バイオスーパーコンピューティング研究会 ウィンタースクール２０１４
２０１４年１月２４日 @ 熱川ハイツ

公開版につき実際に使用したものを編集したものです

バイオインフォマティクスにおける スーパーコンピューティングの過去・現在・未来 （と言う名の自分史と将来への希望）

玉田 嘉紀

東京大学 大学院 情報理工学系研究科

東京大学 THE UNIVERSITY OF TOKYO

Human Genome Center
Institute of Medical Science, University of Tokyo

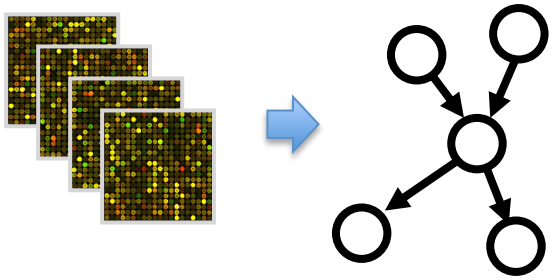
RIKEN

SCLS SUPERCOMPUTATIONAL LIFE SCIENCE

新学術領域研究
「システムがん」

研究テーマ

- 遺伝子ネットワーク推定手法の研究
  - 観測データからその背景となる仕組みを統計科学的・情報科学的に推測する「生命のリバースエンジニアリング」
  - 大規模データ解析



- ベイジアンネットワーク等の構造学習
  - スーパーコンピュータを用いた超高速・超高並列アルゴリズムおよび大規模計算
  - 次世代スパコン「京」

K computer

本日のテーマ

- バイオインフォマティクスにおけるスーパーコンピューティングの過去・現在・未来
  - 「スパコン」という視点で自分の過去を振り返って未来について語りたい.

私的 ~~HPC~~ ヒストリー (1)

- B３～M2の頃（1999～2002年）
  - 東大医科研ヒトゲノム解析センター
    - Sun Ultra Enterprise 10000
      - 64CPUs, 16GB mem, 818GB disk
    - SGI Origin2000
      - 128CPUs, 24GB mem, 636 GB disk
    - ジョブ管理なし
    - プロセス大量投入 or pthread
    - 共通文字列探索
    - データマイニング
    - スパコン？

Human Genome Center
Institute of Medical Science, University of Tokyo

当時

- HPC という言葉はまったく知らなかった.
- データマイニング用並列処理計算フレームワークを PSB の HPC セッションに投稿
  - あたりまえだがリジェクト.

私的 ~~HPC~~ ヒストリー (2)

- D1〜 (2003〜)
  - 東大医科研ヒトゲノム解析センターのスパコンがリニューアル
    - Sun SF15K
      - 96 CPUs, 288GB mem, 10TB disk
    - SGI Origin 3900T
      - 512 CPUs, 512 GB mem, 15TB disk
    - Xeon PC クラスタ
      - 128 CPUs, 256 GB mem, 3 TB disk
    - SF15K & PCクラスタは Sun Grid Engine でジョブ管理.
  - 自分の研究は遺伝子ネットワークへ

私的 ~~HPC~~ ヒストリー (3)

- 2006年〜 2007年
  - ベンチャー企業でスパコン環境＆アプリ開発
    - mini HGC システム的な小規模 Xeon PC クラスタ

私的 HPC ヒストリー (4)

- 2008年
  - 「京」アプリ研究開発プロジェクトに参加
- 2009年〜
  - またヒトゲノム解析センターのスパコンがリニューアル (Shirokane1)
    - Sun Blade (Xeon Linux PC クラスター)
      - 700 ノード〜, Lustre ファイルシステム (1.6PB)
    - Sun Grid Engine
  - 理研RICC
    - あまりにも使い勝手が異なり驚く
      - Xeon 1024ノード（8192コア）

現在

- HGC Shirokane2 （1も併用中）
  - AMD Opteron 494 ノード (16,128コア)
    - mem: 32GB/node
  - Xeon 12 ノード (144 コア)
    - mem: 144GB/node
  - Disk: 4PB
  - Top500 (June 2013)で 468 位.
- 京
- TSUBAME

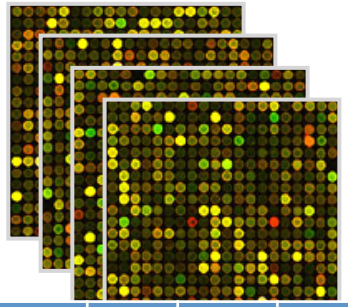
- ちょっとここで研究の話を.

Gene Network Estimation

Gene expression can be measured at mRNA level.

mRNA

DNA chip (microarray)

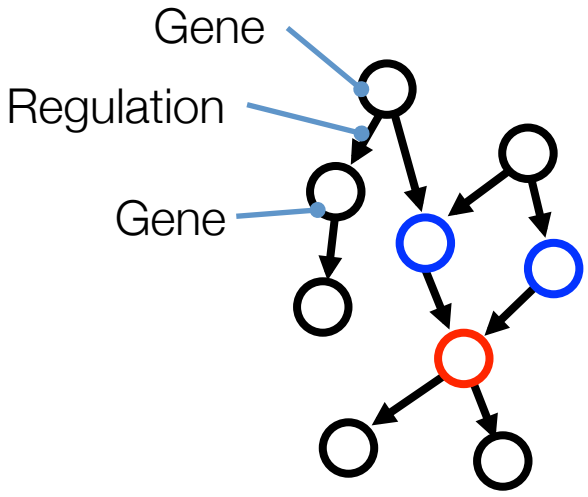


Gene	KD1	KD2	KD3	...
Gene 1	1.45	-1.54	1.23	...
Gene 2	3.21	-2.1	1.44	...
...	...	...	...	...

Gene Expression Data

Estimation

Reverse engineering of the background system



Gene Network

Discovery of a new marker.

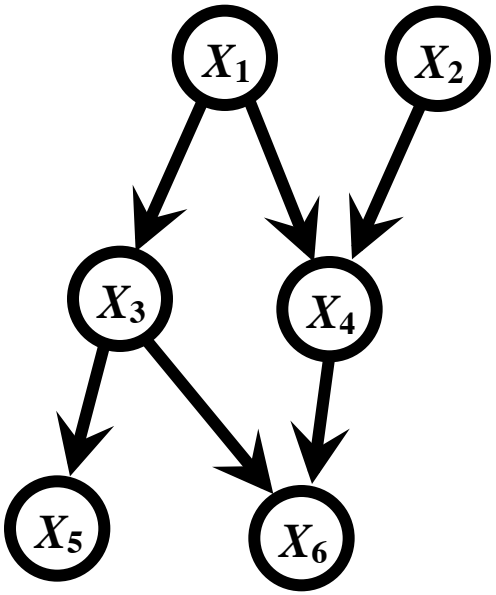
Predict drug side effect.

Comparison of gene regulations between normal versus cancer cells.

遺伝子ネットワークモデル

ベイジアンネットワーク

Node = Gene



Directed Edge = Regulatory Relationships

$X_1, X_2, \ldots, X_p$: Random variables corresponds to p genes (transcripts).

Consider the joint probability of these p variables.

$$\Pr(X_1, X_2, \ldots, X_p)$$

Joint Probability by a DAG (Directed Acyclic Graph)

$$\Pr(X_1, X_2, \ldots, X_6)$$

Product of local probabilities

$$= \Pr(X_1) \times \Pr(X_2) \times \Pr(X_3 \mid X_1) \times \Pr(X_4 \mid X_1, X_2)$$
$$\times \Pr(X_5 \mid X_3) \times \Pr(X_6 \mid X_3, X_4)$$

DAG = Directed graph with *no loops*.

