k$mers in a local region. Thus it is a wrong correction to transform these local solid $k$mers. Compared with the global $k$mer based methods, the global multiple alignment methods, including Coral (Salmela & Schröder 2011), ECHO (Kao et al. 2011) and Karect (Allam et al. 2015), do not rely too much on the selection of $k$mers. Firstly, reads are grouped based on whether they share some $k$mers. Then reads in each group are concatenated to form a long consensus contig, which is assumed error-free. Then, these consensuses are used as references to correct the mismatches in every read. But, the $k$mer grouping can intensify the issue of non-uniform sequencing depths in the contigs, i.e., the error-free assumption on the contigs is too strong and biased.

Our instance-based approach InsEC does not need to define solid or weak $k$mers in the correction step, and thus it can avoid the issue of non-uniform sequencing depths in the global approaches. Although similarly as the multiple sequence alignment methods to implement the alignment process, our InsEC quantifies error probabilities conservatively column-by-column and row-by-row in the alignment array to avoid introducing new errors.

The performance of InsEC is evaluated on the error correction itself as well as on the quality of the resulted assemblies. Extensive experiments demonstrated that our method has superior precision, recall and gain rates over all state-of-the-art error correction methods when tested on reads datasets of lung cancer associated genes. The quality of the assemblies of the reads also become improved after our error correction. We obtained longer and less number of contigs, and the contigs are closer to the ground truth in the simulated datasets. In our SNP case studies, we found that some corrections can happen at the current lung cancer SNP database, implying that instance-based error correction is crucially important for SNP and mutation analysis.

3.2 Methods

A read r is a genomic sequence denoted by $r = r_1 r_2 \cdots r_n$, $r_i \in \Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{N}, \mathrm{T}\}$, where A, C, G and T stand for the nucleotides Adenine, Cytosine, Guanine and Thymine respectively, and the character N stands for uncertain nucleotide; and n is the length of r (e.g., $n = 100$ or 200). Usually, the length of all of the reads from one wet-lab experiment (short read sequencing) is exactly the same. The sequencing errors can be randomly distributed anywhere in r.

Computation required by InsEC consists of two main tasks. One task is to draw relevant reads to a given gene from a WGS sequencing dataset. Through read extraction, a gene-related read dataset is constructed for error correction. The second task is to precisely correct errors on the gene-related subset of reads using fine-grained alignment patterns and statistics.

3.2.1 Reads extraction

Let S be a set of human genomic reads generated by Illumina whole genome sequencing platforms, and let I_g be a reference sequence of our interested gene g. But the reference sequence I_g is assumed *not* error-free. We extract reads

from S which are relevant to the gene sequence I_g for the correction of possible errors in these reads. This subset of reads is denoted by $subset(S, I_g)$. We also assume that the ground truth of gene sequence can vary from different individual samples because of single-nucleotide polymorphism. So the ground truth of gene g, denoted by T_g, should have different nucleotide bases with the reference gene sequence I_g. Under the above two assumptions, reads having a Hamming distance with I_g (i.e., with noise tolerance) are required to move from S to form $subset(S, I_g)$. The Hamming distance threshold is set as 95 so as to have complete relevance of $subset(S, I_g)$ to T_g as much as possible. In this work, we use BWA-MEM (Li 2013) for the read mapping with Hamming distance tolerance. BWA-MEM is a widely-used alignment tool, highly efficient to align short reads against a nucleotide sequence, and it allows mismatches and gaps, which means the extracted subsets of reads may contain insertion and deletion (indel) errors as well. These indel errors are handled at the multiple sequence alignment stage. Insertions are directly removed and the deletions are recovered by the alignment mechanism.

We note that this reads extraction step is very similar to the reads extraction step used in variant calling studies (Van der Auwera, Carneiro, Hartl, Poplin, Del Angel, Levy-Moonshine, Jordan, Shakir, Roazen, Thibault et al. 2013, DePristo et al. 2011). But the purpose and assumptions are polarly different. The purpose of variant calling studies is to identify variations between genomes and the reference genome is assumed to be error-free. But the purpose of our study is to make corrections for the possible errors in the extracted reads, and the reference genome is assumed to be not error-free. Variant calling studies do not have any attempt to correct the possible errors in the extracted reads. Our error-corrected reads can be used for potentially better variant calling analysis.

In the reads extraction step, we actually extend the sequence I_g at both ends with 50 nucleotide bases, to guarantee that some reads crossing the boundary of I_g can be extracted as well. Through the extension of the gene sequence and the noise-tolerant mapping process, more reads are extracted as

far as possible. We note that a few reads mapped to the nucleotide sequence I_g with high mapping scores may belong to other genes (the repetitive areas). So in a further step, we double-check whether a read should be collected in $subset(S, I_g)$.

3.2.2 Error correction step

After $subset(S, I_g)$ is formed, we align all the reads in $subset(S, I_g)$ according to their positions in I_g, and place them one by one in each row in an increasing order of their start position. This sorted organization of $subset(S, I_g)$ is called an alignment array. The alignment array is traversed column-by-column for error correction. Intuitionally, if a base has a very low type frequency in the column, this base (i.e., an outlier) is very likely to be erroneous. The key idea is to detect dominance information in the columns according to the nucleotide type distribution and to locate error bases in the rows according to their error-aware probabilities.

Suppose only four nucleotide types (i.e., A, C, G, and T) are in the reads. For a column of bases in the alignment array, there are four possible cases for the nucleotide type distribution:

- One-type dominance. All or almost all of the bases have the same nucleotide type. For example, 99% of the bases in the column are nucleotide type 'A'; all the other bases ('C', 'G', or 'T') constitute the remaining 1% of the bases. These 1% of the bases are outlier bases or erroneous bases.

- Two-type dominance. All or almost all of the bases are split into two main nucleotide types.

- Three-type dominance. All or almost all of the bases are split into three main nucleotide types.

- Four-type dominance. All of the bases are split into four main nucleotide types.

We say a column is dominated by one or more types of bases if the total count of *the other types* of bases is 0, 1, 2, or 3; or the total percentage of the other types of bases is less than 2% when the total number of bases in the column is 100 or more. These thresholds can be adjusted according to data characteristics.

The respective error correction is as follows:

- Correction for one-type dominance. Suppose the dominant type of bases is X, then change all other type(s) of base(s) to X for correction;
- Correction for two-type dominance. Suppose the two dominant types of bases are X and Y, then change all other type(s) of base(s) to X and Y proportional to the percentages of X and Y;
- Correction for three-type dominance. Suppose the three dominant types of bases are X, Y and Z, then change all other bases to X, Y and Z proportional to the percentages of X and Y and Z;
- Correction for four-type dominance. No correction is needed.

Let $f(X)$ denote the percentage of X in the column, namely the frequency of X. Some examples of the base distribution and error correction are: (i) $f(A) = 99\%$, $f(T) = 0.5\%$, $f(G) = 0.5\%$ (dominated by one type), change all the Ts and Gs to A; (ii) $f(T) = 40\%$, $f(G) = 58\%$, $f(A) = 0.8\%$, $f(C) = 1.2\%$ (dominated by two types), change all the As and Cs to T and G in the ratio 40:58; $f(T) = 40\%$, $f(G) = 58\%$, $f(A) = 2\%$ (dominated by two types), change all the As to T and G in the ratio 40:58; (iii) $f(T) = 40\%$, $f(G) = 41\%$, $f(A) = 18\%$, $f(C) = 1.0\%$ (dominated by three types), change all the Cs to T and G and A in the ratio 40:41:18; $f(T) = 40\%$, $f(G) = 41\%$, $f(A) = 19\%$ (dominated by three types), no change; and (iv) $f(T) = 25\%$, $f(G) = 40\%$, $f(A) = 30\%$, $f(C) = 5\%$ (dominated by four types), no change.

If the minor types of the bases have the same frequency at multiple columns, for a conservative correction, we set priorities to change those bases

at the columns with a less number of dominant types. The order is: one-type dominance is prior to two-type dominance which is prior to three-type dominance. The priority value of base V is set as 0.1 if V is at a one-type dominance column, denoted by $p(V) = 0.1$; set as 0.2 if V is at a two-type dominance column, denoted by $p(V) = 0.2$; and set as 0.3 if V is at a three-type dominance column, denoted by $p(V) = 0.3$.

We then traverse the alignment array row-by-row to make the conservative error correction. For each row, we rank all the bases $r_1 r_2 \cdots r_n$, according to their base type frequency together with their dominance value (i.e., $f(r_i) + p(r_i)$), into an increasing order. Since Illumina sequencing data (used in this work) has an error rate around 0.5% to 2%, the first two per cent of bases in a row are considered as errors. Then these bases are confirmed to change. Before changes, we check the number of dominant types in the column. If there are more than one potential dominant type to correct, we consider its neighbor columns as well. We give a high priority to corrections which is followed by dominant types with large number of bases.

Note that in the situation of two-type or three-type dominance, some of the reads in $subset(S, I_g)$ are not relevant to gene g. They may come from another gene with a repetitive region of g. This issue is not solvable by the reads extraction step; it is only identifiable in the alignment step. In this work, if more than one of bases' probability in the top two per cent bases is larger than the threshold, we assume the read are more likely from the other part of the genome sequence I, instead of from the sequence of the gene I_g. These reads are labeled 'out' and deleted from $subset(S, I_g)$ for the contig construction of gene g. An example of the correction is shown in Figure 3.1. The pseudo code of the correction algorithm is shown in Algorithm 3.1.

3.3 Experiments and Results

We compare the error correction performance of InsEC with instantialized state-of-the-art tools Bcool (Limasset et al. 2020), BFC (Li 2015) and

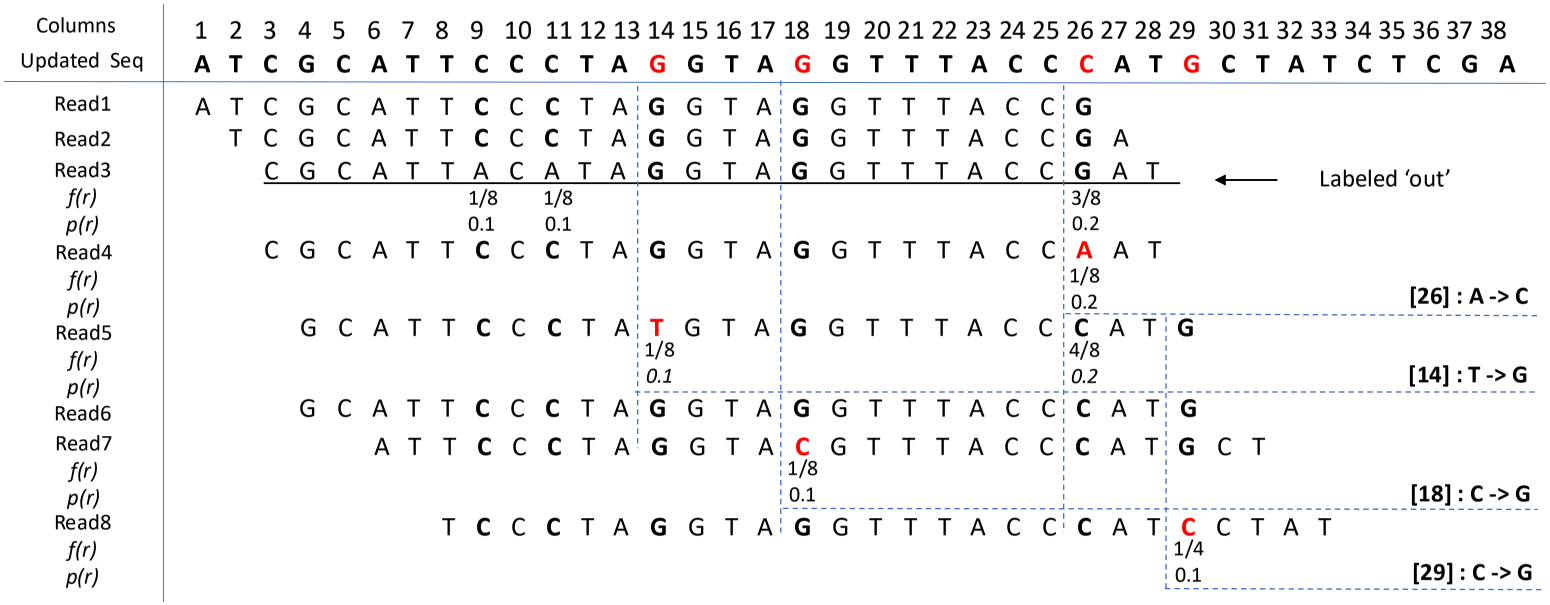

Figure 3.1: An example of read correction on eight reads. The base-type frequency $f(r)$ and dominance value $p(r)$ of a base are shown below that base. For the columns of bases, the dominant nucleotide types are in bold and the erroneous bases are in the red color. The updated sequence is on the top and the correction details are listed in the right. For example, in column 26, there are two dominant nucleotides (G *and* C). $f(C)$ is larger than $f(G)$, so the nucleotide C is used to update the sequence, and the erroneous base A (position[26]) in read4 is corrected to C. For read3, the third base after ranking is below the threshold, so read3 is labeled 'out' and deleted from the extracted subset.

Algorithm 3.1 Error Correction

Input: An extracted read set $\mathcal{R} = \{R^1, R^2, \cdots, R^m\}$, their corresponding alignment position $\{u_i\}_{i=1}^m$ and threshold λ

Output: A corrected read set $\mathcal{S}'$ and an updated nucleotide sequence H

Function ERROR_CORRECTION $(\mathcal{R}, \{u_i\}_{i=1}^m, \lambda)$

begin

$q \leftarrow \max(u_i)$

$H \leftarrow$ empty array ▷ *[r]Store the updated nucleotide sequence

for $j = 1$ **to** $(q + n - 1)$ **do**

$(c, d) \leftarrow (0, 0)$

for $i = 1$ **to** m **do**

if $u_i \leq j \leq (u_i + n - 1)$ **then**

$f(j, r^i_{(j-u_i+1)}) \leftarrow f(j, r^i_{(j-u_i+1)}) + 1$

$c \leftarrow c + 1$

$H(j) \leftarrow \underset{x \in \{A,G,T,C\}}{\text{argmax}} (f(j, x))$

foreach $x \in \{A, T, C, G\}$ **do**

if $f(j, x) > 0$ **then**

$d \leftarrow d + 1$

foreach $x \in \{A, T, C, G\}$ **do**

$f(j, x) \leftarrow f(j, x)/c + d * 0.1$

$\mathcal{S}' \leftarrow \emptyset$ ▷ *[r]Store the corrected reads **for** $i = 1$ **to** m **do**

for $j = 1$ **to** n **do**

$t(j) \leftarrow f(u_i + j - 1, r^i_j)$

$[B, I] \leftarrow \text{sort}(t(1...n))$ ▷ *[r]ascending order; B is sorted array and I is an index array; the k-th smallest is $t(I(k))$ **if** $B(3) > \lambda$ **then**

$r^i_{I(1)} \leftarrow H(I(1) + u_i - 1)$

$r^i_{I(2)} \leftarrow H(I(2) + u_i - 1)$

Append R^i to $\mathcal{S}'$

return $\mathcal{S}'$, H

Coral (Salmela & Schröder 2011). Bcool is the latest method published in year 2020; BFC and Coral are two classical error correction methods, representing the *k*mer based methods and the multi-alignment error correction methods respectively. Our experiments are conducted on both simulated and real sequencing data. The ground truth of the genome sequence is not available for the real datasets, so the simulated datasets are used as a supplement to the real data experiments. With the ground truth provided by the simulated datasets, we are able to evaluate error correction and further assembly performance objectively for all of the methods. Our InsEC method is designed for error correction on disease-causing genes, so seven genes related to lung cancers are selected to illustrate method performance in the following experiments.

3.3.1 Sequencing Read Datasets

Illumina sequencing datasets are available at the Sequence Read Archive (SRA) (`https://www.ncbi.nlm.nih.gov/sra/`); and the simulated Illumina sequencing data can be produced by ART (Huang, Li, Myers & Marth 2011) which is a benchmark tool for the generation of simulated short reads. The real dataset used in this work is ERR174310, which contains paired-end human whole genome deep sequencing reads generated by Illumina HiSeq 2000. We denote this dataset as D0. The two simulated sequencing datasets (denoted by D1 and D2) have the same read length and the same sequencing platform as ERR174310. D1 is a single-end dataset, and D2 is a paired-end dataset, both generated with reference to the standard sequence of human chromosome one.

The genome annotations are obtained from the NCBI (National Center for Biotechnology Information) (`https://www.ncbi.nlm.nih.gov/genome/`), including gene name, gene ID and gene positions. More details of these datasets are shown in **Table 3.1**. The seven genes related to lung cancer in this study are ILR6R, IL10, ATF3, GRIK3, MYCL, PRDX1, and ENO1. All of these genes are located at chromosome one. The nucleotide sequences

of the genes are available at the NCBI gene database (https://www.ncbi.nlm.nih.gov/gene/). The length of these genes ranges from 4,892 to 238,602 bases. See more details of these genes in **Table 3.2**.

Table 3.1: Description of the datasets

Dataset	Real dataset	Simulated dataset	
	D0	D1	D2
Read length	100	100	100
Total reads	586,941,413	23,046,123	23,048,001
Type of reads	paired-end	single-end	paired-end
Accession No.	ERR174310	Simu-Single	Simu-Pair
Reference	Human Genome	Chromosome.1	Chromosome.1

Notes: The latest version of human genome, GRCh38.P13, is used in our experiments as of September 2019.

3.3.2 Evaluation Metrics

The performance is evaluated not only on the error correction but also on the read assembly before and after the error correction.

Metrics for Correction Performance

To assess the accuracy of the correction methods, we use the following three metrics, Precision (*TP/(TP+FP)*), Recall (*TP/(TP+FN)*), Gain (*(TP-FP)/(TP+FN)*). More details are shown in Section 2.3.1.

Metrics for Assembly Performance

To assess the impact of error correction on the assembly results, we compare InsEC with other state-of-the-art methods by standard assembly assessment metrics. We choose SPAdes (Bankevich et al. 2012) to assembly read data

Table 3.2: Genes related to lung cancer on human chromosome one

Gene_ID	Gene_Name	Gene_length	Gene_function
Gene1 (g1)	IL6R	64257	protein_coding
Gene2 (g2)	IL10	4892	protein_coding
Gene3 (g3)	ATF3	55443	protein_coding
Gene4 (g4)	GRIK3	238602	protein_coding
Gene5 (g5)	MYCL	6830	protein_coding
Gene6 (g6)	PRDX1	12011	protein_coding
Gene7 (g7)	ENO1	18250	protein_coding

Notes: The details of genes are from the genome annotation of the latest version GRCh38.P13

before and after error correction, except that the error-free datasets are assembled for the performance assessment as well. To assess our method more specificlly, each nucleotide in the gene sequence updated by InsEC is compared with its in gene reference. On simulated dataset, the ground truth of gene sequence is available, so the more similar the updated sequence with the referferce is, the better performance of assembly is.

- Assembly results comparison: The assembly results are evaluated by QUAST (Gurevich et al. 2013), a quality assessent tools for genome assemblies. Detailed reports include the number of contigs, the largest contigs and N50. A contig is a continuous nucleotide sequences obtained from the assembly process. N50 is defined as the minimum contig length needed to cover 50% of genome.

- The Reference vs the corrected sequence: The nucleotide of gene sequences, updated by our method, are compared with the reference sequence of genes base-by-base. The less difference between the two sequences is, the better assembly performance is.

3.3.3 Performance Evaluation

For each g of the seven lung cancer disease-associated genes, we constructed $subset(D1, I_g)$ and $subset(D2, I_g)$, and conducted instance-based error correction by InsEC. Strictly on these two subsets of reads, we also apply three state-of-the-art global correction methods Bcool (Limasset et al. 2020), BFC (Li 2015) and Coral (Salmela & Schröder 2011) to rectify errors for a fair comparison. This is exactly so called "global approaches can be turned into instance-based approaches" as stated in Introduction. The overall error correction performance by InsEC, Coral, BFC and Bcool on the seven lung cancer disease genes are presented in **Table 3.3**.

