3I1-09

接尾辞木に基づいた疑似バイクラスタ抽出手法

A Method of Finding Pseudo Bi-Clusters Based on Suffix Tree

難波 徹郎　　原口 誠
Tetsuro NAMBA　Makoto HARAGUCHI

北海道大学大学院情報科学研究科コンピュータサイエンス専攻
Division of Computer Science, Graduate School of Information Science and Technology, Hokkaido University

We discuss in this paper a method for finding *Pseudo Bi-Clusters* of gene expression data. For time series data, a linear time algorithm with the help of *suffix tree* has been proposed. Although the algorithm can efficiently enumerate all bi-clusters, we often obtain many overlapping clusters. In order to cope with the difficulties, we introduce a notion of *pseudo bi-clusters* and present an algorithm for finding them with a suffix tree. Some experimental results for gene expression data of ascidian (Hoya) are also presented.

1. はじめに

近年，特定の DNA 配列を検出するプローブをスライドガラスなどの基盤上に配置し，細胞中で発現している遺伝子を検出する DNA マイクロアレイ技術により，数千から数万個の遺伝子発現データを一度に測定することが可能となった．これにより，大量の発現データを短期間で解析し結果を理解することが重要な課題となっている．特に，生物の各発生段階での変動を測定した遺伝子発現データの解析ではその解析のために

同じような発現変動を示す遺伝子群を抽出する

必要がある．そこで発現データにデータマイニングを適用し，遺伝子群を抽出することを目的にクラスタリングなどが行われてきた．しかし，遺伝子発現時系列データは特定の遺伝子が特定のステージに発現することが多く，従来のクラスタリング手法を用いた遺伝子クラスタリングでは有用な結果を得にくい問題があった．この問題を解決するためにデータ行列の行と列，すなわち遺伝子とステージを同時にクラスタリングするバイクラスタリングの適用が試みられている．

特に時系列データのバイクラスタリングにおいては，生物学的な発現プロセスは連続した時間の発現値の増減で示される，という仮定から，連続した列を持つバイクラスタに限定して解析が行われることがある．そのようなバイクラスタリング手法の 1 つに Madeira らの接尾辞木に基づいた線形時間バイクラスタリングアルゴリズムがある [1]．このアルゴリズムは，各遺伝子データを符号列に変換することでそれぞれに共通な部分符号列を探索し，極大なバイクラスタを効率よく判定，抽出することができる．そしてその結果，多くの極大バイクラスタを抽出することができる一方で，同じ発現変動を示すステージの重なりを持つ極大バイクラスタが多く存在する．

本研究では，生物の遺伝子発現時系列データの解析に際し，接尾辞木に基づいた線形時間バイクラスタリングアルゴリズムを利用する．そして抽出した全ての極大バイクラスタそれぞれについて，他の極大バイクラスタとの関係を見ることでその極大バイクラスタの位置づけを考える．そのために本研究では重複の度合が高いバイクラスタ同士を 1 つにまとめる疑似バイクラスタの概念を導入する．そしてバイクラスタ同士の重複の度合として，

同じ発現変動を示すステージの重なり数

に着目し，ステージの重なり数が一定以上のものを 1 つにまとめることで，重複ステージの発現変動を認識できるような疑似バイクラスタを接尾辞木の構造に基づき抽出する．

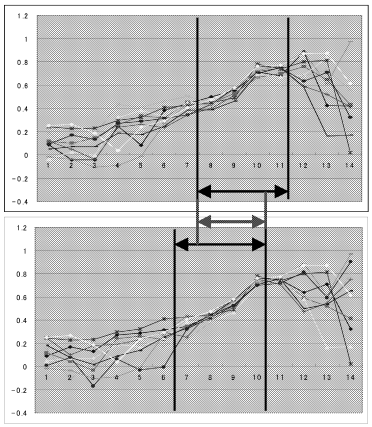


図 1: 発現変動が重なるステージを持つバイクラスタ

連絡先: 原口 誠
北海道大学大学院情報科学研究科コンピュータサイエンス専攻
〒 060-0814 札幌市北区北 14 条西 9 丁目
TEL：011-706-7106
E-mail：mh@ist.hokudai.ac.jp

2. 接尾辞木に基づいたバイクラスタリング

2.1 本研究のバイクラスタ

本研究で対象とする極大バイクラスタは，データのサイズを考慮し，遺伝子数 10 以上，ステージ数 3 以上を持ち，他のバイクラスタに完全に含まれることのないバイクラスタとする．

