개선된 다이나믹 프로그래밍과 품질 정보 및 퍼지 추론 기법을 이용한 DNA 염기 서열 배치 알고리즘

Seung-hwan Lee*, Choong-shik Park, and Kwang-baek Kim***

** Division of Computer and Information Engineering, Silla University*
E-mail: sagara@korea.com, gbkim@silla.ac.kr

*** Department of Computer Engineering, Youngdong University*
E-mail: leciel@youngdong.ac.kr

Abstract

DNA 염기 서열 배치 알고리즘은 분자 생물학 분야에서 단백질과 핵산 서열들의 분석에서 중요한 방법이다. 생물학적인 염기 서열들은 그들 사이의 유사성과 차이점을 나타내기 위해 정렬된다. 본 논문에서는 기존의 DNA 염기 서열 배치 방법을 개선하기 위하여 DP(Dynamic Programming) 알고리즘의 비용증가($O(nm)$) 문제를 해결하는 Quadrant 방법과 품질 정보 및 퍼지 추론 시스템(fuzzy inference system)을 적용한 DNA 염기 서열 배치 알고리즘을 제안한다. 본 논문에서 제안한 DNA 염기 서열 배치 알고리즘은 Quadrant 방법을 적용하여 Needleman-Wunsch의 DP 기반 알고리즘에서의 행렬 생성 단계에서 발생하는 불필요한 정렬 계산을 제거하여 전체 수행 시간을 단축하고, 각 DNA 염기 서열 단편 각각의 길이 차이와 낮은 품질의 DNA 염기 빈도를 퍼지 추론 시스템에 적용하여 지능적으로 갭 비용(gap cost)을 동적으로 조정한다. 제안된 알고리즘의 성능 평가를 위해 NCBI (National Center for Biotechnology Information)의 실제 유전체 데이터로 성능을 분석한 결과, 제안된 알고리즘이 기존의 품질정보만을 이용한 알고리즘보다 개선된 것을 확인하였다.

Keywords:

품질 정보(quality information), 갭 비용(gap cost), 동적 프로그래밍(dynamic programming), 퍼지 추론 시스템(fuzzy inference system), Quadrant

1. 서론

DNA 염기 서열 배치는 두 개 혹은 그 이상의 DNA 서열을 비교 및 정렬하여 상동성 (homology)이 높은 서열들을 알아내 서열의 기능을 유추하거나, 각 서열간의 진화적 연관성이나 관련 기능 등을 예측하기 위한 것이 주 목적이다. 다양한 방법으로 많은 알고리즘이 개선되어 왔으며 이는 분자 생물학 분야에서 매우 중요한 작업이며 생체 정보 처리의 기본이라고 할 수 있다[1-4]. DNA 염기 서열 배치 알고리즘은 최적화 조건의 결과를 구하기 위한 최적화검색(optimized search) 알고리즘과 한 배열과 데이터베이스 전체간의 비교와 같이 빠른 검색이 요구될 때 최적화에 근사한 결과를 얻기 위해 사용하는 휴리스틱 탐색 (heuristic search) 알고리즘으로 구분된다.

생물의 DNA 염기 서열을 밝혀내는 프로그램인 PHRED[5]에서 생성되는 품질 정보를 이용한 기존의 알고리즘[6]은 정확성을 보장하는 Needleman-Wunsch의 동적 프로그래밍 (dynamic programming) 기반 알고리즘[7]으로써 최적화검색에 속한다고 할 수 있다. 기존의 알고리즘은 DNA 서열의 단편 생성시 말단 부분에 발생하는 낮은 품질의 DNA 염기가 존재하는 경우에 정렬에 있어서 오차가 발생한다. 또한 큰 데이터베이스에 적용될 때 DP 알고리즘의 비용증가 문제가 발생하게 된다.

본 논문에서는 기존의 전역배치에서 동적 프로그래밍 알고리즘의 고전적인 문제인 Scoring matrix 생성 단계의 비용증가를 해결하기 위해 불필요한 연산을 줄이는 Quadrant방법[9]과 DNA 염기 서열의 낮은 품질의 빈도수와 쌍 정렬에서 두 서열간의 길이 차이 정보를 퍼지 추론 시스템에 적용하여 갭 비용(gap cost)을 동적으로 조정하는 알고리즘을 제안한다.

2. 관련 연구

DNA는 4종류의 염기 서열로 구성된 이중나선형 구조로써, 이는 A, G, C, T의 문자 집합인 스트링으로 간주할 수 있다. 2절에서는 두개의 유전자 서열을 비교하는 염기 서열 배치와 생물학 데이터의 품질정보를 기술한다.

2.1 전역 염기 서열 배치

염기 서열 배치는 두 개의 염기 서열 S_1과 S_2가 있을 때, 두 개의 서열의 중간이나 끝에 적절하게 갭(gap)이 들어갈 위치를 선택, 삽입하여 두 서열이 같은 길이가 되도록 정렬하는 것이다. 서열 배치 문제는 모든 열에서 불일치를 최소화하는 최적 배치(optimal alignment) 방법을 찾는 문제이다. 이 문제는 전산학에서 오랫동안 연구되어온 스트링 처리분야의 편집 거리(edit distance)를 찾는 알고리즘으로 해결될 수 있는데, 편집 거리란 두 서열 S_1과 S_2에 대해 S_1을 S_2로 바꾸는데 필요한 최소의 편집 연산의 총합으로 정의되며, 편집 연산은 아래의 연산 중 하나에 해당된다.

(1) 삽입(insertion) : S_2에 하나의 문자를 삽입

(2) 삭제(deletion) : S_1에 하나의 문자를 삭제

(3) 변경(change) : S_1의 하나의 문자를 S_2의 다른 문자로 변경

	1	2	3	4	5
S_1	A	G	C	–	T
S_2	–	G	A	C	T

그림 1 – 서열 배치 예제

그림 1에서 '–' 는 갭(gap)이며 2, 5열의 경우와 같이 동일한 문자가 한 열에 위치하는 경우를 일치(match)라고 하고 1, 3, 4와 같이 서로 다른 문자가 한 열에 위치하면 불일치(mismatch)라고 한다. 한 열에 두 염기가 갭인 경우는 허용하지 않는다. 1열에서 A가 삭제(deletion)되고 3열에서 변경(change)되며 4열에서 C가 삽입(insertion) 되므로 두 서열의 편집 거리는 3이다. 염기 서열의 배치 문제는 편집 거리를 구하는 문제와 동일하므로 DP 기법을 적용하여 해결할 수 있다. DP 알고리즘은 2n의 수행시간에 작동하는 $O(2n)$ 알고리즘으로서 프린스턴 대학교의 Richard Bellman에 의해 최적화 문제를 해결하기 위해 소개되었으며, 가장 기본적인 문제의 답부터 계산한 후, 순차적으로 더 큰 문제를 해결하여 전체 문제를 해결하는 알고리즘이다. 이 기법은 크기가 $(S_1+1)\times(S_2+1)$ 인 테이블을 이용하여 각 서열의 길이를 S_1= n, S_2= m 이라고 할 때 $O(nm)$ 시간에 최적 배치를 구한다. 최적 배치는 테이블의 정보를 역으로 추적하여 구할 수 있다[7].