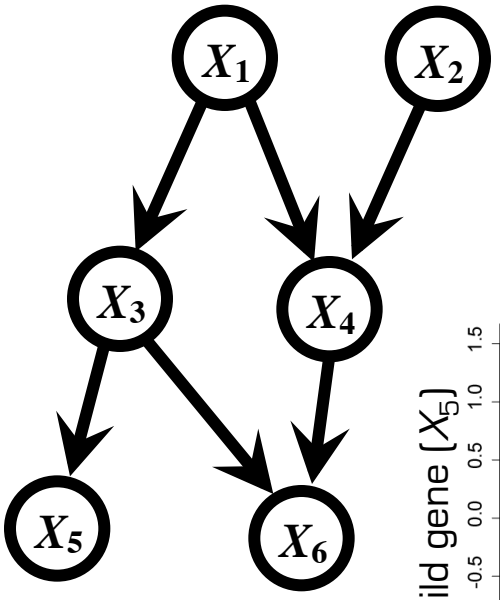
Imoto et al. (2002)

Nonparametric Bayesian Network Model

We use the **nonparametric Bayesian network** as models for gene networks

Joint density function by a DAG (Directed Acyclic Graph)

Node = Gene

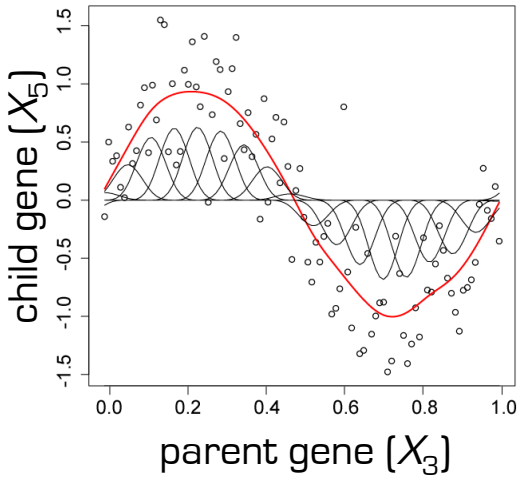


Directed Edge = Regulatory Relationships

$$p(X_1, X_2, \ldots, X_6) = p_1(X_1)\,p_2(X_2)\,p_3(X_3 \mid X_1)\cdots p_6(X_6 \mid X_3, X_4)$$

Nonparametric regression by B-spline

e.g. $x_5 = m(x_3) + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2)$



General Form:

$$x_{ij} = m_{j1}(x_{i1}^{(j)}) + \ldots + m_{jq_j}(x_{iq_j}^{(j)}) + \varepsilon_{ij}$$

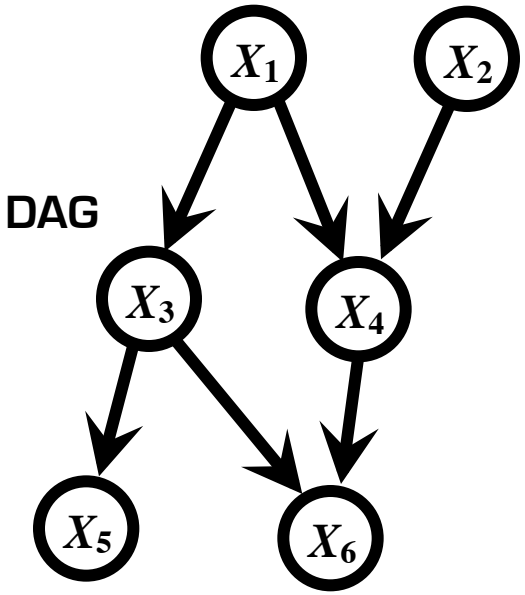
$$\varepsilon_{ij} \sim N(0, \sigma_j^2)$$

$$m_{jk}(x_{ik}^{(j)}) = \sum_{l=1}^{M_{jk}} \gamma_{lk} b_{lk}^{(j)}(x_{ik}^{(j)})$$

coefficients — **B-spline curves**

ベイジアンネットワーク 2

DAG [Directed Acyclic Graph: 非巡回有向グラフ] による同時確率



$$P(X_1, X_2, X_3, X_4, X_5, X_6)$$
$$= P(X_1)P(X_2)P(X_3 \mid X_1)P(X_4 \mid X_1, X_2)P(X_5 \mid X_3)P(X_6 \mid X_3, X_4)$$
$$= \prod_{j=1}^{n} P(X_j \mid Pa(X_j))$$

n : the number of nodes.

$Pa(X_j)$: parents of X_j

Network score = Posterior Probability

$$P(G \mid \boldsymbol{X}) = \frac{P(G)P(\boldsymbol{X} \mid G)}{P(X)}$$
$$\propto P(G)P(\boldsymbol{X} \mid G)$$

G: ネットワーク構造
$\boldsymbol{X}$: 観測データ

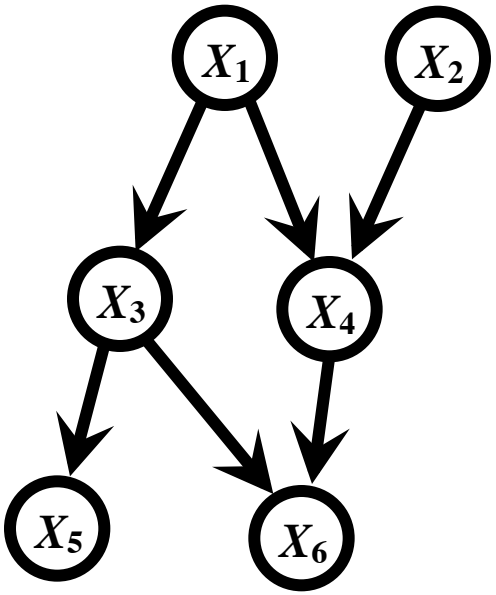
事後確率の高いネットワーク構造＝良いネットワーク構造

問題：スコアの最も良い **DAG** 構造を探したい.

Problem Definition

Posterior probability

$$\pi(G \mid \boldsymbol{X}) \propto \pi(G) \int \prod_{i=1}^{N} \prod_{j=1}^{n} f(x_{ij} \mid \boldsymbol{pa}_{ij}^{G}, \boldsymbol{\theta}_G) \pi(\boldsymbol{\theta}_G \mid \boldsymbol{\lambda}) d\boldsymbol{\theta}_G$$



Network score

$$\mathrm{BNRC}(G, \boldsymbol{X}) = -2 \log \left\{ \pi(G) \int \prod_{i=1}^{N} \prod_{j=1}^{n} f(x_{ij} \mid \boldsymbol{pa}_{ij}^{G}, \boldsymbol{\theta}_G) \pi(\boldsymbol{\theta}_G \mid \boldsymbol{\lambda}) d\boldsymbol{\theta}_G \right\}$$

$$= \sum_{j=1}^{n} s(X_j, Pa^G(X_j), \boldsymbol{X})$$

$$s : V \times 2^V \times \mathrm{R}^{n,N} \rightarrow \mathrm{R} \qquad |V| = n \text{ : nodes}$$

Network estimation (Definition)

$$\hat{G} = \arg\min_{G} \sum_{j=1}^{n} s(X_j, Pa^G(X_j), \boldsymbol{X})$$

subject to G is acyclic.

