Analyzing DNA Double-Strand Break Repair Outcomes with a Custom Needleman-Wünsch-Based Sequence Alignment Tool

Megan Gelement[1]
[1] *Tufts University*
(Dated: February 15, 2023)

Pairwise sequence alignment is a necessary step in DNA repair analysis. A sequence that has undergone repair from a lesion such as a double-strand break must be aligned to a representation of its original source to identify any mutations that occurred in the process. Pairwise sequence alignment methods are commonly used in the identification of common ancestry or function between nucleotide or amino acid sequences. In these cases, researchers seek an optimal alignment that minimizes the misaligned characters between source and query sequences. Multiple such optimal alignments may exist between the same two sequences. When studying DNA repair, small differences between otherwise similarly-optimal alignments may confound characterization of repair pathways. To address this problem, we developed a custom pairwise sequence aligner designed to maximize control over alignment scoring and produce output designed specifically for ease of comparing repair outcomes.

I. INTRODUCTION

Double-strand breaks (DSBs) are detrimental DNA lesions that must be repaired to preserve cellular and genomic integrity [1]. In a DSB, the phosphate backbones of both strands of the double-helix are severed [2]. DSBs occur frequently in humans [1, 3] as a result of internal and external perturbations to the cell ranging from replication errors to viruses to CT scans and cancer therapeutics [2–4]. Eukaryotic cells employ a range of DSB repair mechanisms, and their use depends on a variety of factors, such as availability of helper molecules and phase of the cell cycle at repair time. Homologous recombination (HR) occurs during the S and G2 phases of the cell cycle and requires a template to complete the repair [5]. Non-homologous end-joining (NHEJ) can happen at any point in the cell cycle and does not require a template; instead, broken ends are directly joined to repair the cell [5]. This pathway is not as reliable as HR, which is generally thought of as error-free [6]. A third class of repair, alternative end-joining (alt-EJ), is the most mutagenic of the three and frequently relies on alignment of short stretches of homologous nucleotides on either side of the break to begin repair. This type of alt-EJ, termed microhomology-mediated end joining or MMEJ, has been thought of as a backup to classical NHEJ or HR [7], but recent studies indicate that MMEJ occurs both with and without the conditions for these standard pathways [5, 6]. Understanding these alternative repair pathways is crucial to understanding mutagenic repair.

Failure to faithfully repair DSBs can lead to cell death, and errors in double-strand break repair pathways are implicated in human diseases, including cancers [6, 8, 9]. Thus, fidelity of DSB repair is highly important to cell regulation. Further, understanding DSB repair is integral to the advancement of precision medicine [10].

Studying DSB repair is complicated by a number of factors. Many pairwise sequence alignment algorithms exist to find optimal or approximate near-optimal alignments between sequences [11]. However, more than one optimal alignment may exist between a sequence pair, and small alignment details may have implications in repair event characterization. Next-generation sequencing (NGS) allows researchers to sequence thousands of repair events across samples with relative ease; however, sequencing errors can confound attempts to characterize mutations [7]. To achieve fine control over the alignment scoring scheme and control alignment output, we built a custom Needleman-Wünsch-based tool that leverages base quality and position to score pairwise sequence alignments.

II. METHODS AND DATA

1 **Summary of methods.** Our custom Needleman-Wünsch-based alignment tool takes repaired DSB sequence data as FASTQ files and source sequences as FASTA files, and produces SAM files containing alignment results. The alignment algorithm was parallellized using C++ threads. Test data pre-processing, the basic Needleman-Wünsch algorithm, and the modifications developed for our alignment tool are described below.

2 **Data sources and collection.** Sequence and repair data were collected and pre-processed by Khodaverdian et. al. as described in [12]. A custom construct based on *Iw7* [12, 13] was generated, cloned, and subject to double-strand break at an I- *Sce*I recognition site, incubated to allow for repair, and re-sequenced. Alignments created with Geneious BioMatters software for [12] were used as a benchmark for our aligner.

