International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34670-34674

Performance assessment for main stages in genomic and transcriptomic data processing based upon reads from illumina sequencing technologies

Nelson E. Vera-Parra[1, 2,3], José N. Perez-Castillo[1, 2,5], Cristian A. Rojas-Quintero[1, 2,4]
[1] Universidad Distrital Francisco José de Caldas
Carrera 8 # 40-62
Bogotá, Cundinamarca
Colombia
3239300
[2] GICOGE Research Group
Carrera 8 # 40-62
Bogotá, Cundinamarca
Colombia
3239300
[3] neverap@udistrital.edu.co
[4] carojasq@correo.udistrital.edu.co
[5] nelsonp@udistrital.edu.co

Abstract- This article documents the performance assessment of the main stages and sub stages for the genomic and transcriptomic bioinformatics data analysis with the purpose to create a frame of reference for researchers to get acquainted with the computational requirements of each process and identify bottlenecks representing the future computational challenges originating potential research.

The assessment was focused on four stages: preprocessing, assembly, annotation and mapping, with genomic and transcriptomic datasets of reads of 100 pairs from Illumina sequencing technologies (340,993,796 transcriptomic reads and 354,530,026 genomic reads), the assessment criteria was the RAM usage, the processor usage (number of cores used) and the execution time. The computer used was: 4 processors Intel Xeon E7450 (each one of these with 6 cores with clock speed of 2.4 GHz), available RAM of 256 GB and disk capacity of 1.5 TB. The results show the following:

- The stage with the least computational requirement is quality control and the one with the most demand is the annotation stage. The stage that requires more memory is the normalization stage.
- The annotation and mapping stages are totally parallelized and make the maximum use of the available cores. – The graph build sub-stage in genomic assembly is the process less parallelized and represents a bottleneck.

Keywords- Bioinformatics, Next Generation Sequencing, Assembly, Mapping, Annotation, Preprocessing.

1. Introduction

With the arrival of the next generation sequencing several tools has been created for each stage in genomes and transcriptomes analysis [2], [3]. Most of the time these tools require high performance computational equipment to be executed due to the complexity of the algorithms involved and it's data representation on memory. Some of the stages being executed frequently on genomic or transcriptomic data are:

Quality controls and reads filtering: This stage of the process is also known as preprocessing. In this stage the artifacts are being removed from the datasets before being assembled with the purpose of improving quality of reads which improves precision as well [4]. In this step the normalization process is executed too, this is about the coverage reduction from the genome or transcriptome parts which has been over sequenced leading to a better computational performance of the assembly stage [3] [5].

Assembly: Assembly is the genomic sequence rebuilding process for large genomic sequences from random derived subsequences. This process involves finding the similarity of sequences overlapping one above others to find a relation between the content of the sub sequences; the result of this process is called "Assembly" which is a data structure that groups reads in contigs and contigs in metacontigs [3].

Annotation: Annotation is the process to add biological information to the sequences, specially the gene identification and the discovery of their functions [6].

Mapping: In this stage the reads get aligned to the assembled genome or transcriptome to find the positions on the reads. However, in this stage no relevant information to the researcher is produced, it is one of the most important because it gives information for other analysis, as an example before the quantifications process, differential expression and abundance estimation the reads should be mapped against the assembled transcriptome [7].

Each step for this analysis is composed at the same time of several sub stages which represents all the various computational requirements. Some of these stages or sub stages has been parallelized to make the better use of the potential high performance computational equipment.

In this article is analyzed each stage that make up these analysis collecting data referred to the memory and CPU usage with the purpose of figuring out which of the stages and sub-stages with greater computational requirements and give a point of reference to let other researchers know the information about computational capacity required to analyze their data and identify bottlenecks creating a starting point for future research aiming to optimize such processes.

