アルゴリズム 78−8
（2001．5．18）

ハイブリダイゼーションによるDNA配列決定と文字の分布との関係

土井 晃一郎

東京大学大学院新領域創成科学研究科複雑理工学専攻
〒113-0033 東京都文京区本郷 7-3-1
doi@gi.k.u-tokyo.ac.jp

DNA 配列を決定することはゲノムサイエンスにおいて大変重要な問題である。配列を決定する方法は様々な方法が考えられている。ここではシーケンシング バイ ハイブリダイゼーション (SBH) と呼ばれる DNA チップによって得られる部分文字列の有無のデータによる配列決定の方法について扱う。従来はこの手法の評価にランダムな配列に対しての特定できる割合を考えられていた。本論文では i.i.d. 配列に対して解析を行い、文字の分布の偏りに従い、配列決定の確率が下がっていく傾向があることを理論的に示した。

A relationship between DNA sequencing by hybridization and distribution of the characters

Koichiro Doi

Department of Complexity Science and Engineering,
Graduate School of Frontier Science, University of Tokyo
7–3–1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
doi@gi.k.u-tokyo.ac.jp

Sequencing DNA sequence is a very important problem in genome science. Various sequencing methods were considered. We can get information about the existence of subsequences in DNA sequence by the DNA chip. This paper deals with sequencing by the hybridization data from DNA chip, called Sequencing by Hybridization (SBH). Previous researches assumed that DNA sequences are the same as random sequences for the analysis of SBH. This paper shows the rate of reconstruction of the SBH algorithms for the sequences generated by i.i.d.source is correlated with the bias of distribution of the characters in the sequences.

1 はじめに

DNA 配列のシーケンシング、つまり、DNA 配列がなんであるかを決定していくことはゲノムプロジェクトの中心的な課題であり。DNA 配列決定に対しては様々な方法が考えられ、生物実験も数多く行われてヒトゲノムをはじめとして多くの DNA 配列が決定されてきている。

ここで扱う方法はシーケンシング バイ ハイブリダイゼーション (以下、SBH) と呼ばれている手法である [4, 5, 6, 9]。 この方法は DNA チップによって得られるデータを使用して DNA 配列を決定する手法である。DNA チップとはプローブと呼ばれる短い配列をたくさんくっつけたものであり、ハイブリダイゼーションを行うことにより、DNA 配列上にプローブとマッチしている配列が存在するかどうかが分かる。配列決定をしたい DNA 配列とハイブリダイゼーションを行うこと

により DNA 配列中の部分文字列が分かるので、それによって DNA 配列を決定していく手法である。この手法は、任意のある長さの DNA 配列を高い確率でいかに少ない数のプローブで配列決定ができるかという理論的な解析がなされてきたが、実際には生物実験上のエラーなどの問題があり実用化はあまりなされていなかった。この手法には、どのようなプローブを DNA チップ上におくかというプローブの設計の問題と、チップから得られたデータに対してどのようなアルゴリズムを適用して配列を決定するかという配列決定問題の 2 つの問題がある。この論文では後者について考えている。

既存研究としては、長さ k のプローブをすべて DNA チップ上におく方法が考えられたのが始まりである [5, 6, 9]。つまり、DNA 配列は $\{A, T, G, C\}$ の 4 種類の文字からなる文字列なので 4^k のプローブを使用するものである。これについては詳しい説明ははぶくが、ハミルトンパスが 1 つだけか判定する問題となり、それをオイラーパスが 1 つしかだけしかないか判定する問題に変換できて計算量を減少させることができるという理論的にはきれいなものであるが、冗長性が高く、効率の良いものではないことが理論的にも湿されている ([1, 3])。

この後、$\{A, T, G, C\}$ だけではなく、A, T, G, C どれにもマッチする universal base、A, T だけにマッチする文字などを使用することによってチップの効率を理論的にあげようとする試みがなされた [6, 9]。 この場合にははじめに考えられたチップのようなハミルトンパスといった性質が考えられないので、配列決定をしようとする DNA 配列のはじめの部分はすでに分かっているものと仮定して、そこから 1 つ 1 つ配列を決定していこうとするアルゴリズムで性質が調べられた。これにより、単なる部分文字列を使ったものよりは効率の良いものとなったが、それほど効率のよいものではなかった。

