รายงานวิจัยฉบับสมบูรณ์

โครงการ: การตรวจวัดความหลากหลายทางพันธุกรรมในข้อมูลจีโนมของกลุ่มคน

โดย ดร. เบญจรัตน์ ภู่ภักดี

กันยายน 2555

สัญญาเลขที่ TRG5280016

รายงานวิจัยฉบับสมบูรณ์

โครงการ: การตรวจวัดความหลากหลายทางพันธุกรรมในข้อมูลจีโนมของกลุ่มคน

ดร. เบญจรัตน์ ภู่ภักดี สถาบันวิจัยจุฬาภรณ์

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

(ความเห็นในรายงานนี้เป็นของผู้วิจัย สกว. ไม่จำเป็นต้องเห็นด้วยเสมอไป)

กิตติกรรมประกาศ

งานวิจัยนี้เสร็จสมบูรณ์ได้เพราะผู้วิจัยได้รับการสนับสนุนจากทั้ง สำนักงานกองทุนสนับสนุนการวิจัย และ สถาบันวิจัยจุฬาภรณ์ ให้ได้มีโอกาสได้ทำงานวิจัยนี้ ผู้วิจัยเชื่อและหวังว่าจะสามารถนำไปใช้ให้เกิดเป็นประโยชน์ต่อสังคมโดยรวมได้

ผู้วิจัยขอขอบพระคุณ รองศาสตราจารย์ ดร. คุณหญิงมธุรส รุจิรวัฒน์ สถาบันวิจัยจุฬาภรณ์ และ ศาสตราจารย์ ดร. มูฮัมมัด เจ ซากิ จาก Rensselaer Polytechnic Institute (นิวยอร์ค สหรัฐอเมริกา) เป็นอย่างสูง ที่ได้ให้ความกรุณาเป็นนักวิจัยที่ปรึกษาในโครงการนี้

รหัสโครงการ: TRG5280016
ชื่อโครงการ: การตรวจวัดความหลากหลายทางพันธุกรรมในข้อมูลจีโนมของกลุ่มคน
ชื่อนักวิจัย: ดร. เบญจรัตน์ ภู่ภักดี สถาบันวิจัยจุฬาภรณ์
Email Address: benjarath@cri.or.th
ระยะเวลาโครงการ: ๑๖ มีนาคม ๒๕๕๒ – ๓๐ กันยายน ๒๕๕๕

บทคัดย่อ

ปัจจุบันนี้เทคโนโลยีการถอดรหัสพันธุกรรมทั้งจีโนมได้มีการพัฒนาก้าวหน้าไปมาก ซึ่งช่วยให้การถอดรหัสข้อมูลพันธุกรรมส่วนบุคคลสามารถเป็นไปได้ในราคาที่ถูกและใช้ระยะเวลาสั้นลงเป็นอย่างมาก คาดว่าจะมีการประยุกต์นำเทคโนโลยีดังกล่าวไปใช้ทั่วไปเร็ว ๆนี้ รวมถึงการใช้ความจำเพาะของรหัสพันธุกรรมส่วนบุคคล ร่วมกับการรักษาของแพทย์ เพื่อมุ่งเน้นวิธีการรักษาโรคที่เหมาะสมในแต่ละบุคคลด้วย การพัฒนาอย่างรวดเร็วของเทคโนโลยีการถอดรหัสพันธุกรรมทั้งจีโนม เป็นสาเหตุหนึ่งที่ทำให้ความหลากหลายทางพันธุกรรมแบบ novel Sequence ได้ถูกนิยามอย่างเป็นทางการและได้รับความสนใจในวงการวิจัยมากขึ้นเมื่อไม่นานมานี้ ความหลากหลายทางพันธุกรรมชนิดนี้ สามารถพบได้ในข้อมูลพันธุกรรมมนุษย์แบบ *de novo* ซึ่งงานวิจัยที่ผ่านมาได้พบว่า ในข้อมูลพันธุกรรมของมนุษย์คนหนึ่ง อาจมี novel sequence ได้มากถึง 3 – 5 ล้านเบส ดังนั้นการพัฒนาวิธีวิจัยเพื่อค้นหา novel sequence ในข้อมูลพันธุกรรมส่วนบุคคลอย่างมีประสิทธิภาพและประสิทธิผล จึงมีความสำคัญเป็นอย่างยิ่ง

รหัสพันธุกรรมทั้งจีโนมแบบ *de novo* มีรูปแบบต่างจากรหัสพันธุกรรมทั้งจีโนมแบบ mapping โดยรหัสแบบ *de novo* ประกอบไปด้วย scaffolds และ contigs ซึ่งมีขนาดเล็กใหญ่ต่างกันมาก อีกทั้งไม่มีข้อมูลเบื้องต้นว่า scaffolds และ contigs เหล่านี้มาจากส่วนใดของโครโมโซม การหา novel sequence ต้องใช้ข้อมูลดังกล่าว ซึ่งสามารถหาได้โดยวิธีเปรียบเทียบความเหมือนของรหัสพันธุกรรมแบบ *de novo* กับรหัสพันธุกรรมอ้างอิง (human reference sequence) การเปรียบเทียบดังกล่าวต้องใช้โปรแกรมคอมพิวเตอร์ (sequence aligner) ปัจจุบันนี้ยังไม่มี sequence aligner ที่พัฒนามาโดยเฉพาะเจาะจงเพื่อการเปรียบเทียบดังกล่าว โปรแกรมที่ก้าวหน้าที่สุดเท่าที่จะหาได้ในปัจจุบันยังต้องใช้ computing resources สูง และใช้เวลานานในการคำนวณ

ในงานวิจัยนี้ ได้มีการพัฒนาโปรแกรม **NSIT (Novel Sequence Identification Tool)** ขึ้นเพื่อการเปรียบเทียบความเหมือนของรหัสพันธุกรรมทั้งจีโนมของมนุษย์ แบบ *de novo* กับรหัสพันธุกรรมทั้งจีโนมของมนุษย์ แบบอ้างอิง โดยเป้าหมายหลักคือเพื่อการค้นหา novel sequence ในรหัสพันธุกรรมแบบ de novo โดยเฉพาะ โปรแกรม NSIT สามารถค้นหา novel sequence ได้อย่างแม่นยำและรวดเร็วภายในเวลาไม่กี่ชั่วโมง โดยใช้คอมพิวเตอร์ทั่ว ๆไปได้ ผลการทดลองที่ได้มี sensitivity เกือบ 100% และ precision ประมาณ 90%

คำหลัก: รหัสข้อมูลพันธุกรรมส่วนบุคคลของมนุษย์ แบบ *de novo*, ความหลากหลายทางพันธุกรรม novel sequence, โปรแกรมด้าน bioinformatics

Project Code: TRG5280016
Project Title: Detection of Genetic Variation in Multiple Personalized Genomes
Investigator: Dr. Benjarath Pupacdi Chulabhorn Research Institute
Email Address: benjarath@cri.or.th
Project Period: 16 March 2009 – 30 September 2012

Abstract

Motivation: Recent developments in DNA sequencing technology have been paving the road to faster and cheaper personal genome sequencing. Many useful applications have been anticipated, such as, routinely performing human genome comparisons to detect sequence variation when designing personalized medical treatments. Novel sequences have recently been discovered as a new class of sequence variation detectable in a *de novo* assembly, where as much as 3–5 Mb of novel sequences may be present in a *de novo* personal genome. Detecting them effectively and efficiently is therefore an important task.

A typical *de novo* human genome assembly, unlike a mapping assembly, consists of a very large number of scaffolds and contigs whose lengths vary highly, and chromosomal positions and orientations are unknown. Identification of novel sequences in the assembly requires comparing it to the reference sequence. However, the input characteristics make the problem computationally expensive. Existing state-of-the-art large genome aligners could require several days to run, when applied to this problem.

