HHS Public Access

Author manuscript

IEEE Int Conf Bioinform Biomed Workshops. Author manuscript; available in PMC 2016 August 15.

Published in final edited form as:

IEEE Int Conf Bioinform Biomed Workshops. 2010 December ; 2010: 579–584. doi:10.1109/BIBMW.

2010.5703865

A distributed system for fast alignment of next-generation sequencing data

Jaydeep K. Srimani, Po-Yen Wu, John H. Phan, and May D. Wang

Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, 30332, USA

Abstract

We developed a scalable distributed computing system using the Berkeley Open Interface for Network Computing (BOINC) to align next-generation sequencing (NGS) data quickly and accurately. NGS technology is emerging as a promising platform for gene expression analysis due to its high sensitivity compared to traditional genomic microarray technology. However, despite the benefits, NGS datasets can be prohibitively large, requiring significant computing resources to obtain sequence alignment results. Moreover, as the data and alignment algorithms become more prevalent, it will become necessary to examine the effect of the multitude of alignment parameters on various NGS systems. We validate the distributed software system by (1) computing simple timing results to show the speed-up gained by using multiple computers, (2) optimizing alignment parameters using simulated NGS data, and (3) computing NGS expression levels for a single biological sample using optimal parameters and comparing these expression levels to that of a microarray sample. Results indicate that the distributed alignment system achieves approximately a linear speed-up and correctly distributes sequence data to and gathers alignment results from multiple compute clients.

Introduction

Known as “sequencing by synthesis”, “short read sequencing”, or “massively parallel sequencing”, Next-Generation Sequencing (NGS) refers to high-throughput nucleic acid sequencing technology [1]. NGS is attractive because it has the potential to improve biomarker-based disease diagnostics and treatment with equivalent or better accuracy compared to traditional methods (e.g., microarrays) and with potentially lower cost. Currently, researchers use different NGS systems, e.g., Illumina, Helicos, Roche, and Applied Biosystems (ABI), to produce data unique in certain basic biological properties such as transcript length and nature of sequence errors (e.g., mismatches or insertions/deletions). In addition, numerous algorithms and software applications have recently emerged for conducting high-throughput sequence alignment and assembly [2]. However, the full potential of NGS is dependent on the outcome of several key research directions including (1) improvement of computational efficiency, (2) analysis of the effects of different NGS platforms on alignment and assembly algorithms, and (3) direct quantitative comparison of NGS technology to more established technologies such as microarrays. In this study, we focus primarily on improving computational efficiency of alignment using distributed computing. We validate our distributed system in two ways. First, we use it to

optimize several alignment algorithms for simulated NGS data. Second, we apply it to align real NGS data and compare RNA-Seq gene expression to microarray gene expression.

Analyzing alignment output of NGS data from several NGS platforms (Helicos, Illumina, Roche, etc.) using different alignment software tools can help us gain more insight into alignment algorithm parameters. This is a key step before analyzing real patient sequence data. A distributed alignment system plays an important role in increasing the efficiency of NGS analysis. Each NGS platform produces data with different properties, including read length and error profile. For example, reads generated from Illumina systems tend to be fixed and short in length [3][4]. On the other hand, sequence reads from Roche data are longer [5][6]. Longer sequence reads generally increase alignment time due to the use of hash tables as opposed to seeded template positions [7]. Furthermore, longer sequence reads can negatively affect the alignment quality due to increased likelihood of natural variation, nucleotide polymorphism and sequencing experiment errors. However, longer sequence reads are more likely to align uniquely rather than ambiguously (i.e., alignment to multiple locations). Thus, each platform has its advantages and disadvantages as well as specific behavior with regard to alignment algorithms—i.e., the accuracy of alignment algorithms is affected by the data platform [8].

