제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

RIFLE 알고리즘에 대한 실험 및 성능평가

김동회°, 원영상, 고영웅, 김진
한림대학교 정보통신공학부
e-mail:{kdh,wonys,yuko,jinkim}@hallym.ac.kr

Experiment and Performance Evaluation of RIFLE Algorithm

Dong-Hoi Kim°, Young-Sang Won, Young-woong Ko, Jin Kim
Dept of Computer Engineering, Hallym University

요약

서열의 유사성 검색에 잘 알려진 도구로는 BLAST 와 FASTA 가 있으며 이들 알고리즘은 알려지지 않은 유기체를 sequencing 작업을 통하여 얻어진 염기서열과 유전자 데이터베이스를 대상으로 유사성을 검색한다. 이때 서열의 유사성을 검색하기에 앞서 선행 되어야만 하는 sequencing작업은 시간적인 면에서 상당한 비용을 요구한다. 반면 sequencing 작업을 하기 않고도 간단한 실험에 의해 얻을 수 있는 부분적인 서열정보만을 대상으로 데이터베이스에서 검색 할 수 있는 알고리즘으로 RIFLE가 있다. 본 논문에서는 RIFLE 알고리즘을 구현하고 실험데이터를 생성하여 성능에 대한 분석 평가를 하고자 한다. 성능평가 결과 RIFLE 알고리즘은 시간복잡도 $O(n^2)$으로 빠른 반면 일부 서열에 있어서 실제 유사도에 비해 정확도가 낮게 평가되는 결과가 산출되었다.

1. 서론

생물정보학은 생물학과 관련된 데이터를 컴퓨터를 이용 정리, 분석, 이용하기 위한 연구 분야이다. 그중 서열의 유사성을 검색, 분석하는 문제는 분자생물학의 여러 분야에 걸쳐 상당히 중요시 되는 문제에 해당된다. 서열의 유사성 검색을 위한 대표적인 알고리즘으로는 BLAST[1]와 FASTA[2]알고리즘이 있으며 이들 알고리즘은 유사성 검색을 위해 알려지지 않은 유기체로부터 서열을 얻어내는 sequencing 작업이 선행되어야 한다. sequencing 작업은 시간적으로나 비용적인 면에서 상당한 비용을 요구한다. 반면 RIFLE[3] 알고리즘은 간단한 실험에 의해 얻을 수 있는 부분적인 서열을 대상으로 서열의 유사성을 검색할 수 있으므로 별도의 sequencing 작업을 필요로 하지 않는다. 본 논문에서는 RIFLE 알고리즘에 대한 구현 및 성능평가에 대해 논한다. 2장에서는 RIFLE 알고리즘에 대하여 설명하고 3장에서는 실험에 사용된 서열 데이터베이스 및 질의 데이터 생성 방법에 대하여 설명하고 4장에서는 실험 결과 분석 및 평가에 대하여 설명한다. 마지막으로 5장에서 결론을 맺는다.

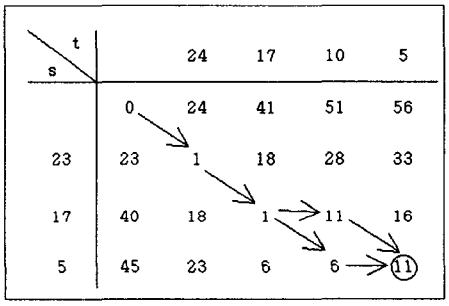
2. RIFLE 알고리즘

RIFLE 알고리즘은 두 개의 restriction pattern에 대한 optimal distance value를 산출하여 유사성 정도를 측정한다. Restriction pattern은 restriction map과 restriction profile로 구분된다. restriction map은 이미 서열을 알고 있는 염기서열 데이터베이스에서 얻은 restriction pattern이기 때문에 제한효소(Enzyme)에 의해 절단이 되었더라도 단편들의 순서는 정해져 있다. 반면 restriction profile은 서열을 알지 못하는 새로운 유기체에 대해서 실험으로 얻어진 길이 정보로서 단편들의 순서는 알 수 없다. REFLE은 restriction map과 restriction profile에 대해 유사성을 측정한다. 즉 REFLE은 순서가 있는 restriction map을 순서 없이 알고리즘에서 정렬 사용함으로서 restriction profile로 사용한다.

