ParsEvalMPI: comparison of gene structure annotations in parallel

Daniel S. Standage

Bioinformatics and Computational Biology

May 3, 2011

Gene prediction (annotation)

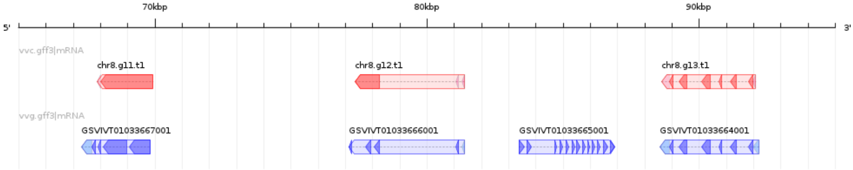
Input

String representing DNA sequence

Output

Regions (coordinates) of sequence that correspond to genes and their structural components

Comparing annotations



Previous implementation

ParsEval

- written in Perl

Previous implementation

ParsEval

- written in Perl
- external dependencies (platform-specific)

Previous implementation

ParsEval

- written in Perl
- external dependencies (platform-specific)
- significant memory demands

Previous implementation

ParsEval

- written in Perl
- external dependencies (platform-specific)
- significant memory demands
- serial execution: run time in minutes to hours

New implementation

ParsEvalMPI

- written in C

New implementation

ParsEvalMPI

- written in C
- external dependencies (cross-platform)

New implementation

ParsEvalMPI

- written in C
- external dependencies (cross-platform)
- reduced memory demands

New implementation

ParsEvalMPI

- written in C
- external dependencies (cross-platform)
- reduced memory demands
- parallel execution: run time in ...

