CS159-T: A History of Parallel Processing in Bioinformatics

David Warshawsky
Computer Science Department
San Jose State University
San Jose, CA 95192
408-924-1000

david.warshawsky@sjsu.edu

ABSTRACT

Parallel processing is revolutionizing the field of bioinformatics by enabling large-scale collection and analysis of biological data. In this paper, the author provides a comprehensive history of parallel processing in bioinformatics, from founding methods to modern high-performance computing. The author discusses the major challenges faced in the parallel processing of bioinformatics data, including data collection, preprocessing, and load balancing. Highlighted in the paper are some of the major tools that take advantage of parallel processing and their effects on sequence analysis. Overall, this paper provides an in-depth perspective on the role of parallel processing in advancing the field of bioinformatics.

INTRODUCTION

In order to understand the role that Parallel Processing plays in bioinformatics, the reader must first understand the way that the biological data is recorded and analyzed to get sequences.

Frederick Sanger is the pioneer of the modern genetics field. He created a method to sequence nucleotides unlike before due to its ability to be automated and precision chain-terminating nucleotides.

To split a double-helix DNA it must be denatured. This requires breaking the hydrogen bonds which hold together the two strands through their complementary nucleotides(A and T, or G and C) on DNA. Heat is a way to split the DNA strand bonds.

A primer, which is a short stretch of DNA that targets a unique sequence and helps identify a unique part of the genome, is added to DNA strands in order to identify the site for DNA replication and provide an attachment point. Heat is removed to allow it to bind to the template strand of DNA. DNA Polymerase makes copies of DNA strands by adding nucleotides to build the corresponding side of the DNA strand. The DNA Polymerase is added to the DNA with the fluorescently labeled dideoxynucleotides and basic nucleotides. Dideoxynucleotides stop the DNA polymerase from continuing to add nucleotides because they are missing a hydroxyl group. Hydroxyl groups are needed to allow the next nucleotide in the new strand to bind to the current nucleotide through a phosphodiester bond. Here the chain doesn't get any bigger.

This added precision over previous methods of sequencing which were more error prone. The process is repeated until many different chains of all sizes are made and so the fluorescently labeled dideoxynucleotides, ddNTPs, are read in order of termination and the whole sequence is finally found. [3]

Amino acids and nucleotides require unique distinction because they can't be distinguished by a microscope so other identification is required. The combination of different methods to sequence in parallel with high-throughput sequencing platforms is known as next-generation sequencing. They build on top of Sanger Sequencing often through the use of fluorescent ddNTPs. This has allowed for sequencing entire genomes, transcriptomes, viruses, and bacteria.

Due to the need for efficiency, the larger the sequence lengths are, the less accurate the sequencing results often are.[4]

Parallel Processing connecting to Hardware

Parallel processing is not only relevant to software in Bioinformatics. The collection of raw data is essential in order to produce precisely calculated nucleotide bases in output file sequences which can then be analyzed in software later. Understanding the collection process and the goal of the analysis informs the analysis methods later on. Some examples include automated Capillary Electrophoresis and Cycle Sequencing.

Automated Capillary Electrophoresis

Also known as electric migration. This is a Sanger Sequencing based technology to analyze the DNA fragments that result from the ddNTPs(fluorescent nucleotides ending the fragment). It is a streamlined design to be able to analyze many samples at once. The DNA fragments in each tube are in specially designed liquid.

Design

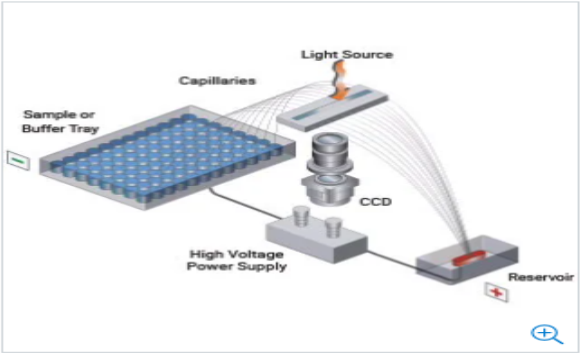

Figure 1. Femto Pulse 12-Capillary Array [1]

The terms in Figure 1's hardware design are described below.

These tubes called *capillaries* are put into every *buffer tray* sample which contains the preprocessed biological fragments.

The *light source* is a laser that stays on which activates the fluorescent dideoxynucleotides.

The *CCD* or charge-coupled device is a silicon chip that has light-sensitive pixels that are registered as an electrical charge.

The *High Voltage Power Supply* creates a charge in the direction going from the tray with the samples which is negatively charged, the anode, and the reservoir at the end which is positively charged(cathode). The smaller fragments go through first while the bigger ones are later [5].

Computation of Nucleotides

A CPU with a high clock rate controls the temperature and power supply to a constant rate needed for the proper flow of the fragments.