そして，抽出した極大バイクラスタそれぞれについて疑似バイクラスタを形成できるかを調べる．同じ発現変動を多くのステージで示すバイクラスタをまとめるため，ステージの重なり具合に注目し，極大バイクラスタそれぞれに対してそのステージ数の過半数で他の極大バイクラスタのステージが重なるときのみ，これをまとめていく．そして，重なり合う極大バイクラスタ全てが共有するステージとそのステージで共有する遺伝子からなる部分を核と呼ぶとき，核を持つときのみその極大バイ

クラスタと，それに重なる極大バイクラスタ群を 1 つの疑似バイクラスタとする．

2.2 接尾辞木

接尾辞木とは，対象となる文字列 s 中の任意の文字から s の末尾までの範囲の文字列を指す接尾辞を索引単位とするデータ構造である．

N 文字からなる文字列 s に対する接尾辞木は，N 個の葉を持つ木構造であり，根を除く内部ノードは 2 つ以上の子ノードへの枝を持つ．そして各枝は空でない文字列によりラベル付けされるが，それらのラベルの先頭文字は常に異なっていなければならない．各葉には番号 i が対応し，根からその葉へのパス上のラベルを連結すると，s 中で i 番目の文字から開始する接尾辞と一致する．

以上が接尾辞木の定義であるが，任意の s について接尾辞木が存在するには文字列 s の末にアルファベットに属さない記号 (終端記号) を加える必要がある．またこの定義から接尾辞木のノード数は $2N-1$ 以下になる．なぜなら，根を除く内部ノードは少なくとも 2 つ以上の分岐を持つため，$N-1$ 個以下でなければ葉の総数が N を越えてしまうからである．ラベルの表現として，文字列の代わりに s におけるその文字列の開始と終了位置からなる番号ペアを用いることで接尾辞木の総サイズは N のオーダーになる．

また，複数の文字列集合に対する接尾辞木の構築は一般化接尾辞木と呼ばれ，それぞれの文字列に違う終端記号を挿入し，それらを繋げて構築する．

2.3 ノードとバイクラスタ

本研究では，各遺伝子のデータを符号化し，それぞれのステージ番号を付与することで各遺伝子の符号列を取得する．そして取得した符号列集合から一般化接尾辞木を構築する．符号列集合 S から構築された一般化接尾辞木を T とする．T のノードを v とし，v を部分木の根としたときの葉の数，すなわち v が持つ葉の数を $L(v)$，T の根から v までの枝のラベル文字列を連結したときの文字数を $P(v)$ とする．このとき，一般化接尾辞木 T の全ての内部ノードはそれぞれ列が連続したバイクラスタと一致する．なぜなら T の内部ノードは，その内部ノードが根の葉を持つ全ての行に共通な部分文字列と一致するからである．さらにこのことから，$L(v)$ はバイクラスタの持つ行の数，すなわち遺伝子数と一致し，$P(v)$ はバイクラスタの持つ列の数，すなわちステージ数と一致するといえる．

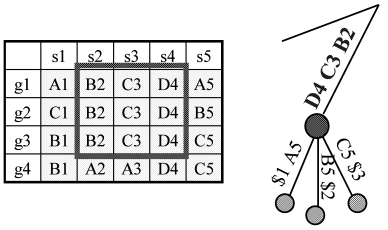


図 2: ノードとバイクラスタ

3. suffix link

接尾辞木は Ukkonen により提案された構築法により N の線形の時間オーダーで構築可能である [2]．Ukkonen 法では文字列 s の先頭文字から 1 文字ずつ入力しながら順次グラフ構造を成長させていく過程で，各ノードから先頭の文字が 1 つ短い接尾辞のノードにリンクが張られており，これを suffix link と呼ぶ．

一般化接尾辞木を T とし，T のノード v_1 がバイクラスタ B_1，ノード v_2 がバイクラスタ B_2 と一致するとする．このとき，もしノード v_1 からノード v_2 へ suffix link が存在した場合，suffix link の定義からバイクラスタ B_2 はバイクラスタ B_1 よりも 1 列少ない．

3.1 極大バイクラスタ判定

先行研究の接尾辞木に基づいたバイクラスタリングでは，suffix link の定義から次のような極大判定を行い，極大バイクラスタを抽出していた [1]．

> もしノード v が内部ノードの場合，ノード u から v へ suffix link があり $L(v) > L(u)$ の時かつその時に限り，ノード v は極大バイクラスタである．

3.2 極大バイクラスタ間の重複探索

本研究では，suffix link とノードの親子関係から極大バイクラスタ同士の重複探索を行う．

一般化接尾辞木 T のノード v_1 が極大バイクラスタ B_1 と一致するとき，ノード v_2 へ suffix link があるとする．このとき，suffix link の定義からバイクラスタ B_2 はバイクラスタ B_1 よりも 1 列右にずれた列からなるバイクラスタで B_1 と重なっており，その重複ステージ数はノード情報から確認できる．

