3K-3

BLASTアルゴリズム全体のハードウェア化による高速化の検討

石川淑† 田中飛鳥† 宮崎敏明†

会津大学大学院コンピュータ理工学研究科†

1. はじめに

Basic Local Alignment Search Tool(BLAST)は最も有名なシーケンスアライメントツールの一つである.シーケンスアライメントとはタンパク質(またはDNA)配列データベース(DB)内のシーケンスと検索対象となるタンパク質(またはDNA)配列(クエリシーケンス)を比較し,配列同士の類似部分検索を行うものである.シーケンスアライメントは,生物学上の進化や遺伝子系図を調べる上で重要であることから,バイオインフォマティクス分野では欠かせない情報となっている.BLAST のハードウェア化による高速化手法は多く提案されてきた[1,2].その殆どは,一部の処理をホストPCで行うというものであり,通信ボトルネックが生じていた.本研究では,BLAST アルゴリズム全体をハードウェア化することによりホストPCとの通信ボトルネックを解消し,高速化を行ったので報告する.

2. BLASTアルゴリズム

BLAST アルゴリズムは,前処理,Seeding(ステップ1),Ungapped extension(ステップ2),Gapped extension(ステップ3)の3つのステップとtraceback 処理からなる.前処理では,検索に用いる隣接ワード(Neighborhood word)と呼ぶ文字列を生成する.隣接ワードはクエリシーケンスと置換行列(Substitution Matrix)を用いて生成される.置換行列とは,タンパク質を構成する20種のアミノ酸同士の類似度を数値化した表であり,一般に20×20の行列形式で表現される.ここでは,比較する2つの配列の類似度を数値化したものをスコアと呼ぶ.置換行列は複数存在するが,本稿では,BLOSUM50 と呼ぶ置換行列を使用する.まず,クエリシーケンスをk文字(通常,タンパク質配列ではk=3,本稿でもk=3を用いる)のクエリワードに分割する.つぎに,そのクエリワードの3文字を他のアミノ酸の20文字と1文字ずつ比較しスコアを計算する.3 文字のスコアの合計が閾値T(通常T=12)以上となったワードが隣接ワードとなる.Seeding(ステップ1)では,前処理で生成した隣接ワードを使用してDBシーケンス内のseedを探す.seed とは,DB シーケンス上で隣接ワードが一致した位置のことである.ステップ2,すなわちUngapped extension ではステップ1で見つかったseed を拡張開始点とし,一致点の拡張を行う.拡張は合計スコアが最大値から閾値 S(ユーザが指定)下がるまで行う.拡張されたシーケンスペアのうちHSP(High scoring segment pair)と呼ぶ組み合わせだけが次のステップ3の入力となる.ステップ3のGapped extension では Smith-Waterman アルゴリズムという動的計画法に基づく手法を使用している.当該手法では,計算処理に2次元データ行列(以下,スコア行列と呼ぶ)を用いる.図1は,HSP をスコア行列の行と列に対応させ,各行列要素(以下,セルと呼ぶ)のスコア計算を行った結果である.1つのセルは,上,斜め上,そして左のセルのスコアを使用して計算された値と,数値0の4つの値を比較し,その最大値をスコアとして該当セルに保持する.図1中の矢印は,どの位置のスコアを使用したのか(経路)を示している.traceback 処理ではステップ3で求めたスコア行列上で,最も高いスコアを持つセルからスコアが0のセルまでの経路を辿ることにより,最適なシーケンスを出力する.図1の例では,“TALILA”と“TAL-LA”が最適なシーケンスとなる.ここで,シーケンスペア中の“-”はギャップと呼ばれ,文字列のずれを示す.

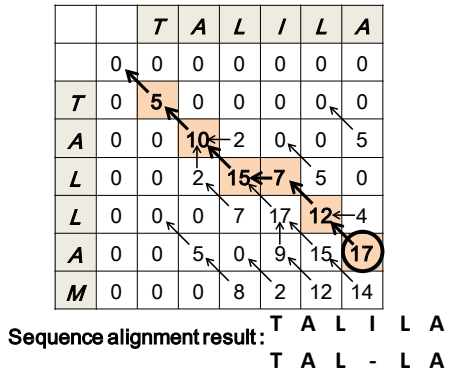

図1 Gapped extension 処理と traceback 処理によって求められた計算結果と最適なシーケンス

3. 提案手法

従来,BLAST のハードウェア化は,前述のステップ3,すなわちGapped extension を中心に行われてきた.そのため,前処理である隣接ワードの生成やtraceback 処理は,ホストPC上のソフトウェアで行う必要があり,PC とハードウェア間で通信ボトルネックが生じる可能性があった.本稿では,前述したtraceback 処理を含むBLAST アルゴリズム全体をハードウェア化することを提案する.ここでは,BLASTP(タンパク質配列のシーケンスアライメントツール)を対象とする.図2に,Gapped extension 処理と traceback 処理を実行する提案回路を示す.Gapped extension 部内の計算には,Processing Element(PE)と呼ぶ簡単な処理を行うプロセッサを