3 **The Needleman-Wünsch algorithm.** The Needleman-Wünsch algorithm is a dynamic programming algorithm for global string alignment developed by Saul B. Needleman and Christian D. Wünsch in 1970 [14]. Originally intended for finding similarities between amino acid sequences, it is used for nucleotide sequence alignment as well. A common purpose of DNA sequence alignment is finding sequence homology - evidence of shared

ancestry through similarity at the nucleotide level [15].

The Needleman-Wünsch algorithm is guaranteed to find the mathematically optimal alignment between two sequences [16]. A two-dimensional array A scores all possible alignments between two sequences x and y. Each cell in the matrix stores the optimal score of aligning $x[0...i]$ with $y[0...j]$. $A(0,0)$ is set to 0. Similarity scores are computed using a scheme that assigns a positive score to a match between two bases and negative scores to mismatches or gaps (insertions or deletions from the source sequence).

Aside from the base case described above, A is scored as follows:

$$A(i,j) = \max \begin{cases} A(i-1,j-1) + s(x_i, y_j) \\ A(i,j-1) + g \\ A(i-1,j) + g \end{cases} \quad (1)$$

where $s(x_i, y_j)$ is the score obtained by aligning the ith character of sequence x with the jth character of sequence y, and g is the cost of inserting a gap in x or in y. $A(n,m)$ stores the optimal score (where, without loss of generality, n = the length of x and m, the length of y). The optimal alignment is obtained by back-tracing from $A(n,m)$. This algorithm runs in $O(n^2)$.

4 **Incorporating the affine gap penalty.**

The original gap penalty described above sets gap costs proportional to gap length. However, the best-scoring alignments produced with such costs tend to result in many short deletions and insertions, often inconsistent with biological expectation [17]. Rather than a gap cost of gk, where g represents the per-character gap cost and k represents the length of the gap, with an affine gap penalty each gap is charged $g_i + g_e k$, where g_i represents a gap initialization penalty and g_e represents a gap extension penalty. Gaps are encouraged to cluster together if these scores are set such that it is more costly to initialize a gap than it is to extend a gap. Past studies and its use in commercial alignment tools suggest that affine gap scoring produces more accurate sequence alignments [17, 18].

In affine gap scoring, we keep three matrices to track the score: one representing alignments ending in a match or mismatch between sequences x and y; one representing alignments ending in a gap in x, and one representing alignments ending in a gap in y. However, because we only need to keep track of whether we begin or extend a gap, we maintain our runtime of $O(n^2)$. The traceback algorithm to retrieve the optimal alignment works as before, but now works between all three matrices [19].

5 **Quality score-aware error detection.**

Repaired read input to our aligner is stored in FASTQ format, a file format for sequencing read data that includes quality scores for each base call [20]. Illumina sequencing, as used in [12], uses PHRED quality scoring. PHRED scores [20, 21] define base quality Q in terms of the probability of error E:

$$Q = -10 \cdot \log_{10}(Pr(E)) \quad (2)$$

Thus at $Q = 10$, the probability of an incorrect base call is 1 in 10; at $Q = 50$, the probability of an incorrect base call is 1 in 100,000, and so on. PHRED scores are typically encoded in FASTQ files as ASCII characters.

