최근 들어 바이오인포매틱스가 크게 주목받고 있는 이유는 인간 유전체 사업이 시작되면서부터 인간의 유전정보가 바로 상업적인 성공과 이어졌기 때문이다. 이 글은 바이오인포매틱스에서 가장 접근하기 쉬운 분야라고 할 수 있는 DNA 서열 분석을 중심으로 설명하고자 한다. 이 글을 통해 컴퓨터 과학에서 많이 연구된 스트링 처리 방법론이 바이오인포매틱스의 유전자 정보처리에 어떻게 활용되는지 잘 살펴보기 바란다.

조환규 · 진희정 hgcho@pearl.cs.pusan.ac.kr · hjjin@pearl.cs.pusan.ac.kr
조환규 씨는 부산대학교 정보컴퓨터 공학부 교수이며, 알고리즘 이론과 컴퓨터 그래픽은 물론 바이오인포매틱스에도 높은 관심을 갖고 있다. 진희정 씨는 부산대학교 전자계산학과 석사 과정에 재학중이다.

특집 II-1 컴퓨터 전공자를 위한 바이오인포매틱스

DNA 서열 분석을 통한 바이오인포매틱스 입문

생물학 교육을 제대로 받지 못한 컴퓨터 전공자가 처음으로 바이오인포매틱스를 접했을 때 받는 느낌은 사뭇 황당하기까지 하다. 먼저 수리과학(Mathematical Science)의 일종인 전산학에서는 항상 모든 것을 정량화할 수 있고 그 사실 여부를 판단할 수 있는 실험도 단시간 내에 가능하다. 이에 비해 생물학을 포함하고 있는 생명과학(Life Science)의 대부분은 즉각적인 실험이나 모두가 인정할 수 있는 어떤 이론적인 사실이 훨씬 적다. 예를 들어 생물의 진화에 관한 이론만 해도 수없이 많으며, 질병 · 원인 · 진단에 관한 의견도 매우 다양하고 많을 수 있다. 쉽게 말하자면 생명과학에 대해서는 '정답'이 없다는 것이다. 이것이 이 분야의 연구를 어렵게도 만들지만 한편으로는 매우 재밌게 만들기도 한다.

지난달 덴마크 올흐스(Arhus) 대학에서 열린 국제 바이오인포매틱스 토론회에서 만난 한 일본인 교수가 우리에게 '바이오인포매틱스를 하니 연구비를 많이 받아서 좋겠다'는 식으로 농담반 진담반의 이야기를 했다. 컴퓨터 분야에서는 이 분야를 볼 때 많은 연구 문제가 남아있으며 돈벌이에 적합한 어수룩한 구석이 다소 많아 보인다는 것이다. 또한 막상 이 분야에 관심이 있는 컴퓨터 과학자는 많지만 실제 어떤 구체적인 일을 맡고 있는 사람은 별로 없기 때문에 컴퓨터 과학이나 IT 분야의 자체 경쟁에 비한다면 바이오인포메틱스에서 컴퓨터 기술 사이의 경쟁은 우리가 보기에도 아직 포화에 이르기까진 멀게만 느껴진다.

현재 바이오인포매틱스에 뛰어드는 컴퓨터 전공자의 수는 급증하고 있으며, 이들 중 많은 수가 서열 분석과 관련된 연구를 하고 있다. 서열 분석은 DNA 서열이나 단백질 서열과 같은 생물학적 서열을 분석하는 것을 말한다. 이것은 아주 기초적이면서도 필수적인 작업이라 할 수 있다. 이러한 분야에 컴퓨터 전공자가 많은 이유는, 서열 분석이 바이오인포매틱스의 한 분야이므로 생물학적 지식이 필요하지만, 비교적 낮은 생물학적인 지식으로도 접근할 수 있는 분야이기 때문이다. 얼마 전 미국의 16개 연구소 · 프랑스 · 독일 · 영국 · 일본 등의 국제 컨소시엄인 인간 유전체 사업(Human Genome Project, http://www.ornl.gov/hgmis), 미국 국립보건원(NIG) 내의 인간 유전체 연구센터(NHGRI, http://www.nhgri.nih.gov/ HGP/), 미국 생명공학 벤처 셀레라(Celera Genomics, CRA, http://www.celera.com/)는 2000년 6월 26일의 초안을 더욱 업데이트한 인간 유전체 지도를 올해 2월 11일 완성했다.

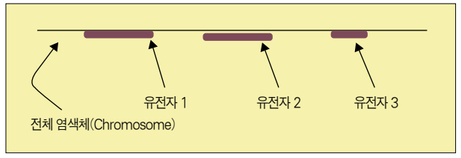
현재 인간 유전체의 염기 서열은 완성됐고, 인간 유전체 이외에도 model organism의 염기 서열도 다수 연구돼 대장균 · 효모 · 초파리 · 선충 등 20여 종의 유전체 염기 서열이 모두 밝혀졌다. 또한 100여 종의 생물 유전체가 연구되고 있다. 이렇게 밝혀진 염기 서열을 바탕으로 유전자의 기능을 연구하고, 개인 · 인종 · 생물 간 유전체 정보를 비교해 차이점을 밝혀내는 연구가 활발히 진행되고 있다. 이러한 연구를 위한 기본 작업이 서열 분석이다.

바이오인포매틱스의 기초 '서열 분석'

일반적으로 컴퓨터 전공자가 이 쪽에서 가장 쉽게 접할 수 있는 분야는 서열 분석(Sequence analysis)이다. '바이오인포매틱스가 어떻게 구성돼 있는가'는 보는 사람마다 다르겠지만, 일단 낮은 차원에서 서열처리에 관한 한 부분이 있을 것이고, 그 상위 단계에서는 각 유전체의 기능, 그들과의 관계, 그리고 그들이 신체내의 조직과 어떻게 결합하고 어떤 반응을 내는가 하는 단계로 분류할 수 있다.

일단 바이오인포매틱스에 관련된 자료 중 더 이상 쪼개질 수 없는 단위는 A(아데닌), G(구아닌), T(티민), C(시토신) 네 개로 분류되는 DNA 시퀀스다. 물론 이들 각 DNA를 다시 분자 구조로까지 분석하는 구조화학 분야가 있지만, 0과 1인 컴퓨터 과학에서 더 이상 분해가 불가능한 것을 하나의 아톰(atom)으로 다루듯 일단 이 네 개의 염기를 아톰으로 다룬다. 그

〈그림 1〉 염색체와 유전자와의 관계

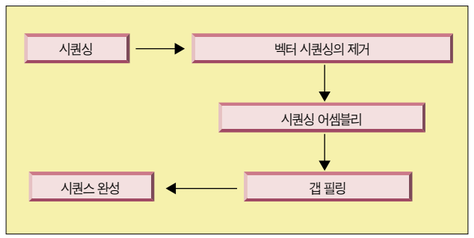


다음 단계는 세 개씩의 염기가 모인 아미노산 레벨이 있으며, 그 다음은 이들의 결집체인 단백질, 그리고 그러한 특정한 기능의 단백질이 모인 기관으로 점점 구성 레벨은 올라간다. 따라서 바이오인포매틱스를 연구하더라도 어떤 레벨에서 하는가에 따라서 입장은 완전히 달라진다. 마치 IT 분야에서 칩 레벨에서 문제를 보는가, 아니면 그 이상의 어셈블리 레벨, 그 다음 단계인 프로그래밍 언어 레벨, 더 나아가서는 시스템 통합(SI, System Integration)에서 문제를 보는가에 따라서 입장이 달라지는 것과 유사하다. 그런데 이 글에서는 상당히 낮은 차원의 서열에 관해 바이오인포매틱스에서 어떤 일을 하는지, 그리고 그러한 일에 컴퓨터 과학은 어떻게 이용되는지 살펴보고자 한다.

유전자 사냥하기

얼마 전 미국의 개인 기업인 셀레라 지노믹스에서 '인간의 염색체 서열을 모두 밝혔다'는 발표를 했다. 이것은 인간이 달에 첫발을 내디딘 일에 비견되기도 하는데, 이 일은 가장 중요하고도 초보적인 일이다. 그 다음에 할 일은 이 서열 중에서 어디에 어떤 유전자가 있는지를 판단하는 것이다. 염색체는 A, G, T, C 네 개의 문자로 구성된 하나의 스트링(String)이다. 왜 거의 모든 생물체의 염색체가 그 많은 위상 구조 중에서 하나의 긴 끈 모양의 스트링으로 돼 있는가는 아직 밝혀지지 않고 있지만 그것이 가장 간단한 구조이므로 진화하기에 용이해 지금까지 남아있는 것으로 추측하고 있다(하나의 세포에서 이 염색체의 DNA 서열을 밝혀내는 일과 서열 분석 알고리즘은 전체 유전체 시퀀싱과 결론의 글을 참조하기 바란다). 한 가지 지적할 것은 공공기관에서 발표한 인간 유전체 사업(HGP)의 인간 염색체 염기 서열이나 셀레라 지노믹스에서

〈그림 2〉 전체 유전체 시퀀싱 단계



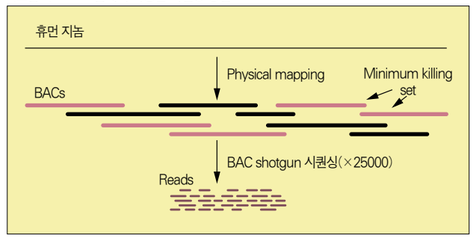
처음으로 발표한 염색체 서열 초안은 아직 미완성이라는 것이다. 다시 말해 알려진 어떤 생물체의 염색체의 서열은 그럴 것이라는 일종의 추측일 뿐, 그것이 정말 그렇다는 것에 대해서는 의문이 있다. 그것은 실제 염색체의 순서를 우리가 눈으로 확인할 방법이 없기 때문에 이러저러한 간접적인 방법을 이용해 그 서열을 추정할 뿐이다. 따라서 이름 있는 연구기관에서 같은 생물체의 염색체 서열을 밝힌다하더라도 그 서열의 순서는 작게는 10% 많게는 약 30%까지 서로 다를 수 있다. 실제 셀레라에서 사용한 방법을 검정한 한 연구논문에 의하면 그 결과는 심한 경우 약 50%까지는 바르지 않다고 하는데 이는 앞으로도 많은 논쟁거리를 남기고 있다.

