

パラメタ化 Burrows-Wheeler 変換の拡張

Eric Osterkamp (University of Münster)

クップル ドミニク (山梨大学)

研究背景

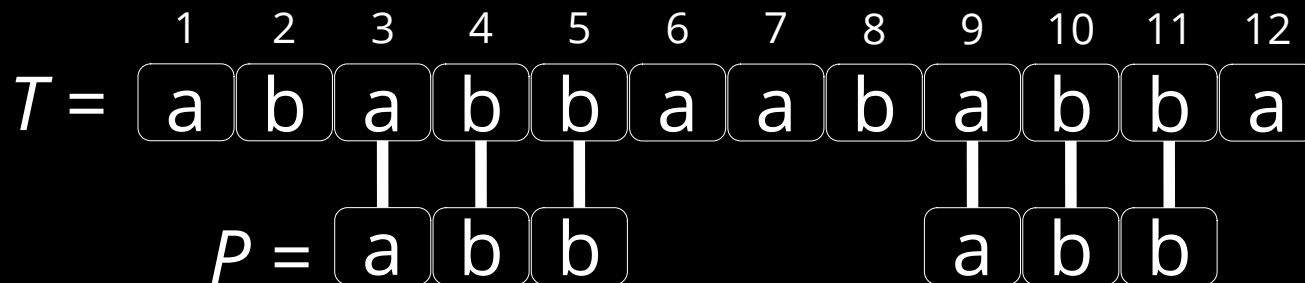
パターン照合問題

- T : テキスト
- P : パターン
- 目的: T の中に P の出現位置を検索する

⇔ つまり、 P とマッチする T の部分文字列を数えたい

パターン照合の例

- Σ : アルファベット, 例: $\Sigma = \{a, b\}$
- $T = ababbaababba$
- $P = abb$
- $T[3..5] = T[9..11] = abb = P$ とマッチする
- 問題: T の中に P は何回出現するの?



答え: 2回!

p マッチ [Baker '93]

パターン照合問題のひとつの一般化はパラメタ化

- そのときアルファベット Σ は2つのアルファベット Σ_s と Σ_p から成り立つ
- $\Sigma := \Sigma_s \cup \Sigma_p$ ただし、 $\Sigma_s \cap \Sigma_p = \emptyset$
 - Σ_s を定数アルファベット (static) と
 - Σ_p をパラメタ化アルファベット (parameterized) を呼ぶ

p マッチ [Baker '93]

- $\Sigma := \Sigma_s \cup \Sigma_p$ ただし、 $\Sigma_s \cap \Sigma_p = \emptyset$
- 2つ文字列 X, Y は p マッチするのは、以下の状況を満たす $f: \Sigma_p \rightarrow \Sigma_p$ の順列が存在する
 - $|X| = |Y|$ (X と Y は同じ長さ)
 - $X[i] \in \Sigma_s$ の場合、 $X[i] = Y[i]$
 - $X[i] \in \Sigma_p$ の場合、 $X[i] = f(Y[i])$
- そのとき、 $X =_p Y$ を書く

