그래프 모형을 활용한 생물 네트워크 분석법 개요

A Survey on Biological Network Analysis with Graph Model

조환규,
hgcho@pusan.ac.kr

1 그래프 이론과 그 응용

그래프 이론(Graph Theory)는 수학의 한 갈래인 조합론(Combinatorial Theory)의 일부로서 매우 오래전 부터 연구되어 왔다. 그래프 이론이 본격적으로 현실문제에 활용되기 시작한 것은 2차대전으로 탄생한 OR(Operations Research)의 영향일 것이다. 예를 들어 지금도 산업공학에서 많이 활용되고 있는 max-flow를 이용한 최적화 계산모형은 그래프의 현실적 활용의 예라고 할 것이다. 이후 계산능력(computation power)의 빠른 발전과 더불어 그래프 이론은 전산학의 전 분야에서 다양하게 활용되기 시작했으며, 지금은 전산학 뿐만 하나라 현대수학에서 중요한 이론적 도구가 되고 있다.

전통적인 그래프 이론과 현대적 그래프 이론[1]을 나누는 기준은 따로 있지는 않지만 전통적인 그래프 이론은 주로 조합론적인 문제, 예를 들면 특정한 성질을 가지는 그래프 구조는 어떤 것인가에 집중하고 있다. 예를 들면 오일러 그래프나 해밀토니안(Hamiltonian) 그래프가 가지는 특성을 규명하는 연구에 여기에 해당한다. 이런 조합론적인 그래프 문제는 대부분을 표준적인 그래프 교재(text)에서 다루고 있는 주제들이다. 그 주제를 나열하면 Degree Sequence, Vertex Coloring, Edge Coloring, Planarity[2], Vertex

[1] 수리과학의 특성상 1980년대 이후에 새롭게 소개된 대부분의 그래프 관련 연구주제는 현대 그래프 이론이라고 볼 수 있다.

[2] 어떤 그래프를 에지의 교차없이 2차원 평면에 그릴 수 있는지의 여부를 확인하는 문제

- edge connectivity 등이다. 그리고 이들에 관련된 알고리즘적 이슈들, 예를 들어 특정한 그래프의 성질을 확인하는데 걸리는 시간복잡도는 무엇인가, 그것이 최적의 복작도 (Optimal Complexity) 인가를 따지는 일들이 여기에 속한다. 일반적으로 볼 때 전통적인 그래프 이론은 특정 그래프 클래스 (graph class) 의 성격을 규명 (Characterization) 하는 작업이라고해도 무방할 것이다. 예를 들어 모든 vertex의 degree가 5이상이면 가장 짧은 Cycle 의 길이는 k 이다 - 와 같은 명제는 특성화의 전형적인 표현이다. 그러나 이런 단순한 성질 규명은 현실에서 별로 유용성이 없다. 근대 그래프 이론은 컴퓨터의 등장으로 이전과 다르게 엄청나게 복잡한 계산문제를 실제 풀 수 있었기 때문에 보다 새로운 주제가 그래프 연구에 나타나기 시작하였다.

1.1 그래프 이론의 동향

최근의 그래프 이론은 앞서 설명한 전통적인 주제 이외에 계산적 측면과 확륙적 모형을 중심으로 그 영역을 확장해왔다. 특히 다른 분야와 마찬가지로 컴퓨터를 이용한 계산과학의 발전으로 이전에는 단순히 모형연구에만 그친 주제들이 현실에 직접 응용되기 시작한다. 그 중 일반적인 그래프 이론 강의[3]에서 다루지 않는 몇 가지 새로운 연구 주제에 대하여 간략히 설명한다.

1.1.1 무작위 그래프 이론 (Random Graph Theory)

전통적인 그래프 이론에서는 정점[4](vertex) 와 에지 (edge) 가 이미 결정적 (deterministic) 으로 주어지고 그 안에서 어떤 특성을 규명하는 작업이 주를 이루었다. 그런데 무작위 (랜덤) 그래프 이론에서 vertex와 edge는 확률변수로 주어지기 때문에 그와 관련된 모든 값들, 예를 들어 연결성 (connectedness), 특정 크기의 subgraph의 존재 등도 확률적으로 결정된다. 랜덤 그래프도 에지가 생성되는 방법에 따라서 Erdös-Rényi 모형이나 Watts-Strogatz(WS) 모형 등으로 그 안에서 다시 세분된다. 이 모형에 대해서는 그림-5에서 다시 설명될 예정이다. 램덤 그래프 모형은 실제 그 연결구조를 확정할 수 없는 자연계,