2.2 품질 정보

DNA 염기 결정 프로그램은 Trace 데이터를 읽어서 DNA 염기들의 서열과 각 DNA 염기에 해당하는 품질 정보(quality score)를 생성한다. 본 논문에서는 염기 결정 프로그램 중 대표격인 PHRED에서 생성된 품질 정보를 다룬다. PHRED에서는 Sequencing machine이 chromatogram의 정점(peak)을 분석하여 trace data를 생성한다. 이 data는 DNA 염기의 서열인 FASTA와 각 염기의 품질 정보인 Quality의 파일을 생성한다. 이상적인 경우에는 그림 2(a)와 같이 trace data에서 모든 정점이 거의 동일한 거리를 두고, 서로 겹치지 않는다. 그러나 실제 trace data는 실험적인 한계로 인하여 오류들이 존재하기 때문에 이상적인 것과는 상이하다. 그림 2(b)와 같이 좋지 않은 품질의 trace data의 특징은 다음과 같다.

(1) 두 정점들 사이의 거리가 일정하지 않고 다양하게 분포 되어 있다.

(2) 둘 이상의 곡선이 비슷한 정점을 가지고 있다.

(3) 네 곡선의 정점이 모두 매우 낮은 경우가 존재한다.

이러한 특징 때문에 해당 위치의 DNA 염기가 무엇인지 확신 할 수 없게 되어 낮은 품질의 DNA FASTA를 얻게 된다.

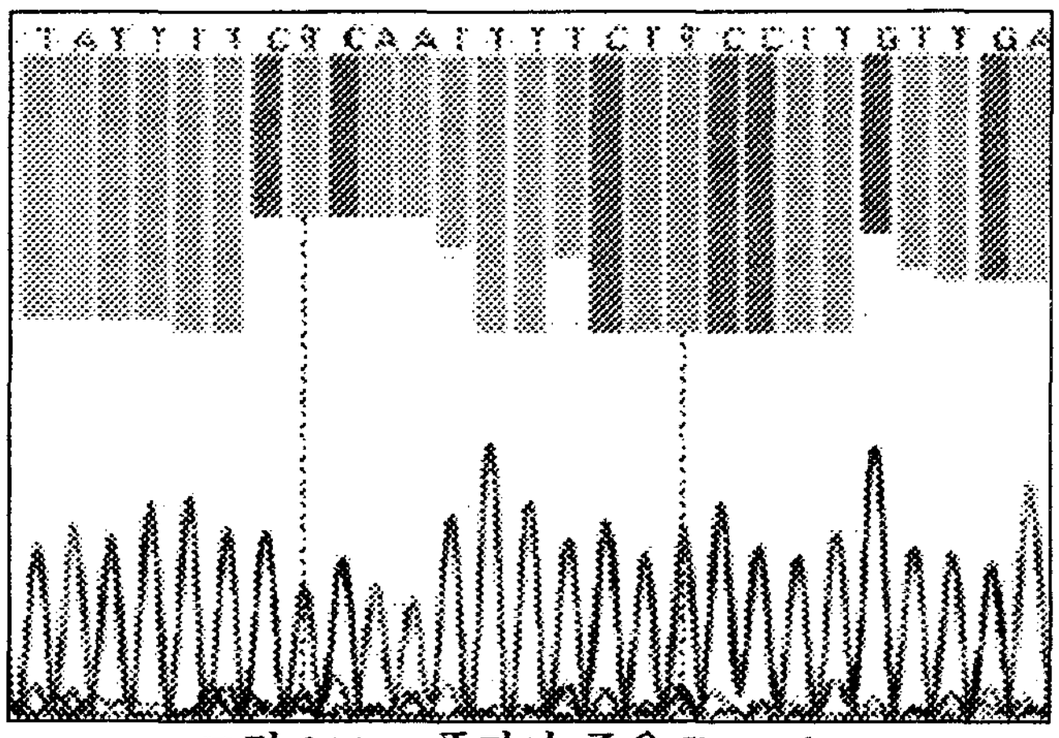

그림 2(a) – 품질이 좋은 Trace data

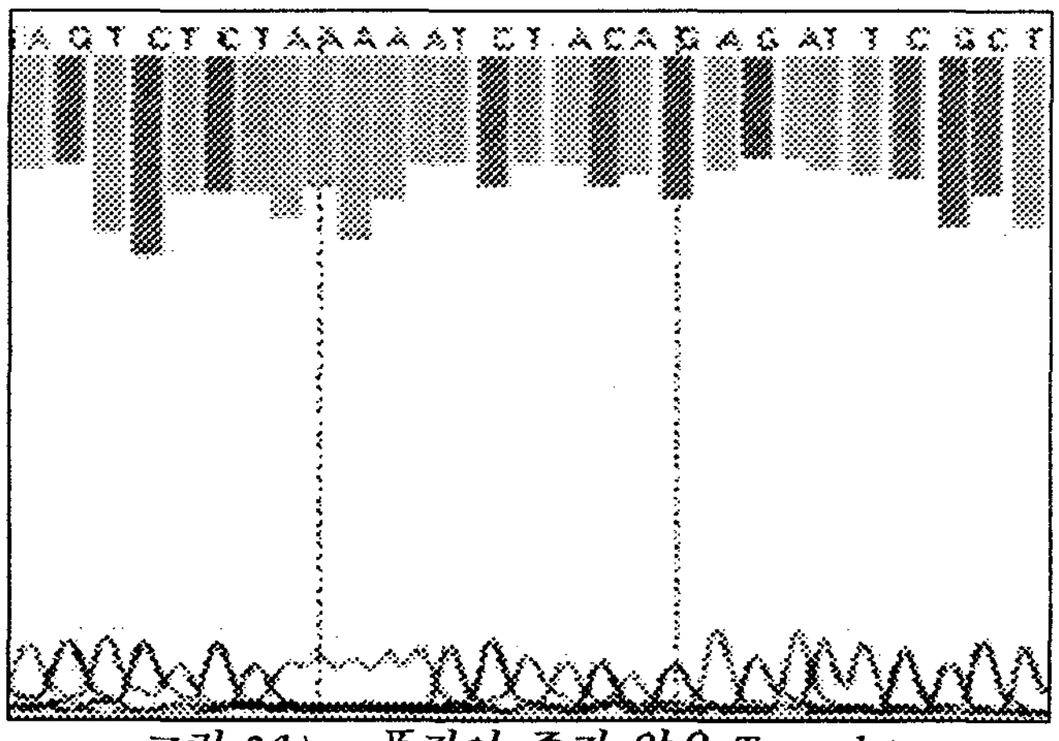

그림 2(b) – 품질이 좋지 않은 Trace data

A의 신호 G의 신호

C의 신호 T의 신호

그림 2 – trace data의 예

그림 2의 상단 문자들은 PHRED가 trace data에서 생성한 DNA 염기 서열 (FASTA)이고 막대 그래프가 품질 점수이다. 품질 점수는 0~99 사이의 값을 가지는데, 해당 위치의 DNA 염기의 실제와 차이를 나타내는 확률과 관계가 있다. NCBI Handbook에 따르면 PHRED에서 품질점수는 $10^{-P/10}$ 의 에러확률에 상응한다. 따라서 품질점수를 Q라 하고 DNA 염기의 에러 확률을 P 라 하면 $Q=-10\times\log_{10}P$ 인 관계가 성립한다. 예를 들어 해당 위치의 염기가 C이고 품질 점수가 20이라면, 그 위치의 염기가 C가 아닐 확률은 0.01이며, 품질 점수가 10이라면 에러 확률은 0.1이 된다. 표 1은 품질 점수에 따른 에러 확률을 나타낸다.