Score-based structure learning (search) of a Bayesian network

Searching DAGs

Nodes=1

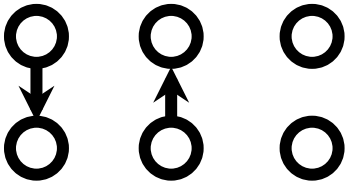
DAGs=1

Searching DAGs

Nodes=1

DAGs=1

Nodes=2



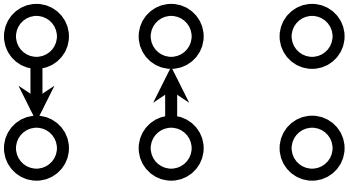
DAGs=3

Searching DAGs

Nodes=1

DAGs=1

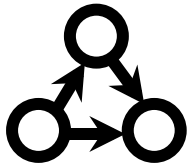
Nodes=2

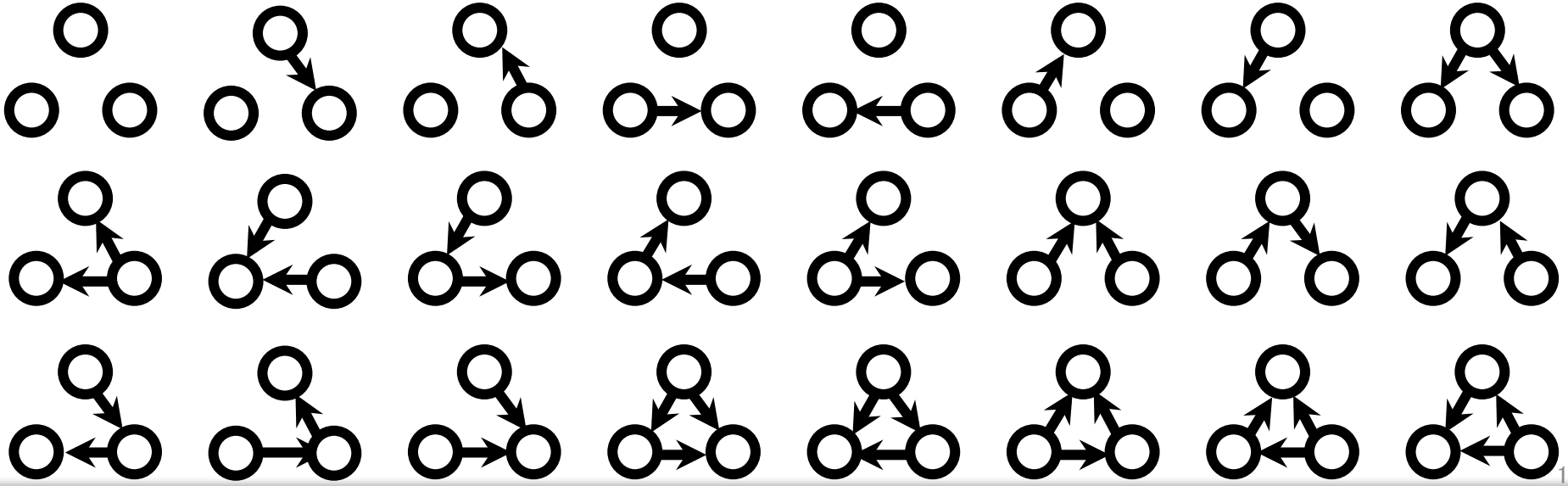


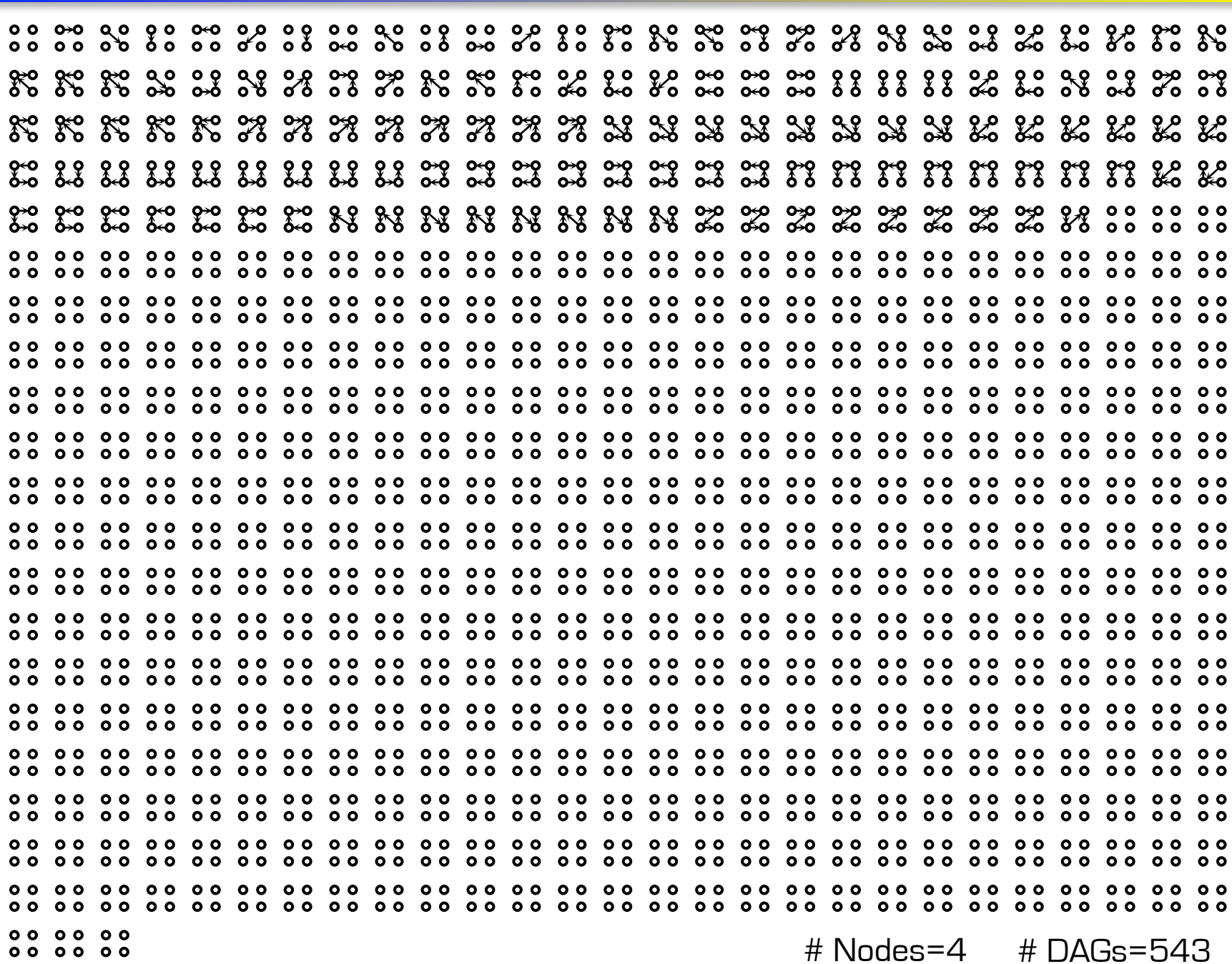
DAGs=3

Nodes=3

DAGs=25






Nodes=4 # DAGs=543

Nodes=5

DAGs = 29,281

Nodes=5

DAGs = 29,281

Nodes=6

DAGs = 3,781,503

Nodes=5

DAGs = 29,281

Nodes=6

DAGs = 3,781,503

Nodes=7

DAGs = 1,138,779,265

Nodes=5

DAGs = 29,281

Nodes=6

DAGs = 3,781,503

Nodes=7

DAGs = 1,138,779,265

Nodes=8

DAGs = 783,702,329,343

もぅﾏｼﾞ無理。。。

Difficulty in Bayesian Network Estimation

A huge number of possible DAGs : Impossible to search the optimal one.

# of nodes	# of DAGs	# of nodes	# of DAGs
1	1	16	$\approx 8.37 \times 10^{46}$
2	3	17	$\approx 6.26 \times 10^{52}$
3	25	18	$\approx 9.93 \times 10^{58}$
4	543	19	$\approx 3.32 \times 10^{65}$
5	29,281	20	$\approx 2.34 \times 10^{72}$
6	3,781,503	21	$\approx 3.46 \times 10^{79}$
7	1,138,779,265	22	$\approx 1.07 \times 10^{87}$
8	783,702,329,343	23	$\approx 6.97 \times 10^{94}$
9	1,213,442,454,842,881	24	$\approx 9.43 \times 10^{102}$
10	$\approx 4.17 \times 10^{18}$	25	$\approx 1.86 \times 10^{111}$
11	$\approx 3.15 \times 10^{22}$	...	...
12	$\approx 5.21 \times 10^{26}$	30	$\approx 2.71 \times 10^{158}$
13	$\approx 1.86 \times 10^{31}$	...	...
14	$\approx 1.43 \times 10^{36}$	...	...
15	$\approx 2.37 \times 10^{41}$	40	$\approx 1.12 \times 10^{276}$

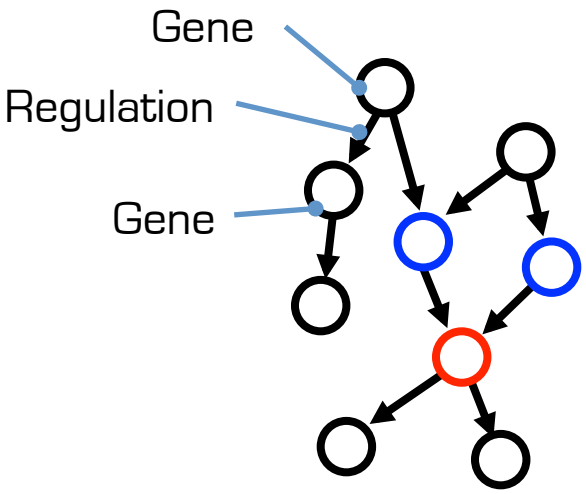
Exceeds the number of atoms in the universe

Optimal search is NP-hard. (Chickering, 1995)

ヒト遺伝子ネットワークへの応用

当初のモチベーション:

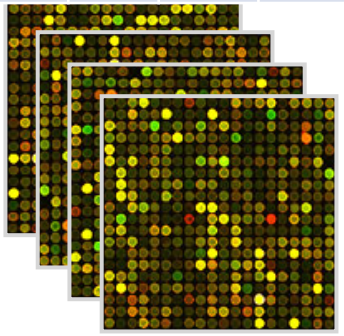
遺伝子の発現を観測したデータから遺伝子間の制御の依存関係を推定したい



Gene Network

Estimation

Gene	KD1	KD2	KD3	…
Gene 1	1.45	-1.54	1.23	…
Gene 2	3.21	-2.1	1.44	…
…	…	…	…	…



Gene Expression Data
(static/dynamic)