[3] 대부분 대학의 학부에서 그래프 이론만 따로 강의하는 전산학 관련 대학은 없다. 보통은 이산치수학에서 부분적으로 다루거나 대학원에서 전통적인 주제 중심의 그래프 이론이 있다. 조합론을 강의하는 수학과에서 그래프 이론을 다루기도 하지만 계산과 알고리즘 측면에서는 많이 다루지 않고 주로 특성규명 (characterization) 에 집중하고 있다.

[4] 본 보고서에서는 정점과 노드를 같은 의미로 사용한다.

금융계, 시회현상을 모델링할 때나 실험이 불가능한 모형의 특징을 예측할 때 매우 유용하게 쓰일 수 있다. 예를 들어 인터넷에서 악성 바이러스가 퍼지는 속도나 예상되는 피해 규모, 또 그것이 안정화되는데까지 걸리는 시간을 예상하는데 랜덤 그래프는 좋은 모형이 된다. 최근에는 온라인 상에서의 이상조직 (outlier) 을 탐지하여 사기조직이나 테러집단을 추적하는데에도 사용된다.

랜덤 그래프 이론은 가장 중요한 그래프 이론의 한 분야가 되고 있다. 특히 같은 실험에 의해서도 서로 다른 결과가 측정되는 분자생물학 연구에는 랜덤 그래프 모형은 가장 중요한 도구가 될 것이므로 랜덤 그래프 분야에 대한 지속적인 연구가 필요하다.

1.1.2 극단 그래프 이론 (Extremal Graph Theory)

그래프의 특성지표 중 가장 대표적인 것이 에지나 정점의 수이다. 그리고 연결도나 그래프 지름 (graph diameter), 가장 긴 사이클, 짧은 사이클 (girth), 클릭 (clique[5]도 중요한 지표이다. 그리고 이런 지표들간에는 다양한 부등식이 성립한다. 예를 들어 에지의 수가 늘어나면 연결도 (vertex connectivity) 도 증가한다. 그리고 최소 차수 ()degree) 가 올라가도 연결도는 증가한다. 이들간의 관계식 중에서 등호가 성립할 때의 조건에 대해서 집중적으로 탐색하는 것이 이 분야의 주제이다.

edge의 수가 N 개 일 때 vertex connectivity의 최대, 최소값을 N 의 함수로 표현하라는 식의 문제는 극단 그래프 관련 문제의 전형적인 형식이다. 또는 vertex connectivity가 k 일 때 edge수의 최소치는 몇 개인가, 이런 류의 문제에 여기에 포함된다. 좀 쉬운 문제로는 다음과 같은 것을 생각해 볼 수 있다. 어떤 그래프 $G(V)$ 의 연결성 (connectedness) 을 보장하기 위한 최소의 에지 갯수는 몇 개인가[6].

극단 그래프 이론의 원조격인 이론은 램지이론 (Ramsey theory[7]) 이다. 이 분야는 어떤 특성 (property $P(a)$) 을 가지기 위해서 (또는 가지지 않기 위해서 $\backsim P(a)$) 필요한 최소한의 원소의 갯수는 몇개인가- 와 같은 식의 문제를 탐구하는 것이다. 가장 잘 알려진

[5] subgraph중에서 complete graph인 것 중 가장 vertex size가 큰 것

[6] $|G| = n$ 이라고 하자. 그래프가 연결되어있지 않으면서 가장 많은 edge를 가지는 경우는 전체가 K_1 하나와 K_{n-1} 로 분리되어 있는 경우이다. 따라서 $n(n-1)/2+1$ 개의 edge를 가지면 반드시 하나로 연결되어 있을 수 밖에 없다. 따라서 그 하한치는 $n(n-1)/2+1$ 이다

[7] 영국 수학자이며 철학자인 Frank P. Ramsey의 이름를 따서 붙인 이름이다. 램지 이론 문제의 전형은 다음과 같다. "how many elements of some structure must there be to guarantee that a particular property will hold?"

