PREMIER Turbo: Probabilistic Error-correction using Markov Inference in Errored Reads using the Turbo principle

(Invited Paper)

Xin Yin*, Zhao Song†, Karin Dorman*‡ and Aditya Ramamoorthy†
*Dept. of Statistics, Iowa State University, Ames, IA 50011
†Dept. of Electrical & Computer Eng. Iowa State University, Ames, IA 50011
‡Dept. of Genetics, Development & Cell Biology Iowa State University, Ames, IA 50011
{xinyin, zhaosong, kdorman, adityar}@iastate.edu

***Abstract*—We present a probabilistic algorithm for error correction for high throughput DNA sequencing data. Our approach leverages our prior algorithm PREMIER where sequencer outputs are modeled as independent realizations of a Hidden Markov Model (HMM) and the problem of error correction is posed as one of maximum likelihood sequence detection over this HMM. In this work we propose an algorithm called PREMIER Turbo which can be viewed as an iterative application of the PREMIER approach. Specifically, we apply error correction in both the forward and the backward directions in a given read. We also present a heuristic inspired by turbo-equalization that incorporates the prior belief on a nucleotide position returned by the Baum-Welch algorithm into the error correction steps. Our approach significantly improves the correction of nucleotides in the beginning of the read. Our test results on the real *C. elegans* and *E. coli* datasets show that PREMIER Turbo achieves a significantly better error correction performance than the other competing methods.**

***Index Terms*—hidden Markov models, DNA sequencing, error correction**

I. Introduction

The advent of novel DNA sequencing platforms has made it feasible to rapidly produce massive amounts of low-cost genomic sequence data [1]. This technology is now an essential tool in many biological and medical studies, with additional applications constantly emerging. Compared with the older Sanger method, the high-throughput capabilities of next-generation sequencing technology are offset by elevated error rates. The dominant observed error varies by platform [2]. For instance, Illumina sequencers almost exclusively make substitution errors, *e.g.* base A called as C, while Ion Torrent machines tend to make insertion/deletion errors, where additional bases are falsely read or valid bases are missed.

Denoising the sequence reads is critical for many downstream analyses, including genome assembly [3] and genetic variation identification [4]. The essence of any error correction scheme is to exploit the high coverage of high-throughput sequencing technology. Each base is sequenced many times, since multiple short reads cover each genome position. Unfortunately, there is no fully reliable alignment information to indicate which reads cover a particular base. In this way, the problem differs from classical error correction.

This work was funded in part by NSF awards DMS-1120597 and CCF-1149860.

Recognizing the importance of error correction in noisy reads, many methods have been developed in recent years to address the problem across various platforms [5]. We briefly review the methods most closely related to our proposed method. When the read length is long, the focus must turn to substrings, called $k$mers of length k, to guarantee sufficient repetition to distinguish error and true bases. Many methods work with $k$mers of just a single length k. It is often assumed that *common* $k$mers, those with high occurrence in the reads, are error-free. Euler [6] corrects a read via the smallest set of corrections that ensures that all $k$mers in the read are common. Reptile [7] and Musket [8] are similar, but more efficient, greedy approaches to make $k$mers common. Hammer [9] identifies $k$mer cliques by linking similar $k$mers, then corrects all members to the clique's consensus $k$mer. Quake [10] iteratively corrects bases by maximizing the a posteriori probability of the true sequence given the observed read until all $k$mers are common. Some methods correct errors in variable-length $k$mers. SHREC [11] uses a suffix trie to correct unusually rare suffixes of common prefixes, while HiTEC [12] uses a suffix array to correct an unlikely base following a common prefix. Both SHREC and HiTEC use a simple probability model to distinguish common from rare suffixes. Two other methods use a probability model to correct a read [13] or $k$mer [14] to the most likely true sequence. All existing $k$mer-based methods, except SHREC, ignore the fact that the $k$mers of a read are not independent observations. Moreover, while some allow arbitrarily complex error models, either all error parameters must be provided *a priori* or the parameter estimation procedure is *ad hoc*.