Program overview

Program overview

- delegation

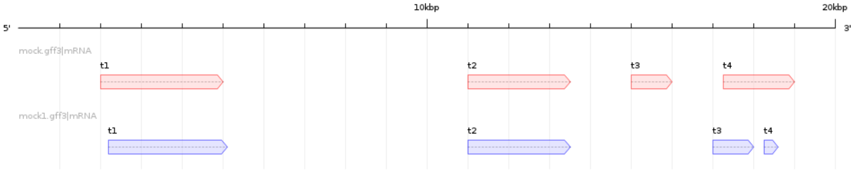
Program overview

- delegation
- local analysis

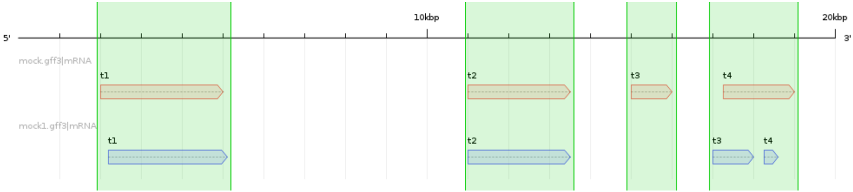
Program overview

- delegation
- local analysis
- global analysis

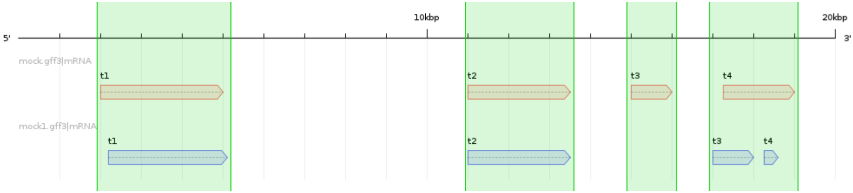
Delegation


10kbp
20kbp
5'
3'
mock.gff3|mRNA
t1
t2
t3
t4
mock1.gff3|mRNA
t1
t2
t3
t4

Delegation

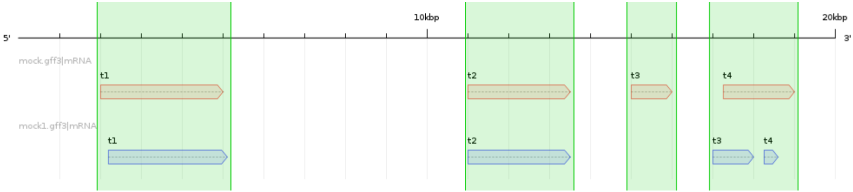

10kbp
20kbp
5'
3'
mock.gff3|mRNA
t1
t2
t3
t4
mock1.gff3|mRNA
t1
t2
t3
t4

Delegation



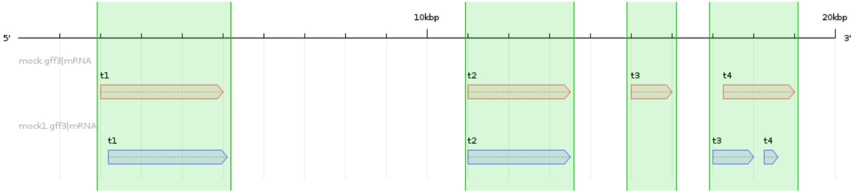
1. all data on all processors

Delegation



1. all data on all processors
2. even distribution of DNA

Delegation



1. all data on all processors
2. even distribution of DNA
3. even distribution of genes

Local and global analysis

- Local analysis

- Global analysis

Local and global analysis

- Local analysis
  - generate local model vector
- Global analysis

Local and global analysis

- Local analysis
  - generate local model vector
  - send local vector to global vector on root processor
- Global analysis

Local and global analysis

- Local analysis
  - generate local model vector
  - send local vector to global vector on root processor
  - analyze local vector, print scores
- Global analysis

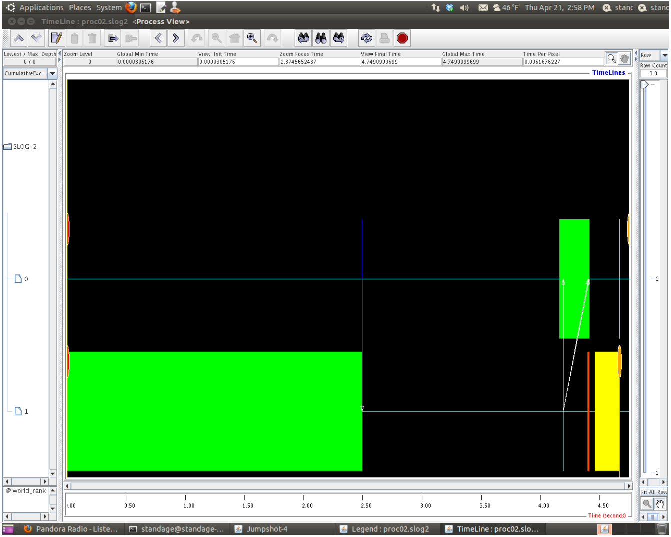
Local and global analysis

- Local analysis
  - generate local model vector
  - send local vector to global vector on root processor
  - analyze local vector, print scores
- Global analysis
  - receive local vectors from each processor

Local and global analysis

- Local analysis
  - generate local model vector
  - send local vector to global vector on root processor
  - analyze local vector, print scores
- Global analysis
  - receive local vectors from each processor
  - analyze combined global vector, print scores

Load balancing

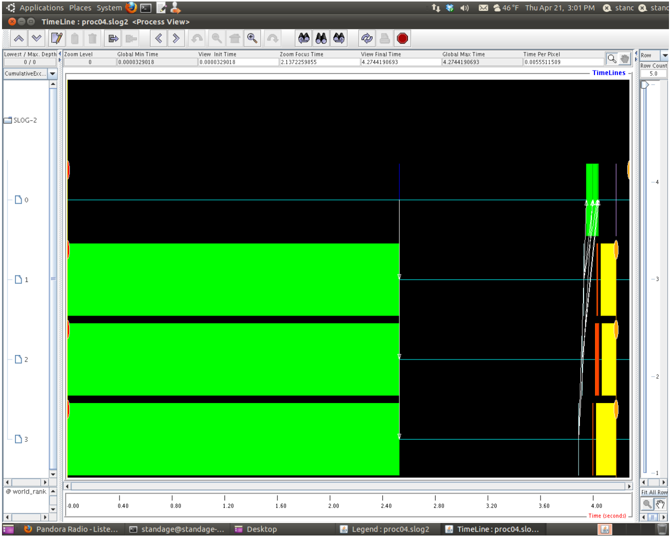


TimeLine : proc02.slog2 <Process View>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 0	0	0.0000305176	0.0000305176	2.3745652437	4.7490999699	4.7490999699	0.0061676227