명제는 6명의 사람이 모이면 그 안에서 반드시 3명 이상은 서로 알거나, 3명 이상은 서로 모르는 사이다 - 라는 명제이다. 이것은 "K_3을 clique으로 가지거나 그 complement graph가 K_3를 가지는 그래프의 최소 크기는 $|G(V)|$=6 이다"- 명제로 정의된다. 램지 이론은 현대 수학에 매우 큰 영향력을 미친 조합론의 한 분야이다.

1.1.3 스펙트럴 그래프 이론 (Spectral Graph Theory)

그래프의 정점간 연결관계는 $|G| \times |G|$의 행렬로 표현할 수 있다. 이 행렬의 특성을 바탕으로 그래프의 다양한 성질을 연구하는 분야를 algebraic graph theory라고 하는데, 그 한 갈래가 이 연구이다. 이 연구는 그래프를 다양한 대수적 방법과 군론(Group Theory)를 활용하여 그 대수적 특성을 분석한다. 주된 연구 소재는 입력 graph의 characteristic polynomial, eigenvalue 등이다. 1950년대부터 시작된 스펙트럴 그래프 이론은 당시에는 마땅한 응용분야가 없어 주목받지 못했지만 현대에 와서 Web graph와 같이 이 이론을 적용할 수 있는 다양한 그래프 모형과 엄청난 파워의 계산[8]이 가능해짐에 따라서 매우 활성화되고 있다.

주어진 그래프의 인접행렬(adjacency matrix)이나 라플라시안 행렬(Laplacian matrix)의 고유값(eigenvalue)은 그래프의 연결 component의 갯수와 그 연결 정도에 대한 흥미로운 정보를 제공해준다. 예를 들어 어떤 그래프의 라플라시안 행렬의 행렬값은 그 그래프에 포함되어 있는 모든 non-isomorphic spanning tree를 갯수와 동일하다는 놀라운 사실[9]도 이 이론으로는 아주 간결하게 증명될 수 있다. 물론 그것을 하나씩 나열하는(enumeration)하는 과정을 갯수를 단순히 산출하는 일과는 다르지만 여하튼 Matrix-Tree Theorem은 spectral graph 이론의 유용성을 보여주는 좋은 예가 된다.

이 이론을 응용하면 우리가 일일이 확인할 수 없는 크기의 그래프, 예를 들어 전 세계의 컴퓨터들이 얽혀있는 Web graph라든지, 또는 Facebook의 친구간 그래프라든지 그 전체의 확정된 모습을 알지 못하는 경우 그 일부분의 행렬구조를 분석하여 전체의 모습을 재구성할 수 있다. 특히 요즘의 사회연결망(Social Network) 연구에서 이 이론은 가장 중요한 분석도구가 쓰이고 있다. 양자화학(Quantum Chemistry) 연구에서도 스펙트럴 그래프 이론은 핵심적인 도구로 활용된다고 한다. SNS 연구가 활발해짐에

[8] 예를 들면 10000×10000 형렬의 determinant 값을 구할 수 있는 일
[9] Matrix-Tree Theorem으로 알려져 있다.

따라서 이 이론은 더욱 주목을 받는 분야가 될 것이다. 대표적인 세계적 대가로는 중국인 F.R.K. Chung을 들 수 있으며 미국수학회(MAA)에서 발간된 그녀의 주요 monograph를 참조하면 이 분야를 보다 체계적으로 접할 수 있을 것이다[1]. 이 분야는 전통적인 그래프 이론(그림으로 표시되는)과 매우 상이하여 대수학에 대한 지식이 많이 요구되어 진입장벽이 상당히 높은 편이다. 입문용 서적으로는 R.B.Bapat이 쓴 수학과 고학년 교재 정도가 가장 적절할 것이다[2].