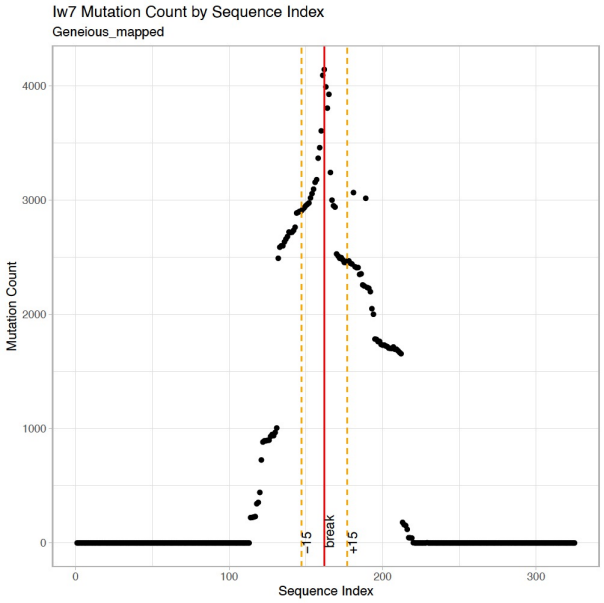
Our aligner allows the user to set a threshold quality score at which mutations are treated as miscalled bases, forcing a mismatch rather than potentially introducing a new gap where one does not belong. In future, the aligner will also flag for the user where the quality threshold was reached.

6 **Sequence position-aware error detection.** Our custom tool also allows the user to update the scoring scheme based on biological context. Mutations found k nucleotides or more from a DSB break site may have a distance factor added to their score to encourage a mismatch over a gap. In a future version of the software, mismatches found some k nucleotides or further from a break site that shows no mutation will be considered PCR artifacts, and the sequence will therefore be considered faithfully repaired. Mismatches found k nucleotides or further from a break site that *does* show evidence of mutation will be flagged for further review. k will be chosen by the user.

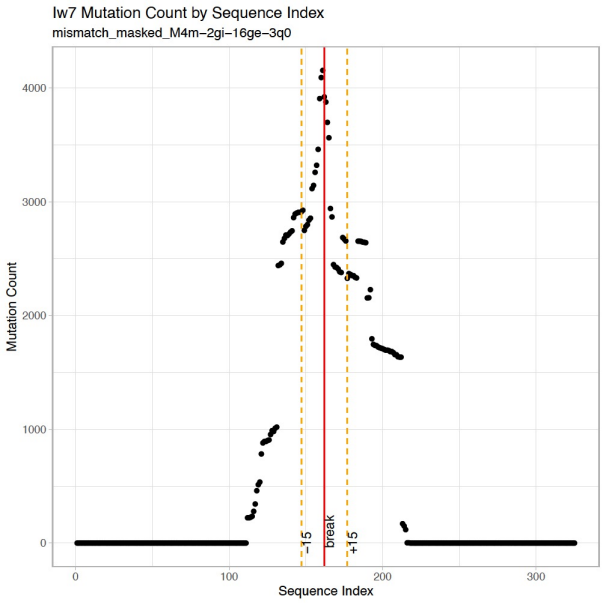
7 **Clearly identifying mismatches with modified CIGAR strings.** A CIGAR ('Compact Idiosyncratic Gapped Alignment Report') string is a shorthand format used to represent pairwise sequence alignments in SAM files. SAM ('Sequence Alignment/Map') is a standard file format for storing sequence alignment data [22, 23]. Standard CIGAR strings are represented as strings of integer-letter pairs, where the integer represents the number of bases that fulfill three potential operations: a match/mismatch, represented with an 'M', an insertion relative to the reference sequence, represented with 'I', and a deletion relative to the reference sequence, represented with 'D.' In our data, the CIGAR string '325M' represents a perfect repair event with no insertions or deletions. The CIGAR string '152M7D166M' represents a CIGAR string with 7 base deletions near the repair junction. However, in either example, this formulation of the CIGAR string masks potential mismatches because 'M' represents either a match *or* a mismatch.

Our aligner produces a more verbose CIGAR string, where an 'X' represents a mismatch (an option not usually taken in commercial alignment software but described in the SAM file format specification [23]). Thus, '325M' *only* represents a repair event that perfectly matches its source. '167M2X156M,' on the other hand, represents a repair event with 2 mismatches. This added operation simplifies analysis of mutations without significantly increasing the size of the output.

