連載

5分で分かる!? 有名論文ナナメ読み

清水佳奈（早稲田大学）

Ferragina, P. and Manzini, G. : Opportunistic Data Structures with Applications

In Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00), IEEE Computer Society, Washington, DC, USA, pp.390-398 (2000)

全文索引

文書から任意の文字の並びを検索することを全文検索と呼ぶ．全文検索を高速に行うためには，あらかじめ文書を検索しやすいように索引化しておけばよい．索引法の１つとして知られる接尾辞配列は，文書中に現れるすべての接尾辞[☆1]を辞書順にソートし，各接尾辞の出現位置を記録したデータ構造である．図 -1 に文書：“ATGAATGCGA$”の接尾辞配列を示す．接尾辞配列では，接頭辞が共通する接尾辞は隣り合うため，パターン“ATG”の存在を確かめるには接尾辞配列を二分探索すればよいのだが，文書の大きさに依存した検索時間が必要となってしまう．本稿では，接尾辞配列と密接な関係がある Burrows-Wheeler 変換（BWT）を用いて文書の大きさに依存しない検索時間を実現した索引法の論文を紹介する．本論文自体の引用件数は1,012にとどまるが[☆2]，この画期的な索引法は後に FM-index と名づけられ 30,000 件を超える研究論文で利用されている．FM-index は一体何に応用されたのか．

☆1 本稿では文字列の i 番目の文字から末尾までの部分文字列と定義する．

☆2 文中の引用数はいずれも Google Scholar Citations（2019 年 2 月 28 日アクセス）による．

図 -1 FM-index による全文検索

新型 DNA シークエンサーの登場と情報爆発

DNA が持つ遺伝情報は 4 種類の文字の並び（以下，塩基配列）として表現される．ヒトを含むいくつかの生物種では，代表的な個体の全塩基配列が読み取られており，さまざまな条件下で得られた塩基配列との比較に用いられる．これを参照ゲノムと呼ぶ．たとえば，ガン細胞で働く遺伝子の塩基配列を読み取れたとしよう．ガンのメカニズム解明には，読み取った塩基配列がどの遺伝子由来なのか（参照ゲノムの何文字目から何文字目に相当するか）知る必要があるが，これは参照ゲノムに対する全文検索により発見できる[☆3]．このように，ゲノム情報解析では全文検索が重要な役割を果たす．FM-index が発表された 2000 年当時，参照ゲノムの全文検索ではクエリの n-gram[☆4] とその近傍文字列をオートマトンで表現して参照ゲノム全体を走査する手法が主流であった．ところが，2000 年代後半に DNA から塩基配列を読み取る装置（DNA シークエンサー）の性能が劇的に向上すると膨大なデータが産出されるようになり，従来の手法では立ち行かなくなってしまった．一体どれほどデータが増加したのか．米国立生物工学情報センターの試算では，過去 15 年間で塩基配列の読み取りコストは約 10 万分の 1 に低下したという．旧型のシークエンサーが 1 回の実験あたり長さ数百〜千程度の塩基配列を数百本出力するのに対し，最新型では長さ数百の塩基配列を数億本も出力する．そのため，検索を効率化するには索引化が必須であった．特に，参照ゲノムが比較的大きい（ヒトの場合 30 億塩基対）ことを考慮すると，索引サイズを抑えつつも，文書サイズに計算量が依存しない方法が望ましい．このような状況下で注目を集めたのが FM-index であった．FM-index による全文検索は BWA や Bowtie といったゲノム情報解析の研究者であれば誰もが知っているソフトウェアに用いられた．BWA と Bowtie の論文は双方とも 2009 年に発表されたが，現在までの引用数はそれぞれ 17,714 と 13,663 に上る．この数値が示す通り，まさに FM-index は現代の生命科学を下支えしているのである．

☆3 実際の解析ではシークエンサーのエラーや生物の個体差などを考慮するためにあいまい検索を行うが本稿での説明は割愛する．

☆4 隣接する n 文字．

BWT の性質

BWT は次の手順により文字列 S を文字列 L に変換する．まず，S の末尾に終端文字を追加し[☆5]，S に現れるすべての接尾辞を辞書順でソートする．次に，各接尾辞の先頭より 1 つ前の文字を取り出して並べた文字列を L とする．ただし，S そのものを接尾辞とする場合は終端文字を取り出す．ここでソートされた接尾辞の先頭文字を並べた文字列を F としよう．L と F は次の性質を持つ．以降では文字列 X の i 番目の要素は X[i] と記述する．性質 1: F[i] は元の文字列 S において L[i] の右隣に位置する．性質 2: L と F では，同じ文字の要素の順位が保存される．たとえば図 -1 では A が 4 つあり，L で最初に現れる A（L[0] に相当）は S[9] に対応するが，F においても S[9] に対応する A（F[1] に相当）は最初に出現する A である．L で 2 番目の A（L[4] に相当）は S[3] に対応するが，S[3] に対応する A（F[2] に相当）は F 上でも 2 番目の A となる．これらの性質により L のみから S を復元できるが本稿では説明しない．