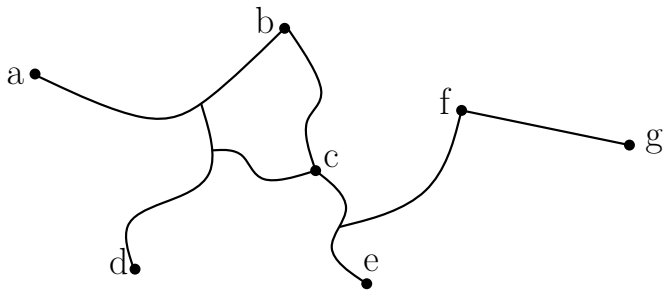
1.1.4 하이퍼 그래프 이론 (Hypergraph Theory)

전기회로를 그래프로 모형화할 때 한가지 어려운 점이 생긴다. 그것은 하나의 edge에 2개 이상의 정점이 연결되기 때문이다. 즉 하나의 등전위 전선 위에 3개 이상의 소자가 연결되기 때문에 일반적인 그래프 이론과 같이 2개의 정점으로 하나의 edge가 이루어지는 형식으로는 표현할 수 없다. 그렇다고 해서 모든 쌍에 대하여 각각의 edge를 지정하면 많은 갯수의 clique이 생기기 때문에 원하는 분석을 할 수 없다. 따라서 하나의 edge에 2개 이상의 정점을 허용하는 보다 일반화된 그래프 모형이 필요하게 되었고 이것을 집중적으로 다루는 분야가 바로 이 연구분야이다. 이런 일반적인 그래프 모형을 하이퍼 그래프라고 부르고, 어떤 연구자들은 fractional graph 라고또 부른다.

이런 모형이 회로도와 같은 데이터를 다룰 때에는 편리하긴하지만 분석에서 많은 어려움이 있다. 기존의 거의 모든 그래프 알고리즘이 두 노드를 연결한 기본 (x, y) 에지 모형을 가정하고 있기 때문에 이 유용한 도구들을 하이퍼 그래프 이론에 바로 활용할 수 없는 단점이 있다. 아래 그림-1는 하이퍼 그래프의 한가지 예를 보여준다. 중간에 vertex 없이 선이 연결된 점은 vertex가 아니며 그 위치는 의미가 없다. 그 점은 3개 이상의 vertex가 하나의 edge에 연결되어 있음을 보여주는 가상의 위치일 뿐이다.

1.1.5 완전 그래프 이론(Perfect Graph Theory)

완전 그래프 이론은 알고리즈믹 그래프 이론(algorithmic graph theory)의 한 분야로서 성질을 규명하는데 집중하는 앞서의 전통적인 그래프 연구와는 목적에서 좀 다르다고 할 수 있다. 알려진대로 대부분은 의미있는 그래프 문제, 예를 들면 그래프 색수문제(graph coloring problem)나 Hamiltonian cycle 문제는 잘 알려진 NP-complete에 속한다. 알고리즘 측면으로 볼 때 그래프 연구는 서로 다른 2방향에서 진행되고 있다. 하나는



$$V = \{a, b, c, d, e, f\}$$
$$E = \{(a, b, c, d),\ (b, c),\ (c, e, f),\ (f, g)\}$$

Figure 1: 두개 이상의 vertex가 하나의 edge를 구성하는 것이 허용되는 하이퍼 그래프 (hypergraph)의 예. 정점은 6개이며 edge는 4개이다.

전통적인 방법으로 그래프 문제의 알고리즘의 복잡도를 다항시간으로 내리려는 노력이 그것이다. 즉 다항시간에 해결이 가능한 다양한 그래프 문제, 특히 실제 현실에서 사용할 수 있는 그래프 문제를 발굴하기 위한 연구가 그것이다. 다른 하나의 연구방향은 일반 그래프가 아니라 그래프 class를 제한해서 기존의 NP-complete가 다항시간에 해결되는 좁은 그래프 클래스를 찾아내고 이것을 확장하는 것이다.

예를 들어 대부분의 그래프의 문제는 트리 (tree)에서는 다항시간에 풀린다. 따라서 tree보다 좀 더 일반적인 그래프 클래스 중에서 대부분의 NP-complete graph problem이 다항시간에 풀리는 그래프를 찾아내서 그것을 확장하는 것은 의미있는 작업이다. 그런데 우리의 예상과는 달리 상당한 제약조건이 있음에도 불구하고 그 문제가 그래도 NP-complete에 남아있는 경우는 허다하다. 예를 들어 평면 그래프 (planar graph)가 전형적인 예인데, 대부분의 NP-complete 문제는 평면 그랴프에서도 변함없이 NP-complete로 남아있다. 아주 부분적인 그래프 클래스에 대하여 다항시간의 알고리즘이 제시되었지만 그런 결과는 매우 trivial한 불과했다. 이후 쿨롱[3] 등 많은 그래프 이론가에 의해서 주요한 NP-complete 그래프 문제가 다항시간에 풀리는 상당히 일반적인 그래프 클래스가 제시되었는데, 그것이 바로 완전 그래프 (perfect graph)라는 새로운 클래스이다.