Approaches based on statistical modeling of the sequencing process have been used for basecalling [15], [16], but not for error correction of reads. In our previous work [17], we presented a hidden Markov model for the DNA sequencer outputs. The reads were corrected by first fitting the HMM model parameters from the observed data and then posing the problem of read correction as one of maximum likelihood sequence detection (MLSD). As the MLSD proves to be

too computationally expensive, we developed low complexity algorithms that outperform competing error correction techniques. In [17], we assumed that the first $k$mer of each read is known.

Main contributions. In this work we propose improved ways to use the HMM for error correction. Specifically, the current model allows for limited errors in the first $k$mer. These errors are corrected by applying our techniques in both forward and the reverse directions of the read. Furthermore, we propose a heuristic based on the turbo principle, whereby the output of the Baum-Welch algorithm (used for fitting HMM parameters) at a given stage is used for error correction in the next stage. Our experiments on *E. coli* and *C. elegans* datasets suggest that the algorithm consistently outperforms other state of the art methods.

II. HMM Modeling

Let $\mathcal{G}$ denote the genome that is being sequenced; $\Omega = \{A, C, G, T\}$ the set of possible bases; $\mathbf{s}$ the unknown true sequence of a fragment of length L from $\mathcal{G}$; $\mathbf{x}$ the sequence reported by the machine for this fragment; and $\mathbf{y}$ the corresponding quality scores. We let $\mathbf{s}_{[i]}$ denote the i-th character of $\mathbf{s}$ and $\mathbf{s}_{[i\ldots j]}$ denotes the substring from position i to j (both $\mathbf{s}_{[i]}$ and $\mathbf{s}_{[j]}$ are included). The tth state (or true $k$mer) is $\mathbf{s}_t = \mathbf{s}_{[t-k+1\ldots t]}$, while $\mathbf{x}_t = \mathbf{x}_{[t-k+1\ldots t]}$ is the observed $k$mer, and $\mathbf{y}_t = \mathbf{y}_{[t-k+1\ldots t]}$ is the corresponding quality score vector. $\mathcal{RC}(\cdot)$ is used to denote the *Reverse Complement* of its argument; for example, the reverse complement for sequence ATT is $\mathcal{RC}(\text{ATT}) = \text{AAT}$. The vector $\boldsymbol{\theta}$ represents all the HMM parameters. The total number of reads to correct is N, and the coverage level is defined as $NL/|\mathcal{G}|$.

We model the sequencer as a HMM; each read is an independent realization of the HMM. A transition is from the state $\mathbf{s}_{t-1}$ to the state $\mathbf{s}_t$. On the tth transition the HMM emits output $(\mathbf{x}_{[t]}, \mathbf{y}_{[t]})$. The model is defined by the following elements.

- State space $\mathcal{K}$, where $|\mathcal{K}| \leq 4^k$.
- Initial state distribution $\pi(\mathbf{s}_t)$ for $\mathbf{s}_t \in \mathcal{K}$.
- Transition distribution $p(\mathbf{s}_{t+1}|\mathbf{s}_t)$, where

$$\sum_{\substack{\boldsymbol{\beta}\in\mathcal{K} \\ \boldsymbol{\beta}[1\ldots k-1]=\boldsymbol{\alpha}[2\ldots k]}} p(\boldsymbol{\beta}|\boldsymbol{\alpha}) = 1, \quad \forall \boldsymbol{\alpha} \in \mathcal{K}.$$