p マッチの例

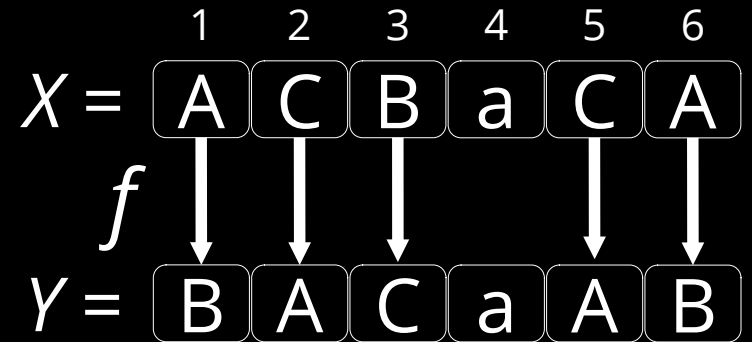
例

- $\Sigma_s = \{a\}, \Sigma_p = \{A, B, C\}$

- $X = ACBaCA$

- $Y = BACaAB$

- $f: A \mapsto B, B \mapsto C, C \mapsto A$ によって $X =_p Y$

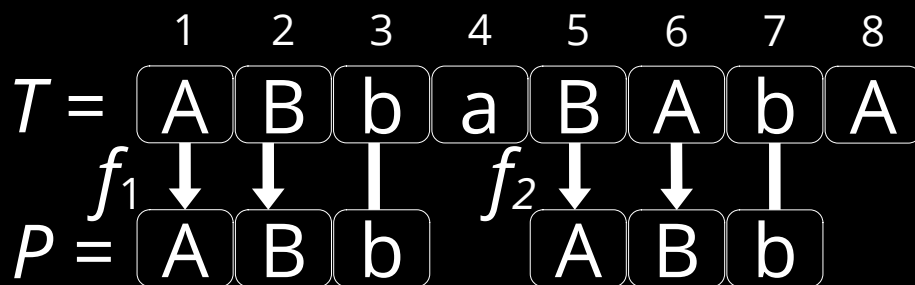


パラメタ化パターン照合問題 (PPM)

- $T[1..n], P[1..m] \in \Sigma^*$
- n は T の長さ、 m は P の長さ
- すべて $T[i..i+|P|-1] =_p P$ を満たすテキスト位置 $i \in [1..n]$ の個数を求める (count query)
(ただし各位置に対して、新しい順列 f を選べる)
- パラメタ化パターン照合問題を PPM (parameterized pattern matching) で省略する

パラメタ化パターン照合問題 (PPM)

- 例: $\Sigma_s = \{a,b\}, \Sigma_p = \{A, B\}$
- $T = ABbBAbAaABBA$
- $P = ABb$
- $T[1..3] =_p T[5..7] =_p ABb = P$ と p マッチする



$$f_1 = \text{id}$$

$$f_2 : A \mapsto B, B \mapsto A$$

PPM の既存研究

データ構造	PPM 時間	引用
suffix tree	$O(m \log \sigma)$	[Baker '93]
suffix array	$O(m + \log n)$	[Deguchi + '08]
position heap	$O(m \log \sigma + m\sigma_p)$	[Diptarama+ 17]
suffix tray	$O(m + \log \sigma)$	[Fujisato+ 21]
DAWG	$O(m \log \sigma)$	[Nakashima+ 22]

$\sigma := |\Sigma|$, アルファベットのサイズ

$\sigma_p := |\Sigma_p|$

$n := |T|$, テキストの長さ

$m := |P|$, パターンの長さ

すべてのデータ構造の
領域は

$O(n \log n)$ ビット

既存研究：省メモリー

- パラメタ化 Burrows-Wheeler transform (pBWT) [Ganguly+ '17]
 - $n \lg \sigma + O(n)$ ビット領域
 - $O(m \log \sigma)$ 時間で PPM を計算できる
- pBWT の簡略化 [Kim, Cho '21]
 - $2n \lg \sigma + O(n)$ ビット領域
 - 入力のビット表現について線形的な領域を取る

PPM の応用

万能な応用:


- 情報検索
- ソフトウェアのメンテナンス
- 剽窃の検出
- 遺伝子構造の解析

応用：重複のコードを発見

下記は X11 のソースコードの2つ場所：

[Baker '97]

```
...                               ...
*pmi++ = *pma++;                 *pmi++ = *pma++;
copy_number(&pmi,&pma,            copy_number(&pmi,&pma,
  pfi->min_bounds.lbearing,      pfh->min_bounds.left,
  pfi->max_bounds.rbearing);     pfh->max_bounds.left);
*pmi++ = *pma++;                 *pmi++ = *pma++;
...                               ...
```