そして最近、[7, 8] で新しいチップとそのデータに対する新しい配列決定アルゴリズムが提唱され、配列決定の能力がかなり伸びて、その情報理論的な最適性も示されている。

しかしながら、今までの研究では A, T, G, C のどの文字も等確率に出てくるようなランダムな配列に対して理論的な解析が行われてきた。この論文では DNA 配列上の文字が i.i.d. 情報源によって生成されると仮定して理論的な解析を行う。

2 準備

まず、この論文で対象とする文字列、DNA チップ、配列決定アルゴリズムについて説明する。ここでは DNA 配列を対象に考えているので、$\Sigma = \{A, T, G, C\}$ の 4 文字からなる文字列を対象とする。

2.1 文字列

本論文では以下のような文字列に対して解析を行っていく。

定義 1 (i.i.d. 配列) 我々が対象としている文字列は定常無記憶情報源 *(i.i.d. source, independently and identically distributed source)* から生成される文字列、つまり、周りの文字とは独立に文字の出現確率分布が定まっている文字列であり、各文字の出現確率は Σ 上の離散確率分布 $p(x)\ x \in \Sigma$ にしたがっている。それをここでは *i.i.d.* 配列と呼ぶ。

この i.i.d. 配列における文字の分布を表すためにエントロピーレートを定義する。

定義 2 (エントロピーレート) *i.i.d.* 配列 S に対するエントロピーレートは

$$H(S) = -\sum_{x \in \Sigma} p(x) \log_2 p(x)$$

である。

また、文字の分布において以下の GC 含量というものが DNA 配列の特徴として大変重要なものであるのでこれも本論文中で扱うことにする。

定義 3 (GC 含量) *GC* 含量とは *DNA* 中の *G,C* の 2 種類の文字の出現する割合である。

2.2 DNA チップ

この論文で扱う DNA チップは以下のようなものである。このチップのデザインは Preparata et al. [8] においてなされたものであり、すべての長さ k の文字列を使った場合を含んだデザインになっている。DNA 配列は A,T,G,C の 4 種類からなっているが、ここではそれに加えて* で表す don't care のように A,T,G,C の 4 種類のどれが来てもマッチする文字を使用して DNA チップを作成する。この*にあたるものは実際の生物実験でも作成することができる。

定義 4 ([7, 8]) 2 つの正の整数のパラメータ s, r を用いて (s, r) で表される *DNA* チップは以下のようなものである。
(s, r)-$probe = X^s(*^{s-1}X)^r$
X は A, T, G, C の 4 種類それぞれに対して全種類のプローブを作成することを表している。

よってチップのサイズは 4^{s+r} である。

例 1 $(3, 4)$ の場合には以下のようになる。
$(3, 4)$-$probe = XXX**X**X**X**X**X$

2.3 配列決定アルゴリズム

ここでは 3 種類の配列決定アルゴリズムを扱う。1 つは Pevzner[6] により、考えられた配列決定アルゴリズムで残りの 2 つはそれを拡張したものである。これらの配列決定アルゴリズムの入出力は以下のようになる。

入力 (s, r)-probe のうちハイブリダイゼーションをしたプローブの集合 S と決定すべき DNA 配列の長さ n と決定すべき DNA 配列の $s(r + 1)$ の長さの正しい接頭辞

出力 DNA 配列が決定できたときにはその DNA 配列、決定できなかったときには決定できたところまでの配列

エラーのないときは入力はすなわち、決定すべき DNA 配列のなかに存在する (s, r) の形をした部分文字列すべての集合ということになる。また、ここで考えられているアルゴリズムは DNA 配列すべてを特定するものではなく決定すべき DNA 配列の $s(r + 1)$ の長さの正しい接頭辞が与えられているときに残りの DNA 配列を決定するアルゴリズムとなっている。
これに関して以下のような 3 種類のアルゴリズムを本論文では考える。

単純アルゴリズム

1. 1 番目から ℓ 番目までの配列が決定されているときに $\ell + 1$ 番目の文字を特定することを考える。$\ell + 1$ 番目に右端が来るようにおいて、$\ell - s(r + 1) + 2$ 番目から ℓ 番目のすでに決定されている配列にちゃんとマッチするプローブの集合 C を S の中からすべてとってくる。

