Bachelor Thesis

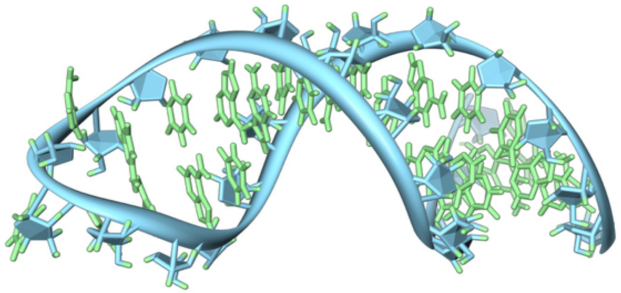
Performance tuning and parallelization of Inchworm Sequence Assembler

Ankur Sharma
Dresden, 10.09.2014
National Institute of Technology Sikkim

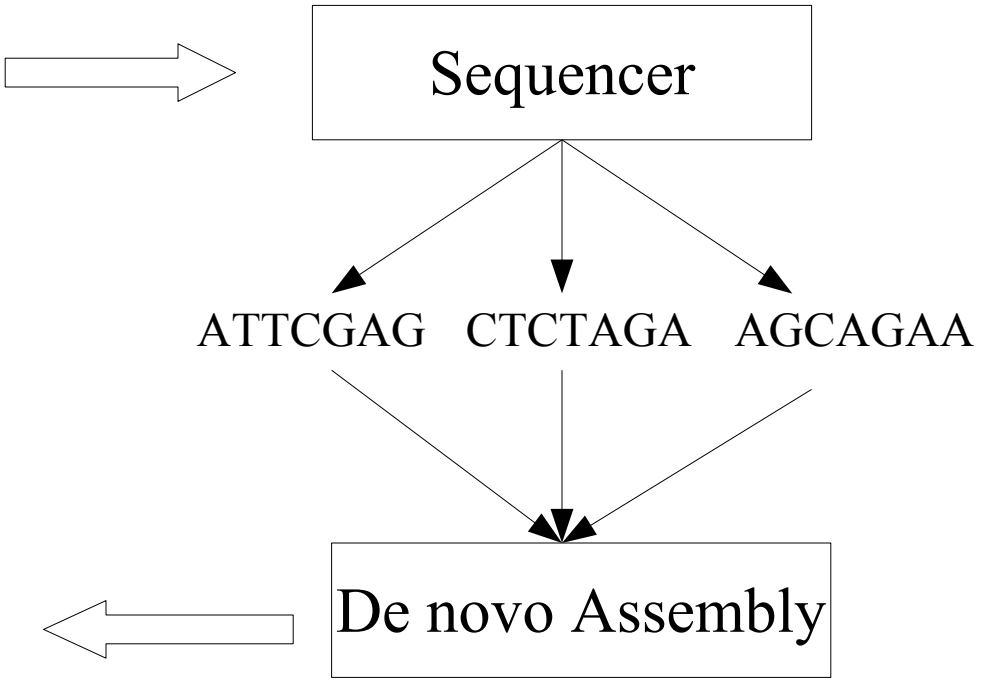
Agenda

- Introduction
- Trinity : Architecture & working
- Inchworm
- Phases & bottlenecks
- Phase-wise optimisation techniques
- Results
- Conclusion & future work

De novo Assembly



RNA Molecule



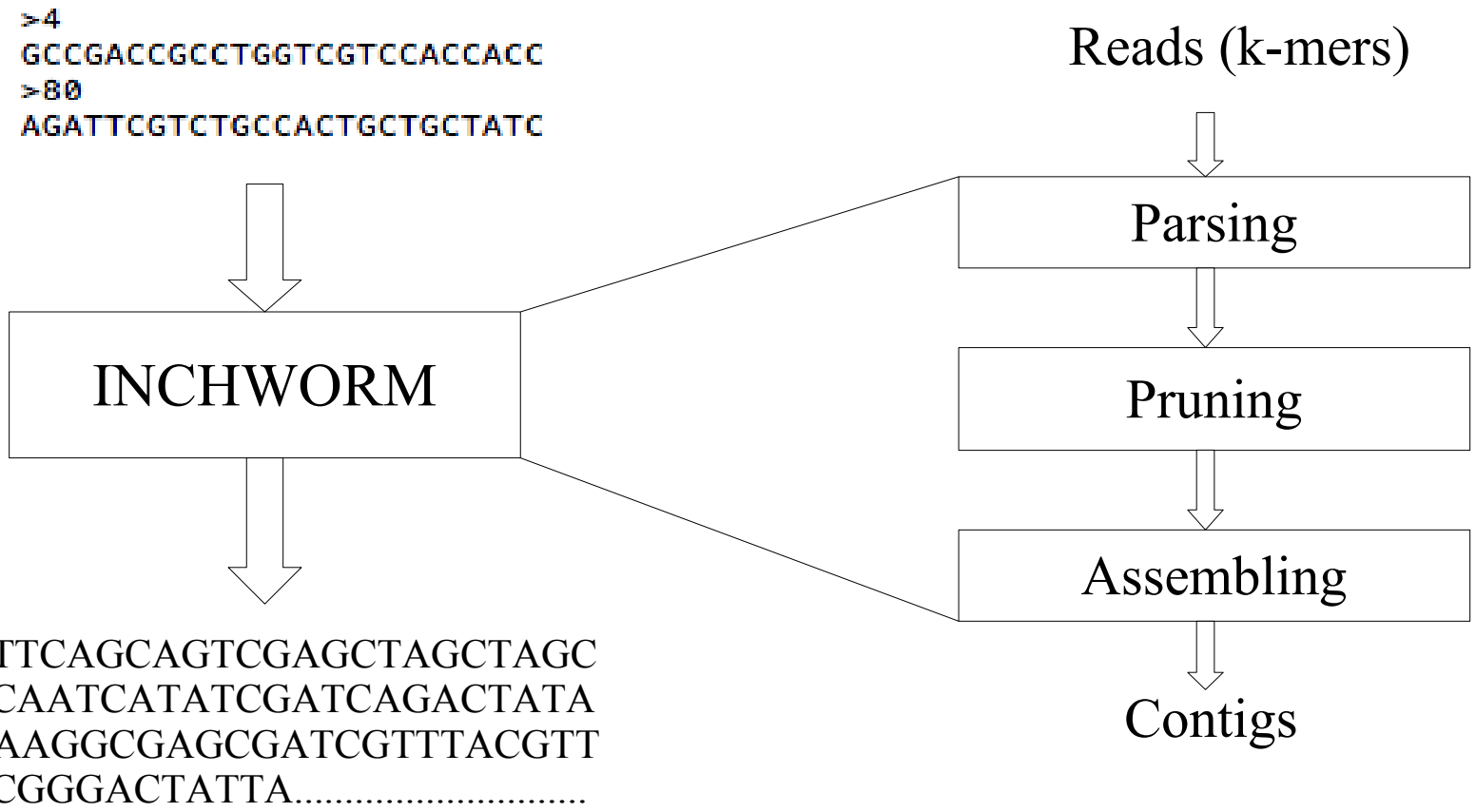
Encoded Representation of RNA (ATTAATAT.....)

Source: [htta]

- It is one of the methods for the efficient and robust de novo reconstruction of transcriptomes from sequence data.

Source: [tr01], [tr02]

Inchworm


>4
GCCGACCGCCTGGTCGTCCACCACC
>80
AGATTCGTCTGCCACTGCTGCTATC
INCHWORM
ATTTCAGCAGTCGAGCTAGCTAGC
ATCAATCATATCGATCAGACTATA
CAAAGGCGAGCGATCGTTTACGTT
TACGGGACTATTA.....
Reads (k-mers)
Parsing
Pruning
Assembling
Contigs

Parsing: Introduction/Bottlenecks

- Reads k-mers one by one from fasta file.
- Parsing can be done in parallel.