표 1- 품질 점수에 따른 에러 확률

품질 점수 (Q)	에러 확률 (P)
10	0.1
20	0.01
30	0.001
40	0.0001
50	0.00001
60	0.000001

PHRED는 염기 서열과 품질 점수 생성 후에 서열을 정돈(trimming)하는 과정을 수행한다. 보통 Trace data의 첫 부분과 끝 부분은 실험적 한계로 인해 많은 오류들을 포함하여 낮은 품질 점수를 갖는다. 많은 오류를 포함하는 단편의 말단 부분은 실험에 좋지 않은 영향을 끼치기 때문에 제거한다. 따라서 실제 사용되는 서열에서 10보다 작은 품질 점수를 가지는 DNA 염기의 비율은 2~5% 정도이다.

3. 제안된 DNA 염기 서열 배치 알고리즘

3.1 품질 정보를 적용한 DNA 염기 서열 배치 알고리즘

DNA 염기 서열은 집합 $\Sigma=\{A, C, G, T\}$ 에 속한 염기들의 나열이다. 갭(gap)은 $\Delta\notin\Sigma$ 로 정의한다. DNA 염기 서열의 A 의 i 번째 DNA 염기는 A_i, 부분 DNA 염기 서열 $A_iA_{i+1}\cdots A_j$ 는 $A[i..j]$ 로 정의한다. 앞으로 품질 정보를 가지는 DNA 염기 서열을 품질 DNA 염기 서열이라 정의하고, 이와 구별하여 품질 정보를 가지지 않는 서열을 일반 DNA 염기 서열로 정의한다. 품질 DNA 염기 서열에서 DNA 염기가 없어서 생기는 공백을 '-'로 표시하고, x 는 해당 위치에서의 대표 DNA 염기로 한다. 본 논문에서는 다음을 가정한다.

- 가정 1 : 어떤 위치에 대표 DNA 염기가 나타날 확률은 보통 0.9보다 크다. 따라서 품질 DNA 염기 서열의 품질 점수는 10보다 크고, 이는 대표 DNA 염기가 나타날 확률이 0.9보다 크다는 것을 의미한다.
- 가정 2 : 대표 DNA 염기 이외의 DNA 염기들과 공백(-)이 나타날 확률은 모두 같다. 시퀀스 해당 위치에 대표 DNA 염기가 A이고 품질 점수가 10이면 에러 확률은 0.1이다. 따라서 해당 위치에 G, C, T, - 이 나타날 확률은 각각 0.025이다. 해당 위치에 대표 DNA 염기가 나타날 확률이 0.9로 매우 크므로 다른 분자들이 나타날 확률은 매우 작다. 따라서 이 확률들은 모두 같다고 가정하더라도 실제 각 문자들이 나타날 확률과 크게 차이가 나지 않는다.

길이가 각각 m 과 n 인 두 DNA 염기 서열 $A=A_1A_2\cdots A_m$ 과 $B=B_1B_2\cdots B_n$ 이 주어졌을 때, 두 품질 DNA 염기 서열의 전역 배치는 $A^*=A_1{}^*A_2{}^*\cdots A_m{}^*$ 와 $B^*=B_1{}^*B_2{}^*\cdots B_n{}^*$ $(n,m\le l)$ 이다. $A_i{}^*$ 와 $B_i{}^*$ 는 각각 A 와 B 의 DNA 염기들 사이에 0개 또는 1개 이상의 공백(Δ)을 삽입함으로써 정렬된다. 공백(Δ)을 삽입해서 배치되는 것은 일반 DNA 염기 서열과 같다. 배치에 삽입되는 공백(Δ)은 그 부분에 항상 어떤 문자도 존재하지 않는다는 것을 의미하므로, 그 위치에 DNA 염기 $x\in\Sigma$ 가 나타날 확률은 0이고 공백(-)이 나타날 확률은 1이다. DNA 염기 쌍 $A_i{}^*$ 와 $B_i{}^*$ 는 그 대표 DNA 염기에 따라서 다음과 같이 세 종류의 매핑 중 하나로 분류된다.

- 정규일치(regular-match): $A_i{}^*\neq\Delta$, $B_i{}^*\neq\Delta$ 이고, $A_i{}^*=B_i{}^*$ 인 경우.
- 정규불일치(regular-mismatch): $A_i{}^*\neq\Delta$, $B_i{}^*\neq\Delta$ 이고, $A_i{}^*\neq B_i{}^*$ 인 경우.
- 정규갭(regular-gap): $A_i{}^*$ 또는 $B_i{}^*$ 가 Δ 인 경우

2.1절의 전역 배치와 마찬가지로 $A_i{}^*=B_i{}^*=\Delta$ 인 경우는 허용하지 않는다. 각 매핑은 해당되는 점수를 가지는데, 일치는 γ, 불일치는 δ, 갭은 μ 를 가진다. 이 γ,δ,μ 를 매핑 점수 인자라 부르는데, 응용(application)에 따라 다양한 값을 가진다. 보통 γ 는 양수이고, δ 와 μ 는 음수 이다. 위 세 매핑을 품질 매핑 이고, 일반 서열의 매핑인 일치, 불일치, 갭을 일반 매핑이라 칭한다.

DNA 염기 서열 A_i^* 와 B_i^* 의 매핑 점수 $S(A_i^*, B_i^*)$ 는 일반 매핑 점수의 기대값으로 정의된다. A_i^* 와 B_i^* 의 실제 DNA 염기 종류에 따라 품질 매핑은 일반 매핑인 일치, 불일치, 갭 중의 하나가 된다. 표 2는 A_i^* 와 B_i^* 가 실제 DNA 염기에 따라 일반 매핑으로 분석되는 결과를 나타낸다.

표 2– 실제 DNA 염기에 따른 일반 매핑

B_i^* \ A_i^*	a	c	t	g	-
a	M	N	N	N	G
C	N	M	N	N	G
T	N	N	M	N	G
G	N	N	N	M	G
-	G	G	G	G	E

- M: 실제 DNA 염기가 같은 경우이다. 따라서 이 경우는 일치 점수 γ를 가진다.
- N: 둘 다 공백이 아니면서 실제 DNA 염기가 서로 다른 경우이다. 불일치 점수 δ를 가진다.
- G: 한쪽은 Σ이고 다른 한쪽은 – 인 경우이다. 갭 매핑으로 갭 점수 μ를 부여한다.
- E: A_i^* 와 B_i^* 가 모두 공백인 경우이다. 이 경우는 이 매핑이 배치 상에서 없는 것으로 간주할 수 있으므로 배치 점수에 영향을 주지 않도록 0의 점수를 부여한다.

따라서 일치 매핑이 될 확률을 $P_m(A_i^*, B_i^*)$, 불일치 매핑이 될 확률을 $P_n(A_i^*, B_i^*)$, 갭 매핑이 될 확률을 $P_g(A_i^*, B_i^*)$ 라 하면 식 (1)과 같이 나타낼 수 있다.

$$S(A_i^*, B_i^*) = \gamma \times P_m(A_i^*, B_i^*) + \delta \times P_n(A_i^*, B_i^*) + \mu \times P_g(A_i^*, B_i^*) \quad (1)$$

A_i^* 의 실제 문자가 $x \in \Sigma$ 일 확률을 α_x, 공백으로 남을 확률을 α_-, B_i^* 의 실제 문자가 $x \in \Sigma$ 일 확률을 β_x, 공백으로 남을 확률을 β_- 로 정의하고 각 품질의 매핑 점수를 구체적으로 계산하면 다음과 같다.