Row Count 3.0

Load balancing



TimeLine : proc04.slog2 <Process View>

Lowest / Max. Depth: 0 / 0

Zoom Level: 0

Global Min Time: 0.0000329018

View Init Time: 0.0000329018

Zoom Focus Time: 2.1372259855

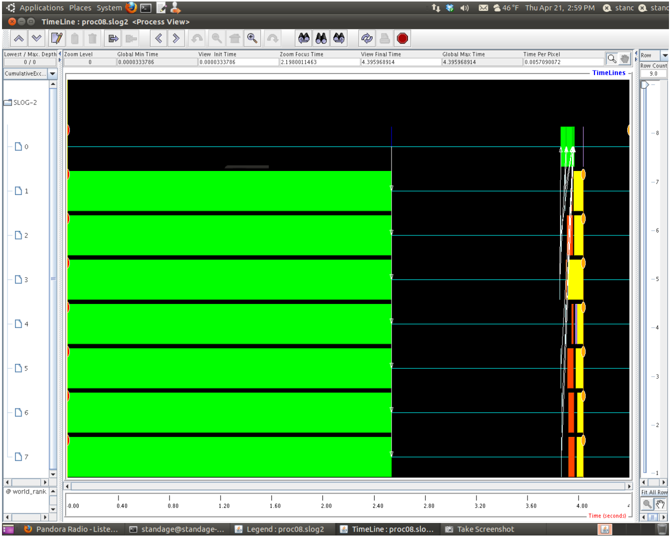
View Final Time: 4.2744190693

Global Max Time: 4.2744190693

Time Per Pixel: 0.0055511509

Row Count: 5.0

Load balancing

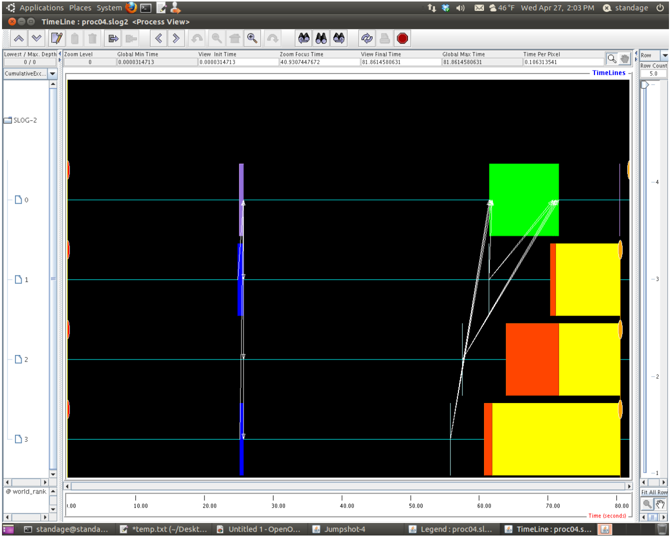


TimeLine : proc08.slog2 <Process View>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 0	0	0.0000333786	0.0000333786	2.1980011463	4.395968914	4.395968914	0.0057090072

