Parallel machine learning approaches for reverse engineering genome-scale networks

Srinivas Aluru
School of Computational Science and Engineering
Institute for Data Engineering and Science (IDEaS)
Georgia Institute of Technology

Motivation

- Arabidopsis Thaliana
  - Widely studied model organism.
  - 125 Mbp genome sequenced in 2000.
  - About 22,500 genes and 35,000 proteins.
- NSF Arabidopsis 2010 Program launched in 2001
  - **Goal:** discover function(s) of every gene.
  - ∼\$265 million funded over 10 years
  - Sister programs such as *AFGN* by German Research Foundation (DFG).

Motivation

- Arabidopsis Thaliana
  - Widely studied model organism.
  - 125 Mbp genome sequenced in 2000.
  - About 22,500 genes and 35,000 proteins.
- NSF Arabidopsis 2010 Program launched in 2001
  - **Goal:** discover function(s) of every gene.
  - ∼\$265 million funded over 10 years
  - Sister programs such as *AFGN* by German Research Foundation (DFG).
- **Status today:** $>$ 30% genes with no known function.

Motivation

- Arabidopsis Thaliana
  - Widely studied model organism.
  - 125 Mbp genome sequenced in 2000.
  - About 22,500 genes and 35,000 proteins.
- NSF Arabidopsis 2010 Program launched in 2001
  - **Goal:** discover function(s) of every gene.
  - $\sim$\$265 million funded over 10 years
  - Sister programs such as *AFGN* by German Research Foundation (DFG).
- **Status today:** $>$ 30% genes with no known function.
- **How can computer science help?**

Motivation

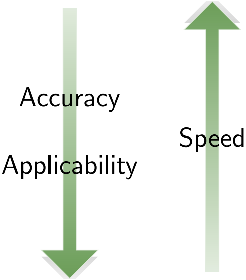
- Arabidopsis Thaliana
  - Widely studied model organism.
  - 125 Mbp genome sequenced in 2000.
  - About 22,500 genes and 35,000 proteins.
- NSF Arabidopsis 2010 Program launched in 2001
  - **Goal:** discover function(s) of every gene.
  - $\sim$\$265 million funded over 10 years
  - Sister programs such as *AFGN* by German Research Foundation (DFG).
- **Status today:** $>30\%$ genes with no known function.
- **How can computer science help?**
  - 11,760 microarray experiments available in public databases.
  - Construct genome wide networks to generate intelligent hypotheses.

Gene Networks

- Structure Learning Methods
  - Pearson correlation (D'Haeseleer *et al.* 1998)
  - Gaussian Graphical Models
    - GeneNet (Schafer *et al.* 2005).
  - Information Theory
    - ARACNe (Basso *et al.* 2005)
    - CLR (Faith *et al.* 2009)
  - Bayesian networks
    - Banjo (Hartemink *et al.* 2002)
    - bnlearn (Scutari 2010)

Gene Networks

- Structure Learning Methods
  - Pearson correlation (D'Haeseleer *et al.* 1998)
  - Gaussian Graphical Models
    - GeneNet (Schafer *et al.* 2005).
  - Information Theory
    - ARACNe (Basso *et al.* 2005)
    - CLR (Faith *et al.* 2009)
  - Bayesian networks
    - Banjo (Hartemink *et al.* 2002)
    - bnlearn (Scutari 2010)



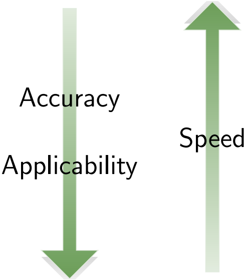
Accuracy

Applicability

Speed

Gene Networks

- Structure Learning Methods
  - Pearson correlation (D'Haeseleer *et al.* 1998)
  - Gaussian Graphical Models
    - GeneNet (Schafer *et al.* 2005).
  - Information Theory
    - ARACNe (Basso *et al.* 2005)
    - CLR (Faith *et al.* 2009)
  - Bayesian networks
    - Banjo (Hartemink *et al.* 2002)
    - bnlearn (Scutari 2010)


Accuracy

Applicability

Speed

Poor Prognosis

- Many do poorly on an absolute basis. One in three no better than random guessing.
- Compromise: Quality of method vs. data scale.

(Marbach *et al.*, *PNAS* 2010; *Nature Methods* 2012)

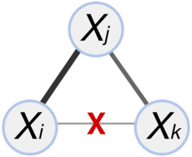
Information Theoretic Approach

- Connect two genes if they are dependent under mutual information

$$I(X_i; X_j) = I(X_j; X_i) = \mathcal{H}(X_i) + \mathcal{H}(X_j) - \mathcal{H}(X_i, X_j)$$

$$\mathcal{H}(X) = -\sum_{X \in X} P_x(X).\log(x)$$

- Remove indirect dependencies by Data Processing Inequality (Basso *et al.* PNAS 2005)



Permutation Testing

- For each (X_i, X_j), compute all $m!$ values of $I(X_i; \pi(X_j))$.

- Accept (X_i, X_j) as dependent if $I(X_i; X_j)$ is greater than at least the fraction $(1 - \epsilon)$ of all tested permutations.