이상을 비유해서 설명하면 다음과 같다. 염색체를 하나의 긴 스트링이라고 하면 유전자는 그 스트링 위에 띄엄띄엄 놓인 Substring이라고 생각하면 무방하다. 이 유전자가 나중에 세포 밖으로 빠져나가서 단백질이 된 뒤에 어떤 기능을 한다. 예를 들어 머리카락 색깔을 결정한다든지, 키와 같은 신체적인 특성부터 아직 논쟁중이지만 성격 · 언어능력 · 음악성 등도 결정할 수 있다. 또는 각종 유전병에 대한 정보도 이 모든 유전자에 쓰여져 있다고 생각하면 된다. 따라서 염색체 서열이 밝혀졌다고 하면 이제는 유전자를 찾아내는 일이 가장 중요한 것이다. 이름하여 유전자 사냥(Gene Hunting)이라고 불리는 이 작업이 지금 각국에서 경쟁적으로 진행중이다.

전체 유전체 시퀀싱

전체 유전체 시퀀싱(Whole genome sequencing)은 'genome(유전체)'란 한 생명체가 갖는 전체 DNA를 말하며, 'gene'과 'chromosome'의 합성어인 'genome'을 독일식으로 발음한 것으로 우리 나라에서는 '유전체'라는 용어로 통일해 사용하고 있다. DNA 염기 서열에는 어떠한 단백질을 만들 수 있는 정보가 들어 있다. 따라서 우리가 DNA 염기 서열을 알면 이 서열에서 어떠한 단백질이 만들어질 것이라는 것을 예측할 수 있게 된다. 얼마 전 발표된 사람의 전체 DNA 염기 서열의 양은 뉴욕 전화번호부 책 200권(30억 bp, bp란 DNA 염기 하나를 1bp라고 한다)에 해당하는 양이라고 한다. 이러한 염기서열의 배열을 알아내는 작업이 전체 유전체 시퀀싱이다. 전체 유전체 시퀀싱의 의의를 간단히 정리하면 다음과 같다.

〈그림 3〉 NHGRI팀에서 사용한 시퀀싱 방법



1 Comparative Sequence Analysis

인종간 · 생물간 혹은 개개인의 유전체 정보를 비교함으로써 많은 정보를 얻어낼 수 있다. 만약, 쥐의 염기 서열의 A라는 부분이 쥐의 눈을 만드는데 관여한다고 하자. 사람의 염기 서열을 쥐와 비교해봤더니 A와 아주 유사한 부분이 사람의 염기 서열에서 나왔다면, 정확하게는 알 수 없지만 그 부분이 사람의 눈을 만드는데 관여할 것이라고 예측할 수 있다.

2 Specific Gene hunting

전체 유전체 시퀀싱이 이뤄지면, 어떤 특정한 유전자를 찾을 수 있다. 예를 들어 제넥셀에서는 초파리를 이용해 치매 · 암 · 비만 · 파킨슨씨병 등을 일으키거나 억제하는 유전자 사냥에 나섰다. 지난 6월 초 세계 최대 규모인 6만 2400여종의 유전자변형 초파리 라이브러리를 완성했으며, 이미 치매와 대장암 관련 유전자 상당수를 1차 발굴했다.

3 Evolutionary Relationship

생물이 진화 과정을 거치면, 염기 서열에 여러 변화(삽입 · 삭제 · 변이 · 서열의 반복 등)가 일어나는데, 이러한 변화는 염기 서열에 그대로 나타나게 된다. 따라서 생물의 염기 서열을 분석하면 그 생물의 진화 과정을 알아낼 수 있다.

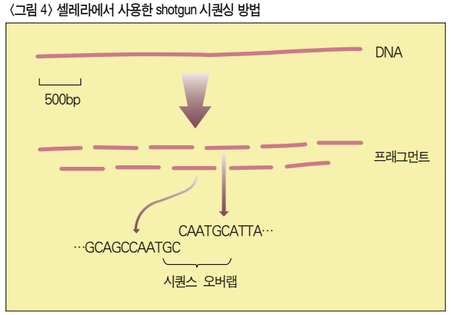
4 각 유전자의 기능 탐색

특정 유전자를 찾아내는 것과 마찬가지로 전체 유전체에서 어느 유전자 부분을 제거했을 경우 어떠한 일이 벌어지는지를 실험으로 관찰함으로써 그 유전자가 어떠한 기능을 하는 것인지를 알 수 있다.

5 Physical Genetic map construction

유전체의 전체적인 모습을 만들어낼 수 있다. 즉, 서열 A라는 부분은 전체 유전체에서 어디에 위치한다는 정보를 알 수 있다.

〈그림 4〉 셀레라에서 사용한 shotgun 시퀀싱 방법



전체 유전체 시퀀싱 단계를 살펴보면 다음과 같다.

1단계, 시퀀싱하기

시퀀싱은 염기 서열의 염기(A, T, G, C)를 해석하는 것이다. 이런 시퀀싱 방법은 NHGRI(국립 인간 지놈 연구소)의 HGP팀에서 사용한 방법과 셀레라에서 사용한 방법으로 나눠볼 수 있다. HGP팀에서는 전체 염기 서열에서 위치를 아는 어느 부분을 잘라내고, 그 잘려진 부분에서 또 위치를 아는 부분을 잘라내 서열을 읽을 수 있을 만큼 작은 조각을 만든다. 이후 그것을 읽어들여 하나로 만들고, 또 그것을 모아 더 큰 조각을 만드는 방식으로 전체 염기 서열을 분석하는 방법을 사용했다. 다시 말하면, 전체 DNA를 우리가 연구하기 좋은 길이로 모두 자른 다음, 염색체 지도를 완성해 우리가 만든 길이의 DNA 절편이 어느 염색체의 어느 부분에 있는지를 확인하고 해당하는 부분을 하나씩 서열 분석하는 방법이다.

이렇게 전체 염기 서열을 분석하면 위치를 아는 서열을 분석하는 것이므로 그만큼 정확할 수 있지만, 이런 방법은 많은 시간을 요구한다. 셀레라에서 사용한 방법을 'shotgun 시퀀싱'이라고 한다. 셀레라는 NHGRI보다 7년이나 늦게 인간 유전체 사업을 시작했지만, NHGRI보다 더 빨리 인간의 유전체를 발표했다. 이렇게 빠른 시간 내에 NHGRI를 따라 잡을 수 있었던 이유가 시퀀싱할 때 슈퍼컴퓨터와 shotgun 방법을 사용했기 때문이다. shotgun 방법은 일단 긴 서열을 무조건 잘게 잘라 그 각각을 해독한 다음 그 조각을 맞춰나가는 방법을 말한다.

이러한 shotgun 방법은 원핵생물(prokaryote, 핵막이 없는 생물로 모두 단세포로 되어 있다)의 유전체를 시퀀싱할 때 그 분석의 표준적인 방법이 된다. 그러나 이 방법은 두 가지 단점이 있다.

1 DNA 조각의 수가 증가할수록 그 조각이 서로 겹칠 수 있는 확률이 많아지기 때문에 정보 분석이 어려워진다.

2 서열에는 반복되는 부분이 있는데, 조각에 반복 서열(repeated sequence)이 있을 경우 정확한 분석이 어렵다.

shotgun 시퀀싱의 단점을 보완하기 위해 물리적인 맵이 필요하고 HGP에서 만든 맵을 사용해 보완했다. 그러면, 이렇게 서열을 작은 단위로 잘라야 하는 이유는 무엇일까. 그것은 현재의 기술로 엄청나게 긴 DNA 서열을 한번에 읽을 수 있는 기계가 없기 때문이다. 그리고 DNA 분석에 사용되는 효소도 한번 DNA에 붙으면 계속 분석이 가능하도록 DNA에 붙어있는 것이 아니고, 어느 정도 복제하면 DNA에서 떨어지기 때문이다. 따라서 DNA 서열을 여러 조각으로 나눠 작업을 해야하고, 잘려진 작은 조각이 전체 서열에서 어느 위치에 해당하는가를 확인하는 작업이 필요한 것이다.

2단계, 벡터 시퀀스의 제거

전체 염기 서열을 분석할 때에는 그 서열을 벡터(vector)에 넣어 여러 개로 복제시켜서 사용한다. 따라서 우리가 얻은 서열에는 벡터의 서열이 들어 있을 수 있다. 따라서 조각 서열을 해석해 그 중 벡터 서열이 들어 있다면 그것을 제거해야 한다.