In this work, we use four popular alignment software tools: BWA [9][10], Novoalign (<http://www.novocraft.com/>), Mosaik (<http://bioinformatics.bc.edu/marthlab/Mosaik>), and SHRiMP [11]. These aligners index the template and then find the optimal locations for each sequence read based on measures such as alignment scores. The computational work for each read is comparable, so we can reduce the computation time by splitting the sequence reads file and aligning each file separately and independently. Each of these alignment software tools can be tuned using different sets of parameters, including nucleotide mismatch penalties, gap open and extension penalties, and hash-based parameters. Exploring the parameter space of each alignment software tool requires intense processing, which is often prohibitive given the complexity of the alignment algorithms. For example, for Novoalign, we have 3 parameters with 10 possible choices. Under this scenario, we need to run 10^3 trials to obtain statistics. If it takes 10 minutes for each run, then it would take several days to complete this simple analysis. Moreover, biomarker identification, which is a chief goal of NGS data analysis, also requires aligning a significant number of NGS reads to large genomic templates. Some datasets require days to process. Thus, computational time is a major constraint for NGS analysis to be of any realistic use in clinical applications.

To alleviate such computational issues, we propose using the Berkeley Open Interface for Network Computing (BOINC) platform to implement a distributed computing system for sequence alignment. BOINC is a volunteer computing scheme in which computationally intense, but modular, tasks can be split amongst any number of client machines, each of which is assigned subtasks according to project need and client resources [12]. We use this platform to divide large sequence reads datasets into smaller units, thereby running alignments in parallel and shortening the time required to finish a given job.

NGS is a recent development in gene sequencing. Thus, our understanding of the computational aspects of NGS analysis still needs to be improved compared to the more

established microarrays. To further understand the potential of this technique, we need a useful tool to more efficiently align NGS data. Our implementation of a distributed system for sequence alignment is advantageous for processing NGS data because it allows us to more effectively explore optimal parameter combinations for sequence alignment software and provides a platform to analyze large datasets efficiently.

Methodology

A. Overview

The distributed alignment application is divided into three tiers: the web interface, the BOINC server, and the client computers (Figure 1). The user selects a sequence reads file, reference file (containing the sequence to which reads will be aligned), aligner, and specific aligner parameters via a simple web interface. These requests are stored in a SQL database for subsequent analysis, tracking, and archival. Each request is then queued into the BOINC system (described more fully in section IIC). The BOINC system splits the original sequence reads file into smaller files of a user-specified size before sending each work unit to a client CPU for processing. Result files are then returned and stored for further processing and archival.

B. NGS Database

Each time a user enters a job via the online interface, the sequence reads file, reference file, aligner, and aligner parameters are stored in a SQL database (“alignments” table, Figure 2). The job record is linked to the specified input files via references to the applicable tables. Additionally, the aligner parameters are written to a job-specific file. Each job has a status bit, initialized to zero, signifying that it has not entered the BOINC processing queue. Each job's status changes first when the job is queued by BOINC, and then when the resultant files are complete. In this way, the user may track job progress.

C. Berkeley Open Interface for Network Computing

We use the Berkeley Open Interface for Network Computing (BOINC) to develop a distributed computing application that speeds the alignment process. As previously discussed, user-specified input files, alignment algorithm, and applicable parameter values are stored in a central SQL database. The BOINC application consists of a separate database, a server and distribution framework, and programs to facilitate data transfer and result validation. The general framework is shown in Figure 3.

The BOINC application has four major components: a work generator, a client program, a validator, and an assimilator. The work generator continuously queries the NGS database for any alignments that have not been processed. When the incoming queue is occupied, the work generator copies the specified parameters from the NGS database to a common file and splits the specified reads file into smaller subfiles, each of a user-specified size. Each sequence reads subfile, paired with its associated parameter file, constitutes a work unit, and is submitted to the BOINC server for distribution to client computers.

The BOINC framework distributes work units to available machines that are “attached” to the BOINC project. Once the sequence reads subfile and parameter file are downloaded by the client, the client program parses the parameter file and runs the specified aligner via a series of system calls. After the alignment is complete, the client uploads results to the BOINC server. Subsequently, the validator program confirms that the results are in the correct format and the assimilator gathers the results from all subfiles associated with a particular alignment job and concatenates them for further processing, analysis, and archival.

Once alignment is finished, post-alignment processing consists of parsing the output file and tabulating the number of ambiguous and unambiguous reads. We define an ambiguous read to be a read that aligns to multiple locations in the reference genome. Analogously, an unambiguous read is one that aligns to only a single location in the reference genome.