RIFLE은 Dynamic programming[4]을 기반으로 한

다. 이 Dynamic programming은 최적인 유사서열을 찾는데 사용되며 서열비교(sequence comparison)와 서열정렬(sequence alignment)에 많이 쓰인다. Dynamic programming을 사용할 경우 문제 해결을 위해 score matrix가 필요하지만, RIFLE은 단편의 길이에 대한 계산임으로 score matrix가 필요 없다. RIFLE은 다음과 같은 과정에 의해서 계산된다.

1) restriction profile s와 restriction map t를 입력받는다.
2) 두 서열 s와 t에 대해서 [그림 1]과 같이 Dynamic programming을 적용한다.
3) score 함수 $f1$ 은 $f1(x,y) = |x-y|$이고 $f1(x,0) = x$ 이다.
4) 다음과 같이 restriction profile과 restriction map이 주어졌을 때
restriction profile s=(23,17,5)
restriction map t=(24,17,10,5)
두 서열의 거리 $Df1(s,t) = 11$이다.



[그림 1] fragment length distance

3. 실험 데이터 생성 방법

실험 데이터는 restriction pattern 1200개와 restriction site map 1200를 대상으로 하였다.

3.1 데이터베이스

실험 데이터베이스는 Ribosomal Data Project(RDP) [5]를 사용하였다. RDP는 16s rRNA 유전자들로 구성된 염기서열 데이터베이스이다. 이 데이터베이스 내의 서열들은 계통학적 발생관계(Phylogenetic relatedness)가 이미 알려져 있다. 실험에서는 서열들 중에서 Bacteriol 부분의 2000개 정도를 실험에 사용하였다. 이 서열들은 공통 조상들로부터 파생되었기 때문에 서열간의 유사도가 높다. 즉 서열의 유사도가 확정된 데이터이다. 이 데이터베이스의 서열들은 GenBank 형식으로 되어 있으며 본 논문에서는 실험을 목적으로 서열들에서 ORF(Open Reading Frame)부분을 추출하여 FASTA 형식으로 전환 사용하였다.

3.2 DB 서열의 길이정보

제한효소를 사용하여 서열들을 절단한 다음 각 조각 패턴들의 길이 정보만을 수치 값으로 변환하여 새로운 길이 정보 데이터베이스를 구축하였다. 실험에 사용된 제한효소는 Hpa II, Hinf I, Rsa I, Hha I 이며 이 제한 효소들은 다음의 특성을 가진다.

Hpa II - C / CGG
Hinf I - G / ANTC
Rsa I - GT / AT
Hha I - GCG / C

각 제한효소들은 서열에서 일치하는 부위를 찾아 '/' 부분을 절단한다. 하나 또는 여러 개의 제한효소를 서열에 적용하였으며 각각의 위치는 Sliding Window 방식을 사용하여 구현하였으며 소스 코드는 [그림 2]와 같다.

```
int cutEnzyme(int ecnt, char enz[10][100], char Seq[10000], char Name[10000]) {
  int i, j, k, misM, cNo, cPos, lPos;

  for(i=0, cNo=0, cPos=0, lPos=0; i<strlen(Seq); i++) {
     for(j=0,misM=1; j<ecnt&&misM==1; j++) {
        for(k=0, misM=0, k<strlen(enz[j])&&misM==0;k++) {
           if(Seq[i+k]!=enz[j][k]&&enz[j][k]!='N') {
              misM = 1;
           }
        }
        if(misM==0) {
           switch(j){
              case 0:
              case 1: cPos = i+1;
                 break;
              case 2: cPos = i+3;
                 break;
              case 3: cPos = i+2;
                 break;
           }
           ptrn[cNo] = cPos - lPos;
           cNo++;
           lPos = cPos;
        }
     }
  }

  ptrn[cNo] = strlen(Seq) - lPos;
  cNo++;

  return cNo;
}
```

[그림 2] 제한효소를 이용한 서열절단 코드

[그림 3]은 env.OPB13 서열을 각 제한효소를 사용해 절단한 결과이다.