G가 완전 (perfect) 그래프라는 것은 G의 모든 subgraph에서 maximum clique number와 chromatic number가 같음을 말한다. 현재까지 알려진 완전 그래프의 종류는 매우 많다. 그 중 중요한 class만 소개하면 구간 그래프 (interval graph), compatibility graph, triangulated graph, chordal graph, permutation graph, k-tree, (line) intersection graph

등이 있다. 그런데 제시된 위의 그래프들은 응용에서 흔히 나타나는 그래프들이다. 이런 현실적으로 쉽게 접하는 그래프에서 이전에는 NP-complete라고 생각되어 시도하지 못한 clique number, graph coloring 등의 문제를 다항시간에 해결할 수 있게 된 것은 큰 의미를 가진다. 완전 그래프 이론는 M.C. Golumbic에 의해서 완성되었는데 이 이론은 그의 역작을 통해서 확인할 수 있다[4]. 이 연구 덕분에 다항시간에 해결할 수 있는 그래프 클래스는 매우 넓어졌다고 할 수 있으며 NP-complete라고 짐작되어 피한 여러 응용분야에서 알고리즘적으로 큰 진보가 이루어졌다. 이 이론은 Parameterized Complexity Theory와 함께 알고리즘 연구의 최신 분야가 되고 있다.

1.2 그래프 문제의 알고리즘적 이슈

그래프는 실생활의 다양한 문제를 해결하기 위한 모형으로 오랫동안 이용되었다. 따라서 그 실생활 문제 대부분은 그에 대응되는 그래프 문제로 변형될 수 있다. 예를 들어 학부수준에서 배우는 최소 스패닝 트리 구성(Minimum spanning tree)이라든지 Traveling Sales Person은 가장 잘 알려진 그래프 문제들이다. 그래프 문제는 특성별로 구분하면 Graph Covering, Partitioning, Subgraph, Supergraphs, Vertex Ordering, Iso-and other Morphisms으로 나뉜다[10].

다양한 그래프 문제 중에서 우리는 네트워크 정렬과 직접적으로 관계가 있는 그래프 동일성(isomorphism)문제를 본 보고서에서 분석할 예정이다. 먼저 그래프 동일성 문제를 셜명한다. 아래 그림-2에서 보면 G_a과 G_2는 vertex label이 없을 때 완전히 동일한 그래프이다. 그러나 그 아래와 같이 만일 vertex에 label이 있는 경우에는 상황은 달라진다. 단 edge에 label이 있는 경우는 고려할 필요가 없다. 왜냐하면 각 edge는 edge의 양 끝 vertex로 완전히 결정되기 때문에 vertex번호가 정해지면 edge label은 의미가 없다. 모든 그래프 연구에서 제일 먼저 결정되어야 할 것은 대상 그래프가 vertex-labelled graph인지 아닌지의 여부이다.

그래프 이론에서 한가지 유의해야 할 점은 induced subgraph와 partial subgraph의 차이를 이해하는 것이다. vertex induced subgraph는 원 그래프에서 vertex 일부분만을 선택하여 subset을 만들 때, 그들 선택된 vertex subset에 속한 edge들은 반드시 모두

[10] 이 기준은 Gary and Jhonson의 기준에 따라서 분류한 것이다.