Row Count 9.0

Load balancing



TimeLine : proc04.slog2 <Process View>

Lowest / Max. Depth: 0 / 0

Zoom Level: 0

Global Min Time: 0.0000314713

View Init Time: 0.0000314713

Zoom Focus Time: 40.9307447672

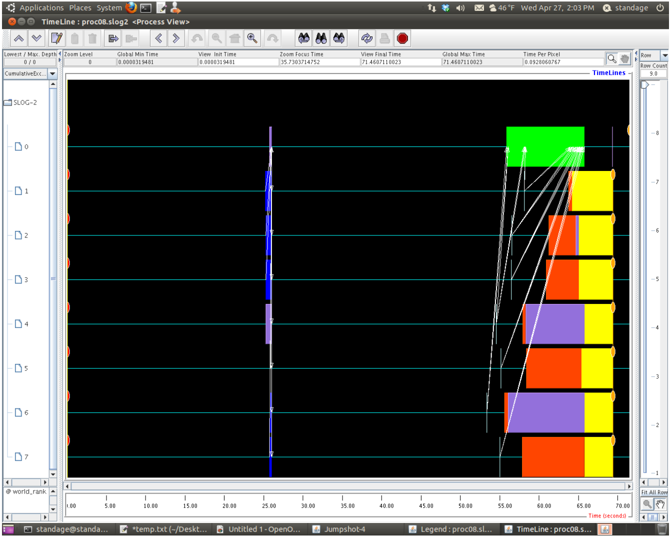
View Final Time: 81.8614580631

Global Max Time: 81.8614580631

Time Per Pixel: 0.106313541

Row Count: 5.0

Load balancing



TimeLine : proc08.slog2 <Process View>

Lowest / Max. Depth: 0 / 0

Zoom Level: 0

Global Min Time: 0.0000319481

View Init Time: 0.0000319481

Zoom Focus Time: 35.7303714752

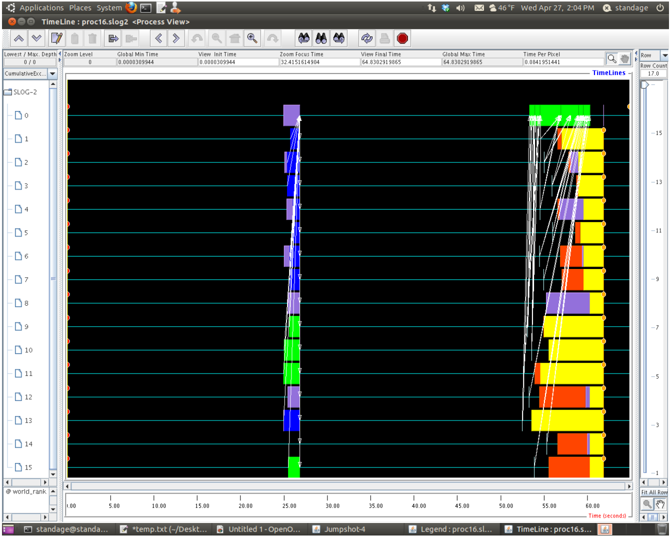
View Final Time: 71.4607110023

Global Max Time: 71.4607110023

Time Per Pixel: 0.0928060767

Row Count: 9.0

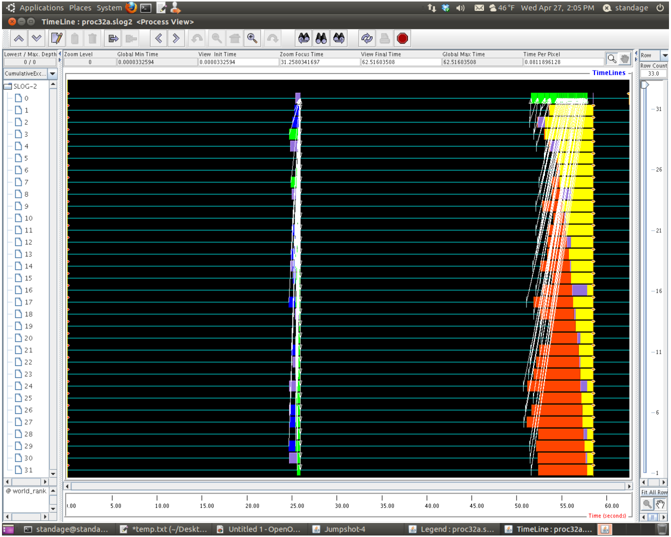
Load balancing



TimeLine : proc16.slog2 <Process View>

Lowest / Max. Depth: 0 / 0
Zoom Level: 0
Global Min Time: 0.0000309944
View Init Time: 0.0000309944
Zoom Focus Time: 32.4151614904
View Final Time: 64.8302919865
Global Max Time: 64.8302919865
Time Per Pixel: 0.0841951441
Row Count: 17.0