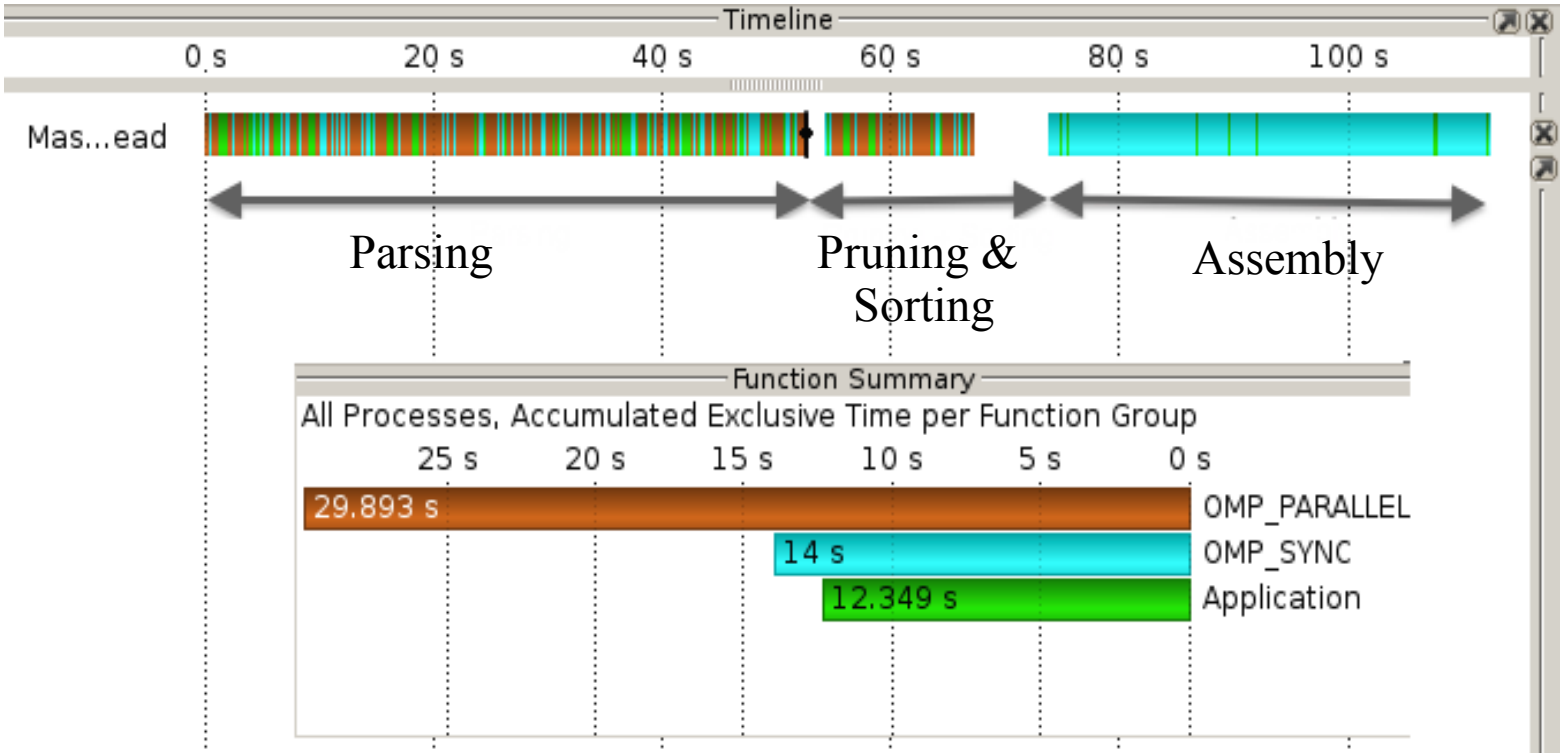


Fig: Trace from original implementation : Dataset 4M

Parsing: Optimization

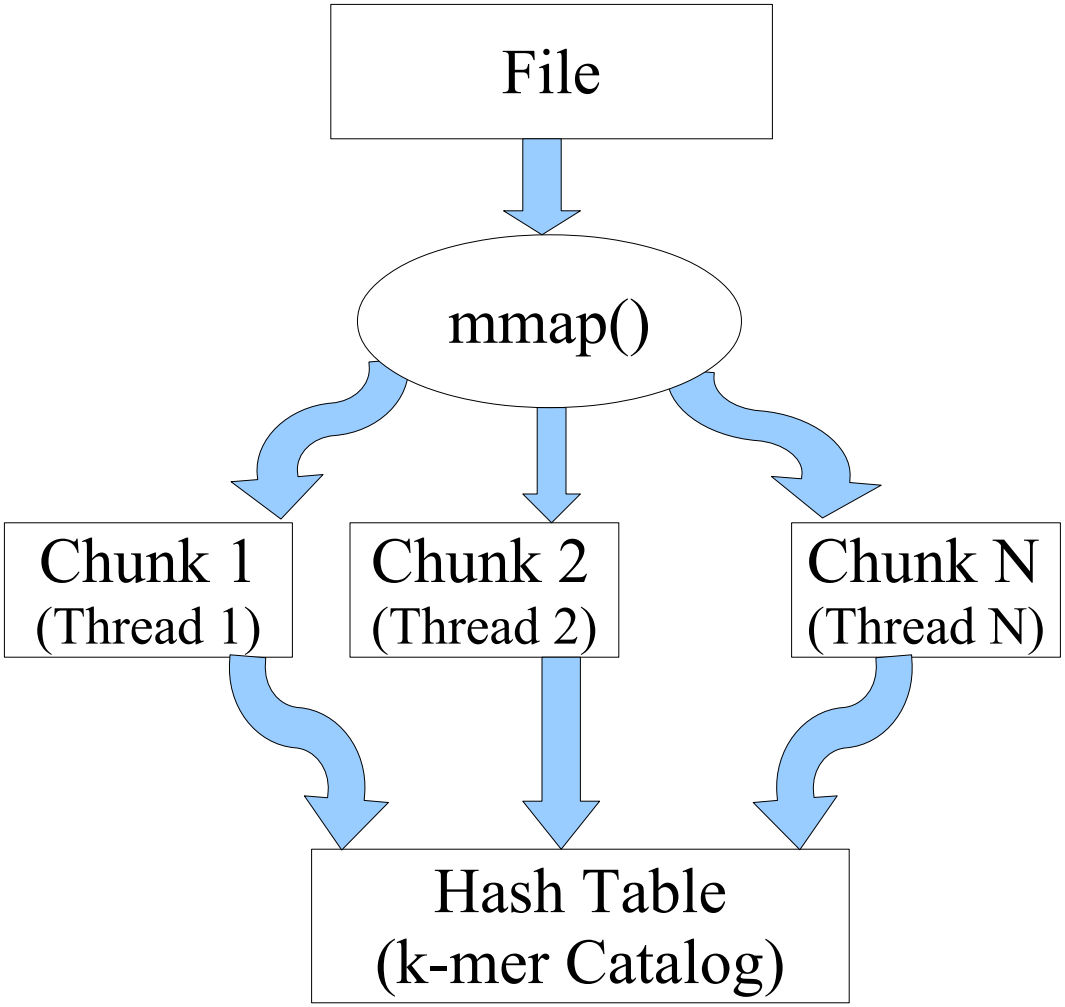
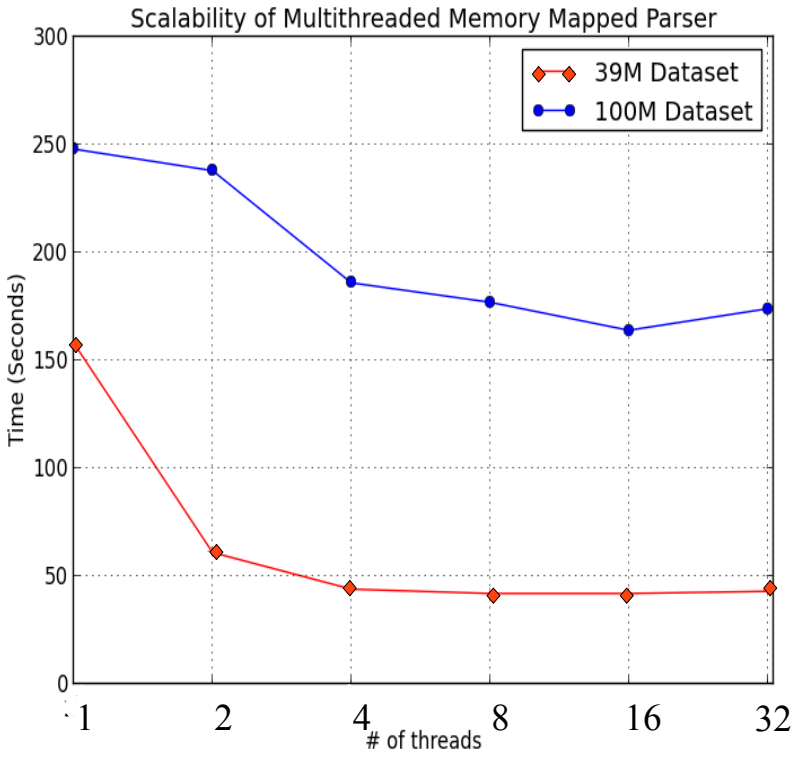
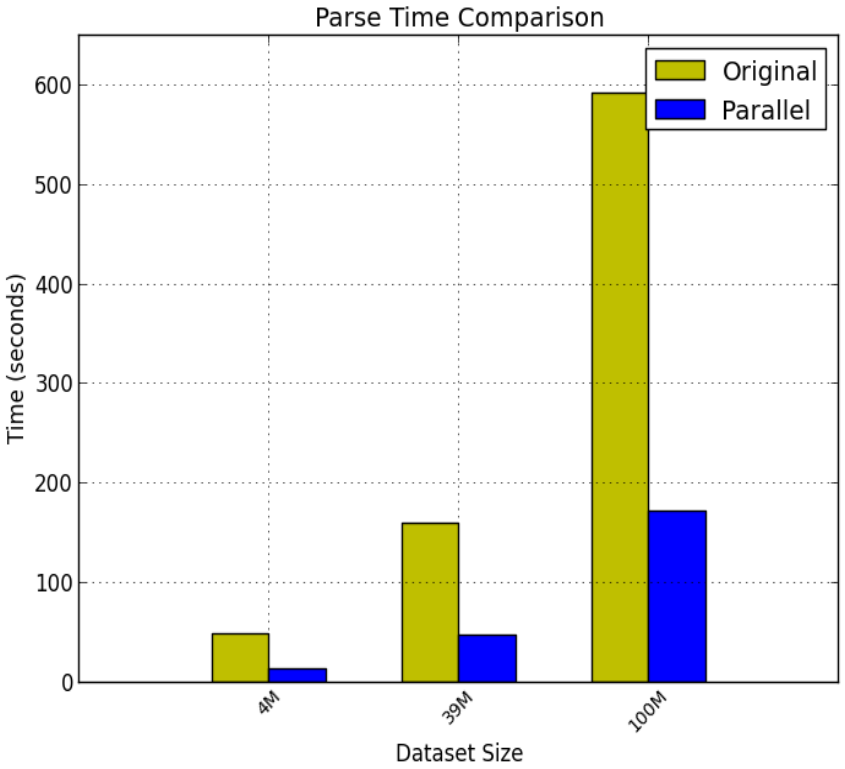


