Optimizing Frequency based Seed Selection for DNA Read Mapping

Webpage: http://snnynhr.github.io/project

Project Description:

DNA sequencing has arisen as an integral tool in the study of biology and medicine. DNA sequencing allows researchers to determine the exact sequence of nucleotides (base-pairs), *Adenine, Guanine, Thymine*, and *Cytosine*, in a strand of DNA and subsequently determine the sequence of any structure from a small gene to an entire genome. These sequences have a myriad of applications in numerous fields: discovering evolutionary patterns among species, tracing human evolution, identifying genetic components to determine the causes of diseases and construct remedying medicine, and identifying primary characteristics of individuals in forensic studies. This has led to a huge demand for fast and efficient genome sequencing algorithms.

Platforms sequence DNA by splitting long strands into small chunks of contiguous base-pairs, typically less than 100 base-pairs, called *reads*. Reads are generated in the billions at a very high rate, and the objective is to reconstruct the original DNA sequence by aligning these short reads together. This is an unparalleled challenge on its own, since the computational complexity of aligning billions of reads is enormous. Sequencing the first human genome took more than a decade and over a billion dollars. Fortunately, there are assembled genomes available today, and given the high rate of similarity between genomes, one can be used as a basis for constructing another. Therefore, reads are mapped to a reference genome to construct the overall DNA sequence.

However, this still poses a very computationally challenging problem. The reference genome is extremely large as the size of the human genome is 3.2 billion base-pairs long. Reads are short and occur very frequently in the genome, so there are a vast number of possibilities where a single read could map to. In addition, DNA sequences differ among species and there can be variations in the DNA due to mutations. This is compounded by errors introduced in the sequencing process: insertions, deletions, and substitutions in the read. Therefore, the process of mapping reads to the reference genome is computationally hard since the mapping process needs to account for DNA variation and sequencing errors and allow a certain number of mismatches.

A popular method used to align reads is the seed-and-extend method. It uses substrings of the read, called *seeds*, to help find correct locations in the reference. The way these seeds are chosen determines how fast the mapper will be. If the seeds are occur very frequently in the genome, there are numerous locations which need to be checked to match where the read fits. Hence, it is optimal to choose a set of seeds which have the least overall frequency.

This problem itself it still computationally hard, so the goal of this project is to develop new heuristics and algorithms to efficiently find less frequent seeds. These new algorithms will be compared with existing seed selection schemes, such as Cheap K-mer selection from FastHASH and optimal Q-gram selection from Hobbes. Ultimately, these algorithms will be integrated into a mapper to improve its efficiency and performance.

The advisor for the project is Professor Onur Mutlu who leads the CMU SAFARI research group, and the point of contact is Hongyi Xin, a 5th year graduate student under him. Prof. Mutlu's group has worked in this area as they created FastHASH. This project will serve as an improvement to this mapper and the future of DNA sequencing.

Project Goals:

100% goal:

Implement the entire seed selection testing framework and implement existing seed selection algorithms. Design heuristics based on seed frequency and run preliminary tests to determine if they are promising. Select a set of heuristics to continue to add and optimize. Finish implementing heuristics and run tests against other seed selection algorithms to determine how effective the new algorithms are. Once this is complete, integrate seed selection algorithms into the overall mapper and test the effectiveness of the mapper. The purpose of the seed selection algorithms is to pick a set of seeds which minimizes the sum of the expected frequencies of the seeds. Therefore the heuristics will be measured by what factor they reduce the frequency sum of the seeds compared to existing algorithms. The mapper incorporating the new algorithms (FastHASH) will be tested against varying read sets and measured for speed, sensitivity, and comprehensive, standard metrics measuring mappers.

75% goal:

If no designed seed selection heuristics show improvement in the overall frequency sum of the seeds, then analyze the algorithms and discuss the potential reasons or flaws for why those approaches may not be appropriate. These flaws can be seen as reasons to show that an approach is most likely not one which will have a positive result, and should be avoided in the future.

125% goal:

In addition to everything laid out in the 100% goal target, if incorporation of the seed selection heuristics into FastHASH is successful, then other seed and extend algorithms can be looked at, like Hobbes and SHRiMP. The new heuristics can be added to these algorithms as well to determine if they provide performance improvements.