Results: We introduce **NSIT (Novel Sequence Identification Tool)**, a software for aligning a *de novo* genome assembly against the reference genome assembly to identify novel sequences. NSIT is fast, accurate, and requires only modest computing resources. To the best of our knowledge, NSIT is the first algorithm designed specifically for this task. Our algorithm requires only a few hours on a commodity desktop and yields high quality results (near 100% sensitivity and about 90% precision).

Keywords: human genome *de novo* assembly, novel sequences, bioinformatics software tool

NSIT: Novel Sequence Identification Tool

1 Introduction

Recent works have identified 3–5 Mb of novel DNA sequences in each of the following 3 individual genomes via *de novo* assembling: YH (male Chinese), NA18507 (male Yoruba), and NA18943 (male Japanese) [14, 6]. Novel sequences are sequences present in at least one human individual but absent in the reference genome, i.e., they can be classified as long insertions. To be biologically meaningful, they are defined to be >100 bp long, with <90% identity to the reference genome, and not known repeats because their paralogs would exist elsewhere in the reference genome rendering them not novel by definition. [14] shows that most novel sequences are individual or population specific and consistent with the known migration paths. They also are predicted to carry potentially functional coding regions. In addition, the human pan-genome is estimated to contain about 19–40 Mb of novel sequences. These recent discoveries show that a considerable amount of undiscovered genetic variation in the human genome remains, and underscore the importance of whole genome sequencing as well as *de novo* assembly.

A *de novo* assembly is constructed by joining sequence reads together without consulting a reference genome which creates new, previously unknown, assembled sequences. A human genome *de novo* assembly typically consists of a few hundred thousand scaffolds and contigs whose lengths vary greatly. Since a reference genome is not used, assembling *de novo* cannot by itself reveal orientations and chromosomal positions of the scaffolds and contigs with respect to the reference genome. Both YH and NA18507 genomes are *de novo* assemblies and were formed using the SOAPdenovo software [15]. In order to identify novel sequences in them, the scaffolds and contigs of each assembly were aligned against the NCBI Build 36 reference sequence using BLAT [11], LASTZ [18], and BLASTn [1], in that order. Approximately 5 Mb of unaligned regions were defined in each assembly as novel sequences. In contrast, the NA18943 genome was assembled via mapping the sequence reads against NCBI Build 36. The unmapped reads were assembled *de novo* using the ABySS [19], SOAPdenovo, and Velvet [23] software. Approximately 3 Mb of novel sequences were identified.

The above approaches discover novel sequences via whole genome assembling. An alternative route is to directly use paired-end read data from next-generation sequencing (NGS) platforms without assembling the entire genome. Recent such developments include VariationHunter [9, 10], MoDIL [13], BreakDancer [3], PEMer [12], GASV [20], HYDRA [16], and Pindel [22]. While these methods address various types of structural variation, novel sequences remain difficult to resolve, particularly when they are longer

than the read length and insert size used in the NGS methods (which are not very long). For example, the recommended read length and insert size for the Illumina Genome Analyzer platform [2] are <100 bp and 200 bp – several kb, respectively. Currently, the only method capable of discovering long insertions in paired-end read data is NovelSeq [8]. The algorithm essentially prunes away sequence reads mappable to the reference genome and assembles *de novo* the unmapped reads. The assembled sequences are then anchored back into the reference genome, thereby discovering both the content and location of the novel sequences. A core strength of NovelSeq is in its ability to perform efficiently without a large computational resource requirement that whole human genome *de novo* assembling imposes. The sets of novel sequences in the NA18507 genome predicted by NovelSeq and by SOAPdenovo, although not identical, largely overlap: 2.63 out of 2.66 Mbp of novel sequences identified by NovelSeq are in the 4.8 Mbp set of novel sequences identified by SOAPdenovo. Similar to other types of structural variants, the novel sequences are difficult to identify *precisely* due to the complex repeating nature of the human genome. In fact, while many types of structural variation can be inferred from non-assembled sequence reads, accurate biological insights at the nucleotide level require substantially more. Thus, a variety of approaches should be used.

De novo assembling of individual genomes is a critical step to fully annotate insertions, deletions, and other structural variants. There are a few large whole genome *de novo* assemblers available. SOAPdenovo was used in [14] to assemble *de novo* the YH and NA18507 genomes. Identification of novel sequences in these *de novo* assemblies were carried out as follows. Firstly, BLAT was used to align the assembly with NCBI Build 36 to assign candidate chromosomal locations for all *de novo* contigs and scaffolds. Next, the state-of-the-art whole-genome aligner LASTZ performs a more detailed alignment in the identified candidate homologous regions. The unaligned regions from LASTZ were considered candidate novel sequences, which were aligned again in more details to the reference sequence via BLASTn. The final unmapped non-repeat regions were reported as novel sequences. Using the three aligners in this order follows the popular trend adopted in most speedy aligners – a faster but less sensitive alignment step is taken first as a heuristic to identify regions likely to be homologous and then more detailed alignments are performed on the previously defined homologous regions.

While BLAT and LASTZ are well-established fast large-genome aligners, they were not created with the purpose of identifying novel sequences in a *de novo* assembly; the algorithms perform computations unnecessary to this problem. In this work, we present a new algorithm called **N**ovel **S**equence **I**dentification **T**ool or ***NSIT***. To the best of our knowledge, NSIT is the first algorithm specialized for identifying novel sequences in a *de novo* assembly. It is magnitudes faster than BLAT and LASTZ and it requires only modest amount of computing resources. We tested NSIT with the YH and NA18507 *de novo* whole genome assemblies. Our experimental results show that NSIT efficiently aligned each assembly to the reference sequence and identified the novel sequences with nearly 100% sensitivity and ∼90% precision, in under 2–3 hours using <2GB of RAM on a 32-bit commodity desktop. We present the details of NSIT in the following section.

2 Methods

In this section, we examine why the existing fast whole-genome aligners BLAT and LASTZ are not suitable for the task of novel sequence identification in a *de novo* whole human genome assembly. Then we present our algorithm NSIT in details.

BLAT [11] was originally designed for aligning several million ESTs and mouse whole-genome random reads (a few hundred bps in length on average) against the reference human genome. The algorithm indexes non-overlapping k-mers of the reference sequence, scans every position of the query rapidly for hits which may include gaps or mismatches, and clumps them together to identify candidate homologous regions. Using non-overlaping k-mers allows the index table for the entire reference genome to fit in just under 1GB of memory. Next, the detailed alignment stage recursively looks for higher-quality hits in the candidate homologous regions, extends and merges the hits into high-scoring pairs (HSPs), and subsequently links them together creating longer alignments. When used in aligning nearly identical sequences, the dynamic progamming stage that finalizes the alignments can be skipped to reduce run time.

LASTZ [18], or originally BLASTZ, is the state-of-the-art pairwise alignment algorithm for large sequences. LASTZ also uses index tables of k-mers. However, unlike BLAT, these k-mers may overlap to examine every reference position for better sensitivity. LASTZ scans every position in a query to search for short matches, which are extended to create HSPs. The HSPs are chained and reduced into single locations called anchors. The anchors are extended to high-score gapped local alignment blocks. Alignment interpolation is performed between the blocks. The entire process is recursively repeated at a higher sensitivity in the unaligned regions until no further alignments can be formed.

BLAT and LASTZ have several alignment steps, which range from faster but less sensitive steps to slower but more sensitive ones. Although the details differ, both algorithms use k-mers to create short alignment seeds, where a seed generally corresponds to several locations on the input sequences. Longer alignment stretches are subsequently constructed via various alignment extension and gap-filling techniques, including dynamic programming. These solutions are appropriate for the programs' intended tasks and can certainly be applied to align a *de novo* assembly with the reference sequence to search for novel sequences. However, they would perform more computations than necessary causing them to be less efficient.

NSIT tackles the problem of aligning a *de novo* human genome assembly with the reference sequence to locate novel sequences in a very efficient manner. The algorithm relies on the fact that two human genomes are nearly identical, therefore once a *good* match is anchored, extending it to create a longer alignment should be relatively straightforward. Applying the domain knowledge specific to the problem allows NSIT to be both fast and sensitive.

