平成 21 年度 RICC 利用報告書

課題名（タイトル）：

大規模遺伝子ネットワーク推定プログラムの研究開発

利用者氏名：

○宮野 悟

玉田 嘉紀

所属　　：

和光研究所　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研究推進グループ　データ解析融合研究開発チーム

１．　本課題の研究の背景および目的

現在，理化学研究所が中心になって開発している次世代スーパーコンピュータを利活用するためのプログラム「次世代計算科学研究開発プログラム」において，申請者らはデータ解析融合チームの一員として研究開発課題として掲げている大規模遺伝子ネットワーク推定プログラムの研究開発を行っている．

申請者らが開発研究している大規模遺伝子ネットワーク推定プログラムは，ノンパラメトリック回帰によるベイジアンネットワークを遺伝子ネットワークモデルとして利用し，遺伝子発現データなどから遺伝子発現の依存関係を表す遺伝子ネットワークを推定するためのものである．ベイジアンネットワークの構造推定は非常に計算に時間がかかるため，スーパーコンピュータを利用した大規模計算が欠かせない．高精度・大規模な遺伝子ネットワークの推定が可能になることにより細胞内での遺伝子の発現の依存関係を予測することが可能になり，薬剤作用機序解明や新規薬剤標的遺伝子の同定などが可能になることが期待される．

２．　具体的な利用内容、計算方法

本研究では，RICC を利用し，申請者らが開発している大規模遺伝子ネットワークプログラムの高並列化を実現する．高並列化が必要なネットワーク推定アルゴリズムとして４種類を課題として挙げているが，今年度中に「発見的アルゴリズム」，および「最適解アルゴリズム」の 4000 並列までの高並列化を RICC 上で実現する．また来年度以降に実現する 8000 並列の超高並列化動作への課題を見出す．開発しているプログラムは 1024 並列までは申請者が所属するヒトゲノムセンターのスーパーコンピュータで高効率並列動作を確認している．

ベイジアンネットワークによるネットワークのスコアは次のように計算される．ネットワーク G 上のノード X_i の直接の親集合を $Pa(X_i)$ と表す．このとき,マイクロアレイデータ D が得られた元でのネットワークの事後確率 $\Pr(G|D)$ に基づくスコア $S(G) = -2\log \Pr(G|D)$ に対して,分解 $S(G) = \mathrm{S}_i\, s(X_i, Pa(X_i))$ を得る．ここで、$s(X_i, Pa(X_i))$ は、ノード X_i とその直接の親集合 $Pa(X_i)$ により定義されるサブネットワークのスコアである．この $S(G)$ を最小にする非閉路有向グラフ（DAG）を求める問題がベイジアンネットワークの構造推定と呼ばれる．各局所スコアは,

$$s(X_i,Pa(X_i)) = -2\log \int \prod_n f_i(x_{nj} \mid pa(x_i)_n,\theta_i)p(\theta_i \mid \lambda_i)d\theta_i$$

の高次積分を Laplace 近似を用いることにより計算される．ここで f_i は，X_i に対するノンパラメトリック回帰に基づく確率モデル，$p(q_i \mid l_i)$ はパラメータ q_i に対する事前確率分布の密度関数である．

最適な（スコアの最小な）ベイジアンネットワーク構造を探索するには, 解候補となる DAG 構造が膨大にあるため，ノード数（遺伝子数）が多い最適なネットワークを探索することは非常に困難な問題である．従って 1000 遺伝子前後の大規模遺伝子ネットワークをベイジアンネットワークを用いて推定する場合は発見的アルゴリズムを用いる．この際，推定されるネットワークの信頼性を確保するために，データセットからリサンプリングを行い再構成したデータで繰り返しネットワーク推定を行い，推定されるモデルの出現頻度を計算するブートストラップ計算を行う．ブートストラップ計算はデータ並列性があるため，１回のネットワーク推定を１つのコアが担当する．この方法によるベイジアンネットワークの構造探索は１回の計算時間が不均一になるため，並列化は単純に計算を分割する

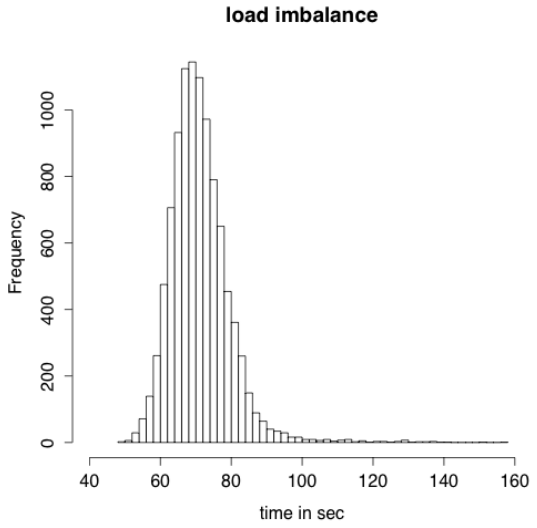

図１．発見的アルゴリズムによる遺伝子ネットワーク推定プログラムの実行時間の分布.

