A Greedy Algorithm for Optimal Recombination

Shiquan Wu and Xun Gu*

Center of Bioinformatics and Biological Statistics
Iowa State University, Ames, IA 50011, USA
{sqwu,xgu}@iastate.edu

Abstract. Let Σ be an alphabet and Σ^n denote the collection of all sequences of length n over Σ. For any $\mathbf{s}_1 = a_1a_2\cdots a_ja_{j+1}\cdots a_n$, $\mathbf{s}_2 = b_1b_2\cdots b_j\ b_{j+1}\cdots b_n \in \Sigma^n$, a recombination of $\mathbf{s}_1$ and $\mathbf{s}_2$ at position j is defined as an operation that crosses $\mathbf{s}_1$ and $\mathbf{s}_2$ at position j and generates $\mathbf{t}_1 = a_1a_2\cdots a_jb_{j+1}\cdots b_n$ and $\mathbf{t}_2 = b_1b_2\cdots b_ja_{j+1}\cdots a_n$. Denote $\mathcal{A}$ and $\mathcal{S}$ two collections of sequences. In this paper, we discuss generating $\mathcal{A}$ from $\mathcal{S}$ by a series of recombinations in minimum number of steps. We present a greedy algorithm for finding the optimal recombination evolutionary history from $\mathcal{S}$ to any tree $\mathcal{A}$ of sequences when $|\mathcal{S}| = 2$.

1 Introduction

Various types of mutations on sequences play an important role in computational biology. Transformations using insertion, deletion, substitution, and reversal are widely studied by applying statistics and algorithms (cf. [1,2]). Recently, much attention has been paid to recombination of sequences. Hein developed some algorithms for recombination problems(cf. [3,4]). Kececioglu and Gusfield discussed a recombination distance problem on generating a third sequence from a given pair of sequences in optimal recombination cost (cf. [5]).

Generally, an edit distance problem of genomes is widely studied and its aim is to find the optimal evolutionary history from some ancestors to some descendents by using certain types of mutations such as insertion, deletion, substitution(point mutation),reversal, etc. Parsimony trees and phylogenetic trees are some intersting problems in the category. In [6], an alignment with recombination is discussed and it is an edit distance problem involved recombinations.

In this paper, we discuss a similar distance problem involved recombinations. The purpose is to generate a collection $\mathcal{A}$ of sequences from another collection $\mathcal{S}$ of sequences by a series of recombinations in minimum number of steps.

1.1 Problem and Example

Definition 1. *Let Σ be an alphabet (e.g., $\Sigma = \{a, c, g, t\}$, etc) and Σ^n the collection of all sequences of length n over Σ. For any $\mathbf{s}_1 = a_1a_2\cdots a_ja_{j+1}\cdots a_n$,*

* Research supported in part by NIH Grant RO1 GM62118 (to X.G.) and NSF of China (19771025). Wu is on leave from Math Dept,N.U.D.T.,Hunan 410073,China.

J. Wang (Ed.): COCOON 2001, LNCS 2108, pp. 86–90, 2001.

$\mathbf{s}_2 = b_1 b_2 \cdots b_j b_{j+1} \cdots b_n \in \Sigma^n$, *a recombination for* $\mathbf{s}_1$ *and* $\mathbf{s}_2$ *at position* j $(1 \le j \le n-1)$ *is defined as the following operation:*

$$\begin{array}{rll} \text{From} & \mathbf{s}_1 = & \underline{a_1 a_2 \cdots a_j}\ \underline{a_{j+1} \cdots a_n} \\ & \mathbf{s}_2 = & \underline{b_1 b_2 \cdots b_j}\ \underline{b_{j+1} \cdots b_n} \\ \text{to} & \mathbf{t}_1 = & \underline{a_1 a_2 \cdots a_j}\ b_{j+1} \cdots b_n \\ & \mathbf{t}_2 = & \underline{b_1 b_2 \cdots b_j}\ \underline{a_{j+1} \cdots a_n}. \end{array}$$