- The *d-neighborhood* of observed $k$mer $\mathbf{x}_t$

$$\mathcal{N}^d(\mathbf{x}_t) = \{\boldsymbol{w} : \boldsymbol{w} \in \mathcal{K} \text{ and } D(\mathbf{x}_t, \boldsymbol{w}) \leq d\}, \quad (1)$$

  where $D(\cdot,\cdot)$ is the Hamming distance function.
- Emission distribution,

$$f_t(\mathbf{x}_t, \mathbf{y}_t \mid \mathbf{s}_t) = q_t(\mathbf{y}_{[t]} \mid \mathbf{x}_{[t]}, \mathbf{s}_{[t]}) g_t(\mathbf{x}_t \mid \mathbf{s}_t), \quad (2)$$

  where we assume these simple forms for $k < t \leq L$:

$$q_t(\mathbf{y}_{[t]} \mid \mathbf{x}_{[t]}, \mathbf{s}_{[t]}) = \begin{cases} q_{t0}(\mathbf{y}_{[t]}) & \mathbf{x}_{[t]} = \mathbf{s}_{[t]}, \\ q_{t1}(\mathbf{y}_{[t]}) & \mathbf{x}_{[t]} \neq \mathbf{s}_{[t]}, \text{ and} \end{cases}$$

$$g_t(\mathbf{x}_t \mid \mathbf{s}_t) \propto \mathbb{1}\{\mathbf{s}_t \in \mathcal{N}^d(\mathbf{x}_t)\} g_{t0}(\mathbf{x}_{[t]} \mid \mathbf{s}_{[t]}). \quad (3)$$

  For the first $k$mer, when $t = k$, we assume that there is at most one error. Furthermore, the first k bases and quality scores are emitted independently,

$$f_k(\mathbf{x}_k, \mathbf{y}_k \mid \mathbf{s}_k) \propto \mathbb{1}\{\mathbf{s}_k \in \mathcal{N}^1(\mathbf{x}_k)\} \times \prod_{t=1}^{k} q_t(\mathbf{y}_{[t]} \mid \mathbf{x}_{[t]}, \mathbf{s}_{[t]}) g_{t0}(\mathbf{x}_{[t]} \mid \mathbf{s}_{[t]}). \quad (4)$$

A restricted version of the above model was presented in [17], where the first $k$mer was assumed to be known. We briefly overview the rationale for our proposed model below. Local dependence in the genome $\mathcal{G}$ is known to exhibit Markovian structure [18], and we leverage this in our model.

For $t > k$, the conditional distribution $g_{t0}(\mathbf{x}_{[t]} \mid \mathbf{s}_{[t]})$ models the probability of emitting $\mathbf{x}_{[t]}$ given that the correct base is $\mathbf{s}_{[t]}$. Note that for each value of $\mathbf{s}_{[t]}$, this is a p.m.f. supported on four symbols and thus has three independent parameters. We expect that the quality score distribution $q_{t0}(q)$ is shifted right of $q_{t1}(q)$, for all t, because the sequencer should assign higher quality scores to error free bases. The emission distribution is further constrained by only allowing $k$mers that lie within a Hamming distance, d, of $\mathbf{s}_t$ to have non-zero emission distributions. Note that for a given $\mathbf{s}_t$ this constraint may not allow the emission of all possible bases. For instance, if there is no $k$mer apart from $\mathbf{s}_t$ in $\mathcal{K}$ within its d-neighborhood, then the only possible emission is $\mathbf{s}_{[t]}$ itself. Thus, the distribution $g_t(\mathbf{x}_t \mid \mathbf{s}_t)$ is obtained by suitably normalizing the RHS of eq. (3). The Illumina sequencer is known to have very few errors in the initial part of the read. Accordingly, we allow at most one error in the first $k$mer.

For computational tractability we restrict the state space $\mathcal{K}$ to be only the set of observed $k$mers, because even for moderate k around 16, 4^k is too big to have a tractable model. Even though $\mathcal{K}$ includes erroneous $k$mers, we hope to identify them during estimation of the HMM. The choice of k depends on two conflicting requirements. On the one hand, we want an accurate model. A small value of k, for instance $k = 3$, will likely result in a $k$mer such as ATC existing in several locations in the original $\mathcal{G}$. In the error correction phase, the sequence detection may result in miscorrections in this situation. For example, if $ATCG$ occurs twice and $ATCT$ occurs once, then true read $ATCT$ may be erroneously corrected to $ATCG$. On the other hand, very large k (though k cannot exceed L), may lead to decreased $k$mer coverage and create problems for parameter estimation. In our experiments, we choose k to optimize performance for our method and all competing methods.