D. System Validation and Case Study

The distributed alignment system saves a tremendous amount of time when processing large NGS datasets. Here, we present a case study to illustrate the utility of the distributed application and to verify the correctness of the alignment results. The case study workflow is shown in Figure 4.

When aligning NGS sequence reads, it is important to tune the algorithm parameters to the data platform. We examine four well-known NGS data systems: Illumina, Helicos, Roche, and ABI. We use simulated read sequences from these four NGS platforms—generated with specific sequence lengths and error profiles—and tune the parameters for each of the four alignment algorithms: BWA, Novoalign, Mosaik, and SHRiMP.

Once the optimal parameters are obtained, we use these parameters to align real biological sequence reads using the BOINC distributed application. Finally, we verify the correctness of the alignment by correlating NGS read counts (also called digital gene expression) to corresponding microarray expression values.

1) Optimization of Alignment Parameters—The general procedure of NGS sequence alignment is as follows. The algorithm first indexes the reference genome template so that sequence reads can be quickly aligned. Second, the aligner constructs a hash table based on sequence reads to speed up the alignment process. Finally, the aligner calculates the alignment score for each possible alignment position of each read. The aligner can report one or more hits for each read based on user settings.

The alignment parameters mainly control the hash table construction and calculation of alignment scores. To find the best parameter sets suitable for a certain NGS platform, we need to determine which parameters are influential to output. Typically, these parameters include match/mismatch penalties, gap open and gap extension penalties, alignment score threshold, number of mismatches allowed, hash table size, and hash key selection, among others.

In this paper, we design experiments to find the best parameter subset for each sequencing platform and aligner. We gather statistics for uniquely and correctly mapped reads, uniquely

but incorrectly mapped reads, reads with multiple hits, and unmapped reads. Based on these statistics, we construct a model describing the parameter relationships, and then apply them to real biological data.

2) Correlation of NGS and Microarray Data—After identifying optimal alignment parameters, we verify these parameters by computing the digital gene expression of an NGS dataset and correlating to microarray gene expression. We use a pair of publicly available NGS and microarray data from the GEO database (<http://www.ncbi.nlm.nih.gov/geo/>, GSE22004) [13]. We use the refseq homo sapiens genome from the National Center for Biotechnology Information (NCBI, <http://www.ncbi.nlm.nih.gov/>) as the reference sequence [14]. The microarray data was assayed using Affymetrix Human Genome U133 Plus 2.0 Array from a similar biological sample. We determine the number of uniquely mapped reads for each gene in the template from NGS alignment outputs. We also extract the expression levels of each gene for the corresponding microarray data. To find a one-to-one mapping between microarray probes and genes in the template, we create a fasta file based on microarray probe sequences and determined the unique hits by aligning to the refseq template. We also filter out genes in refseq that align to more than one probe. There are only 1.34% (731/54675) of probes in the microarray and 22.66% (731/3226) of genes in the template that satisfy our filtering criteria. We correlate expression values between NGS and microarray platforms for these 731 genes.

Results and Discussion

A. Distributed System Performance

We present results for alignment speed-up and data correlation using the Novoalign algorithm. The other aligners behave similarly with an approximately linear reduction in computation time (not shown). For this test case, we use a distributed system with only four compute clients.

Table I lists processing times for various runs of Novoalign. Here we use different parameter sets to demonstrate that speed-up is generally not affected by aligner parameters. Distribution of the Novoalign jobs resulted in an average speed-up factor of 3.550. A perfect linear speed-up 4.0 (distributing to four client machines) does not occur because of some overhead in data processing and distribution. We observe that for these results, the parameter combination does not have an appreciable effect on the speed-up factor. This result is not surprising and serves as a simple verification of the distributed system.

B. System Validation and Case Study

1) Optimization of Alignment Parameters—Each aligner contains its own specialized parameters, but the general procedure to tune these parameters is the same. The procedure is as follows. First, we determine parameters that most affect alignment accuracy. We isolate these parameters and find optimal values. Second, we optimize the remaining parameters jointly. This procedure, however, is not guaranteed to result in the same alignment accuracy for each aligner. We simply use this procedure to find optimal parameter sets for each

alignment tool. The final results still depend on the sequence data as well as alignment software.