It is denoted by $R(\mathbf{s}_1, \mathbf{s}_2; j; \mathbf{t}_1, \mathbf{t}_2)$, *or* $R(\mathbf{s}_1, \mathbf{s}_2; j)$. *Both* $\mathbf{s}_1$ *and* $\mathbf{s}_2$ *are called parents and both* $\mathbf{t}_1$ *and* $\mathbf{t}_2$ *are called children.*

Definition 2. *Let* $\mathcal{A}, \mathcal{S} \subseteq \Sigma^n$. $\mathcal{S}$ *is called a recombination ancestor of* $\mathcal{A}$ *if* $\mathcal{A}$ *can be generated from* $\mathcal{S}$ *by a series of recombinations, i.e., there exist a series of recombinations* $R_i(\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}; j^{(i)}; \mathbf{t}_1^{(i)}, \mathbf{t}_2^{(i)})$ $(i = 1, 2, \cdots, p)$ *such that (1)*$s_1^{(i)}, \mathbf{s}_2^{(i)} \in \mathcal{S} \cup \{\mathbf{t}_1^{(k)}, \mathbf{t}_2^{(k)} |\ 1 \le k \le i-1\}$, *(2)*$\mathcal{A} \subseteq \mathcal{S} \cup \{\mathbf{t}_1^{(i)}, \mathbf{t}_2^{(i)} |\ 1 \le i \le p\}$. $R_i(\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}; j^{(i)}; \mathbf{t}_1^{(i)}, \mathbf{t}_2^{(i)})$ $(i = 1, 2, \cdots, p)$ *is called a recombination process generating* $\mathcal{A}$ *from* $\mathcal{S}$. *It is called optimal if the number* p *of recombination steps is minimized among all such possible recombination processes.*

Recombination Problem 1. Let $\mathcal{A} \subseteq \Sigma^n$ and $\mathcal{S}$ a recombination ancestor of $\mathcal{A}$. Find the optimal recombination process generating $\mathcal{A}$ from $\mathcal{S}$. Denote $n(\mathcal{S}, \mathcal{A})$ the number of optimal recombination steps generating $\mathcal{A}$ from $\mathcal{S}$.

Example 1. Let $\mathcal{A} = \{\mathbf{a}_1 = 01010, \mathbf{a}_2 = 10001, \mathbf{a}_3 = 10110, \mathbf{a}_4 = 11001\}$ and $\mathcal{S} = \{\mathbf{s}_1 = 00101, \mathbf{s}_2 = 11010\}$. We have $n(\mathcal{S}, \mathcal{A}) = 4$ and the following is an optimal recombination process generating $\mathcal{A}$ from $\mathcal{S}$.

$$\begin{array}{rrrl} R_1(\mathbf{s}_1, \mathbf{s}_2;\ 1; \mathbf{a}_1, \mathbf{b}_4)\text{: From} & 0 & 0101 & = b_1 = \mathbf{s}_1 \\ & 1 & 1010 & = b_2 = \mathbf{s}_2 \\ \text{To} & 0 & 1010 & = b_3 = \mathbf{a}_1 \\ & 1 & 0101 & = b_4 \\ R_2(\mathbf{b}_4, \mathbf{s}_2; 2; \mathbf{b}_5, \mathbf{b}_6)\text{: From} & 10 & 101 & = b_4 \\ & 11 & 010 & = b_2 \\ \text{To} & 10 & 010 & = b_5 \\ & 11 & 101 & = b_6 \\ R_3(\mathbf{b}_4, \mathbf{s}_2; 3; \mathbf{a}_3,\ \mathbf{a}_4)\text{: From} & 101 & 01 & = b_4 \\ & 110 & 10 & = b_2 \\ \text{To} & 101 & 10 & = b_7 = \mathbf{a}_3 \\ & 110 & 01 & = b_8 = \mathbf{a}_4 \\ R_4(\mathbf{s}_1, \mathbf{b}_5; 3;\ \mathbf{a}_2, \mathbf{b}_9)\text{: From} & 001 & 01 & = b_1 \\ & 100 & 10 & = b_5 \\ \text{To} & 001 & 10 & = b_9 \\ & 100 & 01 & = b_{10} = \mathbf{a}_2 \end{array}$$