ヒト遺伝子：２万～３万
ヒトタンパク：～１０万

このままでは最適解探索はほぼ不可能

DAG 探索問題

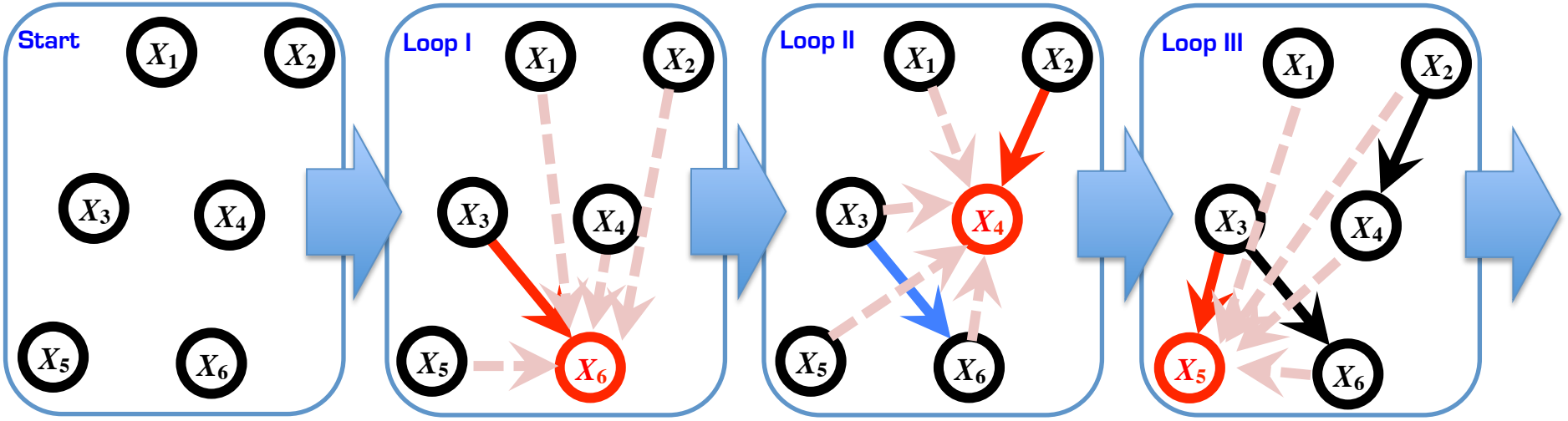
- 発見的方法 (Heuristics Algorithm)
  - Greedy Hill-Climbing Algorithm
  - Neighbor Node Sampling & Repeat Algorithm

- どうしても最適解を見つけたい
  - Optimal search algorithm by dynamic programming

Greedy Hill-Climbing Algorithm (HC)

Algorithm for searching the local optimal DAG structure

Heuristics algorithm applicable to estimate gene networks for ～ **100 genes**.



1. Begins with an empty graph.

2. Visits nodes in a random order.

3. Calculates local scores for all possible candidate parents.

4. Employs the best operation that improves the score.

Add/Delete/Reverse

5. Repeats until any operation can improve the score.

※ Need to check every time whether a cyclic path is made or not.

※ Repeat this many times, then employs the best structure because they are local optimal.

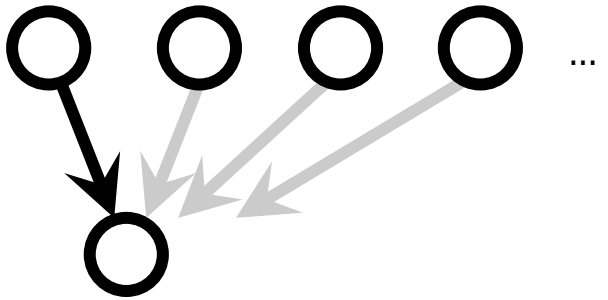
Greedy Hill-Climbing Algorithm (HC)

Still, very slow for large networks, e.g., 1,000 nodes (genes)

1. Restrict parent candidates

All the 1-to-1 scores are calculated and then we use n best scored nodes as parent candidates in the greedy algorithm.

n = 10 ~ 20



2. Restrict maximum number of parents

We allow each node to have at most m parent nodes.

m = 10 ~ 20

3. Repeat the algorithm several times

We repeat the algorithm t times and choose the best scored network as the final result of the algorithm.

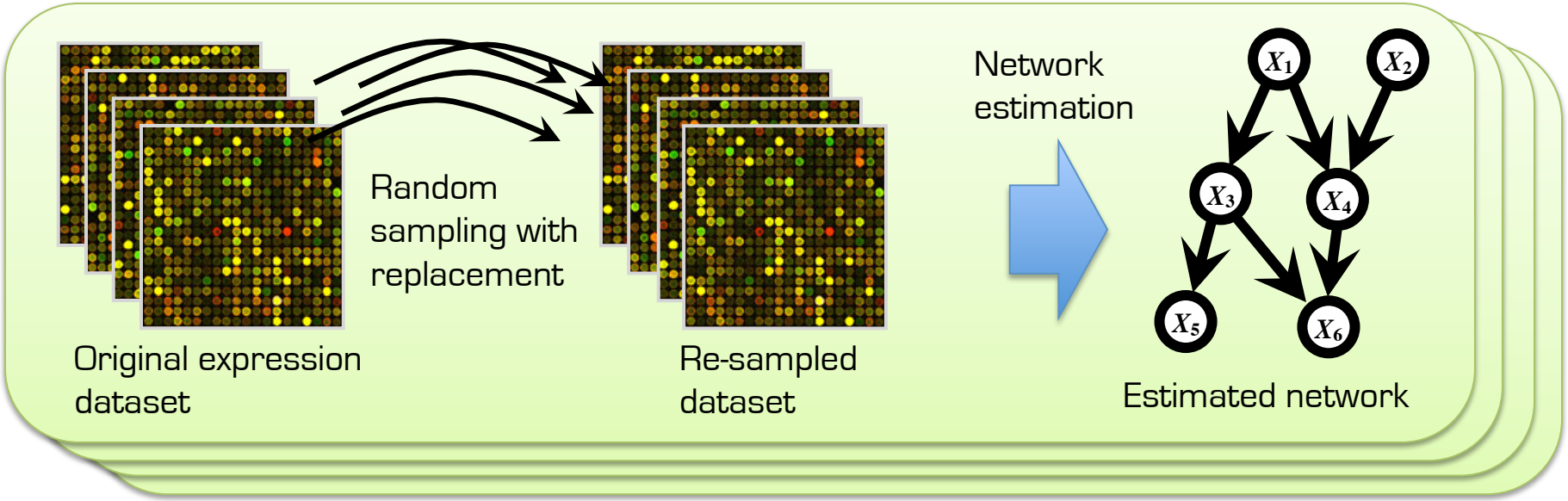
t = 10

The quality of the result is unknown...

HC + Bootstrap

~ 1000 genes

Bootstrapping is required for calculating the reliability of edges.



- Estimate networks **many times for re-sampled dataset.** (1,000 times ~)
- The final structure is determined by the frequencies of edges during the repeated estimation.
- We can perform each network estimation **independently for the re-sampled datasets in parallel.**
  - **Parallelization is easy for Bootstrap HC.**

For Much Larger Networks…

- Home sapiens
  - ~30,000 genes.
  - ~100,000 proteins.

- HC algorithm
  - ~ 1,000 genes.

We want to estimate much larger networks to apply to all the human genes!

2008 年某日

「京」で実現される計算パワーを使ってヒト全遺伝子を含む超大規模遺伝子ネットワークを推定可能なアルゴリズムを開発したい（開発せよ）.