> B_1 と B_2 の重複ステージ数は $P(v_2)$ と一致する．

これは，ノード v_2 の子ノードから葉ノードまでの間にある極大バイクラスタのノードに関しても同様である．一方，ノード v_1 の親ノードから根ノードまでの間にあるノード v_3 が極大バイクラスタ B_3 と一致する場合，B_1 と B_3 の重複ステージ数は $P(v_3)$ と一致する．これは，ノード v_2 の親ノードから根ノードまで間にある極大バイクラスタのノードに関しても同様である．そして，ノード v_1 の子ノードから葉ノードまでの間にあるノード v_4 が極大バイクラスタ B_4 と一致する場合，B_1 と B_4 の重複ステージ数は $P(v_1)$ と一致する．ノード v_1 とその接尾辞に関する suffix link 先ノード全てについて以上のことがいえる．

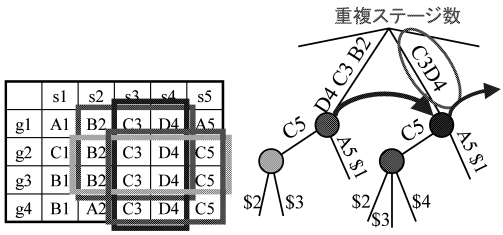


図 3: suffix link による重複探索

4. 疑似バイクラスタ抽出

全ての極大バイクラスタに対して疑似バイクラスタを形成できるかどうかを調べるため，全ての極大バイクラスタのノードから，その接尾辞に関する suffix link だけを順々に辿ることで，各ノードに一致するバイクラスタ同士の重複と核をそれぞれ同時に判定し計算する．

全ての極大バイクラスタのノードから，その接尾辞に関する suffix link だけを順々に辿る過程で，自身の子ノードから

葉ノードまでの間と親ノードから根ノードまで間，リンク先のノードと，その葉ノードまでの間と親ノードから根ノードまで間にあるノードから極大バイクラスタのノードを探索する．そして極大バイクラスタのノードを発見した場合，出発したノードと探索したノードから重複ステージ数を確認し，出発したノードと探索したノードの双方それぞれについてステージ過半数を満たすか計算する．そして重複ステージ数がステージ過半数を満たす場合，そのノードの核を計算する．

全ての極大バイクラスタがそれぞれの suffix link を辿っていく過程で，核が 0 になったノードについては以後重複を確認する必要は無く，核 0 になったノードが出発ノードとして suffix link を辿るときは，その探索先ノードについてのみ重複を確認すればよい．

以上の手順で全ての極大バイクラスタのノードから suffix link を辿り，最終的に核を持った極大バイクラスタとそれに重なる極大バイクラスタを疑似バイクラスタとしてまとめ，これを枚挙する．

なお，全ての極大バイクラスタそれぞれが suffix link を辿り，そのリンク先ノードに関する全ての根ノードから葉ノードを検索するため，データサイズを n，極大バイクラスタ数を m とすると，その計算時間は最大で $O(nm^2)$ である．

5. 実験と考察

本実験では，ホヤの遺伝子発現時系列データ (2340 遺伝子 × 14 ステージ) から一般化接尾辞木を構築し，疑似バイクラスタの抽出を試みた．実験環境は，CPU:Pentium4 2.8GHz, Memory:2046M，OS:WindowsXP Professional，プログラミング言語:Java である．

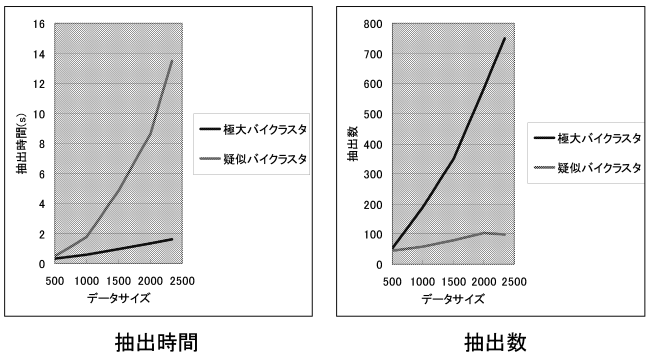
5.1 符号化

本研究では 2340 遺伝子× 14 ステージのホヤの遺伝子発現時系列データに対して，各ステージそれぞれを発現値の大きいものから 234 個ずつ 10 グループに分け，A ～ J の 10 種文字に変換する．そして各文字にステージ番号を付与することで 2340 個の遺伝子の符号列を取得し，一般化接尾辞木を構築した．