3.1.1 정규일치(regular-match)의 경우

A_i^* 와 B_i^* 의 대표 DNA 염기를 a 라 하면 '가정 2 '로 부터 식 (2)를 유도할 수 있다.

$$\frac{1-\alpha_a}{4} = \alpha_c = \alpha_g = \alpha_t = \alpha_-, \qquad \frac{1-\beta_a}{4} = \beta_c = \beta_g = \beta_t = \beta_- \quad (2)$$

표 3은 정규 일치의 확률 표이다.

표 3– 정규 일치의 확률

B_i^* \ A_i^*	A	c	t	g	-
A	$\alpha_a\beta_a$	X	X	X	X
C	Y	Z	Z	Z	Z
T	Y	Z	Z	Z	Z
G	Y	Z	Z	Z	Z
-	Y	Z	Z	Z	Z

표 3에서 각 DNA 염기의 확률은 식 (3)과 같다.

$$a = \alpha_a, \qquad \{c, g, t\} = \frac{1-\alpha_a}{4} \quad (3)$$

표 3의 확률 표의 X, Y, Z는 식 (4)와 같다.

$$X = \frac{(1-\alpha_a)\beta_a}{4}, Y = \frac{\alpha_a(1-\beta_a)}{4}, Z = \frac{(1-\alpha_a)(1-\beta_a)}{16} \quad (4)$$

표 3으로부터 다음 식(5)를 유도한다. Z는 매우 작은 값이기 때문에 식(5)에서 무시한다.

$$P_m(A_i^*, B_i^*) = \alpha_a\beta_a + 3Z \approx \alpha_a\beta_a$$

$$P_n(A_i^*, B_i^*) = 3X + 3Y + 6Z \approx \frac{\alpha_a + \beta_a - 2\alpha_a\beta_a}{4} \times 3 \quad (5)$$

$$P_g(A_i^*, B_i^*) = X + Y + 6Z \approx \frac{\alpha_a + \beta_a - 2\alpha_a\beta_a}{4}$$

따라서 정규 일치의 매핑 점수는 식 (6)과 같이 나타낼 수 있다.

$$S(A_i^*, B_i^*) = \gamma \times \alpha_a\beta_a + \delta \times \frac{\alpha_a + \beta_a - 2\alpha_a\beta_a}{4} \times 3 + \mu \times \frac{\alpha_a + \beta_a - 2\alpha_a\beta_a}{4} \quad (6)$$

3.1.2 정규불일치(regular-mismatch)의 경우

A_i^* 의 대표 DNA 염기를 c, B_i^* 의 대표 DNA 염기를 a라 하면 '가정 2'로 부터 식 (7)를 유도할 수 있다.

$$\frac{1-\alpha_c}{4}=\alpha_a=\alpha_g=\alpha_t=\alpha_-,$$
$$\frac{1-\beta_a}{4}=\beta_c=\beta_g=\beta_t=\beta_- \quad (7)$$

표 4는 정규 불일치의 확률 표이다.

표 4- 정규 불일치의 확률

A_i^* / B_i^*	a	c	t	g	-
a	X	$\alpha_c\beta_a$	X	X	X
c	Z	Y	Z	Z	Z
t	Z	Y	Z	Z	Z
g	Z	Y	Z	Z	Z
-	Z	Y	Z	Z	Z

여기서 각 DNA 염기의 확률은 식 (8)과 같다.

$$c=\alpha_c,$$
$$\{a,g,t\}=\frac{1-\alpha_c}{4} \quad (8)$$

표 4의 확률 표에서 나타나는 X, Y, Z는 식 (9)와 같다.

$$X=\frac{(1-\alpha_c)\beta_a}{4},Y=\frac{\alpha_c(1-\beta_a)}{4},Z=\frac{(1-\alpha_c)(1-\beta_a)}{16} \quad (9)$$

표 4의 확률 표로부터 식 (10)을 유도할 수 있다. Z는 매우 작은 값이기 때문에 식 (10)에서 무시한다.

$$P_m(A_i^*,B_i^*)=X+Y+2Z\approx\frac{\alpha_c+\beta_a-2\alpha_c\beta_a}{4}$$

$$P_n(A_i^*,B_i^*)=\alpha_c\beta_a+2X+2Y+7Z\approx\frac{\alpha_c+\beta_a}{2} \quad (10)$$

$$P_g(A_i^*,B_i^*)=X+Y+6Z\approx\frac{\alpha_c+\beta_a-2\alpha_c\beta_a}{4}$$

따라서 정규 일치의 매핑 점수는 식 (11)과 같이 나타낼 수 있다.

$$S(A_i^*,B_i^*)=\gamma\times\frac{\alpha_c+\beta_a-2\alpha_c\beta_a}{4}+\delta\times\frac{\alpha_c+\beta_a}{2}+\mu\times\frac{\alpha_c+\beta_a-2\alpha_c\beta_a}{4} \quad (11)$$

3.1.3 정규 갭(regular-gap)의 경우

A_i^* 의 대표 DNA 염기를 a, B_i^* 의 대표 DNA 염기를 공백(Δ)이라 하면 B_i^* 가 공백(Δ) 이므로 $\beta_-=1$이다. 정규 갭의 매핑 점수는 식 (12)와 같다.

$$S(A_i^*,B_i^*)=\mu\times(1-\alpha_-)=\mu\times\frac{3+\alpha_a}{4} \quad (12)$$

일반 DNA 염기 서열과 같이 품질 DNA 서열의 배치 점수 $S(A_i^*,B_i^*)$는 배치를 이루는 모든 DNA 염기 쌍들의 매핑 점수의 합으로 정의된다. 즉, 식 (13)과 같이 정의된다.

$$S(A^*,B^*)=\sum_{i=1}^{l}S(A_i^*,B_i^*) \quad (13)$$

가정 1에 의하여 α_a 와 β_a 가 0.9 이상이므로, $\frac{\alpha_a+\beta_a-2\alpha_a\beta_a}{4}$ 는 모든 경우에 0.045 이하의 값으로 매우 작은 값이다. 따라서 이 항(term)을 각 정규 일치, 정규 불일치 식에서 생략하면 좀 더 단순화 된 다음의 정규 일치 매핑 점수 식 (14)와 정규 불일치 매핑 점수 식(15)를 유도 할 수 있다.

$$S(A_i^*,B_i^*)=\gamma\times\alpha_a\beta_a \quad (14)$$

$$S(A_i^*,B_i^*)=\delta\times(\alpha_c+\beta_a)/2 \quad (15)$$

제시된 알고리즘은 길이가 각각 m 과 n 인 두 염기 서열이 주어 졌을 때 서열의 배치 점수가 가장 높은 경우를 최적 배치 (optimized alignment)로 탐색한다. $H_{i,j}$ 를 $A[1..i]$ 와 $B[1..j]$ 의 최적 배치 점수라 할 때, $H_{i,j}$ 는 동적 프로그래밍 기법을 적용하여 계산할 수 있고, 이 기법은 기존의 Needleman-Wunsch 알고리즘과 같은 구조를 가진다. 그림 3은 품질 DNA 염기 서열을 배치하기 위한 점화식이다.

