Bioinformatics Advance Access published May 17, 2011

Sensitive and fast mapping of di-base encoded reads

Farhad Hormozdiari [1]*, Faraz Hach [2]*, S. Cenk Sahinalp [2], Evan E. Eichler [1], Can Alkan [1]†

[1]Department of Genome Sciences, University of Washington, and Howard Hughes Medical Institute, Seattle, WA, USA
[2]Simon Fraser University, School of Computing Science, Burnaby, BC, Canada

Associate Editor: Dr. Alex Bateman

ABSTRACT

Motivation: Discovering variation among high throughput sequenced genomes relies on efficient and effective mapping of sequence reads. The speed, sensitivity and accuracy of read mapping are crucial to determining the full spectrum of single nucleotide variants (SNVs) as well as structural variants (SVs) in the donor genomes analyzed.

Results: We present *drFAST*, a read mapper designed for di-base encoded "color-space" sequences generated with the ABI SOLiD platform. *drFAST* is specially designed for better delineation of structural variants, including segmental duplications, and is able to return *all* possible map locations and underlying sequence variation of short reads within a user-specified distance threshold. We show that *drFAST* is more sensitive in comparison to all commonly used aligners such as Bowtie, BFAST, and SHRiMP. *drFAST* is also faster than both BFAST and SHRiMP and achieves a mapping speed comparable to Bowtie.

Availability: The source code for *drFAST* is available at http://drfast.sourceforge.net

Contact: calkan@u.washington.edu

1 INTRODUCTION

Genomic variation between individuals or across species ranges from single nucleotide polymorphisms (SNPs) and structural variation to larger chromosomal rearrangements Alkan *et al.* (2011). Thanks to the improvements in sequencing technologies, large-scale genome variation studies such as the 1000 Genomes Project 1000 Genomes Project Consortium (2010); Mills *et al.* (2011) have made it possible to better characterize normal human genomic variation and disease Ng *et al.* (2010); Vissers *et al.* (2010); Lupski *et al.* (2010).

The development of high throughput sequencing (HTS) technologies has changed the landscape of genome research. The first commercially available HTS technology was from Roche/454 Life Sciences Margulies *et al.* (2005) and was used to sequence the genome of James Watson Wheeler *et al.* (2008). It was followed by other "second generation" sequencing platforms that generate orders of magnitude more data for a fraction of the cost, such as Illumina Genome Analyzer Bentley *et al.* (2008) and AB SOLiD McKernan *et al.* (2009). Third generation sequencing platforms are now under development, and HeliScope Pushkarev *et al.* (2009) and PacBio RS Eid *et al.* (2009) were recently made available however, for the time being, they produce reads with higher error rates.

Analysis of genomic variation using sequencing starts with mapping the randomly sheared and ideally uniformly sampled DNA fragments from the genome. Different properties and error models of sequence reads generated by these technologies require the development of specialized read mapping algorithms for each platform for accurate read alignment and characterization of genomic variants. This becomes more complicated for short reads: due to repeats and duplications in genomes, they can map to multiple locations with equal sequence identity. Leveraging the high sequence coverage and randomly selecting one "best" location when a read cannot be unambiguously placed has proven to be effective in discovering SNPs and and small indels in relatively non-complex areas of the genome Li *et al.* (2008a); Li and Durbin (2009). However, structural variation detection sensitivity is shown to benefit from tracking all map locations of the reads including suboptimal alignments Hormozdiari *et al.* (2009); Mills *et al.* (2011); Lee *et al.* (2009) and characterization of segmental duplications is extremely resistant against mapping the reads uniquely Alkan *et al.* (2009); Sudmant *et al.* (2010).

Read mappers can be broadly classified into two categories according to the method used to index the reference genome using either hash tables or suffix arrays (compressed through the Ferragina-Manzini index Ferragina and Manzini (2000) with the use of the Burrows-Wheeler Transform Burrows and Wheeler (1994)). Hash-based aligners such as MAQ Li *et al.* (2008a), SHRiMP Rumble *et al.* (2009), mrFAST Alkan *et al.* (2009), mrsFAST Hach *et al.* (2010), and BFAST Homer *et al.* (2009) have poorer performance in comparison to suffix array based aligners (e.g. BWA Li and Durbin (2009), Bowtie Langmead *et al.* (2009)) when dealing with short reads; however their relative performance increases considerably and surpasses the suffix array based aligners when the read length and thus the number of errors (mismatches or indels) that need to be tolerated increase.

In this paper we describe a hash-based read mapping algorithm named "di-base read fast alignment search tool" (*drFAST*) designed for the di-base encoded color-space reads generated with the SOLiD platform. The main advantage of di-base encoding is increased base call accuracy due to each base being represented by two "colors". This helps in differentiating base calling errors from real sequence

*Joint First Authors
†corresponding author. Email: calkan@u.washington.edu

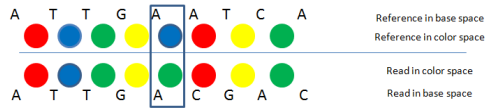

Fig. 1. Translating the read from color-space to base space may result in a new sequence different from the original read if there exists a base call error.

variance, therefore increasing the reliability of detected genomic variants. We show that mapping speed of *drFAST* is higher than other SOLiD-enabled hash-based read mappers, BFAST Homer *et al.* (2009) and SHRiMP Rumble *et al.* (2009), and comparable to suffix array based aligner Bowtie Langmead *et al.* (2009). In addition, *drFAST* was able to map more reads than all the tools we benchmarked. Furthermore, *drFAST* achieves 100% sensitivity if the maximum allowed edit distance is less than L/k, where L is the sequence length and k is the length of the k-mers stored in the hash table (k is set to 12 by default). Coupled with its ability to store all map locations within a user-specified distance threshold and its paired-end mapping capabilities, *drFAST* can be used to characterize segmental duplications Alkan *et al.* (2009); Sudmant *et al.* (2010) and increase sensitivity of structural variation discovery using VariationHunter Hormozdiari *et al.* (2009, 2010).