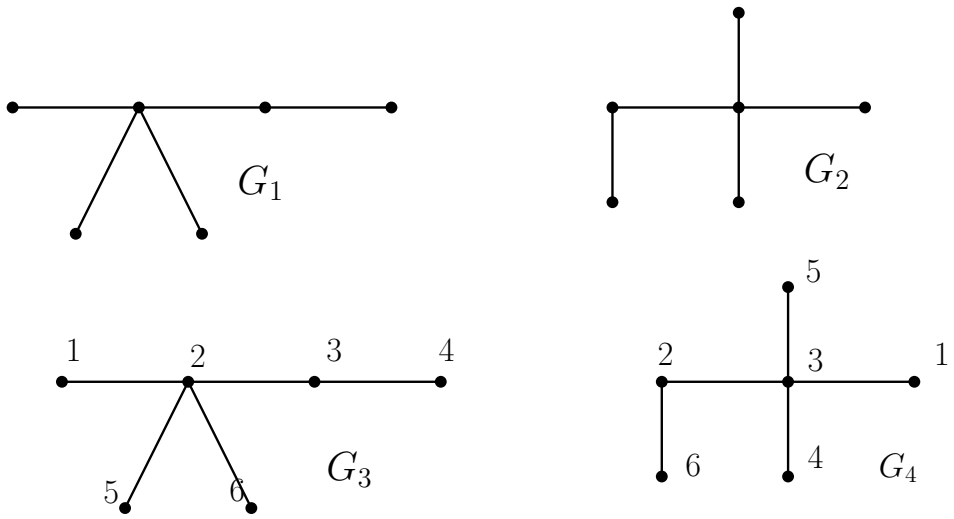


Figure 2: 정점 레이블 그래프(vertex labelled graph)와 레이블이 없이 위상(topology)만 존재하는 그래프. 두 그래프 G_1과 G_2는 모양은 달라도 위상적으로 완전히 동일한 그래프이다.

포함시킨 부분 그래프(subgraph)이다. 아래 그림-3에서 $G_{\{2,3,4,6\}}$은 전체 vertex 집합에서 vertex = { 2,3,4,6 }만을 선택하여 구성한 induced subgraph이다. 만일 여기에서 edge $(2,4)$ 가 빠진다면 induced subgraph는 될 수 없다.

특정 edge 단위로 부분 그래프를 만들 수도 있는데 그것을 partial subgrah라고 한다. 아래 그림-3에서 가장 오른쪽에 있는 그래프 $G_{partial}$가 바로 partial subgrah이다. 우리는 partial subgraph는 vertex induced subgraph의 superset이라는 것을 알 수 있다.

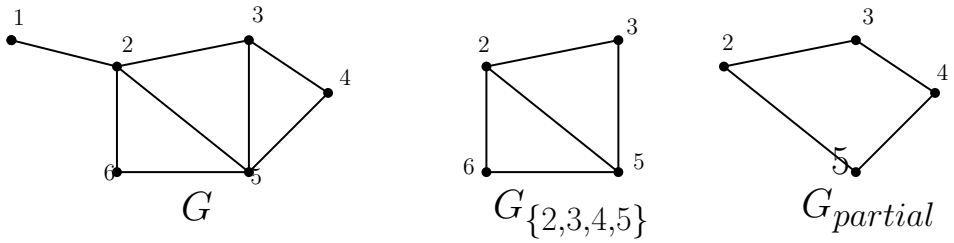


Figure 3: 어떤 그래프 G의 vertex-induced subgraph와 partial subgraph $G_{partial}$.

만일 생물학적 네트워크 (Biological Network, BN)을 그래프로 바꿀 때 node 정보가 확실하면 vertex-labelled 그래프로 모형을 만들어야 한다. 그러나 노드를 이루는 생물학적 개체의 특성이 모호하거나 모두가 같은 기능을 할 경우에는 vertex labelling이 불가능하기 때문에 이 문제는 vertex unlabelled graph문제로 접근해야 한다. 일견

생각하기에 vertex unlabelled graph 문제가 쉬운 것 같지만 본 보고서에서 다루는 그래프 비교의 문제에서는 vertex unlabelled graph가 그것이 아닌 labelled graph보다 훨씬 더 어렵다. vertex labelled graph에서는 단순히 edge (x, y)쌍의 일치여부만 보면 되기 때문에 쉬운 반면, unlabelled graph는 전체 구조를 봐야하기 때문에 훨씬 많은 계산을 요한다. 즉 unlabelled graph에서의 solution space가 labelled graph보다 훨씬 더 크기 때문에 더 어려운 문제가 된다. 아래는 그래프 이론 교재에 나오는 대표적인 문제로서 제시된 그래프 중에서 완전히 같은(isomorphic) 그래프가 무엇인지 알아내는 것이다. 한번에 눈으로 쉽게 풀 수준은 넘어서는 까다로운 문제임을 느낄 수 있을 것이다.