The model parameters for the HMM are fit by using the penalized estimation method that is described in more detail in [17]. The method applies an approximate ℓ_0-like penalty on the transition probabilities so that small transition probabilities are driven to zero to enforce our belief that most $k$mers are unique in genome $\mathcal{G}$. The transition probabilities $p(\boldsymbol{\beta}|\boldsymbol{\alpha})$ are initialized by counting the incidence, $n(\boldsymbol{\alpha}, \boldsymbol{\beta})$, of $k$mer $\boldsymbol{\alpha}$ followed by $\boldsymbol{\beta}$ in all reads. Then we set,

$$p^{(0)}(\boldsymbol{\beta}|\boldsymbol{\alpha}) = \frac{n(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\sum_{\boldsymbol{\beta'}\in\mathcal{K}} n(\boldsymbol{\alpha}, \boldsymbol{\beta'})}, \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{K}, \boldsymbol{\beta}_{[1\ldots k-1]} = \boldsymbol{\alpha}_{[2\ldots k]}. \quad (5)$$

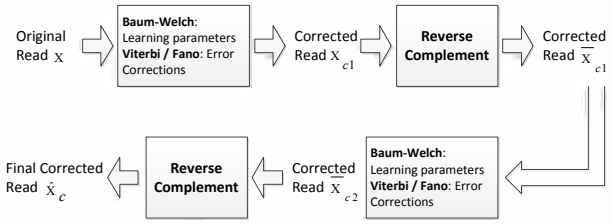

Fig. 1. The forward-backward error correction scheme using reverse complement information

For emissions, we initialize,

$$\begin{aligned} q_{tj}^{(0)}(q) &= 1/Q_{max}, & q \in \{0,1,\ldots,Q_{\max}\} \\ g_{t0}^{(0)}(\beta|\beta') &= \begin{cases} 1-p_e & \text{if } \beta=\beta' \\ p_e/3 & \text{otherwise} \end{cases}, & \beta,\beta' \in \Omega, \end{aligned} \tag{6}$$

where Q_{max} is the maximum quality score, $j \in \{0,1\}$ is an indicator for error, $t \in \{1,2,\ldots,L\}$ is the read position, and p_e is an estimated, average per base error rate. In this paper, we used $p_e = 0.01$.

III. Error Correction Algorithms

The parameters of the HMM are fit based on the penalized estimation procedure discussed in [17]. Then, we attempt to find the maximum likelihood $\mathbf{s}$ that explains the observations $(\mathbf{x},\mathbf{y})$ for each read. In [17] we presented two algorithms, namely A-Viterbi and Fano, that approximate the performance of the optimal Viterbi algorithm. In this work, we propose improved algorithms for error correction using the HMM.

A. Forward and backward error correction

Our first improvement is to use repeated rounds of error correction. Specifically as depicted in Fig. 1, after the original read $\mathbf{x}$ is corrected in the first iteration, producing $\mathbf{x}_{c1}$, we take the reverse complement of $\mathbf{x}_{c1}$ to obtain $\bar{\mathbf{x}}_{c1}$ (the quality score sequence $\mathbf{y}$ is simply reversed). Note that applying this conversion flips the order of nucleotides and $k$mers, *i.e.*, the first $k$mer in $\bar{\mathbf{x}}_{c1}$ is the reverse complement of the last $k$mer in $\mathbf{x}_{c1}$. The error correction procedure is now applied on the reverse complemented data. Such a conversion is useful in our iterative approach since when we employ Baum-Welch and Viterbi/Fano algorithms on $\bar{\mathbf{x}}_{c1}$, the reverse strand supplies upstream context information for correcting errors, and possibly more than one error, in the first $k$mer of $\mathbf{x}$.