III. RESULTS AND DISCUSSION

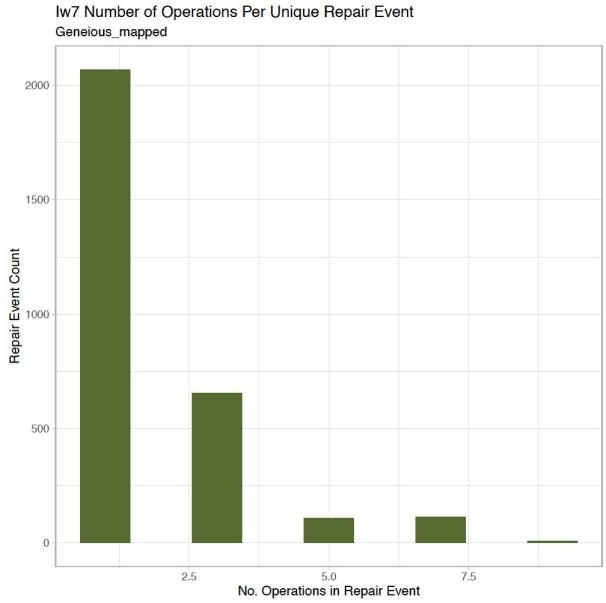


(a) Indel position frequency in alignment to *Iw7* construct made by Geneious BioMatters software as described in [12]

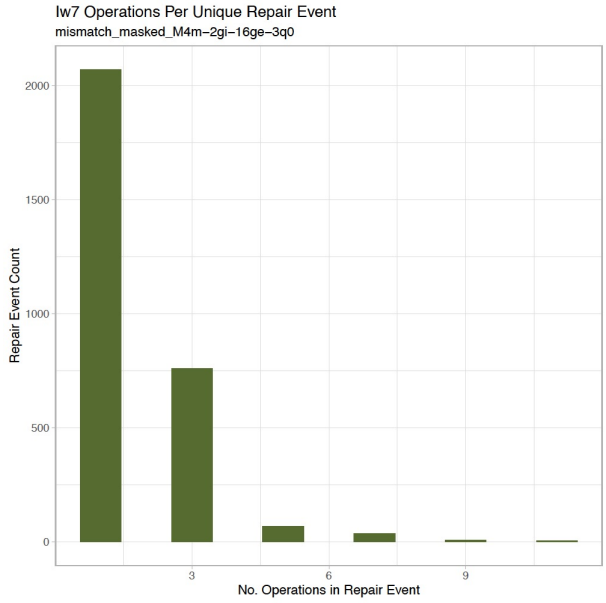


(b) Indel position frequency in alignment to *Iw7* construct made by our Needleman-Wünsch aligner

FIG. 1: The figures above indicate the frequency of indels (insertions or deletions) observed across all reads in construct *Iw7* by position along the source sequence, aligned either with a commercial aligner (a) or our custom aligner (b). The x axes represent base position from 1 to 325. The red vertical line represents the break site; the orange dotted lines represent the position 15 bases to either side of the break site. Mismatches are not included here, as the CIGAR strings produced in [12] did not distinguish between matches and mismatches.



(a) Number of operations (as recorded in CIGAR strings) per unique repair event when repaired construct reads are aligned to their source (*Iw7*) with commercial software ([12])



(b) Number of operations (as recorded in CIGAR strings) per unique repair event when repaired construct reads are aligned to their source (*Iw7*) with our aligner

FIG. 2: The figures above indicate the distribution of operation counts in the CIGAR strings of unique repair events in construct *Iw7*, in both the commercial (a) and custom (b) aligners. Note that in our aligner, mismatches are "masked," or left out of the CIGAR string in order to accurately compare with the prior alignment results, which do not account for mismatches in the CIGAR string. Note that a perfect match has one operation: M. An alignment with a single deletion of 5 bases near the middle of the sequence has 3 operations: M, D, M (ex: 151M5D169M.)