2. Methodology

Methodology used in this assessment involved the following steps: - Selection of stages to evaluate based upon bibliographic references and the collected experience since 2012 in the transcriptomic and genomic data provided in the agreement between the Genetic Institute from Universidad Nacional de Colombia and GICOGE research group from Universidad Distrital of Bogotá. -Datasets selection considering the correspondence to a representative sample by size and quantity of reads. -Tools selection to execute on each genomic and transcriptomic stage based upon their reference number and the applicability to the available datasets. -Software selection for collection and performance data analysis. -Assessment development following a typical workflow for the dataset processing. -Analysis for obtained results and discussion.

Computational Equipment: A computer provided by the CECAD (Universidad Distrital High Performance Computer Center) with 4 Intel Xeon E7450 processors each one with 2.4 GHz clock speed. The amount of RAM memory available was 256 GB. This computer has 1.5 of disk space.

Datasets: The following datasets were chosen from the table 1 corresponding to a representative sample in terms of size and sequencing technology of the data typically analyzed.

Table 1. Data sets used for the assessment. Source: Authors

	Transcriptome	Genome
Number of reads.	**340993796**	**354530026**
Sequencing Technology	**Illumina [8]**	**Illumina**
Organism	**Hydractinia symbiolongicarpus**	**Hydractinia symbiolongicarpus**
Pair of bases	**100**	**100**

Tools for collecting and analyzing data: Data collecting regarding to the performance a GNU/Linux tool called *Collectl* [9] was used which allows to get in detail data related to the machine performance such as RAM memory usage, CPU, I/O and network traffic. In this case the tool only collects data regarding to RAM and CPU because it is the most important data for this assessment. R is used to filter data collected and it's later plotting using the *ggplot2* [10] library.

Workflow: The workflow used for the genomic/ transcriptomic data analysis is shown in the figure 1.

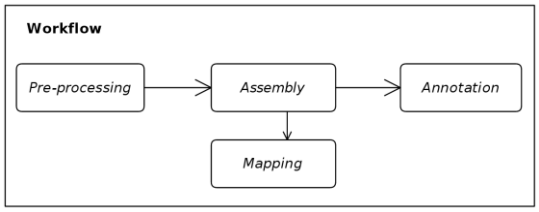

Fig 1. Used workflow for dataset processing. Source: Authors.

Evaluated Software: For both genomic and transcriptomic data the same software was used for many of the stages. The only stage that differs in tools is the assembly stage because this stage uses different algorithms. Below are commonly used software list.

Quality Analysis: FastQC [11], aims to provide and easy way to do some quality control checks on raw sequences from high performance sequencing pipelines. FastQC provides a set of modular analysis that can be used to give a quick view of the data that with any problem that must be taken into account for a deeper analysis. The files supplied as an input were 6 files from reads in FASTQ format.

Normalization: insilico_read_normalization.pl is a script integrated into Trinity's utilities [4] based on Diginorm [5] with the difference that this one allows the parallelization of some process points. Parameters used for genomes an transcriptomes were: Maximum coverage 30, 24 threads, 200GB of RAM for Jellyfish.

Annotation: BLAST (Basic Local Alignment Search Tool) [12] is responsible for finding regions with local similarity between sequences. Specifically *blastx* was used, it allows comparing nucleotides against amino acids. The database used for this analysis was Uniprot [13]. Parameters used were: *evalue* 1e-5, maximum 5 reported sequences and 24 threads.

Mapping: BWA [14] is a software package for mapping sequences of little divergence against a reference genome as the human genome. 24 threads were used for the mapping.

Genome Assembly: ABySS [15] is a de novo parallel assembler for paired sequences, specifically designed for short reads. Parameters used were: 55 for k-mer size, 10 minimum reads to build a contig, 24 threads for processing.