Milestones:

1st Technical Milestone for 15-300:

Read all relevant literature regarding seed selection in DNA sequencing and read papers on the mappers implementing seed selection (FastHASH, Hobbes, SHRiMP, RazerS). Write down a list of potential ideas for heuristics involving seed selection. Read through the existing FastHASH codebase and understand the genome representation and query format. Build the entire testing environment and infrastructure for the seed selection project. This involves interfacing with the FastHASH code to read and parse the genome, run the algorithms on the large DNA read sets, calculate metrics and results for each algorithm, and provide statistics and data regarding the comparison between the existing and developed algorithms.

Bi-weekly Milestones for 15-400:

January 25th: Implement the basic seed selection algorithms: uniform and threshold seed selection. Test the functionality of the seed selection test framework and get preliminary statistics.

February 8th: Implement the seed selection algorithms which will be used as a comparison. These include efficient Q-gram selection from Hobbes and Cheap K-mer selection from FastHASH. Test the algorithms for correctness and run the seed selection test framework for statistics.

February 22nd: Start hashing out the details of the heuristics. Implement preliminary versions of the heuristics and test them out. If they show promising results, continue to work on improving the heuristics. If not, use the test results to determine how to potentially improve the algorithms.

March 14th: Continue to work on the heuristics. Try out ideas from relevant papers read if previous tries are not successful. Build off the algorithms like Cheap K-mer selection. Test again on the read datasets and compare to existing algorithms.

March 28th: Pick out a set of the best heuristics to continue working on. Optimize the heuristics for speed by looking at complexity, cache-efficiency, and possible multithreading expansion. Finish the development of the heuristics.

April 11th: Run final tests comparing the heuristics with existing algorithms. Obtain a multitude of statistical data which can be used to determine if the heuristics perform at a statistically significant level.

April 25th: Integrate seed selection with the overall mapper code (FastHASH). Run basic tests to ensure correctness, and run DNA read tests to determine how effective the new seed selection algorithm with respect to the new mapper.

Literature Search:

The paper on FastHASH, by Xin et. al., which was published by the CMU SAFARI research group, describes improvements to seed-and-extend based mappers. It contains a wealth of information about DNA sequencing, background of the field, the major competing mappers, and access to test data. It will be used as the primary source of background relating to the overarching picture of DNA sequencing and read mapping. FastHASH describes a heuristic for selecting seeds based on frequency, Cheap K-mer selection. This will be used as one of the comparator algorithms in the experiments regarding the new algorithms which are developed. There is a paper about Hobbes, by Ahmadi et. al., which describes another seed-and-extend based algorithm. This paper also includes a frequency based heuristic, efficient Q-gram selection, which will also be used to compare the new algorithms. A recent paper by Xin et. al. on the Optimal Seeds Solver contains interesting insights on results regarding frequency of seeds which can be used to enhance the algorithm. In addition, other seed-and-extend based mappers, such as SHRiMP by Rumble et. al. or RazerS by Weese et. al., may provide useful ideas as inspiration.

Resources Needed:

To evaluate the performance of the new frequency based seed selector in comparison with other methods, the algorithm needs to be integrated into an existing mapper which is based on the seed and extend method. The CMU SAFARI research has already developed such a mapper,

FastHASH, so the code for the new seed selection algorithm will be integrated into FastHASH's seed selection. This will allow for results on mapper related metrics: speed, sensitivity, and comprehensiveness. Ultimately the goal of the project is to apply the new seed selection heuristic to seed-and-extend mappers in order to compare seed-and-extend mappers with BWT (Burros Wheeler transform) based mappers such as BWA, Bowtie, and SOAP2. Therefore, access to working version of BWA, Bowtie, SOAP2, and FastHASH will be needed. FastHASH is available through the CMU SAFARI GitHub releases. The other mappers have publicly available releases as well.

In order to test the new heuristics and algorithms, large datasets for reads and datasets for genomes are needed. Through the 1000 Human Genome Project, 1000 different human genomes are available freely, so this will be used to build prediction database for the algorithm. Large (actual) read datasets are available through FastHASH's test resources. FastHASH was tested on large read datasets, so that test data can be used again and easily accessed. In addition, a large database of simulated reads can be created, by randomly generating strings.

In order to build the prediction database, test the seed selector, and compare it with other algorithms, a large random access memory machine is needed. Loading one genome takes about 4GB of RAM, so including the algorithm's memory usage and the fact the parallel instances may be running at the same time, the memory usage increases to 750-1TB RAM. Fortunately, the CMU SAFARI group has a specialized 1TB RAM machine (32 cores) specifically for running DNA mapping experiments. This will be used for development and testing.