$\gamma>0$: 일치 점수
$\delta<0$: 불일치 점수
$\mu<0$: 갭 점수
Q_x : 문자 x의 품질점점
$P_x=1-10^{-Q_x/10}$

$$S(A_i,B_j)=\begin{cases}\gamma\times P_{A_i},P_{B_j} & if\ A_i\text{와}B_j\text{가 일치}\\ \delta\times(P_{A_i}+P_{B_j})/2 & if\ A_i\text{와}B_j\text{가 불일치}\end{cases}$$

$$H_{0,0}=0$$

$$H_{i,0}=H_{i-1,0}+\mu\times\frac{3+P_{A_i}}{4},(1\le i\le m)$$

$$H_{0,j}=H_{0,j-1}+\mu\times\frac{3+P_{B_i}}{4},(1\le j\le n)$$

$$H_{i,j}=\max\begin{cases}H_{i-1,j-1}+S(A_i,B_j)\\ H_{i-1,j}+\mu\times\frac{3+P_{A_i}}{4}\\ H_{i,j-1}+\mu\times\frac{3+P_{B_j}}{4}\end{cases}$$

$$(1\le i\le m,1\le j\le n)$$

그림 3– 품질 서열 배치를 위한 점화식

3.2 Quadrant 방법을 적용한 동적 프로그래밍(DP)의 비용 감소

동적 프로그래밍은 최적의 비용을 가지는 정렬을 찾아주지만, 연산시간이 많이 소요되는 단점이 있다. 이를 해결하기 위하여 휴리스틱 탐색을 적용하여 최적에 가까운 (near optimal) 배치를 찾아주는 알고리즘이 개발되었는데, BLAST와 FASTA가 그 예이다. 이 알고리즘들은 서열과 정렬에 대한 통계적 정보를 이용하여, 대규모의 DB에 대해 효율적, 실용적인 탐색이 가능하다. 따라서 본 논문에서는 DB에 적용 가능하고 정확도의 신뢰를 갖는 DP 알고리즘에 Quadrant 방법을 적용하여 효율적으로 최적 서열 배치를 수행한다. DP 알고리즘에서 $H_{i,j}$ 의 엔트리는 $H_{i-1,j-1}$, $H_{i,j-1}$, $H_{i-1,j}$ 의 이전 엔트리들로부터 구해지는데, 각각을 1/4분면, 2/4분면, 3/4분면, 4/4분면(자신)으로 두고, 전체를 사분면(quadrant)으로 정의한다. DP 알고리즘의 Scoring matrix의 생성 단계는 quadrant가 위에서 아래, 좌에서 우로 이동하면서 4사분면 값을 채워가는 과정이며, 역 추적(trace back)을 통해 최적 정렬 값을 구하는 알고리즘이다. 그림 4는 quadrant의 4분면을 나타낸다.

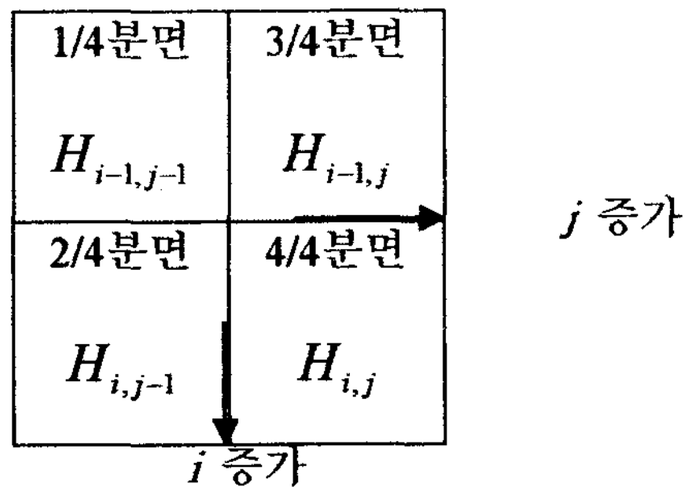

그림 4 - Quadrant의 4분면

Scoring matrix를 계산하기 위해 quadrant는 i방향과 j방향으로 진행하게 되는데, 각 진행방향에 대해 다음과 같이 정의할 수 있다.

정의 1. quadrant의 현재 위치를 $H_{i,j}$, 다음 위치(i+1)를 $H_{i+1,j}+t$ 이라 할 때, $H_{i,j-1}>H_{i-1,j}$ 이면, $H_{i,j-1}+t>H_{i-1,j}+t$ 이다.

정의 2. quadrant의 현재 위치를 $H_{i,j}$, 다음 위치(j+1)를 $H_{i,j+1}+t$ 이라 할 때, $H_{i,j-1}>H_{i-1,j}$ 이면, $H_{i,j-1}+t>H_{i-1,j}+t$ 이다.

정의 1, 정의 2를 통해 quadrant의 i가 증가하는 방향(수직 방향)으로 전개될 때, 처음 위치 $H_{i,j}$ 일 때 2/4분면의 값이 3/4분면의 값보다 작은 곳이 나타나면, 그 후의 $H_{i+1,j}+t$ 일 때도 2/4분면의 값은 3/4분면의 값보다 작다. 따라서, 엔트리 값을 구하는데 있어 불필요 하므로 더 이상 진행할 필요가 없다. 이는 j가 증가하는 방향(수평 방향)일 경우에도 동일하게 적용된다. 즉, $H_{i,j-1}<H_{i-1,j}$ 이면, 항상 $H_{i+1,j-1}<H_{i,j}$ 이고, $H_{i-1,j}<H_{i,j-1}$ 이면, 항상 $H_{i-1,j+1}<H_{i,j}$ 이므로 $H_{i+1,j-1}$ 와 $H_{i,j+1}$ 을 구할 필요가 없게 된다[9]. 그림 5는 정의1, 2에 의하여 quadrant가 진행하는 방향에 대한 전체 알고리즘이다.

```
for all j, m, do
        for all i, i=(k=0), n do
                if H(3/4) < H(2/4)
                        compute Quadrant(4/4);
                end;
                if H(3/4) > H(2/4)
                        save k = i+1;
                        compute Quadrant(4/4);

                else
                        compute Quadrant(4/4);
        end;
end;
```

그림 5 – Quadrant진행에 따른 전체 알고리즘

3.3 퍼지 추론 규칙을 이용한 갭 비용(gap cost)의 동적 조정

DNA 염기 서열 배치에서 갭 발생에 대한 penalty (cost)의 설정에 따라 갭 발생 장소와 길이, 횟수가 결정되어 정렬 형태에 중요한 영향을 주게 된다. 갭의 생물학적 의미는 하나의 DNA 염기 서열에서 삽입(insertion)이 발생했다거나, 다른 염기 서열에서 삭제(deletion)가 발생했다는 것을 의미한다. 이러한 삽입, 삭제는 아미노산의 치환보다는 드문 사건이라고 할 수 있다. 따라서, 이것은 치환의 경우보다 상대적으로 낮은 값, 즉 전체 상동성을 감소시키는 비교적 큰 절대치의 음수 값으로 설정한다. 갭 비용을 객관적으로 설정할 기준은 현재 없다. 따라서 본 논문에서는 중요 매핑 점수 인자인 갭 비용 μ를 퍼지 추론 시스템에 적용하여 염기의 품질정보와 길이에 따라 동적 조정한다. 기존의 매핑 점수 인자의 퍼지 추론 방법[8]은 길이의 한계치를 설정하고 추론하여 길이에 대해 능동적이지 못한 문제점과 최종 산정에 있어서 일치 점수와 갭 비용은 불일치 매핑 점수 결과의 ± 1로 두어 그 비율상의 차이가 없으므로 Scoring matrix를 구성하는 과정에 있어서 큰 변동이 없었고, 결과적으로 매핑 점수가 적절하게 산정되지 않았다. 따라서 본 논문에서는 이러한 문제점들을 개선한다. 퍼지 논리 시스템의 입력은 각 DNA 염기서열의 길이 차이의 비율과 DNA 염기 서열의 낮은 품질의 비율로 하고 출력은 갭 비용 μ이다. 퍼지 논리 시스템은 입력 신호의 퍼지화, 전문가의 지식에 기반을 둔 퍼지 규칙에 의한 퍼지 추론, 비 퍼지화로 구성된다. 퍼지 규칙 기반에서 사용되는 퍼지 제어 규칙은 최소-최대 추론 방법을 적용하여 추론하며, 비 퍼지화는 퍼지추론의 결과인 퍼지 값을 단일 실수 값으로 변화시키는 부분으로 본 논문에서는 식 16과 같은 무게 중심 법을 적용한다.