If $\mathcal{S}$ consists of only two sequences, then Problem 1 can be regarded as one with $|\Sigma| = 2$. For binary case, $\mathcal{S} = \{00\cdots0, 11\cdots1\}$ always generates an arbitrary binary sequence by recombinations. So it is a recombination ancestor for any

$\mathcal{A} \subseteq \{0,1\}^n$. Moreover, given any recombination ancestor $\mathcal{S} = \{s_{i1}s_{i2}\cdots s_{in} | i = 1,2\}$ of $\mathcal{A} = \{a_{i1}a_{i2}\cdots a_{in} | i = 1,2,\cdots,k\}$, for each $1 \leq j \leq n$, we define a function f_j on $\{s_{1j}, s_{2j}\}(s_{1j} \neq s_{2j})$ such that $f_j(s_{1j}) = 0$ and $f_j(s_{2j}) = 1$. Then we get $\mathcal{S}_0 = \{f_1(s_{11})f_2(s_{12})\cdots f_n(s_{1n}), f_1(s_{21})f_2(s_{22})\cdots f_n(s_{2n})\} = \{00\cdots 0, 11\cdots 1\}$ and $\mathcal{A}_0 = \{f_1(a_{i1})f_2(a_{i2})\cdots f_n(a_{in}) | i = 1,2,\cdots,k\}$. And $n(\mathcal{S},\mathcal{A}) = n(\mathcal{S}_0,\mathcal{A}_0)$. For $|\Sigma| = 2$, Recombination Problem 1 is reduced to the following simple form.

Recombination Problem 2. Let $\mathcal{A} \subseteq \{0,1\}^n$ be an arbitrary collection of binary sequences and $\mathcal{S} = \{00\cdots 0, 11\cdots 1\}$. Find the optimal recombination process generating $\mathcal{A}$ from $\mathcal{S}$.

In this paper, we discuss Problem 2 and find algorithms for it.

1.2 Terminology and Notation

Definition 3. *Let* $\mathbf{a} = a_1a_2\cdots a_p\cdots a_q\cdots a_n \in \{0,1\}^n$. $I = \mathbf{a}[p,q] = a_pa_{p+1}\cdots a_q$ *is called an alternative segment of* $\mathbf{a}$ *if* $a_i \neq a_{i+1}$ *for all* $p \leq i \leq q-1$. *An alternative block is a maximal alternative segment. Denote* $I_{\mathbf{a}} = \{I_1, I_2, \cdots, I_k\}$ *the collection of all alternative blocks of* $\mathbf{a}$. *Define the core of* $\mathbf{a}$ *as* $C_{\mathbf{a}} = a[s,t]$, *which is the minimum segment of* $\mathbf{a}$ *containing all* $I_i (1 \leq i \leq k)$. *Define the length of* $\mathbf{a}[p,q]$ *as* $l(\mathbf{a}[p,q]) = q-p$ *and the length of* $I_{\mathbf{a}}$ *as* $l(I_{\mathbf{a}}) = \sum_{j=1}^{k} l(I_j)$. *Denote* $\mathcal{P}(\mathcal{A}) = \{p | 1 \leq p \leq n-1, a_p \neq a_{p+1}$ *for some* $\mathbf{a} \in \mathcal{A}\}$. *For example, if* $\mathbf{a} = 11\underline{10101}1\underline{1010}000\underline{010}0$, *then* $I_1 = \mathbf{a}[3,7] = \underline{10101}$, $I_2 = \underline{1010}$, $I_3 = \underline{010}$. $I_{\mathbf{a}} = \{I_1, I_2, I_3\}$. $C_{\mathbf{a}} = \mathbf{a}[3,17] = \underline{10101}1\underline{101}000\underline{010}$. $l(I_1) = 4$ *and* $l(I_{\mathbf{a}}) = 9$.