2. $|C| = 1$ のときには $\ell + 1$ 番目の文字は $|S|$ に含まれるプローブの右端の文字となる。この文字を $\ell + 1$ 番目の文字と決定して、$\ell = \ell + 1$ とする。してステップ 1 に戻る。$|C| > 1$ のときには特定出来なかったとして終了する。

このアルゴリズムは１つ１つ端から順に文字を決定していく方法である。gap を入れないある長さの文字列すべてをプローブとして使うような DNA チップに対してはオイラーパス問題に変換して解くことが考えられてたが、そうでないものに対してはこのようなアルゴリズムがまず考えられていた。

基本アルゴリズム ([8])

1. 1 番目から ℓ 番目までの配列が決定されているときに $\ell+1$ 番目の文字を特定することを考える。$\ell+1$ 番目に右端が来るようにおいて、$\ell-s(r+1)+2$ 番目から ℓ 番目のすでに決定されている配列にちゃんとマッチするプローブの集合 C を S の中からすべてとってくる。

2. $|C|$ の数にしたがって以下のことを行う。

   - $|C|=1$ のときには $\ell+1$ 番目の文字は $|S|$ に含まれるプローブの右端の文字となる。この文字を $\ell+1$ 番目の文字と決定して、$\ell=\ell+1$ とする。してステップ 1 に戻る。
   - $|C|>1$ のときには C の中のこの位置にマッチするプローブの集合それぞれに対して、幅優先探索のやり方で $\ell-sr+2$ 番目から $\ell+s+1$ 番目までの位置にマッチするプローブの候補、$\ell-s(r-1)+2$ 番目から $\ell+2s+1$ 番目までの位置にマッチするプローブの候補というように探索していく。この幅優先探索は最大深さ $rs+1$ まで行う。
     - 途中で枝が 1 本になれば配列はその枝に DNA 配列が決定され、引き続いてまだ決定されていない部分をステップ 1 に戻って決定していく。
     - 深さ $rs+1$ まで行ってまだ枝が複数存在するときには、その最大一致接頭辞が存在すればその部分の DNA 配列が決定され、引き続いてまだ決定されていない部分をステップ 1 に戻って決定していく。
     - 最大一致接頭辞が存在しなければ配列は決定出来なかったとして終了する。

3. 長さ n まですべて決定できたところで終了する。

このアルゴリズムは [6] でも考えられているような基本的に１つ１つ順番に配列の最初の方から決定していくアルゴリズムである。しかし、それだけでは次にどのプローブが来るのか分からない状況がすぐおこってしまうので、ここではプローブの候補が複数考えられる場合にその先まで読み進めることによって、次に何が来るのか分からないような状況を回避しようとしている。また、このようなアルゴリズムを行うのに適しているようにプローブも設計されている。

拡張アルゴリズム ([7])

1. 最初に基本アルゴリズムを実行する。

2. この基本アルゴリズムが成功すれば配列が特定できて実行終了。失敗したときにはさらに深さ h まで幅優先探索を行っていく。

このアルゴリズムは基本アルゴリズムが失敗したときにさらに先読みを勧めることによってアルゴリズムの成功確率をあげようとするものである。

3　解析

P_i $(i=2,3,...)$ を i.i.d. 配列上の i 個の文字が一致する確率する確率だとする。

$$P_i=\sum_{x\in\Sigma}p(x)^i$$

もし、すべての文字 (A,T,G,C) が $\frac{1}{4}$ の確率で現れているならば、$P_i = (\frac{1}{4})^{i-1}$. $P_{i+1} = \frac{1}{4}P_i$ が成り立つ。しかし、一般にはこの性質は成り立たず、この P_i の大きさが、今後の解析において重要なものとなる。これからの解析では、ある文字列に対する配列決定アルゴリズムの成功確率を示していくのではなく、エントロピーレートの変化に対して、配列決定アルゴリズムの成功確率がどのように変化していくのかを示していくことになる。

最初に i.i.d. 配列における確率分布 $p(x)$ の変化に伴ういくつかの変化を以下の 2 つの補題で示す。

補題 1 一定の長さ m の文字列 S が、$x \in \Sigma$ に対して、$p(x) < \frac{1}{2}$ となっている $p(x)$ に従う定常無記憶情報源によって生成だれた文字列であるとする。このとき、確率分布 $p(x)$ の微少変化に対してエントロピー $H(S)$ は減少し P_i は増加する。その微少変化とは以下の式で表されるものである。

$$\begin{aligned} p'(a) &= p(a) + \varepsilon, \\ p'(b) &= p(b) - \varepsilon, \\ p'(x) &= p(x) \quad x \neq a, b, \\ p(a) &\geq p(b) >> \varepsilon > 0. \end{aligned}$$