- A large sample is used in practice.

Our Approach

We use the following property

$$I(X_i; X_j) = I(f(X_i); f(X_j))$$

where f is a homeomorphism.

We *rank transform* each profile, i.e., we replace $x_{i,l}$ with its rank in the set $\{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$ [Kraskov 2004]

Mutual information computed on rank transformed data. (Zola *et al.*, *IEEE TPDS 2010*)

Our Approach

- Each profile is a permutation of $1, 2, \ldots, m$
- A random permutation of one profile is a random permutation of another
- Use q permutations per pair for a total of $q \times \binom{n}{2}$ permutations
- $I(X_i, X_j) = 2 \times \mathcal{H}(< 1, 2, \ldots, m >) - \mathcal{H}(X_i, X_j)$

Tool for Inferring Network of Genes (TINGe)

Each step is done in parallel:
Input: $M_{n \times m}, \epsilon$
Output: $D_{n \times n}$

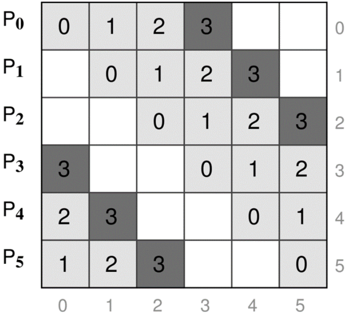
1. read M
2. rank transform each row of M
3. Compute MI between all $\binom{n}{2}$ pairs of genes, and $q \cdot \binom{n}{2}$ permutations
4. find I_0, $\epsilon \cdot q \cdot \binom{n}{2}$ largest value among permutations
5. remove values in D below threshold I_0
6. apply DPI to D
7. write D

Tool for Inferring Network of Genes (TINGe)

- Decomposes D into $p \times p$ submatrices.

- Iteration i: P_j computes $D_{j,(j+i) \bmod p}$

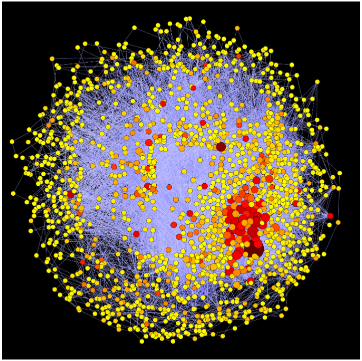
  (Zola *et al.*, *IEEE TPDS* 2010)



	0	1	2	3	4	5	
P_0	0	1	2	3			0
P_1		0	1	2	3		1
P_2			0	1	2	3	2
P_3	3			0	1	2	3
P_4	2	3			0	1	4
P_5	1	2	3			0	5

How Fast Can We Do This?

- 1,024 node IBM Blue Gene/L — 45 minutes (2007)
- 1,024 core AMD dual quad core Infiniband cluster — 9 minutes (2009)



- A single Xeon Phi accelerator chip — 22 minutes (Misra *et al.*, *IPDPS 2013*; *IEEE TCBB 2015*)

Arabidopsis Whole Genome Network

- Dataset
  - 11,760 experiments, each measuring $\sim 22,500$ genes.
  - Statistical normalization (Aluru *et al.*, *NAR* 2013).
- Dataset Classification
  - 9 tissue types (whole plant, rosette, seed, leaf, flower, seedling, root, shoot, and cell suspension)
  - 9 experimental conditions (chemical, development, hormone, light, pathogen, stress, metabolism, glucose metabolism, and unknown)

Dataset combinations

Generated 90 datasets including one for each $\langle$tissue, condition$\rangle$ pair.

Networks Component Analysis

- BR8000

Method	Genes	Edges	Comp.	Largest Comp.	%
GeneNet	4447	15703	791	(3612, 15652)	55.58
ACGN	3977	198848	175	(3787, 198830)	49.71
TINGe	6646	136681	8	(6639, 136681)	83.07
AraNet	7420	142284	325	(7073, 142260)	92.75

- RD26-8725

Method	Genes	Edges	Comp.	Largest Comp.	%
GeneNet	4709	17890	801	(3859, 17839)	53.97
ACGN	4253	319757	183	(4059, 319745)	46.52
TINGe	7049	162091	16	(7034, 162091)	80.79
AraNet	8062	231478	351	(7703, 231468)	92.40

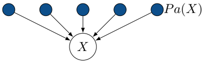
Validation against ATRM

- Arabidopsis Transcription Regulatory Map (Jin *et al.*, 2015)
  - Experimentally validated interactions extracted via text mining.
  - 1431 interactions among 790 genes.
- Results : % of identified interactions vs. cut off distance.