Our method InsEC achieved the best precision, recall and gain rate on all of the datasets. In particular, the average precision, recall and gain rate by our method are much superior respectively by 3.13%, 21.9% and 24.44% to the latest method Bcool on the single-end datasets, and much superior respectively by 4.14%, 2.99% and 7.2% on the paired-end datasets. More importantly, our method improved the gain rates a lot, implying more number of bases are rectified and less number of errors are induced compared with the existing methods. In detail, InsEC improved the gain rates ranging from 9.57% to 24.44% on the single-end datasets, and improved the gain rates ranging from 6.98% to 7.71% on the paired-end datasets. It is noted that the other methods are sensitive to data types. All of the other methods perform better on pair-end datasets than single-end datasets, especially the gain rate improved from 3.28% to 18.31%. While our method InsEC shows good robustness on both single-end and pair-end datasets, achieving the gain rate at 97.55% and 98.62% respectively.

All the experiments were conducted on a computing cluster running Red Hat Enterprise Linux 6.7 (64 bit) with Intel Xeon E5-2695 v3 and 128 GB RAM. We use the Linux/Unix time command to record the system time and memory usage. The average running time (seconds) of InsEC, Coral, BFC and Bcool is 3.2s, 1.55s, 1.02s and 18.92s and the average memory usage (kbytes) is 503,271kb, 419,156kb, 1,109,266kb and 527,268kb respectively.

Table 3.3: Performance comparison of instance-based error corrections

		On single-end reads				On paired-end reads			
		`Ins_EC`	`Coral`	`BFC`	`Bcool`	`Ins_EC`	`Coral`	`BFC`	`Bcool`
Precision (%)	g1	**98.42**	95.95	91.91	93.01	**99.49**	94.46	90.96	89.82
	g2	**100**	99.65	**100**	94.70	**100**	97.39	**100**	98.18
	g3	**99.64**	92.19	93.90	95.48	**99.85**	93.49	94.10	97.97
	g4	**99.93**	94.86	97.34	96.30	**99.97**	95.19	98.18	98.00
	g5	**100**	**100**	**100**	98.68	**100**	90.16	95.00	96.02
	g6	**98.56**	93.36	91.64	87.73	**99.27**	92.30	95.35	91.49
	g7	**100**	99.25	99.87	93.79	**100**	98.57	96.02	95.81
	AVE	**99.51**	96.47	96.38	94.24	**99.80**	94.51	95.66	95.33
Recall (%)	g1	**95.06**	91.06	78.64	79.38	**96.78**	93.92	95.04	86.78
	g2	**97.26**	95.65	71.91	89.63	**99.32**	97.39	95.93	96.42
	g3	**98.07**	97.07	76.00	89.23	**98.48**	97.92	95.49	92.75
	g4	**97.16**	96.97	78.19	91.25	**97.82**	97.05	97.75	93.85
	g5	**99.34**	61.84	69.74	98.03	**99.78**	90.16	94.44	95.73
	g6	**99.60**	96.44	76.91	79.34	**99.69**	97.20	96.65	81.10
	g7	**99.72**	96.81	71.52	86.53	**99.87**	98.41	95.48	89.78
	AVE	**98.03**	90.83	76.13	87.63	**98.82**	96.01	95.83	90.91
Gain (%)	g1	**93.54**	87.95	71.72	79.38	**96.29**	89.66	85.60	86.78
	g2	**97.26**	95.64	71.91	89.63	**99.32**	95.71	95.93	96.42
	g3	**97.71**	89.56	71.06	89.23	**98.34**	92.15	89.50	92.75
	g4	**97.09**	92.76	76.06	91.25	**97.79**	93.29	95.94	93.85
	g5	**99.34**	61.84	69.74	98.03	**99.78**	81.04	89.47	95.73
	g6	**98.15**	91.32	69.90	79.34	**98.96**	91.23	91.94	81.10
	g7	**99.72**	96.79	71.43	86.53	**99.87**	98.39	91.52	89.78
	AVE	**97.55**	87.98	73.11	87.63	**98.62**	91.64	91.42	90.91

Notes: AVE indicates the average score over the seven genes. Bold font indicates the best result in the row.

Our InsEC ranks the second in running time and memory usage.

The global approaches improved when focusing on disease-associated genes

To show the significance of instance-based error correction for the reads related to disease-causing genes, we compare the error correction performance on the whole sequencing datasets with those on the gene-related subsets of reads. After running error correction on the whole datasets D1 and D2, those reads relevant to the given gene g are extracted for performance assessment and comparison. The methods are specially denoted as Bcool_g, BFC_g and Coral_g in this situation. The overall error correction performance for lung cancer-associated genes is presented in **Table 3.4**.

These global error correction methods got improved when directly applied to the subsets of reads related to the gene-associated genes, namely the gain rates by Coral, BFC, and BCOOL are better than their global versions (labeled with _g), increasing the performance from 2.56% to 7.61%.

Performance of read assembly after error correction

To see whether the error correction has impact on the quality of the assemblies, we compare on the number of contigs, the longest contigs and N50 before and after the error correction of D1 and D2. We also construct the assemblies from the error-free read sets (the ground truth is available for the simulated datasets). The best error correction method is expected to have the most similar assembly results to those from the error-free dataset. The differences in the assembly results between the error-free datasets and corrected datasets after error correction by all the methods are listed in **Table 3.5**. There are no differences in assembly results for the other four genes, so their results are not listed in table.

The assembly results get improved after the error correction. In particular, there is an increasing trend at the length of contigs after the error correction, and a decreasing trend at the number of contigs. Compared with

Table 3.4: Performance comparison. Instance-based approach vs global approach

		On single-end reads						On paired-end reads					
		Coral_g	Coral	BFC_g	BFC	Bcool_g	Bcool	Coral_g	Coral	BFC_g	BFC	Bcool_g	Bcool
Precision (%)	g1	97.80	95.95	85.93	91.91	90.21	93.01	97.97	94.46	92.86	90.96	91.12	89.82
	g2	99.66	99.65	94.30	100	94.70	94.70	92.43	97.39	99.00	100	98.18	98.18
	g3	98.81	92.19	94.58	93.90	95.48	95.48	98.41	93.49	98.04	94.10	97.32	97.97
	g4	98.97	94.86	96.45	97.34	96.30	96.30	98.61	95.19	98.02	98.18	98.00	98.00
	g5	98.68	100	100	100	98.01	98.68	96.87	90.16	91.03	95.00	95.02	96.02
	g6	93.33	93.36	77.63	91.64	87.73	87.73	92.97	92.30	89.41	95.35	91.49	91.49
	g7	99.02	99.25	90.27	99.87	93.79	93.79	95.77	98.57	92.39	96.02	95.81	95.81
	AVE	98.04	96.47	91.31	**96.38**	93.75	**94.24**	96.15	94.51	94.39	**95.66**	95.28	**95.33**
Recall (%)	g1	72.48	91.06	75.91	78.64	76.99	79.38	74.61	93.92	92.94	95.04	79.29	86.78
	g2	96.99	95.65	82.94	71.91	89.63	89.63	95.60	97.39	96.26	95.93	96.42	96.42
	g3	88.00	97.07	75.62	76.00	89.23	89.23	90.19	97.92	96.65	95.49	92.47	92.75
	g4	90.26	96.97	78.03	78.19	91.25	91.25	91.47	97.05	96.76	97.75	93.85	93.85
	g5	98.68	61.84	79.61	69.74	97.37	98.03	98.80	90.16	90.67	94.44	94.74	95.73
	g6	67.09	96.44	77.04	76.91	79.34	79.34	68.96	97.20	94.00	96.65	81.10	81.10
	g7	82.35	96.81	80.16	71.52	86.53	86.53	86.31	98.41	93.78	95.48	89.78	89.78
	AVE	85.12	**90.83**	78.47	76.13	87.19	**87.63**	86.56	**96.01**	94.44	**95.83**	89.66	**90.91**
Gain (%)	g1	71.32	87.95	63.48	71.72	68.63	79.38	73.79	89.66	85.79	85.60	71.56	86.78
	g2	96.98	95.64	77.93	71.91	84.62	89.63	88.93	95.71	95.28	95.93	94.63	96.42
	g3	87.66	89.56	71.29	71.06	85.01	89.23	89.57	92.15	94.72	89.50	89.92	92.75
	g4	90.15	92.76	75.16	76.06	87.75	91.25	91.24	93.29	94.81	95.94	91.93	93.85
	g5	97.37	61.84	79.61	69.74	95.39	98.03	96.77	81.04	81.73	89.47	89.77	95.73
	g6	63.58	91.32	54.85	69.90	68.24	79.34	64.52	91.23	82.87	91.94	73.56	81.10
	g7	82.05	96.79	71.52	71.43	80.80	86.53	83.40	98.39	86.05	91.52	85.86	89.78
	AVE	84.16	**87.98**	70.55	**73.11**	81.49	**87.63**	84.03	**91.64**	88.75	**91.42**	85.32	**90.91**

Notes: AVE indicates the average score over the seven genes. Bold font indicates the better result compared methods with its _g version.

Table 3.5: Assembly results compared with the ground truth

	Single-end reads								
	g1			g3			g4		
	NO.	Lar.	N50	NO.	Lar.	N50	NO.	Lar.	N50
Truth	6	24854	11363	3	27822	27822	3	187434	187434
Raw	-1	3170	-1316	0	50	13879	-2	37224	37224
InsEC	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-13**	**-13**
Coral	2	-11485	0	0	-50	-13879	1	15	15
Corel_I	4	-28181	-41672	1	-13824	-13824	0	-186	-186
BFC	-1	3170	-1316	0	50	13879	0	34	34
BFC_I	-2	0	0	0	50	50	0	34	34
Bcool	-1	3198	-1316	0	50	13879	-2	52585	52585
Bcool_I	-1	0	0	0	-6	-6	-1	52505	52505
	Paired-end reads								
	g1			g3			g4		
	NO.	Lar.	N50	NO.	Lar.	N50	NO.	Lar.	N50
Truth	3	40458	40458	2	27893	27893	6	134849	134849
Raw	-4	13097	27287	0	63	63	-3	15530	15530
InsEC	-1	0	0	**0**	**0**	**0**	**0**	**13**	**13**
Coral	0	0	0	0	91	91	-2	302	302
Corel_I	2	-23894	-23894	1	-27650	-27650	2	-23329	-23329
BFC	0	0	0	0	91	91	-2	783	783
BFC_I	-2	0	0	0	69	69	-2	813	813
Bcool	**0**	**0**	**0**	0	91	91	-3	15565	15565
Bcool_I	0	63	63	0	63	63	-2	64126	70640

Notes: Truth row indicates the assembly results of the error-free read data. Other rows show the difference value where value in Truth row minus the current row. NO. indicates the number of contigs. Lar. Indicates the largest length of contigs.

the other error correction methods, InsEC has the most similar assembly results to those from the error-free datasets for 5 of the 6 cases; on the remaining one, the result of our method has only one difference in the number of contigs. Furthermore, we achieved the identical assembly results as those from the error-free datasets g1, g3 and paired-end g3.

The contig quality are shown in **Table 3.6**, where the numbers of base differences between the contigs from our corrected reads and those from the reference sequences are presented.

Table 3.6: The contigs from corrected reads vs the reference sequence

Contig_Q	g1	g2	g3	g4	g5	g6	g7
Single-end_D1	6/64258	M	5/55444	7/238603	M	2/12012	M
Paired-end_D2	6/64258	M	5/55444	6/238603	M	M	M

Notes: The sign 'M' indicates the contig assembled from the corrected reads by our method and the reference sequence are identical. 6/64258 indicates there are 6 different bases in 64258 bases, and similarly for other number combinations.

Most of the contigs assembled from the corrected reads by our method are identical to the reference sequences (see the sign 'M'); while the remaining assemblies have only tiny differences from the reference sequences (e.g., only 7 or 6 base differences over a length of 238,603 bases).

Case studies: error correction at mutation-prone regions in the lung cancer associated genes

On the real sequencing reads dataset D0, we have performed instance-based error correction for the reads relevant to EGFR and KARS which are two genes highly associated with lung cancer (El-Telbany & Ma 2012). Some of our corrections happened at the mutation-prone regions of EGFR. These point mutations or mutation combinations are known (Marchetti,

Martella, Felicioni, Barassi, Salvatore, Chella, Camplese, Iarussi, Mucilli, Mezzetti et al. 2005) to make lung carcinomas more responsive to treatments with tyrosine kinase inhibitors. These mutations are usually at least one base different from a reference sequence, also referred to 'variant calling'. One of the corrections changes A to G at the SNP:rs1476431328 position, located at chr7:55205427. Due to this base correction from A to G, the corresponding amino acid is changed from Asparagine (AAC) to Serine (AGC). If this base is not corrected, the amino acid Asparagine instead of the correct amino acid Serine would be focused in the downstream analysis which may lead to different conclusions about the functions of the protein. This is quite possible because Asparagine and Serine pose their own distinct biophysical properties. Another of our corrections is at SNP:rs781609053 which changes nucleotide T to C. Correspondingly, the amino acid would be changed from Methionine (ATG) to Threonine(ACG). Furthermore a correction was performed at SNP:775317295 which changes nucleotide C to T, implying that the amino acid Proline (CCA) should be changed to Leucine (CTA). The effects of mutations lead to different structures of its coding proteins, thereby affecting its functions (Marks, Colwell, Sheridan, Hopf, Pagnani, Zecchina & Sander 2011), which is shown in Figure 3.2, where we use SWISS-MODEL (Waterhouse, Bertoni, Bienert, Studer, Tauriello, Gumienny, Heer, de Beer, Rempfer, Bordoli et al. 2018) to model the structure of coding protein according to its amino acids sequence. The amplification of gene KARS primarily decides the growth and survival of lung cancer cell lines (Lutterbach, Zeng, Davis, Hatch, Hang, Kohl, Gibbs & Pan 2007). For the reads in D0 that are relevant to KARS, some of our instance-based error corrections also occurred at its SNP positions. The correction from A to G at SNP:rs35225896 changes the corresponding amino acid from Isoleucine (ATA) to Methionine (ATG). Highly accurate sequences near this position should be ensured, as mutations at this position

[0]In Figure 3.2, the mutation bases and changed amino acids are highlighted by green and blue color. The predicted structure of coding proteins are shown in the right side.

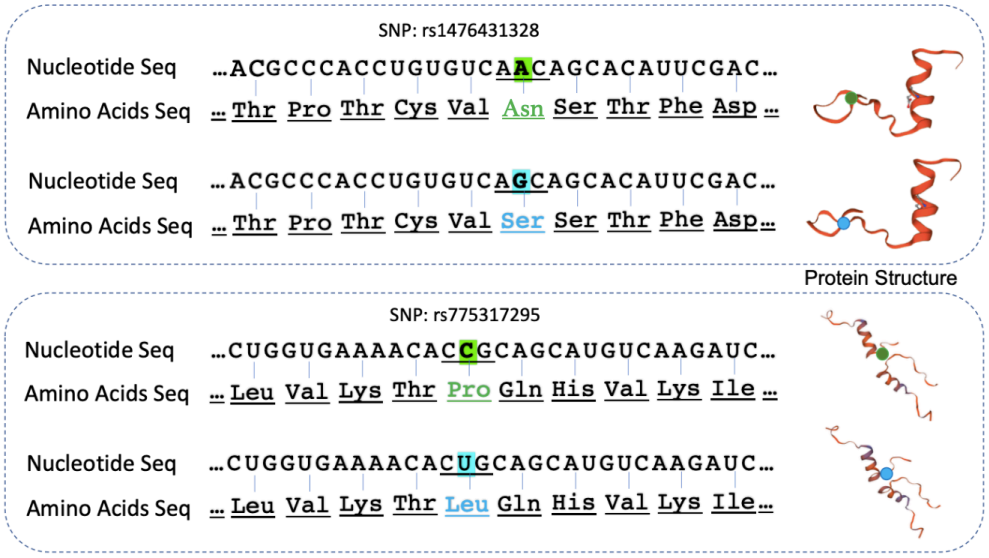

Figure 3.2: Two examples of point mutations in case studies.

are closely related to hereditary cancer-predisposing syndrome, supported by clinical significance and publications (https://www.ncbi.nlm.nih.gov/snp/rs35225896). Error corrections at non-coding regions are important as well. For instance, our correction at SNP:rs11762213 changes the nucleotide from G to A. Though such corrections at non-coding regions do not effect type of amino acids, SNP:rs11762213 is recognized as a predictor of adverse outcomes in clear cell renal cell carcinoma (Hakimi, Ostrovnaya, Jacobsen, Susztak, Coleman, Russo, Winer, Mano, Sankin, Motzer et al. 2016). Thus, high-quality corrections at mutation-prone regions (coding and non-coding regions) are very important for downstream SNP and mutation studies.

3.4 Summary

Our approach (named InsEC) is contrast to the existing error correction methods which all take a global approach to make a genome-wide error correction. Genome-wide error correction is not good enough especially when

the study is focused on disease genes or pathways. InsEC's correction step adequately exploits fine-grained local patterns so as to rectify those errors which were unable to be corrected by the global approach. The reason is that the instance-based approach can significantly moderate the global approach's issue on the non-uniform sequencing depth. We have conducted extensive experiments on simulated single-end and paired-end reads. The performance evaluation confirms that InsEC has much superior precision, recall and gain rate over the state-of-the-art methods on various sets of reads related to lung cancer genes. InsEC can also provide an assembled nucleotide sequence of the corrected reads which is closer to the ground truth than the other methods on the simulated datasets. Our SNP case studies on the real paired-end reads show that the error correction can happen at the mutation-prone bases stored at the current SNP databases, implying that highly accurate instance based approach is particularly useful for SNP and mutation investigations.

In this work, we have proposed a novel approach for short reads error correction. The method is an instance-based approach, or a local approach, to rectify all possible errors in the reads relevant to a disease gene, or a subset of disease-associated genes. Our novel idea is to exploit local sequence features and statistics directly related to these genes. Two main steps can collects reads relevant to a given gene from a WGS dataset through a noise-tolerant mapping technique and take advantage of alignment processes and rectify errors according to fine-grained patterns and statistics. InsEC achieves good performance on both single-end and pair-end datasets, and can also provide an assembled nucleotide sequence for gene sequence studies. This study successfully serves as read preprocess tools to provide high-quality data for targeted genes or genome region research.

Availability

The data material and code are all available at the github link (`https://github.com/XuanrZhang/InsEC.git`).

Chapter 4

Error Correction Method for MicroRNA Sequencing Reads

4.1 Introduction

With rapid developments of sequencing technology, high-throughput platforms have inexpensively produced huge amounts of genomic reads at unprecedented speed (Goodwin, McPherson & McCombie 2016), for example by whole genome sequencing, total RNA sequencing, mRNA sequencing and small RNA sequencing. Recently, sequencing of miRNAs (a special type of small RNA molecules containing about 22 nucleotide bases) has been widely used to examine tissue-specific expression patterns, to identify isomiRs (mature miRNA variants) and to discover previously uncharacterized miRNAs (Yeung et al. 2016, Xiao & MacRae 2019, Giraldez et al. 2018, Tan et al. 2014, Trontti et al. 2018, Fernandez-Valverde et al. 2010). As key regulators in various biological processes, miRNA dysregulation is implicated in many diseases for example cancer and autoimmune disorders (Meng et al. 2017, Liu et al. 2018, Telonis et al. 2017, Dutta et al. 2019, Dai et al. 2019). Numerous studies also reaffirm that miRNA regulatory functions are involved in post-transcriptional gene silencing (PTGS), transcriptional gene silencing (TGS), and transcriptional gene activation (TGA) (Pisignano

et al. 2017, Yang et al. 2020), in which miRNAs bind to nascent RNA transcripts, gene promoter regions or enhancer regions and exert further effects via epigenetic pathways (Liu et al. 2018, Liu et al. 2019).