The electrical force generated pushes the DNA fragments through the capillaries and pulls them towards the reservoir. As the fragments pass through the capillaries, they eventually go through the CCD camera(charge-coupled device) which has light-sensitive pixels that are registered as an electrical charge. There is a laser that is constantly on which causes the fluorescent nucleotides to give off light that the CCD records.

The amount of time the fluorescent nucleotide takes to pass through the camera is calculated and together with the light signal and current, the nucleotide and the length of the fragment is calculated to use for the sequence identification.

This information is shown as peaks that can be analyzed to be put together to get the full sequence. Each capillary has its own laser and camera but can theoretically share a CPU depending on how many pixels are on each CCD camera. This would be SIMD architecture. In Flynn's Taxonomy, the analysis of the data to determine length of the fragment and nucleotide type would be MIMD. The addition of the time between the pixel intensity at the start and end of the CCD would be a SIMD calculation. This is parallelized macro-architecture being coordinated at scale. The storage and analysis of the data requires a powerful processor. Up to 6000 bases can be read per fragment based on this specific CCD size, processor, and analysis speed.

There is no analysis occurring other than measuring the speed of the DNA strand through the camera and the type of fluorescent nucleotide.

Cycle Sequencing

After a DNA fragment is discovered that a bioinformatician would like to sequence, they can amplify it in order to sequence just that fragment. In the past, biologists would have to use bacteria to amplify the fragment. Today, Polymerase Chain Reaction(PCR), which uses primers to bind(anneal) to the region of interest within the DNA fragments can be utilized.

Design

The PCR reaction is designed by using four separate tubes, each containing a different fluorescent dideoxynucleotide (ddNTP) and the four standard deoxynucleotides (dNTPs) When the fluorescent dideoxynucleotides are incorporated into the growing DNA strand, they terminate DNA synthesis at that point.

During the PCR cycle, the enzyme DNA polymerase adds nucleotides to the growing DNA strand until it reaches a ddNTP, which stops the addition of further nucleotides. The ddNTPs are incorporated randomly, resulting in a series of fragments of different lengths, each terminating at a different position.

Computation of Nucleotides

The resulting mixture of DNA fragments is then separated by size using capillary electrophoresis, with the fluorescent signals detected and recorded by a laser-based detection system. The signals are then analyzed to determine the sequence of the DNA fragment.

Cycle sequencing is highly accurate and builds on the methods of Capillary Electrophoresis. It is especially useful for sequencing small fragments of DNA due to higher accuracy with smaller fragments.[6]

Parallel Processing in Software

The larger the sequence size, the less accurate the sequence is because accuracy drops off as less and less nucleotides are available to incorporate into the new strand. Most sequencing methods contain ways of ascertaining the correct nucleotide call likelihood. This is known as quality. The quality of nucleotides can be filtered to remove them and replace them with placeholders. This can be done in parallel by splitting the sequence and running SIMD code on a macro-system or a microarchitecture parallelized pipeline.

Cheaper methods like High-Throughput Sequencing which generates lots of data for small sequences by cyclically reading nucleotides can be used to determine the ratios of nucleotides. They are less accurate due to their speed but if not trying to get exact results, it is a cost-effective and useful preliminary option.

GC Count and Nucleotide Ratio Calculation

There is a powerful ability to utilize parallelization through multiple cpus or cpu cores in Python. Python is the industry standard tool due to its high level of abstraction with packages and simplicity. It makes the workflow much simpler for a bioinformatician and the results are reproducible which is important. Scientific tools can be managed using PIP and Conda package and environment managers. Below the Bio package standards for BioPython.

Python Code Example

Users > David > Documents > fasta_nucleotides.py > ...

```python
from Bio import SeqIO
from Bio.SeqUtils import GC
from multiprocessing import Pool, cpu_count

def process_record(record):
    sequence = str(record.seq)
    gc_content = GC(sequence)
    a_count = sequence.count("A")
    c_count = sequence.count("C")
    g_count = sequence.count("G")
    t_count = sequence.count("T")
    total_count = a_count + c_count + g_count + t_count
    a_freq = a_count / total_count
    c_freq = c_count / total_count
    g_freq = g_count / total_count
    t_freq = t_count / total_count
    return (record.id, a_freq, c_freq, g_freq, t_freq, gc_content)

def main():
    filename = "my_sequences.fasta"
    num_processes = cpu_count() # Number of CPU cores
    with Pool(num_processes) as p:
        records = SeqIO.parse(filename, "fasta")
        results = p.map(process_record, records)
    for result in results:
        print(f"{result[0]}\tA: {result[1]:.2f}\tC: {result[2]:.2f}

if __name__ == "__main__":
    main()
```

Figure 2. Python Script fasta_nucleotides.py sample from CS 123A to analyze GC content and nucleotide ratios.[7]

The code in Figure 2 is used to analyze the nucleotide percentages which can be used to formulate the ratios of nucleotides in Polymerase Chain Reaction(PCR) for shorter subsections of the sequences. GC content of nucleotides are used as unique identifiers to ensure that there isn't any contamination of unwanted fragments from bacteria or viruses. If there is large variation in the different fragment nucleotide percentages, it could mean there has been contamination or incorrect binding of the primers due to heating issues with the analysis machine.