An approach to accelerating BLAST algorithm using dedicated hardware

†Shizuka Ishikawa, †Asuka Tanaka, †Toshiaki Miyazaki

†Graduate School of Computer Science and Engineering, The University of Aizu

複数使用する.図2のように複数のPE同士を線形接続して構成するPEアレイで並列処理を可能とし,ソフトウェア処理に比べ高速化を実現する.また,各PEにはローカルメモリを接続し,計算結果は,ホストPCに転送せずに,当該メモリに格納する.ローカルメモリはM×Nのサイズ(M≧クエリシーケンス長,N≧データベースシーケンス長)の2次元メモリとなっており,各計算結果を図1で示したスコア行列に対応する位置に格納できる.ローカルメモリに格納される各計算結果は,四つ組の情報(行列に対応したクエリ,DBシーケンス中の各1文字,どの位置のスコアを使用したかを示すparent(図1の矢印を数値で表したもの),計算によって求められたスコア)からなる.PEアレイの計算が終了すると,クエリの終端文字を保持するPEから,終了信号と最大スコア,最大スコアが保存されているメモリアドレスがtraceback部に送られ,直ちにtraceback処理が開始される.traceback部では,PEから届いた最大スコアのメモリアドレスを最初の読出しアドレスとして,2次元メモリにアクセスする.メモリから出力された四つ組データはtraceback部に送られる.traceback部は,四つ組データ中のparentを使用して次の読出しアドレスを生成するとともに,他のクエリ,DBの文字データは,そのまま出力する.本操作を,ローカルメモリから読み出した四つ組データのスコアが0になるまで繰り返す.上記の操作に於いて,traceback部から出力されたデータが最適なシーケンスとなる.

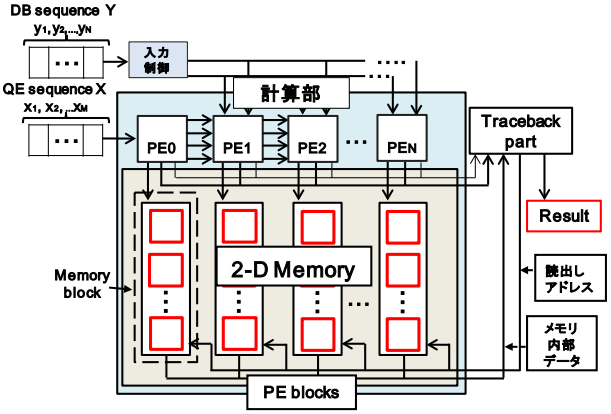

図2 Gapped extension処理とtraceback処理を実行する提案回路

4. 全体構成

図3に提案回路の全体構成を示す.提案回路は,Block 1からBlock 4の4つの部分から成る.各部は,前述した隣接ワードを求める前処理(Block 1),ステップ1,2,3(Block 2～4)に対応している.Block 4には、Gapped extension処理部に加え,traceback処理部も含まれる.前処理部では,入力されたクエリシーケンスを用いて隣接ワードメモリの内容を生成する.この隣接ワードメモリは,Block 2の処理で,何度も参照される.Block 2では,DBシーケンスを3文字に分割し,隣接ワードメモリに送り同一文字の一致点(seed)を探す.Block 3のUngapped extension処理では,パイプライン処理を効率的に行えるように文字列を左右方向に拡張するのではなく,一方向へのみ拡張を行うようにしている.Block 3の処理で求められたHSPは時間調整用のFIFOを介して,最終処理部であるBlock 4へ送られる.Block 4では,Smith-Waterman アルゴリズムに基づいた計算が行われ,計算終了後,最適なシーケンスを出力する.

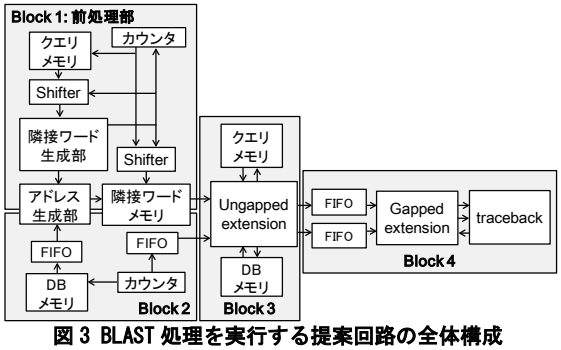

図3 BLAST処理を実行する提案回路の全体構成

5. 評価

動作確認のため,FPGAを用いて提案回路の実装を行った.表1は,図3に示した回路全体の実装結果であり,表2は提案回路とソフトウェアBLASTの実行時間を比較したものである.評価に用いたクエリシーケンス、DBシーケンス長はともに100である.使用したFPGAはCyclone-IV E EP4CE115F29C7であり,回路実装には同社の設計ツールQuartusII 10.1sp1を使用した.提案回路の最大動作周波数は55.11 MHz,必要となったクロックサイクル数は9383クロックであり,実行時間は0.17 msであった.比較対象として、ソフトウェアBLAST (Local BLAST 2.2.25)を使用した.同一入力データに対して,その実行時間は135 msであった.よって,表2に示すように,提案回路はソフトウェアに比べて791倍の高速化を実現した.

表1 前処理部のFPGA回路規模

リソース	使用量	使用率
ロジックエレメント数	41,142	36 %
レジスタ数	3,656	3 %
メモリビット数	653,056	16 %

表2 ソフトウェアBLASTと提案回路の比較

	実行時間	高速化率
ソフトウェアBLAST	135.00 ms	1
提案回路	0.17 ms	791

6. おわりに

BLASTの処理全体を実行するハードウェアアーキテクチャを提案し,ソフトウェアよりも790倍以上高速であることを示した.今後は,PEに接続されているメモリをdual memoryとすることで,計算処理とtraceback処理を並列実行させ,さらなる高速化を目指す.

参考文献

[1] K. Benkrid, Y. Liu, and A. Benkrid, “A Highly Parameterized and Efficient FPGA-Based Skeleton for Pairwise Biological Sequence Alignment,” IEEE Trans. on VLSI Systems Vol. 17, No. 4, pp. 561-570, April 2009. [2] S. Kasap, K. Benkrid, and Y. Liu, “Design and Implementation of an FPGA-based Core for Gapped BLAST Sequence Alignment with the Two-hit Method,” Engineering Letters, Vol. 16, Issue. 3, No. 25, (EL_16_3_25), August 2008.