Fig: Memory mapped parser : Split, Lock & Insert

Parsing: Performance





Parsing: Performance

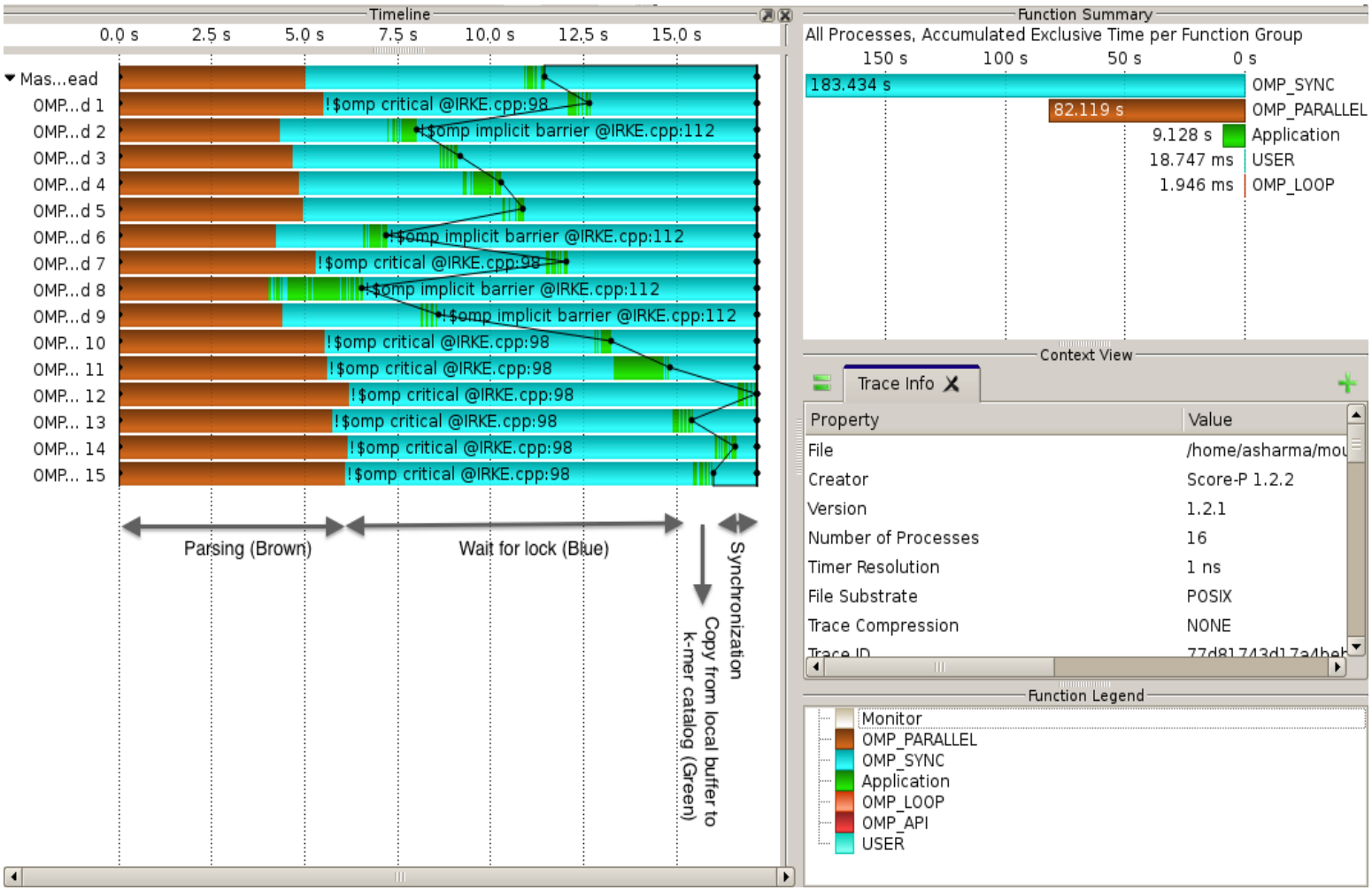


Fig: Trace – Parallel memory mapped parser

Pruning: Introduction/Optimization

- Pruning refers to filtering less complex k-mers out of the assembly procedure.