3단계, 시퀀스 어셈블리

생물체의 전체 유전체 시퀀스를 얻기 위해서는 시퀀싱 방법이 'whole genome shotgun' 방식이든 'clone-by-clone' 방식이든 작은 조각으로 해석한 결과를 하나의 연속된 서열로 만들어야한다. 이 과정을 시퀀스 어셈블리(sequence assembly)라고 한다. 이렇게 하나의 긴 서열을 그 자체로 읽어 정보를 해석하지 않고 작은 조각으로 자른 다음 각각을 해석해 다시 하나로 만드는 작업이 아주 어리석어 보일지도 모른다. 하지만 현재로서는 전체 유전자 서열을 한 번에 읽어들일 수 있는 컴퓨터는 존재하지 않는다. 오늘날의 염기 서열 분석 기술은 한번에 750bp 정도를 읽을 수 있다. 따라서 큰 염기 서열을 읽어야 하는 경우에, DNA를 조각 내어서 염기 서열을 분석하고 이를 조합해 염기 서열을 완성한다. 시퀀스 어셈블리의 기본적인 원리는 작은 조각의 서열 사이의 유사성에 의해 각 조각 사이의 오버랩(overlap)을 찾고, 그 부분을 이어나가는 것이다. 이것은 아주 간단해 보이지만 실제로는 여러 다양한 이유로 아주 어려운 작업이 되며, 셀레라처럼 대규모의 슈퍼컴퓨터가 매우 절실

한 작업이다. 그러면 작은 서열 조각(fragment1, fragment2, fragment3)을 실제 손으로 어셈블리해보자.

```
Fragment1 : ATACATCATAACACTACTTCCTACCCATAAG
Fragment2 : CTTTTAACTTGTTAAAGTCTTGCTTGAATTAAAGACTT
Fragment3 : TTCCTACCCATAAGCTCCTTTTAACTTGTTAAA
```

앞의 세 조각 Fragment1, Fragment2, Fragment3의 오버랩을 각각 나타내면 다음과 같다.

```
(Fragment1)  ATACATCATAACACTACTTCCTACCCATAAG
(Fragment3)                 TTCCTACCCATAAGCTCCTTTTAACTTGTTAAA
(Fragment3)                                 CTTTTAACTTGTTAAAGTCTTGCTTGAATTAAAGACTT
```

앞처럼 오버랩이 된 서열을 하나로 연결하면 다음처럼 하나의 서열이 만들어진다.

```
결과 서열  ATACATCATAACACTACTTCCTACCCATAAGCTCCTTTTAACTTGTTAAAGTCTTGCTTGAATTAAA
```

하지만 이러한 경우는 아주 이상적인 경우다. 시퀀스 어셈블리에서 고려해야 하는 문제 중 가장 중요하면서도 어려운 문제로 반복 서열(repeated sequence)을 들 수 있다. 반복 DNA 염기 서열은 인간 유전체의 넓은 범위에 존재하며 개인에 따라 반복수도 차이가 난다. 이러한 연속 반복 서열은 그 반복 단위의 크기에 따라서 크게 STR(short tandem repeat, micro satellites)과 VNTR(variable number of tandem repeat, minisatellite) 두 종류로 나눌 수 있다. STR은 보통 2~5개의 염기 서열이 반복적으로 나타나는 것을 말하며, VNTR은 9~80개의 염기 서열이 반복적으로 나타나는 것을 말한다. 길이가 4인 염기가 반복적으로 나타나는 STR을 고려해보자.

```
기본 단위가 "AATT"인 STR
STR Sequence  CGGATCAATTAATTAATTAATTAATTAATTAATTGGACCT
```

이 STR 서열을 랜덤하게 잘랐을 때 네 조각(Fragment1, Fragment2, Fragment3, Fragment4)이 나왔고, 이것을 앞의 어셈블리처럼 오버랩을 찾아 하나의 서열로 만든다고 해보자.

```
CGGATCAATTAATTAATTAATTAATTAATTAATTGGACCT

Fragment1   CGGATCAATTAATTAA
Fragment2   AATTAATTAATTAATTAATTAATTAATTAATT
Fragment3   AATTAATTAATTAATTAA
Fragment4   AATTAATTAATTAATTGGACCT
```

앞의 네 조각 Fragment1, Fragment2, Fragment3, Fragment4의 오버랩을 각각 나타내면 다음과 같다.

```
Fragment1   CGGATCAATTAATTAA
Fragment3         AATTAATTAATTAATT
Fragment2         AATTAATTAATTAATTAATTAATTAATT
Fragment4                   AATTAATTAATTAATTGGACCT
```

앞처럼 오버랩이 된 서열을 하나로 연결하면 원래의 STR 서열에서 반복 염기 서열은 모두 8개였다.

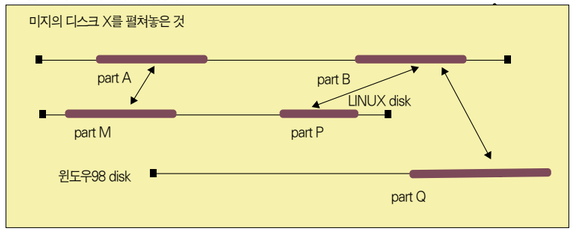
```
결과 서열    CGGATCAATTAATTAATTAATTAATTAATTAATTGGACCT
```

하지만 어셈블리하고 난 결과에는 7쌍이 존재한다. 이처럼 반복 서열이 들어있을 수 있기 때문에 어셈블리 작업이 더욱 어려워진다.

4단계, 갭 필링

시퀀싱을 아무리 잘 하더라도 결과로 나온 서열에는 에러가 있을 수 있다. 그리고 서열을 어셈블리하면 하나로 된 원래의 서

〈그림 5〉 디스크에서의 패턴 찾기



열이 나오는 것이 아니라 커다란 몇 개의 조각 서열이 나온다. 즉, 이들이 서로 연결되지 못하고 이들 사이에 공간(갭, gap)이 생긴다는 것이다. 시퀀싱을 통해 생길 수 있는 에러는 겹치는 클론의 서열을 통해 확인할 수 있다. 서열 사이의 공간은 시퀀스 어셈블리 프로그램과 근접한 클론의 겹쳐지는 부분의 서열을 비교함으로써 확인할 수 있다.

유전자 탐색하기

〈**그림 5**〉의 서열은 어떤 생물체의 염색체 서열인데 이렇게 거대한 서열에서 어떤 부분이 유전자인지, 다르게 말하면 한 유전자의 시작과 끝을 찾아내는 작업을 한다고 생각해보자. 별다른 지식이 없다고 하면 그야말로 망망대해에서 바늘 찾기와 유사할 것이다. 이 문제를 이런 것과 비교해 보자. 예를 들어 어떤 하드 디스크의 알맹이만 우리가 갖고 있다고 생각해 보자. 즉 이 하드 디스크를 어떤 운영체제에서 사용하는지 그리고 이 안에 어떤 파일이 얼마나 들어있는지 전혀 모르는 상태다. 이때 이 디스크 안에 들어있는 의미 있는 정보를 찾아내라는 문제는 마치 임의의 염색체 배열에서 유전자를 찾아내는 작업과 매우 유사하다고 할 수 있다.

우리에게 주어진 디스크의 내용은 오로지 0과 1의 배열만 존재한다. 어디가 디스크의 시작인지, 그리고 어디가 파일 테이블인지 전혀 모르는 상황에서는 매우 어려운 작업일 수밖에 없다. 이 경우 가장 쉬운 방식은 이미 알려진 윈도우 계열, 리눅스 계열 등의 구할 수 있는 모든 운영체제에서 사용되는 디스크를 뜯어와 그 안의 내용과 우리가 지금 밝혀내려는 미지의 디스크 X의 내용을 서로 비교해보는 것이다.

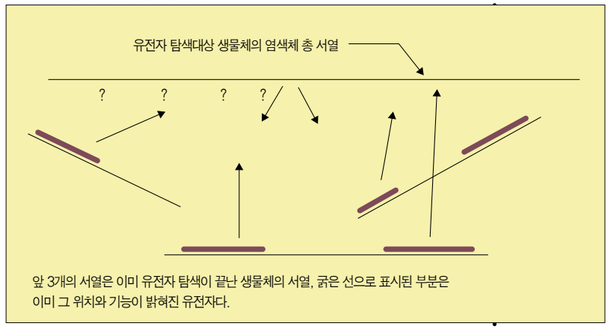
이 상황을 그림으로 살펴보자. 〈**그림 5**〉에서 가장 위에 놓여진 것은 디스크 X를 임의의 위치에서 잘라 펼쳐놓은 것이다. 그리고 그 다음에는 우리가 이미 알고 있는 디스크의 내용을 처음부터 끝까지 0과 1의 배열로 펼쳐놓은 것이다.

그래서 이 디스크 X와 나머지 리눅스와 윈도우 98의 디스크의 0과 1 서열을 비교해본 결과 다음과 같은 '유사한' 부분을 발견할 수 있었다고 가정해보자. 즉 리눅스 디스크의 part M과 디스크 X의 part A의 처음과 끝이 아주 유사했다고 말이다. 그리고 part B와 part P 역시 마찬가지다. 그런데 우리는 리눅스 디스크와 윈도우 디스크의 내용은 이미 알고 있으므로 그 part A, B와 유사한 부분인 M, P, Q가 각 OS에서 어떤 기능을 담당하는 파일인지를 판명하면 쉽게 part A와 B의 기능을 유추할 수 있을 것이다.

생물체의 유전자를 찾아내는 작업은 이와 매우 유사하다. 예를 들어 좀 극단적인 방법으로 어떤 부분이 인간의 눈을 만들어내는 유전자인지 찾아내려는 작업을 생각해보자. 하나의 우악스런 방법은 인간 배아를 선택해 눈을 만들 것이라고 생각되는 유전자 부분을 지워 본다. 그리고 그렇게 키운 수정란으로 생긴 아기를 낳아 그 아이에게 눈이 있는지를 알아보면 가장 확실하게 알 수 있다(특정한 부분의 유전자를 지우는 방법은 그 부분에 해당되는 다른 상보 서열의 유전자를 그쪽에 넣어 결합시키면 서로 상쇄가 돼 지워지는 효과를 거둘 수 있다). 그야말로 공상 과학에서나 나올 수 있는 방법이지만 이 방법은 사용할 수 없다. 그 첫 번째 이유는 독자 여러분도 느꼈을 것이지만 매우 비윤리적이고 위험한 방법이다. 그리고 현실적으로 가능하지 않는 이유로 인간과 같이 고등 동물에서는 그 정도의 유전자 손상이 있으면 대부분 착상되기 전에 유산돼 버리므로 유전적으로 일정 이상 손상된 태아는 태어나지 않는다. 그러나 초파리와 같이 비교적 하등한 곤충 등에서 이러한 유전적 손상에서도 별 무리 없이 후손이 태어나게 된다. 따라서 사람의 눈을 만드는 유전 인자를 찾는 일은 이미 밝혀진 초파리의 눈에 관련된 유전 인자와 유사한 서열이 인간 염색체 서열의 어디에 있는지를 발견하는 일이다.