NSIT algorithm has 3 main steps: *1) k-mer Hash Tables Construction*, *2) Assignment*, and *3) Alignment*. The first step creates indexing data structures, which are 24 k-mer hash tables, one for each reference chromosome. Based on these tables, the *As-*

signment phase quickly and approximately performs all-against-all alignments between the *de novo* and reference sequences, i.e., a few hundred thousand shorter sequences vs. 24 chromosome sequences. We found that most *de novo* sequences align in a certain orientation to a particular chromosome with a statistically significant high score. As a result, they are assigned that specific location for further fine-scale alignment in the *Alignment* phase. A small portion of the *de novo* assembly however cannot be assigned in this manner and must undergo all-against-all fine-scale comparison. The unaligned regions longer than 100 bp become candidate novel sequences and are aligned against the reference genome via BLASTn and searched for repeats via RepeatMasker. The final unaligned regions are identified as novel sequences.

2.1 K-Mer Hash Tables Construction

NSIT indexes all possible k-mers in the reference genome. The first phase constructs 24 hash tables, one for each reference chromosome, linking each k-mer to all of its chromosomal locations. Let $S = s_0 s_2 \dots s_{k-1}$ be a k-mer; Σ denotes the DNA alphabet set $\{A, C, G, T\}$, and $|\Sigma| = 4$. Define m be a 1-to-1 function from Σ to $\{1, 2, 3, 4\}$ such that $m(A) = 0$, $m(C) = 1$, $m(G) = 2$, and $m(T) = 3$. The hash function $h(S)$ is defined as follows:

$$h(S) = \sum_{i=1}^{k} (m(s_i) \times |\Sigma|^{k-i})$$

Simply put, $h(S)$ is an integer that equates the quaternary value of S and serves as the table index. To populate a table H_i for a chromosome R_i, $i \in \{1, 2, \dots, 22, X, Y\}$, R_i is scanned once from left to right, one position at a time, while the k-mer starting positions are recorded into the table. Our hash function allows the first k-mer to be the only one whose hash index is computed in $O(k)$ time. The rest of the k-mers can be computed using $O(1)$ time via bit shifting operations, which helps save time in practice because the algorithm must scan the entire human genome. The total time required to construct all hash tables is linear to the human genome size, i.e., approximately 3 Gb.

As for the space requirement, each table H_i stores $|R_i| - k + 1$ positions using $|\Sigma|^k$ slots. Since $|R_i|$ and $|\Sigma|$ are constant, the space complexity depends directly and exponentially on k. In nature, the DNA nucleotides are not uniformly distributed over the entire genome, as biologically meaningful sequences possess specific patterns. Therefore, we cannot assume that the hash buckets will be of approximately even lengths. When using a particular hash table, NSIT loads the entire table to the memory for fast access. These factors must be considered when selecting an appropriate value for k. A smaller k saves space but retards the search time while a larger k's faster search time comes with a cost of exponentially bigger space requirement as well as a potentially large number of empty hash slots. The goal is to balance between a fast search time and a reasonable space requirement. We investigated different values for k on Chromosome 1 (the largest human chromosome), NCBI Build 36 reference sequence. Table 1 shows the total number of

hash slots and the amount of empty slots experimentally obtained for $k \in [9, 13]$. For $k < 9$, there are no empty slots.

Table 1: Total numbers of hash slots and empty slots from Chromosome 1, NCBI Build 36 reference sequence, for $k \in [9, 13]$

k	**# Total Slots** (4^k)	**# Empty Slots**
9	262,144	10,486 (4%)
10	1,048,576	188,744 (18%)
11	4,194,304	1,803,551 (43%)
12	16,777,216	12,079,596 (72%)
13	67,108,864	60,397,978 (90%)

The table suggests $k = 9$ or 10 to be reasonable choices. We found experimentally that $k = 10$ works well and used this value across all our experiments; the largest hash table, H_1, requires about 1GB of memory. Twenty-four hash tables corresponding to each reference chromosome are constructed and stored on disk one at a time, therefore the memory requirement for this phase does not exceed 1GB (approximately). The hash tables are used in memory during the subsequent steps, also one at a time. Overall, the hash table construction has a one-time cost.

2.2 Assignment Phase

A *de novo* human genome assembly typically contains a large number of contigs and scaffolds whose lengths vary greatly. For example, there are 136,926 contigs and 48,160 scaffolds in the YH assembly. The sequence lengths are in the [100, 4,532] and [108, 3,449,040] bp ranges respectively. By way of *de novo* assembling, chromosomal positions as well as orientations of these sequences are not available. A main challenge is in effectively and efficiently selecting candidate homologous regions in the reference genome to align with the query sequences: it would certainly be computationally expensive to do all-against-all detailed comparisons between these hundreds of thousands *de novo* sequences and the 24 reference chromosomes. The goal of the *Assignment Phase* is to quickly identify the appropriate target reference chromosomes and orientations for each of the query *de novo* sequences. The focus here is on speed and not sensitivity. Detailed alignments follow in the next phase.

Since any two human genomes are $\geq 99\%$ identical in the DNA bases, a large number of k-mer matches between them can surely be identified, and aligning any two such highly homologous regions using either back-to-back consecutive k-mers or non-consecutive k-mers should likely yield similar results. Figure 1, case 1, illustrates having a perfect match between the query sequence Q and reference sequence R. Let $k = 2$ for the sake of simplicity, aligning R and Q from left to right with consecutive k-mers AA, GG, CC, and so on would yield the same alignment as aligning them via non-consecutive k-mers with a 5-bp skip between them, i.e., AA, TA, TT, and so on. This means, for our purposes, a sparse subset of k-mers similar to {AA, TA, TT, ...} can be used to align reasonably well a query sequence to its appropriate homologous regions. Using

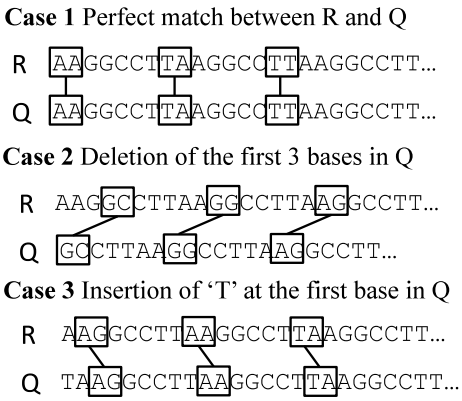

Figure 1: Three cases of non-consecutive k-mer matching between a reference sequence R and a *de novo* query sequence Q. Case 1 illustrates a perfect match. Cases 2 and 3 show a deletion and an insertion in the query sequence, respectively. Each case requires a different sparse set of k-mers.

non-consecutive k-mers can vastly improve both time and space requirements because the number of k-mers involved can be reduced by several times. A query region that is not truly homologous would simply not match with the set, while a truly homologous region would almost always match, with the exception of genetic variants such as SNPs, insertions, and deletions. Cases 2 and 3 in Figure 1 demonstrate the scenarios when there is a deletion or an insertion in the query sequence. Case 2 shows a deletion of the first three bases in Q, which results in another sparse set of k-mer matches {GC, GG, AG, ...}. Similarly, case 3 shows an insertion of a base 'T' at the start of Q and results in another sparse set of k-mer matches {AG, AA, TA, ...}.