B. A turbo equalization inspired heuristic

Note that the Baum-Welch algorithm not only returns the parameter estimate $\hat{\boldsymbol{\theta}}$ but also the marginal posterior probability of the true base at every position [19]. Let

$$\psi_t(\mathbf{s}_t) = \begin{cases} p(\mathbf{s}_k|\mathbf{x},\mathbf{y},\boldsymbol{\theta}), \text{ if } t=k \\ p(\mathbf{s}_{[t]}|\mathbf{x},\mathbf{y},\boldsymbol{\theta}), \text{ if } k<t\le L. \end{cases} \tag{7}$$

When applying the iterative approach, we have observed improved results with the sequential decoding based Fano algorithm, if the path metric (refer to [17] for details) is augmented with an additive term $\log_2\left[\psi_{t+1}(\mathbf{s}_{t+1})\right]$ from the previous iteration. Specifically, after the first forward iteration depicted in Fig. 1 is run, we have the terms corresponding to eq. (7) on all the first $k$mers and the subsequent bases. For the next backward iteration, we fit the model using the Baum-Welch algorithm. Following this, when the Fano algorithm is utilized for correcting a given read, the path metric at any stage has an additional term corresponding to eq. (7). This can be treated as a turbo-equalization based heuristic where the information from the previous round is treated as an independent belief about the basecall.

IV. Experimental Results and Methodology

We compared the performance of our proposed methods with the following previously published methods: HiTEC (1.0.2), Reptile (1.0.1), Shrec (2.2), Quake (0.3) and Musket (1.0.7), using two benchmark datasets generated from real Illumina sequencing projects. Their summary statistics are outlined in Table I.

The "ground truth" sequencing errors were identified using short-read alignment. We aligned the *E. coli* (Accession ERR022075) and the *C. elegans* (Accession SRR065390) to their reference genomes (NC.000913 and PRJNA13758 release WS236, respectively) using Burrows Wheeler Aligner (BWA). We ran the aligner with default parameters, while setting the maximum edit distance to 10 for both datasets. Next, we kept reads that (1) uniquely map to certain genomic regions on the reference genomes (1.0Mbp–1.5Mbp for *E. coli*, 1Mbp–3Mbp on *Chr I* for *C. elegans*); (2) do not contain insertion or deletion errors, as reported by BWA; and (3) do not contain undetermined "N" bases. Finally, we discarded reads without mate-pairs. The reads in the resulting benchmark set fulfill the requirements of all error correction methods used in this study. Errors were tallied as mismatches between the selected reads and the reference. At the end of this process the maximum number of errors in an observed read was 29.

A. Software parameter tuning

For HiTEC, the genome length and error rate parameters were set to the true values as in Table I. Quake was run with $k$mer length $k = 13$ for *E. coli* dataset, and $k = 14$ for *C. elegans* dataset, values selected using the guideline provided in [10]. Several neighboring k for Quake were tested as well. Quake failed to estimate the model parameters for some k, but among those that worked, performance did peak at the recommended k. SHREC was run with default parameters, as no instructions were provided on tuning the parameters. For k-spectrum based methods that do not automatically determine $k$mer length, such parameters (k for HMM and Musket, or $(k, step)$ for Reptile) were optimized using a grid-search approach ($13 \le k^h \le 19, 13 \le k^m \le 21$, for the *E. coli* data, and $19 \le k^h \le 25, 17 \le k^m \le 27, 9 \le k^r = step^r \le 13$, where the superscripts h, m and r denote the method HMM, Musket and Reptile respectively). The optimal values found were $k^h = 16, k^m = 16, k^r = step^r = 9$ for the *E. coli* dataset, as well as $k^h = 21, k^m = 23, k^r = step^r = 11$ for *C. elegans* dataset. The model complexity parameter d, denoting maximum errors allowed per kmer, is set to $d = 8$ for both HMM and Musket, and $d = 4$ for Reptile. Reptile makes additional assumptions about the distribution of errors