Fine-granulated analysis of miRNA reads at single-base resolution for uncovering their isoforms (isomiRs) and alternative splicing is one of the most frontier research areas in this field (Liao et al. 2018, Tan et al. 2014, Liu, Lai & Guo 2020, Bilanges et al. 2019, Sänger et al. 2020, Pillman et al. 2019, Hoefer 2020). IsomiRs vary in size and base content, due to the alternative and imprecise cleavage of Drosha and Dicer, or the turnover of miRNAs (Neilsen, Goodall & Bracken 2012, Hoefer 2020). IsomiRs have been classified into four categories: 5' trimmed isomiRs, 3' trimmed isomiRs, 3' addition isomiRs, and polymorphic isomiRs (Lan, Peng, McGowan, Hutvagner & Li 2018). 5'/3' trimmed or addition isomiRs are defined as those miRNA sequences which have one or more bases trimmed or added respectively at the 5' or 3' end from the canonical miRNAs, while polymorphic isomiRs usually have substitution mutations, causing one or more bases different from the canonical miRNA. For such broad range of miRNAome analysis, super high-quality sequencing data is demanded because the definitions are very sensitive to the base positions—one base difference can lead to entirely different read categorization.

High-throughput sequencing technology produces short reads containing approximately 1% erroneous bases (Salk, Schmitt & Loeb 2018, Goodwin et al. 2016, Laehnemann, Borkhardt & McHardy 2016) such as aberrations of substitutions, base insertions, or deletions (indels). A previous study reported that the error percentage of most Illumina reads is approximately 0.5% at best (Mardis 2013). These randomly distributed errors or even erroneous bases at only one position can cause lowered copy numbers for miRNA reads, and thus affect the calculation of miRNA expression levels and differential folds (Bartel 2004, Chekulaeva & Filipowicz 2009, Yu, Pillman, Neilsen, Toubia, Lawrence, Tsykin, Gantier, Callen, Goodall & Bracken 2017, Telonis & Rigoutsos 2018, van der Kwast, Woudenberg, Quax

& Nossent 2020). Suppose an miRNA isoform has 100 copies expressed in a diseased cell, if there are substitution errors happened in 5 copies of them during the sequencing, 3 deletion errors in the other copies, and two insertion errors as well, then the total copy number would be counted as 90 which is away from the ground truth. Further, the data may lead to wrong identification of isomiRs without correction of these errors. For example, an miRNA isoform containing the deletion errors would be wrongly identified as a 5' trimmed isoform of a canonical miRNA; If the errors occur at the seed region of an miRNA (conserved region of miRNAs), its target specificity analysis would be affected, potentially increasing the number of target transcripts (Cloonan, Wani, Xu, Gu, Lea, Heater, Barbacioru, Steptoe, Martin, Nourbakhsh et al. 2011, Mullany, Herrick, Wolff & Slattery 2016, Neilsen et al. 2012). Although current research adopts "abandon ambiguity reads or noise reads" to avoid misinterpreting erroneous sequence variants (ESVs) as isomiRs, the approach inevitably losses a part of the precious raw data (Guo & Chen 2014, Ebhardt, Tsang, Dai, Liu, Bostan & Fahlman 2009). It is demanded to develop sophisticated algorithms to rectify these aberrations for truth-closer analysis of miRNA reads in the wide range of applications.

None of the existing error correction methods suits well for miRNA sequencing data, since they have not considered the unique characteristics of miRNA reads (short length and varying per read coverage). Besides, most of the methods, designed for DNA or mRNA sequencing reads, only focus on the correction of substitution errors and do not support indels error correction. So far, these methods have taken two streams of different correction ideas. The first one is a kmer-based error correction idea, represented by BCOOL (Limasset et al. 2020), BFC (Li 2015), ACE (Sheikhizadeh & de Ridder 2015), and BLESS (Heo et al. 2014). The step is to examine the frequencies of kmers to distinguish between solid and weak kmers according to a fixed global frequency threshold. Then the solid kmers (assumed as error-free) are referred as templates to rectify weak kmers (assumed error-containing)

to obtain correct reads. The second one is a multi-alignment based error correction approach, represented by coral (Salmela & Schröder 2011), ECHO (Kao et al. 2011), and Karect (Allam et al. 2015). These methods usually group those reads sharing the same kmer and then concatenate such a group of reads to form a long consensus contig. The contig is assumed error-free to correct erroneous bases. There are also a few methods designed for RNA sequencing reads error correction, for example Seecer (Le, Schulz, McCauley, Hinman & Bar-Joseph 2013) and Rcorrector (Song & Florea 2015). These approaches do not work for miRNA sequencing data error correction. For example, the consensus idea is not applicable to miRNA data because each read already encompasses one entire miRNA sequence. Our study did verify that the existing methods tend to significantly under-correct the errors and are prone of introducing tremendous number of new errors.

We present an error RECtification method for miRNA sequencing reads (named miREC), which is the first tool to address the problem of miRNA sequencing errors. Unlike the existing methods which have the primary goal of correcting substitution errors, our miREC concentrates more on insertion and deletion errors for excellent correction performance. The novel step of our method is the use of a 3-layer (k-1)mer-kmer-(k+1)mer lattice structure to maintain the frequency differences of the kmers (Figure 4.1). These superset-subset frequency differences are very effective to detect the errors especially the indel errors. The lattice structure is also a moving structure where k is set continuously from a small number to a big number 23 or 25 for a full coverage of error correction. Extensive tests on both simulated and wet-lab (experimentally catalogued) miRNA sequencing datasets show that miREC can excel performance in all of precision, recall and gain.

4.2 Methods

An miRNA sequencing read r is a sequence $r_1 r_2 \cdots r_n$, $r_i \in \Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{N}, \mathrm{T}\}$, where A, C, G and T stand for the nucleotide bases Adenine,

Cytosine, Guanine and Thymine respectively, and the character N stands for a uncertain nucleotide; n is the length of r. Usually, the length n of an miRNA read ranges from 15 to 28 in a dataset, but each read encompasses one entire miRNA. A kmer *substringk* is a contiguous subsequence in a read r.

4.2.1 A 3-layer Kmer Lattice Structure

Given an miRNA sequencing read multi-set RS and a setting k, the copy count (or frequency) of a distinct read r in RS is the total number of its copies in RS, and the copy count (or frequency) of a distinct kmer in RS is the total number of its copies in RS. KMC3 (Kokot, Długosz & Deorowicz 2017) is used by this work as a kmer counter for these calculations. Consider a kmer *substringk*, this kmer's k-neighborhood is defined as the set of kmers $H(k, substringk)$ containing all possible distinct kmers of RS that each have only one base difference from *substringk*. Similarly, *substringk*'s (k-1)-neighborhood is defined as the set of (k-1)mers $H((k-1), substringk)$ containing all possible distinct (k-1)mers of RS each of which is an immediate subset of *substringk*, and *substringk*'s (k+1)-neighborhood is defined as the set of (k+1)mers $H((k+1), substringk)$ containing all possible distinct (k+1)mers of RS each of which is an immediate superset of *substringk*.

For example, when the kmer is given as GTC and assume that all its proper supersets and subsets exist in RS, then its (k+1)-neighborhood $H(4, GTC) = \{\underline{A}GTC,\ \underline{T}GTC,\ \underline{C}GTC,\ \underline{G}GTC,\ G\underline{A}TC,\ G\underline{T}TC,\ G\underline{C}TC,\ G\underline{G}TC,\ GT\underline{A}C,\ GT\underline{T}C,\ GT\underline{C}C,\ GT\underline{G}C,\ GTC\underline{A},\ GTC\underline{T},\ GTC\underline{C},\ GTC\underline{G}\}$. Its (k-1)-neighborhood $H(2, GTC) = \{TC,\ GC,\ GT\}$. These three neighborhoods of kmer *substringk* can be combined and it is called a 3-layer kmer lattice structure of *substringk*. A schematic example of this lattice structure is shown in Figure 4.1. [1]

[1]The red * symbol represents a nucleotide A, G, T, or C.

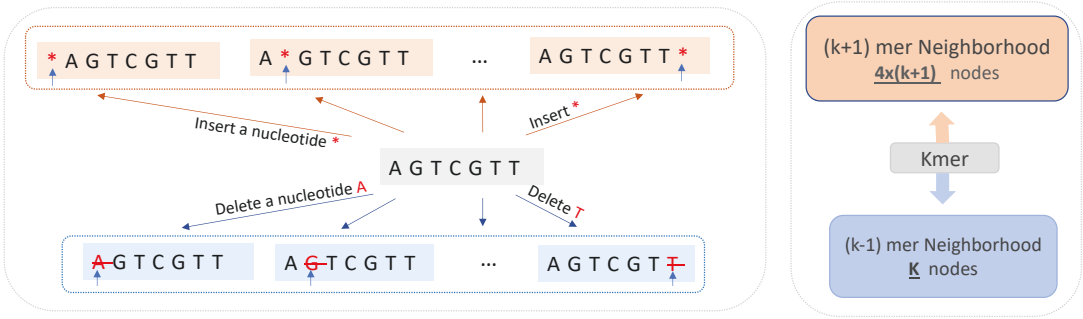

Figure 4.1: A 3-layer kmer lattice structure.

4.2.2 Error Correction Steps

The first step of the algorithm is to rectify substitution errors in RS. The algorithm traverses all of the distinct kmers. If a kmer $substringk$ has a frequency lower than a threshold τ (a small integer like 1, 2, or 3) and there exist at least one kmer in $substringk$'s k-neighborhood $H(k, substringk)$ whose frequency is larger than τ, we conjecture that $substringk$ contains a substitution error. We choose the kmer with the highest frequency in $H(k, substringk)$ as template to rectify the erroneous base in $substringk$. In the case where more than one kmer neighbors have the same high frequency, we choose the smallest kmer according to the alphabetical order as the template. After the change in $substringk$, those reads in RS containing the original $substringk$ are changed accordingly; some of them may become identical with other reads in RS. We introduce a double-checking technique to decide whether we eventually accept the correction — we double-check the updated frequencies of the distinct reads in the updated RS. Only when the corrected reads become identical with a read having a frequency higher than τ, we confirm the correction; Otherwise, we abandon the modification. With this double-checking strategy, we can avoid the issue of over-correction.

The second step is to rectify indel errors in the updated RS after the correction of substitution errors. The procedure is similar to correcting the

substitution errors. But the concept is fundamentally different. We traverse all of the distinct kmers in the updated RS. If a kmer $substringk$ has a frequency lower than a threshold τ and there exist at least one kmer in $substringk$'s (k-1)-neighborhood $H((k-1), substringk)$ whose frequency is larger than τ, we conjecture that $substringk$ contains an insertion error. On the other hand, if there exists at least one kmer in $substringk$'s (k+1)-neighborhood $H((k+1), substringk)$ whose frequency is larger than τ, we conjecture that $substringk$ contains a deletion error. We choose the kmer with the highest frequency in $H((k-1), substringk)$ or in $H((k+1), substringk)$ as template to rectify the insertion error in $substringk$ or to add the deleted base into $substringk$. After the change in $substringk$, those reads in RS containing the original $substringk$ are changed accordingly; some of them may become identical with other reads in RS. Again we use the double-checking strategy to decide whether we eventually accept the correction. We iterate these two steps by setting k from k_1 (usually 8) to k_{end} (usually 20 or 25). Setting a start k as 8 is because of that we find low-frequency kmers (e.g., frequency equal to 1) at this k but we cannot find such low-frequency ($< \tau$) kmers for k = 7. Starting from k = 8, we correct substitution errors first, then we perform the indel error correction, till k reaches k_{end}. Our method is named miREC built from a 3-layer kmer lattice structure for effective correction of miRNA sequencing errors especially those insertion and deletion errors. The pseudo code of our algorithm is shown in Algorithm 4.1.

Our miREC has been implemented as a software prototype. It provides several parameters for users to specify their tasks. Three most useful settings are: the error types, the frequency threshold τ, and the kmer range $[k_1, k_{end}]$. miREC has two running modes: one is for the substitution error correction only, the other is for the correction of both indel and substitution errors. Based on our experience, the frequency threshold τ is best recommended as 5 by default, and the kmer range parameter is set as $[8, 15]$. The higher frequency τ is set, the bigger number of bases might be considered as errors.

Algorithm 4.1 miRNA Sequencing Reads Error Correction

Input: A read set $\mathcal{RS} = \{r^1, r^2, \cdots, r^n\}$, a frequency border τ, a k value region (k_1, k_{end})

Output: A corrected read set $\mathcal{S}$

Function ERROR_CORRECTION $(\mathcal{RS}, (k_1, k_{end}), \tau)$

```
begin
    H_r[1...m] ← hash table ▷ *[r]read info H_a[1...m], H_b[1...m], H_c[1...m] ← hash table ▷ *[r]kmer info; H_.[i] is an array; Each element of H_.[i] is a tuple composed of sequence and its frequency for k = k_1 to k_end do
        H_a[1...m] to KMC3(RS, k)
        H_b[1...m] to KMC3(RS, k − 1)
        H_c[1...m] to KMC3(RS, k + 1)
        for i = 1 to n do
            (s, f) ← Count(r^i)
            Append (s, f) to H_r[s].f
        C_kmer ← ∅          ▷ *[r]the kmer with highest frequency for i = 1 to n do
            foreach kmer ∈ r^i do
                if H_a[kmer].f < τ then
                    C_kmer ← FindNeighbor(H_a)
                    C_r ← Replacekmer(C_kmer, r^i)
                    if H_r[C_r].f > τ then
                        r^i ← C_r
        for i = 1 to n do
            foreach kmer ∈ r^i do
                if H_a[kmer].f < τ then
                    C_kmer ← FindNeighbor(H_c, H_c)
                    C_r ← Replacekmer(C_kmer, r^i)
                    if H_r[C_r].f > τ then
                        r^i ← C_r
    S ← RS return S
```

Thus, users should be cautious about using a too large frequency threshold to avoid over-correction.

Every iterative step of miREC with the increasing length of kmer each time by 1 in the range $[k_1, k_{end}]$ actually corrects different amounts of errors. As shown in Figure 4.2, after five consecutive lengths of k are iterated, about 99.61% of substitution errors, 88.77% of insertion errors and 94.63% of deletion errors can be corrected on average over 12 wet-lab salmon datasets (Table 2) if k_1 is set as 8. With more loops of correction, more erroneous bases are detected and corrected. As each iterative loop consumes the same order of time complexity, users are suggested to narrow the kmer range (by setting k_{end} smaller) to shorten the program running time while correcting almost all of the errors for those miRNA sequencing datasets of huge size.

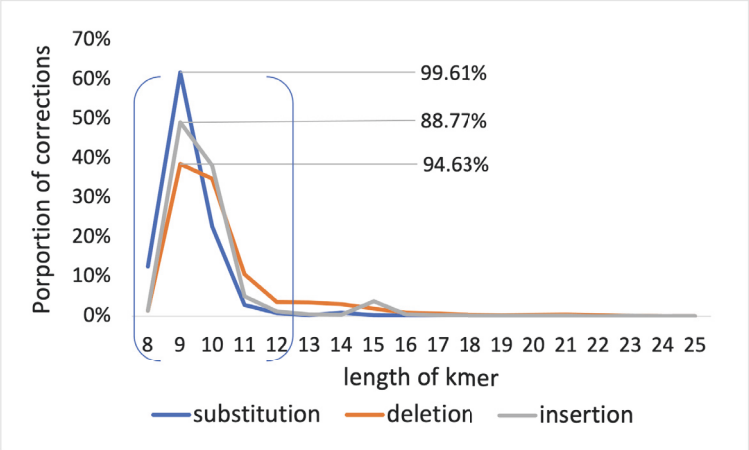

Figure 4.2: The numbers of corrected bases vary at different lengths of kmers.

The source codes of miREC are publicly available online at `https://github.com/XuanrZhang/miREC`.

4.3 Experiments and Results

In this section, we introduce data materials and related experiments used to verify our method's error correction ability. Our experiments includes

five parts. The first part is about the correction performance on the 8 simulated datasets; the second part is about correction performance on the wet-lab miRNA sequencing datasets after a small number of artificial errors are injected; the third part is about copy abundance recovery, entropy change and rectification site summary on the recently published salmon fish miRNA sequencing datasets after error correction; the fourth part provides detailed case studies on the change of isomiR families, tissue-specific isoforms, differentially expressed biomarkers and rare-miRNA quantity enhancement after the error correction on some of the wet-lab datasets, including the human miRNA sequencing datasets. The fifth part presents our verification results on sequencing reads datasets of 963 miRXplore Universal Reference miRNAs (three replicates) and their spike-in at eukaryotic cells.

4.3.1 Sequencing Read Datasets

To evaluate the performance of error correction methods, simulated datasets are required and the ground truth of the errors should be known. We introduce a novel process to generate simulated datasets that would have a close nature to wet-lab miRNA sequencing reads. We have two considerations in the process. One is to computationally replicate lab-verified miRNA sequences as templates to form the basic sequences of the simulated datasets, then we duplicate these basic sequences such that the copy counts of them follow a real distribution from a wet-lab dataset of miRNA sequencing reads. In fact, we replicated the mature miRNA sequences in miRBase (Kozomara, Birgaoanu & Griffiths-Jones 2019) as the templates, and made the copy count distribution of these template sequences to follow the distribution drawn from a typical miRNA dataset under accession number SRR866573. In other words, the sequences in our simulated datasets are not random sequences (they are real lab-verified miRNA sequences); their copy count distribution is not random either. Then we injected random errors into the simulated datasets under an error rate of 0.1% per base (Laehnemann et al. 2016). Specifically, we randomly selected two reads from every 100

reads in the dataset; then for each selected read, we injected an erroneous base (substitution, deletion or insertion) randomly at any position of the read. We recorded all of these randomly and purposely injected errors for performance evaluation.

Considering that some existing methods only support substitution error correction, we synthesized 8 simulated datasets: four datasets containing substitution errors only (denoted as D_sub1, D_sub2, D_sub3 and D_sub4), and four datasets containing a mixture of 80% substitution and 20% indel errors (denoted as D_mix1, D_mix2, D_mix3 and D_mix4). More details of the simulated datasets are shown in Table 4.1.

Table 4.1: Description of our simulated datasets

		ID	Total erroneous bases	Per read error rate
Simulated Datasets	subs only	D_sub1	3071	3.03%
		D_sub2	3022	2.98%
		D_sub3	2973	2.93%
		D_sub4	3124	3.08%
	mix errors	D_mix1	1602,213,211	2.00%
		D_mix2	1618,188,206	1.98%
		D_mix3	1598,184,177	1.93%
		D_mix4	1625,226,217	2.04%

Notes: '_sub' means datasets contain substitution errors only and '_mix' means datasets contain both substitution and indel errors. Total erroneous bases list substitution, insertion and deletion errors respectively.

Wet-lab miRNA sequencing datasets for our performance evaluation are all downloaded from the Sequence Read Archive (SRA) (`https://www.ncbi.nlm.nih.gov/sra/`) under the accession numbers SRP022967, SRP296813 and SRP288246. These datasets have been originally studied for topics related to salmon fish miRNAs (Woldemariam, Agafonov, Høyheim,

Houston, Taggart & Andreassen 2019, Andreassen, Worren & Høyheim 2013), human beta cells, or Alzheimer's disease.

Table 4.2: Description of twelve wet-lab salmon miRNA sequencing datasets and four human miRNA datasets.