그런데 이렇게 고등한 인간의 유전자 서열과 그에 비해 열등한 초파리의 유전자와 유사성을 우리가 판단할 수 있을 정도로 충분히 있는가 하는 일인데 불행히도(?) 상당히 유사함이 밝혀졌다. 몇 년 전에는 개구리의 눈에 관련된 유전자를 초파리의 염색체에 삽입해 개체를 발생시킨 결과 초파리의 다리와 날개 겨드랑이 부분에도 눈이 달려있는 기괴한 모습의 초파리를 탄생시킬 수 있었다. 이 사실이 말하는 것은 비록 개구리의 눈을 담당하는 유전자가 초파리의 것은 아니지만 초파리 안에서 별 무리 없이 반응한다는 사실을 뒷받침해주는 것이고, 이 사실은

〈그림 6〉 유전자에서 유사한 서열 찾기



이 두 종의 생물체가 하나의 공통된 조상으로부터 진화했다는 매우 강력한 증거이다.

스트링 매칭 알고리즘

이미 사람들은 하등한 동물에 대해서는 실험을 통해 매우 많은 유전자를 발견했다. 예를 들어 대장균의 경우에는 거의 모든 유전자에 대해 대장균 염색체에 어디에 위치하며, 그 기능이 무엇인지에 대해 상당한 수준까지 밝혀진 상태다. 따라서 고등 동물의 유전자 사냥의 핵심은 바로 이미 밝혀진 하등 생물의 염색체 서열과의 비교에 있으며 그 비교의 핵심은 바로 스트링 매칭(String Matching) 알고리즘에 있다. 특히 그 서열의 크기가 몇백 메가 단위의 A, G, T, C 서열에서 특정한 패턴을 얼마나 정확하게 빨리 찾아내는가 하는 것이 가장 중요한 작업이다. 물론 유전자를 찾아내는 방법은 이외에도 유전자가 시작되고 끝나는 부분의 특정한 패턴을 조사해 유전자를 검색하는 언어적인(Linguistic) 방법에 은닉 마르코프 모델, 또는 이미 밝혀진 유전자 서열에서 공통의 패턴을 기계학습을 통해 추출한 뒤에 이를 다른 미지의 DNA 서열에 적용시키는 방법 등에 사용된다.

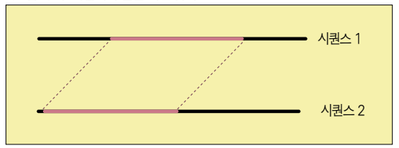
그런데 왜 생물체의 염색체 서열에서 유전자를 발견하는 것이 중요한 것일까? 이유는 인체의 모든 기능을 담당하는 최소한의 가치인 동시에 특정 질병의 치료를 위해 각 유전자에서 만들어지는 단백질의 기능을 차단하거나 강화하는 약물을 만들어야 하기 때문이다. 이렇듯 특정한 병을 일으키거나 그것을 방지하는 유전자의 위치와 그 내용을 판별하는 것은 과학적으로나 상업적으로 가장 중요한 작업이다. 예를 들어 어떤 특정한 유전자의 서열을 알아내거나 그 대사기능을 특허로 걸어둔다면 그 사실을 이용해 약품을 만들거나 치료하는 모든 행위는 일정 이상의 특허료를 반드시 지불해야 한다. 쉽게 말해 하나의 유전자는 바로 돈 덩어리 그 자체라고 보아도 별 무리가 없을 것이다. 이상의 모델 생물(초파리나 쥐, 개구리)을 이용해 유전자를 추적하는 과정을 도식으로 나타내면 **〈그림 6〉**과 같다.

즉 이미 알려진 생물체의 유전자 서열과 '유사한' 서열이 사람 염색체 서열의 어디에 있는가, 또는 사람의 DNA 배열 가운데 유전자로 의심이 가는 부분과 유사한 DNA 배열을 어떤 생물체가 갖고 있으며 그 기능은 무엇인가를 계산하는 문제가 바로 유전자 사냥의 핵심이다. 그런데 여기까지 이해했다면 우리 컴퓨터 전공자는 당장이라도 뭔가를 해볼 수 있겠지만, 문제는 생물학적인 의미로 그 서열의 '유사성'을 어떻게 계산하는가에 있다. 즉 AAATCCTA가 AAATTA와 더 유사한지 아니면 AAATCCCTA와 유사한지를 알 길이 없다. 즉 어떤 경우에는 어떤 배열에서 CC가 두 개 빠지는 것이 진화적으로 더 가능성이 있을 수도 있으며, 아니면 한 서열이 새로운 C가 하나 더 추가(insertion)되는 것이 더 가능성이 있다고 볼 수도 있다. 문제는 우리가 전 지구적 진화 과정을 관찰할 수 없었기 때문에 DNA 서열 사이의 실제적인 유사성을 알 수 없다는 것이다. 따라서 생물학자가 생각하는 의미 있는 유사성을 정의하는 방법은 매우 다양하고 그 절대적인 기준은 없다. 게다가 이 문제를 어렵게 하는 다른 이유는 각 생물체의 염색체 서열이 매우 길다는 것, 각 유사성을 비교할 기준이 매우 다양하다는 것, 그리고 이미 밝혀져 있는 서열 역시 충분한 정도의 오류가 있을 수 있다는 것 때문이다. 이제까지 독자 여러분은 바이오인포매틱스의 가장 기초적인 작업이 두 개의 DNA 스트링끼리 서로 비교하는 것이고 왜 그것이 중요한지에 대해 개략적으로 이해했을 것이다. 이제 그 대표적인 스트링 처리방법의 하나 하나에 대해 살펴보자.

바이오인포매틱스에 쓰이는 스트링 처리 방법론

일반적으로 컴퓨터 과학에서는 바이오인포매틱스 이전에 텍스트 정보 처리를 위해 많은 스트링 처리 알고리즘을 연구해왔다. 예를 들어 http://www-igm.univ-mlv.fr/~lecroq/string/에 가면 현재까지 컴퓨터 과학에서 개발한 거의 모든 스트링 알고리즘이 소개돼 있으며 관련된 도구까지 받아 올 수 있다.

〈그림 7〉 두 서열에서 같은 패턴 찾기



주로 컴퓨터 과학에서 스트링 매칭을 연구할 때의 주안점은 처리속도, 특히 그 알고리즘의 점근적 복잡도(Asymptotic Complexity)의 개선에 있다고 볼 수 있다. 즉 바이오인포매틱스에서 사용되는 스트링 처리 방법론의 주안점은 누가 얼마나 생물학적인 의미를 알고리즘에 반영해 궁극적으로 생물학자를 만족시키는가(to make biologists satisfied)에 달려있으므로 여간 어렵지 않다. 특히 최근에는 스트링 알고리즘만을 연구하는 학교가 스트링학을 일컫는 Stringolgy라는 단어를 만들어 본격적으로 다양한 변화의 스트링 알고리즘을 연구하고 있다. 이 그룹을 이끌고 있는 집단은 체코 프라하 기술대학(Prague Technical University)의 스트링 그룹(The Prague Stringology CLub)인데 http://cs.felk.cvut.cz/psc/#Links에 가면 관련된 많은 자료를 얻을 수 있다.