NSIT utilizes all three cases of sparse k-mer sets to quickly and approximately locate a query's candidate homologous regions. For each reference chromosome R_i, H_i is loaded into the memory such that only the k-mer positions contributing to the selected sparse sets are filtered in. The underlying logic is to only partially load the k-mers and save on computational time and space while still incurring enough exact matches that result in real alignments. Let a user-defined parameter $skipSize$ denote the number of bases skipped between any two k-mers in a particular set. The k-mer set in case 1 should definitely be used, i.e., loaded from the hash tables into main memory, for the obvious reason. To cater for insertions or deletions and increase our chances of finding exact matches, the algorithm shifts some bases and loads the respective sparse sets of k-mers into the memory as well. The more k-mer sets loaded, the more sensitive the program becomes, and vice versa. Let another user-defined parameter $setNum$ be the number of k-mer sets to be loaded such that $setNum$ is a positive integer and $(k + skipSize)$ is divisible by it. Let $(k + skipSize)/setNum = loadSkip$. For each R_i, only the positions divisible by $loadSkip$ from its hash table are allowed into the memory. This guarantees that strictly $setNum$ sparse sets of k-mers, with $skipSize$ bases between any two k-

mers in a particular set, are loaded into the main memory. The first k-mer in any two consecutive sets are $loadSkip$ bases apart. For example, let $k = 10$ and $skip = 80$. If $loadSkip = 90$, only 1 k-mer set $\{0, 90, 180, \ldots\}$ would be loaded. If $loadSkip = 45$, 2 k-mer sets $\{0, 90, 180, \ldots\}$ and $\{45, 135, 225, \ldots\}$ would be loaded, and so on.

For each chromosome R_i, its hash table is loaded in the above manner into memory. The program scans each query Q_j from left to right and attempts to align it against R_i by matching as many k-mers between them as possible. In other words, the alignment is maximal in both directions. We found that if such maximally aligned regions are sufficiently long, they often only occur in exactly one, i.e., unique, location on R_i. Since the two genomes are highly homologous, it translates that chances are the alignment only occurs once between the pair. It is a well accepted principle that long maximal unique matches are good alignment anchors and generally lead to a true alignment [4, 5]. $skipSize$ bases are skipped after every k-mer match. Once no more k-mer matches can be found for the current stretch of alignment, the process simply restarts at the base following the last aligned region until the end of Q_j is reached. The search is applied independently to both orientations of Q_j. In a particular orientation, any nucleotide base is never scanned more than once. There is no minimum length requirement for an alignment in this step. Due to the low sensitivity, many spurious matches are expected to arise. Figure 2 demonstrates how NSIT removes them and computes the alignment score between R_i and Q_j. Since the longest aligned region is most likely to constitute a true global alignment, it is used as a heuristic for a true alignment. In this example, it is A_1. Spurious alignments are filtered out based on the principle that true aligned regions should approximately lie along the diagonal of the longest alignment, plus or minus the insertion and deletion ranges. The aligned regions A_2 and A_3 represent the cases when there are insertions and deletions in the query. Define the distance between any two aligned regions to be the difference between their y-intercepts, i.e., their locations on R_i. Any aligned region within $threshold$ distance from the longest aligned region are kept and others are discarded. In this example, the candidate true alignment contains the regions A_1, A_2, and A_3. We experimentally found that the $threshold$ value between 0.1–1 Mb works well in real data. The alignment score is defined to be the number of bases included in the candidate true alignment. Each query maintains a total of 48 scores, one for a particular reference chromosome in a specific orientation.

The assignment of target chromosome(s) and orientation(s) to a query Q_j is done as follows. Let $score_{(ij,p)}$ be the alignment score between R_i and Q_j in the orientation p where $i \in \{1, 2, 3, \ldots, X, Y\}$ and $p \in \{+, -\}$. Since a spurious alignment may occur randomly due to a variety of independent factors with no one factor being the most important, it is reasonable to expect the distribution of the scores from chance matches of any Q_j to be approximately Gaussian. In addition, the true alignment score of Q_j should deviate significantly from the mean score of the false alignments. To assess whether Q_j truly maps to R_i in the orientation p, let $score_{avg}$ and $score_{SD}$ be the mean and standard deviation of all scores $score_{(ik,q)}$, such that $j \neq k$ and $p \neq q$. The *Assignment Phase* assigns any chromosome R_i in orientation p to Q_j if and only if $(score_{(ij,p)} - score_{avg}) \geq 2.5 score_{SD}$, which means $score_{(ij,p)}$ has p-value < 0.01, i.e., is

highly statistically significant, signaling a real alignment. We show in the Results section that the majority of scaffolds and contigs were correctly, quickly, and uniquely assigned a chromosome and an orientation in this manner. Due to the approximate nature of the algorithm in this phase, some queries especially ones that are repeat rich are assigned more than one chromosome and/or orientation. Some are incorrectly assigned and some simply could not be assigned. These latter two cases warrant further all-against-all detailed alignments in both directions in the next phase, however we found that they amount to a very low percentage and most of them are very short. Therefore, the overall run time does not suffer in practice. Detailed alignments between a query and its assigned reference homologous region(s) are carried out in the next phase.

The time and space complexity for the *Assignment Phase* can be analyzed as follows. Let n be the number of bases in the human genome, i.e., $n = 3 \times 10^9$. All hash tables are scanned once when being loaded, so this takes $O(n)$ time. The *de novo* assembly is scanned once in each orientation, for each chromosome, therefore in total this takes $O(24 \times 2n)$ time. A maximal unique aligned region is created by intersecting the position lists between the current aligned region and the next k-mer until the intersection contains a unique location. It was found that nearly 80% of the human genome is composed of unique 25-mers [15]. Therefore, we can generally expect that each maximal unique aligned region is created in $O(1)$ time. Filtering out spurious matches takes $O(n)$ time and finding statistically significant aligned regions takes $O(1)$ time. Therefore, in total this phase takes $O(n)$ time. Since sparse sets of k-mers are used one chromosome at a time and a query is analyzed also one at a time, the memory required for this phase is quite affordable, i.e., $<$ 1GB.

2.3 Alignment Phase

In the *Alignment Phase*, the algorithm aligns the *de novo* contigs and scaffolds to their target chromosomes and orientations in the manner similar to the *Assignment Phase*. The key difference is that the entire hash tables, i.e., all possible k-mers, are used in order to increase the alignment sensitivity. Although higher sensitivity generally implies more run time and that there remain some short query sequences to be aligned to all chromosomes after the previous phase, the majority of *de novo* query sequences are assigned to a unique chromosome and orientation. Thus, this phase approximately requires $O(n)$ time. Resulting aligned regions are extended by direct base comparison as far as possible in both directions. Some alignment overlaps may result and are resolved by giving a higher priority to the longer alignment. The heuristic nature of the *Assignment Phase* may also cause some false assignments. We consider any scaffold or contig not aligned to its assigned chromosome for more than half its length to be false and realign it to all reference chromosomes in both orientations for further validation. In addition, we found that *de novo* assembling may occasionally fuse DNA sequences from multiple chromosomes together into longer scaffolds. Large (>2.5 kb) unaligned regions in a query may not actually be novel but rather simply belong to another chromosome and warrant further fine-scale alignments to the chromosomes other than what was assigned to them. NSIT therefore subsequently aligns these regions against all reference chromosomes

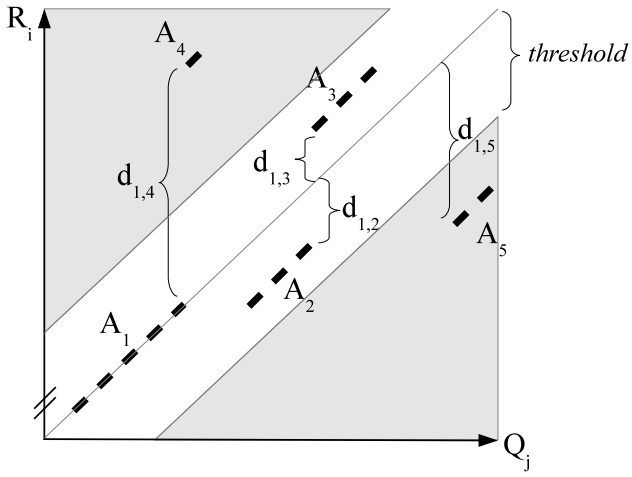

Figure 2: Alignment between a reference chromosome R_i and a query sequence Q_j using non-consecutive k-mers. The initial alignment consists of five smaller aligned regions A_1 to A_5. A_1 is the longest region and used as a heuristic for a true alignment. The distances $d_{1,2}$ to $d_{1,5}$ measure the differences between the y-intercepts of A_1 to A_2 and A_5 respectively. The aligned regions A_1, A_2, and A_3 are kept as candidates for the true alignment because $d_{1,2}$ and $d_{1,3} \leq threshold$. The alignment score is the number of DNA bases covered in the true alignment candidate regions.

in both orientations and the alignment with the highest score is chosen as the true alignment. Finally, the unaligned regions in the queries are passed on as novel sequence candidates to BLASTn, which performs the last fine-scale sequence alignment between these candidates and the reference human genome. The unaligned regions are searched for repeats via RepeatMasker. Final unaligned regions that are at least 100 bp long and not known repeats are reported as novel sequences.