Transcriptome: Assembly: Trinity [4] is a new method for efficient robust transcriptome rebuild. This software combines three independent modules: Inchworm, Chrysalis and Butterfly applied sequentially to process large amount of RNA-Seq reads. Broadly the process works as described: Inchworm: Assemblies the RNA-Seq data into unique sequences transcripts, creating transcripts for a dominant isoform. Chrysalis: Groups the Inchworm contigs and build the De Bruijn's graph for each group. Each group represents the transcriptomal complexity for a given gene. Butterfly: Process the graphs individually in parallel, tracing the reads and the reads pairs within the graph and reports the most probable transcripts. Parameters used were: 24 threads and 200GB of RAM Jellyfish [16].

3. Results and Discussion

Genomic data processing.

Figures 2 and 3 shows the analysis global results of genomic data for RAM and CPU use.

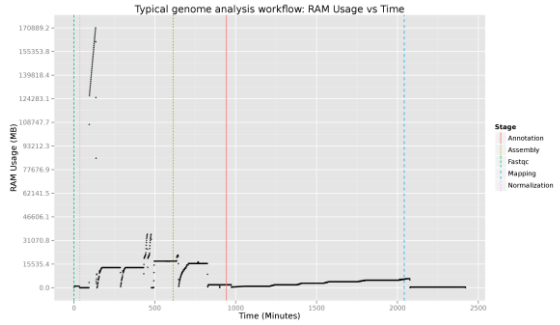

Fig 2. Results for memory use in genome processing stages. Source: Authors.

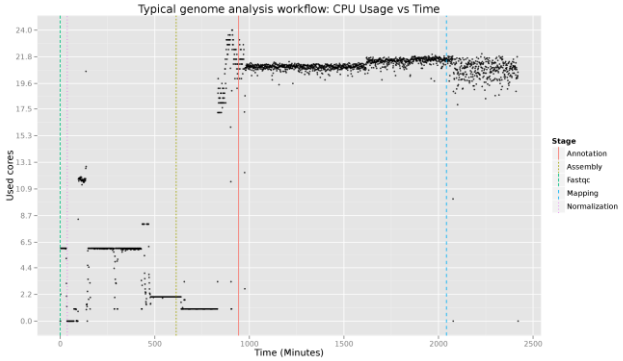

Fig 3. Results for CPU use in genomic processing stages. Authors.

The table 2 shows relevant information of this analysis. Source: Authors.

Data processing results clarifications: The quality analysis made with FASTQC reported that the reads had good quality therefore it was not necessary to remove adapters or elimination of low quality reads. Once the normalization process were 56,969,624 i.e. 16% of the original reads. 126,482 hits were found in the annotation stage.

Transcriptomic data processing: Figures 4 and 5 shows the global transcriptomic analysis in the use of RAM and CPU correspondingly.

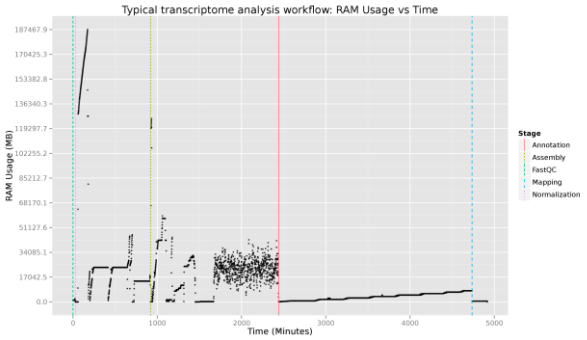

Fig 4. Results of RAM usage in transcriptome processing stages. Source: Authors.

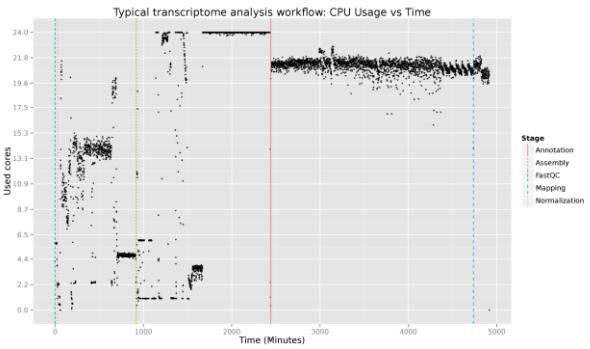

Fig 5. Results of CPU usage in in transcriptome processing stages. Source: Authors

Table 3 shows relevant information of the transcriptomic analysis.