2 METHODS

For each read sequenced from a donor genome, a mapping algorithm aims to find locations in a "reference genome" where the read can be aligned exactly or within a small number of errors in the form of substitutions or insertions/deletions (indels). *drFAST* is a read mapper designed for color-space reads generated with the SOLiD platform, and finds *"all"* possible map locations for each read of length r in the reference genome within a user-specified e mismatches.

drFAST is a seed-and-extend type algorithm and it builds an index of the reference genome by creating a collision-free hash table for all subsequences of length k (k-mers) of the reference genome. To map the reads, it first partitions each read to $(e+1)$ k-mers and searches for each of these k-mers in the hash table. For each *hit* in the hash table, it then tests if the remainder of the read can be "extended" by aligning to the reference genome starting at the determined hit location.

How exactly this is done is described below:

2.1 Genome transformation

The sequence data produced with the SOLiD platform are in *color-space* format ($S = \{0,1,2,3\}^*$), where the reference genome sequence is in *letter-space* (i.e. $R = \{A,C,G,T\}^*$). Each color encodes two adjacent base pairs in the read, and each base pair is represented by two colors. Transformation of reads from color-space to letter-space before mapping may result in generating incorrect reads where base call errors exist, as depicted in Figure 1. To avoid such incorrect decoding of reads, we translate the reference genome to color-space and use this transformed genome to create the index.

2.2 Indexing the reference genome

drFAST creates a collision-free hash table for all k-mers in the reference genome. Each entry of this index is a 2-tuple $\tau = (s, L)$, where s is a k-mer from the genome ($k = 12$ by default) and L is a list of all positions of the genome starting with this subsequence. The index is maintained in lexicographically sorted order with respect to their subsequences. For a reference genome of length n, the upper bound for the size of its index is $O(n)$; but due to the repetitive nature of genome sequences, the index size is smaller in practice.

2.3 Indexing the reads

drFAST partitions each read of length r into $e+1$ non-overlapping blocks of length k where e is the user-specified maximum Hamming distance allowed for mapping. In the case where $k \leq \lfloor r/(e+1) \rfloor$ the pigeon hole principle guarantees that at least one of these blocks maps to the reference genome with no errors. Similar to the indexing described in Section 2.2, *drFAST* creates an index of blocks computed from all reads in 2-tuples $\tau_r = (s, L_r)$, where L_r denotes the list of reads that include the k-mer s.

2.4 Searching

drFAST compares the reference genome index keys with read index keys to find the locations in the reference genome where a read can be mapped with at most e errors. For each partition of the read, *drFAST* first finds the locations of the reference genome with the identical subsequence (same keys). It then tries to extend the location through sequence alignment of the reads to the genome, and reports those locations where the Hamming distance of the alignment is at most e. A simple loop scans both indices (both are lexicographically sorted); if the keys of the indices are the same (same subsequence) for entries $\tau = (s, L)$ in the reference and $\tau_r = (s, L_r)$ in the read index. Then all entries in L are candidate map locations for each read entry in L_r, thus the entire list L should be compared to L_r (extending step).

Similar to mrsFAST Hach *et al.* (2010), *drFAST* performs *"all-to-all"* list comparison using a recursive divide-and-conquer strategy that guarantees cache obliviousness; i.e. asymptotically minimizing the number of costly cache misses Frigo *et al.* (1999).

2.5 Extending

The final step is to verify if each read can be aligned to candidate map locations within the user-specified error threshold e. *drFAST* aims to align the color-space read (S_c) to the letter-space sequence (S_l). The aligning process can be considered as finding a letter-space read S'_l that aligns to S_l, and highly similar to S_c if transformed to color-space:

$$argmax_{S'_l} \left(Sim(S_l, S'_l) + Sim(S_c, CCG(S'_l)) \right) \qquad (1)$$

where CCG is the function that transforms the letter-space to color-space as defined by the SOLiD technology, and Sim is the similarity function.

Maximizing the similarity between two sequences is equivalent to minimizing their distance. We use Hamming distance (i.e. the number of mismatches) as the distance measure between two sequences.

$$argmin_{S'_l} \left(Diff(S_l, S'_l) + Diff(S_c, CCG(S'_l))\right) \quad (2)$$

To address the problem, *drFAST* introduces two efficient methods.

2.5.1 Method I: Dynamic Programming. Let $\sum = \{A, C, G, T\}$, and $\sigma, \sigma' \in \sum$, and let $Score(i, \sigma)$ indicate the optimal alignment of two subsequences $S_l[1..i]$ and $S_c[1..i]$ (from the first to the i^{th} character) while σ is the last character of S'_l. We then define

$$Score(i, \sigma) = {}_{d(S_l[i],\sigma)+min_{\sigma'}\{Score(i-1,\sigma')+d(S_c[i],CCG(\sigma'\sigma))\}} \quad (3)$$

where $d(a, b) = 1$ if $a \neq b$, and $d(a, b) = 0$ otherwise.

The detailed version of equation 3 is as follows:

$$Score(i, \texttt{'A'}) = d(S_l[i], \texttt{'A'}) + min \begin{cases} Score(i-1,\texttt{'A'})+d(S_c[i],\texttt{'0'}) \\ Score(i-1,\texttt{'C'})+d(S_c[i],\texttt{'1'}) \\ Score(i-1,\texttt{'G'})+d(S_c[i],\texttt{'2'}) \\ Score(i-1,\texttt{'T'})+d(S_c[i],\texttt{'3'}) \end{cases}$$

$$Score(i, \texttt{'C'}) = d(S_l[i], \texttt{'C'}) + min \begin{cases} Score(i-1,\texttt{'A'})+d(S_c[i],\texttt{'1'}) \\ Score(i-1,\texttt{'C'})+d(S_c[i],\texttt{'0'}) \\ Score(i-1,\texttt{'G'})+d(S_c[i],\texttt{'3'}) \\ Score(i-1,\texttt{'T'})+d(S_c[i],\texttt{'2'}) \end{cases}$$

$$Score(i, \texttt{'G'}) = d(S_l[i], \texttt{'G'}) + min \begin{cases} Score(i-1,\texttt{'A'})+d(S_c[i],\texttt{'2'}) \\ Score(i-1,\texttt{'C'})+d(S_c[i],\texttt{'3'}) \\ Score(i-1,\texttt{'G'})+d(S_c[i],\texttt{'0'}) \\ Score(i-1,\texttt{'T'})+d(S_c[i],\texttt{'1'}) \end{cases}$$

$$Score(i, \texttt{'T'}) = d(S_l[i], \texttt{'T'}) + min \begin{cases} Score(i-1,\texttt{'A'})+d(S_c[i],\texttt{'3'}) \\ Score(i-1,\texttt{'C'})+d(S_c[i],\texttt{'2'}) \\ Score(i-1,\texttt{'G'})+d(S_c[i],\texttt{'1'}) \\ Score(i-1,\texttt{'T'})+d(S_c[i],\texttt{'0'}) \end{cases}$$

As shown in Figure 2 each column is dependent only to the previous one. This property not only helps to formulate the problem using dynamic programming but also enables us to calculate the value in each column simultaneously. If the value of column i^{th} is loaded to an SSE register available in commodity hardware, we can compute the value of the $(i+1)^{th}$ column through a single CPU instruction.