- Basically done in three different ways:
  - Based on abundance (Incorporated in Parsing)

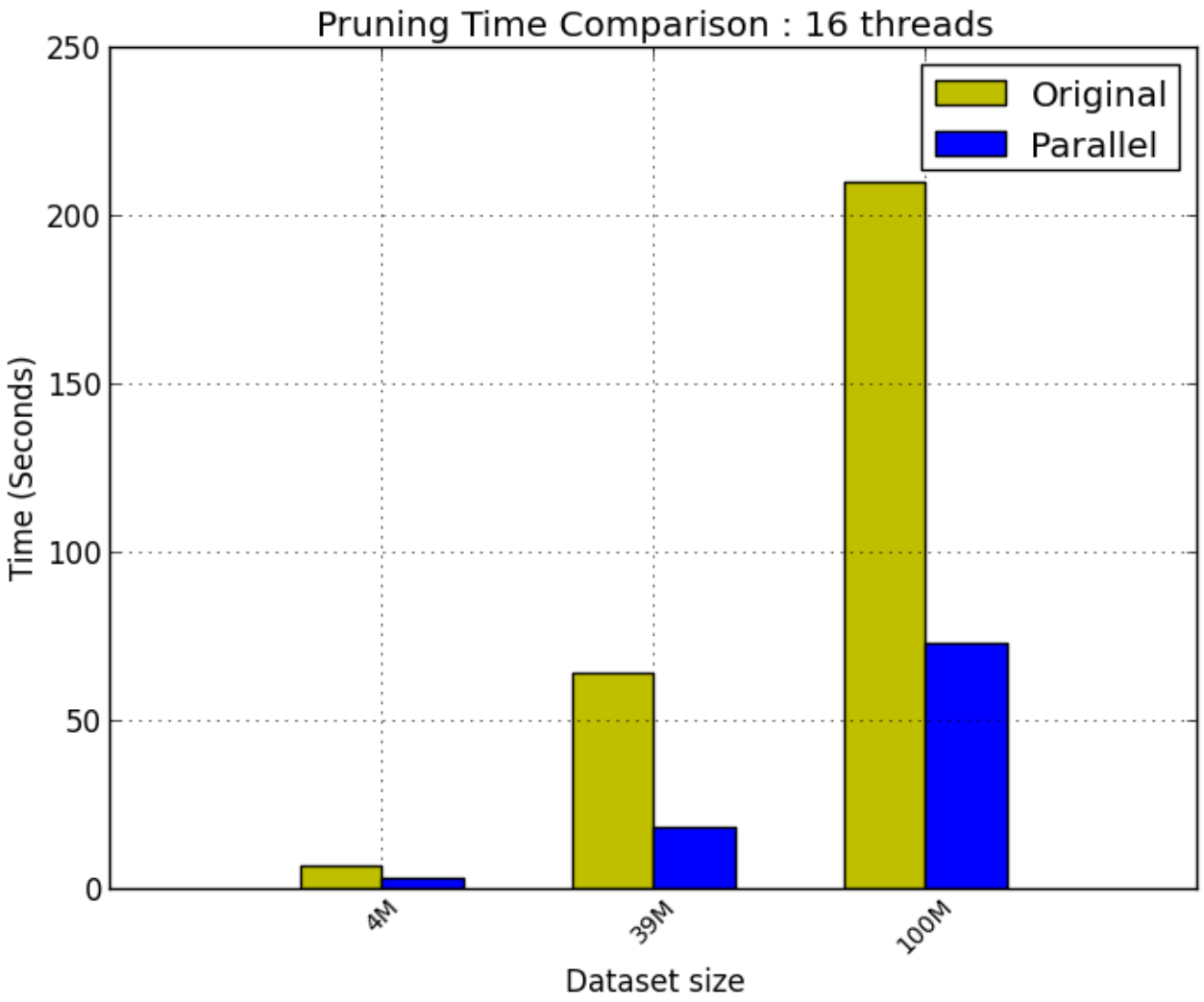
  - Based on entropy (Incorporated in Parsing)

$$Entropy = \sum_{x \in \{G,A,T,C\}} \frac{count(x)}{kmer\_length} * log_2\left(\frac{kmer\_length}{count(x)}\right)$$

  - Based on ratio evaluated for each extension k_1, k_2, k_3 & k_4 of k-mer k.