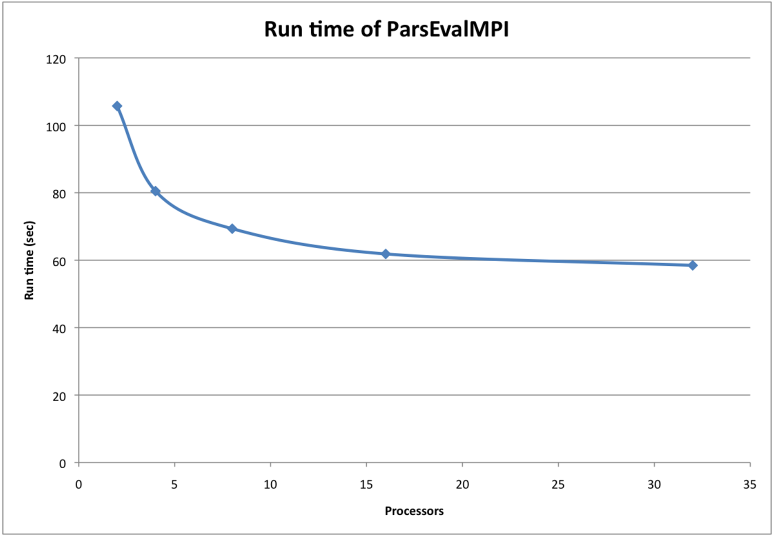
Load balancing



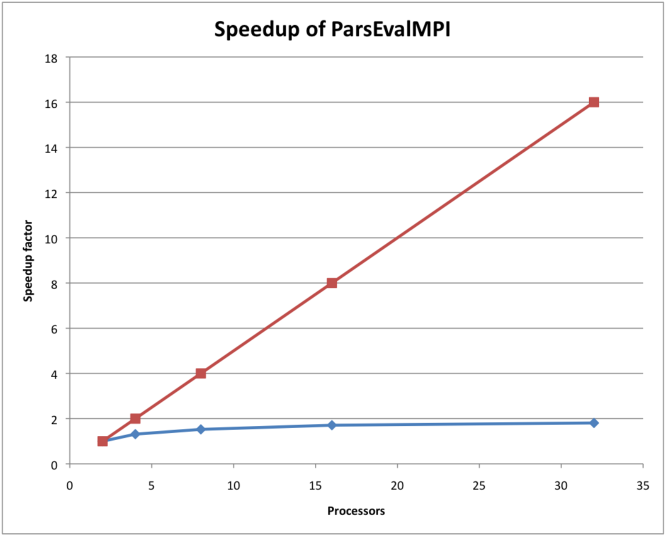
TimeLine : proc32a.slog2 <Process View>

Lowest / Max. Depth: 0 / 0
Zoom Level: 0
Global Min Time: 0.0000332594
View Init Time: 0.0000332594
Zoom Focus Time: 31.2580341697
View Final Time: 62.51603508
Global Max Time: 62.51603508
Time Per Pixel: 0.0811896128
Row Count: 33.0

Scalability



Scalability



Serial optimization

Serial optimization

- Good
  - native data types
  - static arrays

Serial optimization

- Good
  - native data types
  - static arrays
- Bad
  - pointers, dynamic arrays
  - dynamic data structures

Serial optimization

- Good
  - native data types
  - static arrays
- Bad
  - pointers, dynamic arrays
  - dynamic data structures
- Ugly
  - copying data

Conclusions

Conclusions

- Excellent data distribution, load balancing

Conclusions

- Excellent data distribution, load balancing
- Very poor scaling properties

Conclusions

- Excellent data distribution, load balancing
- Very poor scaling properties
    - maximum scaling factor of 2?!?!

Conclusions

- Excellent data distribution, load balancing
- Very poor scaling properties
  - maximum scaling factor of 2?!?!
  - perhaps try OpenMP

Conclusions

- Excellent data distribution, load balancing
- Very poor scaling properties
  - maximum scaling factor of 2?!?!
  - perhaps try OpenMP
- Significant improvement