The minimum value of $Score(|S_l|,' A')$, $Score(|S_l|,' C')$, $Score(|S_l|,' G')$, and $Score(|S_l|,' T')$ is the score of the best translation of S_c to S'_l.

Figure 3 shows an example of aligning a letter-space and a color-space sequence using the dynamic programming described in equation 3. The minimum value in the last column represents the score of the best alignment. Using the backtracking pointer, we can then recover the best alignment sequence.

REMARK 1. *The dynamic programming formulation in Equation (3) will find the optimal solution to the objective function in Equation (2) if the costs of mismatches and read errors are equal to one.*

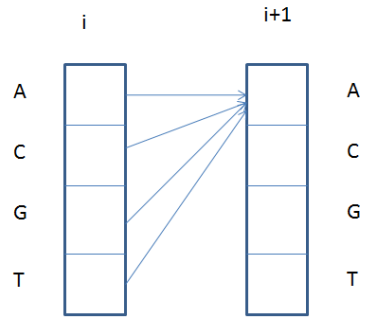

Fig. 2. Computation of $Score(i + 1,' A')$ is only dependent to column i, which is computed in the previous step.

	A	T	T	G	A	A	T	C	A
A	0	2	2	2	0	1	3	3	1
C	1	2	2	2	2	1	3	1	3
G	1	2	2	0	2	2	2	3	3
T	1	0	0	2	2	2	1	3	3

Fig. 3. The dynamic programming table generated to align ATTGAATCA and 30121321 (0=blue, 1=green, 2=yellow, 3=red). The arrows represent the best alignment between the two sequences.

REMARK 2. *Dynamic programming described in (3) can be modified to handle any cost function for mismatches and read errors.*

Note that the equation (3) uses Hamming distance but it can be easily generalized for edit distance to allow indels.

$$Score(i, j, \sigma) = min \begin{cases} Score(i-1,j-1,\sigma')+d(S_c[j],CGG(\sigma\sigma'))+d(S_l[i],\sigma) \\ Score(i-1,j,\sigma')+d(S_l[i],\sigma) \\ Score(i,j-1,\sigma')+d(S_c[j],CGG(\sigma\sigma')) \end{cases}$$

2.5.2 Method II: Transformation Based Detection. The second method is based on the theoretical design aspect of color-space reads McKernan *et al.* (2009). A string of colors $c_1c_2c_3 \ldots c_k$ can also be treated as transformations. For example, $C102$ can be written as $f_2(f_0(f_1(C)))$ where the transformation of the colors is applied one after the other. This specific transformation converts C to G, acting as color 3 ($C102 = C3 = G$). For any other base pair, color string 102 will behave exactly as color 3.

The set of color operations is isomorphic to the "Klein Four Group" Armstrong (1988); McKernan *et al.* (2009). The Klein Four Group is the symmetry group of a rectangle, which has four elements: the identity, the vertical reflection, the horizontal reflection, and a 180 degree rotation. In other words, given the four bases in the corners of a rectangle, each color operation has a one-to-one correspondence with one of the Klein Group elements (see Table 2.5.2). The Klein Four Group is closed under its elements meaning that if a b are two elements of this group $a \oplus b$ and $b \oplus a$ ($a \oplus b$ means a followed by b) is also an element of the this group. It also has associative, identity, reverse and commutative properties. This means that any sequence of color operations can be considered as one color operation.

(a)		(b)	
A	C	T	G
G	T	C	A

Table 1. Applying color transformation '3' (a) is the same as applying 180^o rotation (b).

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Table 2. Addition Table Code for Strings of Colors

We use this property of the color-space reads to detect mismatches. Let two sets of color operations of the same length exist ($c_1 \dots c_k$ and $r_1 \dots r_k$) with different starting color ($c_1 \neq r_2$). For both sets, if any two consecutive colors are replaced with their equivalent (closure property) starting from left hand side, you will end up with one at the end. If the last color matches with no intermediate matching colors then these two operations show a mismatch of length $k-1$. To illustrate this, consider two color operations 313 and 100. For simplicity, we also consider a leading base C. After applying the color operations, strings GTA and AAA will be generated respectively. It can be seen that the last base pair generated using both operations is A and intermediary base pairs are not matching. These two sets of operations have the same transformation, thus although they generate different sets of base pairs in middle, the final "product" is the same character.

THEOREM 3. *Let $c = c_1c_2c_3c_k$ be a k-color substring of a read aligned with the corresponding color-space reference $r = r_1r_2r_3r_k$. Then c encodes an isolated $(k-1)$-base change if and only if the base position preceding c is not a variant, and the following two equations hold under the Color Addition Table 2.5.2:*

$$\sum_{j=1}^{k} c_j = \sum_{j=1}^{k} r_j$$

For all i from 1 to $k-1$:

$$\sum_{j=1}^{i} c_j \neq \sum_{j=1}^{i} r_j$$

We use Theorem 3 as the basis of our validation function (i.e. extending step). If there is a color mismatch between the read and the reference genome, we consider the next $2e$ colors to test if there exists any same color transformation of size at most $2e$ between the read and the genome. Considering a window of limited length, this sometimes may cause incorrect classification of a long stretch of mismatches as two independent read error calls. We refine such calls at the final step.

3 ADDITIONAL FEATURES

Parallelization: An *embarrassingly-parallel* wrapper for *drFAST* can be easily written to split the reads into smaller "chunks" ($\sim$1-5 million reads per file) and align on cluster nodes. This approach is the best practice because:
1) *drFAST* requires $<$ 700 MB to load the genome and its index and only a total of $\sim$ 1.3 GB of memory to map 1 million reads to the genome.
2) Mapping of each read is independent from mapping the others (except in the case of paired-end sequences where both ends need to be processed in the same chunk).