Definition 4. *Let* $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$ *with* $C_{\mathbf{a}} = \mathbf{a}[p,q]$ *and* $C_{\mathbf{b}} = \mathbf{b}[u,v]$. *If* $u \leq p \leq q \leq v$ *and* $\mathbf{a}[p,q] = \mathbf{b}[p,q]$, *we say that* $\mathbf{b}$ *covers* $\mathbf{a}$, *denoted by* $\mathbf{a} \subseteq \mathbf{b}$. $\mathbf{b}$ *and* $\mathbf{a}$ *are called disjoint or independent if* $q \leq u$, *or* $v \leq p$, *denoted by* $\mathbf{a} \cap \mathbf{b} = \emptyset$. *If* $q \leq u$, *we say* $\mathbf{a} < \mathbf{b}$. *For any alternative segments (or blocks)* I *and* J, *we similarly define* $I \subseteq J$, $I \cap J = \emptyset$, *and* $I < J$.

Definition 5. *Denote* $\mathcal{C}(\mathcal{A}) = \{C_{\mathbf{a}} | \mathbf{a} \in \mathcal{A}\}$. $\mathcal{A}$ *is called a nest of sequences if* $\mathbf{a} \subseteq \mathbf{b}$, *or* $\mathbf{b} \subseteq \mathbf{a}$ *for any* $\mathbf{a}, \mathbf{b} \in \mathcal{A}$. *It is called a tree of sequences if either* $\mathbf{a} \cap \mathbf{b} = \emptyset$, *or* $\mathbf{a} \subseteq \mathbf{b}$, *or* $\mathbf{b} \subseteq \mathbf{a}$ *for any* $\mathbf{a}, \mathbf{b} \in \mathcal{A}$. *For* $\mathbf{a}, \mathbf{b} \in \mathcal{A}$, *if* $\mathbf{b} \subseteq \mathbf{a}$ *and there exists no* $\mathbf{c} \in \mathcal{A}$ *such that* $\mathbf{b} \subseteq \mathbf{c} \subseteq \mathbf{a}$ *with* $\mathbf{c} \neq \mathbf{a}$ *and* $\mathbf{c} \neq \mathbf{b}$, *then* $\mathbf{b}$ *is called a branch of* $\mathbf{a}$. *Denote* $B_{\mathbf{a}}$ *the collection of all branches of* $\mathbf{a}$ *in* $\mathcal{A}$. *A branch is called a leaf (or youngest branch) if it has no branch in* $\mathcal{A}$. $\mathbf{a} \in \mathcal{A}$ *is called a root (or oldest branch) if it is not a branch of any other sequences in* $\mathcal{A}$.

1.3 Related Work

Our recombination problem is a generalization of the following problem(cf.[5]).

Recombination Distance. Given $\mathbf{a} = a_1a_2\cdots a_m$, $\mathbf{b} = b_1b_2\cdots b_n$, and $\mathbf{c} = c_1c_2\cdots c_k$, find the minimum cost recombination to produce $\mathbf{c}$ from $\mathbf{a}$ and $\mathbf{b}$.

The goal is to generate a third sequence from a given pair by recombinations consisting of multiple crossovers and point mutations. In our problems, we define a recombination consisting of a single crossover with no point mutations. We discuss a more general problem to construct the optimal recombination spanning history from one family of sequences to another one. Its general case with multiple crossovers and point mutations is NP-complete(cf.[5,6]).

2 Theorems and Algorithms

In this section, we show some theorems on optimal recombination processes and design a greedy algorithm for finding the optimal recombination process for a tree of binary sequences. We always assume $\mathcal{S} = \{00\cdots0,\ 11\cdots1\}$.

Theorem 1. *Let* $\mathbf{a} \in \{0,1\}^n$. *Then* $n(\mathcal{S}, \{\mathbf{a}\}) = l(I_{\mathbf{a}})$.