TABLE I
BENCHMARK SEQUENCING DATASETS

Dataset	Region length	Read length	Number of reads	Coverage	Error rate (%)
C. elegans (SRR●6539●)	2000000	100 bp	975984	49x	0.51
E. c●li (ERR●22●75)	500000	100 bp	4812400	963x	0.58

TABLE II
ERROR CORRECTION RESULTS FOR *C. elegans* 100BP DATASET

	ce	ζ	fa	η
Fanop	462639 (444027)	0.9257 (0.8884)	16532 (12694)	0.8926 (0.8630)
A-Viterbi	462394 (444134)	0.9252 (0.8887)	15291 (12030)	0.8946 (0.8646)
Musket	447088	0.8946	11402	0.8718
HiTEC	451225	0.9028	73219	0.7563
Shrec	239590	0.4794	591795	-0.7047
Quake	84275	0.1686	3260	0.1621
Reptile	386217	0.7728	57310	0.6581

in a $k$mer, but $d = 4$ for Reptile is as close to $d = 8$ for other methods as we can get given these restrictions.

For given k and d, the maximum likelihood estimates of the HMM parameters were estimated by the Baum-Welch algorithm with penalized likelihood as described in [17]. We fixed the penalty parameter γ at $\gamma = 10^{-18}$ to approximate an ℓ_0-type penalty while avoiding numerical underflow. We then computed λ so the ℓ_0 penalty (nearly) zeros out $k$mers with expected occurrence counts less than the first valley of the $k$mer coverage histogram. (For detailed discussion of such thresholds, see [12].) Using this strategy, the λ values for our datasets were set to $\lambda = 600$ for *E. coli* and $\lambda = 125$ for *C. elegans*. The Fano algorithm also requires the specification of the bias parameter and step size. While the performance of the algorithm is somewhat insensitive to step size, the bias parameter has to be set based on an inspection of the statistics of increments of the log-likelihood of the corrected paths returned by the A-Viterbi algorithm (owing to space limitations we are unable to discuss this point in detail).

B. Discussion of results

We summarized the error correction results in Table II and III, and the methods can be compared on the basis of following metrics: ce, number of errors correctly recovered; the sensitivity $\zeta \triangleq ce/e$, where e as the total number of errors; number of new errors falsely introduced, fa; and gain $\eta \triangleq (ce - fa)/e$, measuring the effectiveness of error correction. The method Fanop refers to the algorithm discussed in section III-B.

The results in parentheses indicate the result of the forward iteration for the HMM (and correspond to the approach in [17]). It can be noted that the second iteration provides an additional 3% improvement in the *C. elegans* dataset and a 1% improvement in the *E. coli* dataset. Though not shown in the table, we also noted an improvement of over 2% for the Fanop approach for the *C. elegans* dataset when compared to the original Fano algorithm of [17]. It is worth noting that our approaches consistently outperform the other competing methods, with a greater edge on more complex genomes like *C. elegans*. A-Viterbi and Fano perform almost equally well, presumably because setting $d = 8$ permits A-Viterbi to explore most of the relevant 4-ary branches in the trellis.

TABLE III
ERROR CORRECTION RESULTS FOR *E. coli* 100BP DATASET

	ce	ζ	fa	η
Fanop	2747787 (2712971)	0.9802 (0.9678)	3104 (2716)	0.9791 (0.9668)
A-Viterbi	2756519 (2721929)	0.9833 (0.9710)	3026 (2708)	0.9822 (0.9700)
Musket	2738171	0.9768	1000	0.9764
HiTEC	2541891	0.9068	1573	0.9062
Shrec	2669575	0.9523	1970	0.9516
Quake	2041845	0.7284	215	0.7283
Reptile	2704302	0.9647	6722	0.9623

REFERENCES

[1] M. L. Metzker, "Sequencing technologies - the next generation." *Nat Rev Genet*, vol. 11, pp. 31–46, 2010.