Paired-end Mapping: SOLiD, like most other HTS technologies can generate paired-end (PE) sequences. A pair of PE sequences are generated from the prefix and suffix of the same sheared DNA fragment, thus they can be used to increase mapping accuracy, and discover structural variation Alkan *et al.* (2011); Mills *et al.* (2011). Current implementation of *drFAST* supports tracking the paired-end information, enabling direct use of VariationHunter for structural variation Hormozdiari *et al.* (2009) and transposon insertion Hormozdiari *et al.* (2010) discovery, as well as NovelSeq Hajirasouliha *et al.* (2010) for characterization of novel sequence insertions.

4 RESULTS

To measure the performance of *drFAST*, we compared its two variants to popular color-space read mappers currently available.

Benchmarked Software:

- *drFAST*-DP (Dynamic programming variant) (version 0.0.0.2);
- *drFAST*-CT (Color transformation variant) (version 0.0.0.0);
- BFAST Homer *et al.* (2009) (version 0.6.4);
- Bowtie Langmead *et al.* (2009) (version 0.12.0);
- SHRiMP Rumble *et al.* (2009) (version 2.0.1);
- SOCS Ondov *et al.* (2008) (version 2.0.3);
- Mapreads McKernan *et al.* (2009) (version 2.4.1);
- PerM Chen *et al.* (2009) (version 0.3.3);

Parameters: We used the following parameter settings for these mappers:

- *drFAST*: e=2,3 (error threshold for different runs).
- BFAST: Parameters recommended in the BFAST manual.
- Bowtie: n,v=2,3 (error threshold for different runs); -a (for reporting all); -S (output in SAM format); -C (color-space mapping).
- SHRiMP: -m 1 (score 1 for match); -i -1 (score -1 for mismatch) -x -1 (score -1 for read error); -U (ungapped alignment) -o 10000 (maximum number of alignments for a read); -N 1 (number of threads); -h 96% ($\geq$96% alignment identity).
- SOCS: -x 0 (number of bases to trim); -s 2 (mismatch sensitivity); -t 4 (mismatch tolerance); -m 0 (maximum number of alignments for a read, 0 indicates to report all); -T 1 (number

of threads); -N 1 (number of nodes); -l yes (consider the lower case bases in genome).

- Mapreads: S=0 (color-space mapping); M=2 (number of mismatches allowed); A=2 (count adjacent mismatches as one mismatch); Z=10000 (maximum number of alignment for a read).
- PerM: –seed S20 (full sensitivity for 2 SNPs); -v 4 (number of mismatches).

We used the same parameters (for reporting "all" mapping locations) when available to ensure a fair comparison.

Note that BWA and MAQ are not considered here since they ignore the first two characters of SOLiD reads.

Data, Reference Genome and Computing Power: We used both simulated and real data sets for comparisons. We simulated three sets, each with 4 million reads of length 50 bp sampled randomly from chromosome 1 of human reference genome (NCBI build 35) as follows:

- **Set 1:** We transformed the reads to color-space with no color errors and no mismatches.
- **Set 2:** Reads are transformed with two color errors. To achieve this, we transformed the reads to color-space and then changed the color of two arbitrarily selected non-consecutive positions. Note that if two color errors are consecutive, this might make it impossible to distinguish a read error from a SNP.
- **Set 3:** Generated with no color errors but one SNP.

In addition, we randomly selected 1 million (50bp long) reads from publicly available color-space reads generated from the genomes of NA18507 McKernan *et al.* (2009) (SRX004555), NA10847 (SRX008164), and NA12156. We used the human reference genome (NCBI build 35, unmasked) as the reference genome in all our experiments. The benchmarking results we report are performed on a server with 64-bit Intel Xeon processor and 8 GB of RAM.

Time, Accuracy and Sensitivity Results: We give the comparison results for all the mappers above with respect to the proportion of the reads that have at least one map location on the reference genome (sensitivity), total number of map locations found (comprehensiveness), and time needed to map the reads.

Table 3 shows the results on simulated data sets with error threshold of 2 (color errors and mismatches), except in the case of PerM where we allowed up to four mismatches due to recommendations of its developers. *drFAST* maps *all* of the reads from simulated data sets back to the reference genome very efficiently. The closest competitor to *drFAST* appears to be Bowtie, which is, in general, slower than *drFAST*-CT and is not 100% sensitive.Although Bowtie with a parameter setting of v=2 seems to map each read to more locations than *drFAST*, when no substitutions are present (Set 1), or a single color error is added (Set 2), this is simply due to Bowtie not being stringent on the number of errors it permits disregarding the parameter setting; we noticed that there are mapping locations with more than five color errors.

When the reads involve a nucleotide substitution (Set 3), the number of mapping locations are lower than that of *drFAST*. What is more interesting is the number of reads that can be mapped to the reference genome. It seems like Bowtie can map at most 61.9% of the reads even when they include a single color error (Set 2), in contrast, *drFAST* (both variants) map 100% of the reads. When the errors are in the form of nucleotide substitutions, the proportion of reads mapped by Bowtie drops to 51.66%.

Data Set	Mapper	Time (min.)	Map Locations	Reads Mapped (%)
Set 1	*drFAST*-DP	65	138,715,908	**100**
	drFAST-CT	40	137,483,484	**100**
	BFAST	88	8,803,840	96.1
	Bowtie v=2	17	25,581,176	99.4
	Bowtie n=2	67	168,307,651	99.4
	SHRiMP	414	13,961,155	99.8
	SOCS	45	13,357,519	**100**
	Mapreads	50	55,569,848	**100**
	PerM –seed S20 -v 4	17	14,441,796	96.2
Set 2	*drFAST*-DP	42	37,652,313	**100**
	drFAST-CT	26	36,458,468	**100**
	BFAST	101	8,098,581	98.0
	Bowtie v=2	13	9,738,234	60.8
	Bowtie n=2	31	57,550,920	61.9
	SHRiMP	519	11,977,512	99.8
	SOCS	90	12,909,860	**100**
	Mapreads	31	21,749,155	**100**
	PerM –seed S20 -v 4	15	12,679,070	98
Set 3	*drFAST*-DP	47	76,588,622	**100**
	drFAST-CT	32	75,970,911	**100**
	BFAST	105	8,982,132	97.4
	Bowtie v=2	16	11,030,554	49.4
	Bowtie n=2	43	70,508,835	51.66
	SHRiMP	472	11,859,215	99.8
	SOCS	96	9,780,960	**100**
	Mapreads	37	29,799,473	**100**
	PerM –seed S20 -v 4	15	13,140,561	97.5

Table 3. Performance results of all tested color-space read aligners on simulated data with error threshold of 2 mismatches. In the case of PerM, we allowed for mapping with up to four mismatches as recommended by its developers, yet its sensitivity failed to reach 100%. Reads are simulated from human reference genome build 35 (chromosome 1). Set 1: no errors, Set 2: color errors, Set 3: substitutions.