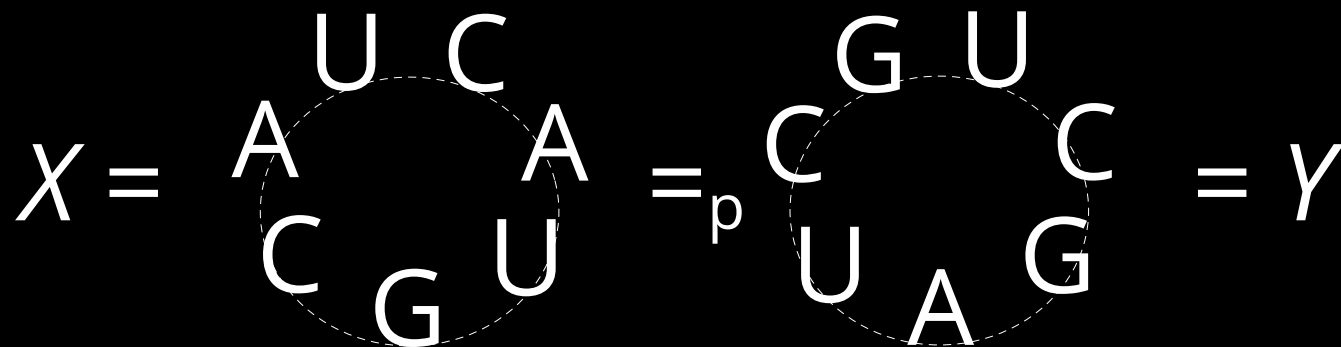
$\Sigma_p = \{pfi, pfh, lbearing, rbearing, \dots\}$ を文字列アルファベットに見なすと

f は以下のように左側を右側へ写像できる：

- $pfi \mapsto pfh$
- $rbearing \mapsto right$
- $lbearing \mapsto left$

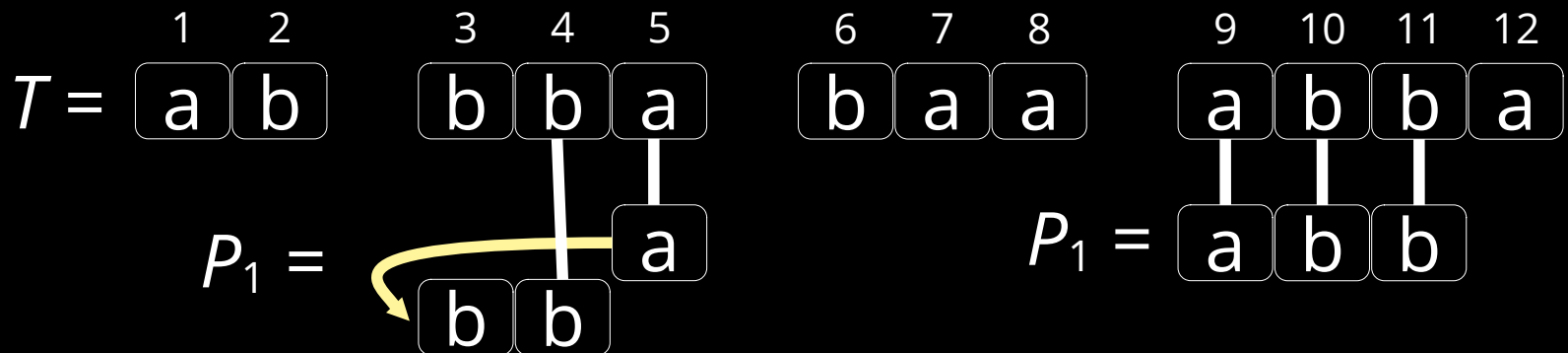
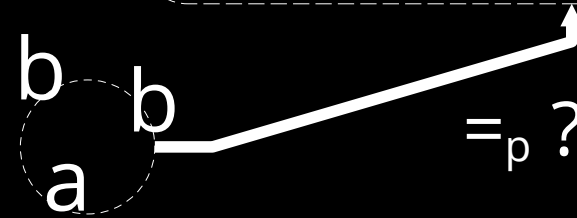
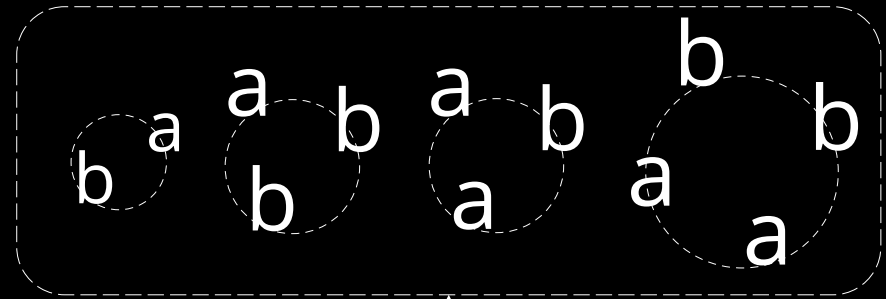
応用： バイオインフォマティクス