$$Ratio = \frac{kmer\_count}{dominant\_count}$$

Pruning: Performance Gain



Assembly: Introduction

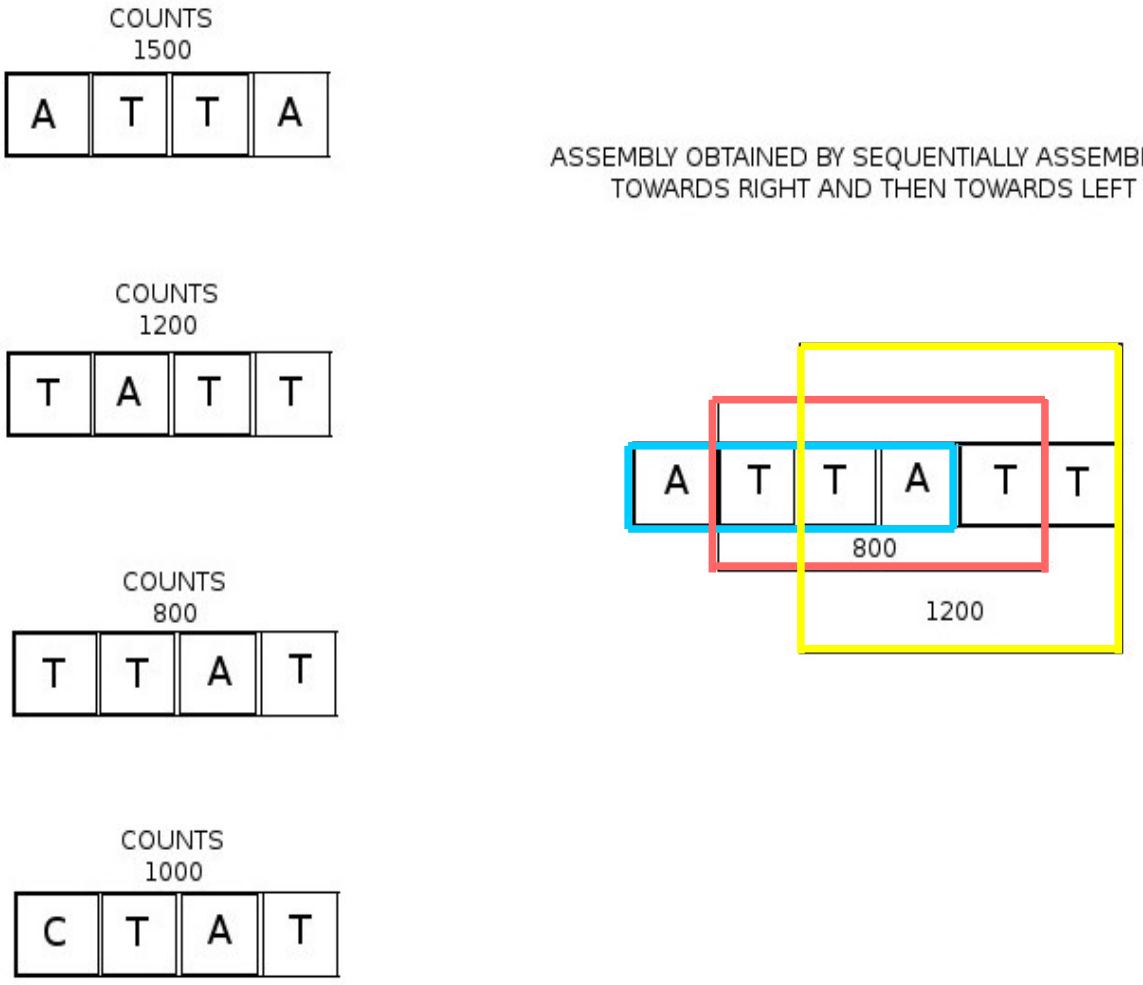


Fig: Graphical description of assembly algorithm

Assembly: Ambiguity

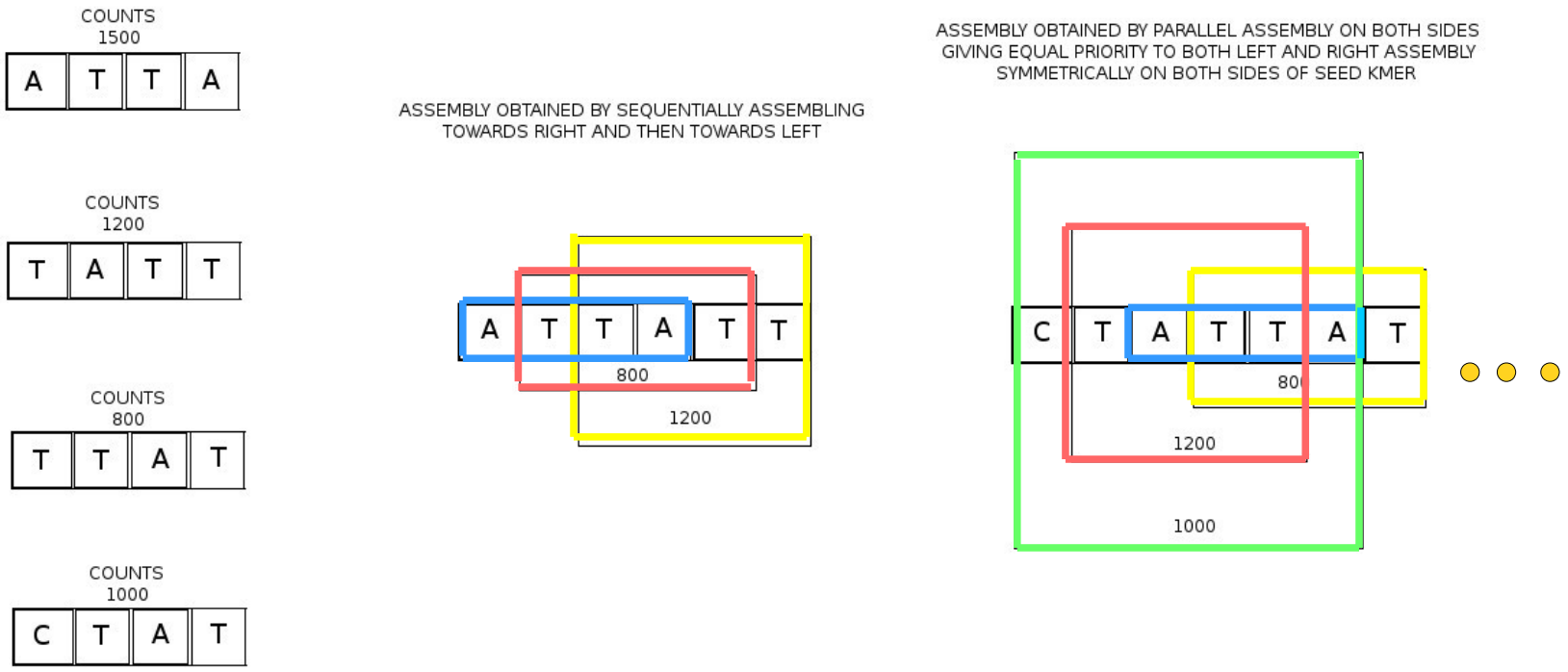


Fig: Ambiguity in assembly algorithm

Assembly: Optimization