Since Bowtie was the closest competitor to *drFAST*, we performed another experiment on the same data sets by increasing the error threshold to 3 (Table 4). Interestingly for this setting, the proportion of reads mapped by Bowtie is 99.4%, almost matching the 100% mapping sensitivity of *drFAST*. However, both in terms of time and the number of map locations *drFAST* (both variants) perform better than Bowtie, especially when errors (Set 2 for color errors and Set 3 for nucleotide errors) are present.

As all three sets are generated from chromosome 1 with at most two errors added, a sensitive mapper should be able to map all reads to chromosome 1 when the error threshold is set to 2. In order to experimentally check the accuracy of all locations found by *drFAST*, we simulated the corresponding Illumina reads (letter-space) and aligned to chromosome 1 using mrsFAST. As seen in Table 5 for Sets 1 and 3, *drFAST* finds slightly more mapping

Data Set	Mapper	Time (min.)	Map Locations	Reads Mapped (%)
Set 1	*drFAST*-DP	88	364,601,231	**100**
	drFAST-CT	75	363,472,241	**100**
	Bowtie v=3	60	56,407,732	99.4
	Bowtie n=3	101	252,735,117	99.4
Set 2	*drFAST*-DP	42	118,053,818	**100**
	drFAST-CT	45	113,349,741	**100**
	Bowtie v=3	45	23,365,015	99.4
	Bowtie n=3	50	111,931,387	99.4
Set 3	*drFAST*-DP	47	215,274,860	**100**
	drFAST-DP	50	215,261,940	**100**
	Bowtie v=3	48	28,321,746	99.4
	Bowtie n=3	75	137,015,425	99.4

Table 4. Performance comparison on simulated data sets between *drFAST*-DP, *drFAST*-CT and Bowtie where error threshold is set to three mismatches.

Data Set	Mapper	Time (min.)	Map Locations	Reads Mapped (%)
Set 1	mrsFAST	20	135,450,193	100
Set 3	mrsFAST	20	75,115,629	100

Table 5. Number of mapping locations reported by mrsFAST for the same set of simulated reads in letter-space.

locations than mrsFAST, where the sensitivities of both aligners are 100%. The reason *drFAST* can find more mapping locations for SOLiD reads compared to the corresponding Illumina reads is because *drFAST* could map a read like `T0000` to two different positions with base pair contents `TTTT`, and also `CCCC` when one color error is "corrected". This is not the case with letter-space reads generated by a platform like Illumina Genome Analyzer. Although it would not be correct to arbitrarily select one "version" above the other, or returning both alignments as possibilities, we propose to correct such artifacts by incorporating the base pair quality values. This problem will arise only in polyN regions, thus, we propose to disable error correction of polyN reads where the base quality value of the first base is sufficiently high (i.e. $q > 30$).

BFAST and SHRiMP results are not presented for the three real data sets (Table 6) due to: (i) in our experiments, BFAST terminated with error in indexing step, (ii) SHRiMP requires 16 GB of main memory for alignment. Furthermore, both programs are much slower than *drFAST* or Bowtie. As a result, we only compared *drFAST* with Bowtie with an error threshold of 2 (see Table 6) and an error threshold of 3 (See Table 7). In five out of the six cases, *drFAST* maps significantly more reads, and to substantially more locations, in comparable time. The performance of the two programs are comparable only for NA18507 for n=2, in terms of mapped reads; however, *drFAST*-CT is slightly faster on this data set.

Data Set	Mapper	Time	Map Locations	Reads Mapped (%)
NA18507	*drFAST*-DP	114	189,276,027	36.8
	drFAST-CT	54	149,362,540	35.6
	Bowtie v=2	21	64,092,233	27.8
	Bowtie n=2	63	202,948,323	35.2
	SOCS	320	21,081,941	35.3
	Mapreads	80	17,032,680	37.3
	PerM –seed S20 -v 4	76	10,068,062	35.3
NA10847	*drFAST*-DP	200	667,928,813	47.1
	drFAST-CT	100	512,599,230	45.9
	Bowtie v=2	84	280,928,112	38.0
	Bowtie n=2	91	270,996,634	36.0
	SOCS	420	53,668,622	44.8
	Mapreads	140	39,589,079	48.5
	PerM –seed S20 -v 4	100	20,699,652	44.8
NA12156	*drFAST*-DP	136	491,158,791	33.5
	drFAST-CT	91	440,317,111	32.5
	Bowtie v=2	99	329,916,108	25.0
	Bowtie n=2	99	318,621,596	23.7
	SOCS	400	38,246,530	31.2
	Mapreads	110	22,182,469	35.1
	PerM –seed S20 -v 4	140	10,798,496	31.2

Table 6. Performance comparison on real data sets between *drFAST*-DP, *drFAST*-CT, Bowtie, SOCS, and PerM on 1 million randomly selected reads from three different sequencing experiments. We set the error threshold to 2 bp for all aligners, except PerM, where we set the threshold to four as per the PerM developers suggested. We then removed the alignments with more than 2 bp mismatches for comparison purposes.

Data Set	Mapper	Time (min.)	Map Locations	Reads Mapped (%)
NA18507	*drFAST*-DP	154	309,994,599	41.5
	drFAST-CT	61	302,237,779	40.5
	Bowtie v=3	63	145,473,423	37.1
	Bowtie n=3	78	290,357,005	36.35
NA10847	*drFAST*-DP	300	1,121,281,408	52.1
	drFAST-CT	141	1,092,259,727	51.6
	Bowtie v=3	182	565,114,739	47.8
	Bowtie n=3	76	270,885,799	35.8
NA12156	*drFAST*-DP	310	655,648,865	42.4
	drFAST-CT	120	639,667,174	39.5
	Bowtie v=3	187	585,191,747	34.6
	Bowtie n=3	98	318,527,434	23.6

Table 7. Performance comparison on real data sets between *drFAST*-DP, *drFAST*-CT and Bowtie on 1 million randomly selected reads from three different sequencing experiments. We set the error threshold to 3 bp.