〈표 1〉 쥐와 사람의 염기서열 비교

쥐의 염기 서열	사람의 염기 서열
AAATACCTAATTCCCTAAACCCGAAACCGGTTTCTCTGGTTG	AGGGCTTTCGGCCCCCTCGTGCTATAGGTTCAAGGTTTATCCAACC
AAAATCATTGTGTATATAATGATAATTTTATCGTTTTTATGT	CCGAGCATGGTCTGAGCAGCCACCCGACTCAAATACACTATAAAAA
AATTGCTTATTGTTGTGTGTAGATTTTTTAAAAATATCATTT	TGTGGAAAAACATCACACCATACTTCATTAACCAAAACACGGTCTA
GAGGTCAATACAAATC**CTGTTTATGTTTGGACATTTATTGTC**	ATTTTTTTCCAAATAAGACCCTCCTCCCTCTTGTTACACCACGTAA
ATTCTTACTCCTTTGTGGAAATGTTTGTTCTATCAATTTATC	ATAAGGGACCATGATATCCAAGATTAGTTATGGAACAATGTCGTAT
TTTTGTGGGAAAATTATTTAGTTGTAGGGATGAAGTCTTTCT	AATTTCCTGGAAATCTCTCATCCCCGCATAATTGATTTGTGTACTG
TCGTTGTTGTTACGCTTGTCATCTCATCTCTCAATGATATGG	TCAAAATTTGAGTGCTCACTTCCATCAAGGATCTCATTGAAATCTC
GATGGTCCTTTAGCATTTATTCTGAAGTTCTTCTGCTTGATG	TCCAAATCATCCATGGTTTATTTCTAAACAATGGTGAATCATGATG
ATTTTATCCTTAGCCAAAAGGATTGGTGGTTTGAAGACACAT	ATAACATATATCCTCCCAAAGTGTTTTTCTTTCCTCTGCAAAATTG
CATATCAAAAAAGCTATCGCCTCGACGATGCTCTATTTCTAT	ATGCATAAACGAAAGTACAGAAAAATTCATTTTCTCCTCCTTCCAA
CCTTGTAGCACACATTTTGGCACTCAAAAAAGTATTTTTAGA	CAAAACAGAACAAGTGATCATTTGACTGCTCTTTCTCTCCAAACTA
TGTTTGTTTTGCTTCTTTGAAGTAGTTTCTCTTTGCAAAATT	CCGAAATGCGGCCTAACCGGTGTGATTCATAATTTTTCATAGATTA
CCTCTTTTTTTAGAGTGATTTGGACTATTTCTTGTGGTTTTC	CCAACCCTAAAAAGTTAATGACATAATACCAGATGCCTGAACTTCC
TTTCCTTCACTTAGCTATGGATGGTTTATCTTCATTTGTTAT	TTGACCCTCGTCTCTAAAAGACAATCAAACTAATGGAACAATGTCG
ATTGGATACAAGCTTTGCTACGATCTACATTTGGGAATGTGA	TATAATTTCCTGGAAATCTCTCATCCCCGTTTAGTCACACTGTTGT
GTCTCTTATTGTAACCTTAGGGTTGGTTTATCTCAAGAATCT	AATTGGTTTGGTCACAATTTGGTGACATCTTTTGGCTAATCATAAA
TATTAATTGTTTGGATGATTCAAGACTTCTCGGTACTGCAAA	CCTAATTAATTACCAAATTTGAAACTCTAGTAAACCACGTAAATAA
GTTCTTCCGCCTGATTAATTATCCATTTTACCTTTGTCGTAG	GGGACCATGATATCCAAGATTAGTTATGGAACAATGTCGTATAATT
ATATTAGGTAATCTGTAAGTCAACTCATATACAACTCATAAT	TCCTGGA**CTGTTTATGTTTGGACATTTATTGTCATTCTTACTCCTT**
TTAAAATAAAATTATGATCGACACACGTTTACACATAAAATC	**TGTGGAAATGTTTGTTCTATCAATTTATCTTTTGTGGGAAAATTAT**
TGTAAATCAACTCATATACCCGTTATTCCCACAATCATATGC	**TTAGTTGTAGGGATGAAGTCTTTCTTCGTTGTTGTTACGCTTGTCA**
TTTCTAAAAGCAAAAGTATATGTCAACAATTGGTTATAAATT	**TCTCATCTCTCAATGATATGGGATGGTCCTTTAGCATTTATTCTGA**
ATTAGAAGTTTTCCACTTATGACTTAAGAACTTGTGAAGCAG	**AGTTCTTCTGCTTGATGATTTTATCCTTAGCCAAAAGGATTGGTGG**
AAAGTGGCAACACCCCCACCTCCCCCCCCCCCCCCCACCCC	**TTTGAAGACACATCATATCAAAAAAGCTATCGCCTCGACGATGCTC**
CCAAATTGAGAAGTCAATTTTATATAATTTAATCAAATAAAT	**TATTTCTATCCTTGTAGCACACATTTTGGCACTCAAAAAAGTATTT**
AAGTTTATGGTTAAGAGTTTTTTACTCTCTTTATTTTTCTTT	**TTAGATGTTTGTTTTGCTTCTTTGAAGTAGTTTCTCTTTGCAAAAT**
TTCTTTTTGAGACATACTGAAAAAAGTTGTAATTATTAATGA	**TCCTCTTTTTTTAGAGTGATTTGGA**AATCTCTCATCCCCGCATAAT
TAGTTCTGTGATTCCTCCATGAATCACATCTGCTTGATTATT	TGATTTGTGTCTAGTAAACCACGTAAATAAGGGACCATGATATCCA
GCAACACGAGATCAGACGATAGTCTTTCATC	AGATTAGTTATGGAACAATGTCGTAT

바이오인포매틱스 길라잡이 1

우선 바이오인포매틱스를 공부하고자 하는 사람에게 도움이 될 수 있는 책을 소개하고자 한다(필자가 직접 공부하면서 도움을 받은 책인 만큼 여러분에게도 도움이 됐으면 하는 바람이다). 각 책의 소개에는 레벨을 표시해 뒀는데, 이는 우리의 개인적인 의견이므로 책을 읽는 사람에 따라서 다를 수 있다. 레벨은 모두 5단계로 돼있으며, 1 · 2는 학부 과정 학생이 볼 수 있는 책이며, 3 · 4는 현업에서 일하는 사람이 보기에 좋은 책이다. 4 · 5는 가장 어려운 책으로 생물학뿐만 아니라 컴퓨터 관련의 알고리즘 · 자료구조 · 파일 시스템 · 운영체제와 같은 많은 지식이 밑바탕 되어야 이해할 수 있는 책이다.

B1 GENOMES(T.A BROWN, WILEY BIOS, Level 3)

바이오인포매틱스를 처음 공부할 때 꼭 봐야한다는 소개를 보고 읽게 된 책이다. 이 책에는 분자생물학 전공 학생을 대상으로 만들어진 책이다. 이 책에는 유전체에 대한 소개뿐만이 아니라, 유전체 분석의 여러 실험 방법에 대한 자세한 소개가 나온다. 전산 전공 학생에게는 다소 어려운 책이지만, 유전체의 여러 실험에 대해 많이 배울 수 있다. 생물학에 대한 정보가 없는 필자와 같은 경우라면, 혼자서 보기에는 다소 어려움이 있을 것이다. 필자의 경우에는 바이오인포매틱스에 관심이 있는 생물학을 전공한 사람과 함께 세미나를 통해 공부했다.

B2 유전 3, 기능 유전체학(한국유전학회 지음, 월드 사이언스, Level 2)

유전체학의 기술적인 혁신을 소개하고 있는데, 바이오인포매틱스 분야의 발전속도를 생각해 볼 때 현재까지의 모든 기술에 대해 소개하지는 못했다. 하지만 바이오인포매틱스에서 필수적인 접근 방식에 대해서 알 수 있다.

B3 분자생물학(심웅섭 · 안정선 · 안태인 · 이동희 · 이주현 · 정혜문 · 허윤강 지음, 월드사이언스, Level 1)

분자생물학의 기본적인 개념이 쉽게 설명돼 있는 책이다. 따라서 처음 바이오인포매틱스를 접했을 때, 생물학적 지식이 없다면 많이 당황하게 되는데 이때 기본적인 지식을 습득하기에 좋다.

B4 생명합성에의 길(나가쿠라 이사오 지음, 박택규 옮김, BLUE BACKS, Level 1)

유전자, 단백질, 바이러스 등과 같은 생명합성에 대해 나와 있다. 이 책을 읽으면 생명이 어떻게 합성되는가에 대한 기본적인 개념을 알 수 있다. 이 책은 아주 작고 얇기 때문에 부담이 없다. 필자의 경우에는 학교에서 집에 갈 때 버스에서 잠깐씩 읽었다.

B5 유전자에 관한 50가지 기초지식(가와카미 마사야 지음, 박경숙 옮김, BLUE BACKS, Level 1)

유전자가 어떻게 복제되는가와 같은 정말 기초적인 내용 50가지를 다룬 책이다. 이 책은 '생명합성에의 길'과 같은 시리즈 중 하나다. 이 시리즈에는 '유전자가 말하는 생명의 모습', 'RNA 이야기', '단백질이란 무엇인가'와 같은 책이 있다. 이들 모두 아주 기본적인 것을 소개하고 있으며, 필자가 처음 바이오인포매틱스에 대해 공부할 때 기본적인 개념을 쉽게 정리할 수 있게 도와준 책이다. 처음 바이오인포매틱스를 접할 때 쉽게 읽을 수 있는 책이라고 생각한다. 주의할 점은 BLUE BACKS에서 나온 이 시리즈는 일반인이 접근하기는 쉽지만, 바이오인포매틱스 분야의 발전 속도가 빨라서 현재의 기술에 대한 내용은 부족한 편이다.

B6 Biological sequence analysis : Probabilistic models pf proteins and nucleic acids(Richard Durbin 외 3명 지음, Cambridge University Press, Level 5)

통계학적 모델인 Hidden Markov Model을 이용해 서열을 분석하는 여러 방법에 대해서 다룬다. 이 책을 보기 위해서는 서열 분석 방법인 서열정렬(Sequence Alignment)과 통계적인 지식, 그리고 기본적인 생물학적 서열에 관련된 지식을 미리 갖춰야한다. 필자의 경우에는 쉽지 않은 책이었다.

B7 Bioinformatics : Sequence and Genome Analysis(David W.Mount 지음, COLD SPRING HARBOR LABORATORY PRESS, Level 3)

가장 최근에 나온 책으로 서열과 구조 분석을 위한 방법을 이해할 수 있다. 여러 바이오인포매틱스에 관련된 데이터베이스와 그에 대한 플랫 파일의 설명, 서열 정렬과 같은 분석에 대한 설명이 나와있다. 따라서 실제 산업현장에서 일하는 사람에게 도움될 것 같다.

B8 Algorithms on Strings, Trees and Sequence: Computer Science And Computational Biology(Adn Gusfield 지음, CAMBRIDGE UNIVERSITY PRESS, Level4, 5)

서열 분석에 꼭 필요한 서열 분석의 여러 알고리즘과 자료구조에 대해 설명하고 있다. 서열 분석을 위한 여러 알고리즘과 데이터베이스 검색 등과 같은 것에 대해 공부하려고 할 때 많은 도움을 받을 수 있다. 이 책을 보기 위해서는 기본적으로 전산학 지식인 자료구조와 알고리즘에 매우 밝아야한다. 컴퓨터 전공자가 접근하기에 아주 좋은 책인 것 같다.