8 **Our custom aligner produces results consistent with out-of-the-box results.** In order to determine whether our aligner was successful, we asked whether it was capable of producing similar alignments to that of the Geneious BioMatters alignment software previously used by Khodaverdian et. al. [12]. We asked whether our aligner produced mutations that generally clustered near the repair junction (which is biologically expected), and how much the number of operations in alignment CIGAR strings varied. The intuition for this last question is that a complex CIGAR string with many operations (match, insertion, deletion, and in our case, mismatch) represents less faithful repair than a CIGAR string with few operations. While there is no ground truth to which to compare the results of either aligner, our aligner should avoid introducing excessive complexity to sequence alignments.

To address these questions, we compared alignment results of repaired *Iw7* reads produced by Khodaverdian et. al. in [12] to those of our new aligner on the same reads. An initial test of 81 different scoring schemes was used to explore the space. A scheme in which a match was assigned +4, a mismatch, -2, gap initialization, -16, and gap extension, -3 was chosen for comparison to the results of the Geneious alignment using default parameters. In order to fairly compare CIGAR strings between the two aligners, mismatches were "masked," or treated as matches when constructing CIGAR strings with our aligner.

The resulting CIGAR strings were then used for comparison as shown in Fig. 1 and Fig. 2, above. Fig. 1 demonstrates that the distribution of mutation placement is comparable between the two aligners. Fig. 2 demonstrates that CIGAR string complexity is similar between the two aligners.

9 **Current and future work will tune alignment parameters to existing data and explore metrics for characterization of DSB repair.** Further development will identify the optimal scoring scheme for a set of DSB repair data produced by Hanscom et. al. in [5], as well as for an additional set of custom constructs currently in development. The optimal set of scoring parameters for our data can be learned by minimizing error E, where for each construct:

$$E = \sum_i 325M - s_i \tag{3}$$

where each source sequence is 325 bases long, M is the "match" score between a pair of bases, and s_i is the Needleman-Wünsch alignment score between unique repair product i and its source sequence. This will require fixing M and setting a minimum threshold for mismatch and gap initialization/extension penalties.

Once a scoring scheme is chosen, the next step (currently in development but outside the scope of this report) is to develop metrics by which to characterize repair fidelity (how close are repair products to their source sequences?) and diversity (how much do repair products vary?) both within and across 1,100+ constructs.

IV. CONCLUSION

We have developed a Needleman-Wünsch sequence alignment tool with an affine gap penalty and the capacity to adjust the scoring scheme according to base quality and proximity to the repair junction. The tool runs in the same $O(n^2)$ as the standard Needleman-Wünsch algorithm and allows fine-grain control over scoring. Its primary use will be in the analysis of existing repair data produced by Hanscom et. al. in an earlier study [5], as well as on an additional dataset to be produced this year, with the goal of expanding characterization and analysis of sequence characteristics that may influence DSB repair.

ACKNOWLEDGMENTS

This research was performed in collaboration with Dr. Lenore Cowen, Dr. Mitch McVey, and Nick Woodward (Tufts University). Funding was provided by the MSCS Consortium/NSF DREAM program. This paper was formatted in Overleaf using the following guide by Jennifer E. Hoffman (Harvard University, October 10, 2022): http://hoffman.physics.harvard.edu/example-paper/

[1] T. Hanscom and M. McVey, Regulation of Error-Prone DNA Double-Strand Break Repair and Its Impact on Genome Evolution, Cells **1657**, doi:10.3390/cells9071657 (2020).

[2] A. Mehta and J. E. Haber, Sources of dna double-strand breaks and models of recombinational dna repair, Cold Spring Harbor perspectives in biology **6**, a016428 (2014).

[3] L. Deriano and D. B. Roth, Modernizing the nonhomologous end-joining repertoire: alternative and classical nhej share the stage., Annual review of genetics **47**, 433 (2013).