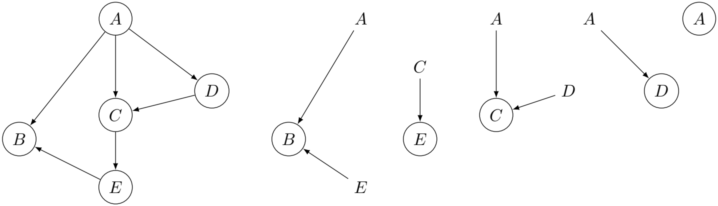
Method	Cut off Distance		
	1	2	3
ACGN	4.13	14.26	25.02
GeneNet	5.77	35.54	61.65
TINGe	9.43	50.66	97.11
AraNet	14.88	43.26	85.34

Score-based Bayesian Network Structure Learning

- Scoring Function : $s(X, Pa(X))$
  - Fitness of choosing set $Pa(X)$ as parents for X



- Score of a network N



$$Score(N) = \sum_{X_i} s(X_i, Pa(X_i))$$

Bayesian Network Modeling

- Bayesian Networks
  - DAG N and joint probability P such that $X_i \perp\!\!\!\perp ND(X_i)|Pa(X_i)$
  - Super exponential search space in n: $\frac{n!2^{\frac{n}{2}(n-1)}}{rz^n}$ possible DAGs over n variables, $r \approx 0.57436$, $z \approx 1.4881$ (Robinson, 1973)
  - NP-hard even for bounded node in-degree (Chickering *et al.*, 1994)]

- Optimal Structure Learning
  - Serial: $O(n^2 2^n)$; $n = 20$ in ≈ 50 hours (Ott *et al.*, *PSB* 2004).
  - Work-optimal Parallel Algorithm (Nikolova *et al.*, *HiPC* 2009).

- Heuristic Structure Learning
  - Serial: $n = 5000$ in ≈ 13 days (Tsamardinos *et al.*, *Mach. Learn.* 2006)
  - Genome-scale: 13,731 human gene network estimated by 50,000 random subnetworks of size 1,000 each (Tamada *et al. TCBB* 2011)

Our Heuristic Parallel Algorithm

1. Conservatively estimate *candidate parents* set $CP(X)$ for each X
    - Use pairwise mutual information (Zola *et al. TPDS* 2010)
    - Symmetric: $Y \in CP(X) \Rightarrow X \in CP(Y)$
2. Compute *optimal parents* sets (OPs) from CPs using exact method
    - Directly compute OPs from small CPs ($|CP(X)| \leq t$)
    - Reduce large CPs by using

      $$CP(Y) \leftarrow CP(Y) \setminus \{X \in CP(Y) \mid Y \in OP(X)\}$$

    - Select top t correlations for still large CP sets
    - Directly compute OPs from the now small CPs
3. Detect and break cycles

(Nikolova *et al. SC 2002*)

Our Heuristic Parallel Algorithm

1. Conservatively estimate *candidate parents* set $CP(X)$ for each X
    - Use pairwise mutual information (Zola *et al. TPDS* 2010)
    - Symmetric: $Y \in CP(X) \Rightarrow X \in CP(Y)$
2. Compute *optimal parents* sets (OPs) from CPs using exact method
    - Directly compute OPs from small CPs ($|CP(X)| \leq t$)
    - Reduce large CPs by using

$$CP(Y) \leftarrow CP(Y) \setminus \{X \in CP(Y) \mid Y \in OP(X)\}$$

    - Select top t correlations for still large CP sets
    - Directly compute OPs from the now small CPs
3. Detect and break cycles

(Nikolova *et al. SC 2002*)

Key Ideas

- Combine the precision of *Optimal Learning* with scalability of *Heuristic Learning*.
- Push limit on t using massive parallelism.

Proposed Hypercube Representation

- Compute $CP(X_i) \rightarrow OP(X_i)$.

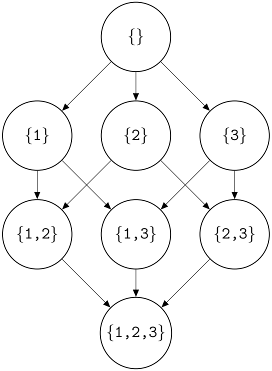
$$OP(X_i) = \underset{A \subseteq CP(X_i)}{\arg\max} \quad s\left(X_i, A\right)$$

Proposed Hypercube Representation

- Compute $CP(X_i) \to OP(X_i)$.

$$OP(X_i) = \underset{A \subseteq CP(X_i)}{\arg\max} \; s\left(X_i, A\right)$$

- But, more efficient to compute $s(X_i, A)$ from $s(X_i, B)$ where $B \subset A$.



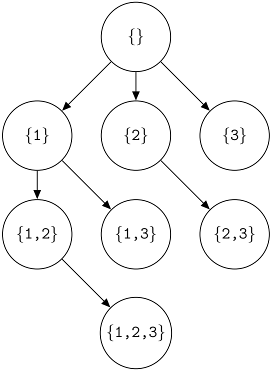
Reusing Computations

- Compute $CP(X_i) \rightarrow OP(X_i)$.

$$OP(X_i) = \underset{A \subseteq CP(X_i)}{\arg\max} \; s\left(X_i, A\right)$$

- But, more efficient to compute $s(X_i, A)$ from $s(X_i, B)$ where $B \subset A$.

- Depth First traversal to cap memory usage.