- RNA の塩基対関係 f : [Shibuya' 04]
- $X = \text{AUGCAUC}$
- $Y = \text{CGAUCGU}$
- $f(X) = Y$
- f :
 - $A \mapsto C$,
 - $U \mapsto G$,
 - $C \mapsto U$,
 - $G \mapsto A$
- しかし、RNA で円形のパターン照合する希望がある!



円形のパターン照合 (CPM)

- $T = \{ab, bba, baa, abba\}$
- $P_1 = abb$
 $(=_c T[3..5] =_c T[9..11])$
- $=_c$ は円形的に一致することを示す
 (c : circular)



CPM の既存研究

CPM の索引構造

$\sigma := |\Sigma|$, アルファベットのサイズ
 $n := |T|$, テキストの長さ
 $m := |P|$, パターンの長さ

一つのテキスト [Booth '80]

データ構造	CPM 時間	引用
suffix automaton	$O(m \lg \sigma)$	[Lothaire '05]
suffix tree	$O(m \lg n)$	[Iliopoulos, Rahman '08]
suffix tree	$O(m \lg \sigma)$	[Jin, Adjeroth '11]
suffix array	$O(m \lg n)$	[Iliopoulos+ '17]

すべてのデータ構造の領域は $O(n \log n)$ ビット

CPM count の既存研究

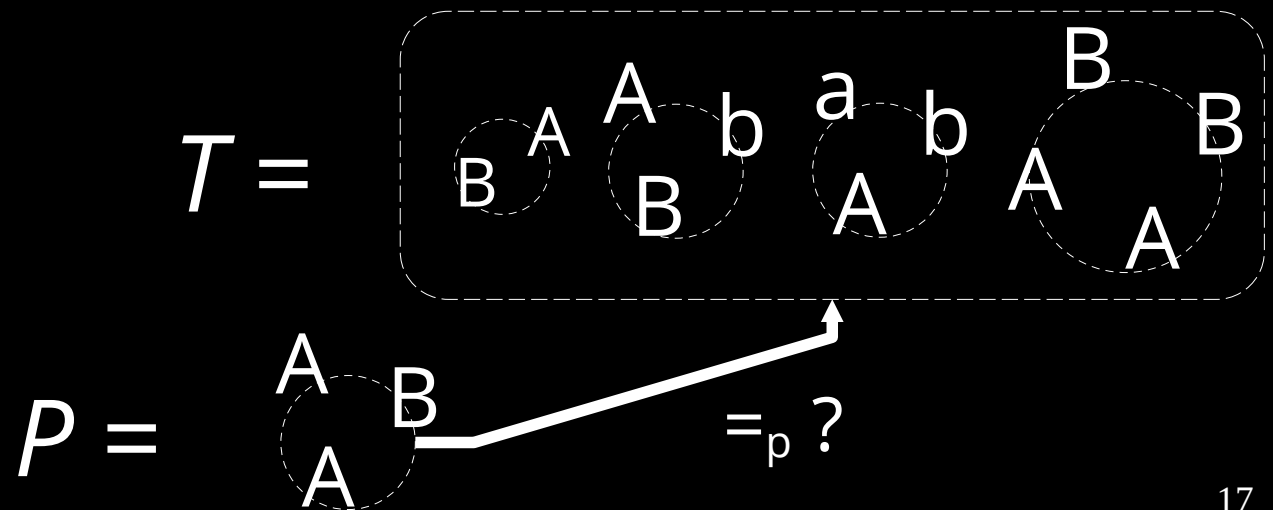
- 複数のテキストに対して CPM 照合
 - extended BWT [Mantaci+ '07]
 - 領域・時間は不明
 - circular BWT [Hon+ '11]
 - $n \log \sigma (1 + o(1)) + O(n) + O(d \log n)$ ビット
 - $O(m \log \sigma)$ 時間で CPM count
- d : テキストの個数
 n : すべてのテキストの文字の個数の総和