High Performance Computing

Assembling a genome for an organism that hasn't yet been sequenced is known as De Novo Assembly or from the beginning assembly. It is an incredibly computation heavy task that can take years to run in series but days to run in a HPC depending on the size of the sequence that has been split into sequenced fragments. The mapping algorithms must take the different fragment sequences and find overlapping sections and build decision trees from the resulting sequences after having already aligned the different sequences.

Parallelized Data Preprocessing

If sequences are heavily similar, they can be grouped which makes downstream analysis simpler so that way less alignment occurs between redundant data.

A tool in the free online Bioinformatics HPC sharing platform Galaxy called FASTQC can be used for identifying highly repeated sequence data. It is hosted online for people learning to do bioinformatics or without access to cloud computing otherwise.

The raw sequences can be run through FASTQC which makes use of parallelization in order to identify similarities. One important consideration is identifying the primers and adapter sequences which are present in large amounts during polymerase chain reaction(PCR) amplification. [2]

Methods of Parallelization

Parallelization can either be instruction-level or data level.

Bioinformatics data can be split up into smaller pieces in order to determine what the original sequence is. If the sequences are too large, the accuracy is low therefore enzymes often are used to split up the sequence randomly into different sized fragments.

Dynamic Programming methods are implemented in order to keep track of which parts have already been compared and what the results are. This saves time for identifying the repetitive sequences where each sequence can be split up to do comparison on a cpu or a cpu core. The memory can be shared but write access is restricted to one thread.

The comparison returns values to a table which say if the split sequence parts are equal(say three nucleotides at a time). This can later be used for another analysis. Hash codes could be implemented for a sequence.

Job scheduling management or algorithms make use of mutex conditions in order to prevent jobs from running at the same time on the same resources. Given the importance of accurate results, the resources are allocated to a specific job in a macro architecture instead of having inter-instruction parallelization in a pipeline. The job scheduling for Galaxy is first in first out with some specific rules to prevent deadlock so that way all jobs get a chance to run. [4]

SUMMARY/CONCLUSION

Bioinformatics is a field that requires large signal data collection that requires analysis in order to sequence a genome or produce all sorts of biological materials or sequences. Proper analysis is critical and produces large amounts of raw sequence data. The physical signals require macro-architecture parallelized hardware for proper identification to take place. Unique identifiers and ratios allow for further identification of sequences and proper nucleotide ratios in the lab to catalyze proper nucleotide binding to prevent an under or overabundance of nucleotides. Repetitive data is harmful to analysis efficiency and must be properly identified before alignment of sequences to find related organisms. The macro-architecture of HPCs require resources to be dedicated and rededicated for proper accuracy and preventing race conditions and deadlock which can wreak havoc on currently executing computationally expensive tasks.

REFERENCES

Agilent Technologies. Femto Pulse 12-Capillary Array [Online image].https://www.agilent.com/cs/publishingimages/12-capillary-array-zoomtb-320x320-agilent.jpg.[Accessed: Apr. 23, 2023].

Galaxy Team. 2018. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Commun. ACM 61, 8 (August 2018), 48–54. DOI:https://doi.org/10.1145/3233027

Gauthier, J., Vincent, A.T., Charette, S.J., & Derome, N. (2018). A brief history of bioinformatics. Briefings in Bioinformatics, 21(6), 1-9. doi: 10.1093/bib/bby063. PMID: 30084940.

Lee, W. "BIOL CS 123B: Bioinformatics II" [website]. San Jose State University, accessed April 22, 2023. Available at: https://sites.google.com/sjsu.edu/biolcs123b-sjsu/home?authuser=1.

Promega Corporation. 2019. What Is Capillary Electrophoresis? [Video].[Video].Retrieved from https://www.youtube.com/watch?v=x7PUqNA0eOA&ab_channel=PromegaCorporation

Thermo Fisher Scientific. (n.d.). Sequencing Reaction for Sanger Sequencing. Retrieved April 22, 2023, from https://www.thermofisher.com/us/en/home/life-science/sequencing/sanger-sequencing/sanger-dna-sequencing/sequencing-reaction-sanger-sequencing.html

Wesley, L. CS 123A Bioinformatics I. San José State University. Fall 2022. Based on code from the author's previous project.