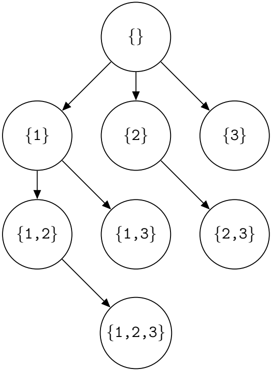
Reusing Computations

- Compute $CP(X_i) \rightarrow OP(X_i)$.

$$OP(X_i) = \underset{A \subseteq CP(X_i)}{\arg\max} \; s\left(X_i, A\right)$$

- But, more efficient to compute $s(X_i, A)$ from $s(X_i, B)$ where $B \subset A$.

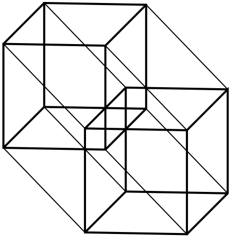
- Depth First traversal to cap memory usage.



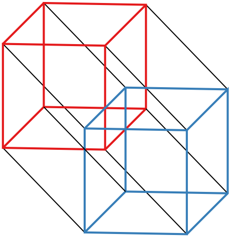
Challenges

1. Available parallelism limited by number of genes.
2. Workload varies exponentially.

Work Decomposition

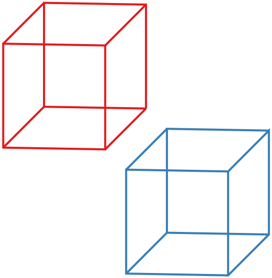


Work Decomposition



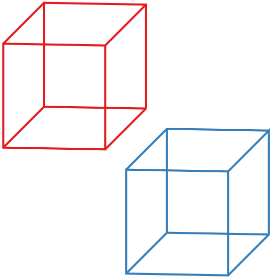
- Maximum unit of work set as *r*-dimensional hypercube.

Work Decomposition



- Maximum unit of work set as r-dimensional hypercube.
- Larger Hypercubes are split into r-dimensional sub-hypercubes.

Work Decomposition



- Maximum unit of work set as *r*-dimensional hypercube.
- Larger Hypercubes are split into *r*-dimensional sub-hypercubes.
- Direct access to subhypercube facilitated by computing the root.

Key Idea

Significantly increases parallelism with negligible compromise on reuse.

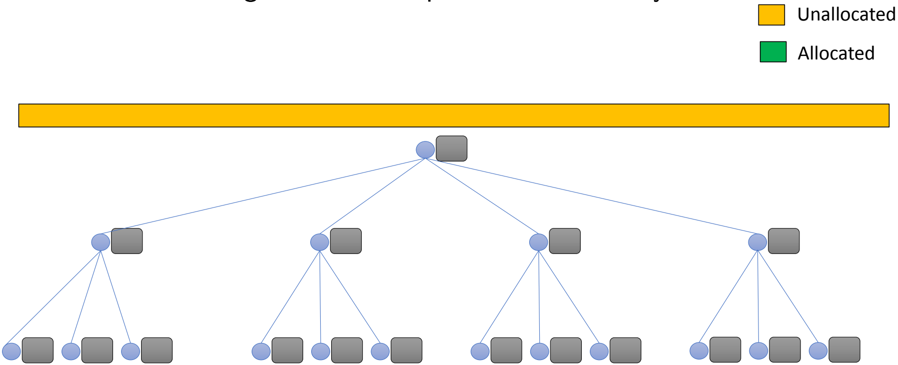
Work Distribution and Load Balancing

- Variable sized loads even when hypercube sizes are same.

Work Distribution and Load Balancing

- Variable sized loads even when hypercube sizes are same.
- Dynamic Scheduling over a processor tree.

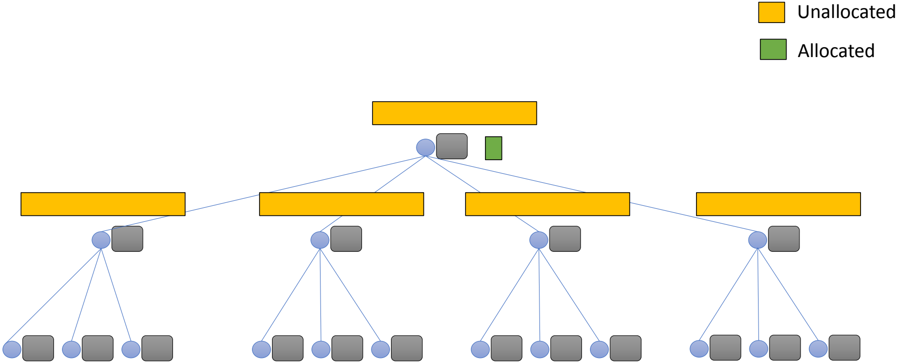
Arrangement of compute nodes as *k*-ary tree



Work Distribution and Load Balancing

- Variable sized loads even when hypercube sizes are same.
- Dynamic Scheduling over a processor tree.