B9 BIOINFORMATICS : A PRACTICAL GUIDE TO THE ANALYSIS OF GENES AND PROTEINS(Andreas D. Baxevanis, B.F.Francis Ouellette 지음, WILEY-INTER SCIENCE, Level 2, 3)

인터넷을 통해 여러 정보를 제공해주는 생물학적 데이터베이스(GenBank, PDB...)의 데이터 구조와 그것을 사용하는 방법 등을 소개하고 있다. 이 외에도 DNA나 단백질의 서열을 예측하는 방법 등을 소개하고 있어 바이오인포매틱스의 여러 소프트웨어를 이해하는데 좋다. 이 책을 보기 위해서는 기본적으로 유닉스 · 리눅스 · 데이터베이스 · 파일 시스템과 같은 컴퓨터 시스템에 대한 지식이 있어야 한다.

B10 Computational Molecular Biology: An Algorithmic Approach(Pavel A.Pevzner 지음, Mit Press, Level 5)

시퀀싱, 서열 정렬, DNA 어레이 등과 같은 비교 생물학을 생물학적 특성과 함께 전산학적 관점(알고리즘)에서 기술한 책이다. 따라서 이 책은 생물학자에게는 전산학적인 전문 기술을 제공해줄 수 있고, 전산학자에게는 바이오인포매틱스에서 필요한 알고리즘뿐만 아니라 그 분야에서 어떠한 일을 할 수 있는가에 대한 안목을 심어줄 수 있다. 이 책을 보기 위해서는 기본적으로 전산학적 지식이 필요하기 때문에, 비전공자가 보기에는 힘든 책이다. 이 책은 대학원 수업 시간에 2/3정도를 배웠다.

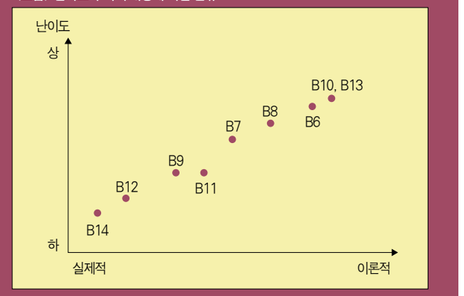
B11 Introduction to Computational Molecular Biology(SETU BAL, MEIDANIS 지음, PWS, Level 3)

생물학에 관련된 내용이 40% 정도이고, 나머지 60% 정도가 전산학에 관련된 이야기가 나온다. 이 책은 학부 3 · 4학년 정도의 학생이 볼 수 있는 어렵지 않은 책으로 처음 바이오인포매틱스를 접할 때 보기에 좋다.

B12 Bioinformatics Methods and Protocols(Stephen Misener, Stephen A.Krawetz 지음, Humana Press, Level 2)

여러 bio 패키지들을 자세하게 소개한 실무적인 책이다. 바이오인포매틱스의 본질적인 문제보다는 실제 여러 툴을 사용하는 사람이나 툴을 개발하려는 사람에게 필요한 책이다. 여러 툴에 대한 소개가 나오기 때문에 어렵지는 않다.

〈그림〉 난이도와 책의 내용에 따른 분류



B13 Bioinformatics: The Machine Learning Approach(Pierre Baldi, Soren Brunak 지음, Level 4, 5)

머신 러닝 알고리즘과 이를 이용한 유전자 사냥과 같은 알고리즘에 대해 다룬다. 이 책은 computational molecular biology 분야에서 오랫동안 연구되어 온 'classic' 문제에 대해 알 수 있는 유용한 책이다. 이 책을 이해하기 위해서는 통계학의 Hidden Markov Model과 컴퓨터 분야의 자료구조와 알고리즘과 같은 기본적인 내용을 바탕으로 neural network과 machine learning에 대해 알고 있어야 이해하기 쉽다.

B14 Develoing Bioinformatics Computer Skills(Cyntbia Gibas, Per Jambeck 지음, 오라일리, Level 1, 2)

바이오인포매틱스를 처음 접하는 사람에게 기본적으로 권하는 책이다. 이 책에는 컴퓨터를 전공하는 사람에게 바이오인포매틱스에서 컴퓨터로 할 수 있는 기본적인 기술에 대해 나와있다. 웹에서 볼 수 있는 생물학적 연구, 서열 분석, 데이터베이스와 같은 여러 툴에 대해서도 소개하고 있다. 어렵지 않은 책이므로 바이오인포매틱스를 접하는 사람이라면 한번쯤 읽어 볼 만한 책이다.

다음의 〈**그림**〉은 앞에서 소개한 여러 책 중에서 전산학에 관련된 책을 난이도와 책의 내용에 따라 분류해 본 것이다. 처음 바이오인포매틱스를 접하는 사람이라면 쉬우면서도 바이오인포매틱스에서 실제 사용되고 있는 여러 툴에 관한 책을 살펴봄으로써 바이오인포매틱스에 대해 이해를 넓혀 가는 것이 좋을 것 같다.

다(이미 시퀀스 1 서열의 붉은 부분이 어떠한 기능을 하는지 알고 있으므로). 그러면, 실제 서열을 보면서 이러한 부분을 찾아보자. 다음에 있는 두 서열을 쥐와 사람의 서열의 일부분이라고 가정해보자. 이때 두 서열의 같은 부분을 찾아보려고 한다. 직접 서열을 보면서 눈으로 찾아보자.

예로 든 쥐의 서열과 사람의 서열은 같은 부분이 붉은 색으로 미리 표시돼 있다. 이런 표시가 없었다면 찾을 수 없었을 것이다. 우리는 exact matching 알고리즘을 사용해 두 서열 사이에 존재하는 같은 패턴의 서열을 찾아낼 수 있다.

최장 공통 스트링

최장 공통 스트링(Longest Common Subsequence)이라는 것은 두 서열 A=a[1]a[2]...a[n], B=b[1]b[2]...b[m]이 주어졌을 때 두 서열 사이의 공통되는 서열 사이에 가장 긴 것을 말한다. L(A[1,i],B[1,j])을 A서열의 A[1]부터 A[i]까지와 B[1]부터 B[j]까지의 서열의 LCS의 길이라고 정의하면, LCS의 길이를 구하는 식은 다음과 같다.

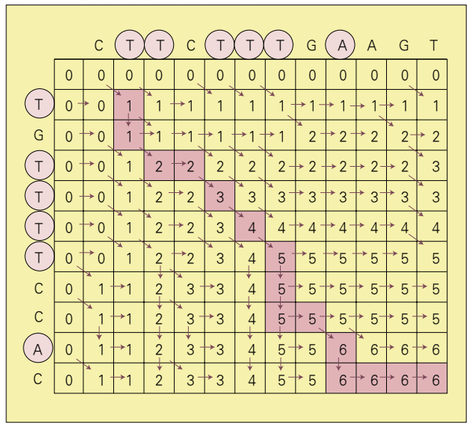
```
i)  L(A[1,i],B[1,j]) = L(A[1,i-1],B[1,j-1]) + 1              , if A[i]==B[j]
ii) L(A[1,i],B[1,j]) = max(L(A[1,i-1],B[1,j]),L(A[1,i],B[1,j-1])) , if A[i]!=B[j]
```

앞의 식을 보면 알겠지만, 우리는 두 서열의 LCS를 구할 때

〈리스트 1〉 LCS 프로그램의 수도 코드

```
/* 세 가지 방향 정의. 현재 행렬에서의 위치 (i, j)
 ↖ : (i-1, j-1)의 값을 선택한 경우
 ↑ : (i  , j-1)의 값을 선택한 경우
 ← : (i-1, j  )의 값을 선택한 경우 */
LCS(S1,S2)                    // S1, S2는 입력으로 들어오는 두 서열
for i←1 to n
   si,0 ←0                    // si,j는 LCS 길이가 저장되는 행렬
for i←1 to n                  // 두 서열 S1, S2를 처음부터 살펴본다.
   for j←1 to n
      if S1i = S2j            // 현재 두 서열의 각 위치에 있는 문자가 같은 경우
         si,j ←si-1,j-1 + 1   // LCS 길이를 하나 증가한다.
         bi,j ←↖              // (i,j)의 결과는 (i-1, j-1)에서부터 왔다는 것을 표시한다.
      else if si-1,j ≥si,j-1  // L(S1[1,i-1],S2[1,j])이 L(S1[1,i],S2[1,j-1])보다 큰 경우
         si,j ← si-1,j        // L(A[1,i-1],B[1,j]) 값을
         bi,j ←↑              // (i,j)의 결과는 (i-1, j)에서부터 왔다는 것을 표시한다.
      else
         si,j ←si,j-1         // L(S1[1,i],S2[1,j-1])이 L(S1[1,i-1],S2[1,j])보다 큰 경우
         bi,j ←←              // (i,j)의 결과는 (i, j-1)에서부터 왔다는 것을 표시한다.
return s and b
```

〈그림 8〉 LCS를 구하기 위한 다이내믹 프로그램



다이내믹 프로그래밍을 이용한다. 행렬 M을 중간 값이 저장되는 행렬이라고 하자. LCS의 길이를 구할 때에는 두 서열을 처음부터 끝까지 보면서 행렬 M의 값을 구해 다 넣을 때까지만 해보면 된다. 하지만 그 서열을 구할 때에는 행렬 M을 구할 때 그 값이 어느 곳에서부터 온 값인지를 알 수 있도록 왔던 방향을 저장하고 행렬 M을 다 완성하면, 행렬(n,m)에서부터 (1,1)까지 그 방향을 따라 가보면 알 수 있다. **〈리스트 1〉**은 LCS의 길이와 서열을 모두 알기 위한 수도 코드다.

그러면, SeqA=TGTTTTCCAC, SeqB=CTTCTTTGAAGT일 때 LCS를 구해보자. 다음과 같이 행렬 M을 구할 수 있다.

따라서, LCS의 길이는 5이고, LCS=TTTTA이다.