と効率が悪い可能性がある．従って１つのプロセスが専属的にジョブを他のプロセスに割り当てる仕事を担当する．この方法により，1024 並列まで高効率で並列動作することを確認している． 最適ネットワーク探索アルゴリズムにおいても，上記同様に今年度の高並列化はブートストラップによる並列計算を行う． プログラムの実装は C 言語を用いて行い，並列化は MPI を使用した.

３． 結果

発見的方法を用いたネットワーク推定アルゴリズムでは 200 遺伝子 200 サンプルのシミュレーションデータを用いて開発した並列プログラムの並列性能の評価を行った．ブートストラップ計算の繰り返し数として 10,000 回及び 20,000 回の計算を 1（計算）ノード 8 コアから 512 ノード 4096 コアを用いた．評価として次のように定義される並列化効率を用いた．$T(n_p)$ を n_p 個のコア数を使用してアルゴリズムの実行に必要な時間とする．ここではアルゴリズムの純粋な並列性能を評価したいので，MPI の初期化（MPI_Init ルーチン）と終了（MPI_Finalize ルーチン）の間の実行時間のみを計測する．通常は 1 コア実行時との比較を行うが，1 コア実行時に RICC の利用時間制限である 72 時間を超えるため，8 コア実行時との比較を行う．したがって，n_p コア使用時の高速化率 $S(n_p)$ は，ここでは $S(n_p) = T(8)/T(n_p) \times 8$ と定義する．評価に用いる並

表１．並列化効率と単純分割との比較

	10,000 回		20,000 回	
Cores	Eff	Ratio	Eff	Ratio
8	1.00	0.89	1.00	0.89
16	1.07	0.95	1.07	0.95
32	1.10	0.99	1.10	0.99
64	1.12	1.01	1.12	1.01
128	1.12	1.02	1.12	1.02
256	1.11	1.04	1.13	1.03
512	1.10	1.06	1.12	1.05
1024	1.06	1.07	1.09	1.07
2048	0.97	1.03	1.05	1.07
4096	0.83	N/A	0.99	N/A

列化効率 $E(n_p)$ は $E(n_p) = S(n_p)/n_p$ と定義する.

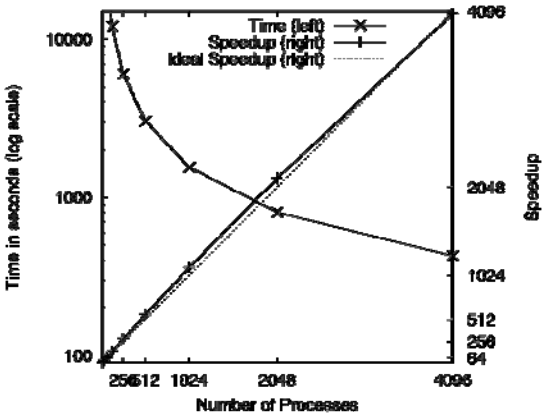

図２．発見的アルゴリズムの 20,000 回ブートストラップ計算の並列化効率.

理想的な並列化の元では $E(n_p) = 1$ となる．アルゴリズムは乱数を用いるため理想的な状態においても必要な計算時間は変化する．図 1 はヒトゲノム解析センターの計算機システムで測定した各繰り返し計算（分割可能な最小単位の計算）の実行時間の分布である．最短約 49 秒から最長 156 秒と各リサンプリングデータに対する推定時間が大きく異なっていることが分かる．従って，並列化効率の測定にはアルゴリズム及び計算機システム自体による実行毎の計算時間ばらつきを考慮し，各コア数 10 回計測し，その平均を用いる．ただし本報告書には提出期限の関係上，最初の 5 回分の計測で取得した結果を掲載する．コア数は 8, 16, 32, 64, 128, 128, 256, 512, 1024, 2048, および 4096 コアそれぞれでアルゴリズムの実行時間を計測した.

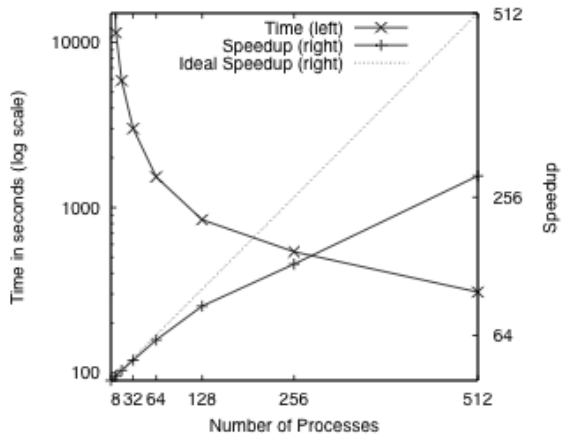

図３．並列版最適化ネットワーク探索アルゴリズムにおける並列化効率.