Proof. $\mathbf{a}$ is optimally generated by $R(\mathbf{s}_1, \mathbf{s}_2; p; \mathbf{b}_1, \mathbf{c}_1)$, $R(\mathbf{b}_1, \mathbf{c}_1; p+1; \mathbf{b}_2, \mathbf{c}_2)$, $R(\mathbf{b}_2, \mathbf{c}_2; p+2; \mathbf{b}_3, \mathbf{c}_3)$, $\cdots\ \cdots$, $R(\mathbf{b}_{q-p-1}, \mathbf{c}_{q-p-1}; q-1; \mathbf{b}_{q-p}, \mathbf{c}_{q-p}), \cdots$ over all positions in $\mathcal{P}(\{\mathbf{a}\})$. Therefore, $n(\mathcal{S}, \{\mathbf{a}\}) = l(I_{\mathbf{a}})$.

The series of recombinations in the proof is called a recombination extension and denoted by $ext(\mathbf{s_1}, \mathbf{a})$. Generally, if $\mathbf{a} \subseteq \mathbf{b}$ and $\mathbf{a}$ has been generated, then $\mathbf{b}$ can be generated by series of recombinations on the positions in $\mathcal{P}(\{\mathbf{b}\})$ but not $\mathcal{P}(\{\mathbf{a}\})$, called an recombination extension from $\mathbf{a}$ to $\mathbf{b}$ and denoted by $ext(\mathbf{a}, \mathbf{b})$.

Theorem 2. *If* $\mathcal{A}$ *is a nest, then* $n(\mathcal{S}, \mathcal{A}) = |\mathcal{P}(\mathcal{A})|$.

Proof. By Theorem 1 and recombination extensions.

Theorem 3. *Let* $\mathcal{A}_1, \mathcal{A}_2 \subseteq \{0,1\}^n$ *be independent nests, i.e.,* $\mathbf{a} \cap \mathbf{b} = \emptyset$ *for any* $\mathbf{a} \in \mathcal{A}_1, \mathbf{b} \in \mathcal{A}_2$. *Then* $n(\mathcal{S}, \mathcal{A}_1 \cup \mathcal{A}_2) = |\mathcal{P}(\mathcal{A}_1)| + |\mathcal{P}(\mathcal{A}_2)|$.

Proof. Apply Theorem 2 to $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.

Theorem 4. *Let* $\mathcal{A}_1, \mathcal{A}_2 \subseteq \{0,1\}^n$ *be independent nests and* $\mathbf{a} \in \{0,1\}^n$. *For any* $\mathbf{b}_1 \in \mathcal{A}_1$ *and* $\mathbf{b}_2 \in \mathcal{A}_2$, *(1)*$\mathbf{b}_1 < \mathbf{b}_2$ *and (2)*$\mathbf{b}_1, \mathbf{b}_2 \subseteq \mathbf{a}$. *Then* $n(\mathcal{S}, \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{\mathbf{a}\}) = |\mathcal{P}(\{\mathbf{a}\})| + 1$.

Proof. Choose a position p between $\mathcal{A}_1$ and $\mathcal{A}_2$, then $C_{\mathbf{a}}$ is partitioned into C_1 and C_2. Choose $\mathbf{a}_1$ and $\mathbf{a}_2$ with $C_{\mathbf{a}_1} = C_1$ and $C_{\mathbf{a}_2} = C_2$. Then the theorem follows by applying Theorem 2 to both $\mathcal{A}_1 + \{\mathbf{a}_1\}$ and $\mathcal{A}_2 + \{\mathbf{a}_2\}$ with one more recombination $R(\mathbf{a}_1, \mathbf{a}_2; p)$, i.e., $|\mathcal{P}(\{\mathbf{a}_1\})| + |\mathcal{P}(\{\mathbf{a}_2\})| + 1 = |\mathcal{P}(\{\mathbf{a}\})| + 1$ steps.

Theorem 5. *Let* $\mathcal{A}$ *be a tree. Then* $n(\mathcal{S}, \mathcal{A}) = |\mathcal{P}(\mathcal{A})| + \sum_{\mathbf{a} \in \mathcal{A}} (|B_{\mathbf{a}}| - 1)$.