補題 1 の証明

文字列 S' を確率分布 $p'(x)$ に従う定常無記憶情報源から生成される i.i.d. 配列であるとする。同様に $P'_i = \sum_{x \in \Sigma} p'(x)^i$ と定義する。ε は微少変化であるのでテイラー展開を使って変化量を近似することが出来る。エントロピーレートの変化量は以下のようになる。

$$\begin{aligned} H(S') - H(S) &= -((p(a)+\varepsilon)\log_2(p(a)+\varepsilon) - p(a)\log_2(p(a))) \\ &\quad +(p(b)\log_2(p(b)) - (p(b)-\varepsilon)\log_2(p(b)-\varepsilon)) \\ &\approx -\varepsilon\log_2(p(a)) - \varepsilon/\ln 2 + \varepsilon\log_2(p(b)) + \varepsilon/\ln 2 \leq 0 \end{aligned}$$

同様に P_i の変化量も計算してやる。

$$\begin{aligned} P'_i - P_i &= (p(a)+\varepsilon)^i - p(a)^i - (p(b)^i - (p(b)-\varepsilon)^i) \\ &\approx i\varepsilon p(a)^{i-1} - i\varepsilon p(b)^{i-1} \geq 0 \end{aligned}$$

よって、$H(S)$ は減少、P_i は増加する。 □

補題 2 一定の長さ m の文字列 S が、$x \in \Sigma$ に対して、$p(x) < \frac{1}{2}$ となっている $p(x)$ に従う定常無記憶情報源によって生成だれた文字列であるとする。このとき、確率分布 $p(x)$ の微少変化に対して $P'_i - P'_{i+1}$ は増加する。その微少変化とは以下の式で表されるものである。

$$\begin{aligned} p'(a) &= p(a) + \varepsilon, \\ p'(b) &= p(b) - \varepsilon, \\ p'(x) &= p(x) \quad x \neq a, b, \\ p(a) &\geq p(b) >> \varepsilon > 0,\ p(a) + p(b) < \frac{2}{3}. \end{aligned}$$

補題 2 の証明

補題 1 と同様に変化量を計算してやると、

$$
\begin{aligned}
& P_i' - P_{i+1}' - (P_i - P_{i+1}) \\
\approx\ & i\varepsilon(p(a)^{i-1} - p(b)^{i-1}) - (i+1)\varepsilon(p(a)^i - p(b)^i) \\
=\ & \varepsilon(p(a)-p(b))(i(p(a)^{i-2} + p(a)^{i-3}p(b) + \cdots + p(b)^{i-2}) - (i+1)(p(a)^{i-1} + \cdots + p(b)^{i-1}) \\
=\ & \varepsilon(p(a)-p(b))((i-(i+1)p(a))(p(a)^{i-2} + p(a)^{i-3}p(b) + \cdots + p(b)^{i-2}) - (i+1)p(b)^{i-1}) \\
\geq\ & \varepsilon(p(a)-p(b))((i-(i+1)(p(a)+\frac{b}{i-1}))(p(a)^{i-2} + p(a)^{i-3}p(b) + \cdots + p(b)^{i-2}) \\
& -(i+1)p(b)^{i-1}) \\
\geq\ & 0.
\end{aligned}
$$

よって、$P_i - P_{i+1}$ は増加している。

□

これら 2 つの補題を使って、単純アルゴリズムにおける配列決定の成功確率の変化に関する定理を証明する。

定理 1 一定の長さ m の文字列 S が、$x \in \Sigma$ に対して、$p(x) < \frac{1}{2}$ となっている $p(x)$ に従う定常無記憶情報源によって生成された文字列であるとする。このとき、確率分布 $p(x)$ の微少変化に対して文字列 S の単純アルゴリズムの DNA 配列の特定率は減少する。その微少変化とは以下の式で表されるものである。

$$
\begin{aligned}
p'(a) &= p(a) + \varepsilon, \\
p'(b) &= p(b) - \varepsilon, \\
p'(x) &= p(x) \quad x \neq a, b, \\
p(a) &\geq p(b) >> \varepsilon > 0\ (p(a) + p(b) \leq \frac{2}{3}).
\end{aligned}
$$