Tissue	Total reads	Unique reads	Accession ID
Liver	1,446,902	64,593	SRR866573
Liver	1,647,133	75,273	SRR866579
Spleen	8,597,057	295,940	SRR866583
Spleen	2,236,013	89,165	SRR866587
Kidney	10,065,660	243,430	SRR866589
Head kidney	7,375,957	246,444	SRR866590
Heart	2,812,993	118,366	SRR866605
Brain	6,331,448	132,558	SRR866611
Intestine	12,428,822	197,094	SRR866612
White muscle	5,972,384	142,444	SRR866613
Gills	6,240,735	132,038	SRR866614
One day old individual	18,041,561	172,048	SRR866615
human beta cells datasets			
In low glucose	63,008,516	5,803,166	SRR13208981
In high glucose	33,444,257	1,856,318	SRR13208980
human brain datasets			
Sample of aged 75	11,849,807	635,169	SRR12881030
Sample of aged 94	17,250,812	361,039	SRR12881018

This work used 12 salmon miRNA sequencing datasets which were acquired from particular salmon tissues, including liver, spleen, kidney, heart, brain, etc; and used two human beta-cell miRNA sequencing datasets which are about miRNA expression comparison between those cells incubated with a solution of low glucose (2 mM) and those with a high glucose (20 mM) in extracellular vesicles. The two other human miRNA sequencing datasets

analyzed here are about brain samples related to post-mortem Alzheimer's disease. One is from a male patient aged 75, the other is from a male patient aged 94. All the reads in the above datasets contain the sequences of adaptors; we used the cutadapt tool (Martin 2011) to remove the adaptors before our error correction. More details of these cleaned datasets are shown in Table 4.2.

To rigorously evaluate the error correction performance, we also randomly and purposely inject a small number of errors into these wet-lab datasets, rather than the simulated datasets, to see whether our algorithm can detect and correct these errors with ground truth, together with other errors without ground truth. Only when all of these artificial errors in the real-life miRNA sequencing reads can be detected and corrected, the corrections on the other bases (without ground truth) can be highly trustable. This small number of artificial errors constitutes only 0.5% of total corrections in each dataset to avoid changing the original nature of the data. We have done these for three salmon datasets (liver, heart and spleen tissues), and one human brain miRNA dataset from the male patient aged 75. For each of these datasets, we randomly injected small numbers of errors twice.

4.3.2 Evaluation Metrics

As the ground truth of the errors in the simulated datasets are known, we can use recall, precision and gain to compare the correction performance between different methods. On the wet-lab miRNA sequencing datasets, we measure the copy count changes of the reads, the entropy changes of the whole set of reads, and locations of the rectifications to understand the importance of error correction. There is no recall or precision performance on the wet-lab miRNA sequencing datasets, because the ground truth of error distributions is unknown.

Performance evaluation metrics on the simulated miRNA sequencing datasets

To assess the accuracy of the correction methods, we use the following three metrics, Precision (*TP/(TP+FP)*), Recall (*TP/(TP+FN)*), Gain (*(TP-FP)/(TP+FN)*). More details are shown in Section 2.3.1.

Metrics used for performance evaluation on wet-lab miRNA sequencing datasets

We examine the changes of miRNA copy counts and dataset entropy changes before and after error correction for multiple salmon fish miRNA sequencing datasets. Besides, we also summarize the position information of the corrections in the reads to record the proportion of corrections in the seed region. The concept of entropy was first introduced by a physicist Rudolf clausius (Clausius 1879), to measure a system's thermal energy per unit temperature. The amount of entropy is also a measure of the molecular disorder or randomness of a system. Here, we regard our sequencing read dataset as a system, and use the concept of entropy to define dataset entropy. For dataset entropy, we combine all low-frequency reads which are more likely to contain errors, to interpret the quality of datasets. More specificly, we define the miRNA count, dataset entropy and errors in the seed region as follows.

- miRNA count: the copy count of miRNA appearing in the datasets, which is corresponding to miRNA expression level or miRNA abundance.

- Dataset entropy: $-\sum_{i=1}^{n} p_i \cdot \log p_i$, where p_i is the proportion of reads whose frequency is small than i. We calculate the entropy for low-frequency reads and sum up to interpret the degree of disorder in the read dataset. When the entropy turns to be small, it means the certainty of the miRNA expression becomes higher.

- Errors in seed region: erroneous bases in the seed region, which is a conserved sub-sequence of miRNA (mostly situated at positions 2-8). Precise bases in the seed region are vital since the seed sequence must be perfectly complementary with its target mRNA to make the miRNAs function.

4.3.3 Performance Evaluation

Our analysis and results are presented in four main parts. The first part is about the correction performance on the 8 simulated datasets; the second part is about correction performance on the wet-lab miRNA sequencing datasets after a small number of random errors are injected; the third part is about expression abundance recovery, entropy change and rectification site summary on the recently published salmon fish miRNA sequencing datasets after error correction; the fourth part provides detailed case studies on the change of isomiR families, tissue-specific isoforms, differentially expressed biomarkers and rare-miRNA quantity enhancement after the error correction on some of the wet-lab datasets, including the human miRNA sequencing datasets.

Gain, recall and precision performance on the simulated miRNA sequencing datasets

The correction performance of our miREC is presented in Table 4.3 in comparison with algorithms Karect (Allam et al. 2015), Coral (Salmela & Schröder 2011), BFC (Li 2015), Rcorrector (Song & Florea 2015) and Bcool (Limasset et al. 2020). Coral and Karect are multi-alignment based error correction methods. BFC is a representative of the kmer based error correction methods. BFC requires a prior-setting of the k parameter; the best k in this work is 21 (namely, under other k settings, BFC did not exceed the performance of when k = 21). Karect is one of a few correction tools which supports the correction of indel errors. Rcorrector, a RNA reads error correction method, has a performance higher than another RNA correction

method Seecer (Le et al. 2013). Rcorrector also needs to set the k parameter and the best k in this work is 17. Even using the optimal k settings, only a few bases can be corrected by Rcorrector. A very recent error correction algorithm Bcool (Limasset et al. 2020), which uses a de Bruijn graph as the platform to correct errors, could not detect any errors in the simulated datasets. This surprising performance is not included in the table.

Our method miREC has excelled in the correction performance:

- It did not introduce any new error, namely, it achieved the same gain and recall rates on all of the 8 datasets;
- It detected and corrected almost all of the errors including the indel errors; the recall rate ranges between 96.0% - 97.9%; the precision ranges between 98.6% - 99.5%;
- It improved the overall data quality remarkably: (1) from every 50 reads containing one error to every 1300 reads containing one error for the four error-mixed datasets; and (2) improved the data quality from every 30 reads containing one error to every 1650 reads containing one error for the four substitution-only datasets.

The average recall and gain rate of miREC are much superior to Karect (the second-best method) respectively by 3.28% and 3.66% on the four substitution-only datasets, and by 19.44% and 28.7% on the four error-mixed datasets. Specifically, the average recall rates of miREC are 97.83% and 96.12% on the four D_sub datasets and on the four D_mix datasets respectively, which are 16.25%, 87.96% and 3.28% (on the D_sub datasets) and 27.86%, 87.21% and 19.44% (on the D_mix datasets) better than BFC, Coral and Karect. This implies that there are lots of errors un-detected by these baseline methods meanwhile these introduced a lot of new errors (gains and recall not equal). The multi-alignment method performed worst on these miRNA datasets. A possible reason is that the alignment strategy could not differentiate miRNA reads well due to the short length of miRNAs.

Table 4.3: Outstanding error correction performance by our miREC in comparison with the best available tools

	Gain(%)					Recall(%)					Precision(%)				
	miREC	BFC	Cor	Kar	Rcor	miREC	BFC	Cor	Kar	Rcor	miREC	BFC	Cor	Kar	Rcor
D_s1	**97.88**	69.85	5.44	94.2	5.99	**97.88**	70.63	9.96	94.43	5.99	99.64	82.07	67.4	**99.76**	92
D_s2	**97.98**	84.98	4.8	94.44	5.43	**97.98**	85.7	9.46	94.87	5.43	99.5	99.08	65.75	**99.55**	91.62
D_s3	**97.61**	83.99	5.01	94.01	5.85	**97.61**	84.76	9.72	94.55	5.85	99.42	99.06	66.9	**99.43**	91.1
D_s4	**97.86**	84.51	5.83	94.01	5.19	**97.86**	85.21	10.34	94.37	5.19	99.48	99.07	68.87	**99.63**	90
AVE	**97.83**	80.83	5.27	94.17	5.62	**97.83**	81.58	9.87	94.55	5.62	99.51	94.82	67.23	**99.59**	91.18
D_m1	**95.96**	65.93	2.22	65.98	0.05	**95.96**	67.11	9.22	75.64	0.05	**98.78**	95.78	55.16	88.67	100
D_m2	**95.73**	68.24	1.04	66.95	0.1	**95.73**	69.43	8.2	77.53	0.1	**98.67**	96.28	52.05	87.99	100
D_m3	**96.02**	68.56	1.43	71.47	0.05	**96.02**	69.78	8.88	78.61	0.05	**98.58**	96.68	53.05	91.67	100
D_m4	**96.76**	65.57	2.47	65.28	0	**96.76**	66.73	9.33	74.95	0	**99.12**	96.17	57.27	88.57	0
AVE	**96.12**	67.07	1.79	67.42	0.05	**96.12**	68.26	8.91	76.68	0.05	**98.79**	96.23	54.38	89.22	75

Notes: AVE indicates the average score over the four datasets. Bold font indicates the best result in the row. Cor, Kar and Rcor stand for the Coral method, the Karect method and the Rcorrect method respectively. D_s indicates datasets containing only substitution errors, while D_m indicates datasets containing mixed substitution, insertion and deletion errors. The underline 100 precision of Rcor on D_m1, D_m2 and D_m3 stands for only one, two and one base is corrected respectively.

Rcorrector had a very low recall and gain performance as well, that means most of the errors were not detected by the method.

The performance of miREC is robust across all the 8 datasets including the four mixed-error datasets, in contrast to the baseline methods which exhibited a poor performance on the detection and correction of the indel errors. The gain rate of BFC drops from 80.83% (on the four D_sub datasets) to only 67.07%(on the four D_mix datasets), and the gain of Coral drops from 5.27% to 1.79%. It suggests that the performance of these methods on the substitution error correction was interrupted and affected by the addition of the indel errors into the datasets. As real-life wet-lab sequencing reads more or less company with a small amount of indel errors, our miREC provides an unalterable advantage over the baseline methods for the correction of all types of aberrations.

Correction performance on wet-lab miRNA sequencing datasets injected with small numbers of random known errors

We made 27 random base modifications (total 21 substitutions, 3 insertions and 3 deletions) at the salmon liver miRNA sequencing dataset (SRR866573). These random modifications introduced/injected 18 *genuine* errors into the dataset, where a random base modification is *not* considered as a genuine error if its correspondingly modified read becomes identical with another read having a high frequency (i.e., copy count > 5).

Our algorithm corrected all of these 18 genuine errors (100% recall). For example, the read @SRR866573.64765 (TGCGGACCAGGGGAATCCGACT) had a random deletion at position 5, becoming TGCGACCAGGGGAATCCGACT; our miREC detected this error and restored it to its original base. As another example, the read @SRR866573.212344 (AAGCTGCCAGCTGAAGAACTG) had a random substitution from C to G at position 8, becoming AAGCTGCGAGCTGAAGAACTG; our miREC corrected it successfully. The read @SRR866573.1103128 (AAGCGGGCCCCCAAACTTCTGT)

had a random insertion of G at position 16, becoming AAGCGGGCCCCCAAAGCTTCTGT; again, our miREC successfully detected this error and corrected it. For the remaining 9 randomly injected base modifications, they did not cause genuine errors because each of their reads was transformed into another read that has a high copy count in the same dataset. For example, the read @SRR866573.360151 (ATGACCTATGAATTGACAGCCT) had a random substitution from T to C at position 21 (the last position). With this modification, the read becomes another read ATGACCTATGAATTGACAGCCC which has 156 copies. This modification was unable to be restored to its original base because every kmer in ATGACCTATGAATTGACAGCCC was highly frequent (at least 156 copies), namely, containing no error. Note that this modification should not be restored to ensure no over-correction would happen in practice, otherwise the correction would be guilty. For performance comparison, the second-best method Karect was applied to the same error-injected salmon liver dataset, but it corrected only 5 of the 18 genuine errors.

We repeated this test with another round of random base modifications at SRR866573 (total 28 modifications including 20 substitutions, 6 insertions and 2 deletions). Our miREC detected and corrected all of the 20 genuine errors (100% recall again). In comparison, Karect corrected only 9 of them.

Similarly, our miREC corrected all of the genuine errors caused by small numbers of random base modifications at other wet-lab miRNA sequencing datasets (40 substitutions, 3 insertions and 6 deletions; or second round 45 substitutions, 7 insertions and 7 deletions at the salmon heart dataset. 38 substitutions, 4 insertions and 4 deletions; or second round 43 substitutions, 6 insertions and 6 deletions at the salmon spleen dataset). However, Karect corrected only 8 of the 27 genuine errors or only 8 of the 35 errors on these two error-injected salmon heart datasets, and had similar performance on the two error-injected salmon spleen datasets.

On the two human brain datasets, our miREC achieved the same perfect

performance (100% recall) to correct all of the genuine errors caused by small numbers of random base modifications (about 300 base modifications which had resulted in 130 and 120 genuine errors). However, Karect could only fix 12 or 20 genuine errors in these two datasets. Our source codes for the random error injection into wet-lab miRNA sequencing datasets and more detailed correction results are available at github link `https://github.com/XuanrZhang/miREC`.

Changes in isoform abundance, whole set entropy and base positions after error correction at the salmon fish miRNA sequencing reads

The perfect recall performance on the small numbers of errors injected into wet-lab miRNA sequencing datasets and the excellent gain performance on the simulated datasets are strong combined evidence to convince our correction results on wet-lab datasets where the ground truth of errors are not available. The salmon liver miRNA sequencing dataset (SRR866573) has a total of 900,814 reads, containing 32,972 distinct reads before error correction. After error correction by our miREC, there are only 27,299 distinct reads some of which gained plenty of abundance. In other words, most of the error-contained reads were corrected and turned to be identical with some other reads, making the abundance merging meanwhile the disappearance of the originally error-contained reads. See Figure 4.3 for an average percentages of the distinct miRNAs over the 12 datasets that have a high- or low-level abundance recovery. There are around 47.3% of the distinct miRNAs whose copy counts have increased by more than 10% after the corrections, in particular, about 5.5% of the distinct miRNAs have obtained above 50% abundance increase. These corrections are useful to draw more reliable conclusions about miRNA discovery or isomiR classification or tissue-specific biomarker discovery (case studies presented later).

The abundance recovery of the miRNA isoforms after error rectification in a dataset implies that the numbers of distinct reads are decreased as reported

above. We present Figure 4.4 to illustrate the overall entropy change of every entire dataset before and after the error correction to quantify this point. On average the entropy of the 12 datasets is shrank by 15.11% when the parameter k of miREC ranges from 8 to 20, and the entropy score decreased by 14.51% when k ranges from 8 to 25. These entropy declines (with slight variance) in the 12 datasets theoretically mean that the certainty of the miRNA expression level is greatly improved. In other words, our miREC can enhance the data quality in the perspective of a lower entropy or a higher certainty.

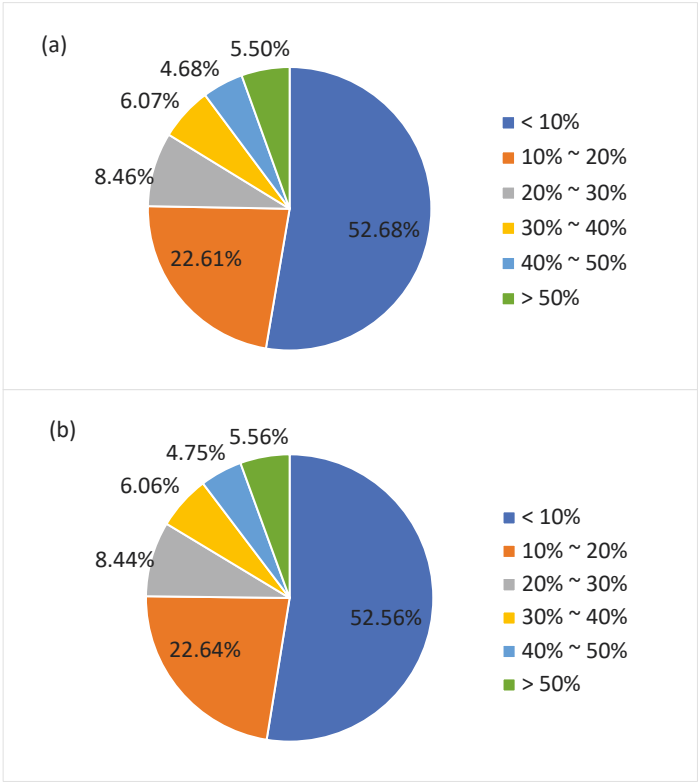

Figure 4.3: Proportions of unique-read count are changed compared with uncorrected data in average of 12 salmon datasets. (a) The miREC runs with continuous k value from 8 to 20. (b) The miREC runs with continuous k value from 8 to 25.

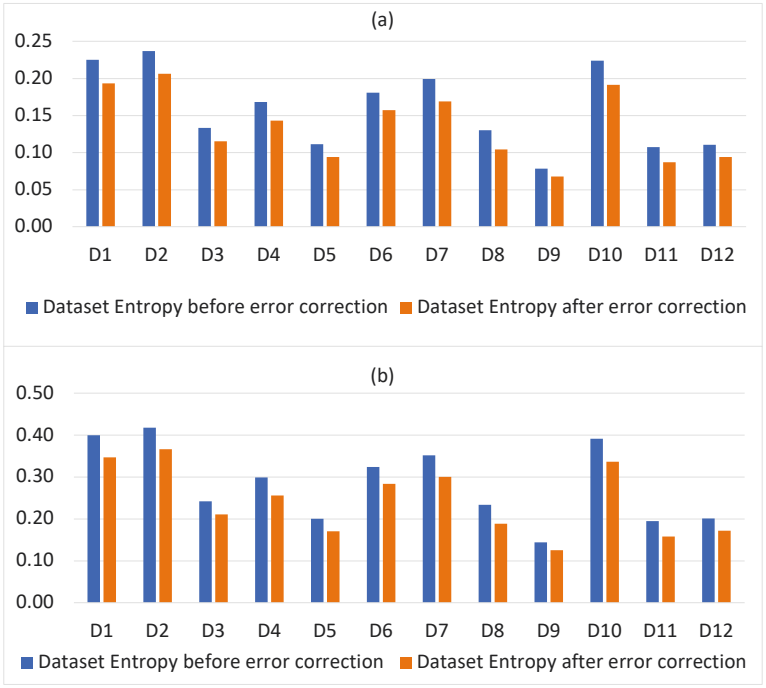

Figure 4.4: Dataset entropy changes before and after the error correction by miREC on the 12 salmon miRNA datasets. (a) when the continuous k settings from 8 to 20; (b) when the continuous k settings from 8 to 25.

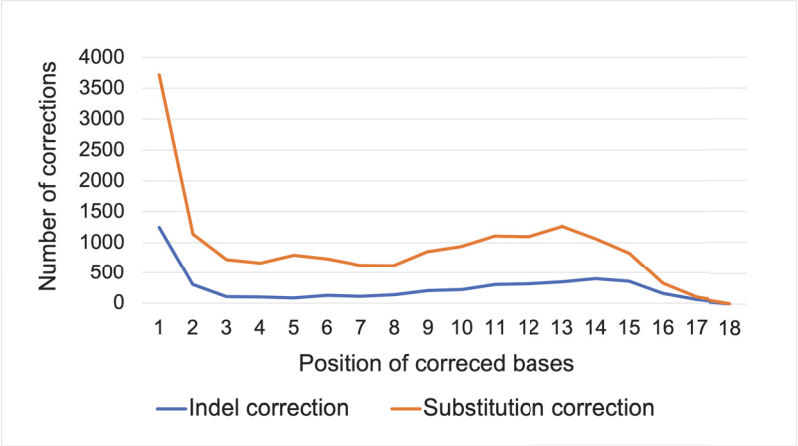

Figure 4.5: The distributions of correction positions.