Proof. By Theorem 4 and induction on $|\mathcal{A}|$. Choose a root $\mathbf{a} \in \mathcal{A}$ and denote $B_{\mathbf{a}} = \{\mathbf{d}_1 < \mathbf{d}_2 < \cdots < \mathbf{d}_p\}$. We define $ext(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_p; \mathbf{a})$ as the following recombination process: (1)Choose a position p_i between $\mathbf{d}_i$ and $\mathbf{d}_{i+1}$ $(1 \le i \le p-1)$, (2)Partition $C_{\mathbf{a}}$ by all p_i into C_i and choose $\mathbf{b}_i$ with $C_{\mathbf{b}_i} = C_i (1 \le i \le p)$, (3)Make $ext(\mathbf{d}_i, \mathbf{b}_i)$ to get $\mathbf{b}_i$ from $\mathbf{d}_i (1 \le i \le p)$, (4)Make the recombinations: $R(\mathbf{b}_1, \mathbf{b}_2; p_1; \mathbf{f}_1, \mathbf{g}_1)$ and $R(\mathbf{f}_{i-1}, \mathbf{b}_i; p_i; \mathbf{f}_i, \mathbf{g}_i)$ $(2 \le i \le p-1)$ with $\mathbf{b}_j \subseteq \mathbf{f}_i$ $(1 \le j \le i-1)$. Then $\mathbf{f}_{p-1} = \mathbf{a}$. $ext(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_p; \mathbf{a})$ generates $\mathbf{a}$ from $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_p$ in optimal steps. $\mathcal{A}$ is then generated from branches to roots by induction.

Based on Theorem 5, we now design a greedy algorithm for finding the optimal recombination spanning evolutionary history generating a tree $\mathcal{A}$ from $\mathcal{S}$.

Greedy Algorithm
Input: $\mathcal{A}$ (nest/tree).
Output: Optimal recombination process.
Step 1 Find $B_{\mathbf{a}}$ for all $\mathbf{a} \in \mathcal{A}$ and set $E = \emptyset$.
Step 2 While $\mathcal{A} - E \neq \emptyset$ {
Choose a leaf (youngest branch) $\mathbf{a} \in \mathcal{A} - E$ and
denote $B_{\mathbf{a}} = \{\mathbf{d}_1 < \mathbf{d}_2 < \cdots < \mathbf{d}_p\}$.
Generate $\mathbf{a}$ by $ext(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_p; \mathbf{a})$.
$E = E + \{\mathbf{a}\}$.
}

Theorem 6. *Greedy Algorithm finds the optimal recombination process for a tree $\mathcal{A}$ and the run time is $O(|\mathcal{A}|^2 n)$.*

Proof. For each pair $\mathbf{a}$ and $\mathbf{b}$ of sequences, the algorithm compares n positions to determine whether $\mathbf{a} \subseteq \mathbf{b}$ or not. There are $O(|\mathcal{A}|^2)$ pairs of sequences. Therefore the run time is $O(|\mathcal{A}|^2 n)$.

3 Conclusion

The greedy algorithm is designed to generate a tree from two sequences. The most interesting part of the problem is to generate an arbitrary collection of sequences from a given recombination ancestor. Our discussion may be applied to some problems in human SNP(single nucleotide polymorphism) genome project.

References

1. Durbin, R., Eddy, S. R., Krogh, A. and Mitchison, G., 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids,*. Cambridge University Press.
2. Gusfield, D., 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press.
3. Hein, J., 1990. *Reconstruct evolution of sequences subject to recombination, Mathematical biosciences* 98:185-200.
4. Hein, J., 1993. *A heuristic method to reconstruct the history of sequences subject to recombination, Journal of Molecular Evolution* 36: 396-450.
5. Kececioglu, J. and Gusfield, D., 1994. *Reconstructing a history of recombinations from a set of sequences, 5th Annual ACM-SIAM Symposium on Discrete Algorithm* Arlington, Virginia, pp.471-480. Also in Discrete Applied Mathematics 88(1998)239-260.
6. Wang,L., B. Ma and M. Li, 2000, *Fixed topology alignment with recombination, Discrete Applied Mathematics* 104: 281-300