Hinf I Enzyme을 적용한 Fragment Length

```
env.OPB13
1342
```

Hpa II Enzyme을 적용한 Fragment Length

```
env.OPB13
61 179 17 348 103 13 143 211 21 11 212 23
```

Hpa II, Hinf I Enzyme을 동시에 적용한 Fragment Length

```
env.OPB13
61 108 71 17 222 126 103 13 143 108 69 34 21 11 60 152 23
```

Hpa II, Hinf I, Rsa I Enzyme을 동시에 적용한 Fragment Length

```
env.OPB13
61 179 17 348 103 13 143 211 21 11 212 23
```

Hpa II, Hinf I, Rsa I, Hha I Enzyme을 동시에 적용한 Fragment Length

```
env.OPB13
61 179 17 348 103 13 143 211 21 11 212 23
```

[그림 3] 제한효소를 이용한 길이정보 추출 예

실험에 사용된 서열들에 대하여 [그림 3]과 같은 길이정보 만을 갖는 데이터베이스를 구축하였다.

3.3 query 서열 길이정보

본 논문에서는 query 서열을 실제 생물학적 실험에 의해서 구한 것이 아니라 서열 데이터베이스 내의 서열들로부터 얻어진 restriction site map에서 각각의 단편들에 0~5%의 랜덤 오차를 적용하여 생성하였다. [그림 4]는 본 실험에서 restriction site map들에 대하여 restriction pattern으로 변환하기 위해 작성된 코드의 일부이다.

```
#defint EB 5

(생략)

tmpfrag = strtok(fSet, " ");

while( tmpfrag != NULL) {
      dbSeq[dfno] = atoi(tmpfrag);
      dfno++;
      tmpfrag = strtok(NULL, "");
}

fprintf(qF, "%s" , dname);

for(k=0; k<dfno; k++) {
      size = dbSeq[k];
      valueEB = dbSeq[k] + (rand() % EB) *size/100;
      if(k==(dfno-1))
          fprintf(qF, "%d" , valueEB);
      else
          fprintf(qF, "%d " , valueEB);
}
```

[그림 4] restriction site map을 restriction pattern으로 변환하는 소스

[그림 5]는 [그림 4]의 코드를 사용 변환된 결과이다.

변환되기 전의 restriction site map

```
env.OPB13
61 108 71 17 222 126 103 13 143 108 69 34 21 11 60 152 23
```

변환된 restriction pattern

```
env.OPB13
62 109 72 17 228 126 104 13 148 109 70 34 21 11 61 153 23
```

[그림 5] 변환 결과

4. 실험결과 및 분석

RIFLE 알고리즘의 실험은 하나의 서열이 모든 데이터베이스를 대상으로 RIFLE 알고리즘을 적용하여 Distance를 구하고 이 Distance로 유사성 검사 결과에 대한 Rank를 적용한 후 Rank 20 까지 검색하고 실제 유사도와 비교하여 얼마나 신뢰성 있는 검색을 하는지를 알아보았다. [그림 6]은 env.OPB13 서열에 대한 RIFLE에서 검색된 Rank 결과와 실제 유사도를 비교한 결과이다.