3 Results

In this section, we discuss the performance of NSIT, specifically the quality of the results, the run time, and the computing resource consumption. All of our experiments were performed on a 32-bit Linux machine with Intel ® Xeon ® 3.00 GHz quad-core processors, 4GB of RAM, and 500GB of hard disk. The amount of RAM utilized across all experiments never exceeded 2GB. Despite being tested on a multi-core machine, NSIT does not use parallel processing.

We experimented with both the YH (Asian male) and NA18507 (African male) *de novo* whole genome assembly data. As earlier described, both assemblies consist of a large number of scaffolds and contigs of highly variable lengths and unknown orientations. These two genomes were sequenced from two different research facilities and possess an important distinct characteristic such that the YH assembly has a longer N50

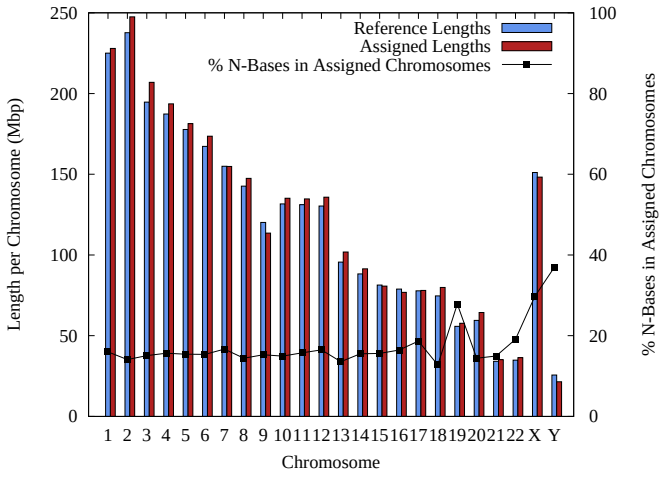

Figure 3: Supporting evidence that the *Assignment Phase* works correctly, despite using a heuristic to achieve its rapid speed. 1) The total lengths of YH *de novo* sequences assigned per chromosome closely mirror the true reference chromosome lengths (with 95% sequence identity on average). 2) The amount of N-bases in the assigned sequences are consistent with the amount of known repeats, e.g., Chromosomes 19, X, and Y are repeat-rich.

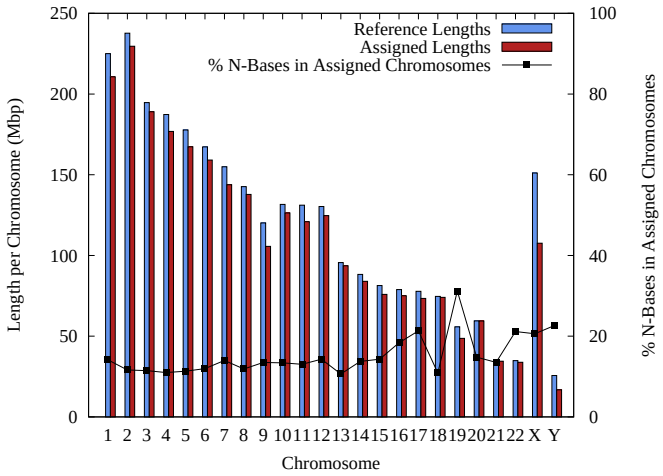

Figure 4: Supporting evidence that the *Assignment Phase* works correctly, despite using a heuristic to achieve its rapid speed. 1) The total lengths of NA18507 *de novo* sequences assigned per chromosome closely mirror the true reference chromosome lengths (with 95% sequence identity on average). 2) The amount of N-bases in the assigned sequences are consistent with the amount of known repeats, e.g., Chromosomes 19, X, and Y are repeat-rich. In addition, chromosomes 16, 17, and 22, which have been reported to be enriched with segmental duplications, were the only other three chromosomes whose N-base percentages exceeded the mean percentage.

size than the NA18507 assembly does for both contigs (7.4 kb vs. 5.9 kb) and scaffolds (446.3 kb vs. 61.9 kb). It was reported in [15] that this is likely due to the longer average read length (55 bp vs. 35 bp) and longer paired-end insert size (9.6 kb vs. 2 kb) of the Asian genome sequencing data. A further inspection of the two assemblies reveals that there exist about 3.5 times fewer scaffolds in YH than there are in NA18507. The maximum scaffold length in YH is also about 4.2 times longer (8.2 kb vs. 3.5 Mb). Based on this information, we speculated that the YH assembly would be an easier data set to decipher with NSIT than the NA18507 assembly due to its fewer and longer scaffolds, hence fewer alignment scores to process.

3.1 Assigned Chromosomes

We evaluate the performance of the *Assignment Phase* in this section. The goal of this phase is to speedily assign a correct reference chromosome and orientation to each contig and scaffold of the input *de novo* genome assembly. As described in the Methods section, the phase's fast speed relies on the heuristic that any two human genomes are highly homologous and aligning them even with a large gap between every 2 k-mers should still suffice to detect true alignments. NSIT was run on the YH *de novo* assembly, with $k = 10$, $skipSize = 80$, and $loadSkip = 5$, i.e., a 80-bp gap was allowed between any two 10-mers constituting an alignment, and the hash tables were loaded with 5 times less reference information than what the tables actually store. Figure 3 shows the following result statistics. The total number of nucleotide bases assigned per chromosome in comparison to the reference chromosome lengths are displayed with the pairing bars. The data elicit striking resemblance between the two lengths on every chromosome. In addition, the assigned sequences added up to 95% sequence identity to the reference chromosomes on average. These results strongly support our hypothesis that the *de novo* sequences were assigned correctly. Nearly all (99%) of the total number of bases in the YH *de novo* assembly were assigned at least a chromosome and an orientation and the majority of them (96%) were assigned *uniquely*, i.e., in precisely one orientation on a particular chromosome. The unassigned *de novo* sequences were mostly very short, i.e., their N50 size was under 4 kb. Therefore, even though they needed to undergo all-against-all alignments in the *Alignment Phase*, they would not require much run time.