We found that the aberrations could occur at every base position of the miRNAs. But, one-third of the errors are detected and corrected at the seed region of the miRNAs (Figure 4.5). These corrections at the seed region provide great benefits for miRNAs' target prediction analysis. There are also a high percentage of the indel or substitution corrections at position 1 which is a base position very sensitive to the definition of trimmed or addition isomiRs.

4.3.4 Case Studies

In this section, we demonstrate case studies related to isomiR families, tissue-specific miRNAs and rare-miRNA quantity recovery. We show examples of miRNAs whose copy counts have changed a lot after error correction. We also show examples of tissue-specific miRNAs after error correction, and describe the change in the ranking lists of differentially expressed miRNAs.

Case study 1: big abundance recovery.

In the salmon heart dataset, a read *TTGGTCCCCTTCAACCAGCTGTAAT* (mapped to miR-133a-1 in miRBase (Kozomara et al. 2019)) had 10 copies. Our miREC detected 13 erroneous reads related to this miRNA. Eight substitution errors happened at position 24 base A (sequenced to G or T), and five happened at position 25 base T (sequenced to A or G). After our correction, the abundance level of miR-133a-1 increased from 10 to 23, a 130% expression recovery. Other two miRNAs (ssa-miR-133a-3p and ssa-miR-133a-5p) from the same miRNA family also recovered their expression abundance. See the copy counts and change details in Table 4.4. We note that currently annotated functions of miR-133a-1 are related to conventional central osteosarcoma and heart conduction disease (Stelzer, Rosen, Plaschkes, Zimmerman, Twik, Fishilevich, Stein, Nudel, Lieder, Mazor et al. 2016, Andreassen et al. 2013). With the refined expression understanding, its functions can be re-examined more deeply.

Another example in Table 4.5, read *ATCCCGGACGAGCCCCCAA*, had 18 copies and its abundance increased to 31 after miREC correction. The aberrations include four deletion errors at position 1 (base A deleted), four substitution errors at position 19 base A (sequenced to C) and two insertion errors at position 1 and 2 (base A inserted). This error distribution implies that the sequencing mistakes can occur at multiple base positions with multiple times; and that our miREC is capable of correctly detecting these errors and performing accurate corrections.

For comparison, we tested the second-best method Karect on this salmon heart dataset to see whether the same mistakes could be corrected. Take the cases in Table 4.4 as example, only three erroneous reads of the first read *TTGGTCCCCTTCAACCAGCTGTAAT* were detected by Karect (we detected 13); none of the erroneous reads of the other two reads in the table were detected. Only one of the four related erroneous bases was corrected by Karect, while all of the related erroneous bases were corrected by our method.

Case study 2: miRNA isoforms and editing events.

Editing events and isoform variations at the cleavage sites can cause slight but important difference in many miRNA sequences (Martí, Pantano, Bañez-Coronel, Llorens, Miñones-Moyano, Porta, Sumoy, Ferrer & Estivill 2010). In the salmon fish heart miRNA sequencing dataset (SRR866605), canonical miRNA read *ATCCCGGACGAGCCCCCAA* co-exists with five isoforms having copy counts 9, 1489, 16, 4 or 4; There are also three singleton reads having an editing distance with this canonical miRNA (Table 4.5). Our miREC grouped all of these reads and detected some of them as erroneous reads. After error correction, the abundance of the canonical miRNA increased from 18 copies to 31; the first three isoforms' abundance increased from 9 to 17, from 16 to 20 and from 1489 to 1513. The abundance recovery of the canonical miRNA is owned to the erroneous base correction of the 11 reads listed in the last five rows of Table 4.5.

Table 4.4: Changes in the read counts of some miRNAs

Sequence	Read Count Before Correction	Read Count After Correction	Abundance Change Percentage	isoforms/editing events
TTGGTCCCCTTCAACCAGCTGTAAT	10	23	130.00%	
TTGGTCCCCTTCAACCAGCTGTA-T	44	45	2.27%	5' deletion
TTGGTCCCCTTCAACCAGCTGTAA-	107	111	3.74%	3' deletion
Related Erroneous Reads List				
TTGGTCCCCTTCAACCAGCTGTAGT	4	0	error removal	substitution error
TTGGTCCCCTTCAACCAGCTGTATT	4	0	error removal	substitution error
TTGGTCCCCTTCAACCAGCTGTAAA	3	0	error removal	substitution error
TTGGTCCCCTTCAACCAGCTGTAAG	2	0	error removal	substitution error

The performance by the Karect method shows that only one of the eleven erroneous reads was corrected. Only the first and second miRNA sequence (Table 4.5) have different read counts after Karect's correction. The copy count of the first read was increased by 1 and the copy count of the second read was increased by 20, missing lots of corrections. A more interesting part of the error correction is that the 11 erroneous reads of the canonical miRNA contain not only substitutions, but deletion and insertion errors distributed at multiple base positions. In particular, more than one third of erroneous bases happened at the seed region, important for gene target binding analysis.

Case study 3: upside down change in differential expression analysis.

Analysis on tissue-specific uniquely expressed or top-ranked differentially expressed miRNAs in a specific tissue or at a disease stage is very sensitive to the sequencing data quality (Telonis et al. 2017). Some uniquely expressed miRNAs can be identified only after error correction.

In our differential expression analysis between the salmon heart and brain tissues (SRR866605 vs SRR866611), we found that $5,675$ miRNA did not co-exist in the two datasets, and the number of common miRNAs was reduced from $16,443$ to $10,768$ after error correction. For example, a read *TGAGGTAGTTGGTTGTATGGTG* (mapped to ssa-let-7d-5p in miRBase), had 4 copies in the heart dataset and 26 copies in the brain dataset before correction, while the number of its copies changed to zero in the heart dataset and changed to 30 in the brain dataset after error correction. Two more examples: A read *CTTTCAGTCGGATGTTTGCACCA* (mapped to ssa-miR-30d-3p in miRBase) had 152 copies in the heart dataset and 2 copies in the brain data before correction, while its quantity was changed to 155 in the heart dataset and to zero in the brain dataset. Another read *TTGCATAGTCACAAAAATGATC* (mapped to ssa-miR-153a-3p in miRBase) had 3 copies in the heart dataset and $14,434$ copies in the brain dataset before correction, while the quantity dropped to zero in the heart

Table 4.5: isomiRNA detection

Sequence	Read Count Before Correction	Read Count After Correction	Abundance Change Percentage	isoforms/editing events
ATCCCGGACGAGCCCCCAA	18	31	72.22%	
ATCCCGGACGAGCCCCCA-	1489	1513	1.61%	deletion
ATCCCGGACGAGCCCCCAT	16	20	25.00%	substitution
ATCCCGGACGAGCCCCCAAA	9	17	88.89%	insertion
Related Erroneous Reads List				
-TCCCGGACGAGCCCCCAA	4	0	error removal	deletion error
ATCCCGGACGAGCCCCCAC	4	0	error removal	substitution error
AATCCCGGACGAGCCCCCAA	1	0	error removal	insertion error
ACCCCGGACGAGCCCCCAA	1	0	error removal	substitution error
AATCCCGGACGAGCCCCCAA	1	0	error removal	insertion error

Table 4.6: Rank changes of the top-10 common miRNAs in salmon heart and brain tissues after error correction

miRNA sequence	After_rank	Before_rank
TCTTTGGTTATCTAGCTGTATG	1	2
TCTTTGGTTATCTAGCTGTAT	2	3
TTTGTTCGTTCGGCTCGCGTT	3	5
TCTTTGGTTATCTAGCTGTA	4	8
TTGCATAGTCACAAAAGTGATC	5	6
TCTTTGGTTATCTAGCTGTATGA	6	7
TGGAAGACTAGTGATTTTGTTG	7	10
TAAAGCTAGAGAACCGAATGTA	8	11
TAAGGCACGCGGTGAATGCC	9	12
ATGGCACTGGTAGAATTCACT	10	13

Notes: After_rank indicates the rank after error correction, while Before_rank indicates the rank before error correction

dataset but increased to 14,498 in the brain dataset after error correction.

Top-rank differentially expressed miRNAs can become low-ranked ones, and vice versa after error correction. The reason is that the expression folds of miRNAs between two tissue types or between two disease stages are sensitive to the copy counts after erroneous reads are corrected in the two classes. We compared the expression folds of common miRNAs between the salmon heart tissue and brain tissue before and after our error correction. Table 4.6 presents the list of 10 miRNAs whose expression folds between the two tissues are top-ranked after the error correction, in comparison with their ranking positions before the error correction. The two ranking lists are quite different. For example, the rank of ssa-miR-9a-5p (*TCTTTGGTTATCTAGCTGTA*) is reverted from rank 8 to 4. Furthermore, the originally top-ranked number-1, number-4 and number-9 miRNAs are all dropped below rank-10 after error correction. In

detail, the original top-one miRNA (*TCTTTGGTTATCTAGCTGTATGT*) had 16,776 copies in the brain tissue. However, the corrected top-one miRNA is *TCTTTGGTTATCTAGCTGTATG*, whose copy count is 48,092 in the brain tissue after error correction. It is interesting to note that:

- The two miRNAs only have one base difference at the 3' end. The corrected top-one miRNA after error correction has one base trimmed at the 3' end, compared to the original top-one ranked miRNA. The two miRNAs can be recognized as 3' end trimmed/addition isoforms each other.
- The original top-ranked miRNA and the corrected top-one miRNA have a huge abundance difference (31316 copies = 48092 − 16776) in the brain tissue. One is extremely high-level expressed; the other is median-level expressed. This suggests that we would concentrate on wrong top-ranked miRNA biomarkers if the sequencing reads had not been cleaned by the error correction algorithms.

New top-ranked tissue-specific miRNAs (or called no-presence miRNAs or tissue- and disease-subtype dependent miRNAs by (Telonis et al. 2017)) were found in the heart tissue (SRR866605) after error correction when the liver tissue (SRR866579) was compared. Table 4.7 presents two rankings of top-15 miRNAs specifically expressed in the heart tissue before and after error correction. Without our correction, the top-1, top-10 and top-13 tissue-specific miRNAs in salmon heart would be not detected because erroneous reads which are identical with these reads also exist in the liver tissue. Moreover, after our error correction, the quantity of the top-ranked miRNAs increases. These recovered expression levels and accurate abundance measurement would make more convincing conclusions in the down stream analysis.

Table 4.7: Ranking position change of tissue-specific miRNAs in the heart tissue (vs the liver tissue) before and after error correction

miRNA Sequence	Rank After Correction	Rank Before Correction	Read Count Before Correction	Read Count After Correction	Read Count Increase
TTAAGACTTGTAGTGATGTTT	1	out of scope	47546	47583	37
TGGAATGTAAAGAAGTATGTAT	2	1	12650	12728	78
TTTGGTCCCCTTCAACCAGCTG	3	2	4954	4985	31
TTGGTCCCCTTCAACCAGCTG	4	3	2522	2541	19
TTAAGACTTGCAGTGATGTT	5	4	1665	1677	12
ACAGCTCATCCATTGGTC	6	5	1174	1188	14
TGGAATGTAAAGAAGTATGTA	7	6	879	892	13
AACATCACTTTAAGTCTCTGCT	8	7	876	892	16
TTGGTCCCCTTCAACCAGCTGTA	9	8	835	856	21
TGAGGTAGTTGGTTGTATTGTTT	10	out of scope	780	791	11
TGGACGGAGAACTGATAAGGG	11	9	693	702	9
TTAAGACTTGTAGTGATGTTTAA	12	10	685	698	13
TGAGGTAGTTGGTTGTATTGT	13	out of scope	659	666	7
TAAAGGGAATTTGCGACTGTTA	14	11	622	635	13
TGGAATGTAAAGAAGTATGTATT	15	12	616	629	13

Case study 4: Class-specific miRNAs and rare-miRNA analysis for human miRNA sequencing datasets.

Ranking positions of class-specific miRNAs and rare miRNA quantity recovery analysis are also conducted for human miRNA sequencing datasets (acquired from beta cells and brain samples). The human beta cells were incubated with solution of low glucose or high glucose. It's expected to reveal novel differentially expressed miRNAs between these two classes. We found that the number of distinct reads decreased by 8.85% from 5,803,166 to 5,289,466 in the low glucose solution cell, and reduced by 12.44% from 1,856,318 to 1,625,453 in the high glucose solution cell after error correction. For the top-ranked differentially expressed miRNAs between the two datasets, only slight rank changes were observed (Table 4.8). Some of the top-ranked miRNAs were just swapped ranking positions within top 10 after error correction. The copy counts of these top-ranked miRNAs all had small increases after the error correction. Note that these changes on glucose-level specific miRNAs in human beta cells after error correction is not as big as those changes made in the salmon heart-head tissue pair comparison by our error correction. However, such big changes on age-specific miRNAs in brain samples can be observed again when we compared between miRNA

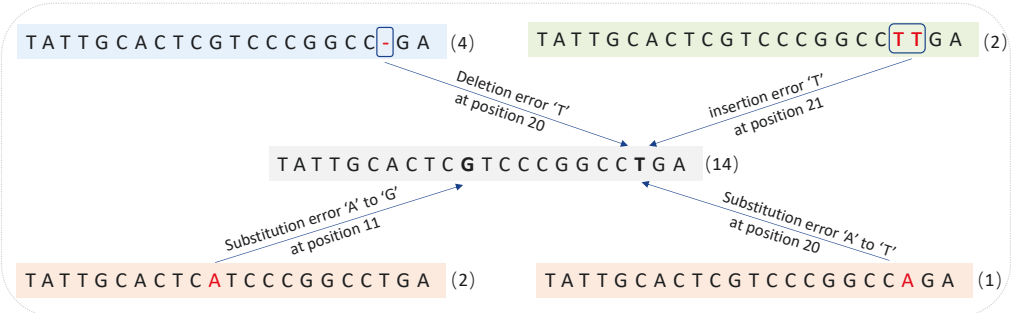

Figure 4.6: A rare miRNA in the Alzheimer's disease patient aged 94 showing significant copy count change after error correction.

Table 4.8: Copy count change and ranking position change of top-10 differentially expressed miRNAs in the high glucose incubated human beta cell after error correction, and those in the low glucose incubated human beta cell after error correction.

miRNA Sequence **among high glucose**	Rank After	Rank Before	Read Count Before	Read Count After	Read Count Increase
GTGGGGCCACGAGCTGAGTGCGT	1	1	86	92	6
AGCAGGGTCGGGCCTGGTTAGTA	2	2	68	69	1
GAGTTCGCGCTTTCCCCT	**4**	**3**	61	69	8
GCCGCAGGTGCAGATCTTGGTGG	**3**	**4**	62	65	3
TCGGGCCTGGTTAGTACTTGGAT	5	5	60	62	2
GTGGAGCCTGCGGCTTAAT	6	6	60	61	1
TCGGAAGCTAAGCAGGGTCGGGCC	7	7	57	57	0
GGCTCAGCGTGTGCCTACC	8	9	55	56	1
GTCTACGGCCCTACCACCCTGAACG	9	8	54	55	1
GTCGGGCCTGGTTAGTACTTGGA	10	10	51	54	3
miRNA Sequence **among low glucose**	Rank After	Rank Before	Read Count Before	Read Count After	Read Count Increase
CGCCCGTCCCCGCCCCTT	1	1	640	651	11
TAGGGGTATGATTCTCGCTTCGG	2	2	582	586	4
GCCCGTCCCCGCCCCTT	3	3	565	572	7
CGCCCGTCCCCGCCCCTTGCC	4	4	484	489	5
CCAGTGGTTGTCGACTTGCG	5	5	428	436	8
CTCAGGTGCCCGAGGCCGAA	6	6	371	376	5
AAGACGGAGAGGGAAAGAG	7	7	317	324	7
ACGGGGAGGGCGGCGCCGCCGCC	**8**	**9**	292	300	8
TTCGGCTGAGTTCGTGATGGATTTG	**9**	**8**	297	298	1
CCACCGCCCGTCCCCGCCCCTTG	10	10	272	278	6

Table 4.9: Ranking position changes of age-specific miRNAs in brain tissue from an Alzheimer male patient aged 94 (vs a patient aged 75) before and after error correction

miRNA Sequence	Rank After Correction	Rank Before Correction	Read_C Be_Correction	Read_C Af_Correction	Read_C Increase	Original Read_C(aged 75)
TTTCTCACTACTGCACTTGACTAGTC	1	out of scope	1416	1477	61	3
TTTCTCACTACTGCACTTGACC	2	out of scope	839	877	38	2
TTTCTCACTACTGCACTTGAC	3	out of scope	817	857	40	2
TTTCTCACTACTGCACTTGACTAG	4	out of scope	726	761	35	3
TTTCTCACTACTGCACTTGACA	5	1	719	746	27	0
TTTCTCACTACTGCACTTGACTAGT	6	out of scope	593	628	35	4
TGAGGTAGTACGTTGTATAGT	7	2	521	533	12	0
TTTCTCACTACTGCACTTGACTA	8	out of scope	381	410	29	2
TGAGGAAGTAGGTTGTAGAGTT	9	3	365	375	10	0
TGAGGTAGTACATTGTATAGT	10	out of scope	339	352	13	4

Notes: Read_C indicates read count, Be_Correction means before correction and Af_Correction means after correction.

sequencing reads of an Alzheimer's disease patient aged 75 and a patient aged 94. The number of distinct miRNA reads decreased by 33.6% from 361,039 to 239,667 in the patient aged 75, and decreased by 16.1% from 635,169 to 532,708 in the patient aged 94, after error correction. Table 4.9 provides two rankings of top-10 age-specific miRNAs expressed only in the patient aged 94 before and after correction. New top-ranked age-specific miRNAs were identified in the patient aged 94. Without our error correction, top-1, 2, 3, 4, 6, 8 and 10 age-specific miRNAs would not be detected because erroneous reads which are identical with these reads also exist in the patient aged 75, with copy counts 3, 2, 2, 3, 4, 2 and 4 respectively.

Discovery of rare miRNAs is of strong interests. We examined the copy counts of low-expression miRNAs (or rare miRNAs) before and after error correction in the Alzheimer's disease patient aged 94. Note that all these rare miRNAs here are defined to have no expression in the patient aged 75. Table 4.10 shows top-10 copy count greatly-changed rare miRNAs before and after error correction. It suggests that the copy counts of these rare miRNAs were all enhanced by about 2 or 3 folds after error correction.

Figure 4.6 depicts how the quantity of a rare miRNA is enhanced from 14 copies to 23 in the correction process. The corrections were involved with four types of erroneous reads: four reads with a deletion error (labeled in blue), two reads with an insertion error (labeled in green), one read with a substitution error from A to G at position 11 and two reads with a substitution error from A to T at position 20 (labeled in orange). Our miREC can detect all of these erroneous reads and corrected them to recover this rare miRNA's quantity.

Verification results on the sequencing reads of the 963 miRXplore Universal Reference miRNAs (pure control and spike-in)

Our algorithm was tested on the sequencing reads of an equimolar mixture of synthetic miRNAs from the miRXplore Universal Reference that consists of 963 miRNAs from human, mouse, rat and viral sources (three replicate

samples miRXploreUR rep1-3 corresponding to GSE139936.GSM4149813, GSE139936.GSM4149814 and GSE139936.GSM4149815 (Hu, Yim, Ma, Huber, Davis, Bacusmo, Vermeulen, Zhou, Begley, DeMott et al. 2021)). The test was to verify

- whether our detected erroneous reads can be each corrected into one of the 963 miRNA sequences, and
- whether any new sequences are introduced into the read dataset after the correction.

An ideal performance should be: every error-corrected read is turned to be an exact copy of one of the 963 miRNA sequences, and previously non-existing reads are never created by the correction step.