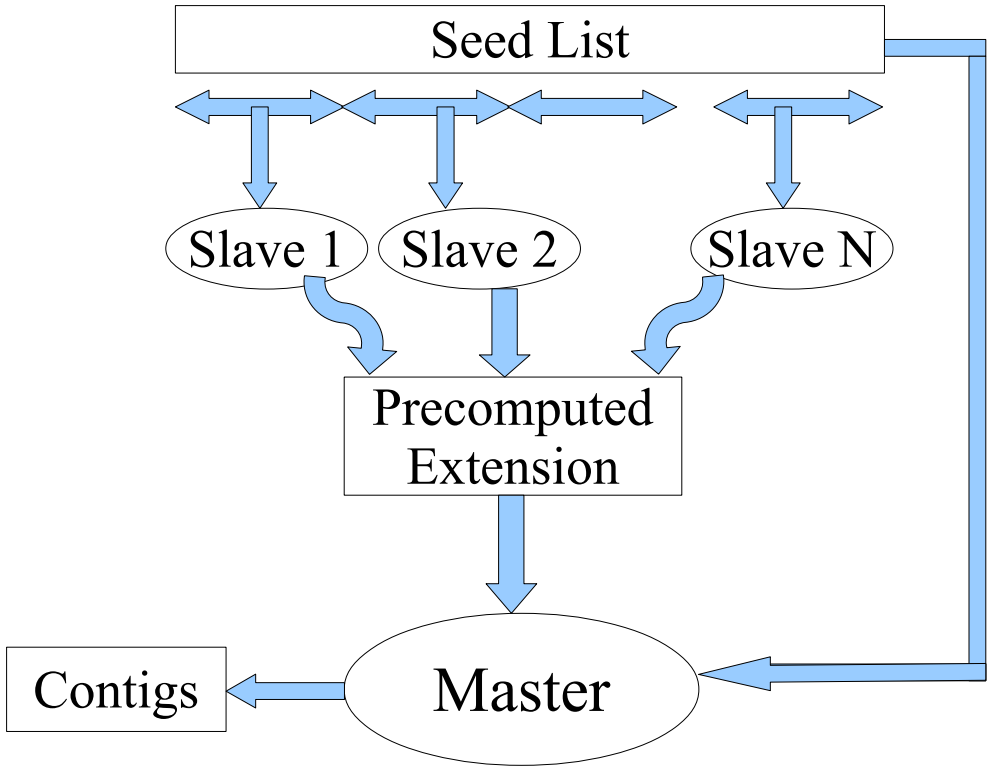


Fig: Graphical representation of Master-Slave assembly approach

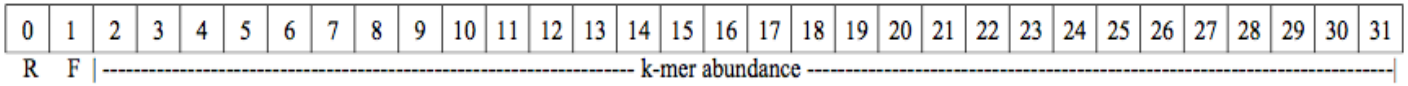
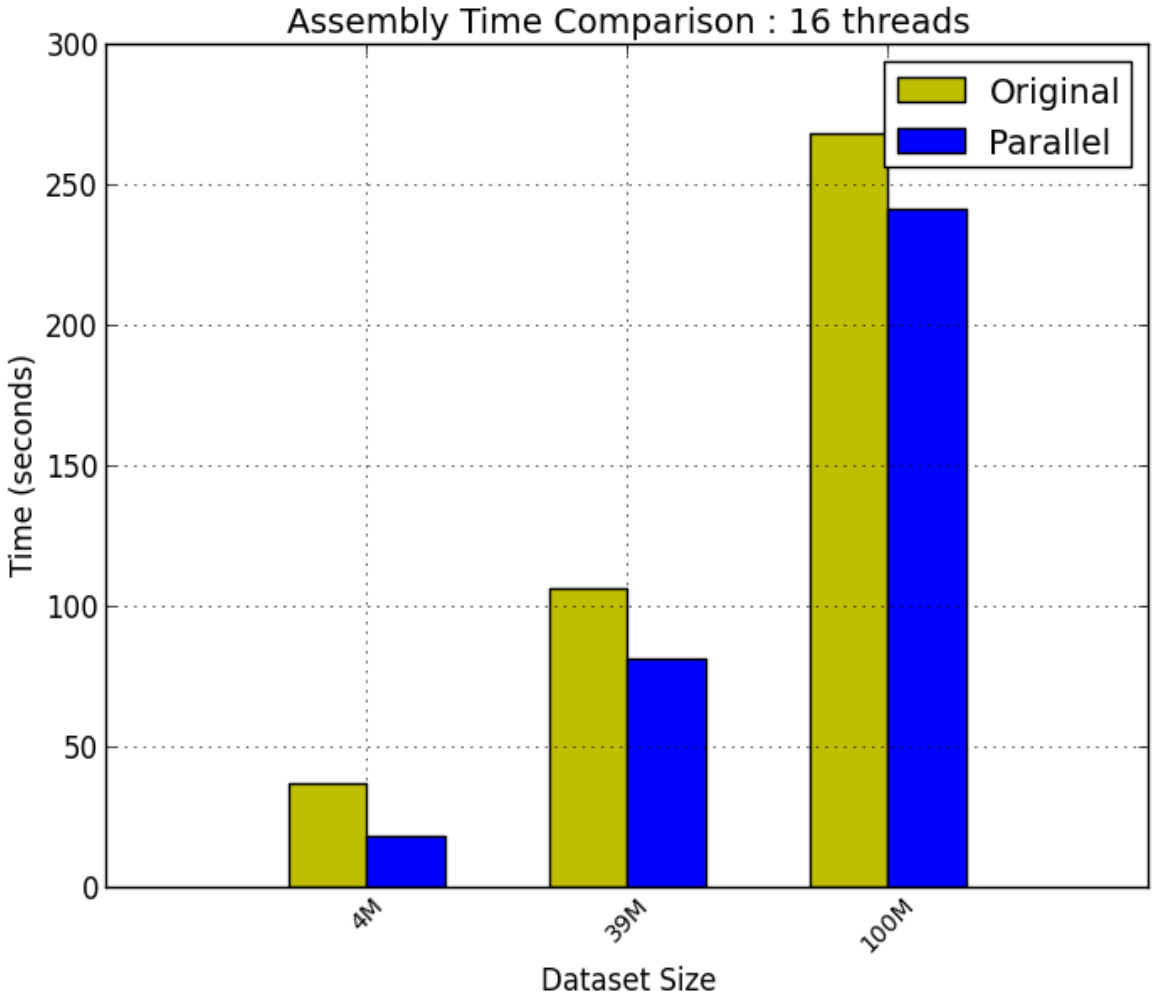


Fig: Marking used k-mers

Assembly: Performance



Summary & Comparison

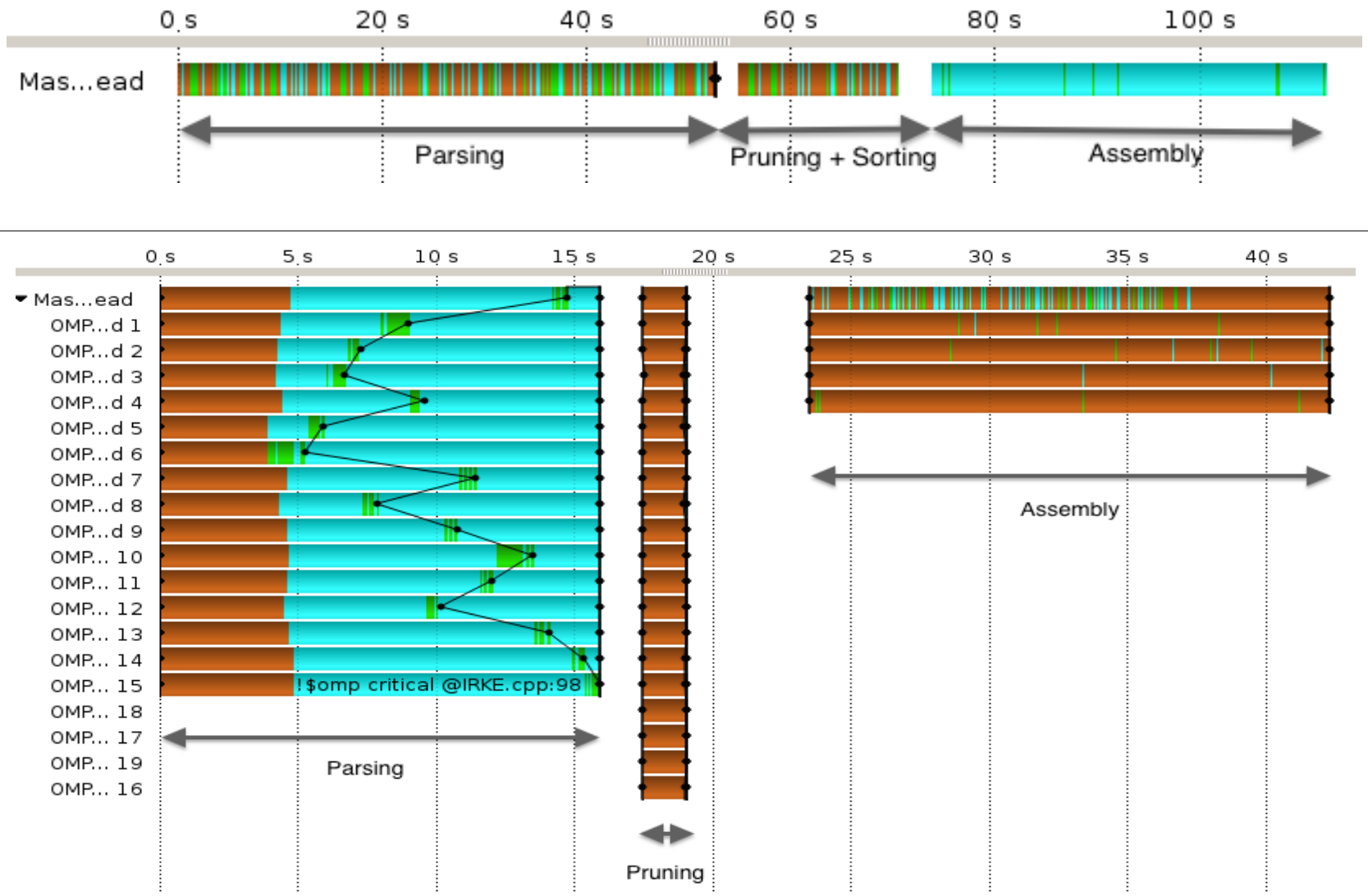
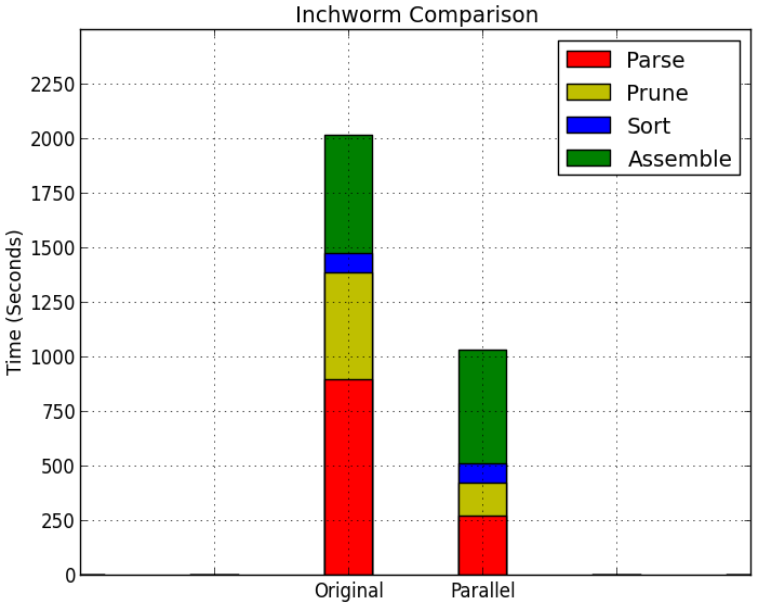