Table 2. Summary of relevant results by genomic analysis stages. Source: Authors.

Stage	Sub stage	RAM		Number of Cores		Time
		Mean	Maximum	Mean	Maximum	(Minutes)
Quality Analysis		1030MB	2000MB	5	7	33
Normalization	Jellyfish	140GB	172GB	12	24	51
	Kmer Stats	21GB	32GB	6.7	8	327
	NCBK Normalization	15.5GB	17GB	2	2	172
Assembly	Read and reads load.	16GB	16GB	1.2	16	68
	Generation of adjacency	16GB	17GB	1	1	93
	"Popping bubbles"	2GB	2GB	1	1	20
	Vertices Union.	2GB	2GB	1	1	5
	Contigs union using paired-end information	2GB	2GB	20	24	327
Annotation	N/A	3.5GB	6GB	22	24	1113
Mapping	N/A	600MB	640MB	23	24	342

Table 3. Summary of relevant results for stages in transcriptome analysis. Source: Authors

Stage	Sub stage	RAM		Cores Used		Time (Minutes)
		Mean	Maximum	Mean	Maximum	
Quality Analysis	N/A	1000MB	1250MB	5	6	29
Normalization	Jellyfish	163GB	190GB	15	24	126
	Kmer Stats	25GB	52GB	15	16	504
	NCBK Normalization	16GB	20GB	5	6	210
Assembly	Inchworm	50GB	70GB	4	7	161
	Butterfly	33GB	45GB	18	24	94
	Chrysalisis	10GB	12GB	22	24	110
	Chrysalisis: Graph From FASTA	22GB	34.5GB	13	24	187
	Chrysalisis: Reads to transcripts	150MB	200MB	3	5	153
	Chrysalisis: Quantify Graph	27GB	50GB	23	24	592
Annotation	N/A	4.5GB	8GB	22	24	2400
Mapping	N/A	510MB	550MB	22	24	183

Data processing results clarifications: Like the case of genome the reads came with good quality, so it was not necessary to apply any process of removing of reads because of low quality. The normalization process left 64,852,526 i.e. 19% of the original ones. Generated contigs on the assembly stage were 572,794 with N50 of 1,326. A total of 697,273 hits were obtained by comparing against the Uniprot database.

4. Results Analysis

For both types of data (transcriptomic and genomic), the analysis stage is the one with the less time and resources requirements, because it does not have complex algorithms. Its parallelism depends upon the number of files provided, in this case 5 transcriptome reads and 6 with genome reads were provided.

Normalization is one of the stages with the most RAM requirement because Jellyfish builds a k-mer catalogue of the sequences and all of this catalogue is loaded into the RAM. The RAM usage should be kept on mind due to the reception of all over sequenced reads, if this stage were not performed all the computational cost related to RAM would be on the assembly stage.

Thanks to the previous data normalization which led to a redundancy reduction, the assembly stage has a lower computational cost. For both genome and transcriptome assembly is noticed that the stages with more parallelism and moderate RAM usage correspond to the contigs construction. Regarding to the transcriptome this stage called Chrysalisis which is responsible for processing graphs individually in parallel. Related to the genome, the stage with more parallelism is the one that analyzes the contigs extensions using the De Bruijn's graphs.

For both genome and transcriptome was noted that the stage that consumes more computational time is the annotation stage, this stage is completely parallelized and its duration depends mainly on the database which is compared and the number and the size of the sequences being compared. As the same as the annotation stage, the mapping stage is completely parallelized and its duration depends mainly on the number of reference reads and the number of contigs generated. Note that this assessment did not take into account the indexing stage and the conversion to other file formats.