[2] N. J. Loman, R. V. Misra, T. J. Dallman, C. Constantinidou, S. E. Gharbia, J. Wain, and M. J. Pallen, "Performance comparison of benchtop high-throughput sequencing platforms." *Nat Biotechnol*, vol. 30, pp. 434–439, 2012.

[3] S. L. Salzberg, A. M. Phillippy, A. Zimin, D. Puiu, T. Magoc, S. Koren, T. J. Treangen, M. C. Schatz, A. L. Delcher, M. Roberts, G. Marais, M. Pop, and J. A. Yorke, "Gage: A critical evaluation of genome assemblies and assembly algorithms." *Genome Res*, vol. 22, pp. 557–567, 2012.

[4] A. Wilm, P. P. K. Aw, D. Bertrand, G. H. T. Yeo, S. H. Ong, C. H. Wong, C. C. Khor, R. Petric, M. L. Hibberd, and N. Nagarajan, "Lofreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets." *Nucleic Acids Res*, vol. 40, pp. 11 189–11 201, 2012.

[5] X. Yang, S. P. Chockalingam, and S. Aluru, "A survey of error-correction methods for next-generation sequencing." *Brief Bioinform*, 2012.

[6] M. J. Chaisson, D. Brinza, and P. A. Pevzner, "De novo fragment assembly with short mate-paired reads: Does the read length matter?" *Genome Res*, vol. 19, pp. 336–346, 2009.

[7] X. Yang, K. Dorman, and S. Aluru, "Reptile: representative tiling for short read error correction," *Bioinformatics*, vol. 26, pp. 2526–2533, 2010.

[8] Y. Liu, J. Schröder, and B. Schmidt, "Musket: a multistage k-mer spectrum based error corrector for illumina sequence data." *Bioinformatics*, vol. 29, pp. 308–315, 2013.

[9] P. Medvedev, E. Scott, B. Kakaradov, and P. Pevzner, "Error correction of high-throughput sequencing datasets with non-uniform coverage." *Bioinformatics*, vol. 27, pp. i137–i141, 2011.

[10] D. R. Kelley, M. C. Schatz, and S. L. Salzberg, "Quake: quality-aware detection and correction of sequencing errors." *Genome Biol*, vol. 11, p. R116, 2010.

[11] J. Schröder, H. Schröder, S. J. Puglisi, R. Sinha, and B. Schmidt, "Shrec: a short-read error correction method." *Bioinformatics*, vol. 25, pp. 2157–2163, 2009.

[12] L. Ilie, F. Fazayeli, and S. Ilie, "HiTEC: accurate error correction in high-throughput sequencing data." *Bioinformatics*, vol. 27, pp. 295–302, 2011.

[13] E. Wijaya, M. C. Frith, Y. Suzuki, and P. Horton, "Recount: expectation maximization based error correction tool for next generation sequencing data." *Genome Informatics*, vol. 23, pp. 189–201, 2009.

[14] X. Yang, S. Aluru, and K. S. Dorman, "Repeat-aware modeling and correction of short read errors." *BMC Bioinformatics*, vol. 12 Suppl 1, p. S52, 2011.

[15] S. Das and H. Vikalo, "Onlinecall: fast online parameter estimation and base calling for illumina's next-generation sequencing," *Bioinformatics*, vol. 28, pp. 1677–1683, 2012.

[16] T. Wu and H. Vikalo, "Joint parameter estimation and base-calling for pyrosequencing systems," *IEEE Trans. Signal Process.*, vol. 60, pp. 4376 –4386, Aug. 2012.

[17] X. Yin, Z. Song, K. S. Dorman, and A. Ramamoorthy, "PREMIER - PRobabilistic Error-correction using Markov Inference in Errored Reads," in *Proc. IEEE Intl. Symp. on Info. Theory*, July 2013, pp. 1626–1630.

[18] E. Picardi and G. Pesole, "Computational methods for ab initio and comparative gene finding," *Methods Mol. Biol.*, vol. 609, pp. 269–284, 2010.

[19] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257 –286, 1989.