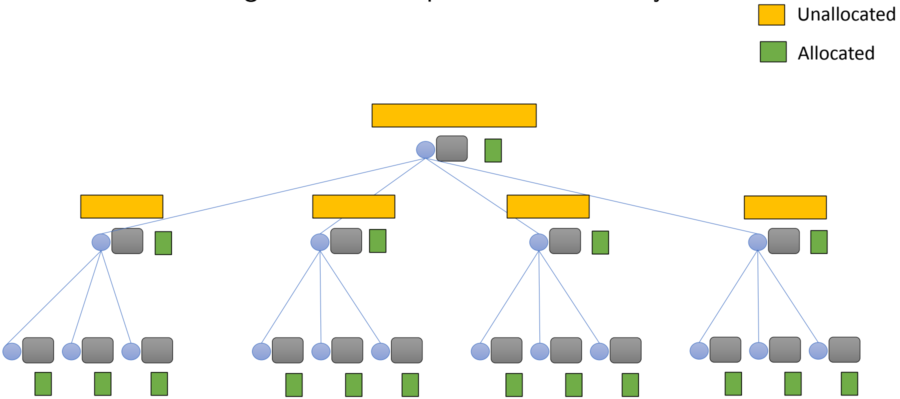
Arrangement of compute nodes as *k*-ary tree



Work Distribution and Load Balancing

- Variable sized loads even when hypercube sizes are same.
- Dynamic Scheduling over a processor tree.

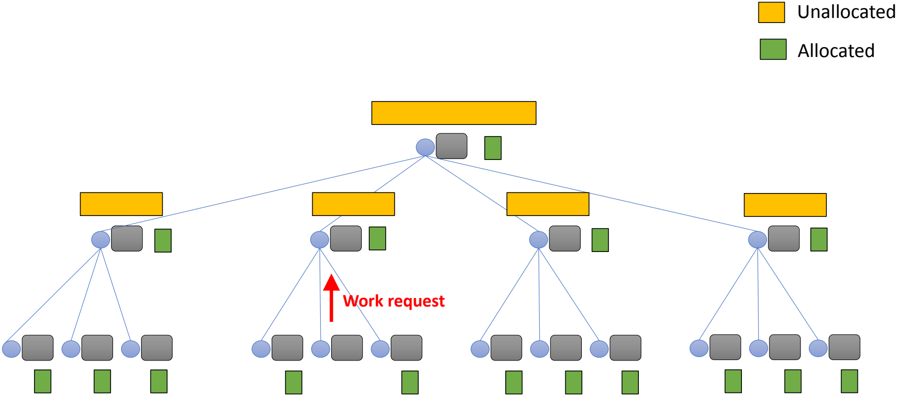
Arrangement of compute nodes as *k*-ary tree



Work Distribution and Load Balancing

- Variable sized loads even when hypercube sizes are same.
- Dynamic Scheduling over a processor tree.

Arrangement of compute nodes as *k*-ary tree



(Pamnany *et al. ISC 2015*)

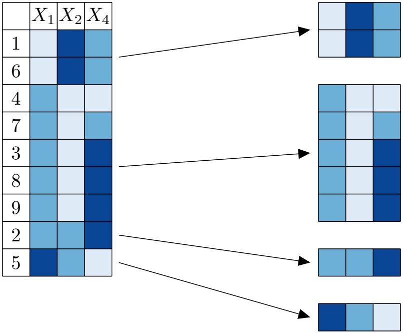
Score Computation

To compute $s(X_4, \{X_1, X_2\})$, estimate $\tilde{P}(X_4 | \{X_1, X_2\})$.

	X_1	X_2	X_4
1	░	▓	▒
2	▒	▒	▓
3	▒	░	▓
4	▒	░	░
5	▓	▒	░
6	░	▓	▒
7	▒	░	▒
8	▒	░	▓
9	▒	░	▓

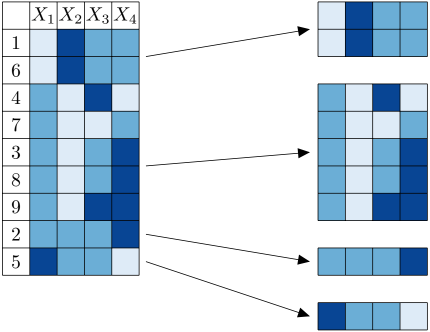
Score Computation

To compute $s(X_4, \{X_1, X_2\})$, estimate $\tilde{P}(X_4 | \{X_1, X_2\})$.



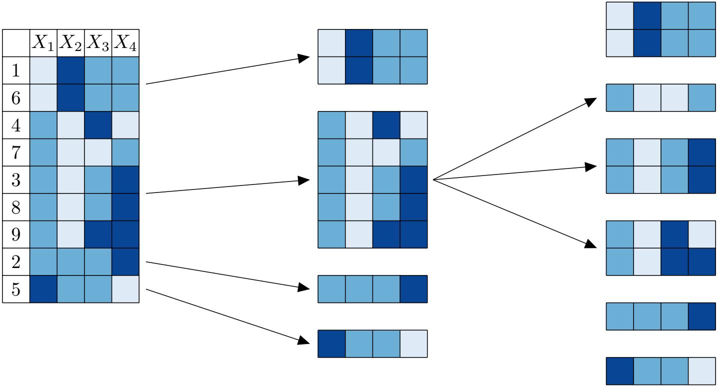
Score Computation

To compute $s(X_4, \{X_1, X_2, X_3\})$, estimate $\tilde{P}(X_4 | \{X_1, X_2, X_3\})$.



