パラメタ化 Burrows–Wheeler 変換の拡張

Eric M. Osterkamp and Dominik Köppl

概 要

Burrows–Wheeler 変化 (BWT) はパターン照合のための簡潔な索引構造として知られている．今回は、パターン照合の２つの特殊な場合に注目する：複数の文字列にたいして円形のパターン照合とパラメタ化のパターン照合である．前者は拡張 BWT (extended BWT)，後者はパラメタ化 BWT (parameterized BWT) と呼ばれる BWT の変種で効率的に行うことができる．自然な拡張は２つのパターン照合の混合である．その斬新な照合問題で，eBWT と pBWT とそれぞれの問題に対している索引構造の合体 epBWT を考案し，パラメタ化円形の照合について，epBWT の性質を解析する．

キーワード： 拡張 BWT，パラメタ化のパターン照合，円形のパターン照合

1 序論

大規模データで検索を効率的に行うため，簡潔な索引構造がよく利用される．簡潔な索引構造の中では、FM-index [4] はよく知られている．FM-index は Burrows–Wheeler 変換 (BWT) から成る．パターン対して，FM-index でパターンの出現の個数を検索できる．count のため， BWT を rank/select データ構造で索引しても十分であるが，count は BWT の中の空間で出現の個数を表現する．必要な道具が少ないため，FM-index の性質は簡潔なデータ構造として相応しい．しかし，FM-index は単純なパターン照合だけ対応できる．

円形のパターン照合は生命情報学と幾何学的計算でよく利用されている [6]．例えば，様々なウイルスの遺伝子のデータは円形の文字列で表現できる．円形の照合とは，テキストの端が繋いでいると考えて，任意の接尾字と接頭辞をパターンの出現の候補と見なす．簡潔な領域を考えて，Mantaci et al. [9] は複数の文字列を提案した拡張 BWT (extended BWT) で索引し，円形のパターン照合を行うことができた．拡張 BWT は複数の問題で応用され [11, 12], 効率的な構築が研究された [2, 3].

その一方で，遺伝子の RNA 解析のために，パラメタ化照合 [1] というパターン照合が必要であると考えられる [13]．また BWT に基づいた索引構造が提案され [5], パラメタ化 BWT (parameterized BWT) と呼ばれる．Mendivelso et al. [10] はパラメタ化のデータ構造を纏めた．近年，Iseri et al. [7] は pBWT の効果的な構築を提案した．

パラメタ化照合は以下のようにパラメタ化一致で定義された．パターンはテキストの部分文字列 S とパラメタ化一致するとは文字の全単射でパターンを S に写像できること．

より詳細には，固定するアルファベットを Σ_s ，置き換えを考えるアルファベットを Σ_p とする．また，パラメタ化文字列を $\Sigma_s \cup \Sigma_p$ 上の文字列とする．パラメタ化一致するとは，二つのパラメタ化文字列 S, T に対し，Σ_p から Σ_p への全単射が存在することをいう．

例 1. $\Sigma_s := \{a, b\}, \Sigma_p := \{A, B\}$ とし，$S :=$ `aABAbA`， $T :=$ `aBABbB` の二つのパラメタ化文字列を考える．`A` について `A` → `B`, `B` → `A` を考えると $S \to$ `aBABbB` $= T$ となるので S, T はパラメタ化一致している．

パラメタ化照合のため，様々の索引が考案されたが,[1] $O(n \lg n)$ ビット領域以下の領域を持つ少メモリーの索引であれば pBWT に限りと考えられる．

[1] Mendivelso et al. [10] はパラメタ化のデータ構造を纏めた．

2　予備知識

Σ をアルファベットとし，Σ^* の要素を**文字列**と呼ぶ．文字列 V の長さを $|V|$ と表記する．文字列 $V = XYZ$ について X, Y, Z をそれぞれ文字列 V の**接頭辞**，**部分文字列**，**接尾辞**と呼ぶ．$V[i]$ は V 中の i 文字目の文字，$V[i..j]$ は V の i 文字目から j 文字目までの部分文字列を表す．$j = |V|$ のとき，$V[i..j]$ は開始位置 i を持つ接尾辞であり，$V[i..]$ とも書く．ϵ は空文字列を示す．