Fig: Comparison between present and parallel implementation

Results

(a)



(b)

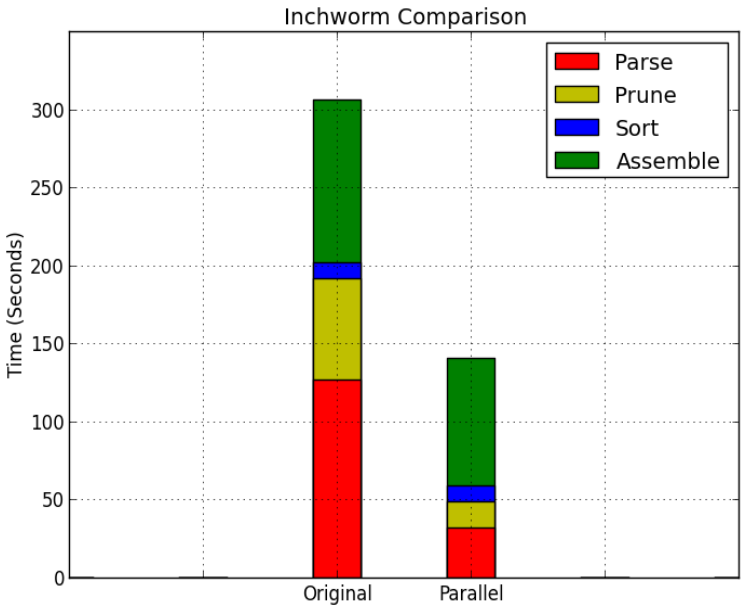


Fig: Parallel Inchworm v/s Original Implementation with datasets (a) *Schizosaccharomyces pombe* (b) *Drosophila melanogaster*

Dataset Name	# Contigs	# Base Pairs	# Max Length	# N50 Count	# Runtime (secs)
39M[1]	128,426	24,051,266	11,235	609	306
39M[2]	150,342	30,947,748	11,235	646	142
Schizo[1]	1,448,757	73,681,874	17,806	49	2019
Schizo[2]	1,758,751	102,085,873	17,708	51	1040

Fig: Statistical comparison of parallel and original implementation (1) Current Implementation (2) Parallel Implementation

Future Works

- Using a lock free hash table like hopscotch [HTS08, Ihht14] to avoid overhead of locking the k-mer catalog.
- Tuning the parallel prune method to avoid removing extra k-mers by keeping track of already removed k-mers.
- Optimize master slave approach to get better speed up.

References

[tr01] Manfred G Grabherr, Brian J Haas, Moran Yassour, Joshua Z Levin, Dawn A Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, Zehua Chen, Evan Mauceli, Nir Hacohen, Andreas Gnirke, Nicholas Rhind, Federica di Palma, Bruce W Birren, Chad Nusbaum, Kerstin Lindblad-Toh, Nir Friedman, and Aviv Regev. Full-length transcriptome assembly from rna-seq data without a reference genome. Nature Biotechnology 29, pages 644 – 652, 2011.

[tr02] Robert Henschel, Matthias Lieber, Le-Shin Wu, Phillip M. Nista, Brian J. Haas, and Richard D. LeDuc. Trinity rna-seq assembler performance optimization. In Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridg- ing from the eXtreme to the Campus and Beyond, XSEDE '12, pages 45:1–45:8, New York, NY, USA, 2012. ACM.

[HPY+ 13] Brian J. Haas, Alexie Papanicolaou, Moran Yassour, Manfred Grabherr, Philip D. Blood, Joshua Bowden, Matthew Brian Couger, David Eccles, Bo Li, Matthias Lieber, Matthew D. MacManes, Michael Ott, Joshua Orvis, Nathalie Pochet, Francesco Strozzi, Nathan Weeks, Rick Westerman, Thomas William, Colin N. Dewey, Robert Henschel, Richard D. LeDuc, Nir Friedman, and Aviv Regev. De novo transcript sequence reconstruction from rna-seq: reference generation and analysis with trinity. In Nature Protocols, volume 8, pages 1494 – 1512, 2013.

[HST08] Maurice Herlihy, Nir Shavit, and Moran Tzafrir. Hopscotch hashing. In Gadi Taubenfeld, editor, Distributed Computing, volume 5218 of Lecture Notes in Computer Science, pages 350–364. Springer Berlin Heidelberg, 2008.

[htta] http://en.wikipedia.org/wiki/RNA. Wikipedia.

[httb] http://www.gnu.org/software/libc/manual/. Memory map: Eff ciency and malloc.

[lHht14] libhash : Hopscotch hash table. https://code.google.com/p/libhhash/wiki/intro, June 2014.

[Jac09] Adam Jacobs. The pathologies of big data. Commun. ACM, 52(8):36–44, August 2009.

[TW00] T.D. Tock and T.K. Wong. Eff cient hash table for use in multi-threaded environments, September 5 2000. US Patent 6,115,802.

[ZWK+11] Qiong-Yi Zhao, Yi Wang, Yi-Meng Kong, Da Luo, Xuan Li, and Pei Hao. Optimizing de novo transcriptome assembly from short-read rna-seq data: a comparative study. BMC Bioinformatics, December 2011.

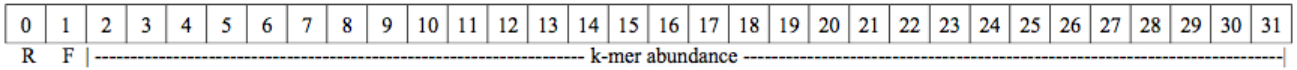
THANK YOU

Assembly: Optimization

- If k-mer has independent occurrence in both direction, we can use the k-mer in both directions.
- If assembly needs to be symmetrical, why give preference to forward direction? We can assemble both directions independently in parallel.
- As the k-mer length grows, the probability that the same k-mer is an extension of multiple k-mers decreases. So very less chance of conflict.
- If we can perform forward and reverse extension in parallel, we can use a master-slave approach to produce precomputed results.

Assembly: Master-Slave Approach

- Seeds are distributed amongst slaves to evaluate precomputations.
- Seperate slaves for computing forward and reverse extensions.
- Master uses the precomputed extensions directly if they are valid.
- To ensure that the same k-mer is not used again, 2 bits of the datastructure holding the k-mer abundance is used as flags.



- F is set, when the k-mer is used in forward assembly, and R when it is used in reverse assembly.
- Only master marks the k-mers to avoid race conditions and slaves reject these k-mers in further extension

Assembly: Master-Slave Approach

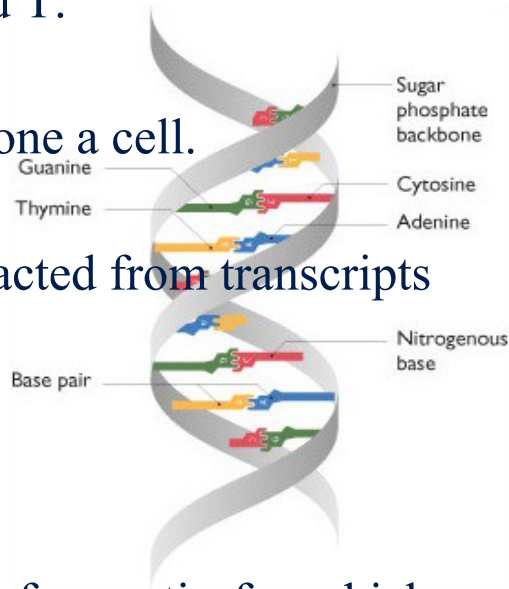
- Only master marks the k-mers to avoid race conditions and slaves reject these k-mers in further extension.
- Master verifies that extensions precomputed by workers does not contain any already used k-mers.
- If it finds such a conflict, it drops the precomputed extension and recomputes it itself.
- Only 1% of total computation is done by master. 99% of the precomputed results are correct.
- But verification process takes a lot of time which counters the speedup obtained.