RICC における測定の結果，10,000 回の繰り返し計算では，4096 並列時の並列化効率は 0.83 であった（表１）．ブートストラップ計算１セット分の総演算時間は 803,781 秒（= 239 時間）であった． 表中の”Ratio”はジョブを単純分割した場合との速度の比較である．低並列時においてはプロセスをジョブ割り当て専用に使用することから単純分割と比較して速度が低下するが，高並列時には最大で１割程度実行速度が向上することが分かった．10,000 回実行の場合の高並列時の並列化効率低下の主な原因は分割可能な計算の最短時間が 1 分半と全体の実行時間に対して比較的長いことに因る．実際 20,000 回実行時には 4096 並列時において並列化効率は 0.99 となりほぼ線形の高速化率を高並列時に達成できることを確認した．20,000 回実行時の並列化効率を図 3 に示す. 1000～2000 程度の並列実行時では並列化効率が 1.0 を越えることが分かった. これは主に，ジョブ割り当て専用にプロセスを割り当てていることに因る高並列時の高効率化が理由である.

最適ネットワーク探索アルゴリズムにおいてもブートストラップ計算プログラムを開発し，同様の並列化効率評価を行った．最適ネットワークアルゴリズム用には 20 遺伝子 50 サンプルのデータを用いた．20,000 回の繰り返し計算を行い，4096 コア使用時の並列化効率は 0.945 であった．ブートストラップ計算 1 セット分の総演算時間は 1,265,731 秒(= 352 時間)であった.

以上より，発見的アルゴリズム及び最適ネットワーク探索アルゴリズムを用いたブートストラップ計算による遺伝子ネットワーク推定プログラムにおいて，2009 年度末までの目標としていた 4096 並列までの高効率並列動作を，本研究実施前に開発したプログラムを改良することなく実現・達成することができた.

予定していた実行回数以下で高効率を達成することが出来たため，より高並列での実行及び，最適ネットワーク並列探索アルゴリズムの RICC での並列動作実行時間の基礎データを取得した．具体的には，発見的アルゴリズムにおいて RICC の空き時間を利用し，7000 コアでの並列動作に成功した．この際の並列化効率は 0.92 であった．また並列版最適ネットワーク探索アルゴリズムでは図３に示すような並列化効率のデータを取得した．並列化版最適ネットワーク探索は並列動作のためにシングル版と比較して高並列ほど無駄な計算が増える．アルゴリズムの詳細は現在論文執筆中のため，ここでは詳細は省くが，簡便に説明すると動的計画法を用いた探索の並列化アルゴリズムである．22 遺伝子 50 サンプルのシミュレーションデータを用いて，高並列時における並列化効率の低下を測定した．1 コア（シングル）実行時 88,072 秒（約 24.5 時間）で計算が完了し, 128, 256, および 512 コア使用時に並列化効率はそれぞれ 0.815, 0.636, および 0.558 となった. 以上より，現時点では 256 コア使用が本アルゴリズムの現実的な並列化効率の上限と言える．また遺伝子数を増やした際の評価も行っており，27 遺伝子を 1024 コア用いて 9,565 秒(2 時間 40 分)で計算が終了した. 今後はハイブリッド MPI 化などにより，高効率化を目指しており，今年度の RICC 利用により，今後の次世代スーパーコンピュータ向けアプリケーション開発のための重要な結果を蓄積することが出来た.

４． 今後の計画・展望

計画では，平成 22 年度は 6 月までに本研究で用いたブートストラップ計算による遺伝子ネットワーク推定プログラムの 8000 並列での高効率動作を達成する予定である．また全ゲノムサイズ遺伝子ネットワーク推定プログラムの高並列版プログラムの開発・評価および上位構造に基づく最適ネットワーク探索アルゴリズムの並列アルゴリズムの開発を行う.

５． RICC の継続利用を希望の場合は、これまで利用した状況（どの程度研究が進んだか、研究においてどこまで計算出来て、何が出来ていないか）や、継続して利用する際に行う具体的な内容

本研究の最終的な目標は次世代スーパーコンピュータにおける 10,000 並列以上での高効率動作可能な並列プログラムの研究・開発である．現在は当初の計画通り 4000 並列まで達成し，次年度で 8000 並列を目指す．最適ネットワーク探索アルゴリズムおよび全ゲノムサイズ遺伝子ネットワーク推定プログラムにおいてはハイブリッド MPI 化等の高並列化のためのアイディアが必要となるので，これらのプログラムの研究開発を行い，RICC を用いて高並列動作の評価を行う．

６．　一般利用で演算時間を使い切れなかった理由

RICC 利用前までに開発したプログラムを改良せずに高効率を達成できたため，プログラムの改良と測定を繰り返す必要が無く予定していた演算時間が余った．現時点では結果の概略を得るために当初予定していた１０回の測定中５回しか行っていないが，３月中の残り演算時間を使用して測定予定である．

平成 21 年度　RICC 利用研究成果リスト

【国際会議、学会などでの口頭発表】

Tamada, Y., Supercomputing on Gene Network Estimation with Bayesian Networks, The 2nd NUS-UT Workshop on Computational Systems Biology, Feb. 2010, Tokyo.