In addition to the mirroring lengths, the amounts of N-bases in the *de novo* sequences assigned per chromosome correspond with the repeat content of the chromosomes, which further supports that these sequences were mapped correctly. In general, it is more difficult to sequence and assemble genomic areas that are repeat-rich and these areas are usually denoted as N-bases in the final assembly. For the same experiment above, we displayed the proportions of N-bases in the assigned *de novo* sequences as the line graph in Figure 3. The mean percentage was 17.7%. The standard deviation (SD) was 5.8%. The results showed that chromosomes 19, X, and Y were assigned the most N-bases, which were 27.9%, 29.8%, and 36.9% respectively. In other words, these amounts were >1SD, >2SD, and >2SD away from the mean, which are considered significant. Such statistics conform with what have been reported in literature. [7] shows that nearly 55%

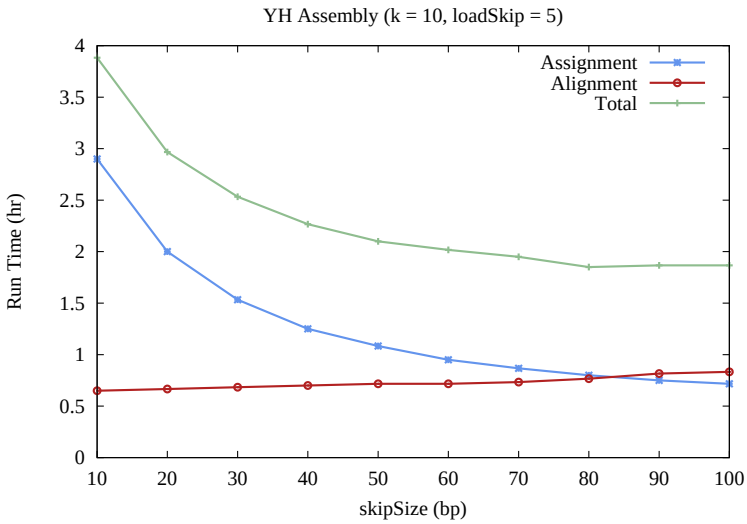

Figure 5: NSIT run time for YH *de novo* assembly. The best total run time of about 2 hours was achieved at $skipSize = 80$.

of the human chromosome 19 consists of repetitive elements. The human chromosome X is also reported in [17] to be highly enriched in interspersed repeats (56%), in comparison to the genome average of 45%. [21] also found that the human chromosome Y is unusually repeat-rich as most of it consists of genes in the form of palindromes. Our results firmly illustrate that NSIT maps the scaffolds and contigs onto the reference genome correctly.

The NA18507 genome exhibited similar results. As much as 95% of its bases were assigned at least a chromosome and an orientation, and the majority of them (96%) were assigned uniquely. The mean and SD percentages of assigned N-bases were 15.2% and 5%. Chromosomes 19, X, and Y were assigned 31.2%, 20.6%, and 22.6% or >2SD, >1SD, and >1SD away from the mean respectively. (Figure 4).

We now examine the run time of NSIT at various *skipSize* values for both YH and NA18507 *de novo* assemblies. The larger value *skipSize* takes, the lesser amount of reference information loaded into the main memory and used in the hash tables, and therefore the faster the *Assignment Phase* becomes. The *Assignment Phase* took 48 and 54 minutes at the data sets' best total run times, i.e., at $skipSize = 80$ and 60 for YH and NA18507, respectively (See Figure 5 above and Figure 6). The African genome took longer to run due to its higher number of *de novo* contigs and scaffolds, and thus more alignment scores to process. In addition, the shorter scaffold sizes prohibited it from taking advantage of larger *skipSize* values. However, a good balance must be acquired because using a larger *skipSize* value, i.e., less amount of reference information, could result in a larger amount of *de novo* sequences needing all-against-all alignments in the next phase, which increases the run time, as evident from the figure. We elaborate further on this in the next subsection. Overall, a *skipSize* value between 50–90 is recommended.

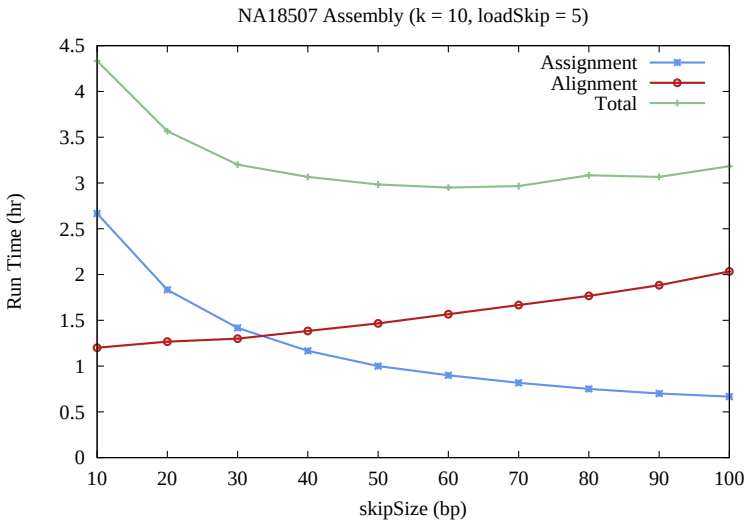

Figure 6: NSIT run time for NA18507 *de novo* assembly. The best total run time of about 3 hours was achieved at $skipSize = 60$.

3.2 Results of the Alignment Phase

The run time of the *Alignment Phase* depends directly on 3 groups of input *de novo* sequences: the correctly assigned sequences, the incorrectly assigned sequences, and the unassigned sequences. The majority (>95%) of *de novo* contigs and scaffolds are assigned candidate homologous regions in the previous phase. Using *all* possible reference k-mers, the *Alignment Phase* aligns these sequences to their corresponding target regions and the remaining unassigned sequences to the entire reference genome, in both orientations. The majority of run time is spent on the alignment of the former group; this marks the run time baseline.

We found that very few (1–3% and 3–8% of the YH and NA18507 genome sizes, respectively) and short (N50 size of 1–6 kb and 2–7 kb for YH and NA18507, respectively) *de novo* sequences constituted the unassigned group across all *skipSize* values. Larger *skipSize* values give rise to more and longer unassigned sequences as well as higher run times for the *Alignment Phase*, and vice versa. These unassigned sequences were found to have rich N-base contents, i.e., approximately 50% in YH and 25% for NA18507, in comparison to their respective genome N-base contents of 18% and 15%. The N-base regions generally represent the areas more difficult to sequence and align, thus it is not surprising that the unassigned sequences produced such ratios. Since the YH assembly is comprised of larger scaffolds, it was easier to assign and hence its unassigned sequences showed better statistics, e.g., 2% vs. 5% of the genome total sizes and 4 kb vs. 4.5 kb N50 sizes, at their best total run times.

Nearly all of the assigned sequences were found to have been assigned correctly. However, since the *Assignment Phase* is heuristic-based, false assignments of target

regions can occur. The falsely assigned sequences shared almost identical features with the unassigned sequences above; only they were shorter and fewer. Across all *skipSize* values in Figure 5, the N50 sizes of these sequences were just 300 bp–1 kb and 3-6 kb and they amounted to smaller fractions of <1% and 1–3%, for YH and NA18507 respectively. Again, larger *skipSize* values imply more incorrectly mapped sequences and higher run times. The N-base characteristics also did not vary from the results above. The YH assembly again had better statistics for this group of sequences than the NA18507 assembly did: total amounts of 0.9% vs. 3% of the genome total sizes and N50 sizes of 814 bp vs. 5.6 kb, at their best total run times.

A lesson learned from above is that the high N-base content causes the sequences to be more difficult to align, particularly with sparse k-mer sets. Fortunately, sequencing technologies are rapidly improving and we can possibly expect a decreasing trend for the amount of N-bases in a finished *de novo* assembly. Therefore, the amount of unassigned and falsely assigned sequences are likely to reduce and contribute to a more negligible portion of the run time in the future, i.e., the *Alignment Phase* run time would become more stable across different *skipSize* values as technology progresses, which would result in a faster total run time of NSIT.

It is worth mentioning that some long (>2.5 kb) stretches of DNA from multiple chromosomes were found to have been fused together into a single scaffold during the *de novo* assembling process. As a result, some of these regions would be unaligned to the originally assigned target regions and appear as possible candidates for novel sequences even though they truly are not novel. NSIT separates such sequences out and realigns them to the whole genome in order to locate their true homologs. For all of our experiments, this group of sequences consistently amounted to very little (<1%) of the total genome size and their N50 sizes were also consistently short, i.e., approximately 4.6 kb. These sequences neither had any N-bases nor contributed to a significant portion of the phase's run time, however, pruning them out reduced the amount of false novel sequence candidates by up to 20 Mb, which is a significant amount considering that the total size of novel sequences in each *de novo* assembly is only ∼5 Mb.