근사 스트링 매칭

근사 스트링 매칭(Appprixiimate String matching)은 k개까지의 mismatch를 허용하는 것이다. 두 서열 Seq1=AAGTTGCAAT, Seq2=AAGATCCATT가 주어지고 세 개까지 mismatch를 허용한다고 하자. 이 두 스트링을 매칭시켜 보면 다음과 같이 나온다.

```
Seq1    AAGTTGCAAT
Seq2    AAGATCCATT
```

Exact matching을 사용한다면 Seq1과 Seq2는 같지 않다고 나올 것이다. 하지만 세 개까지 mismatch를 허용하기 때문에 이 두 스트링 Seq1과 Seq2는 매칭이 된다고 볼 수 있다. 이러한 근사 스트링 매칭을 서열 분석에서 사용하는 이유는 서열이 서로 완전히 같지 않더라도 그 기능이나 구조가 같을 수 있기 때문이다. 즉, 어떤 서열을 다른 서열과 비교할 때 그와 유사한 서열도 찾아낼 수 있어야하기 때문이다.

갭이 있는 스트링 매칭

생물학적 두 서열에서 스트링 매칭을 할 때, 각 서열들이 진화 과정에서 삽입(insert, 서열에 염기가 서열 안으로 들어오는 것)이나 삭제(delete, 있던 염기가 삭제될 수 있다), 치환(replacement, 염기들의 위치가 바뀌는 것)이 일어나기 때문에 완전히 같은 서열을 매칭하는 것은 사용하기에 좋지 않다. 이러한 단점을 해결하기 위해 갭이라는 것이 도입됐다. 갭은 비교하는 서열 사이에서 서로 다른 것이 나왔을 때 그 사이를 메워주는 것을 말한다. 다음의 두 서열 A와 B를 살펴보자.

```
Sequence A   ATACATCATAACACTACTTCCTACCCATAAGCTC
Sequence B   ATACATCATAACACTACTCCTCCCATAAGCTC
```

이 두 서열을 첫 염기 서열부터 차례대로 맞춰 보면 다음과 같이 된다.

```
Sequence A   ATACATCATAACACTACTTCCTACCCATAAGCTC
Sequence B   ATACATCATAACACTACTCCTCCCATAAGCTC
```

하지만 이 두 서열에 갭을 허용해 매칭하면 다음과 같이 된다.

```
Sequence A   ATACATCATAACACTACTTCCTACCCATAAGCTC
Sequence B   ATACATCATAACACTACT-CCT-CCCATAAGCTC
```

또한 두 서열 전체가 매칭되는 것을 볼 수 있다. 따라서 갭을 사용하지 않고 서열을 매칭하는 것보다 갭을 두는 것이 두 서열에서 더 의미 있는 매칭을 찾을 수 있다. 이처럼 정렬에서 갭이 발생하면 갭 페널티(gap penalty)를 주게 된다. 갭 페널티는 삽입 혹은 삭제에 의해 생기는 갭에 얼마의 감점을 줄 것인가를 정하는 것이다.

서열 정렬

서열 정렬(Sequence alignment)이란 단백질 서열이나 핵산 서열 사이의 상관 관계를 나타내는 것이다. 따라서 관심 대상이 하나의 서열과 유사성이 높은 서열을 알아내 그 서열의 기능을 유추하거나, 관련되는 서열 사이의 진화적 연관성 같은 것을 예측하기 위해 사용된다. 물론 유사성이 없음에도 불구하고 그 구조나 기능이 유사한 경우도 종종 있다. 서열 정렬은 1970년 Needleman과 Wunsch의 다이내믹 프로그래밍 기법에 의한 pairwise alignment가 발표된 이후, Smith와 Waterman 등이 일반적인 길이의 갭에 대해 이 알고리즘을 확장함으로써, 두 서열 사이의 정렬을 위한 실용 가능한 프로그램이 구현되기 시작했다. 이들 연구에 기반해 전산학의 그래프 알고리즘 이론이나, 패턴 매칭 알고리즘이 주로 생물학 패턴의 검색에 많이 응용돼 여러 응용 알고리즘이 개발됐으며, 데이터베이스의 개발과 함께 대량의 데이터를 대상으로 빠른 실행 시간이 요구돼 여러 heuristic 알고리즘이 개발됐다. 이러한 서열 정렬은 정렬하는 서열의 개수에 따라, pairwise alignment와 multiple alignment로 나눠지며, 상동성의 종류에 따라 global alignment와 local alignment로 나뉘어진다. pairwise alignment는 두 개의 서열에 대해 정렬하는 것을 말하며, multiple alignment는 세 개 이상의 서열에 대한 정렬을 말한다. Global alignment는 비교하는 서열 전체를 모두 비교해 유사도를 측정하는 것을 말하며, local alignment는 유사성이 높은 각 서열의 부분만을 생각하는 것을 말한다.

주어진 서열의 유사한 정도를 표현하기 위해서 두 서열 사이의 유사성을 숫자로 표현한다. 즉 높은 숫자는 높은 유사성을 갖게 되는 것이다. 가장 간단한 방법으로는 두 서열이 길이가 같을 때, 두 서열을 배열한 후에 단순히 서로 같은 염기를 세어서 그것을 유사도로 삼을 수 있다. 이러한 방법은 간단하면서도 유용해 보이지만, 거의 모든 서열이 정렬할 때 그 길이가 서로 다르고(커다란 서열을 랜덤으로 잘라서 하기 때문에), 각 서열이 진화 과정에서 삽입(insert)이나 삭제(delete) 또한 치환

(replacement)이 일어나기 때문에 사용하기에 좋지 않다. 이러한 단점을 해결하기 위해 갭이라는 것이 도입됐다. 서열을 정렬할 때, 갭이 발생하면 gap penalty를 주게 된다. Gap penalty는 삽입 혹은 삭제에 의해 생기는 갭에 얼마의 감점을 줄 것인가를 정하는 것이다. 현재의 통계적 계산으로는 gap penalty를 얼마나 주어야하는가에 대한 정확한 해답은 없지만, 여러 실험적 사실에 의해 그 값을 지정해 사용한다. 실제 염기나 단백질 서열이 각각 치환되는 확률은 각각의 서열의 진화적 · 물리적 · 화학적 성질에 따라 많은 차이를 나타낸다. 그래서 실제 염기 서열의 비교에는 각각의 서열의 치환 확률을 행렬로 만들어 사용한다. 이것을 scoring matrix라고 한다. 따라서 어떠한 scoring matrix를 선택하느냐에 따라 분석 결과에 많은 차이가 나타날 수 있다. 이러한 scoring matrix는 진화이론에도 적용된다.

전체 서열 정렬

두 시퀀스를 비교하는 경우, 서로 동일한 종류의 핵산이나 단백질 시퀀스에 이어서 비교하는 시퀀스를 전체적으로 비교해 최대의 상동성이 나타나도록 alignment하는 경우, 그러한 상동성을 global homology라고 하며, 이러한 alignment를 전체 서열 정렬(Global Alignment)이라고 한다.

길이가 1보다 큰 임의의 두 prefixes(0:s:i와 0:t:j) i와 j가 있고, i와 j보다 작은 prefixes의 적절한 배열은 이미 알고 있다고 가정한다. 이 경우 i와 j의 최적화 배열은 다음에 나오는 세 가지의 경우 중 하나다.

```
치환(replacement) 혹은 일치(match), (si = tj)
삭제(deletion), (si , -)
삽입(insertion), (-, tj)
```

앞의 세 경우의 다이내믹 프로그래밍을 이용해 서열을 정렬할 수 있다.

```
GA(0:s:i, 0:t:j) = max {
   GA(0:s:i, 0:t:j) + w(si, tj),
   GA(0:s:i-1, 0:t:j) + w(si, -),
   GA(0:s:i, 0:t:j-1) + w(-, tj)
}
```

앞의 식에서 GA(0:s:i, 0:t:j)는 두 서열 집합 s와 t에서 i와 j번째 서열까지의 전체 서열 정렬을 했을 경우의 점수 중 가장 높은 것을 의미한다. 이와 같은 식을 이용해 각각 m과 n개의 염기 서열을 가진 비용의 최소 값은 (e+1)*(n+1)개의 행렬(matrix)로 나타낼 수 있다. 예를 들어 두 서열 s(ATTGCA)와 t(TCGCCT)가 있고, match(A,A)=2, match(G,G)=2, match(T,T)=1, match(C,C)=1이며 gap insertion/deletion은 -1의 penalty가 있으며 unmatched pair에 대해서는 0의 penalty가 있다고 하자. 여기서 s와 t를 각각 행렬의 축으로 하고 앞의 식을 적용해 각각의 행렬의 항을 채워나가면 다음과 같은 행렬을 만들 수 있다. 다음의 **〈그림 9〉**는 global alignment하는 모습이다. 대각선은 일치 혹은 치환을, 수평선은 삽입을, 수직선을 삭제를 나타냄을 알 수 있다. **〈그림 9〉**를 살펴보면 두 서열 s와 t의 적절한 배열을 위한 가중치 함수의 값은 2라는 것을 알 수 있다.

대각선에 의해 표시된 행로에 의해 s와 t를 배열하면 다음과 같다.

```
s   ATTGC-A
t   TC-GCCA
```

이렇게 두 서열의 정렬을 보여주기 위해서는 score matrix를 만들 때 각각의 score가 어느 곳에서 왔었는지를 알고 있어야 한다. 따라서 score matrix 외에 같은 크기의 방향을 나타내는 matrix도 하나 더 필요하게 된다. 각 위치의 score를 결정할 때 그 값이 어느 곳(방향으로, 수직방향, 왼쪽 대각선방향)에서 왔는지를 방향을 나타내는 matrix에 저장하고 모든 위치의 score를 결정하면 방향 matrix의 (n,m)에서부터 (1,1)까지 거꾸로 따라가면, 두 서열의 정렬된 모습을 얻을 수 있다. 앞의 다이내믹 프로그램 알고리즘은 서열 길이를 L이라고 할 때, 각각 L2에 비례하는 실행 시간과 메모리를 요구한다. 데이터베이스를 대상으로 상동성을 검색하는 경우, N*L2(N : 데이터베이스에 있는 서열의 수)에 비례하는 실행 시간이 필요하므로 용량이 큰 서버에서도 실행에 대한 부담이 매우 크다고 할 수 있다. 부분적인 유사도 검색에 대한 필요성과 이와 같은 실행 시간의 문제를 해결하기 위한 해결책으로 heuristic을 사용한 빠른 알고리즘이 개발돼 BLAST, FASTA 등의 프로그램으로 사용되고 있다. 메모리 사용에 대한 문제의 해결책으로는 linear space를 사용하는 정렬 알고리즘이 개발됐다.