$$y^* = \frac{\sum \mu(y_i) x_i}{\sum \mu y_i} \tag{16}$$

퍼지 입력과 출력 소속 함수는 그림 6, 7과 같다. 그림 6(a)의 L은 비교될 DNA 염기 서열과의 길이 차이를 나타내며 식(17)에 의해 계산한다.

$$L = \frac{|m-n|}{\max(m,n)} \times 100 \tag{17}$$

그림 6(b)에서 Q는 낮은 품질의 빈도이며 식(18)에 의해 계산한다. 본 논문에서 낮은 품질의 염기는 품질점수 20점 이하의 염기를 의미한다.

$$Q = \frac{Number\ of\ Lowquality}{m+n} \times 100 \tag{18}$$

식(17)과 식(18)에서 m, n은 각 DNA 염기 서열 A와 B의 길이이다. 그림 6(a)에서 Low 구간은 길이 차이가 많이 나지 않는 구간이고, Middle 구간은 단편의 길이 차이가 중간 정도인 구간이고, High 구간은 길이 차이가 많이 나는 구간이다.

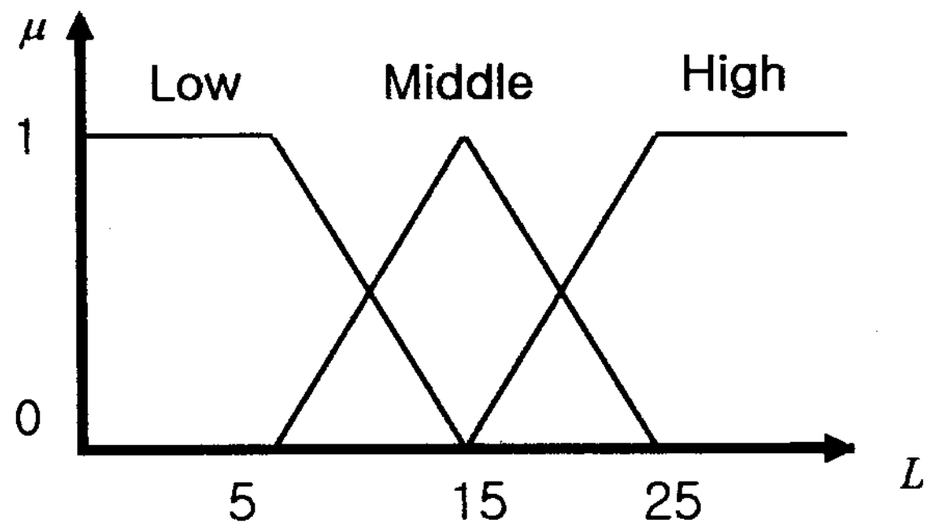

그림6(a) – 길이 차이 비율

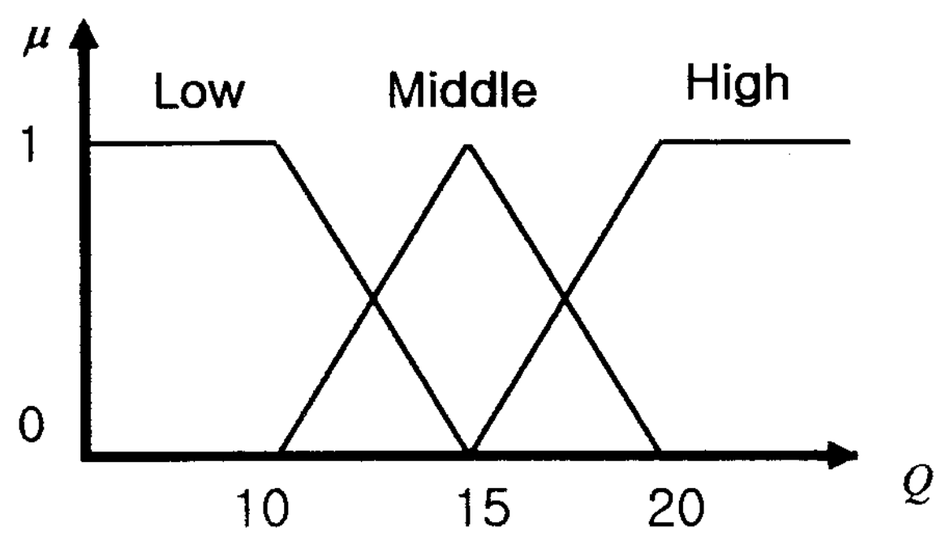

그림6(b) – 낮은 품질 빈도

그림 6 – 갭 비용에 대한 입력 소속함수

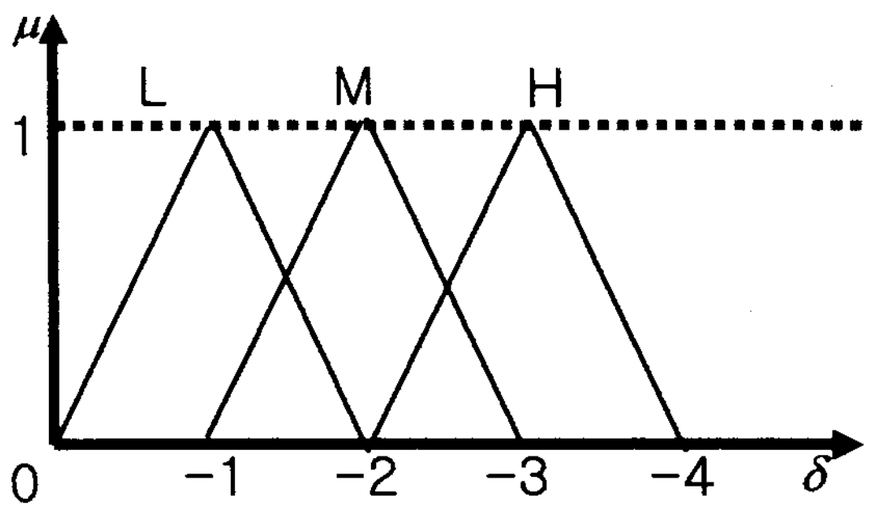

그림 7 – 갭 비용에 대한 출력 소속함수

본 논문에서 갭 비용에 대한 퍼지 제어 규칙은 표 5와 같다.

표 5 – 갭 비용(gap cost)에 대한 퍼지 추론 규칙

L \ Q	Low	Middle	High
Low	L	L	M
Middle	L	M	H
High	M	H	H

4. 실험 및 결과 분석

실험 환경은 Intel Pentium-VI 2GHz CPU와 512M RAM이 장착된 IBM 호환 PC상에서 VC++ 6.0으로 구현하였다. 실험에 사용된 DNA 데이터는 NCBI (http://www.ncbi.nlm.nih.gov/Traces/trace.fcgi)에서 실제 유전체 데이터를 받아 실험에 적용하였다. 본 논문에서는 다음과 같이 3가지 실험으로 비교 분석 한다.