Here, we focus on Novoalign to show how we select optimal parameters with simulated data, and then apply them to real NGS data. Using this case study, we can demonstrate the utility of the system and verify that the data distribution and collection is sound.

In Novoalign, three parameters significantly influence alignment accuracy. They are threshold (or highest alignment score acceptable for the best alignment), gap open, and gap extension penalties. In our experimental design, we use a relatively large step size for threshold compared to the step size of the other two parameters. We also assume threshold has little interaction with gap open and extension penalties. Therefore, we can vary threshold first to obtain the optimal value and then vary gap open and extension penalties together.

By counting the statistics of uniquely and correctly mapped reads, uniquely but incorrectly mapped reads, reads with multiple hits, and unmapped reads, we can conclude that the optimal threshold should be 120. We then vary gap open and extension penalties together and visualize the results using a heat map (Figure 5).

Generally, high gap extension penalty and low gap open penalty perform best. But this is true only under specific constraints. The gap extension penalty can be set to

$$-10\log_{10}(P_{\text{gap}2}/P_{\text{gap}1}) \quad (1)$$

where $P_{\text{gap}1}$ is the probability of a single base indel (insertion or deletion) and $P_{\text{gap}2}$ is the probability of a two-base indel mutation. The gap open penalty can be set to

$$-10\log_{10}(P_{\text{gap}}) - G_{\text{extend}} \quad (2)$$

where P_{gap} is the probability of an indel mutation and G_{extend} is the gap extension penalty (<http://www.novocraft.com/>).

The high accuracy of parameters centered at the diagonal suggests that the aligner either prefers higher gap extension penalty combined with lower gap open penalty, or prefers lower gap extension penalty combined with higher gap open penalty. The overall optimum occurs at the combination of the first case—high gap extension penalty and low gap open penalty.

2) Correlation of NGS and Microarray Data—We have a relatively thorough understanding of microarray data and are generally confident in reported microarray gene expression values. Thus, we expect NGS gene expression values to roughly correlate with microarray gene expression [15][16]. Figure 6 and Figure 7 are scatter plots of NGS data versus microarray data in the log scale. Each data point corresponds to a specific gene or microarray probe. By performing this correlation, we also validate the BOINC distributed alignment software.

The positive correlation indicates that NGS expression values closely mirror microarray gene expression for these particular datasets. We found that increasing the threshold of the Novoalign software slightly increases correlation of NGS and microarray expression (from 0.716 to 0.750). This result was expected since increasing threshold leads to an increased number of unique hits in an alignment. This gives us more reliable results as the number of samples increases. Here we treat microarray data as a point of reference for comparison only because of the established use of microarrays in gene expression analysis. However, due to the known limitations of microarrays, other references may be used for comparison in the future.

Conclusion

We developed a distributed system to efficiently analyze large next-generation sequencing datasets. This system is based on the Berkeley Open Interface for Network Computing (BOINC), a highly scalable volunteer computing system. The software system includes a web interface that simplifies alignment algorithm parameter selection, job management, and result interpretation. We validate the system by optimizing alignment parameters using simulated NGS data. Subsequently, we use the optimal parameters to align NGS data generated from real biological samples. Positive correlation of the NGS data to microarray samples from similar biological samples suggests that the software is correctly distributing data to and collecting alignment results from multiple parallel client machines. In our work, the distributed system can reduce the computation time by an approximately linear factor determined by the number of available compute clients. Compared to other related work [17] [18] that focus on multiple sequence alignment and DNA sequence similarity search, we obtain similar improvement in terms of job time reduction. In addition, the scalability of the BOINC platform will allow us to maintain computational efficiency even as the size of NGS data continues to grow.

Acknowledgments

This research has been supported by grants from The Parker H. Petit Institute for Bioengineering and Bioscience (IBB), National Institutes of Health (NIH) (Bioengineering Research Partnership R01CA108468, Center for Cancer Nanotechnology Excellence U54CA119338), National Cancer Institute (NCI); Georgia Cancer Coalition (Distinguished Cancer Scholar Award to MW), Georgia Research Alliance; Hewlett-Packard and Microsoft Research. The authors would also like to thank C.F. Quo for valuable comments.