Table 4.10: Copy count enhancement of 10 rare miRNAs after error correction in the human brain dataset related to an Alzheimer's disease patient aged 94

miRNA Sequence	read count _Before	read count _After
TCATTGGTTATCTAGCTGTATGC	6	18
TAGAACTTCGTCGAGTACGCTC	9	26
AAAAGCTGGGTTGAGAGGGCGTGA	6	17
AGCAGGACGGTGGCCATGGA	8	22
TGAGGCAGTAGGTTGTGTGGTTAT	6	16
TCCAGCATCAGTGATTTTGTTGT	6	16
TCACAGACAGCCGGTCTCTTTT	6	16
GTTGGTCCGAGTGTTGTGGGC	6	16
TCCCCGGCATCTCCACCAT	9	23
AGGAGATGGAATAGGAGCTTGA	8	20

Notes: _After indicates after error correction, while _Before indicates before error correction

Table 4.11: Correction performance by miREC in comparison with Karect on the sequencing reads of the 963 miRXplore Universal Reference miRNAs (pure control and spike-in)

Dataset (Total read count)	Method	Number of detected bases for correction	Total read count of the 963 miRNAs			Introduced new sequences	Distinct reads with the same frequencies before and after correction								
			before correction	after correction	Pct(%) increased		Total count	Minimum editing distance ##							
								## =0		## =1		## =2		## >=3	
								Count	Pct(%)	Count	Pct(%)	Count	Pct(%)	Count	Pct(%)
D18-6962_1	miREC	43,362	221,657	265,076	19.59	0	231,792	24	0.0104	1,809	0.7804	47,949	20.6862	182,010	78.5230
(544,056)	Karect	127,642		239,882	8.22	37,678	162,730	10	0.0061	11,031	6.7787	21,538	13.2354	130,151	79.9797
D18-6962_2	miREC	43,122	223,921	267,094	19.28	0	233,418	23	0.0099	1,813	0.7767	47,832	20.492	183,750	78.7214
(547087)	Karect	129,410		241,979	8.06	38,501	163,525	8	0.0049	11,095	6.7849	21,609	13.2145	130,813	79.9957
D18-6963_1	miREC	35,211	150,886	186,126	23.36	0	181,380	35	0.0193	1,310	0.7222	34,202	18.8565	145,833	80.4019
(402349)	Karect	108,583		167,678	11.13	30,885	124,456	11	0.0088	8,058	6.4746	13,726	11.0288	102,661	82.4878
D18-6963_2	miREC	35,006	152,201	187,237	23.02	0	180,491	35	0.0194	1,301	0.7219	34,154	18.9228	144,999	80.3359
(401407)	Karect	109,344		168,894	10.97	31,159	123,387	11	0.0089	8,040	6.5161	13,704	11.1065	101,632	82.3685
D18-6964_1	miREC	39,369	192,095	231,512	20.52	0	215,161	25	0.0116	1,372	0.6377	39,914	18.5508	173,850	80.8000
(490577)	Karect	113,706		209,543	9.08	32,412	154,167	12	0.0078	9,488	6.1544	17,004	11.0296	127,663	82.8083
D18-6964_2	miREC	39,210	194,177	233,441	20.22	0	213,499	26	0.0122	1,401	0.6562	39,802	18.6427	172,270	80.6889
(488317)	Karect	115,845		211,474	8.91	33,398	151,458	11	0.0073	9,513	6.2809	17,069	11.2698	124,865	88.4420
D19-10246	miREC	89,301	208,219	240,828	15.66	0	77,335	16	0.0207	1,782	2.3043	11,129	14.3906	64,408	83.2844
(767426)	Karect	85,491		224,089	7.62	14,462	78,927	7	0.0089	9,750	12.3532	14,004	17.7430	55,166	69.8950

[a] The parameter of kmer range of miREC: [8, 25].

[b] The parameters of Karect: -matchtype=hamming -celltype=haploid.

[c] D18-6962_1 (180719Ded_D18-6962_1_sequence.fastq) and D18-6962_2 (180719Ded_D18-6962_2_sequence.fastq) from GSE139936.GSM4149813;
D18-6963_1 (180719Ded_D18-6963_1_sequence.fastq) and D18-6963_2 (180719Ded_D18-6963_2_sequence.fastq) from GSE139936.GSM4149814;
D18-6964_1 (180719Ded_D18-6964_1_sequence.fastq) and D18-6964_2 (180719Ded_D18-6964_2_sequence.fastq) from GSE139936.GSM4149815;
D19-10246 (D19-10246.assembled.2NN.fastq) from GEO accession GSE159434.

The correction performance by miREC in comparison with Karect (the best literature method (Allam et al. 2015)) are shown in Table 4.11. On the sequencing dataset named D18-6962_1 of GSE139936.GSM4149813, our algorithm detected a total of 43362 errors. After correction, the correspondingly rectified reads were each exactly matched with one of the 963 miRNA sequences. The total read count of the 963 miRNAs was therefore increased by about 19.59% (see Supplementary file S1 for details). During this correction step, previously non-existing reads were never generated/created. In fact, the number of distinct reads was decreased from 259867 to 212093. On the other hand, almost all (99.22%) of the remaining unchanged 231792 distinct reads were not considered as the erroneous reads of the 963 miRNAs by our algorithm. This is reasonable because each of them has a minimum editing distance of 2 or bigger with any of the 963 miRNA sequences. These remaining reads also have an extremely low counts such as 1, 2 or 3. They can be considered as noisy reads which may be caused by the library preparation noise or contaminates.

Karect detected total 127642 errors, but only 18225 of them were corrected into the sequencing reads of the 963 miRNAs, increasing their read counts by 8.22% in total. Meanwhile, the other base modifications have introduced a pool of 37678 new sequences which did not exist in the dataset before Karect's correction.

From these comparisons, we note that our algorithm miREC has corrected almost all of those reads which should be rectified and that miREC has never introduced previously non-existing reads. This is true for all other datasets listed in Table 4.11. However, Karect introduced large pools of new reads which have never existed in the original reads set; also Karect corrected less than half of those reads which should be rectified.

On a spike-in sample of the 963 miRNAs at human cells (GSE159434.D19-10246.assembled.fastq (Hu, Yim, Huber, Bacusmo, Ma, DeMott, Levine, de Crécy-Lagard, Dedon & Cao 2019)), our algorithm detected 89301 erroneous reads of the 963 miRNAs. After correction, their read counts

increased by 15.66% in total. The algorithm did not generate any previously non-existing reads, but decreased the number of distinct reads by 45189, greatly diminishing the uncerternty/entropy of the data set. On the other hand, Karect detected and corrected 15885 erroneous reads of the 963 miRNAs, making their read counts increased by 7.62% in total. However, Karect created 14462 new reads which were non-existing previously.

These comparative results on both the control and spike-in sample demonstrate that our modified reads are genuine correction and that our algorithms do not generate any previous non-existing reads after the correction process.

4.4 Summary

In this work, we have proposed an miRNA sequencing error correction method named miREC, which is the first tool to address the error correction problem in the area. The novelty of the method is a 3-layer kmer-(k+1)mer-(k-1)mer lattice structure to hold the kmer's supersets and subsets' frequency differences which underline the locations of the errors and the correcting templates. Our miREC has showed excellent performance to rectify not only substitution errors but also indel errors at both simulated and real miRNA sequencing datasets. The experiments conducted with different running parameters showed that the miREC is insensitive to datasets and it has good robustness to guarantee high-quality correction performance. With the precise aberration correction and free of new error introduction, we are able to conduct ultrafine analysis on miRNA sequencing data at the single base resolution. The method is immediately applicable to miRNA sequencing datasets from the fields of plant biology and cancer biology which are worth future investigation in detail.

Availability

The data material and code are all available at the github link (`https://github.com/XuanrZhang/miREC.git`).

Chapter 5

Extended Error Correction Method for Small RNA Sequencing Reads

5.1 Introduction

Small RNAs are non-coding RNA molecules with short lengths (usually smaller than 200 nucleotides), mainly including microRNA, siRNA (small interfering RNA), piRNA (piwi-interacting RNA), snoRNA (small nucleolar RNA), tRNA-derived small RNA (tsRNA), srRNA (small rDNA-derived RNA) and rasiRNA (repeat associated small interfering RNA). All of these small RNA play important roles in molecular function. For example, miRNAs, as a very famous category in the small RNA family functions in RNA silencing and post-transcriptional regulation of gene expression (Bartel 2018). miRNAs function via base-pairing with complementary sequences within mRNA molecules (Bartel 2009). As a result, these mRNA molecules are silenced. miRNAs resemble the small interfering RNAs (siRNAs) of the RNA interference (RNAi) pathway, except miRNAs derive from regions of RNA transcripts that fold back on themselves to form short hairpins, whereas siRNAs derive from longer regions of double-stranded

RNA (Bartel 2004). Some studies show that piRNA and rasiRNA have been identified in the mentioned process and contribute to the RNAi (RNA interference) (Gunawardane, Saito, Nishida, Miyoshi, Kawamura, Nagami, Siomi & Siomi 2007). Small RNA related research, sequencing and analysis attract more and more attention especialy in human diseases (e.g. breast cancers) (Wu, Lu, Li, Lu, Guo & Ge 2011).

Small RNA sequencing (RNA-seq) technique makes related research available. Firstly, small RNA species are isolated and then RNA-seq can query thousands of small RNA with unprecedented sensitivity and dynamic range. With small RNA-Seq we can discover novel miRNAs and other small non-coding RNAs and examine the differential expression of all small RNAs in any sample. We can characterize variations such as isomiRs with single-base resolution and analyze any small RNA or miRNA without prior sequence or secondary structure information. However, sequencing data generated by machine goes with sequencing errors. Even though, the error rate can be low at 0.1% per base, erroneous bases in datasets still can cause wrong conclusion, especially in variation detection and isomiRs which is just single-base differences. Thus, high-quality and error-free small RNA sequences are required to correct for better investigations on small RNA editing events, small RNA isomiRs, differential expression of all small RNAs and novel small RNA discovery. With more convincing sequencing data, we can obtain a better understanding of how cells are regulated or misregulated under pathological conditions, thereby proposing better solution for RNA-related disease treatment.

Our method introduces a novel method for small RNA error correction which supports substitution, insertion and deletion error rectification. Compared with the miRNA error correction method, this method is more robust by supporting all kinds of small RNA sequencing (read length from 20 to 200 nucleotides). Furthermore, we improve the 3-layer lattice structure and combine it by reads with the same length n, length $(n+1)$ one and length $(n-1)$ one, which dramatically increases the method's efficiency. Finally, to

make a fine correction, we propose to do proportional correction. Specifically, in the correction phase, we do not correct all potential erroneous copies to the top one candidate; Instead, we divide corrections into top 3 candidates proportionally to remain all possible recovery. With this improvement, the method achieves high error correction performance.

5.2 Methods

An small RNA sequencing read r is a sequence $r_1 r_2 \cdots r_n$, $r_i \in \Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{N}, \mathrm{T}\}$, where A, C, G and T stand for the nucleotide bases Adenine, Cytosine, Guanine and Thymine respectively, and the character N stands for a uncertain nucleotide; n is the length of r. Usually, the length n of an small RNA read ranges from 20 to 200 in a dataset, and each read encompasses entire small RNA nucleotide info. Our method, SRNAEC (Small RNA Error Correction), uses a 3-layer lattice structure to select convincing candidates and combines proportional correction strategy to achieve optimal rectification.

5.2.1 A 3-layer Read Lattice Structure

Given an small RNA sequencing read multi-set S and a setting k, the copy count (or frequency) of a distinct read r in S is the total number of its copies in S. Consider a read r with length n, this r's neighborhood is defined as the set of reads $H(n, r)$ containing all possible distinct reads of S that each have only one base difference from the r. Similarly, r's $(n-1)$-neighborhood is defined as the set of read with length $(n-1)$, $H((n-1), r)$ containing all possible distinct reads with $(n-1)$ length of S each of which is an immediate subset of the read, r, and r's $(n+1)$-neighborhood is defined as the set of reads with length $(n+1)$, $H((n+1), r)$ containing all possible distinct reads with length $(n+1)$ of S each of which is an immediate superset of the r.

For example, if a small RNA read with length $n(n = 3)$ is given as GTC and assume that all its proper supersets and subsets exist in S, then

its $(n+1)$-neighborhood $H(4, GTC) = \{\underline{A}GTC,\ \underline{T}GTC,\ \underline{C}GTC,\ \underline{G}GTC,\ G\underline{A}TC,\ G\underline{T}TC,\ G\underline{C}TC,\ G\underline{G}TC,\ GT\underline{A}C,\ GT\underline{T}C,\ GT\underline{C}C,\ GT\underline{G}C,\ GTC\underline{A},\ GTC\underline{T},\ GTC\underline{C},\ GTC\underline{G}\}$. Its $(n-1)$-neighborhood $H(2, GTC) = \{TC,\ GC,\ GT\}$. These three neighborhoods of the read r can be combined and it is called a 3-layer read lattice structure of r. By considering 3-layer reads, our method can support insertion and deletion error correction at the same time. Note that in real cases, the length of small RNA read usually range from 20 to 200 nt. A schematic example of this lattice structure is shown in Figure 5.1. [1]

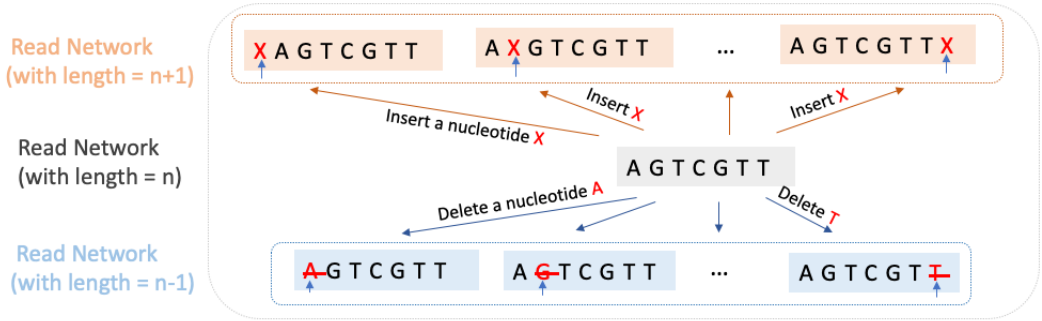

Figure 5.1: A 3-layer read lattice structure.

5.2.2 Proportional Correction

After erroneous read detection, related 3-layer lattice structures are constructed to guide corrections. Unlike previous correction methods, we do not correct the same erroneous reads to the top voted candidate instead of proportional corrections.

Our algorithm traverses all of distinct reads and compares it frequency with a threshold τ. If a read's frequency is lower then τ, the read is more likely to contain errors. For each of these read, all potential candidate reads are sorted by their frequencies and the top three candidates are selected for

[1] The red * symbol represents a nucleotide A, G, T, or C.

further proportional correcitons. Note that only reads with frequency larger than τ are able to be sorted. For every distinct erroneous read, we correct it differently according to their copy numbers.

- If the copy number of a erroneous read is one, proportional correction can not be triggered and the erroneous read is corrected to the top one potential candidates.

- If the number of potential candidate reads is one, proportional correction can not be triggered and all erroneous reads are corrected to the only one potential candidate.

- If the copy number of a erroneous read and the number of potential candidate reads are both greater than one, proportional correction can be triggered. For each of the top three candidates, we calculate the proportion of its frequency in the total frequency of them, and then divide all erroneous read proportionally into the top three candidates.

With proportional correction strategy, over-correction for a specific small RNA read can be avoided and read quantity recovery can cover more reads including high frequency reads as well as their low frequency isoforms or family members. The pseudo code of our algorithm is shown in Algorithm 5.1.

Our SRNAEC has been implemented as a software prototype. It is very easy to use and only one parameter (the frequency threshold τ) might need to tune when you run it. Based on practical experience, the frequency threshold τ is best recommended as 5 by default. The higher frequency τ is set, the bigger number of bases might be considered as errors. Thus, users should be cautious about using a too large frequency threshold to avoid over-correction. The SRNAEC also supports multi-threads to optima computing ability and achieve high-efficiency correction performance.

Algorithm 5.1 Small RNA Sequencing Read Error Correction

Input: A read set $\mathcal{RS} = \{r^1, r^2, \cdots, r^n\}$, a frequency border τ

Output: A corrected read set

Function ERROR_CORRECTION ($\mathcal{RS}, \tau$)

```
begin
    H[1...m] ← hash table   ▷ *[r]kmer info; H[i] is an array; Each element
     of H[i] is a tuple composed of sequence, its frequency and id array
     C_read          ▷ *[r]An two dimentional array store read and its frequency
     Candidates                              ▷ *[r]An array store candidate reads
     foreach r ∈ RS do
        if H[r].f < τ then
            C_read ← FindNeighbor(r);
            foreach read ∈ C_read do
                tmp ← H[r].f * Proportion(read);
                while tmp ≥ 1 do
                    Candidates ← Add(read);
                    tmp − −;
            for i = 1 to H[r].f do
                RS[H[r].id[i]] ← Candidates[i];
    Function FindNeighbor
     foreach read ∈ Neighbor(r) do
        if H[read].f > τ then
            Reads ← Add(read);
    Sort(Reads) by its frequency;
     return Reads
    return RS
```

5.3 Experiments and Results

5.3.1 Read Datasets

To evaluate the performance of error correction methods, simulated datasets are required and the ground truth of the errors should be known. We conducted our method on both simulated datasets and wet-lab raw sequencing read datasets.

Simulated datasets

Since there is no available small RNA read simulation tool for using, we introduce a novel process to generate simulated datasets that would have a close nature to wet-lab small RNA sequencing reads. We have two considerations in the process. One is to computationally replicate lab-verified small RNA sequences as templates to form the basic sequences of the simulated datasets, then we duplicate these basic sequences such that the copy counts of them follow a real distribution from a wet-lab dataset of small RNA sequencing reads. In fact, we replicated the small non-coding RNA sequences in DASHR 2.0 (Kuksa, Amlie-Wolf, Katanić, Valladares, Wang & Leung 2019) as the templates, which includes miRNA, piRNA, siRNA, snoRNA, tsRNA and srRNA, and made the copy count distribution of these template sequences to follow the distribution drawn from a typical smallRNA dataset under accession number SRR6317802. In other words, the sequences in our simulated datasets are not random sequences (they are real lab-verified small RNA sequences); their copy count distribution is not random either. Then we injected random errors into the simulated datasets under an error rate of 0.2% per base (Laehnemann et al. 2016). we randomly injected an erroneous base (substitution, deletion or insertion) at any position of the read. We recorded all of these randomly and purposely injected errors for performance evaluation.

Followed mentioned steps, we synthesized 4 simulated datasets with subs and indels error (denoted as D1, D2, D3 and D4). More details of the

Table 5.1: Description of used datasets

Datasets		ID	Total Reads	Total erroneous bases	Per read error rate
	simu	D1	2,620,163	261,449	0.263%
		D2	2,620,163	262,457	0.263%
		D3	2,620,163	262,734	0.264%
		D4	2,620,163	262,499	0.270%
		D5	2,620,163	523083	0.527%
		D6	2,620,163	524407	0.529%
		D7	2,620,163	524868	0.529%
		D8	2,620,163	523284	0.528%
	raw	Accession ID	Total reads	sample tissue	-
		SRR6317801	31,103,535	flower	-
		SRR6317802	21,831,615	flower	-
		SRR6317805	30,924,779	pod	-
		SRR6317806	30,862,811	pod	-

Notes: 'simu' means simulation datasets and 'raw' means raw sequencing datasets.

simulated datasets are shown in Table 5.1.