5.2 実験結果

データサイズの比較による計算量と，実際の疑似バイクラスタ抽出結果により実験評価を行う．

5.21 データサイズによる比較まず，疑似バイクラスタ抽出時間と抽出数を 14 ステージの遺伝子数の比較により測定した．



抽出時間　　抽出数

図 4: データサイズと計算量

計算時間は最大で $O(nm^2)$ であるが，これは大まかには $O(n^3)$ と同等であり，実際の抽出時間もそれを表す結果となった．一方抽出数は，極大バイクラスタ数が増えても疑似バイクラスタの抽出数は増えず，これはデータが増えるにつれ核の共通な遺伝子を持つことが難しくなったためだと思われる．

5.22 抽出結果2340 遺伝子× 14 ステージのデータから，全ノード数 : 48077，suffix link 数 : 12836 の一般化接尾辞木が構築され，極大バイクラスタ : 750 個，疑似バイクラスタ : 98 個を抽出した．また，1 つの疑似バイクラスタを形成した極大バイクラスタの平均数 : 3.3，核のステージ数平均は 2.3 であった．図 5 に疑似バイクラスタとそれを形成した極大バイクラスタのグラフを示す．

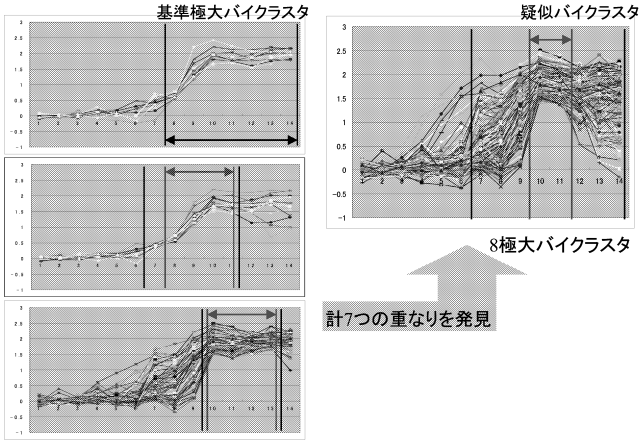


図 5: 疑似バイクラスタ

5.3 考察

全ての極大バイクラスタそれぞれに対して重なりを探索したため計算量が大きくなったが，ノード探索の順番を変えてある程度無駄な計算を減らせる可能性がある．しかしデータの列数によっては極大バイクラスタのノード数が増えることもあり，データの種類によっても計算時間は左右されると考えられる．

また，極大バイクラスタ 750 個のうち 98 個しか疑似バイクラスタを形成しなかった．その疑似バイクラスタは重なり数平均が約 3 で，小さな極大バイクラスタの集まりであることが多かった．これは小さな極大バイクラスタほど核を持ちやすいためであると考えられる．本研究の核の定義では，大きなバイクラスタほど重複数が増えるために核を持ちにくく，ステージ重複数の条件をさらに上げる必要がある．

なお，符号化の粒度が半分の 5 種文字と倍の 20 種文字で符号化したデータで実験した結果も，同様の特徴が見られた．

6. まとめと今後の課題

本研究では，接尾辞木に基づいたバイクラスタリングによるホヤの遺伝子発現時系列データの解析にあたり，抽出された極大バイクラスタそれぞれに対して同じ発現変動を示すステージの重複に着目して疑似バイクラスタ抽出を行った．

今後の課題としてはまず計算量の軽減がある．本研究では全ての極大バイクラスタそれぞれに疑似バイクラスタ抽出を行ったため，suffix link を利用してもどうしても計算量が大きくなる．そこで，疑似バイクラスタをそれぞれの極大バイクラスタで考えるのではなく，重複条件を相互に過半数で重なるようなものにすることで計算の組み合わせ数を減少できる可能性があ

る．また本研究のアルゴリズムにも無駄に計算している部分があると思われるのでその点を解決する必要がある．

次に符号化の問題がある．本研究で利用したバイクラスタリングアルゴリズムは符号化方法やその粒度の設定に実験結果が大きく左右される．そこで，よりデータの特性を考えた符号化をするためにデータの散らばり具合や密度の概念を取り入れて符号化する必要がある．

そして，このように得られた結果が生物学的に意味を持つのか，有用な結果が得られているのかは分からず，専門家に実験結果データの解析を御願いする必要がある．

7. 謝辞

本研究に関して，接尾辞木構築アルゴリズムに関する貴重な御助言と御指導頂きました北大大学院情報科学研究科・喜田拓也准教授，データの提供をして頂いた北大創生科学共同研究機構・安住薫助教に深く感謝し御礼申し上げます．

参考文献

[1] Sarac C.Madeira, Arlindo L. Oliveira. A Linear Time Biclustering Algorithm for Time Series Gene Expression Data. In Proc. of WABI2005, p39-52, 2005.

[2] E. Ukkonen. On.line construction ofsuffix.trees. In Proc. of Algorithmica, Vol. 14, p249-260, 1995.