RIKEN

ISLiM

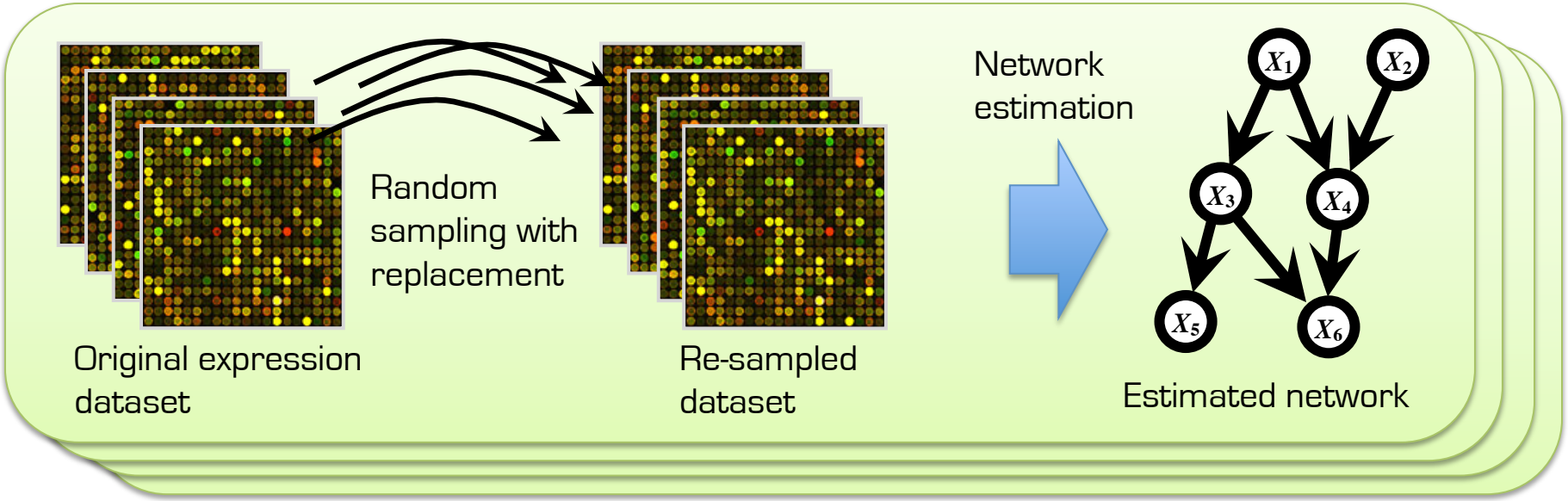
理化学研究所 次世代計算科学研究開発プログラム
データ解析融合研究開発チーム

「京」のアプリケーション開発プロジェクト

HC + Bootstrap

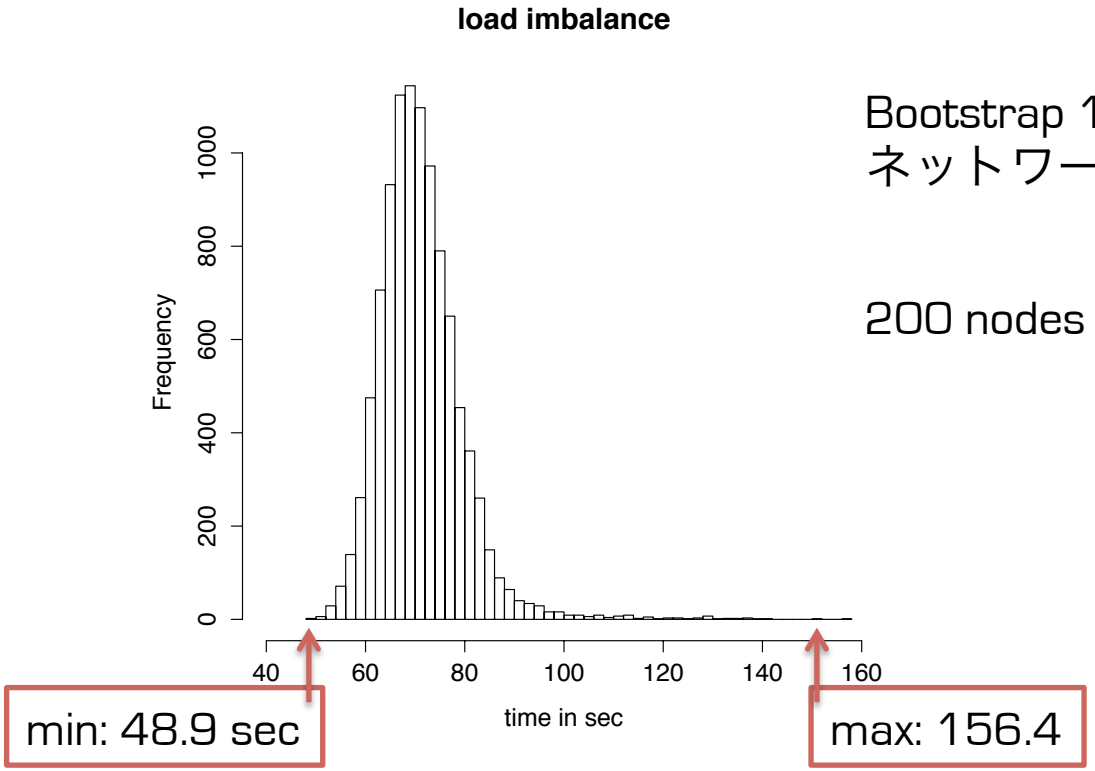
～ 1000 genes

Bootstrapping is required for calculating the reliability of edges.



- HC+Bootstrap アルゴリズムの超並列化
- 24,576 ノード（196,608 コア）で動作するソフトウェアを開発
- 従来1,000回のブートストラップ→ 1,000,000 回が現実的に.

Load Imbalance in HC Bootstrap



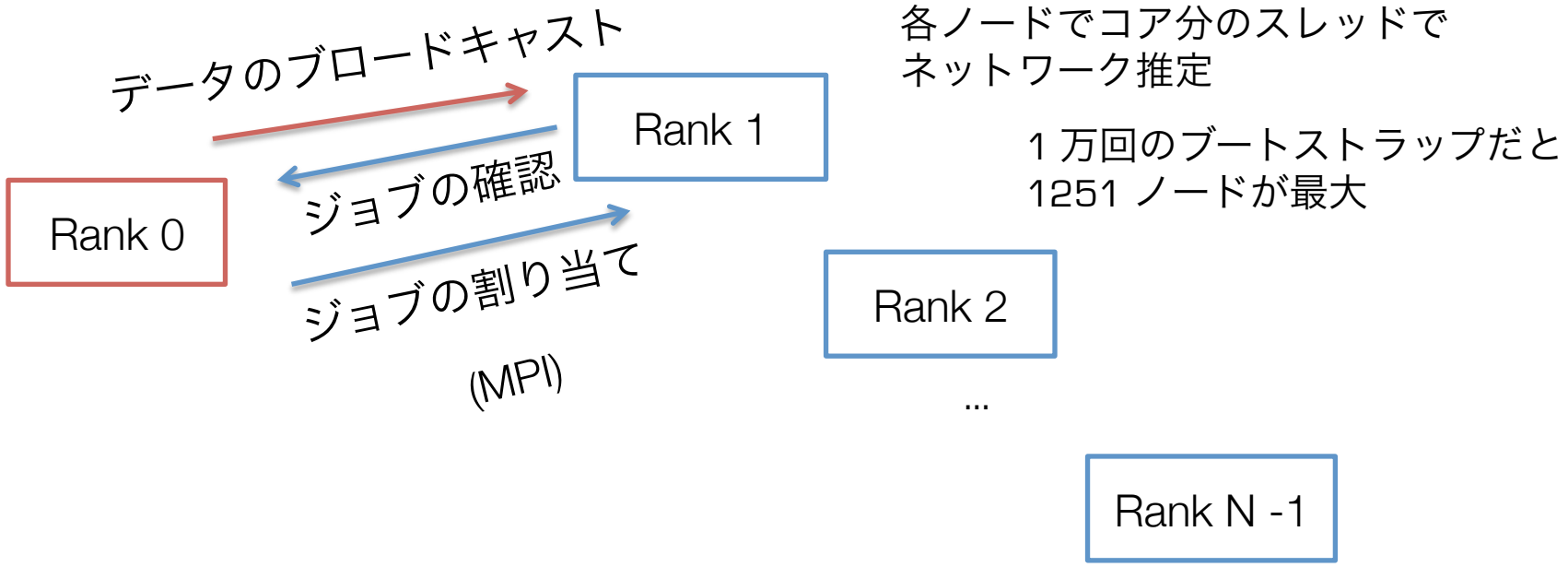
Bootstrap 1 回の
ネットワーク推定時間

200 nodes & 200 samples

- 単純に分割すると load imbalance により効率が悪い.
  - 高並列時に顕著
- 1 プロセスをジョブ割り当て専用に使用.
  - 8%弱の並列化効率の改善

京のために何をしたか 1

- 自前ジョブ割り当てツール
  - Grid Engine のアレイジョブ的なものがない.
    - bulk job? 最大 15 本？しかも12ノード以上？
    - こっちは1コアジョブを 100,000 個投げたいのに. . .



京のために何をしたか２

- ネットワークスコア計算部のハンドチューニング
  - 手動ループアンローリング
- ハイブリッド並列化
  - 最初は１ノード１スレッドで１ネットワークだったのを８スレッド８ネットワークに.
- FLOPS/PEAK
  - 2%前後 ⇒ 10%前後
- MIPS/PEAK
  - 20%弱? ⇒ 33%前後

For Much Larger Networks…

- Home sapiens
  - ~30,000 genes.
  - ~100,000 proteins.