文字列 V に対して，$V = U^k$ を書くと必ず $V = U$ と $k = 1$ である場合，V を**原始的** (primitive) と呼ぶ．任意の文字 V に対して，唯一の原始的な文字列 U と唯一整数 k が $V = U^k$ を満たす．そのとき， U と k をそれぞれ $\mathrm{root}(V)$ と $\exp(V)$ で示す．任意の文字列 V に対して， V^ω は V の無限の連結を示す，すなわち，$V^\omega = VVVV\cdots \in \Sigma^\omega$ である．

2.1　パラメタ化文字列

パラメタ化文字列は固定アルファベット Σ_s と パラメタ化文字列 Σ_p で成り立つ．ただし， $\Sigma_s \cap \Sigma_p = \emptyset$.

パラメタ化照合を効率良く行うための符号化として prev 符号が存在している．Kim と Cho [8] が定義したパラメタ化文字列 V の *prev* **符号** $\langle V\rangle$ は以下の状況を満たす長さ $|V|$ を持つ文字列 $\langle V\rangle \in \Sigma_s \cup [1..|V|-1] \cup \{\infty\}$. すべて $i \in [1..|V|]$ に対して，

$$\langle V\rangle[i] = \begin{cases} V[i], V[i] \in \Sigma_s \quad \text{のとき} \\ \infty, V[i] \in \Sigma_p \wedge V[i] \neq V[j] \forall j \in [1..i-1] \text{ のとき} \\ i - \mathrm{FPQ}_{V[i]}(V[..i-1], i-1)\text{，その他.} \end{cases}$$

この論文では，∞ はすべて整数とアルファベット記号よる大きい記号を示し，$(\{\infty\} \cup \mathbb{N}) \cap \Sigma_s = \emptyset$ を満たす，ただし $\mathbb{N}$ は整数の照合を示す．

2.2　ω 順序のペラメタ化

円形のパターン照合を行うため，Mantaci et al. [9] は文字列の ω 順序を定義した．文字列の ω **順序**を以下のように，パラメタ化文字列に対して拡張できる．

各パラメタ化文字列 V に対して，$\langle V^\omega\rangle := \langle V\rangle \cdot \left(\langle V\rangle[|V|+1..|VV|]\right)^\omega$ と $[\![V^\omega]\!] := [\![V]\!]^\omega$ を定義する．任意のパラメタ化文字列 U と W，$U \preceq_\omega W$ の時かつその時に限り $\langle U^\omega\rangle < \langle W^\omega\rangle$ または $\mathrm{root}([\![U]\!]) = \mathrm{root}([\![W]\!])$. $\preceq_\omega$ はパラメタ化文字列照合 $(\Sigma_s \cup \Sigma_p)^*$ の順序に成り立つ．$U =_\omega W$ の意味は，$U \preceq_\omega W$ かつ $U \neq_\omega W$ となる．

2.3　計算設定

入力は複数の文字列 $T_1, \cdots, T_d$ と設定する．ただし d は入力文字列の個数と n は入力文字列の長さの総和を示す．$T := T_1 \cdots T_d$ は入力文字列を連結，$\mathrm{ENC}_T := [\![T_1]\!] \cdots [\![T_d]\!] \in \left(\Sigma_s \cup [1..\left|\Sigma_p\right|]\right)^n$ はそれぞれのテキストの prev 符号の連結を示す．

以下の 2 つの写像を定義する．

写像 $C : [1..n] \to \{T_j[k..|T_j|]T_j[1..k-1] : k \in [1..|T_j|], j \in [1..d]\}$ は T の位置 $p \in [1..n]$ を循環文字列 $T_j[k..|T_j|]T_j[1..k-1]$ に写像する，ただし $T_j[k..]$ は $T[p..]$ の接頭辞，厳密に言えば $\sum_{x=1}^{j-1}|T_x| < p \leq \sum_{x=1}^{j}|T_x|$ を満たす．