定理 1 の証明

離散確率分布 $p'(x)$ に従って定常無記憶情報源から生成される文字列を S' とする。S と S' とのエントロピーレートの変化を計算する。

単純アルゴリズムが失敗する場合は、ある 2 つのプローブが対象 DNA 配列にマッチし、そのプローブの 最後の文字以外は同じ文字で最後の文字だけ異なるような場合である。

ここでは以下の 2 つの場合について考える。

最初に 2 つのプローブが対象 DNA 中で重なり合わない場合である。このとき、2 つのプローブの選択の数は $\binom{(m-s(r+1)+1)}{2} - (m - s(r+1) + 1)$ 以下になる。ある 2 つの位置 $i, j (i + s(r+1) \leq j)$ にマッチするプローブが最後の文字以外は同じ文字で最後の文字だけ異なる確率は

$$P_2^{s+r-1}(1 - P_2)$$

である。

よって、この場合の確率は

$$\left(\binom{(m - s(r+1) + 1)}{2} - (m - s(r+1) + 1)\right) P_2^{s+r-1}(1 - P_2)$$

である。

2 番目の場合は注目する 2 つのプローブが対象 DNA 配列中で重なりがある場合である。この場合にはプローブ $i, j\ (1 < i < j < m - s(r+1))$ に対して以下の 3 つの場合を考える。

Case 1: プローブ i の最初の s 文字がプローブ j と重なる場合 $(0 < j - i < s)$

$j - i \leq \lceil \frac{s}{2} \rceil$ の場合、 $P_{s+1}P_2^{r-1}(1-P_2) + P_{\lfloor \frac{s+2}{2} \rfloor}P_{\lceil \frac{s+2}{2} \rceil}P_2^{r-1}(1-P_2) + \cdots$

$j - i > \lceil \frac{s}{2} \rceil$ の場合、 $\sum_{j=i+\lceil \frac{s}{2} \rceil + 1}^{s}(P_3)^{s-(j-i)}P_2^{j-i-(s-(j-i))}P_2^{r-1}(1-P_2)$

Case 2: プローブ i の中の最後の r 文字がプローブ j と重なりがある場合 $(j = i + hs, (1 \leq h \leq r))$

$h \leq \lceil \frac{r}{2} \rceil$ の場合、 $P_2^{s-1}(P_{r+1} - P_{r+2}) + P_2^{s-1}P_{\lfloor \frac{r+3}{2} \rfloor}(P_{\lceil \frac{r+3}{2} \rceil - 1} - P_{\lceil \frac{r+3}{2} \rceil}) + \cdots$

$h > \lceil \frac{r}{2} \rceil$ の場合、 $\sum_{h=\lceil \frac{r}{2} \rceil + 1}^{r} P_2^{s-1}(P_3)^{r-h}P_2^{h-(r-h)-1}(P_2 - P_3)$

Case 3: それ以外の場合、 $(s-1)(r-1)P_2^{s+r-3}P_3(1-P_2) + (s-1)P_2^{s+r-2}(P_2 - P_3)$

ここで重要なのは上記の確率中の項はすべて P_i、 $P_i - P_{i+1}$、 $1 - P_2$ の 3 つの項の組み合わせで構成されていることである。$1 - P_2$ 以外の式はすべて微少変化 ε に対して増加する。 (Lemma 1、 2 より) $1 - P_2$ はその逆で減少するが、しかし、$P_2(1-P_2)$ という式を考えれば $P_2(1-P_2)$ は増加する。なぜなら $P_2 < \frac{1}{2}$ だからである。よって、すべての項が微少変化 ε に対して増加する。よって、失敗する確率は増加する。

□

基本アルゴリズムと拡張アルゴリズムに対しても同様の性質が成り立つ。証明は省略するが、定理 1 と同じように確率のそれぞれの項が増加するのか調べていけばよい。

系 1 一定の長さ m の文字列 S が、$x \in \Sigma$ に対して、$p(x) < \frac{1}{2}$ となっている $p(x)$ に従う定常無記憶情報源によって生成された文字列であるとする。このとき、確率分布 $p(x)$ の微少変化に対して文字列 S の基本アルゴリズムの *DNA* 配列の特定率は減少する。その微少変化とは以下の式で表されるものである。

$$\begin{aligned} p'(a) &= p(a) + \varepsilon, \\ p'(b) &= p(b) - \varepsilon, \\ p'(x) &= p(x) \quad x \neq a, b, \\ p(a) &\geq p(b) >> \varepsilon > 0 \ (p(a) + p(b) \leq \frac{2}{3}). \end{aligned}$$