Wet-lab sequencing datasets

Wet-lab small RNA sequencing datasets for our performance evaluation are all downloaded from the Sequence Read Archive (SRA) (`https://www.ncbi.nlm.nih.gov/sra/`) under the accession numbers SRR6317801, SRR6317802, SRR6317805 and SRR6317806. These datasets have been originally studied for topics related to small RNA of lupinus luteus (Glazinska, Kulasek, Glinkowski, Wysocka & Kosiński 2020). They are all small RNA sequence data from lupinus luteus pod and Lupinus luteus flower. More details of these datasets are shown in Table 5.1.

Table 5.2: Details of wet-lab sequencing datasets with a few known errors

Dataset	Accession ID	Tissue	Number of subs	Number of ins	Number of del	Total errors
R1	SRR6317801	flower	7456	958	916	9330
R2	SRR6317802	flower	5089	683	647	6419
R3	SRR6317805	pod	7426	952	892	9270
R4	SRR6317806	pod	7428	912	904	9244

Notes: 'subs' means injected substitution errors, 'ins' means injected insertion errors, 'del' means injected deletion errors and 'Total errors' means the number of all injected errors.

Wet-lab sequencing datasets with a few known errors

To rigorously evaluate the error correction performance, we also randomly and purposely inject a small number of errors into these wet-lab datasets, rather than the simulated datasets, to see whether our algorithm can detect and correct these errors with ground truth, together with other errors without ground truth. Only when all of these artificial errors in the real-life small RNA sequencing reads can be detected and corrected, the corrections on the other bases (without ground truth) can be highly trustable.

This small number of artificial errors constitutes only 0.3% of total corrections in each dataset to avoid changing the original nature of the data. We have done these for all wet-lab sequencing datasets, we randomly injected small numbers of errors twice referring to steps in simulation datasets. More details of wet-lab datasets with injected errors are in the Table 5.2.

5.3.2 Evaluation Metrics

As the ground truth of the errors in the simulated datasets are known, we can use recall, precision and gain to compare the correction performance between different methods. On the wet-lab small RNA sequencing datasets, we measure the copy count changes of the reads, the entropy changes of the whole set of reads, and locations of the rectifications to understand the importance of error correction. There is no recall or precision performance on the wet-lab small RNA sequencing datasets, because the ground truth of error distributions is unknown.

Performance evaluation metrics on the simulated sequencing datasets

To assess the accuracy of the correction methods, we use the following three metrics, Precision (*TP/(TP+FP)*), Recall (*TP/(TP+FN)*), Gain (*(TP-FP)/(TP+FN)*). More details are shown in Section 2.3.1.

Metrics used for performance evaluation on wet-lab small RNA sequencing datasets

We examine the changes of small RNA copy counts and dataset entropy changes before and after error correction for multiple salmon fish small RNA sequencing datasets. Besides, we also summarize the position information of the corrections in the reads to record the proportion of corrections in the seed region. The concept of entropy was first introduced by a physicist Rudolf clausius (Clausius 1879), to measure a system's thermal energy per unit temperature. The amount of entropy is also a measure of the molecular disorder or randomness of a system. Here, we regard our sequencing read dataset as a system, and use the concept of entropy to define dataset entropy. For dataset entropy, we combine all low-frequency reads which are more likely to contain errors, to interpret the quality of datasets. More specificly, we define the small RNA count, dataset entropy and errors in the seed region as follows.

- small RNA count: the copy count of small RNA appearing in the datasets, which is corresponding to small RNA expression level or small RNA abundance.
- Dataset entropy: $-\sum_{i=1}^{n} p_i \cdot \log p_i$, where p_i is the proportion of reads whose frequency is small than i. We calculate the entropy for low-frequency reads and sum up to interpret the degree of disorder in the read dataset. When the entropy turns to be small, it means the certainty of the small RNA expression becomes higher.

5.3.3 Performance Evaluation

Our analysis and results are presented in three main parts. The first part is about the correction performance on the 4 simulated datasets; the second part is about correction performance on the wet-lab miRNA sequencing datasets after a small number of random errors are injected; the third part

is about expression abundance recovery, entropy change on the recently published small miRNA sequencing datasets after error correction and related case studies.

Correction performance on simulation datasets

The correction performance of our SRNAEC is presented in Table 5.3 in comparison with algorithms Karect (Allam et al. 2015), Coral (Salmela & Schröder 2011), and BFC (Li 2015). Coral and Karect are multi-alignment based error correction methods. BFC is a representative of the kmer based error correction methods. BFC requires a prior-setting of the k parameter; the best k in this work is 21 (namely, under other k settings, BFC did not exceed the performance of when k = 21). Karect is one of a few correction tools which supports the correction of indel errors.

Our method SRNAEC has excelled in the correction performance:

- It did not introduce any new error, namely, it achieved the highest gain and recall rates on all simulated datasets;
- It detected and corrected almost all of the errors including the indel errors; the recall rate ranges between 99.85% - 99.86%; the precision ranges between 99.90% - 99.91%; the gain rate ranges between 99.79% - 99.82%;

The average recall and gain rate of SRNAEC are much superior to Karect (the second-best method) respectively by 51.85% and 11.82% on all datasets. Specifically, the average recall rates of SRNAEC are 99.86%, which are 33.34%, 50.36% and 51.75% better than BFC, Coral and Karect. This implies that there are lots of errors undetected by these baseline methods meanwhile these introduced a lot of new errors. The kmer based method BFC performed worst on these datasets. More details are in the Table 5.3

The performance of SRNAEC is robust across all datasets. Futher experiments on real-life wet-lab sequencing datasets also prove that.

Table 5.3: Outstanding error correction performance by our SRNAEC in comparison with the best available tools

	Gain(%)				Recall(%)				Precision(%)			
	SRNAEC	BFC	Cor	Kar	SRNAEC	BFC	Cor	Kar	SRNAEC	BFC	Cor	Kar
D1	**99.82**	65.78	76.38	79.64	**99.86**	66.44	49.43	48.04	**99.91**	82.89	89.82	88.19
D2	**99.79**	66.79	76.58	79.65	**99.85**	66.61	49.58	47.97	**99.90**	83.4	89.83	88.29
D3	**99.81**	65.69	75.3	79.51	**99.86**	66.58	49.54	48.01	**99.90**	82.85	89.76	87.65
D4	**99.81**	66.82	76.37	79.43	**99.85**	66.46	49.45	48.02	**99.90**	83.41	89.71	88.18
AVE	**99.81**	66.27	76.16	79.56	**99.86**	66.52	49.50	48.01	**99.90**	83.14	89.78	88.08
D5	**99.81**	75.11	85.61	79.91	**99.85**	63.94	49.79	47.98	**99.90**	87.56	92.80	89.96
D6	**99.81**	75.26	85.29	79.75	**99.86**	63.98	49.61	47.98	**99.90**	87.63	92.65	89.87
D7	**99.82**	75.54	85.67	79.67	**99.86**	63.88	49.59	47.89	**99.91**	87.77	92.76	89.83
D8	**99.81**	75.06	85.55	79.80	**99.85**	63.94	49.67	47.98	**99.90**	87.53	92.77	89.90
AVE	**99.81**	75.24	85.53	79.78	**99.86**	63.94	49.66	47.96	**99.90**	87.62	92.75	89.89

Notes: AVE indicates the average score over the four datasets. Bold font indicates the best result in the row. Cor and Kar stand for the Coral method, and the Karect method respectively.

Table 5.4: Error correction results on raw sequencing datasets with injected errors

	SRNAEC		Coral		BFC		Karect	
Total genuine errors	Total corrections	Gain rate	Total corrections	Gainf rate	Total corrections	Gainf rate	Total corrections	Gain rate
8269	7265	87.86%	919	11.11%	3725	45.05%	3936	47.60%
5571	3726	66.88%	505	9.06%	2142	38.45%	2226	39.96%
7938	6886	86.75%	950	11.97%	3775	47.56%	3962	49.91%
8530	7328	85.91%	989	11.59%	4534	53.15%	4623	54.20%
Average		81.85%		10.94%		46.05%		47.92%

Table 5.5: Changes of unique small RNA reads and entropy

	Total Unique small RNA Before correction	Total Unique small RNA After correction	Unique read Reduction	Entropy Reduction (%)
R1	11,787,926	10,185,465	1,602,461	13.59%
R2	11,063,723	10,076,326	987,397	8.92%
R3	12,672,715	11,111,546	1,561,169	12.32%
R4	13,862,064	12,206,205	1,655,859	11.95%
Averge	12346607	10894886	1451722	11.69%

Correction performance on wet-lab miRNA sequencing datasets injected with small numbers of random known errors

To competenvise evaluation correction performance on real datasets, we propose an error injection strategy to inject a few known errors and labled these erroneous reads. Through examine correction performance on these reads, correction performance on the whole datasets can be supported.

As we mentioned in Section 5.3.1, we injected 0.3 percent of known errors of four wet-lab sequencing datasets R1 to R4, specifically 9330, 6419,9270 and 9244 errors in them. These random modifications introduced/injected errors into the datasets, where a random base modification is *not* considered as a genuine error if its correspondingly modified read becomes identical with another read having a high frequency (i.e., copy count > 5). There are 7265, 3726, 6886, and 7328 genuine errors in R1 to R4 datasets. Referring to the results in Table 5.4, We can see that our algorithm corrected 81.85 per cent of all genuine errors, the second best method, the karect only correct 47.92 per cent of these genuine errors, in average.

From the table 5.5, we can see the number of unique reads decrease $1,451,722$ in average and the entropy of datasets decrease almost 12% in average. These entropy declines in the datasets theoretically mean that the certainty of the small RNA expression level is greatly improved. In other words, our method can enhance the data quality in the perspective of a lower entropy or a higher certainty.

5.4 Summary

In this chapter, we developed SRNAEC, a small RNA sequencing error correction method, by constructing the 3-layer read lattice structure, which achieve high-efficiency error correction. Proportional correction strategy is used to guarantee all small RNA's quantity recovery. Parallel computing is implemented in the approach to accelerate the correction process. Experimental results evaluating on both simulated and raw sequencing

datasets have achieved outstanding correction performance, and it is much superior to performances of the state-of-the-art methods: Karect, Coral and BFC.

Availability

The data material and code are all available at the github link (`https://github.com/XuanrZhang/SRNAEC.git`).

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has mainly addressed three research problems on genomic sequencing data error correction, namely error correction for instance cases, error correction for microRNA sequencing reads and error correction for small RNA sequencing reads. The proposed methods for solving these problems are detailed in Chapters 3-5. In the following content, the results and findings of each research problem are summarized.

In Chapter 3, we proposed an instance-based strategy to correct errors. It provides high-quality reads for any given instance case and is implemented as a tool named InsEC. It is designed to correct errors in reads related to instance cases (*e.g.*, a set of genes or a part of the genome sequence. The instance-based strategy makes it possible to make use of data traits only related to an instance, which guarantees that we can approach the ground truth of the instance case and then achieve better error correction performance. In the instance extraction step, all reads related to a given instance are extracted by using read mapping strategies. In the correction step, we take advantage of alignment processes and correct errors according to the alignment. Besides, statistical models are used to avoid induced errors as well. Intensive experiments are conducted with other state-of-the-art

methods on both simulated and real datasets. The results demonstrate the superiority of our method, which achieves the best error correction performance (*e.g.*, precision, recall, and gain rate in average) and further assembly results (*e.g.*, N50, the length of contig, and contig quality).

Chapter 4 developed the first method for miRNA read error correction. Existing error correction methods do not work for miRNA sequencing data attributed to miRNAs' length and per-read-coverage properties distinct from DNA or mRNA sequencing reads. Although the error rate can be low at 0.1%, precise rectification of these errors is critically essential because isoform variation analysis at single-base resolution such as novel isomiR discovery, editing events understanding, differential expression analysis, or tissue-specific isoform identification is very sensitive to base positions and copy counts of the reads. We present a novel lattice structure combining kmers, (k-1)mers, and (k+1)mers to address this problem. Moreover, the method is particularly effective for the correction of indel errors. Extensive tests on datasets having known ground truth of errors demonstrate that the method is able to remove almost all of the errors, without introducing any new error, to improve the data quality from every-50-reads containing one error to every-1300-reads containing one error. Studies on wet-lab miRNA sequencing datasets show that the errors are often rectified at the 5' ends and the seed regions of the reads and that there are remarkable changes after the correction in miRNA isoform abundance, the volume of singleton reads, overall entropy, isomiR families, tissue-specific miRNAs, and rare-miRNA quantities.

Chapter 5 introduces a novel method for minor RNA error correction which supports substitution, insertion, and deletion error rectification. Compared with the miRNA error correction method, this method is more robust by supporting all kinds of small RNA sequencing (read length from 20 nt to 200 nt). Furthermore, we improve the three-layer lattice structure and combine it by reads with the same length, length plus one, and length minus one, which dramatically increases the method's efficiency.

Finally, to make a fine correction, we consider RNA's isoform and propose correction proportionally. Specifically, in the correction phase, we do not correct all potential erroneous copies to the top one candidate; Instead, we divide corrections into top 3 candidates proportionally to remain all possible recovery. With this improvement, the method achieves high error correction performance, and its precision, recall, and gain rate are superior to all other existing error correction methods. Extensive experiments on simulation and raw sequencing data prove our method's ability. Thus, our error correction method does help improve data quality and necessary for all downstream analyses.

6.2 Summary of Important Results

The main contributions of **the instance-based error correction method** are listed.

- We proposed the first instance-based algorithm to solve the problem of short read error correction related to any instance case (*e.g.*, a set of genes or a part of the genome sequence). The novel idea of the algorithm is concentration on data traits only related to specific instances.
- The algorithm achieves the best performance compared with the state-of-the-art methods not only in correction but also further assembly results.

The main contributions of **the microRNA error correction method** are listed.

- We proposed the first algorithm to solve the problem of correcting errors (substitutions and indels) in microRNA sequencing reads. The novel idea of the algorithm is a 3-layer kmer lattice structure.
- The algorithm did not introduce any new error; It detected and corrected almost all of the errors, including the indel errors; the recall

rate ranges between 96.0% - 97.9%; the precision ranges between 98.6% - 99.5%;

- Identified significant changes in isoform abundance, whole set entropy and base positions after error correction on salmon fish miRNA sequencing reads; and studied class-specific miRNAs and rare miRNAs on human brain and beta cells miRNA sequencing datasets after our error correction.

The main contributions of **the extended error correction method** are listed.

- We proposed the first algorithm to solve the problem of correcting errors in all types of small RNA sequencing reads, supporting substitutions and indels correction. The novel idea of the algorithm is a 3-layer read lattice structure and proportional correction strategy;
- The algorithm achieved outstanding and robust correction performance; It detected and corrected almost all of the errors including the indel errors; More specifically, the average recall rate is 99.86%; the average precision is 99.9% and the average gain rate is 99.81%;
- Identified significant changes in small RNA abundance and whole set entropy after error correction on wet-lab small RNA sequencing reads;

6.3 Perspectives and Future Research

In addition to the encouraging results and findings, there are still some problems as well as challenges need to be addressed in the future.

First, in Chapter 3, the parallel implement version can be extended on the previous InsEC version. Also, some high-frequency used downstream analysis tools can be combined with the InsEC for comprehensive use. With the wide application of the third-generation sequencing technology, long

reads are generated and they suffer from high error rate. It would be interesting to take advantage of the second-generation technology to help long reads correction. Furthermore, combining high-quality long read data can dramatically improve the precision of genome assembly.

Secondly, in Chapter 4 and Chapter 5, further analysis pipelines of small RNA analysis can be combined and conduct more case studies to convincing correction performance. In future work, we will achieve a more comprehensive tool and can easily suit to all correction requirements.

Bibliography

Akogwu, I., Wang, N., Zhang, C. & Gong, P. (2016), 'A comparative study of k-spectrum-based error correction methods for next-generation sequencing data analysis', *Human genomics* **10**(2), 20.

Allam, A., Kalnis, P. & Solovyev, V. (2015), 'Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data', *Bioinformatics* **31**(21), 3421–3428.

Andreassen, R., Worren, M. M. & Høyheim, B. (2013), 'Discovery and characterization of mirna genes in atlantic salmon (salmo salar) by use of a deep sequencing approach', *BMC genomics* **14**(1), 482.

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D. et al. (2012), 'Spades: a new genome assembly algorithm and its applications to single-cell sequencing', *Journal of computational biology* **19**(5), 455–477.

Bartel, D. P. (2004), 'Micrornas: genomics, biogenesis, mechanism, and function', *cell* **116**(2), 281–297.

Bartel, D. P. (2009), 'Micrornas: target recognition and regulatory functions', *cell* **136**(2), 215–233.

Bartel, D. P. (2018), 'Metazoan micrornas', *Cell* **173**(1), 20–51.

Bashir, N., Ragab, E., Khabour, O., Khassawneh, B., Alfaqih, M. & Momani, J. (2018), 'The association between epidermal growth factor receptor (egfr) gene polymorphisms and lung cancer risk', *Biomolecules* **8**(3), 53.

Beerenwinkel, N. & Zagordi, O. (2011), 'Ultra-deep sequencing for the analysis of viral populations', *Current opinion in virology* **1**(5), 413–418.

Bilanges, B., Posor, Y. & Vanhaesebroeck, B. (2019), 'Pi3k isoforms in cell signalling and vesicle trafficking', *Nature Reviews Molecular Cell Biology* **20**(9), 515–534.

Cainap, C., Balacescu, O., Cainap, S. S. & Pop, L.-A. (2021), 'Next generation sequencing technology in lung cancer diagnosis', *Biology* **10**(9), 864.

Chandran, A. (2018), Overview of next-generation sequencing technologies and its application in chemical biology, *in* 'Advancing Development of Synthetic Gene Regulators', Springer, pp. 1–41.

Chekulaeva, M. & Filipowicz, W. (2009), 'Mechanisms of mirna-mediated post-transcriptional regulation in animal cells', *Current opinion in cell biology* **21**(3), 452–460.

Chopra, R., Burow, G., Farmer, A., Mudge, J., Simpson, C. E., Wilkins, T. A., Baring, M. R., Puppala, N., Chamberlin, K. D. & Burow, M. D. (2015), 'Next-generation transcriptome sequencing, snp discovery and validation in four market classes of peanut, arachis hypogaea l.', *Molecular Genetics and Genomics* **290**(3), 1169–1180.

Clausius, R. (1879), *The mechanical theory of heat*, Macmillan.

Cloonan, N., Wani, S., Xu, Q., Gu, J., Lea, K., Heater, S., Barbacioru, C., Steptoe, A. L., Martin, H. C., Nourbakhsh, E. et al. (2011), 'Microrna and their isomirs function cooperatively to target common biological pathways', *Genome biology* **12**(12), 1–20.

Dai, F.-Q., Li, C.-R., Fan, X.-Q., Tan, L., Wang, R.-T. & Jin, H. (2019), 'mir-150-5p inhibits non-small-cell lung cancer metastasis and recurrence by targeting hmga2 and β-catenin signaling', *Molecular Therapy-Nucleic Acids* **16**, 675–685.

DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., Del Angel, G., Rivas, M. A., Hanna, M. et al. (2011), 'A framework for variation discovery and genotyping using next-generation dna sequencing data', *Nature genetics* **43**(5), 491.

Dutta, R. K., Chinnapaiyan, S. & Unwalla, H. (2019), 'Aberrant micrornaomics in pulmonary complications: Implications in lung health and diseases', *Molecular Therapy-Nucleic Acids* **18**, 413–431.

Ebhardt, H. A., Tsang, H. H., Dai, D. C., Liu, Y., Bostan, B. & Fahlman, R. P. (2009), 'Meta-analysis of small rna-sequencing errors reveals ubiquitous post-transcriptional rna modifications', *Nucleic acids research* **37**(8), 2461–2470.

El-Telbany, A. & Ma, P. C. (2012), 'Cancer genes in lung cancer: racial disparities: are there any?', *Genes & cancer* **3**(7-8), 467–480.

Fernandez-Valverde, S. L., Taft, R. J. & Mattick, J. S. (2010), 'Dynamic isomir regulation in drosophila development', *Rna* **16**(10), 1881–1888.