We also compared the amount of memory used by each program when mapping 1 million reads to the human reference genome assembly (Table 4).

[1] SOCS memory usage is a parameter set by user, we used 5 Gigabytes of memory in our experiments

[2] Mapreads memory usage is a parameter set by user, we used 8 Gigabytes of memory in our experiments

Mapper	Memory Usage
drFAST-DP	1.3 GB
drFAST-CT	1.3 GB
BFAST	$\geq$ 10 GB
Bowtie	4.5 GB
SHRiMP	16 GB
SOCS	$\geq$ 5 GB [1]
PerM	2 GB
Mapread	$\geq$ 8 GB [2]

Table 8. Memory required by each software to map 1 million 35-base reads to human reference genome. The memory requirement increases with the number of reads and/or the read length, this increase is typically linear with the increase in the number of basepairs in the data set.

One important issue to note is that the *drFAST* aligner is aimed at SV/CNV inference and it does not return mapping quality values, which are still essential for accurate SNP detection. However, *drFAST* also returns "best" map locations for paired-end and matepair reads in addition to all possible discordant configurations where "best" is defined as the mapping with the lowest Hamming distance and with span size closest to the library average. Future releases of *drFAST* will have the capability of returning mapping quality for these best map locations, which will effectively increase the appeal of *drFAST*, and users will be able to use it for both structural variation discovery through multi-mapping paired-end and matepair reads, and SNP discovery. Until this feature is available in *drFAST*, one may need to run other aligners in parallel to discover SNPs.

5 CONCLUSION

This is an exciting time for genomics research. The amount of available 1000 Genomes Project Consortium (2010) and anticipated Genome 10K Community of Scientists (2009) sequence data now arms us to expand our understanding human variation, disease susceptibility, and genome evolution. Although there are inherent accuracy and bias problems associated with different sequencing platforms Smith *et al.* (2008), we can also leverage the different "strengths" of these technologies to increase confidence and comprehensiveness of SNP Nothnagel *et al.* (2011) and structural variation Mills *et al.* (2011) discovery.

For species where a reference genome is available as in human, mapping sequence reads to this reference assembly is the first step in genome analysis. Sensitivity and accuracy, as well as the speed of read alignment, are crucial for precise characterization of genomic variants. To this end, many mapping algorithms were developed Li *et al.* (2008a,b); Li and Durbin (2009); Li *et al.* (2009); Alkan *et al.* (2009); Hach *et al.* (2010); Homer *et al.* (2009); Rumble *et al.* (2009) focusing mainly on the Illumina Genome Analyzer data, and very little effort was devoted to analyze color-space reads generated with the SOLiD platform McKernan *et al.* (2009); Rumble *et al.* (2009); Homer *et al.* (2009). The main limitation of the SOLiD-aware read aligners is that they were not optimized for structural variation detection (except for SHRiMP Rumble *et al.* (2009), which is more powerful in mapping to more complex areas of the genome), and they are unusable for segmental duplication analysis due to their unique mapping approach Alkan *et al.* (2009). On the other hand, by tracking all possible map locations and underlying sequence variation, *drFAST* provides an opportunity to better access and increase "mappability" in repeat and duplication-rich areas of the genome that are known to harbor much structural variation Kidd *et al.* (2008). Although the **sensitivity** of *drFAST* is higher than the other aligners, we also demonstrate **speed enhancements** of both dynamic programming and color transformation versions. Through its readiness to be integrated to VariationHunter Hormozdiari *et al.* (2009) for more sensitive SV discovery, to NovelSeq Hajirasouliha *et al.* (2010) to characterize novel sequence insertions, and usability for segmental variation detection Alkan *et al.* (2009). *drFAST* is an important step forward for recovering additional genetic variation from di-base encoded color-space sequencing.

ACKNOWLEDGMENTS

Funding: Natural Sciences and Engineering Research Council of Canada (NSERC to S.C.S. in parts); Bioinformatics for Combating Infectious Diseases (BCID to S.C.S. in parts); Michael Smith Foundation for Health Research grants (to S.C.S. in parts); US National Institutes of Health (grants HG004120 and HG005209 to E.E.E.). E.E.E. is a Howard Hughes Medical Institute Investigator.

Conflict of Interest: EEE is an SAB member of the Pacific Biosciences.

REFERENCES

1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**(7319), 1061–1073.

Alkan, C., Kidd, J. M., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., Kitzman, J. O., Baker, C., Malig, M., Mutlu, O., Sahinalp, S. C., Gibbs, R. A., and Eichler, E. E. (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature Genetics*, **41**(10), 1061–1067.

Alkan, C., Coe, B. P., and Eichler, E. E. (2011). Genome structural variation discovery and genotyping. *Nat Rev Genet*, **12**(5), 363–376.

Armstrong, M. (1988). Groups and symmetry. In *Springer Verlag*, page 53.