FM-index

次の関数 FM を想定する．FM は L 上の位置 i と文字 x が与えられたとき，i 番目以降で最初に x となる要素を見つけ，その要素が F 上の何番目の要

☆5 図の例では S にあらかじめ終端文字 $ が追加されている．

素に対応するか計算する．i 番目以降に x となる要素がない場合は F 上で最後に x が現れる位置の次を返す．FM-index は関数 FM によってクエリを右端の文字から順に絞り込むように検索する．それでは，図 -1 の S から文字列 ATG を検索してみよう．結果は F の区間 $[f, g)$ [☆6] であらわされる．まず，G を検索する．f=FM(0, G), g=FM(L の長さ =11, G) とすれば，f は L で最初に現れる G=L[1]=F[6] を指し，g は最後に現れる G=L[5]=F[8] の次を指す．f=6, g=9 となる．では，文字列 TG が現れる場所を探すにはどうすればよいか．性質 1 を思い出してほしい．F の区間 [6, 9) に対応する S 上の文字はすべて G であるが，L ではその 1 つ前の文字を持つのだから，L の区間 [6, 9) は S 上の文字列 *G に対応している（* は任意の文字を示す）．それではこの区間から TG を探し出すにはどうしたらよいだろうか．ここで f=FM(6, T), g=FM(9, T) を計算すると，f は L の区間 [6, 9) で最初の T に対応する F の要素の位置 (つまり f=9)，g は最後に T が現れる位置の次（つまり g=11）を指す．これで S 上の文字列 TG に対応する F の区間を知ることができた．L の区間 [9, 11) は S 上の文字列 *TG に対応するため，f=FM(9, A), g=FM(11, A) を計算すれば S 上の文字列 ATG に対応する F の区間が分かる．

それでは関数 FM はどのように計算するのか．まず，S について L と F を作成する．次に，L について i 番目よりも前に現れる文字 x の個数（すなわち i 番目以降で最初に現れる x の順位）を事前計算しておく．ゲノムの場合は A, T, G, C それぞれについて計算をする．計算結果は Rank(i, x) と記述する．また，各文字の数も事前計算しておき C(x) と記述する．これらの事前計算から FM を計算できる．FM(i, x) を計算してみよう．F では x は x よりも辞書順で小さい文字が並んだ後に出現する．たとえば，G は $, A, C が順に並んだ後に出現する．そのため G の出現位置は C($)+C(A)+C(C) 以降となる．FM を計算するには連続する G の中で知りたい要素の順位が分かればよい．ここで性質 2 より，F 上の順位は L 上の順位 Rank(i, x) と同一であることがいえる．よって，FM は ΣC(y)+ Rank(i, x) で計算できる．(y は x よりも辞書順で小さい文字とする)．たとえば，FM(6, G) は，C($)+C(A)+C(C)+Rank(6, G)=1+4+1+3=9 と計算できる．FM の計算は S の長さに依存しない．よって，FM-index の計算量は文書サイズに依存しないのである．誌面の都合上，本稿では概略のみを紹介した．詳細は FM-index とその周辺技術に関する優れた書籍 [☆7] を参照していただきたい．

FM-index と類似の索引法はその後も多数発表されている．個人ゲノムデータの蓄積が進んだ近年では，多数の個人ゲノムを参照ゲノムとして利用する試みが注目を集めており，FM-index を有向グラフに拡張した GCSA（Generalized Compressed Suffix Array）や一塩基多型の行列を BWT と類似のデータ構造で索引化する PBWT（Positional Burrows-Wheeler Transform）などが発表されている．さて，世の中には面白いアルゴリズムが多数存在するが，実問題を解くにあたっては単純な方法と泥臭いヒューリスティックでそこそこ良い性能が達成できてしまい，美しい手法の必要性を説明するのに苦労することがしばしばある．FM-index はそのアルゴリズムの鮮やかさもさることながら，新型 DNA シークエンサーのデータ解析というキラーアプリに恵まれた点においても特別だと思うのである．

(2019 年 3 月 14 日受付)

清水　佳奈（正会員） shimizu.kana@waseda.jp

2006 年早稲田大学より博士（工学）取得．同年，産業技術総合研究所入所．2013 ～ 2015 年メモリアルスローンケタリング癌センター客員研究員．2016 年早稲田大学基幹理工学部情報理工学科准教授．2018 年より同学科教授．バイオインフォマティクスの研究に従事．

☆6 左は閉区間，右は開区間とする．

☆7 岡野原大輔：高速文字列解析の世界，岩波書店（2010）．定兼邦彦：簡潔データ構造，共立出版（2018）．