系 2 一定の長さ m の文字列 S が、$x \in \Sigma$ に対して、$p(x) < \frac{1}{2}$ となっている $p(x)$ に従う定常無記憶情報源によって生成された文字列であるとする。このとき、確率分布 $p(x)$ の微少変化に対して文字列 S の拡張アルゴリズムの *DNA* 配列の特定率は減少する。その微少変化とは以下の式で表されるものである。

$$\begin{aligned} p'(a) &= p(a) + \varepsilon, \\ p'(b) &= p(b) - \varepsilon, \\ p'(x) &= p(x) \quad x \neq a, b, \\ p(a) &\geq p(b) >> \varepsilon > 0 \ (p(a) + p(b) \leq \frac{2}{3}). \end{aligned}$$

これらの定理や系により、エントロピーと DNA 配列の特定率に完全な相関はいえないが、ある程度の関係がいえ、例えば以下のような性質が成り立つ。

系 3 以下の式が成り立つとき、エントロピーレートは配列決定アルゴリズムの特定率に相関している。

$$p(A) = p/2, p(T) = p/2, p(G) = \frac{1-p}{2}, p(C) = \frac{1-p}{2}\ (0 < p < \frac{1}{2})$$

つまり、G と C、A と T の出現確率が同じならば、GC 含量が 50%のときに最も DNA 配列の特定率が高くなり、文字の分布が偏るにつれて特定率は低くなっていく傾向があることが分かる。このように文字列中の文字の分布が偏ると配列の特定率が低くなっていくことが分かった。

4 まとめ

本論文では、DNA 配列決定の 1 手法であるシーケンシングバイハイブリダイゼーションに対して、その配列決定する能力と、DNA 配列における文字の分布との関係を調べた。完全にエントロピーの増減と SBH の成功確率が相関する訳ではないが、ある程度関係があることが分かった。また、今までの解析で対象にしていたすべての文字が $\frac{1}{4}$ の確率でででてくる場合が 1 番都合の良い場合で実際には偏りのある場合を考慮しなければいけない。本論文で得られたような分布の偏りに従い配列の特定率が下がっていくという性質は実際の DNA 配列上における計算機実験でも示している ([2])。

ここでは、定常無記憶情報源を考えたが、マルコフ情報源などから生成される文字列についても同じような性質があるのか調べることを予定している。また、これらの性質を調べることにより、実際の DNA 配列の特徴づけも考えていく。

謝辞 日本学術振興会特別研究員として科学研究費補助金の援助を受けた。

参考文献

[1] R. Arratia, D. Martin, G. Reinert, and M. S. Waterman, Poisson process approximatation for sequence repeats, and sequencing by hybridization, *Journal of Computational Biology*, 3, 3, pp.425–463, 1996.

[2] K. Doi, A computer aided design of large scale biological experiments based on subsequence properties, A doctoral thesis, University of Tokyo, 2001.

[3] M. Dyer, A. Frieze, and S. Suen, The probability of unique solutions of sequencing by hybridization *Journal of Computational Biology*,1,2, pp.105–110, 1994.

[4] P. A. Pevzner, *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, 2000.

[5] P. A. Pevzner, l-tuple DNA sequencing: computer analysis, *Journal of Biomolecular Structure & Dynamics*, 7, 1, pp.63–73, 1989.

[6] P. A. Pevzner and R. J. Lipshutz, Towards DNA sequencing chips, *Proc. of the 19th International Conference on Mathematical Foundations of Computational Science*, Lecture Notes in Computer Science, 841, pp.143–158, 1994.

[7] F. P. Preparata and E. Upfal, Sequencing-by-hybridization at the information-theory bound: an optimal algorithm, *Journal of Computational Biology*,7,3, pp.621–630, 2000.

[8] F. P. Preparata, A. M. Frieze, and E. Upfal, On the power of universal bases in sequencing by hybridization, *Journal of Computational Biology*,6,3, pp.361-368, 1999.

[9] M. S. Waterman, *Introduction to Computational Biology*, Chapman and Hall, 1995.