Introduction

- Discuss Trinity and one of its three modules.
- Describe present implementation, performance analysis and bottlenecks of Inchworm.
- Highlight new parallel approach of optimizing sub modules/phases of Inchworm.
- Compare parallel and present implementation.

Important Terms

- Nucleotide : These are organic molecules that serve as a building block of large nucleic acid molecules like DNA and RNA. They carry packets of energy within the cell and regulates metabolism. IUPAC has assigned standard symbols for different types of nucleotides like A, G, C and T.

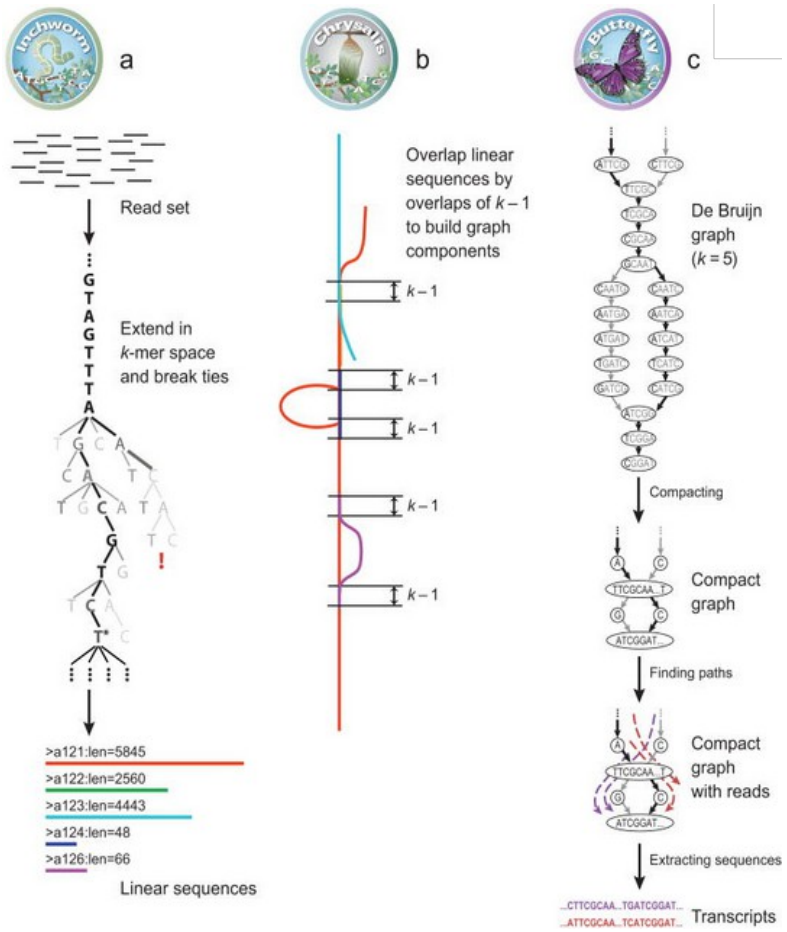


- Transcripts : It is a set of all RNA molecules produced in one a cell.
- Read : It is a short sequence of nucleotides which are extracted from transcripts using high throughput sequencers.
- K-mer : It is a k-tuple that represents a read or a sequence.
- N50 : For a set of contig, the N50 represents the length of a contig for which the number of contigs larger in length is same as the number of contigs of smaller length.
- De Buijn Graph : It is a directed graph that represents overlapping between the sequences.

Source : [htta]

Trinity: Modules

- Inchworm assembles the RNA-seq data into the unique sequences of larger linear transcripts.
- Chrysalis clusters the Inchworm contigs into clusters and constructs complete de Bruijn graphs for each cluster.
- Butterfly then processes the individual graphs in parallel, tracing the paths that reads and pairs of reads take within the graph for ultimately reporting full-length transcripts.



Source: [tr01], [tr02]

Parsing: Bottlenecks

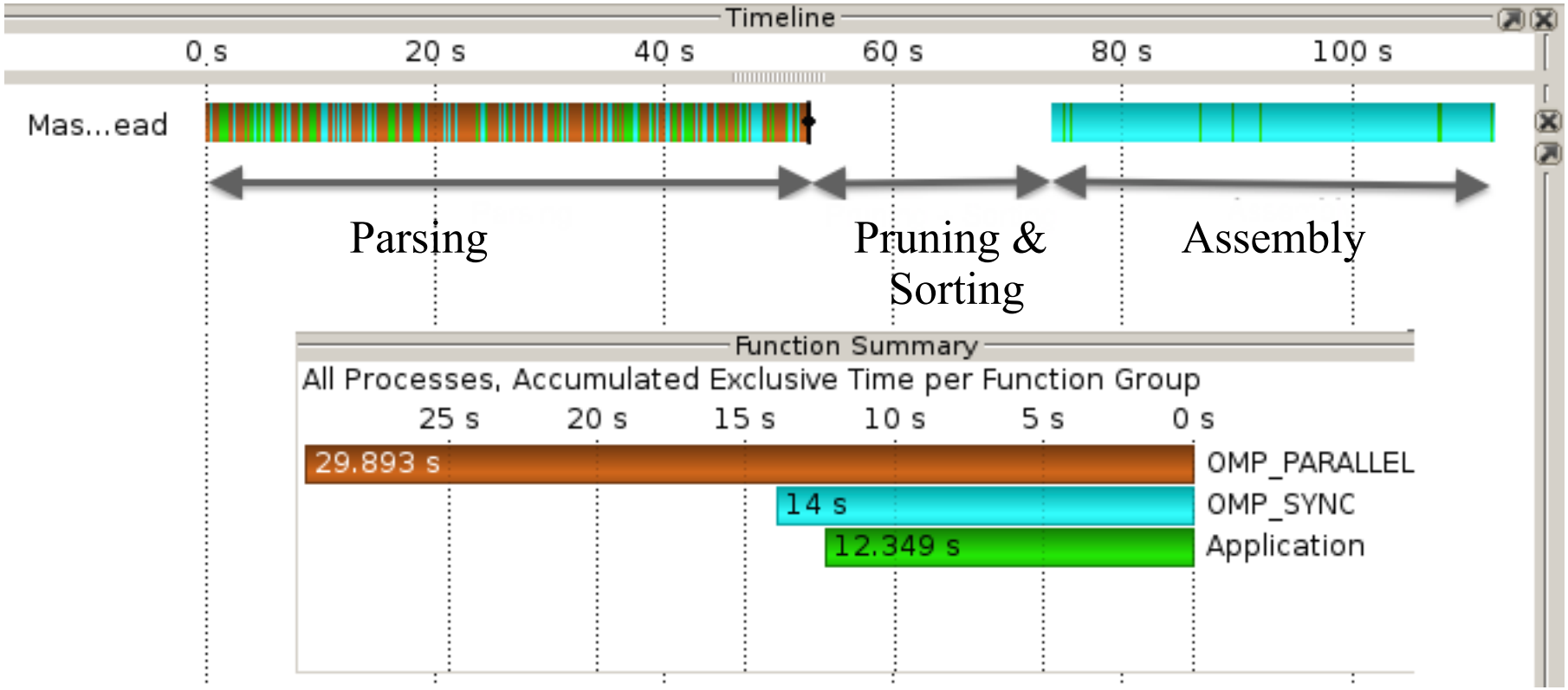


Fig: Trace from original implementation : Dataset 4M