1) 실험 **1.** 길이 차이가 많이 나고 낮은 염기의 빈도가 다소 높은 경우인 유전체 데이터인 gnl|ti|1147316797와 gnl|ti|1147316725의 서열배치
2) 실험 **2.** 서로 짝을 이루는 유전체 데이터인 gnl|ti|1147316842와 gnl|ti|1147316796의 서열배치
3) 실험 **3.** 한 쌍의 임의의 유전체 데이터인 gnl|ti|1147316818와 gnl|ti|1147316792의 서열배치

표 6은 실험 및 결과 분석에 사용된 각 유전체의 정보를 나타내었다.

표 6-실험에 사용된 DNA 염기서열의 세부 사항

염기 서열 명	A	G	C	T	평균 품질	길이
gnl|ti|1147316797	230	179	112	156	38	677
gnl|ti|1147316725	146	108	92	131	38	477
gnl|ti|1147316842	89	77	104	150	32	420
gnl|ti|1147316796	153	99	77	91	37	420
gnl|ti|1147316818	193	137	116	140	38	586
gnl|ti|1147316792	144	121	157	214	37	636

제안된 알고리즘을 구현한 프로그램 화면은 그림 8와 같다.

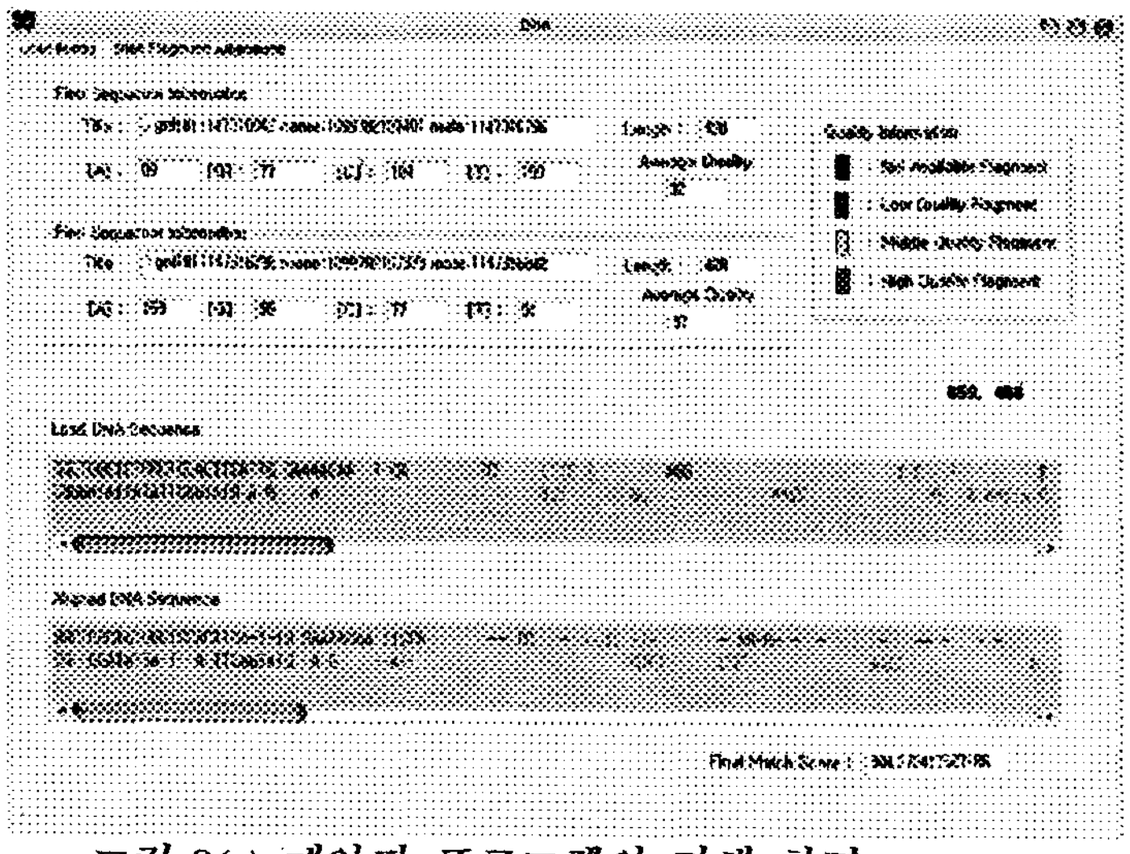

그림 8(a)-제안된 프로그램의 전체 화면

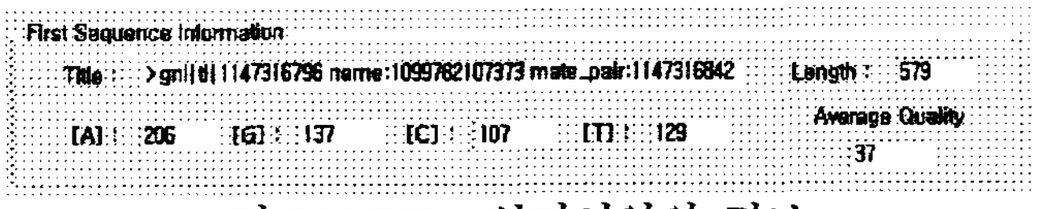

그림 8(b)-DNA 염기서열의 정보

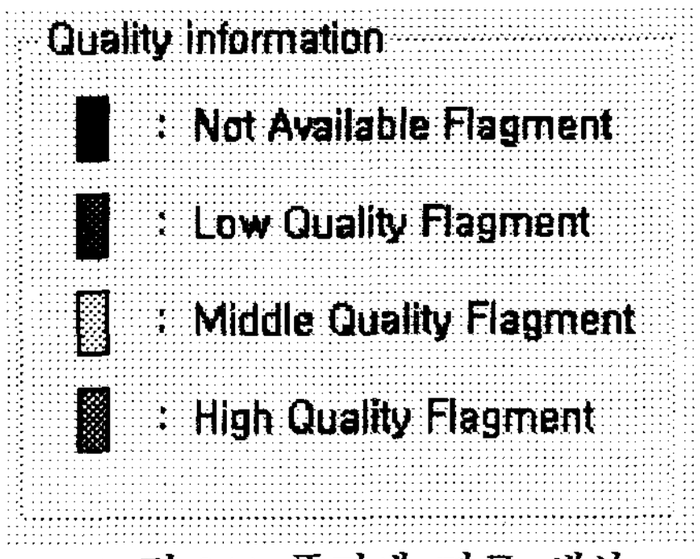

그림 8(c)-품질에 따른 색상

그림 8-제안된 DNA 염기 서열 배치 프로그램

제안된 방법을 구현한 화면은 각 염기의 이름과 길이 각 A, G, C, T 염기의 개수를 볼 수 있도록 하였고 그림 8(a)에서 프로그램 하단에 DNA 염기 서열을 불러온 화면과 정렬된 결과를 표시한다. 이 때 각 염기는 품질에 따른 색상으로 표시된다. 그림 8(c)의 품질에 따른 색상 기준은 표 7를 따른다.