Overall, NSIT correctly assigns most of *de novo* scaffolds and contigs to their true homologous regions and these assigned sequences contribute to the majority of the *Alignment Phase*'s run time. More sequences are unassigned or incorrectly assigned as a larger *skipSize* value is used, thus increasing the phase's run time as displayed in Figure 5. At their best total run times, the *Alignment Phase* took 46 and 94 minutes for the YH and NA18507 assemblies respectively. We expect the run time of this phase to decrease as sequencing technology progresses. At their best total run times, YH and NA18507 required 2 and 3 hours, respectively.

3.3 Selecting the Appropriate *loadSkip* Value

Selecting the appropriate value for the *loadSkip* parameter has a crucial impact on the run time of NSIT. As previously described, $(k+skipSize)/loadSkip$ sparse sets of k-mer positions are used, therefore a larger *loadSkip* indicates more sparsely loaded hash tables, which speed up the assignment time. However, this could lower the program's sensitivity,

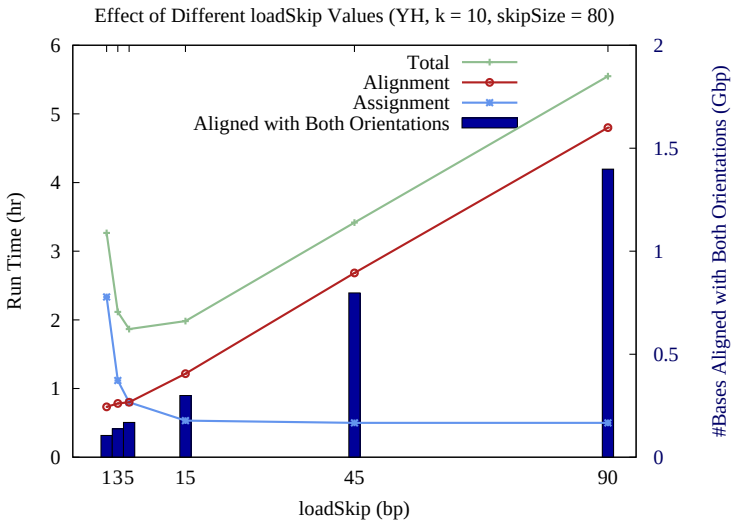

Figure 7: A larger *loadSkip* value indicates more sparsely loaded hash tables which speed up the assignment time. However, it also lowers the sensitivity, causing more *de novo* sequences to not be correctly assigned and need to be aligned in both orientations in the *Alignment Phase*, which slows down the alignment time. With $k = 10$ and $skipSize = 80$, the best balance for the YH assembly was achieved at $loadSkip = 5$ where the size of the loaded hash tables were reduced by a factor of 5 from their actual sizes.

causing more *de novo* sequences to be incorrectly assigned or unassigned entirely. As a correction, they are realigned against the entire reference genome in the *Alignment Phase* to obtain their true homologous regions. We found that the number of sequences requiring the correction very quickly increased as *loadSkip* took on larger values (Figure 7). This directly impacted the run time. At $k = 10$ and $skipSize = 80$, the best balance for YH was achieved at $loadSkip = 5$, i.e., the size of the loaded hash tables were reduced by a factor of 5 from their actual sizes. We also found that this value worked well for all our experiments. Since other human genome *de novo* assemblies would likely exhibit similar characteristics as in YH and NA18507, we recommend assigning a small value between 3–15 to the *loadSkip* variable.

3.4 Novel Sequences

The goal of the NSIT algorithm is to identify novel sequences in a *de novo* genome assembly. The novel sequences are defined to be the *de novo* sequences which are >100 bp long, with <90 % identity to the reference genome, and not known repeats. The *de novo* genome regions unaligned by NSIT are treated as novel sequence candidates, which we post-processed with BLASTn and RepeatMasker to obtain the final list of novel sequences. Using BLAT, LASTZ, and BLASTn, [14] identified 5,125,070 bp and 4,798,833 bp of novel sequences, not present in the NCBI Build 36 reference human genome, for the *de novo* YH and NA18507 assemblies respectively. Our experimental results showed that NSIT identified these novel sequences in both assemblies quite accurately and efficiently, i.e., 98% and 99% sensitivity and 92% and 93% precision for YH and NA18507, within a few hours on a commodity desktop. The details of the results are as follows.

For the YH genome, NSIT identified between 64.1–65.9 Mb of novel sequence candidates across all *skipSize* values. Notice that the amount did not vary significantly. Specifically, 65.2 Mb of candidates were found at $skipSize = 80$. Similar to [14], we set BLASTn thresholds to allow only alignments with identity > 90% and E-value < 1e-20, and aligned the 65.2 Mb candidate sequences against Build 36 database. About 12.3 Mb were found to be unaligned, thus reducing the size of novel sequence candidates. Nearly all, i.e., 5,009,244 bp, out of the reported 5,125,070 bp of novel sequences were included in these unaligned sequences, indicating a high sensitivity of 98%. This entire process was relatively fast (8 mins). We found via BLASTn that nearly 80% of the rest of candidates were present in the GRCh37 and HuRef assemblies, leaving about 1.5 Mb of unidentified candidate bases. To avoid known repeats, RepeatMasker was run on the 1.5 Mb of sequences, resulting in 475 kb of unmasked sequences, which we considered as the false positive bases. Note that the unmasked sequences were quite short, with an average size of 150 bp, and should not be difficult to prune out in practice. The precision of NSIT for YH is therefore 92%. We do not use the specificity to measure the result quality because the number would not be informative, due to the enormous size of the non-novel sequences. Figure 8 graphically displays the above results.

NA18507 was processed in the same manner and exhibited similar results. Between 55.5–68.5 Mb of novel sequence candidates were identified across all *skipSize* values, and

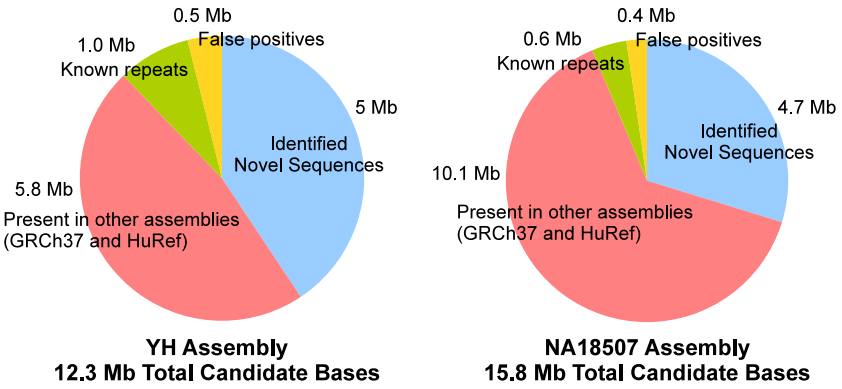

Figure 8: The figure illustrates the total candidate novel sequence bases, as a result of NSIT and BLASTn, from both input *de novo* assemblies. Nearly all of the previously reported novel sequence bases were identified. A large portion of the candidate bases was found present in other existing human genome assemblies, i.e., GRCh37 and HuRef. After excluding these portions and removing the known repeats, around 0.4–0.5 Mb of false positive novel sequence bases remained, resulting in nearly 100% sensitivity and 92% and 93% precision for the YH and NA18507 assemblies respectively.

59.8 Mb were identified at $skipSize = 60$. Aligning via BLASTn reduced the number of candidate bases to 15.8 Mb. 4,730,719 bp out of the reported 4,798,833 bp of novel sequences were present, yielding 99% sensitivity. The other 10.1 Mb of the candidates were found in GRCh37 and HuRef, and about 0.6 Mb were known repeats. The precision for the NA18507 assembly was 93%.