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., Hall, K. P., Evers, D. J., Barnes, C. L., Bignell, H. R., Boutell, J. M., Bryant, J., Carter, R. J., Cheetham, R. K., Cox, A. J., Ellis, D. J., Flatbush, M. R., Gormley, N. A., Humphray, S. J., Irving, L. J., Karbelashvili, M. S., Kirk, S. M., Li, H., Liu, X., Maisinger, K. S., Murray, L. J., Obradovic, B., Ost, T., Parkinson, M. L., Pratt, M. R., Rasolonjatovo, I. M. J., Reed, M. T., Rigatti, R., Rodighiero, C., Ross, M. T., Sabot, A., Sankar, S. V., Scally, A., Schroth, G. P., Smith, M. E., Smith, V. P., Spiridou, A., Torrance, P. E., Tzonev, S. S., Vermaas, E. H., Walter, K., Wu, X., Zhang, L., Alam, M. D., Anastasi, C., Aniebo, I. C., Bailey, D. M. D., Bancarz, I. R., Banerjee, S., Barbour, S. G., Baybayan, P. A., Benoit, V. A., Benson, K. F., Bevis, C., Black, P. J., Boodhun, A., Brennan, J. S., Bridgham, J. A., Brown, R. C., Brown, A. A., Buermann, D. H., Bundu, A. A., Burrows, J. C., Carter, N. P., Castillo, N., Catenazzi, M. C. E., Chang, S., Cooley, R. N., Crake, N. R., Dada, O. O., Diakoumakos, K. D., Dominguez-Fernandez, B., Earnshaw, D. J., Egbujor, U. C., Elmore, D. W., Etchin, S. S., Ewan, M. R., Fedurco, M., Fraser, L. J., Fajardo, K. V. F., Furey, W. S., George, D., Gietzen, K. J., Goddard, C. P., Golda, G. S., Granieri, P. A., Green, D. E., Gustafson, D. L., Hansen, N. F., Harnish, K., Haudenschild, C. D., Heyer, N. I., Hims, M. M., Ho, J. T., Horgan, A. M., Hoschler, K., Hurwitz, S., Ivanov, D. V., Johnson, M. Q., James, T., Jones, T. A. H., Kang, G.-D., Kerelska, T. H., Kersey, A. D., Khrebtukova, I., Kindwall, A. P., Kingsbury, Z., Kokko-Gonzales, P. I., Kumar, A., Laurent, M. A., Lawley, C. T., Lee, S. E., Lee, X., Liao, A. K., Loch, J. A., Lok, M., Luo, S., Mammen, R. M., Martin, J. W., McCauley, P. G., McNitt, P., Mehta, P., Moon, K. W., Mullens, J. W., Newington, T., Ning, Z., Ng, B. L., Novo, S. M., O'Neill, M. J., Osborne, M. A., Osnowski, A., Ostadan, O., Paraschos, L. L., Pickering, L., Pike, A. C., Pike, A. C., Pinkard, D. C., Pliskin, D. P., Podhasky, J., Quijano, V. J., Raczy, C., Rae, V. H., Rawlings,

S. R., Rodriguez, A. C., Roe, P. M., Rogers, J., Bacigalupo, M. C. R., Romanov, N., Romieu, A., Roth, R. K., Rourke, N. J., Ruediger, S. T., Rusman, E., Sanches-Kuiper, R. M., Schenker, M. R., Seoane, J. M., Shaw, R. J., Shiver, M. K., Short, S. W., Sizto, N. L., Sluis, J. P., Smith, M. A., Sohna, J. E. S., Spence, E. J., Stevens, K., Sutton, N., Szajkowski, L., Tregidgo, C. L., Turcatti, G., Vandevondele, S., Verhovsky, Y., Virk, S. M., Wakelin, S., Walcott, G. C., Wang, J., Worsley, G. J., Yan, J., Yau, L., Zuerlein, M., Rogers, J., Mullikin, J. C., Hurles, M. E., McCooke, N. J., West, J. S., Oaks, F. L., Lundberg, P. L., Klenerman, D., Durbin, R., and Smith, A. J. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, **456**(7218), 53–59.

Burrows, M. and Wheeler, D. (1994). A block sorting lossless data compression algorithm. *Digital Equipment Corporation Technical Report*, **124**.

Chen, Y., Souaiaia, T., and Chen, T. (2009). PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics*, **25**(19), 2514–2521.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., Dewinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden, D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J., and Turner, S. (2009). Real-time DNA sequencing from single polymerase molecules. *Science*, **323**(5910), 133–138.

Ferragina, P. and Manzini, G. (2000). Opportunistic data structures with applications. *Proc. of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*, page 390.

Frigo, M., Leiserson, C. E., Prokop, H., and Ramachandran, S. (1999). Cache-oblivious algorithms. In *40th Annual Symposium on Foundations of Computer Science*, pages 285–297, New York, New York.

Genome 10K Community of Scientists (2009). Genome 10K: a proposal to obtain whole-genome sequence for 10,000 vertebrate species. *J Hered*, **100**(6), 659–674.

Hach, F., Hormozdiari, F., Alkan, C., Hormozdiari, F., Birol, I., Eichler, E. E., and Sahinalp, S. C. (2010). mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nat Methods*, **7**(8), 576–577.

Hajirasouliha, I., Hormozdiari, F., Alkan, C., Kidd, J. M., Birol, I., Eichler, E. E., and Sahinalp, S. C. (2010). Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, **26**(10), 1277–1283.

Homer, N., Merriman, B., and Nelson, S. F. (2009). BFAST: An Alignment Tool for Large Scale Genome Resequencing. *PLoS ONE*, **4**(11), 12.

Hormozdiari, F., Alkan, C., Eichler, E. E., and Sahinalp, S. C. (2009). Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Research*, **19**(7), 1270–1278.

Hormozdiari, F., Hajirasouliha, I., Dao, P., Hach, F., Yörükoglu, D., Alkan, C., Eichler, E. E., and Sahinalp, S. C. (2010). Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics [ISMB]*, **26**(12), 350–357.

Kidd, J. M., Cooper, G. M., Donahue, W. F., Hayden, H. S., Sampas, N., Graves, T., Hansen, N., Teague, B., Alkan, C., Antonacci, F., Haugen, E., Zerr, T., Yamada, N. A., Tsang, P., Newman, T. L., Tzn, E., Cheng, Z., Ebling, H. M., Tusneem, N., David, R., Gillett, W., Phelps, K. A., Weaver, M., Saranga, D., Brand, A., Tao, W., Gustafson, E., McKernan, K., Chen, L., Malig, M., Smith, J. D., Korn, J. M., McCarroll, S. A., Altshuler, D. A., Peiffer, D. A., Dorschner, M., Stamatoyannopoulos, J., Schwartz, D., Nickerson, D. A., Mullikin, J. C., Wilson, R. K., Bruhn, L., Olson, M. V., Kaul, R., Smith, D. R., and Eichler, E. E. (2008). Mapping and sequencing of structural variation from eight human genomes. *Nature*, **453**(7191), 56–64.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, **10**(3), R25.

Lee, S., Hormozdiari, F., Alkan, C., and Brudno, M. (2009). MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nat Methods*, **6**(7), 473–474.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Li, H., Ruan, J., and Durbin, R. (2008a). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*, **18**(11), 1851–1858.

Li, R., Li, Y., Kristiansen, K., and Wang, J. (2008b). SOAP: short oligonucleotide alignment program. *Bioinformatics*, **24**(5), 713–714.

Li, R., Yu, C., Li, Y., Lam, T.-W., Yiu, S.-M., Kristiansen, K., and Wang, J. (2009). SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, **25**(15), 1966–7.