env.OPB13										
Rank	Hpa II				Hpa II, Hinf I, Hha I, Rsa I				Similarity	
	Name	Dist	Frag	Sim	Name	Dist	Frag	Sim	Name	Sim
1	env.OPB13	14	12	100	env.OPB13	19	17	100	env.OPB13	100
2	AF018192	14	12	98	AF018192	19	17	98	AF018192	98
3	AF018196	14	12	98	AF018196	147	16	98	AF018196	98
4	AJ240998	159	12	71	env.WCHB25	158	24	66	AF068801	88
5	Mlb.organ2	164	11	71	AB015887	163	18	66	AF068807	88
6	AB015560	177	11	71	Hp.foetida	166	17	68	AF068791	87
7	Y14312	184	11	73	Mlb.organ2	167	17	71	Aqu.pyrophL	76
8	Srh.teran2	184	13	70	AF068807	168	18	68	Hdg.subterL	76
9	Srh.terang	184	13	70	AF033558	168	18	70	str.EM_17	75
10	Z78203	184	13	70	AF036343	169	18	66	env.OPB_5	74
11	env.RB_40	191	10	69	Mlb.radtol	171	18	71	env.OPB23	74
12	AJ231189	193	10	67	Mlb.spF73	171	18	71	AF068788	74
13	U58018	194	10	71	Mlb.mesph1	171	18	72	env.OPB45	73
14	Mlb.spF48	194	10	72	Mlb.spF18	171	18	72	AJ009501	73
15	Mlb.extor4	194	10	72	D.rediodur	172	17	68	Tdv.TGE_P1I	73
16	Mlb.GK101	194	10	72	Blb.BF15	173	18	71	AJ237665	73
17	Mlb.rhodi2	194	10	72	Blb.PC3039	177	18	70	Tt.subtrrnI	73
18	Mlb.GK118	194	10	71	AJ009501	180	16	73	Ctm.prtlyt	73
19	AJ009481	196	11	67	AJ009451	181	18	73	Rht.marin2I	73
20	AJ009451	200	12	73	U81656	182	17	67	Rht.marinuI	73

[그림 6] RIFLE 결과와 실제 유사도 비교 결과

[그림 6]의 결과에서 보는 것과 같이 실제 유사도 순위와 RIFLE에서 제시된 유사도 Rank와의 차이가 나는 것을 볼 수 있다. RIFLE 알고리즘은 휴리스틱 알고리즘이므로 정확한 순서를 기대하기는 어렵다. 즉 연구대상은 전체 서열의 유사도가 아닌 유사성이 높은 서열들이다. 따라서 위의 예는 적절한 결과를 얻었다고 볼 수 있다. 그러나 실험 결과 분석 과정 중 두 가지 문제점을 발견할 수 있었다. 첫 번째 문제는 [그림 7]에서 보는 바와 같이 AF068791과 유사한 서열이 3개가 있다. 하나는 자신이고 다른 두 개는 98%로 상당히 유사하다. 반면 유사도가 높은 AF068807서열(유사도 98%)의 경우 RIFLE에서는 RANK가 11로 밀려난 것을 볼 수 있다. 이 경우 실제 유사도를 측정해 보지 않은 상태라면 유사도가

현저하게 떨어지는 다른 서열이 마치 AF068807서열보다 더 유사하다고 착각할 수 있다.

AF068791(15)						
#	similarity		RIFLE			
	name	%	name	dist	frag	%
1	AF068791	100	AF068791	22	15	100
2	AF068801	99	AF068801	24	15	99
3	AF068807	98	T.spTok8	188	14	73
4	AF018196	88	D.geother2	190	16	70
5	AF018192	88	D.geother3	190	16	70
6	env.OPB13	87	D.geother4	190	16	70
7	Aqu.pyroph0	79	D.geother5	190	16	70
8	Hdg.subterN	78	str.EM_17	192	14	76
9	str.EM_17	76	AF020205	194	15	73
10	Tt.maritimL	76	T.spNMX2	194	15	74
11	Tt.subtrrnK	75	AF068807	204	14	98
12	env.OPB45	75	Tt.subtrrn	206	16	75
13	env.OPB14	74	T.spTok20	212	15	73
14	T.thmoph27J	74	T.spTok3	212	15	73
15	T.thmoph13J	74	T.spW28	212	15	73
16	T.spFiji3AJ	74	T.spYSPID	220	17	73
17	T.spNMX2	74	T.aquaticu	220	17	73
18	T.spHS	74	T.spZHGI	222	16	73
19	AB020888	74	env.OPB80	224	17	71
20	T.flavus2	74	env.OPB90	224	17	71