아래 그림-4에서 보면 3개의 그래프는 모두 같은 수의 vertex를 가지고 있다. 또한 모든 vertex의 degree(차수)도 동일하다. 일단 G_3와 G_2가 같지 않음은 쉽게 알 수 있다. G_2는 bipartite graph 이기 때문에 모든 cycle의 길이는 짝수이다. 그러나 G_3을 보면 Cycle의 길이가 5인 것도 존재하기 때문에 두 그래프는 같은 그래프가 될 수 없다.

이와 같이 두 그래프가 다른 것을 밝히는 것은 하나의 반례적 특정(counter example characteristics)만 보여주면 되지만, 두 그래프가 같다는 것은 벍히는 것은 구체적인 isomorphism function (각 vertex가 어떤 vertex 로 대응되는지를 mapping하는 작업)을 제시해야하기에 반증보다 훨씬 더 복잡하고 어려운 일이다. 이 문제의 exhaustive solution에는 모든 가능한 쌍의 경우를 다 확인해보는 $n!$번의 검증작업이 필요하다. 이 문제, 즉 두 개의 vertex unlabelled graph가 서로 isomorphic인지의 여부를 밝히는 문제가 NP-complete 문제인지도 아직 밝혀지지 않은 상황이다. 이와 유사한 subgraph isomorphism 문제, 죽 주어진 QUERY graph G_q가 어떤 target graph G_u안에 embedded 되어있는지의 여부는 NP-complete 문제임이 밝혀졌다. 그래프 비교에서 전체로 비교하는 문제와 부분적으로 비교하는 문제는 이후 Network Alignment 문제에서 Global alignment, Local alignment로 나누어서 설명한다.

끝으로 그래프 모형을 생물학적 네트워크에 응용할 때 고려해야할 점을 나열하면 다음과 같다.

1. 생물학적 실험결과를 바탕으로 그래프 모형을 만들면 많은 잡음 신호가 들어간다. 즉 수많은 false data가 포함된 불완전한 그래프라는 것을 인지해야 한다. 비록 제시된 그래프 모형은 Solid하게 보일지라도.

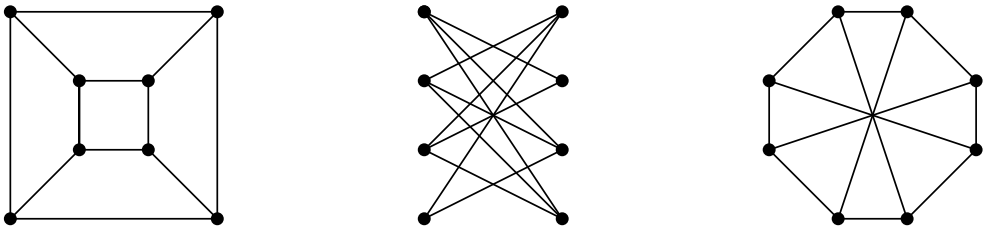


Figure 4: 대표적인 그래프 isomorphism test 문제. 문제는 주어진 3개의 그래프 G_1 G_2 , G_3 중에서 isomorphic한 그래프 쌍이 존재한다면 그것을 찾는 것이다.

2. 실험이 가능한, 예를 들어 단백질 상호작용 실험을 할 수 있는 경우가 수(edge)가 실험 개체(vertex, 예를 들면 단백질)수의 제곱으로 늘어나기 때문에 모든 것을 실험으로 검증하기는 불가능하다. 따라서 그것을 바탕으로 만든 그래프 모형에는 상당히 많은 true data 가 빠져있는 셈이므로 빠져있는 생물학적 사실을 그래프 모형과 계산적 기법을 동원하여 추론하는 것이 가장 중요한 이슈가 된다.

3. 설사 완벽한 생물학적 그래프 모형이 만들어졌다고 하더라고, 그 그래프에서 뭔가 가치있는 것을 알고리즘으로 찾아내거나(주로 optimization solution을 이용해서), 성능의 근사치가 보장된 solution을 찾아내는 일은 현실적 계산이 불가능한 NP-complete 이상의 복잡도를 가진 문제들이므로 최적의 답을 구해야 한다는 생각을 버려야 한다.