5. Conclusions

Related to the RAM use the conclusion is as follows: The stage with the major use of memory is normalization, due that this one requires to load all the raw lectures in the memory. The stages with the least use of memory are the analysis and mapping stages. This conclusions are valid for genomic and transcriptomic data.

About the execution time the conclusion is as follows: The annotation stage is the one which requires more processing time mostly for the size of the reference database and the genome o transcriptome size to annotate. The stage with the least execution time for genomic and transcriptomic data is the data analysis stage.

Regarding to parallelization we conclude the following: The stages where the most parallelization is present are mapping and annotation stages for both types of data, this is mainly because of the process independence. The sub stage where the less parallelization of algorithms is used is the graph construction for genomic data assembly.

6. Acknowledgements

To the high performance computer center of Universidad Distrital (CECAD) for the given equipment to make the development of this project. To the Genetics Institute from Universidad Nacional (IGUN), and especially to evolutive immunology group (GIE) for giving the necessary data to perform this evaluation.

References

[1] J. Zhang, R. Chiodini, A. Badr and G. Zhang, 'The impact of next-generation sequencing on genomics', Journal of Genetics and Genomics, vol. 38, no. 3, pp. 95-109, 2011.

[2] Z. Wang, M. Gerstein and M. Snyder, 'RNA-Seq: a revolutionary tool for transcriptomics', Nat Rev Genet, vol. 10, no. 1, pp. 57-63, 2009.
[3] S. Schuster, 'Next-generation sequencing transforms today's biology', Nat Meth, vol. 5, no. 1, pp. 16-18, 2007.
[4] B. Haas, A. Papanicolaou, M. Yassour, M. Grabherr, P. Blood, J. Bowden, M. Couger, D. Eccles, B. Li, M. Lieber, M. MacManes, M. Ott, J. Orvis, N. Pochet, F. Strozzi, N. Weeks, R. Westerman, T. William, C. Dewey, R. Henschel, R. LeDuc, N. Friedman and A. Regev, 'De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis', Nature Protocols, vol. 8, no. 8, pp. 1494-1512, 2013.
[5] C. T. Brown, A. Howe, Q. Zhang, A. B. Pyrkosz, and T. H. Brom, "A reference-free algorithm for computational normalization of shotgun sequencing data" arXiv preprint arXiv:1203.4802, 2012.
[6] C. Rojas-Quintero, J. Perez-Castillo and N. Vera-Parra, 'Massive Automatic Functional Annotation. (MAFA)', in 2nd International Work-Conference on Bioinformatics and Biomedical Engineering, Granada, 2014.
[7] A. Oshlack, M. Robinson and M. Young, 'From RNA-seq reads to differential expression results', Genome Biol, vol. 11, no. 12, p. 220, 2010.
[8] T. Illumina, 1st ed. An introduction to next-generation sequencing technology: Illumina, 2011.
[9] J. Clarke, Oracle exadata recipes. New York: Apress, 2013, pp. 411-444.
[10] H. Wickham, Ggplot2. Dordrecht: Springer, 2009.
[11] S. Andrews et al., "Fastqc: A quality control tool for high throughput sequence data," Reference Source, 2010.
[12] S. F. Altschul, T. L. Madden, A. A. Sch¨affer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," Nucleic acids research, vol. 25, no. 17, pp. 3389–3402, 1997.
[13] 'The Universal Protein Resource (UniProt) 2009', Nucleic Acids Research, vol. 37, no., pp. D169-D174, 2009.
[14] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," Bioinformatics, vol. 25, no. 14, pp. 1754–1760, 2009.
[15] J. Simpson, K. Wong, S. Jackman, J. Schein, S. Jones and I. Birol, 'ABySS: A parallel assembler for short read sequence data', Genome Research, vol. 19, no. 6, pp. 1117-1123, 2009.
[16] G. Marcais and C. Kingsford, 'A fast, lock-free approach for efficient parallel counting of occurrences of k-mers', Bioinformatics, vol. 27, no. 6, pp. 764-770, 2011.