3.5 Comparison with BLAT and LASTZ

In identifying novel sequences from the *de novo* YH and NA18507 assemblies, [14] first used BLAT with -fastmap option enabled to assign candidate chromosomal locations to all contigs and scaffolds. LASTZ, which is the state-of-the-art pairwise aligner for large genomes, was subsequently used to align the *de novo* sequences to their candidate locations in the NCBI Build 36 reference assembly. The sequences unmapped by LASTZ were considered candidate novel sequences. They were aligned against the reference genome again using BLASTn, for more fine-scale alignments. The BLASTn parameters were set to allow only alignments with identity $> 90\%$ and E-value $<$ 1e-20. Unaligned *de novo* sequences that were >100 bp were reported as novel sequences.

BLAT and LASTZ are both well-established fast aligners for large sequences. However, they were not specifically designed for the task of detecting novel sequences in a *de novo* large genome assembly; to achieve the same results, these algorithms perform excessive unnecessary steps and inevitably are an overkill for the task. With the identical computing environment used for NSIT experiments, we attempted to run BLAT with -fastmap option for the YH assembly against the reference human chromosome 1. The

process continued for over 16 hours and did not finish. We later learned[1] that BLAT and LASTZ required about 25 and 300 CPU days in total, respectively, for the experiments in [14] on comparable machines (the LASTZ processes were parallelized).

4 Discussion and Conclusion

Technological advances over the last decade have finally led us into an era of personal genomics, where DNA sequence data can be generated at an ultra high speed, and unprecedented low cost. In addition to helping us understand our genome variation better, there is a great hope that these technologies would positively thrust personalized healthcare much ahead by making individual-based treatments more realizable. Sequence comparison of the human genomes provides a means to detect sequence variation constituting unique genotypes. In addition to SNPs, CNVs, insertions, and deletions, novel sequences have recently been classified as a new type of genomic sequence variant detectable in *de novo* genome assemblies. As much as 3–5 Mb of novel sequences per person have been found in several personalized genomes.

Fast and accurate identification of novel sequences is therefore an important task. Certain characteristics of *de novo* whole human genome assembly data have made the problem computationally daunting. None of the currently existing fast large genome aligners cater exactly to this problem; they require much long run time. At the rate that personal genomes are emerging, it is absolutely essential to be able to perform basic genome data analysis, like detecting novel sequences, quickly and inexpensively.

NSIT was designed specifically for this task. By applying appropriate domain knowledge, the algorithm is able to detect novel sequences in a *de novo* human genome assembly with great efficiency and accuracy. In our experiments, NSIT achieved nearly 100% sensitivity and very high precision, and finished in 2–3 hours on a typical desktop computer. Thus, NSIT is a practical and useful tool for next generation sequencing data analysis. In addition, although our focus has been on *de novo* human genome assembly, we do not foresee any restriction that would prevent NSIT from being successfully applied to *de novo* assembly from other species.

References

[1] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[2] David R. Bentley, Shankar Balasubramanian, Harold P. Swerdlow, Geoffrey P. Smith, John Milton, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, Nov 2008.

[1] Personal Communication

[3] Ken Chen, John W. Wallis, Michael D. McLellan, David E. Larson, Joelle M. Kalicki, et al. Breakdancer: an algorithm for high-resolution mapping of genomic structur al variation. *Nat Meth*, 6(9):677–681, Sep 2009.

[4] A. L. Delcher, S. Kasif, R.D. Fleischmann, J. Peterson, O. White, and S.L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–2376, 1999.

[5] Arthur L. Delcher, Adam Phillippy, Jane Carlton, and Steven L. Salzberg. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research*, 30(11):2478–2483, 2002.

[6] Akihiro Fujimoto, Hidewaki Nakagawa, Naoya Hosono, Kaoru Nakano, Tetsuo Abe, et al. Whole-genome sequencing and comprehensive variant analysis of a japanese individual using massively parallel sequencing. *Nat Genet*, 42(11):931–936, Nov 2010.

[7] Jane Grimwood, Laurie A. Gordon, Anne Olsen, Astrid Terry, Jeremy Schmutz, et al. The DNA sequence and biology of human chromosome 19. *Nature*, 428(6982):529–535, 2004.

[8] Iman Hajirasouliha, Fereydoun Hormozdiari, Can Alkan, Jeffrey M. Kidd, Inanc Birol, Evan E. Eichler, and S. Cenk Sahinalp. Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, 26(10):1277–1283, 2010.

[9] Fereydoun Hormozdiari, Can Alkan, Evan E. Eichler, and S. Cenk Sahinalp. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Research*, 19(7):1270–1278, 2009.

[10] Fereydoun Hormozdiari, Iman Hajirasouliha, Phuong Dao, Faraz Hach, Deniz Yorukoglu, Can Alkan, Evan E. Eichler, and S. Cenk Sahinalp. Next-generation variationhunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics*, 26(12):i350–i357, 2010.

[11] W. James Kent. BLAT — The BLAST– like alignment tool. *Genome Research*, 12(4):656–664, 2002.

[12] Jan Korbel, Alexej Abyzov, Xinmeng Mu, Nicholas Carriero, Philip Cayting, Zhengdong Zhang, Michael Snyder, and Mark Gerstein. Pemer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology*, 10(2):R23, 2009.

[13] Seunghak Lee, Fereydoun Hormozdiari, Can Alkan, and Michael Brudno. Modil: detecting small indels from clone-end sequencing with mixtures of distributions. *Nat Meth*, 6(7):473–474, Jul 2009.

[14] Ruiqiang Li, Yingrui Li, Hancheng Zheng, Ruibang Luo, Hongmei Zhu, et al. Building the sequence map of the human pan-genome. *Nat Biotech*, 28(1):57–63, Jan 2010.

[15] Ruiqiang Li, Hongmei Zhu, Jue Ruan, Wubin Qian, Xiaodong Fang, Zhongbin Shi, Yingrui Li, Shengting Li, Gao Shan, Karsten Kristiansen, Songgang Li, Huanming Yang, Jian Wang, and Jun Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 2010.

[16] Aaron R. Quinlan, Royden A. Clark, Svetlana Sokolova, Mitchell L. Leibowitz, Yujun Zhang, et al. Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Research*, 20(5):623–635, 2010.

[17] Mark T. Ross, Darren V. Grafham, Alison J. Coffey, Steven Scherer, Kirsten McLay, et al. The DNA sequence of the human X chromosome. *Nature*, 434(7031):325–337, 2005.

[18] Scott Schwartz, W. James Kent, Arian Smit, Zheng Zhang, Robert Baertsch, Ross C. Hardison, David Haussler, and Webb Miller. Human–mouse alignments with BLASTZ. *Genome Research*, 13(1):103–107, 2003.

[19] Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, and nan Birol. Abyss: A parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.

[20] Suzanne Sindi, Elena Helman, Ali Bashir, and Benjamin J. Raphael. A geometric approach for classification and comparison of structural variants. *Bioinformatics*, 25(12):i222–i230, 2009.

[21] Charles A. Tilford, Tomoko Kuroda-Kawaguchi, Helen Skaletsky, Steve Rozen, Laura G. Brown, et al. A physical map of the human Y chromosome. *Nature*, 409(6822):943–945, 2001.

[22] Kai Ye, Marcel H. Schulz, Quan Long, Rolf Apweiler, and Zemin Ning. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25(21):2865–2871, 2009.

[23] Daniel R. Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829, 2008.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ (กำลังอยู่ในขั้นตอน submission)

- **Pupacdi B**, Javed A, Zaki M J, Ruchirawat M., NSIT: Novel Sequence Identification Tool, *In Submission*

2. Poster Presentations

- **Pupacdi B**, Javed A, Zaki M J, NSIT: Novel Sequence Identification Tool (abstract/program #1743/W) Presented at **the 60th Annual Meeting of The American Society of Human Genetics**, November 3, 2010, Washington DC.

3. Invited Talks

- A fast and accurate approach to detect novel sequences in a *de novo* human genome assembly, **Structure Discovery in Biology: Motifs, Networks & Phylogenies Seminar**, SchlossDagstuhl, Saarbrucken, Germany, June 2010
- A software for scalable and efficient comparison of multiple human genome assemblies, **Harvard School of Public Health**, Harvard University, Boston, USA, October 2009