写像 $\mathrm{R}_T : [1..n] \to [1..n]$ は以下の条件を満たす．すべての異なる $i, j \in [1..n]$ に対して，

$$\mathrm{R}_T(i) < \mathrm{R}_T(j) \Leftrightarrow C(i) \prec_\omega C(j) \vee (C(i) =_\omega C(j) \wedge i < j)$$

i	$C(i)$	$\mathrm{ENC}_T[i]$	$\langle C(i)^\omega\rangle[..8]$	$\mathrm{R}_T^{-1}(i)$	$\varphi_T(\mathrm{R}_T^{-1}(i))$	$\mathrm{L}_T[i]$	$\langle C(\mathrm{R}_T^{-1}(i))^\omega\rangle[..8]$
1	AB	2	∞∞222222	7	6	1	ab∞ab3ab
2	BA	2	∞∞222222	8	7	a	b∞ab3ab3
3	AbB	2	∞b∞3b33b	4	3	2	b∞∞b33b3
4	bBA	b	b∞∞b33b3	6	8	b	∞ab3ab3a
5	BAb	2	∞∞b33b33	3	5	2	∞b∞3b33b
6	Aab	1	∞ab3ab3a	10	9	2	∞1∞13131
7	abA	a	ab∞ab3ab	12	11	2	∞1∞13131
8	bAa	b	b∞ab3ab3	5	4	b	∞∞b33b33
9	ABBA	2	∞∞131313	9	10	1	∞∞131313
10	BBAA	1	∞1∞13131	11	12	1	∞∞131313
11	BAAB	2	∞∞131313	1	1	2	∞∞222222
12	AABB	1	∞1∞13131	2	2	2	∞∞222222

図 1: 文字列集合 {AB, AbB, Aab, ABBA} の epBWT.

R_T は順序ランク $\in [1..n]$ を T の位置 $\in [1..n]$ に写像する順列である.

2.4 BWT の backward search でパターン照合

任意のパラメタ化文字列 P 対して，P **空間** とは以上の条件を満たす最大の空間 $[\ell..r]$．各 $i \in [\ell..r]$ に対して $\langle P\rangle$ は $\langle C(\mathrm{R}_T^{-1}(i))^\omega\rangle$ の接頭辞であるような空間のことをいう．ただし，ε 空間は $[1..n]$ とする．

P と T はパラメタ化の文字を両方に持たない場合，任意 $k \in [2..|P|]$ に対して，BWT の LF 写像で $P[k..]$ 空間で $P[k-1..]$ 空間を計算できるため，BWT が P 空間で count を計算できる．

P または T がパラメタ化文字列とすると，BWT のように epBWT で LF 写像に基づいて，P 空間を計算できる．

3 索引構造の定義

提案する epBWT が 3 つの構造 $\mathrm{L}_T, \mathrm{F}_T$ と LCP_T^ω で成り立つ．定義のため，ENC_T が複数の原始的な文字列から成り立つように書き換える．

$$\mathrm{ENC}_T = \mathrm{root}([\![T_1]\!])^{\exp([\![T_1]\!])} \cdots \mathrm{root}([\![T_d]\!])^{\exp([\![T_d]\!])}.$$

各構造が下記のように定義されている．

1. $\mathrm{L}_T \in \left(\Sigma_s \cup [1..\left|\Sigma_p\right|]\right)^n$ は $\mathrm{L}_T[i] := \mathrm{ENC}_T[\varphi_T(\mathrm{R}_T^{-1}(i))]$ をすべて $i \in [1..n]$ に対して満たす文字列．ただし，$\varphi_T : [1..n] \to [1..n]$ は以下の 2 つ条件を満たす.
   (a) 任意の ENC_T 位置 $i \in [1..n]$ に対して，i から ℓ 番目の $\mathrm{root}([\![T_k]\!])$ が始まる (すなわち，$i = 1 + \sum_{x=1}^{k-1} |T_x| + \sum_{y=1}^{\ell-1} |\mathrm{root}([\![T_k]\!])|$) と，$\varphi_T(i) := i + |\mathrm{root}([\![T_k]\!])| - 1$.
   (b) そうでなければ，$\varphi_T(i) := i - 1$ とする.
2. $\mathrm{F}_T \in \left(\Sigma_s \cup [1..\left|\Sigma_p\right|]\right)$ は $\mathrm{F}_T[i] := \mathrm{ENC}_T[\mathrm{R}_T^{-1}(i)]$ で定義され，$\mathrm{F}_T[\mathrm{LF}_T(i)] = \mathrm{L}_T[i]$ をすべて $i \in [1..n]$ に対して満たす.