Lupski, J. R., Reid, J. G., Gonzaga-Jauregui, C., Deiros, D. R., Chen, D. C. Y., Nazareth, L., Bainbridge, M., Dinh, H., Jing, C., Wheeler, D. A., McGuire, A. L., Zhang, F., Stankiewicz, P., Halperin, J. J., Yang, C., Gehman, C., Guo, D., Irikat, R. K., Tom, W., Fantin, N. J., Muzny, D. M., and Gibbs, R. A. (2010). Whole-genome sequencing in a patient with Charcot-Marie-Tooth neuropathy. *N Engl J Med*, **362**(13), 1181–1191.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J.-B., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**(7057), 376–380.

McKernan, K. J., Peckham, H. E., Costa, G. L., McLaughlin, S. F., Fu, Y., Tsung, E. F., Clouser, C. R., Duncan, C., Ichikawa, J. K., Lee, C. C., Zhang, Z., Ranade, S. S., Dimalanta, E. T., Hyland, F. C., Sokolsky, T. D., Zhang, L., Sheridan, A., Fu, H., Hendrickson, C. L., Li, B., Kotler, L., Stuart, J. R., Malek, J. A., Manning, J. M., Antipova, A. A., Perez, D. S., Moore, M. P., Hayashibara, K. C., Lyons, M. R., Beaudoin, R. E., Coleman, B. E., Laptewicz, M. W., Sannicandro, A. E., Rhodes, M. D., Gottimukkala, R. K., Yang, S., Bafna, V., Bashir, A., MacBride, A., Alkan, C., Kidd, J. M., Eichler, E. E., Reese, M. G., De La Vega, F. M., and Blanchard, A. P. (2009). Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Research*, **19**(9), 1527–41.

Mills, R. E., Walter, K., Stewart, C., Handsaker, R. E., Chen, K., Alkan, C., Abyzov, A., Yoon, S. C., Ye, K., Cheetham, R. K., Chinwalla, A., Conrad, D. F., Fu, Y., Grubert, F., Hajirasouliha, I., Hormozdiari, F., Iakoucheva, L. M., Iqbal, Z., Kang, S., Kidd, J. M., Konkel, M. K., Korn, J., Khurana, E., Kural, D., Lam, H. Y. K., Leng, J., Li, R., Li, Y., Lin, C.-Y., Luo, R., Mu, X. J., Nemesh, J., Peckham, H. E., Rausch, T., Scally, A., Shi, X., Stromberg, M. P., Sttz, A. M., Urban, A. E., Walker, J. A., Wu, J., Zhang, Y., Zhang, Z. D., Batzer, M. A., Ding, L., Marth, G. T., McVean, G., Sebat, J., Snyder, M., Wang, J., Ye, K., Eichler, E. E., Gerstein, M. B., Hurles, M. E., Lee, C., McCarroll, S. A., Korbel, J. O., and Project, . G. (2011). Mapping copy number variation by population-scale genome sequencing. *Nature*, **470**(7332), 59–65.

Ng, S. B., Bigham, A. W., Buckingham, K. J., Hannibal, M. C., McMillin, M. J., Gildersleeve, H. I., Beck, A. E., Tabor, H. K., Cooper, G. M., Mefford, H. C., Lee, C., Turner, E. H., Smith, J. D., Rieder, M. J., Yoshiura, K.-I., Matsumoto, N., Ohta, T., Niikawa, N., Nickerson, D. A., Bamshad, M. J., and Shendure, J. (2010). Exome sequencing identifies mll2 mutations as a cause of kabuki syndrome. *Nat Genet*, **42**(9), 790–793.

Nothnagel, M., Herrmann, A., Wolf, A., Schreiber, S., Platzer, M., Siebert, R., Krawczak, M., and Hampe, J. (2011). Technology-specific error signatures in the 1000 Genomes Project data. *Hum Genet*.

Ondov, B. D., Varadarajan, A., Passalacqua, K. D., and Bergman, N. H. (2008). Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications. *Bioinformatics*, **24**(23), 2776–2777.

Pushkarev, D., Neff, N. F., and Quake, S. R. (2009). Single-molecule sequencing of an individual human genome. *Nat Biotechnol*, **27**(9), 847–850.

Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A., and Brudno, M. (2009). SHRiMP: Accurate Mapping of Short Color-space Reads. *PLoS Computational Biology*, **5**(5), 11.

Smith, D. R., Quinlan, A. R., Peckham, H. E., Makowsky, K., Tao, W., Woolf, B., Shen, L., Donahue, W. F., Tusneem, N., Stromberg, M. P., Stewart, D. A., Zhang, L., Ranade, S. S., Warner, J. B., Lee, C. C., Coleman, B. E., Zhang, Z., McLaughlin, S. F., Malek, J. A., Sorenson, J. M., Blanchard, A. P., Chapman, J., Hillman, D., Chen, F., Rokhsar, D. S., McKernan, K. J., Jeffries, T. W., Marth, G. T., and Richardson, P. M. (2008). Rapid whole-genome mutational profiling using next-generation sequencing technologies. *Genome Res*, **18**(10), 1638–1642.

Sudmant, P. H., Kitzman, J. O., Antonacci, F., Alkan, C., Malig, M., Tsalenko, A., Sampas, N., Bruhn, L., Shendure, J., Project, . G., and Eichler, E. E. (2010). Diversity of human copy number variation and multicopy genes. *Science*, **330**(6004), 641–646.

Vissers, L. E. L. M., de Ligt, J., Gilissen, C., Janssen, I., Steehouwer, M., de Vries, P., van Lier, B., Arts, P., Wieskamp, N., del Rosario, M., van Bon, B. W. M., Hoischen, A., de Vries, B. B. A., Brunner, H. G., and Veltman, J. A. (2010). A de novo

paradigm for mental retardation. *Nat Genet*, **42**(12), 1109–1112.

Wheeler, D. A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A., He, W., Chen, Y.-J., Makhijani, V., Roth, G. T., Gomes, X., Tartaro, K., Niazi, F., Turcotte, C. L., Irzyk, G. P., Lupski, J. R., Chinault, C., zhi Song, X., Liu, Y., Yuan, Y., Nazareth, L., Qin, X., Muzny, D. M., Margulies, M., Weinstock, G. M., Gibbs, R. A., and Rothberg, J. M. (2008). The complete genome of an individual by massively parallel DNA sequencing. *Nature*, **452**(7189), 872–876.