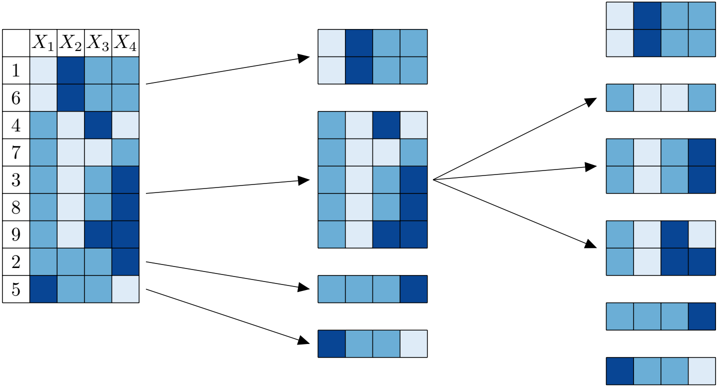
Score Computation

To compute $s(X_4, \{X_1, X_2, X_3\})$, estimate $\tilde{P}(X_4 | \{X_1, X_2, X_3\})$.



Score Computation

To compute $s(X_4, \{X_1, X_2, X_3\})$, estimate $\tilde{P}(X_4 | \{X_1, X_2, X_3\})$.



Key Idea

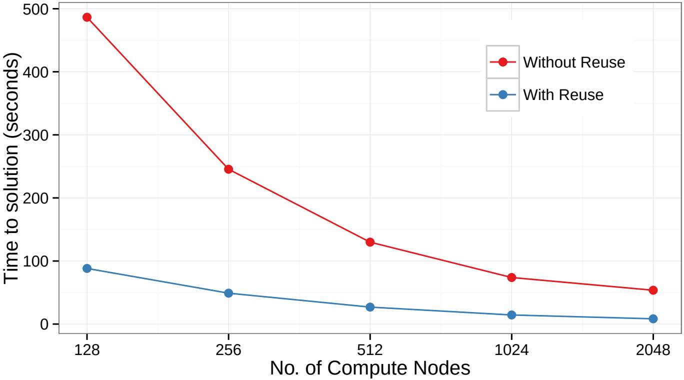
Vectorization: Score function dominates execution time.

Target Supercomputers

- Tianhe-2, National University of Defense Technology, Changsha.
- Stampede, Texas Advanced Computing Center, Austin.

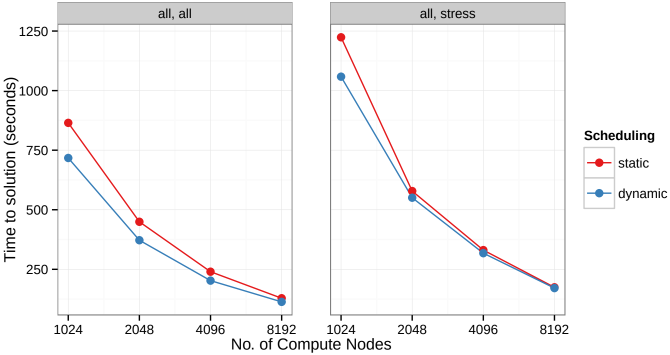
	Node configuration	
	Tianhe-2 (54.9 PF)	Stampede (8.5 PF)
CPU	Intel Xeon E5-2600	Intel Xeon E5-2680
CPU Frequency	2.2 GHz	2.7 GHz
No. of CPUs	2	2
DRAM	64 GB	32 GB
Coprocessors	Intel Xeon Phi 31 S1P	Intel Xeon Phi SE10P
Coprocessors frequency	1.09 GHz	1.09 GHz
No. of Coprocessors	3	1
Coprocessor Memory	8 GB	8 GB
Cores per node	192 $(2 \times 12 + 3 \times 56)$	76 $(2 \times 8 + 60)$
Threads per node	696	256

Performance Benefit of Reuse



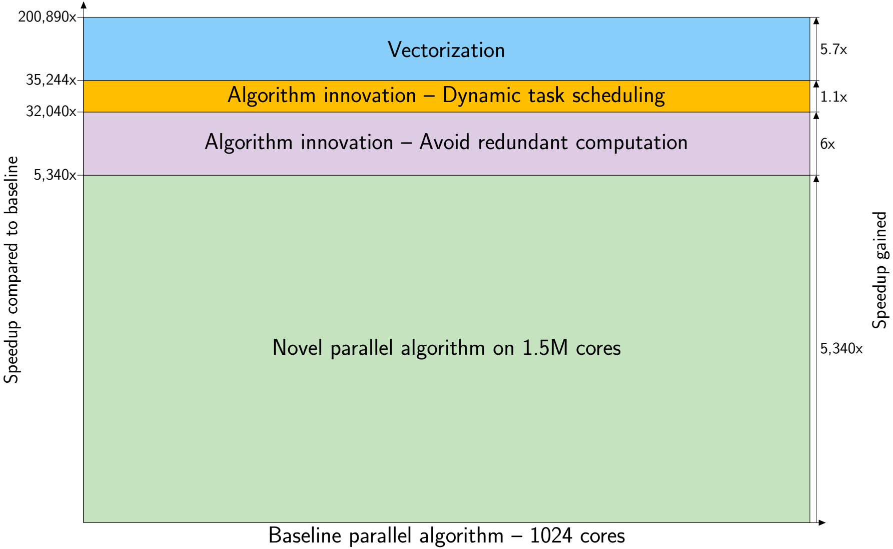
- 4.8-6.4x Speedup due to reuse of computation.