References

1. Metzker M. Sequencing technologies—the next generation. *Nature Reviews Genetics*. 2009; 11:31–46.
2. Mardis E. The impact of next-generation sequencing technology on genetics. *Trends in Genetics*. 2008; 24:133–141. [PubMed: 18262675]
3. Smith D. Rapid whole-genome mutational profiling using next-generation sequencing technologies. *Genome research*. 2008; 18:1638. [PubMed: 18775913]
4. Quail MA. A large genome center's improvements to the Illumina sequencing system. *Nat Methods*. Dec.2008 5:1005–10. [PubMed: 19034268]
5. Quince C. Accurate determination of microbial diversity from 454 pyrosequencing data. *nature methods*. 2009; 6:639–641. [PubMed: 19668203]
6. Margulies M. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*. Sep. 2005 437:376–80. [PubMed: 16056220]

7. Ewing B. Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. *Genome research*. 1998; 8:175. [PubMed: 9521921]
8. Harismendy O. Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol*. 2009; 10:R32. [PubMed: 19327155]
9. Li H, Durbin R. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*. 2010; 26:589. [PubMed: 20080505]
10. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*. Jul.2009 25:1754–60. [PubMed: 19451168]
11. Rumble S. SHRiMP: accurate mapping of short color-space reads. *PLoS Comput Biol*. 2009; 5:e1000386. [PubMed: 19461883]
12. Anderson D. BOINC: A system for public-resource computing and storage. 2004:4–10.
13. Guo H. Mammalian microRNAs predominantly act to decrease target mRNA levels. *Nature*. Aug. 2010 466:835–40. [PubMed: 20703300]
14. Pruitt KD. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes transcripts and proteins. *Nucleic Acids Res*. Jan.2007 35:D61–5. [PubMed: 17130148]
15. Bradford J. A comparison of massively parallel nucleotide sequencing with oligonucleotide microarrays for global transcription profiling. *BMC genomics*. 2010; 11:282. [PubMed: 20444259]
16. Git A. Systematic comparison of microarray profiling real-time PCR and next-generation sequencing technologies for measuring differential microRNA expression. *Rna*. 2010; 16:991. [PubMed: 20360395]
17. Lopes H. A distributed approach for a multiple sequence alignment algorithm using a parallel virtual machine. *IEMBS*. 2005:2843.
18. Nordin A. Parallel Guided Dynamic Programming Approach for DNA Sequence Similarity Search. *JCEE*. 2009; 1(4):402.

Figure 1. Distributed Sequence Alignment System Workflow

Figure 2. NGS Database

Figure 3. BOINC Framework

Figure 4. Case study workflow for optimization of alignment algorithm parameters and correlation analysis of gene expression

Figure 5. Heat map of the percentage of unique and correct hits. The default values for gap open and extension are 40 and 15 respectively. We vary the parameter values around default values with step sizes of 3 and 1, respectively

Figure 6.

Scatter plot comparing NGS reads counts and microarray expression levels. After data transformation on both datasets, we get a correlation coefficient of 0.71595, which suggests good correlation between NGS and microarray gene expression. Here we use Novoalign with a threshold of 80. (NGS Platform: Illumina Genome Analyzer II (Homo sapiens); Microarray Platform: Affymetrix Human Genome U133 Plus 2.0 Array)

Figure 7.

Scatter plot comparing NGS reads counts and microarray expression levels using Novoalign with a more “optimal” threshold of 120. In this case, the correlation coefficient is 0.7503, which suggests good correlation. (NGS Platform: Illumina Genome Analyzer II (Homo sapiens); Microarray Platform: Affymetrix Human Genome U133 Plus 2.0 Array)

Table I
Distributed Sequence Alignment Speed-Up

Novoalign			
Round	<i>Single Machine Operating Time</i>	<i>BOINC System Operating Time</i>	<i>Speed-up Factor</i>
1 [-oSAM-rA-R4-t SO]	1770.5 s	505.8 s	3.500
2 [-oSAM-rA-R4-t 120]	5941 s	1650 s	3.600