今回の話題

CPM \cap PPM =

円形のパラメタ化パターン照合問題

- 複数のテキストに対して CPM \cap PPM を計算

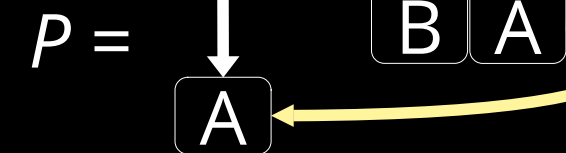
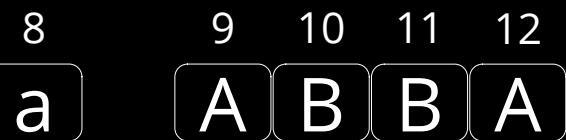
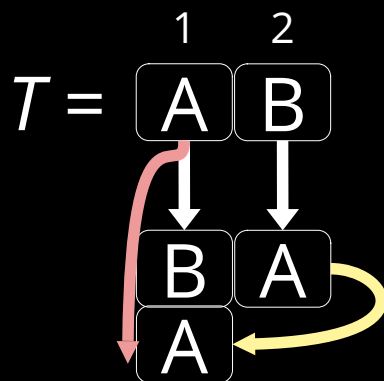
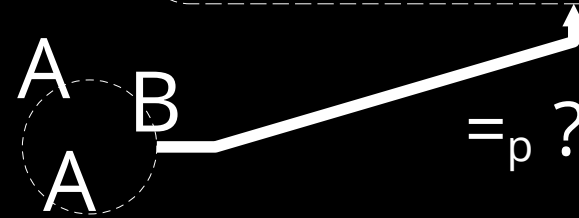
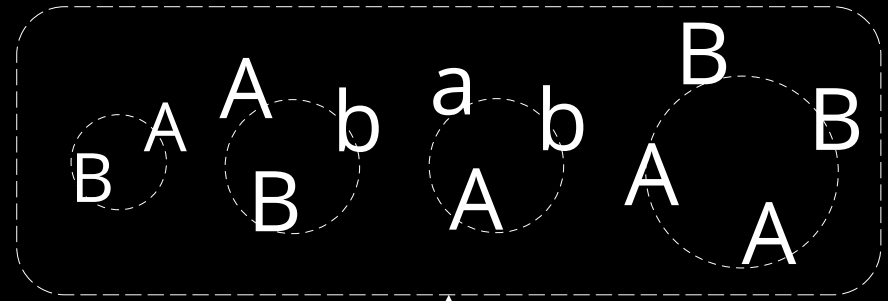


円形のパラメタ化パターン照合: 例

- $T = \{AB, bBA, bAa, ABBA\}$

- $P = BAA$

($=_{cp} T[9..11] =_{cp}$
 $T[11..12] T[9]$)