Strong Scaling on Tianhe-2

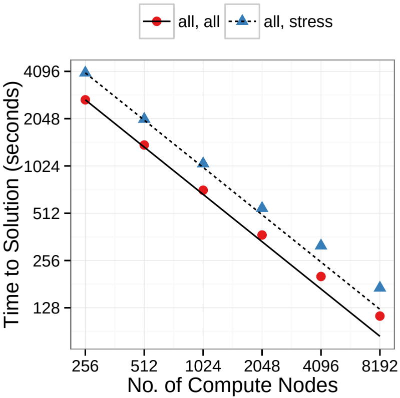


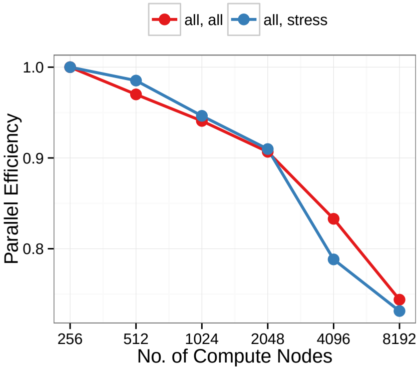
- 7-18 % improvement by dynamic scheduling in all cases except – 8192 nodes for the ⟨all,stress⟩ dataset

Where does the speedup come from?



Parallel Efficiency





Full Application Runs

	all,all	seedling,all	root,all	all,stress
Genes (n)	$14,330$	$13,590$	$15,236$	$15,216$
Experiments (m)	$11,760$	$4,933$	$1,939$	$2,476$
Genes with $\lvert CP\rvert \leq t$	$13,922$	$13,086$	$14,340$	$13,293$
Genes with reduced CP	408	504	896	$1,923$
Genes with truncated CP	241	15	293	$1,376$
Run-time on STP (sec)	$1,947$	269	501	$2,352$
Run-time on TH-2 (sec)	113.4			171.2
Billion scores/s (TH-2)	12.3			42.9

(Misra *et al. SC 2014*, best paper finalist)

GeNA — Gene Network Analyzer

Adopted from page rank (Haveliwala, *IEEE Trans. Knowledge Data Engg. 2003*)

Assign transition probabilities:

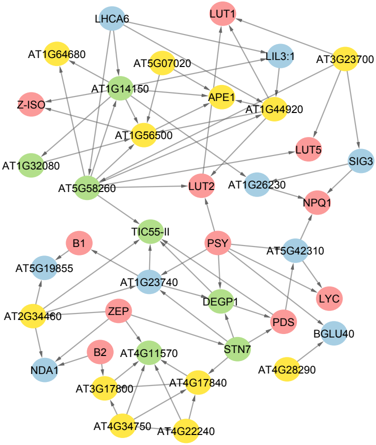
$$\omega(i,j) = \frac{D[i,j]}{\sum_{k:(i,k)\in N} D[i,k]}$$

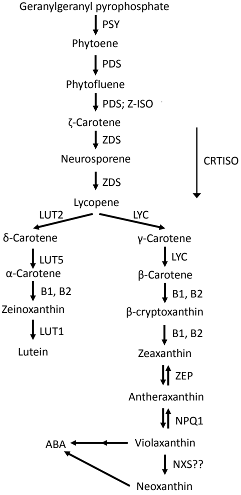
Compute ranks:

$$R(j)^{(k+1)} = (1-\alpha)\cdot\left(\sum_{i:(i,j)\in N} \omega(i,j)\cdot R(i)^{(k)}\right) + \alpha\cdot p(j)$$

Return connected subnetwork with high ranked genes.

Carotenoid Subnetwork and Pathway

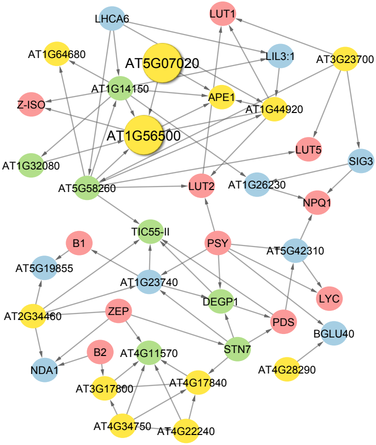


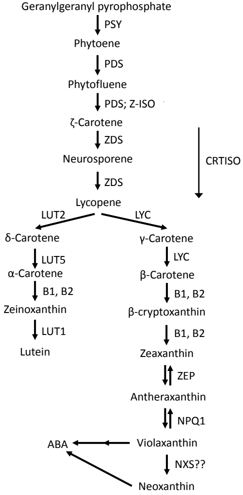


Geranylgeranyl pyrophosphate
PSY
Phytoene
PDS
Phytofluene
PDS; Z-ISO
ζ-Carotene
ZDS
Neurosporene
CRTISO
ZDS
Lycopene
LUT2
LYC
δ-Carotene
γ-Carotene
LUT5
LYC
α-Carotene
β-Carotene
B1, B2
B1, B2
Zeinoxanthin
β-cryptoxanthin
LUT1
B1, B2
Lutein
Zeaxanthin
ZEP
Antheraxanthin
NPQ1
ABA
Violaxanthin
NXS??
Neoxanthin