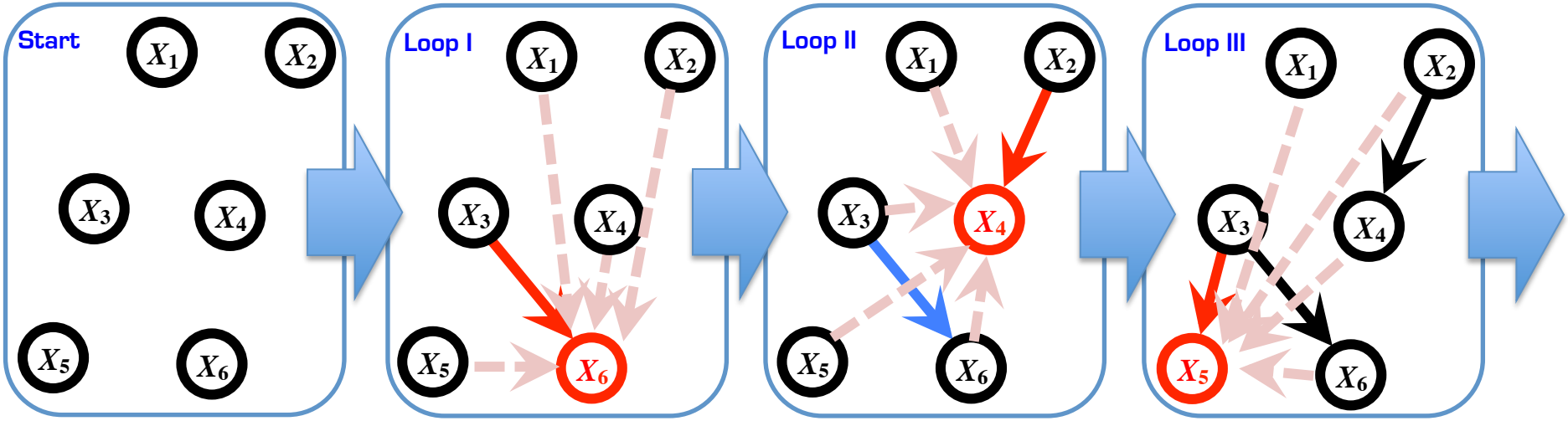
- HC algorithm
  - ~ 1,000 genes.

We want to estimate much larger networks to apply to all the human genes!

Greedy Hill-Climbing Algorithm (HC)

Algorithm for searching the local optimal DAG structure

Heuristics algorithm applicable to estimate gene networks for ～ 100 genes.



1. Begins with an empty graph.

2. Visits nodes in a random order.

4. Employs the best operation that improves the score.

Add / Delete / Reverse

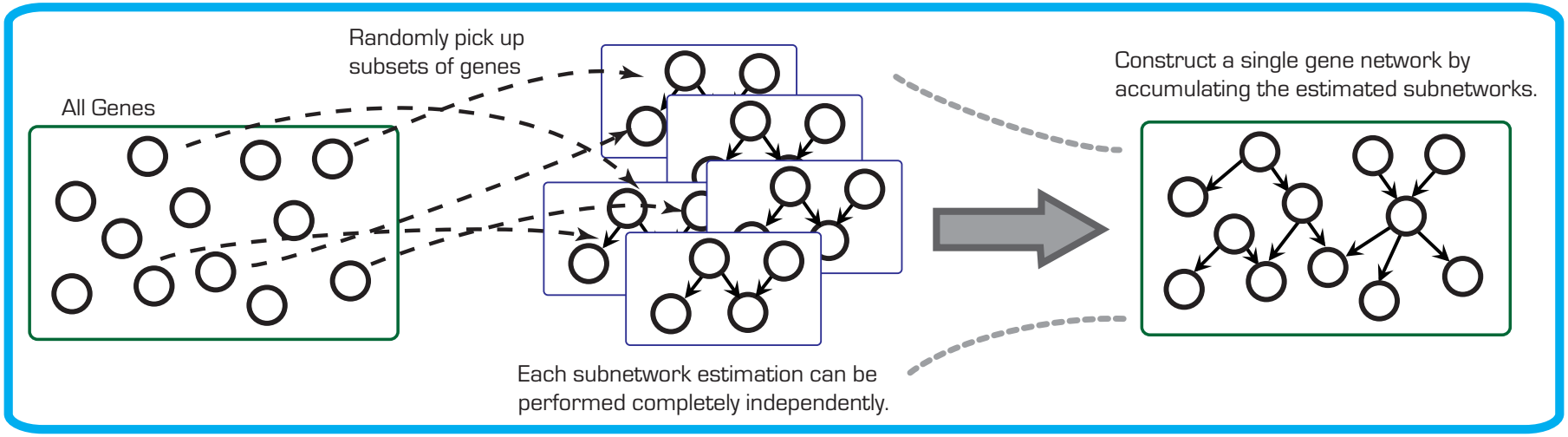
HCは逐次アルゴリズム＝並列化が難しい

※ Need to check every time whether a cyclic path is made or not.

※ Repeat this many times, then employs the best structure because they are local optimal.

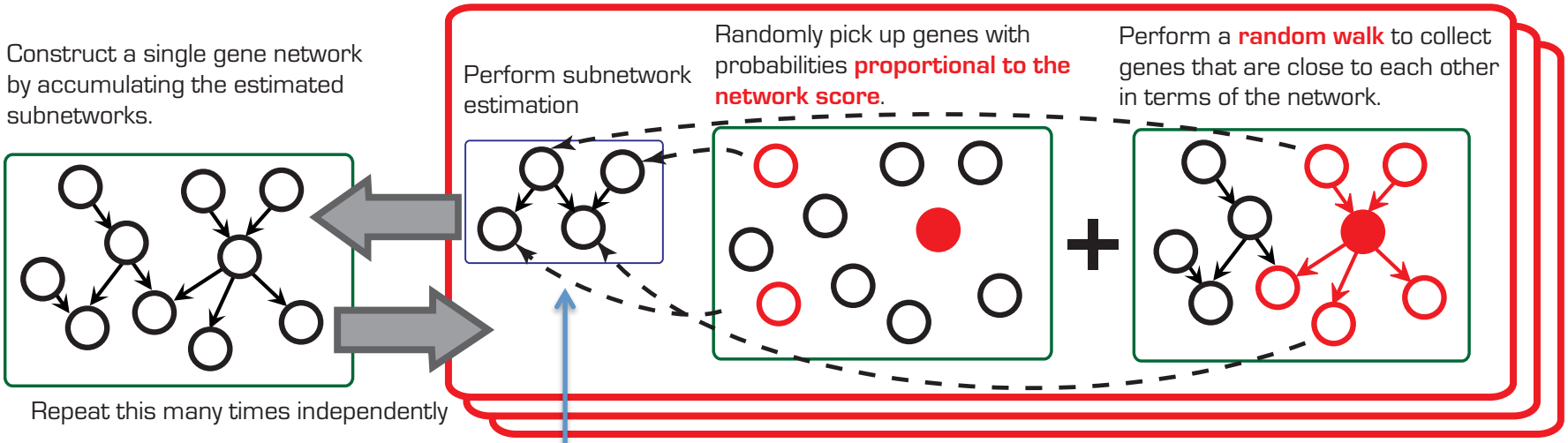
Naïve Algorithm: RSR Algorithm

Random Sampling & Repeat (RSR) Algorithm



- Estimate **subnetworks repeatedly** for subset of genes selected at random.
- Each subnetwork estimation can be performed **completely independently**.

Neighbor Node Sampling & Repeat Algorithm



Sub-network estimation is performed by the HC algorithm.

To determine the final network structure, NNSR uses a threshold and collects edges whose estimation frequencies during the sub network estimation are greater than the threshold.