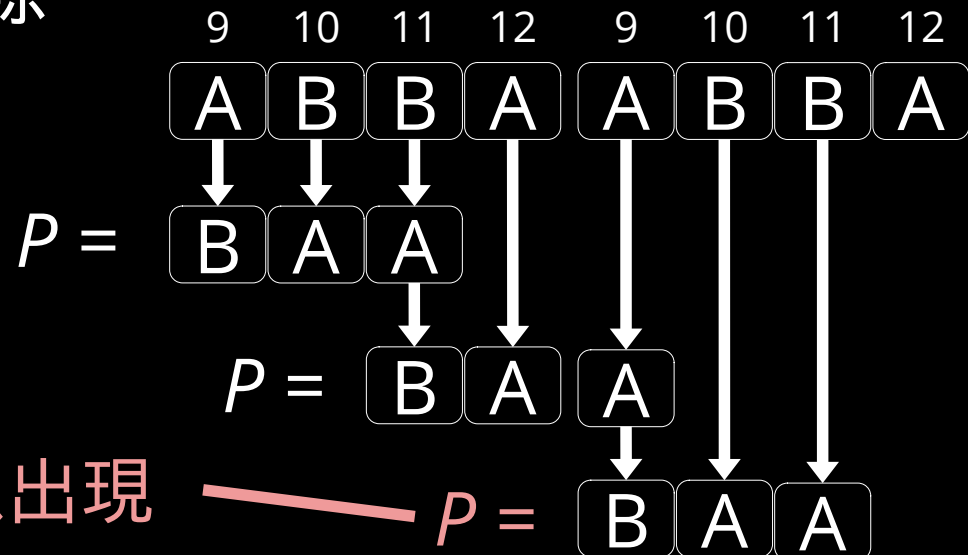
本研究の方針

- [Kim, Cho '21] の pBWT を CPM 問題に拡張する
- CPM 問題の方針: P を円形的にテキスト T の中に検索したい場合は、
 - 1) 厳密パターン照合を連結 $T \cdot T$ を行ったあと、
 - 2) 数えすぎた出現を削除

例:

• $T = ABBA$

• $P = BAA$



pBWT から epBWT までへ

pBWT をよりわかりやすく定義するため、
2つの文字列の符号化を提案する

1) prev 符号化 $\langle T \rangle$ [Baker '93]

2) 橋本符号化 $\llbracket T \rrbracket$ [Hashimoto+ '22]

prev 符号化 $\langle T \rangle$

- prev 符号化 $\langle T \rangle$ は任意の文字列 T に対して、すべての Σ_p の文字を前の出現の距離に変更する
ただし、最差の出現を ∞ で変更
- $\langle T \rangle[i] = T[i]$ ただし、 $T[i] \in \Sigma_s$ の場合
- $\langle T \rangle[i] = \ell, T[i] \in \Sigma_p \wedge T[i - \ell] = T[i] \wedge T[i] \notin T[i - \ell + 1..i - 1]$ の場合
- $\langle T \rangle[i] = \infty$ (その他)

$T = \text{ABbBAAbAaABBA}$

例: $\langle T \rangle = \infty\infty\text{b}24\text{b}2\text{a}2613$
 $\langle T[2..] \rangle = \infty\text{b}2\infty\text{b}2\text{a}2613$

橋本符号化 « T »

- « T »[i] = T [i] ただし、 T [i] $\in \Sigma_s$ の場合
- « T »[i] = c ただし、 T [i] $\in \Sigma_p$ かつ
($T \cdot T$)[j] = T [i] を満たす最小 $j \in [i+1..]$ として
($T \cdot T$)[$i..j$] の中に出現する異なる Σ_p の文字の個数は c
- 例: $T = \text{ABbBAbAaABBA}$,
« T » = 21b21b1a2121
« $T[2..]$ » = 1b21b1a2122

定理: $X =_p Y \Leftrightarrow \langle X \rangle = \langle Y \rangle \Leftrightarrow \llbracket X \rrbracket = \llbracket Y \rrbracket$

pBWT

- $RA : [1..n] \rightarrow [1..n]$ 順列 : $\langle T[RA[1]..] \rangle < \langle T[RA[2]..] \rangle < \dots < \langle T[RA[n]..] \rangle$
- $pBWT[i] := \langle T[RA[i]..] \cdot T[1..RA[i]-1] \rangle$ は長さ n を持つ文字列
- 観察: pBWT で CPM を行いたい場合、 $\langle T \rangle$ の代わりに $\langle T \cdot T \rangle$ を索引しても良い
- 素朴な方法は領域を2倍増やす!

まとめ

$\sigma := |\Sigma|$, アルファベットのサイズ
 $n := |T|$, テキストの長さ
 $m := |P|$, パターンの長さ

- CPM n PPM 問題について索引構造 epBWT を提案した
- $2n \lg \sigma + O(n)$ ビット領域
- $O(m \log \sigma)$ 時間で CPM n PPM count を計算できる

方針

- [Kim, Cho '21] の pBWT に基づいて
- 順列 RA の代わりに [Mantaci+ '07] の extended BWT の順序でソートする