표 7- 품질에 따른 색상 분류

	품질 점수
Not available	0
Low quality	1~20
Middle quality	21~40
High quality	41~99

4.1 Quadrant 방법을 적용한 Scoring Matrix의 계산

기존의 동적 프로그래밍 기법에서는 $O(nm)$ 의 시간 복잡도를 가지는 연산으로 Scoring matrix를 계산하고 대략 $O(n+m)$ 의 비용으로 역 추적을 통해 최종 정렬을 구했으나, 본 논문에서는 Quadrant방법을 통해 Scoring matrix의 계산량을 대폭 줄였다. 표 8에서 Scoring matrix를 구하는 기존의 방법과 Quadrant방법의 계산 횟수 실험 결과를 비교하였다.

표 8 – Scoring matrix 연산 횟수 비교

	기존의 연산 횟수	Quadrant방법의 연산 횟수
실험 **1**	322929	193757
실험 **2**	176400	91728
실험 **3**	372696	167713

표 8의 실험 결과에서 기존의 방법에 비해 Quadrant방법의 연산횟수가 더 효율적임을 확인할 수 있다. 알고리즘의 시간은 1:1 쌍 정렬이기 때문에 실질적인 차이를 보기 힘들었으나, 차후 알고리즘이 BLAST와 FASTA와 같은 DB기반의 검색을 위한 구현 시 많은 시간상의 차이를 보일 것으로 예상된다. 그림 9은 Needleman-Wunsch 알고리즘에서 Quadrant 방법을 적용하지 않은 경우와 Quadrant 방법을 적용한 경우의 Scoring matrix의 예이다.

		G	C	T	G	G	A	A	G	G
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
G	-1	2	1	0	-1	-2	-3	-4	-5	-6
C	-2	1	4	3	2	1	0	-1	-2	-3
A	-3	0	3	3	2	1	3	2	1	0
G	-4	-1	2	2	5	4	3	2	4	3
A	-5	-2	1	1	4	4	6	5	4	3
G	-6	-3	0	0	3	6	5	5	7	6
C	-7	-4	-1	-1	2	5	5	4	6	6

그림 9(a) – Quadrant 방법을 적용하지 않은 경우

		G	C	T	G	G	A	A	G	A
	0	-1	-2							
G	-1	2	1	0						
C	-2	1	4	3	2					
A		0	3	3	2	1				
G		-1	2	2	5	4	3	2	4	
A			1	1	4	4	6	5	4	
G				0	3	6	5	5	7	6
C						5	5	4	6	6

그림 9(b) – Quadrant 방법을 적용한 경우

그림 9-각 알고리즘의 Scoring matrix 연산

그림 9(b)에서 빈 영역은 Scoring matrix 생성 과정에서 역 추적에 영향을 주지 않으므로 구할 필요가 없는 부분이다. 그림 9에서 회색 배경은 실제 역 추적이 발생하는 영역이다. 빈 영역의 계산 없이 정상적인 역 추적을 통해 기존의 방법과 동일한 최적 정렬을 수행하는 것을 확인할 수 있다.

4.2 퍼지 추론 규칙을 이용한 갭 비용 동적 조정 결과 분석

본 논문에서 퍼지 추론을 이용하지 않는 기존 방법의 매핑 점수는 일치 $\gamma = 1$, 불일치 $\delta = -1$, 갭 $\mu = -2$ 로 두고, 퍼지 추론 규칙을 적용한 알고리즘에서는 일치, 불일치 점수는 동일하게 설정하고 갭 비용 μ 는 퍼지 추론 규칙에 의하여 [-4, -1]로 동적으로 조정한다. 표 9는 각 실험에 대한 결과이다. 최종 배치 점수는 식(13)을 통해 계산한다.

표 9 – 두 알고리즘의 배치 결과 비교

	실험	일치	불일치	갭	점수
기존의 방법	1	363	77	272	509.16
	2	218	177	48	232.56
	3	321	247	84	351.60
퍼지 추론 방법	1	332	140	207	417.55
	2	242	130	94	304.27
	3	369	148	186	493.72

표 9에서 실험 1은 각 염기 쌍의 길이 차이가 많이 나므로 (677, 477로 200 bp 차이) 전역 정렬에서 해당 차이만큼 갭이 채워지게 되고, 품질 정보가 낮은 염기가 전체 염기의 10% 비율이었다. 따라서 갭 비용을 음의 방향(negative) 으로 증가하여 penalty를 부여할 필요가 있다. 퍼지 추론을 적용한 최종 정렬 결과에서 배치 점수가 낮게 산정 되었음을 볼 수 있다. 실험 2에서는 서로 짝(pair)을 이루는 염기로 길이차이가 없으며 낮은 품질의 염기가 7%비율이었다. 퍼지 추론 규칙에 의해서 갭 penalty가 줄어들었고 좀 더 최적의 배치를 이루었다. 실험 3에서는 낮은 품질의 염기가 5% 였으며 길이 차이는 7% 였다. 실험 3 역시 기존의 방법보다 퍼지 추론을 적용한 방법이 좀더 최적 배치를 이루었다. 따라서 본 논문에서 제안한 DNA 염기 서열 배치 알고리즘이 기존의 방법보다 효율적인 계산으로 최적 정렬을 구했으며 퍼지 추론 시스템을 적용하여 품질과 길이에 따른 갭 비용의 동적 조정으로 보다 지능적인 DNA 염기 서열 배치를 보였다.

5. 결론

DNA 염기 서열 배치 알고리즘은 다양한 방법으로 개선되고 발전해왔다. 본 논문에서는 기존의 동적 프로그래밍 기법 기반의 전역 배치 알고리즘에 Quadrant방법을 적용하여 계산 복잡도를 효율적으로 줄였으며 PHRED 염기 결정 프로그램에서 생성된 품질 정보를 이용한 서열 배치 알고리즘에 퍼지 추론 시스템을 적용하여 갭 비용을 동적 조정하는 알고리즘을 제안하였다. 실험 및 분석 결과에서 상대적으로 낮은 품질과 길이 차이가 나는 서열, 혹은 그렇지 않은 DNA 염기 서열을 구분하여 효과적으로 배치 하였고 효율적인 계산으로 Scoring matrix를 계산함으로써 기존의 알고리즘보다 효율성과 배치율이 개선된 것을 확인하였다.

참고문헌

[1] Waterman, M. S., Introduction to Computational Biology, Champman and Hall, 1995.

[2] Gusfield, D., Algorithms on Strings, Trees and Sequences: Computer science and Computational Biology, Cambridge University Press, 1997.

[3] Apostolico, A. and Giancarlo, R., "Sequence Alignment in Molecular Biology," Journal of Computational Biology, Vol.5, No.2, pp. 173-196, 1998.

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

[5] Ewing, B., Hillier, L., Wend, M.C. and Green, P., "Base-calling of automated sequencer traces using phred. 1. accuracy assessment," Genome Research, Vol.8, No.3, pp.175-185, 1998.

[6] 나중채, 노강호, 박근수, "품질 정보를 이용한 서열 배치 알고리즘", 정보과학회논문지, 32권. 제11호, pp. 578-586, 2005.

[7] Needleman, S.B. and Wunsch, C.D., "A general method applicable to the search for similarities in the amino acid sequences of two proteins," Journal of Molecular Biology, Vol.48, pp.443-453, 1970.

[8] 이상수, 박충식, 김광백, "품질 정보 및 퍼지 추론 기법을 이용한 DNA 염기 서열 배치 알고리즘", 한국지능정보시스템학회추계학술발표논문집, pp. 201 -209, 2006.

[9] 안희국, 노희영, "유전자서열 정렬을 위한 Dynamic Programming Algorithm의 개선", 기초과학연구, Vol.16, pp. 89~105, 2005.