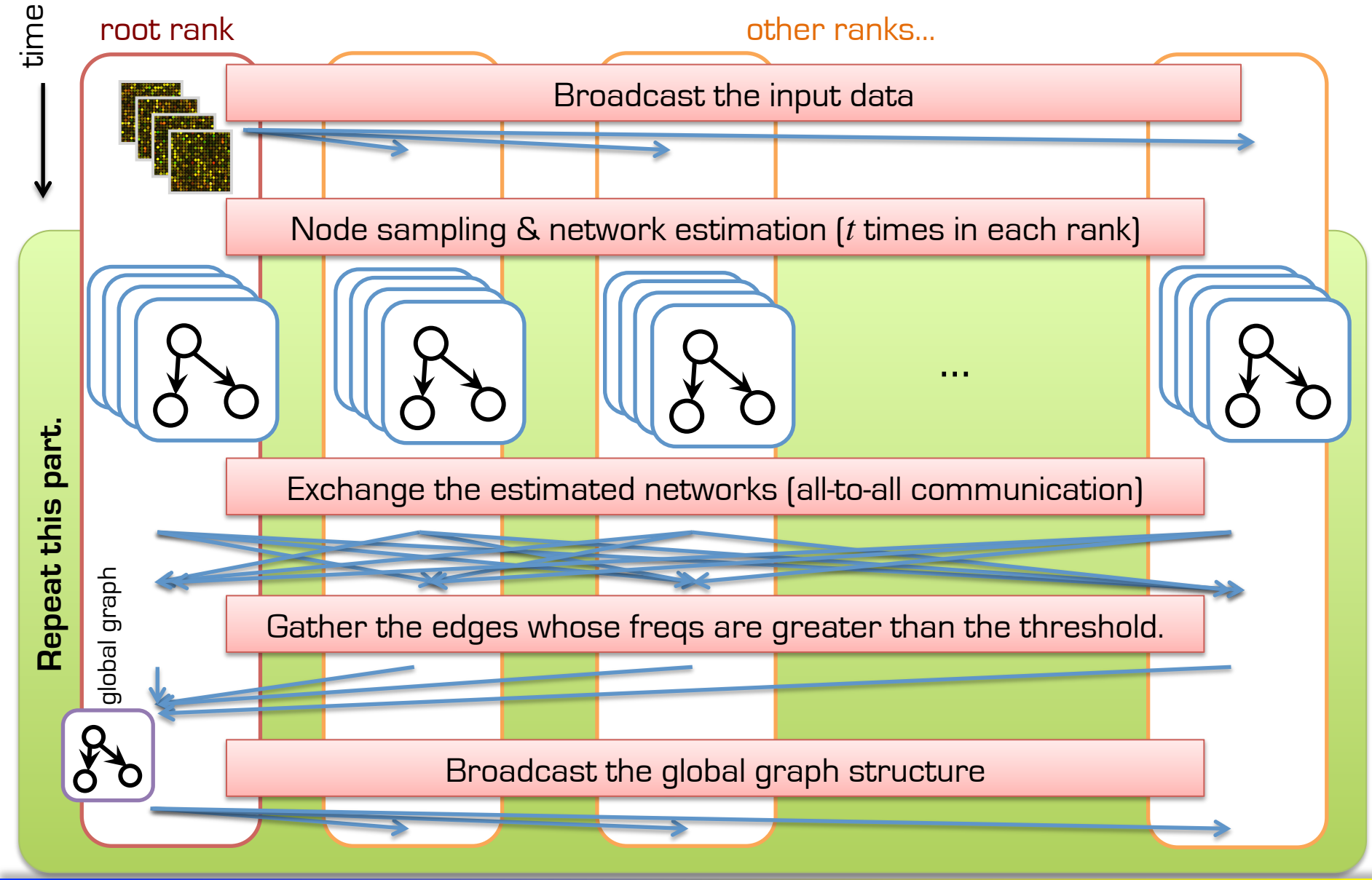
Final Structure $G=(V,E)$: $E=\{(u,v): g_{uv}/c_{uv} \geq \theta\} \quad (u<v)$

g_{uv} : # of estimated (u,v)

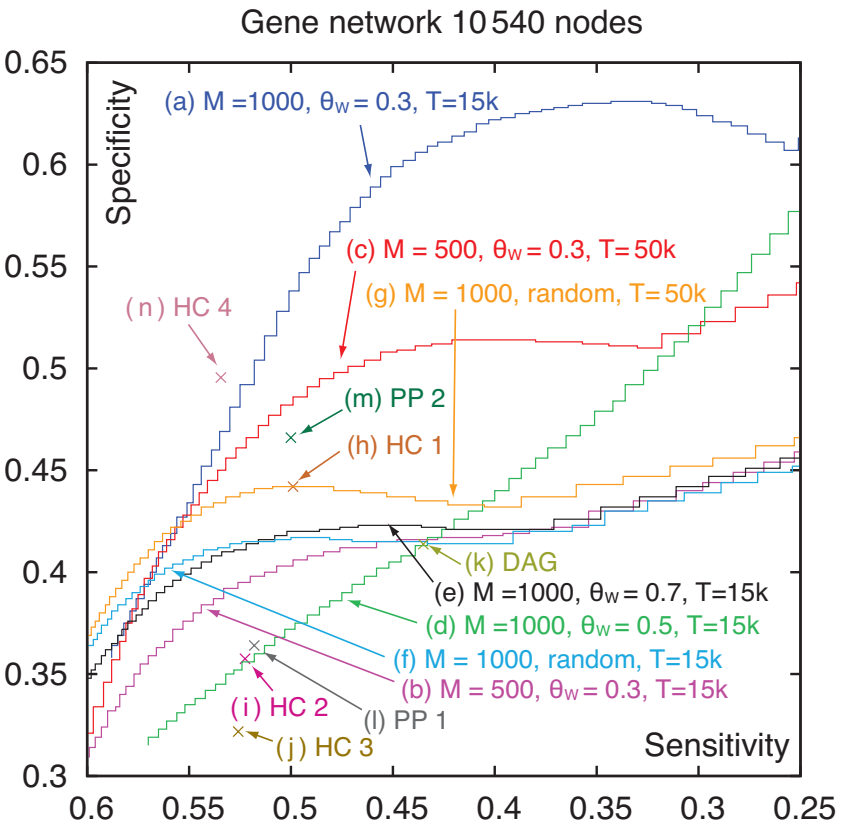
c_{uv} : # of times selected both nodes in a sub network

θ : threshold

Parallelization



Simulation Results



# of samples: 500

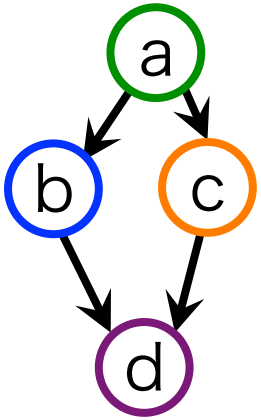
Sp=TP/(TP+FP) Sn=TP/(TP+FN)

- Simulation on the artificial network and expression data.
- Sp and Sn were compared by NNSR, RSR and HC.
- (n) HC4 is a 40 hr execution of HC while (a) took 2 hrs with 200 cores.

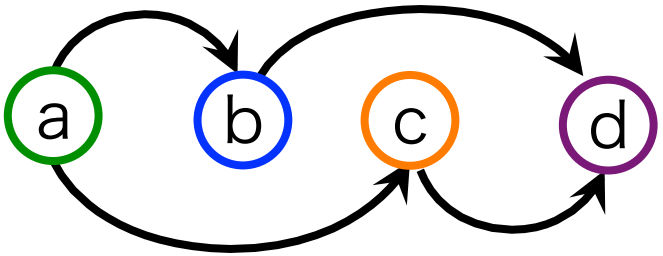
Optimal Search (OS) algorithm

Ott, S., Imoto, S., and Miyano, S., (2004). Finding optimal models for small gene networks. *Pacific Symposium on Biocomputing*, **9**, 557-567.

DAG = there exists a topological ordering



All edges are connected from left to right

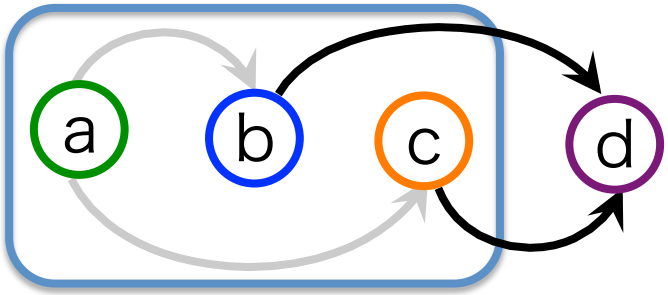


Optimal Permutation

Optimal Structure = +

Optimal Parents

Parent candidate of node d.



Both can be calculated by dynamic programming

Para-OS Algorithm

Largest optimal Bayesian network size until 2010:

Silander, T. and Myllymäki, P. (2006): **n = 29** using **100GB HDD**.

(in 1 week)

Para-OS algorithm (Tamada et al., 2011)

HGC Supercomputer System

Dual Xeon 5450 (3GHz): 8 cores / node

32 GB / node (4 GB / core)

256 cores / 32 nodes (max 1TB)

n = 31 using 453.4GB

6.3 days by continuous model

2.9 days by discrete model

n = 32 using 836.1GB

PREVIOUS WORLD RECORD

5.8 days by discrete model

CURRENT RECORD n = 33 by O. Nikolova.