[그림 7] RIFLE 결과 분석 1

두 번째 문제는 [그림 8]에서 보는 바와 같이 Rank 4를 기록한 env.WCHB25 서열은 AF018192 서열과는 restriction pattern의 개수가 많이 차이가 난다. 실제 검색 대상인 AF018192 서열은 17개의 단편을 갖고 있는 반면 env.WCHB25 서열은 24개의 단편을 갖고 있다. 또한 실제 유사도가 66%로 낮은 서열이 상위에 Rank되는 것을 볼 수 있다.

AF018192						
#	similarity		RIFLE			
	name	%	name	dist	frag	%
1	AF018192	100	AF018192	22	17	100
2	AF018196	99	env.OPB13	22	17	98
3	env.OPB13	98	AF018196	150	16	99
4	AF068801	89	env.WCHB25	155	24	66
5	AF068791	88	AJ009501	171	16	73
6	AF068807	88	AF033558	171	18	71
7	Aqu.pyrophL	76	AF068807	171	18	88
8	Hdg.subterL	76	D.radiodur	175	17	69
9	str.EM_17	75	AF068801	197	19	89
10	AJ237665	73	Y13595	199	19	71
11	Tt.subtrrnI	73	T.sp_ac2	199	19	71
12	Tt.maritimI	73	AF068791	215	19	88
13	Ctm.prtlytI	73	D.murrayi2	215	17	69
14	env.OPB45	73	D.radiodu2	217	16	69
15	AJ009501	73	D.prtlytic	219	19	69
16	Nsp.moscovI	73	D.murrayi3	225	16	69
17	AJ224039	73	D.murrayi1	225	16	69
18	AJ224042	73	AJ237665	225	18	73
19	Tdv.TGE_P1I	73	AJ224041	227	18	72
20	AF018195	72	str.EM_17	257	19	75

[그림 8] RIFLE 결과 분석 2

이러한 방식의 실험으로 100개의 서열들에 대해서 여러 가지의 제한 효소들을 적용한 결과 D.radiopug 서열만을 제외한 나머지 서열에서 유사한 문제점을 발견할 수 있었다.

5. 결론

본 논문에서는 간단한 생물학적 실험에 의해 얻어지는 서열에 대하여 제한 효소를 이용하여 단편의 길이 정보를 구하고 이 단편 길이만으로 유사성 검색을 하는 RIFLE 알고리즘에 대해서 분석을 하였다. 실험을 위해 길이정보를 가지는 데이터베이스와 0~5% 오차를 가지는 query 서열을 생성하고 이 데이터를 이용 REFLE 알고리즘을 구현해 적용하여 결과를 산출하였다. 실험결과 RIFLE은 Dynamic Programming을 기반으로 한 알고리즘이기 때문에 시간 복잡도는 $O(n^2)$으로 빠른 검색속도를 가진다. 반면 RIFLE이 산출해낸 결과는 실험결과 및 분석에서 본 것과 같이 실제 유사성이 높은 서열이 다른 서열보다 하위에 Rank되는 경우와 실제 유사성이 낮은 서열이 상위에 Rank되는 경우가 발생하는 것을 확인 할 수 있었다.

참고문헌

[1] http://www.ncbi.nlm.nih.gov

[2] Pearson, W. R. "Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms." Genomics, 11 (3), 635-50. 1991

[3] Hermjakob, H and Giegerich, R. and Arnold, W "RIFLE: Rapid Identification of Microorganisms by Fragment Length Evaluation" AAAI Press Menlo Park, CA, USA P:131-137

[4]Pearson, W. R. and Miller, W. Dynamic programming algorithms for biological sequence comparison. Methods Enzymol, 210, 575-601. 1992

[5] http://rdp.cmd.msu.edu/download/SSU_rRNA/alignments