Findlay, G. M., Daza, R. M., Martin, B., Zhang, M. D., Leith, A. P., Gasperini, M., Janizek, J. D., Huang, X., Starita, L. M. & Shendure, J. (2018), 'Accurate classification of brca1 variants with saturation genome editing', *Nature* **562**(7726), 217.

Frazer, K. A. (2012), 'Decoding the human genome', *Genome research* **22**(9), 1599–1601.

Fung, C., Zhou, P., Joyce, S., Trent, K., Yuan, J.-M., Grandis, J. R., Weissfeld, J. L., Romkes, M., Weeks, D. E. & Egloff, A. M. (2015),

'Identification of epidermal growth factor receptor (egfr) genetic variants that modify risk for head and neck squamous cell carcinoma', *Cancer letters* **357**(2), 549–556.

Giraldez, M. D., Spengler, R. M., Etheridge, A., Godoy, P. M., Barczak, A. J., Srinivasan, S., De Hoff, P. L., Tanriverdi, K., Courtright, A., Lu, S. et al. (2018), 'Comprehensive multi-center assessment of small rna-seq methods for quantitative mirna profiling', *Nature biotechnology* **36**(8), 746–757.

Glazinska, P., Kulasek, M., Glinkowski, W., Wysocka, M. & Kosiński, J. G. (2020), 'Luludb—the database created based on small rna, transcriptome, and degradome sequencing shows the wide landscape of non-coding and coding rna in yellow lupine (lupinus luteus l.) flowers and pods', *Frontiers in Genetics* **11**, 455.

Goodwin, S., McPherson, J. D. & McCombie, W. R. (2016), 'Coming of age: ten years of next-generation sequencing technologies', *Nature Reviews Genetics* **17**(6), 333.

Greenfield, P., Duesing, K., Papanicolaou, A. & Bauer, D. C. (2014), 'Blue: correcting sequencing errors using consensus and context', *Bioinformatics* **30**(19), 2723–2732.

Gunawardane, L. S., Saito, K., Nishida, K. M., Miyoshi, K., Kawamura, Y., Nagami, T., Siomi, H. & Siomi, M. C. (2007), 'A slicer-mediated mechanism for repeat-associated sirna 5'end formation in drosophila', *science* **315**(5818), 1587–1590.

Guo, L. & Chen, F. (2014), 'A challenge for mirna: multiple isomirs in mirnaomics', *Gene* **544**(1), 1–7.

Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. (2013), 'Quast: quality assessment tool for genome assemblies', *Bioinformatics* **29**(8), 1072–1075.

Hackl, T., Hedrich, R., Schultz, J. & Förster, F. (2014), 'proovread: large-scale high-accuracy pacbio correction through iterative short read consensus', *Bioinformatics* **30**(21), 3004–3011.

Hakimi, A. A., Ostrovnaya, I., Jacobsen, A., Susztak, K., Coleman, J. A., Russo, P., Winer, A. G., Mano, R., Sankin, A. I., Motzer, R. J. et al. (2016), 'Validation and genomic interrogation of the met variant rs11762213 as a predictor of adverse outcomes in clear cell renal cell carcinoma', *Cancer* **122**(3), 402–410.

Heo, Y., Wu, X.-L., Chen, D., Ma, J. & Hwu, W.-M. (2014), 'Bless: bloom filter-based error correction solution for high-throughput sequencing reads', *Bioinformatics* **30**(10), 1354–1362.

Hoefer, I. E. (2020), 'Isolating functional (iso) mirna targets during ischemia', *Molecular Therapy* **28**(1), 7–8.

Hu, J. F., Yim, D., Huber, S. M., Bacusmo, J. M., Ma, D., DeMott, M. S., Levine, S. S., de Crécy-Lagard, V., Dedon, P. C. & Cao, B. (2019), 'Sequencing-based quantitative mapping of the cellular small rna landscape', *bioRxiv* p. 841130.

Hu, J. F., Yim, D., Ma, D., Huber, S. M., Davis, N., Bacusmo, J. M., Vermeulen, S., Zhou, J., Begley, T. J., DeMott, M. S. et al. (2021), 'Quantitative mapping of the cellular small rna landscape with aqrna-seq', *Nature Biotechnology* pp. 1–11.

Huang, W., Li, L., Myers, J. R. & Marth, G. T. (2011), 'Art: a next-generation sequencing read simulator', *Bioinformatics* **28**(4), 593–594.

Ilie, L., Fazayeli, F. & Ilie, S. (2010), 'Hitec: accurate error correction in high-throughput sequencing data', *Bioinformatics* **27**(3), 295–302.

Ilie, L. & Molnar, M. (2013), 'Racer: rapid and accurate correction of errors in reads', *Bioinformatics* **29**(19), 2490–2493.

Kamps, R., Brandão, R. D., Bosch, B. J., Paulussen, A. D., Xanthoulea, S., Blok, M. J. & Romano, A. (2017), 'Next-generation sequencing in oncology: genetic diagnosis, risk prediction and cancer classification', *International journal of molecular sciences* **18**(2), 308.

Kao, W.-C., Chan, A. H. & Song, Y. S. (2011), 'Echo: a reference-free short-read error correction algorithm', *Genome research* **21**(7), 1181–1192.

Kokot, M., Długosz, M. & Deorowicz, S. (2017), 'Kmc 3: counting and manipulating k-mer statistics', *Bioinformatics* **33**(17), 2759–2761.

Kozomara, A., Birgaoanu, M. & Griffiths-Jones, S. (2019), 'mirbase: from microrna sequences to function', *Nucleic acids research* **47**(D1), D155–D162.

Kuksa, P. P., Amlie-Wolf, A., Katanić, Ž., Valladares, O., Wang, L.-S. & Leung, Y. Y. (2019), 'Dashr 2.0: integrated database of human small non-coding rna genes and mature products', *Bioinformatics* **35**(6), 1033–1039.

Laehnemann, D., Borkhardt, A. & McHardy, A. C. (2015), 'Denoising dna deep sequencing data—high-throughput sequencing errors and their correction', *Briefings in bioinformatics* **17**(1), 154–179.

Laehnemann, D., Borkhardt, A. & McHardy, A. C. (2016), 'Denoising DNA deep sequencing data—high-throughput sequencing errors and their correction', *Briefings in Bioinformatics* **17**(1), 154–179.

Lan, C., Peng, H., McGowan, E. M., Hutvagner, G. & Li, J. (2018), 'An isomir expression panel based novel breast cancer classification approach using improved mutual information', *BMC medical genomics* **11**(6), 118.

Le, H.-S., Schulz, M. H., McCauley, B. M., Hinman, V. F. & Bar-Joseph, Z. (2013), 'Probabilistic error correction for rna sequencing', *Nucleic acids research* **41**(10), e109–e109.

Li, H. (2013), 'Aligning sequence reads, clone sequences and assembly contigs with bwa-mem', *arXiv preprint arXiv:1303.3997* .

Li, H. (2015), 'Bfc: correcting illumina sequencing errors', *Bioinformatics* **31**(17), 2885–2887.

Li, R., Li, Y., Fang, X., Yang, H., Wang, J., Kristiansen, K. & Wang, J. (2009), 'Snp detection for massively parallel whole-genome resequencing', *Genome research* **19**(6), 1124–1132.

Liao, Z., Li, D., Wang, X., Li, L. & Zou, Q. (2018), 'Cancer diagnosis through isomir expression with machine learning method', *Current Bioinformatics* **13**(1), 57–63.

Lim, E.-C., Müller, J., Hagmann, J., Henz, S. R., Kim, S.-T. & Weigel, D. (2014), 'Trowel: a fast and accurate error correction module for illumina sequencing reads', *Bioinformatics* **30**(22), 3264–3265.

Limasset, A., Flot, J.-F. & Peterlongo, P. (2020), 'Toward perfect reads: self-correction of short reads via mapping on de bruijn graphs', *Bioinformatics* **36**(5), 1374–1381.

Liu, C.-H., Wang, Z., Huang, S., Sun, Y. & Chen, J. (2019), 'Microrna-145 regulates pathological retinal angiogenesis by suppression of tmod3', *Molecular Therapy-Nucleic Acids* **16**, 335–347.

Liu, H., Lei, C., He, Q., Pan, Z., Xiao, D. & Tao, Y. (2018), 'Nuclear functions of mammalian micrornas in gene regulation, immunity and cancer', *Molecular cancer* **17**(1), 64.

Liu, H.-P., Lai, H.-M. & Guo, Z. (2020), 'Prostate cancer early diagnosis: circulating microrna pairs potentially beyond single micrornas upon 1231 serum samples', *Briefings in Bioinformatics* .

Liu, Y., Schröder, J. & Schmidt, B. (2012), 'Musket: a multistage k-mer spectrum-based error corrector for illumina sequence data', *Bioinformatics* **29**(3), 308–315.

Liu, Y., Zhang, X., Zou, Q. & Zeng, X. (2020), 'Minirmd: accurate and fast duplicate removal tool for short reads via multiple minimizers', *Bioinformatics* . btaa915.

Lutterbach, B., Zeng, Q., Davis, L. J., Hatch, H., Hang, G., Kohl, N. E., Gibbs, J. B. & Pan, B.-S. (2007), 'Lung cancer cell lines harboring met gene amplification are dependent on met for growth and survival', *Cancer Research* **67**(5), 2081–2088.

Marchetti, A., Martella, C., Felicioni, L., Barassi, F., Salvatore, S., Chella, A., Camplese, P. P., Iarussi, T., Mucilli, F., Mezzetti, A. et al. (2005), 'EGFR mutations in non–small-cell lung cancer: analysis of a large series of cases and development of a rapid and sensitive method for diagnostic screening with potential implications on pharmacologic treatment', *Journal of Clinical Oncology* **23**(4), 857–865.

Mardis, E. R. (2013), 'Next-generation sequencing platforms', *Annual review of analytical chemistry* **6**, 287–303.

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R. & Sander, C. (2011), 'Protein 3d structure computed from evolutionary sequence variation', *PloS one* **6**(12), e28766.

Martí, E., Pantano, L., Bañez-Coronel, M., Llorens, F., Miñones-Moyano, E., Porta, S., Sumoy, L., Ferrer, I. & Estivill, X. (2010), 'A myriad of mirna variants in control and huntington's disease brain regions detected by massively parallel sequencing', *Nucleic acids research* **38**(20), 7219–7235.

Martin, M. (2011), 'Cutadapt removes adapter sequences from high-throughput sequencing reads', *EMBnet. journal* **17**(1), 10–12.

Meng, L., Liu, C., Lü, J., Zhao, Q., Deng, S., Wang, G., Qiao, J., Zhang, C., Zhen, L., Lu, Y. et al. (2017), 'Small rna zippers lock mirna molecules and block mirna function in mammalian cells', *Nature communications* **8**(1), 1–10.

Metzker, M. L. (2010), 'Sequencing technologies—the next generation', *Nature reviews genetics* **11**(1), 31.

Millot, G. A., Carvalho, M. A., Caputo, S. M., Vreeswijk, M. P., Brown, M. A., Webb, M., Rouleau, E., Neuhausen, S. L., Hansen, T. v. O., Galli, A. et al. (2012), 'A guide for functional analysis of brca1 variants of uncertain significance', *Human mutation* **33**(11), 1526–1537.

Minoche, A. E., Dohm, J. C. & Himmelbauer, H. (2011), 'Evaluation of genomic high-throughput sequencing data generated on illumina hiseq and genome analyzer systems', *Genome biology* **12**(11), R112.

Mullany, L. E., Herrick, J. S., Wolff, R. K. & Slattery, M. L. (2016), 'Microrna seed region length impact on target messenger rna expression and survival in colorectal cancer', *PloS one* **11**(4), e0154177.

Needleman, S. B. & Wunsch, C. D. (1970), 'A general method applicable to the search for similarities in the amino acid sequence of two proteins', *Journal of molecular biology* **48**(3), 443–453.

Neilsen, C. T., Goodall, G. J. & Bracken, C. P. (2012), 'Isomirs–the overlooked repertoire in the dynamic micrornaome', *Trends in genetics* **28**(11), 544–549.

Pillman, K. A., Goodall, G. J., Bracken, C. P. & Gantier, M. P. (2019), 'mirna length variation during macrophage stimulation confounds the interpretation of results: implications for mirna quantification by rt-qpcr', *RNA* **25**(2), 232–238.

Pisignano, G., Napoli, S., Magistri, M., Mapelli, S. N., Pastori, C., Di Marco, S., Civenni, G., Albino, D., Enriquez, C., Allegrini, S. et al. (2017),

'A promoter-proximal transcript targeted by genetic polymorphism controls e-cadherin silencing in human cancers', *Nature communications* **8**(1), 1–16.

Ravindran, S. (2019), 'Fixing genome errors one base at a time', *Nature* **575**, 553–555.

Salk, J. J., Schmitt, M. W. & Loeb, L. A. (2018), 'Enhancing the accuracy of next-generation sequencing for detecting rare and subclonal mutations', *Nature Reviews Genetics* **19**(5), 269.

Salmela, L. (2010), 'Correction of sequencing errors in a mixed set of reads', *Bioinformatics* **26**(10), 1284–1290.

Salmela, L. & Schröder, J. (2011), 'Correcting errors in short reads by multiple alignments', *Bioinformatics* **27**(11), 1455–1461.

Salzberg, S. L., Phillippy, A. M., Zimin, A., Puiu, D., Magoc, T., Koren, S., Treangen, T. J., Schatz, M. C., Delcher, A. L., Roberts, M. et al. (2012), 'Gage: A critical evaluation of genome assemblies and assembly algorithms', *Genome research* **22**(3), 557–567.

Sänger, L., Bender, J., Rostowski, K., Golbik, R., Lilie, H., Schmidt, C., Behrens, S.-E. & Friedrich, S. (2020), 'Alternatively spliced isoforms of auf1 regulate a mirna-mrna interaction differentially through their ygg motif', *RNA biology* .

Schirmer, M., Sloan, W. T. & Quince, C. (2012), 'Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes', *Briefings in bioinformatics* **15**(3), 431–442.

Schröder, J., Schröder, H., Puglisi, S. J., Sinha, R. & Schmidt, B. (2009), 'Shrec: a short-read error correction method', *Bioinformatics* **25**(17), 2157–2163.

Sheikhizadeh, S. & de Ridder, D. (2015), 'Ace: accurate correction of errors using k-mer tries', *Bioinformatics* **31**(19), 3216–3218.

Slatko, B. E., Gardner, A. F. & Ausubel, F. M. (2018), 'Overview of next-generation sequencing technologies', *Current protocols in molecular biology* **122**(1), e59.

Song, L. & Florea, L. (2015), 'Rcorrector: efficient and accurate error correction for illumina rna-seq reads', *GigaScience* **4**(1), s13742–015.

Song, L., Florea, L. & Langmead, B. (2014), 'Lighter: fast and memory-efficient sequencing error correction without counting', *Genome biology* **15**(11), 509.

Stelzer, G., Rosen, N., Plaschkes, I., Zimmerman, S., Twik, M., Fishilevich, S., Stein, T. I., Nudel, R., Lieder, I., Mazor, Y. et al. (2016), 'The genecards suite: from gene data mining to disease genome sequence analyses', *Current protocols in bioinformatics* **54**(1), 1–30.

Tan, G. C., Chan, E., Molnar, A., Sarkar, R., Alexieva, D., Isa, I. M., Robinson, S., Zhang, S., Ellis, P., Langford, C. F. et al. (2014), '5 isomir variation is of functional and evolutionary importance', *Nucleic acids research* **42**(14), 9424–9435.

Telonis, A. G., Magee, R., Loher, P., Chervoneva, I., Londin, E. & Rigoutsos, I. (2017), 'Knowledge about the presence or absence of mirna isoforms (isomirs) can successfully discriminate amongst 32 tcga cancer types', *Nucleic acids research* **45**(6), 2973–2985.

Telonis, A. G. & Rigoutsos, I. (2018), 'Race disparities in the contribution of mirna isoforms and trna-derived fragments to triple-negative breast cancer', *Cancer research* **78**(5), 1140–1154.

Trontti, K., Väänänen, J., Sipilä, T., Greco, D. & Hovatta, I. (2018), 'Strong conservation of inbred mouse strain microrna loci but broad variation

in brain micrornas due to rna editing and isomir expression', *RNA* **24**(5), 643–655.

Van der Auwera, G. A., Carneiro, M. O., Hartl, C., Poplin, R., Del Angel, G., Levy-Moonshine, A., Jordan, T., Shakir, K., Roazen, D., Thibault, J. et al. (2013), 'From fastq data to high-confidence variant calls: the genome analysis toolkit best practices pipeline', *Current protocols in bioinformatics* **43**(1), 11–10.

van der Kwast, R. V., Woudenberg, T., Quax, P. H. & Nossent, A. Y. (2020), 'Microrna-411 and its 5-isomir have distinct targets and functions and are differentially regulated in the vasculature under ischemia', *Molecular Therapy* **28**(1), 157–170.

Wang, Y., Freedman, J. A., Liu, H., Moorman, P. G., Hyslop, T., George, D. J., Lee, N. H., Patierno, S. R. & Wei, Q. (2017), 'Associations between rna splicing regulatory variants of stemness-related genes and racial disparities in susceptibility to prostate cancer', *International journal of cancer* **141**(4), 731–743.

Waterhouse, A., Bertoni, M., Bienert, S., Studer, G., Tauriello, G., Gumienny, R., Heer, F. T., de Beer, T. A. P., Rempfer, C., Bordoli, L. et al. (2018), 'Swiss-model: homology modelling of protein structures and complexes', *Nucleic acids research* **46**(W1), W296–W303.

Woldemariam, N. T., Agafonov, O., Høyheim, B., Houston, R. D., Taggart, J. B. & Andreassen, R. (2019), 'Expanding the mirna repertoire in atlantic salmon; discovery of isomirs and mirnas highly expressed in different tissues and developmental stages', *Cells* **8**(1), 42.

Wu, Q., Lu, Z., Li, H., Lu, J., Guo, L. & Ge, Q. (2011), 'Next-generation sequencing of micrornas for breast cancer detection', *Journal of biomedicine and biotechnology* **2011**.

Xiao, Y. & MacRae, I. J. (2019), 'Toward a comprehensive view of microrna biology', *Molecular cell* **75**(4), 666–668.

Yamamoto, T. (2021), 'Genomic aberrations associated with the pathophysiological mechanisms of neurodevelopmental disorders', *Cells* **10**(9), 2317.

Yang, A., Shao, T.-J., Bofill-De Ros, X., Lian, C., Villanueva, P., Dai, L. & Gu, S. (2020), 'Ago-bound mature mirnas are oligouridylated by tuts and subsequently degraded by dis3l2', *Nature Communications* **11**(1), 1–13.

Yang, X., Dorman, K. S. & Aluru, S. (2010), 'Reptile: representative tiling for short read error correction', *Bioinformatics* **26**(20), 2526–2533.

Yeung, C. L. A., Co, N.-N., Tsuruga, T., Yeung, T.-L., Kwan, S.-Y., Leung, C. S., Li, Y., Lu, E. S., Kwan, K., Wong, K.-K. et al. (2016), 'Exosomal transfer of stroma-derived mir21 confers paclitaxel resistance in ovarian cancer cells through targeting apaf1', *Nature communications* **7**(1), 1–14.

Yu, F., Pillman, K. A., Neilsen, C. T., Toubia, J., Lawrence, D. M., Tsykin, A., Gantier, M. P., Callen, D. F., Goodall, G. J. & Bracken, C. P. (2017), 'Naturally existing isoforms of mir-222 have distinct functions', *Nucleic acids research* **45**(19), 11371–11385.

Zhao, L., Chen, Q., Li, W., Jiang, P., Wong, L. & Li, J. (2017), 'Mapreduce for accurate error correction of next-generation sequencing data', *Bioinformatics* **33**(23), 3844–3851.