Pink – Seed genes; Green – In associated pathways; Blue – Have related GO terms; Yellow – No known function

Carotenoid Subnetwork and Pathway

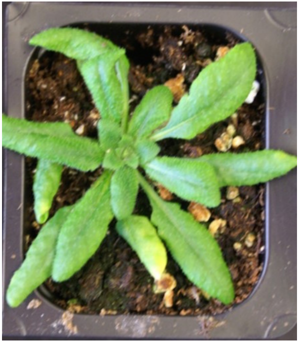




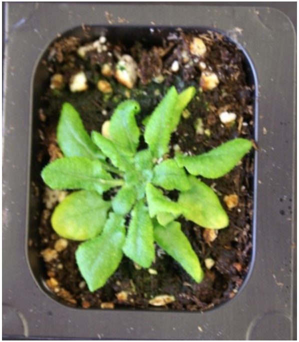
Geranylgeranyl pyrophosphate
PSY
Phytoene
PDS
Phytofluene
PDS; Z-ISO
ζ-Carotene
ZDS
Neurosporene
CRTISO
ZDS
Lycopene
LUT2
LYC
δ-Carotene
γ-Carotene
LUT5
LYC
α-Carotene
β-Carotene
B1, B2
B1, B2
Zeinoxanthin
β-cryptoxanthin
LUT1
B1, B2
Lutein
Zeaxanthin
ZEP
Antheraxanthin
NPQ1
ABA
Violaxanthin
NXS??
Neoxanthin

Pink – Seed genes; Green – In associated pathways; Blue – Have related GO terms; Yellow – No known function

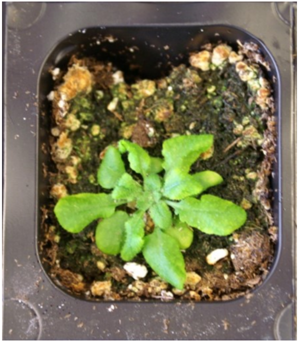
Arabidopsis Knockout Mutants



Wild Type

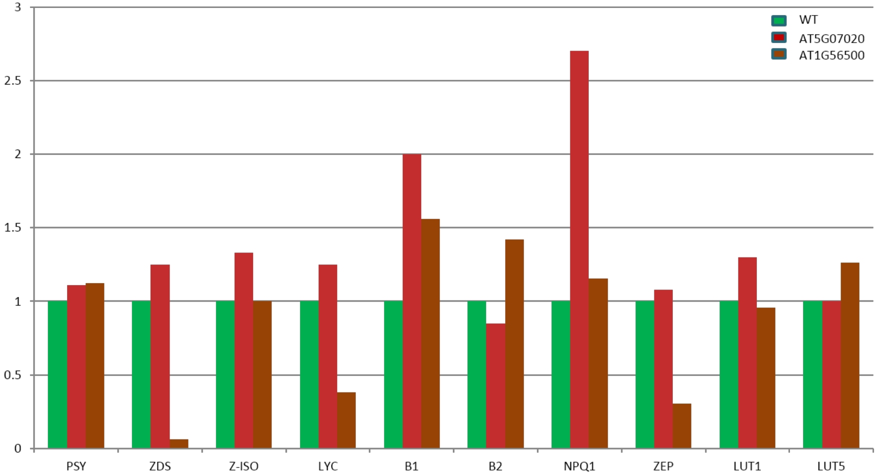


AT1G56500



AT5G07020

Experimental Validation



Network Driven Biology Research

- M. Aluru, J. Zola, D. Nettleton and S. Aluru, “Reverse engineering and analysis of large genome-scale gene networks,” *Nucleic Acids Research*, Vol. 41, No. 1, pp. e24, doi: 10.1093/nar/gks904, 2013.
- H. Guo, L. Li, M. Aluru, S. Aluru and Y. Yin, “Mechanisms and networks for brassinosteroid regulated gene expression,” *Current Opinion in Plant Biology*, Vol. 16, 9 pages, 2013.
- X. Yu, L. Li, J. Zola, M. Aluru, H. Ye, A. Foudree, H. Guo, S. Anderson, S. Aluru, P. Liu, S. Rodermel and Y. Yin, “A brassinosteroid transcriptional network revealed by genome-wide identification of BES1 target genes in Arabidopsis thaliana,” *The Plant Journal*, Vol. 65, No. 4, pp. 634-646, 2011.

Acknowledgements

Group Members:

- Sriram Chockalingam
- Wasim Mohammed
- Olga Nikolova
- Jaroslaw Zola

Collaborators:

- Maneesha Aluru (Bio)
- Yanhai Yin (Bio)
- Daniel Nettleton (Stat)
- Sanchit Misra (Intel)
- Kiran Pamnany (Intel)

Funding

Research supported by NSF CCF-0811804, IOS-1257631, and Intel PCC.