3. 構造は以下の整数配列 LCP_T^ω である．$\mathrm{LCP}_T^\omega \in ([0..|\Sigma_p|])^n$ は以下の状況を満たす．$\mathrm{LCP}_T^\omega[1] := 0$ とすべて $i \in [2..n]$ に対して，

$$\mathrm{LCP}_T^\omega[i] := \mathrm{lcp}_\omega^\infty\left(C\left(\mathrm{R}_T^{-1}(i-1)\right), C\left(\mathrm{R}_T^{-1}(i)\right)\right).$$

ただし，任意のパラメタ化文字列 U と W に対して，$\mathrm{lcp}_\omega^\infty(U,W)$ は $\langle U\rangle$ と $\langle W\rangle$ の最長共通接頭辞で出現する記号 ∞ の個数を出力する．

例 2. 図 1 は例 $T_1 = \mathtt{AB}, T_2 = \mathtt{AbB}, T_3 = \mathtt{Aab},$ と $T_4 = \mathtt{ABBA}$ に対して以上の定義さるたデータ構造を示す．ただし $\Sigma_s := \{\mathtt{a}, \mathtt{b}\}, \Sigma_p := \{\mathtt{A}, \mathtt{B}\}$.

定義したデータ構造において，FM-index のように検索を行うことができ， count を答えられる．

定理 3. $\mathrm{L}_T, \mathrm{F}_T, \mathrm{LCP}_T^\omega$ とのデータ構造があるとする．任意の文字 $c \in \Sigma_s \cup \Sigma_p$, P 空間 を cP 空間に更新できる．P は最短の入力テキストより短い場合、計算時間は FM-index の計算時間に等しい。

謝辞

本研究は JSPS 科研費 JP23H04378 と JP21K17701 の助成を受けたものです．

参考文献

[1] B. S. Baker. A Theory of Parameterized Pattern Matching: Algorithms and Applications. In *Proceedings of ACM*, pages 71–80. Association for Computing Machinery, 1993.

[2] H. Bannai, J. Kärkkäinen, D. Köppl, and M. Piatkowski. Constructing the Bijective and the Extended Burrows-Wheeler Transform in Linear Time. In *Proceedings of CPM*, volume 191 of *LIPIcs*, pages 7:1–7:16, 2021.

[3] C. Boucher, D. Cenzato, Z. Lipták, M. Rossi, and M. Sciortino. Computing the Original eBWT Faster, Simpler, and with Less Memory. In *Proceedings of SPIRE*, volume 12944 of *LNCS*, pages 129–142. Springer, 2021.

[4] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings of FOCS*, pages 390–398, 2000.

[5] A. Ganguly, R. Shah, and S. V. Thankachan. pBWT: Achieving Succinct Data Structures for Parameterized Pattern Matching and Related Problems. In *Proceedings of SODA*, pages 397–407, 2017.

[6] C. S. Iliopoulos and M. S. Rahman. Indexing circular patterns. In *Proc. WALCOM*, volume 4921 of *LNCS*, pages 46–57, 2008. doi: 10.1007/978-3-540-77891-2_5.

[7] K. Iseri, T. I, D. Hendrian, D. Köppl, R. Yoshinaka, and A. Shinohara. Breaking a Barrier in Constructing Compact Indexes for Parameterized Pattern Matching. *CoRR*, abs/2308.05977, 2023.

[8] S.-H. Kim and H.-G. Cho. Simpler FM-index for parameterized string matching. *Information Processing Letters*, 165:106026, 2021.

[9] S. Mantaci, A. Restivo, G. Rosone, and M. Sciortino. An extension of the Burrows-Wheeler Transform. *Theoretical Computer Science*, 387(3):298–312, 2007.

[10] J. Mendivelso, S. V. Thankachan, and Y. Pinzón. A brief history of parameterized matching problems. *Discrete Applied Mathematics*, 274:103–115, 2020.

[11] N. Prezza, N. Pisanti, M. Sciortino, and G. Rosone. SNPs detection by eBWT positional clustering. *Algorithms for Molecular Biology*, 14, 2019.

[12] A. Salgado, F. Fernandes, and A. T. Freitas. CSA-MEM: Enhancing Circular DNA Multiple Alignment Through Text Indexing Algorithms. In *Proceedings of ISBRA*, volume 14248 of *LNBI*, pages 509–517, 2023.

[13] T. Shibuya. Generalization of a Suffix Tree for RNA Structural Pattern Matching. *Algorithmica*, 39(1):1–19, 2004.