〈그림 9〉 global alignment

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

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

부분 정렬

두 시퀀스의 어떤 부분 시퀀스가 높은 상동성을 갖는가를 나타내기 위해 alignment하는 경우, 그러한 상동성을 local homology라고 하고, 그 alignment를 부분 정렬(Local Alignment)이라고 한다. 높은 global homology를 나타내는 같은 기능을 가진 단백질이나 핵산의 경우, local homology는 global homology를 보이는 부분과 유사한 부분을 나타내게 된다. 즉, global alignment와 local alignment가 유사한 결과를 보여주게 된다. 한편 기능적으로 일부분만이 관련되어 있어, 진화적으로도 일부분만이 상동성을 보여줄 수 있을 경우는 부분 정렬을 실행하는 것이 효과적일 것이다. 또한 유사한 기능의 시퀀스나 global homology가 낮아 쉽게 상동성을 나타내지 못하지만 일부분에서는 높은 상동성을 보이는, 즉 일부분에서 높은 local homology가 있는 경우도 부분 정렬을 실행하는 것이 효과적이다. 따라서, 많은 경우 부분 정렬이 상동성 서열을 검색하기 위한 용도로 사용된다. 부분 정렬의 다이내믹 프로그래밍 식은 전체 서열 정렬 식에서 max 값을 구할 때 '0' 값이 하나 추가된다. 이것은 이전까지 정렬했던 결과를 고려하지 않고 현재부터 새롭게 정렬할 수 있다는 뜻이다.

〈표 2〉 입력 서열

서열	결과
Seq1	MAVDPPKADPKGSSGGFHRKSADDPKNGGPALGSREDQSAKAGCCDPPK
Seq2	DPPKADPKGGGFHRNARKKDNGGPALGSREDQSAKAGGCCQWER
Seq3	MAVDPPKADPKGSSGGFHRKSADDPKNGGPALGSREDQSAKAGCCQWER

〈표 3〉 결과 서열

서열	결과
Seq1	MAVDPPKADPKGSSGGFHRKSADDPKNGGPALGSREDQSAKAG-CCDPPK
Seq2	---DPPKADPKG--GGFHRNARKK-DNGGPALGSREDQSAKAGGCCQWER
Seq3	MAVDPPKADPKGSSGGFHRKSADDPKNGGPALGSREDQSAKAG-CCQWER

```
LA(0:s:i, 0:t:j) = max {
   0,
   LA(0:s:i, 0:t:j) + w(si, tj),
   LA(0:s:i-1, 0:t:j) + w(si, -),
   LA(0:s:i, 0:t:j-1) + w(-, tj)
}
```

Multiple Alignment

앞의 여러 방법은 두 서열이 주어졌을 때 그 유사도를 구하는 방법이었다. 이러한 방법을 pairwise alignment라고 한다. 이와는 달리 세 개 이상의 여러 서열이 주어졌을 때, 그 서열을 비교하는 것이 multiple alignment다. 단백질 서열의 비교는 단백질 사이의 구조나 기능과 같은 생물학적 유사성을 찾는 것이 목적이다. 생물학적으로 유사한 단백질이 서열의 강한 유사성을 보이지 않을 수도 있고, 서열이 매우 다를지라도 그 구조나 기능적으로 유사할 수도 있다. 만약 서열의 유사성이 적을 경우, pairwise alignment는 서열의 유사한 부분만을 찾아내기 때문에 생물학적으로 연관성이 있는 서열을 찾아낼 수 없을 것이다. 따라서 종종 많은 서열을 동시에 비교해 pairwise alignment에서 찾을 수 없었던 유사성을 찾기도 한다. 이는 패밀리(family) 분석, 계통 관계(phylogeny) 분석, 도메인(domain) 분석 등의 기능 분석(functional analysis) 연구를 위해 매우 다양하게 사용된다.

다이내믹 프로그래밍으로 길이가 n인 k개의 서열을 정렬하는 문제를 해결할 수 있다. 하지만 이것은 O((2n)k)만큼의 실행 시간이 들것이다. 따라서 길이가 긴 k개의 서열을 exact

multiple alignment하는 것은 실행할 수 없고, suboptimal한 multiple alignment를 위해 많은 heuristic한 방법이 나오고 있다. 다음은 세 서열 Seq1, Seq2, Seq3에 대해 multiple alignment를 한 결과다.

바이오인포매틱스에 도전하라

최근 들어 바이오인포매틱스를 연구하는 컴퓨터 전공자가 늘어나고 있다. 우리가 처음 바이오인포매틱스를 접하면서 가장 어려웠던 점은 생물 전공자들의 말을 이해할 수 없었다는 것이다. 아주 간단한 예로 생물분야에 '모티프'라는 것이 있다. 이것은 서열에서는 '보존되는 서열의 부분(conserved "blocks" of sequences)'이라는 의미고, 구조에서는 '몇 개의 2차 구조가 특정한 모양으로 배열되어 이룬 구조(combination of a few secondary structure with a specific geometric arrange ment)'로 여러 단백질에서 공통적으로 발견되고 기능 혹은 구조적 역할을 수행하는 것을 말한다. 하지만 처음 이 말을 들었을 때 나는 X윈도우를 떠올렸다. 앞서도 말했지만 바이오인포매틱스를 알기 위해서는 생물학적인 지식이 필요하다. 아마도 이것이 바이오인포매틱스를 돌파하는 첫 번째 관문인데 너무 짧은 시간에 많은 것을 배우려고 하기보다는 짬짬이 시간 나는 대로 쉬운 책부터 읽고, 생물학 관련 연구자와 자주 대화를 하면서 귀를 열어 가는 지혜가 필요할 것이다. 또한 생명과학 연구자는 IT 사람에게 뭔가 정량화한 표현과 자료를 제시하는 일에 좀 더 익숙해져 할 것이다. 지금까지 살펴봤듯이 바이오인포매틱스는 컴퓨터 과학자 없이는 발전할 수 없으며, 이 학문의 발전은 상당 기간동안 지속될 것이다. 따라서 전산 전공자가 연구하기에 아주 좋은 학문이라고 생각한다. 우리는 이 글을 읽는 많은 재능 있는 전산 전공자가 바이오인포매틱스라는 새로운 학문에 관심을 갖고 도전해보기를 바란다.

정리 : 조규형 jokyu@sbmedia.co.kr

참고자료

1. Computational Molecular Biology; An Algorithmic Approach ,Pavel A.Pevzner 지음, Mit Press
2. GENOME, T.A.BROWN, JOHNWILEY&SONS(ASIA) PTE LTD
3. Computing in Science and Engineering , May/June 1999 (Vol. 1, No. 3), pp. 33-43
4. Algorithms on Strings, Trees and Sequence; Computer Science And Computational Biology,
5. Adn Gusfield 지음, CAMBRIDGE UNIVERSITY PRESS

바이오인포매틱스 길라잡이 2

웹을 통해서도 바이오인포매틱스에 관련된 많은 내용을 볼 수 있다. 다음은 바이오인포매틱스에 관련된 몇 가지 웹 사이트를 소개하겠다.

◆ 국립 보건원(http://www.nih.go.kr/)

각종 질병의 원인 규명을 위한 연구와 보건, 복지 분야 종사자의 교육훈련을 실시하는 국립기관으로서 전염병과 비전염병 질환의 효과적인 예방, 진단, 치료법에 대한 연구와 보건, 복지분야 종사자의 전문적 지식과 소양을 함양하기 위한 교육을 담당하고 있다

◆ 생명공학연구소(http://www.kribb.re.kr/)

유전자 은행, Bio-Information Center, 유전체 사업단, 생물다양성 정보네트워크 BIKNet 등의 내용을 볼 수 있다.

◆ 한국과학기술정보연구원(http://www.kisti.re.kr/)

지식 정보시대 연구개발 활동에 반드시 필요한 첨단정보 데이터베이스, 지식정보에 부가가치를 더하는 기술정보 분석 서비스, 만들어진 정보의 신속한 유통을 위한 초고속 연구망, 그리고 정보와 과학기술의 한계를 극복하는 슈퍼컴퓨팅 파워를 제공하는 국립연구원이다.

◆ 포항공과대학 생물학정보연구센터(http://bric.postech.ac.kr/)

생물학 연구지원을 위한 온라인 저널, 문헌정보, 관련 사이트, 연구자원 데이터베이스와 각종 동향정보를 제공해준다. 또한 바이오인포매틱스에 관련된 여러 응용 프로그램을 분류해 소개하고 있다.

◆ 서울대 장병탁 교수 홈페이지(http://bi.snu.ac.kr/~btzhang/)

진화 계산의 Bayesian 이론, probabilistic neural networks, bioinformatics와 biocomputing의 응용 프로그램에 대해 연구하는 곳이다.

◆ bioinformatics 온라인 강의(http://s-star.org/lectures)

Bioinformatics 관련 온라인 강의가 있다. 강의 내용은 아주 수준 있고 의미 있는 내용이다. 하지만, 녹음 상태가 그리 좋지 않아 듣기가 쉽지는 않다.