[4] H. H. Chang, N. R. Pannunzio, N. Adachi, and M. R. Lieber, Non-homologous dna end joining and alternative pathways to double-strand break repair, Nature reviews Molecular cell biology **18**, 495 (2017).

[5] T. Hanscom, N. Woodward, R. Batorsky, A. J. Brown, S. A. Roberts, and M. McVey, Characterization of sequence contexts that favor alternative end joining at cas9-induced double-strand breaks, Nucleic Acids Research **50**, 7465 (2022).

[6] A. Sfeir and L. S. Symington, Microhomology-mediated end joining: a back-up survival mechanism or dedicated

pathway?, Trends in biochemical sciences **40**, 701 (2015).
[7] A. J. Brown, A. T. Al-Soodani, M. Saul, S. Her, J. C. Garcia, D. A. Ramsden, C. Her, and S. A. Roberts, High-throughput analysis of dna break-induced chromosome rearrangements by amplicon sequencing, in *Methods in Enzymology*, Vol. 601 (Elsevier, 2018) pp. 111–144.
[8] T. Hanscom and M. McVey, Regulation of error-prone dna double-strand break repair and its impact on genome evolution, Cells **9**, 1657 (2020).
[9] M. Tarsounas and P. Sung, The antitumorigenic roles of brca1–bard1 in dna repair and replication, Nature Reviews Molecular Cell Biology **21**, 284 (2020).
[10] T. Gillyard and J. Davis, Dna double-strand break repair in cancer: A path to achieving precision medicine, International review of cell and molecular biology **364**, 111 (2021).
[11] W. Haque, A. Aravind, and B. Reddy, Pairwise sequence alignment algorithms: a survey, in *Proceedings of the 2009 conference on Information Science, Technology and Applications* (2009) pp. 96–103.
[12] V. Y. Khodaverdian, T. Hanscom, A. M. Yu, T. L. Yu, V. Mak, A. J. Brown, S. A. Roberts, and M. McVey, Secondary structure forming sequences drive SD-MMEJ repair of DNA double-strand breaks, Nucleic Acids Research **45**, doi:10.1093/nar/gkx1056 (2017).
[13] Y. S. Rong and K. G. Golic, The homologous chromosome is an effective template for the repair of mitotic dna double-strand breaks in drosophila, Genetics **165**, 1831 (2003).
[14] S. B. Needleman and C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of molecular biology **48**, 443 (1970).
[15] W. R. Pearson, An introduction to sequence similarity ("homology") searching, Current protocols in bioinformatics **42**, 3 (2013).
[16] C.-T. Lee and S.-L. Peng, A pairwise alignment algorithm for long sequences of high similarity, in *Information and Communication Technology: Proceedings of ICICT 2016* (Springer, 2018) pp. 279–287.
[17] S. F. Altschul, Generalized affine gap costs for protein sequence alignment, Proteins: Structure, Function, and Bioinformatics **32**, 88 (1998).
[18] M. A. Zachariah, G. E. Crooks, S. R. Holbrook, and S. E. Brenner, A generalized affine gap model significantly improves protein sequence alignment accuracy, Proteins: Structure, Function, and Bioinformatics **58**, 329 (2005).
[19] C. Kingsford, General and affine gap penalties, https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/gaps.pdf.
[20] P. J. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants, Nucleic acids research **38**, 1767 (2010).
[21] S. Ambardar, R. Gupta, D. Trakroo, R. Lal, and J. Vakhlu, High throughput sequencing: an overview of sequencing chemistry, Indian journal of microbiology **56**, 394 (2016).
[22] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup, The sequence alignment/map format and samtools, bioinformatics **25**, 2078 (2009).
[23] T. S. F. S. W. Group, Sequence alignment/map format specification, https://samtools.github.io/hts-specs/SAMv1.pdf (2022).