Performance Test (1)

Scalability to the number of cores

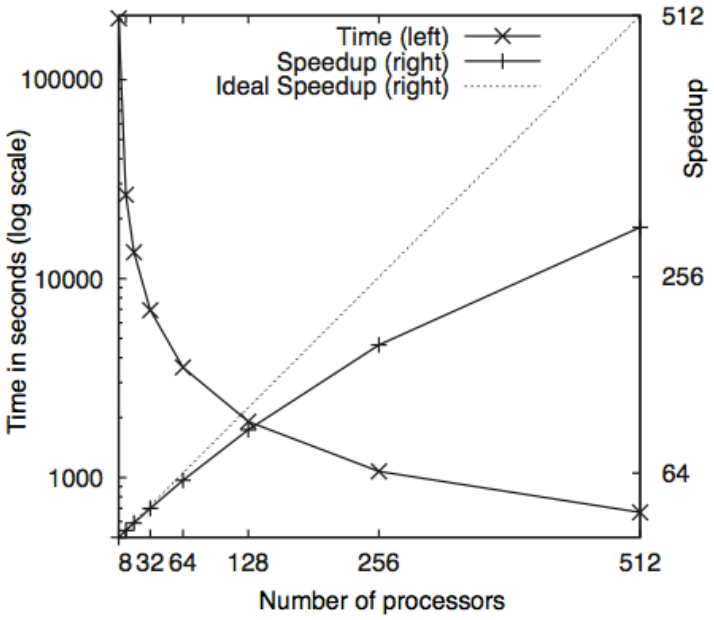
Tested on RIKEN RICC (Fat-tree network)

Dual Xeon 5570 (2.93GHz): 8 cores / node

12GB / node (1.5 GB / core)

55 hr by 1 CPU core

515 sec by 1024 CPU cores

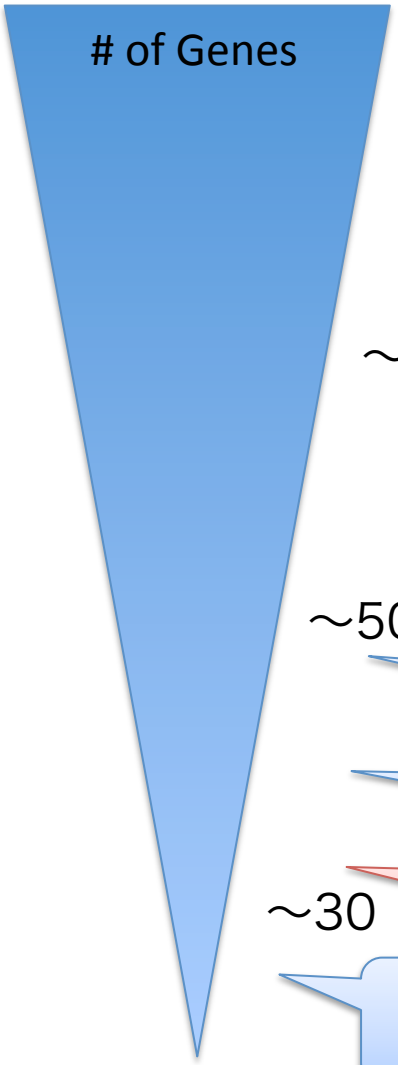


n_p	$T(n_p)$	$S(n_p)$	$E(n_p)$	$T_c(n_p)$	$R_c(n_p)$
1	203297.04	1.00	1.00	0.00	0.00
8	26367.58	7.71	0.96	145.24	0.01
16	13563.89	14.99	0.94	490.24	0.04
32	6932.09	29.33	0.92	398.29	0.06
64	3574.89	56.87	0.89	302.46	0.08
128	1909.72	106.45	0.83	269.90	0.14
256	1072.85	189.49	0.74	249.06	0.23
512	667.38	304.62	0.59	251.46	0.38
1024	515.29	394.53	0.39	305.12	0.59

Figure 3: Scalability test results for $n = 23$ and $N = 50$ with $\sigma = 3$. We did not present the result for $n_p = 1024$ in the graph on the left-hand side because the speedup was too low.

Network Size and Algorithms

Different search algorithms are developed depending on the size of networks



2,000～20,000

Neighbor Node Sampling & Repeat (NNSR) algorithm (Tamada et al., 2011)

～2,000

Greedy Hill-climbing Algorithm (HC) + Bootstrap (Imoto et al. 2002)

～500

Extended COS (Kojima et al. 2010)

Constrained Optimal Search algorithm (COS) (Perrier et al. 2008)

～30

Parallel OS (Para-OS) (Tamada et al. 2011)

Optimal Search algorithm (OS) by Dynamic Programming (Ott et al. 2004)

SiGN （サイン）

- SiGN: A collection of large-scale gene network estimation software designed for utilizing super computers.
  - SiGN-BN: Bayesian networks （ベイジアンネットワーク）
  - SiGN-SSM: State space models （状態空間モデル）
  - SiGN-L1: L1-regularization based models （L1正則化）

SiGN Web Site: http://sign.hgc.jp/

未来の話

京が使いにくい点1

- 計算時間は計算してみないと分からない
  - 京：２４時間
  - HGC：最大２ヶ月（大多数は４８時間）

- 大量に細切れジョブを流したい
  - 一説によると（伝聞）
    - 東大情報基盤センター：40万ジョブ／年
    - HGC：２３００万ジョブ／年
  - バイオインフォでは今までもこれからも EP が主.

京が使いにくい点2

- Java, R, …
  - R は gcc でコンパイルして動くようになったが，，，fcc ではだめ（SIMD使えず）.

- 整数演算　も　速くして.

ゲノム解析ではどうだ

- なんとか無理矢理使おうとしている
- 計算結果のマージ作業がボトルネックになったり．．
  - つまり ディスク I/O
- 既存アプリを組み合わせて使うので移植＆動作確認が大変

まとめ？要望？

- 解析を行っているのは（うちのグループでは）統計解析の専門家.
  - スクリプト程度は書けるがC/Fortranで並列プログラムは書けない.
- 各工程で使える解析プログラムは次々と新しいものがでる.
- 主要スクリプト言語に job 管理システムのAPI や連携可能なインターフェイスを用意して欲しい. . .
  - 一つのジョブが数日から数週間走る.

富豪的HPC

- ゲノム解析・バイオインフォはこれにつきる.
  - つまり，時間をかけてチューニングなど速度のための工夫をしない.
  - あくまでバイオロジー

ハードウェアについては？

- ディスク I/O
- 自分に限って言えば,

もっと

- もっとバイオインフォマティシャン，京を使えばいいのに．．．
  - 使いにくくてもリソースは圧倒的
- だけど．．．
  - 結局やっぱりツールが動かない．．．
  - 随時申請できないとか．．．
  - 余分に覚えること，気をつけないといけないことが多すぎ．
    - ステージングとか，ステージングとか，ステ (rya

夢は

- 夢はライフサイエンスのジョブで京やポスト京を埋めたい.
  - そうじゃないと話を聞いてもらえない.
  - もっと目立たないとナショプロスパコンは永遠に自分が使いやすいものにならない. .
- ユーザ増やなきゃ
  - どうやって？
  - 京はやっぱり速いよ！

最後に

- いろいろ言いましたが．．．
- 待つか，自分が作るか
  - 「未来を予測する最善の方法は、それを発明することだ」アラン・ケイ １９７１
- 自分はなんでも自分でやりたい

Acknowledgments 1

Computational time was provided by the Super Computer System, Human Genome Center, Institute of Medical Science, The University of Tokyo, and RIKEN Integrated Cluster of Clusters (RICC) system.

Collaborators

Human Genome Center, IMS, The University of Tokyo

Prof.
Satoru Miyano

Assoc. Prof.
Seiya Imoto

Lecturer
Rui Yamaguchi

Assist Prof.
Teppei Shimamura

Dr. Ayumu Saito
Dr. Yuichi Shiraishi
Mr. Ken'ich Chiba

Dr. Atsushi Niida

Univ. Auckland (NZ)

Assoc. Prof.
Cristin Print

Dr.
Hiromitsu Araki
Now in Kyushu U.

Univ. Cambridge (UK)

Reader
D S Charnock-Jones

Muna Affara, Ben Dunmore,
Debby Sanders, and Sally Humphreys

Cell Innovator Inc.
Yuki Tomiyasu & Kaori Yasuda

Kyushu University
Kosuke Tashiro & Satoru Kuhara

Acknowledgments 2

理化学研究所 次世代計算科学研究開発プログラム
「次世代生命体統合シミュレーションソフトウェアの研究開発」

HPCI 戦略プログラム分野1「予測する生命科学・医療および創薬基盤」

新学術領域研究
「システムがん」

システム的統合理解に基づくがんの先端的診断、治療,
予防法の開発　「計算とシミュレーションによるがんシステム学の創成」
（研究代表者：宮野 悟）

科研費「汎用自動チューニング機構を実現するためのソフトウェア基盤の研究」
（研究代表者：須田 礼仁）

JST CREST 「進化的アプローチによる超並列複合システム向け開発環境の創出」
（研究代表者：東北大学 滝沢 寛之）

本発表の結果の一部は，理化学研究所が実施している京速コンピュータ「京」の試験利用によるものです.