1.3 그래프와 네트워크 : 용어의 상이점

많은 연구 보고서나 논문에서 보면 어떤 경우에는 그래프, 어떤 경우에는 네트워크라는 용어가 쓰인다. 둘의 차이에 대하여 명확한 규정은 없지만 본 연구자의 경험으로 볼 때 그래프는 위상적 중요성을 강조할 때 쓰이는 표현이고 네트워크는 그 내부의 Dynamics를 강조할 때 쓰이는 용어라고 판단된다. network flow analysis가 전형적인 예가 될 수 있을 것이다. 즉 시간의 흐름에 따라서 특정 edge를 통하여 정보가 전달되고 또는 그 edge가 사라지고 각 노드상의 정보가 자주 바뀌는 경우에는 Network이라는 용어가 자주 쓰인다. 한편 전체 연결도(connectivity)와 같이 특정한 노드의 위상적 중요성과 같이 정적인 모형에서 각 노드가 차지하는 구조적 문제를 다루는 경우에는 보통 graph라는 표현을

쓰는 편이다. 생물학적으로 볼 때 각 기관이 서로 엮여있는 모습은 그래프 이론적 접근이 필요하지만 세포내에서의 신호전달, 유전자들간의 시간에 따른 상호작용은 Genetic Network으로 표현되고 있듯이 내부의 동적인 변화를 강조할 경우에는 network이라는 표현이 더 보편적이다.

2 그래프 유사도 계산문제

생물 그래프를 비교하는 것은 그 안에 숨어있는 분자생물학적으로 의미있는 과정을 찾아내기 목적이다. 그러나 우리가 가지고 있는 생물 데이터 자체에는 피할 수 없는 실험오류와 잡음, 그리고 전 과정을 모두 확인할 수 없는 본질적인 문제가 내재하고 있기 때문에 같은 데이터를 이용한 분석도 관찰자에 따라서 전혀 다른 결과로 나타나기도 한다[3]. 그 이유는 우리가 연구하고자 하는 생물 네트워크 연구에는 기본적으로 그 모형에 대한 가정을 이미 하고 있기 때문이다. 우리가 가정한 모형 자체가 달라지면 그 결과도 당연히 달라진다. 따라서 이런 모형으로 쓸 수 있는 다양한 그래프 모형에 대하여 먼저 살펴보고자 한다.

2.1 네트워크 모형의 설정

가장 일반적인 모형은 앞에서 설명한 vertex unlabelled (labelled) graph가 있다. 이 모형은 우리가 탐구하고자 하는 그래프의 전체 위상을 우리가 완벽하게 알고 있다는 것을 가정하고 있다. 다르게 표현하자면 탐구대상이 그래프로 완벽하게, explicitly expressed된 모형이다. 주로 적은 갯수의 노드로 구성된 일반적인 그래프가 여기에 속한다. 예를 들어 지하철 노선이라든지, 작은 집단내에서 친분관계, 또는 상하관계를 그래프로 표현한 것이 여기에 속한다. 문제는 이렇게 표현하기가 불가능한 그래프를 다룰 때 우리가 어떤 그래프 모형을 가정해야 하는가이다. 그 모형은 크게 무작위 그래프 모형 (Random graph), Small-world Network, Power-law Network[11], 기하기반 네트워크 (Geometric Network)로 나뉜다. 물론 각 모형들은 세부적으로 매개변수를 어떻게 설정하는가에 따라서 새로운 변형 모델의 네트워크로 발전시킬 수 있다. 아래 그림-5은 각각의 예를 보여주고 있다.

[11] 또는 Scale-Free Network이라고도 불린다. 연구분야에 따라서 다르게 불린다. 보통은 Complex Network 이라면 이런 모형을 가정한다고 생각하면 된다.

무작위 그래프는 말대로 각 edge가 임의의 노드를 연결하는 방식으로 만들어지는 것이다. 즉 두 vertex x, y 를 연결하는 edge (x, y)는 확률변수로 주어진다. 그 중에서 가장 단순한 모형은 Erdös-Rényi Random graph model로서 $G(N, p)$로 표현된다. 여기서 N은 vertex의 갯수 즉 $|G| = N$이고 임의의 edge (x, y)가 존재할 확률은 uniform distribution의 확률 $0 < p < 1$로 주어진다. 만일 $p = 0$이면 edge가 하나도 없이 N개의 vertex만 존재하는 NULL graph, 또는 N개의 정점을 가진 complete graph의 complement[12] graph인 $K(N)^c$이 된다. 그리고 $p = 1$ 이 되면 complete graph K_N 이 된다. 이 그래프의 expected number of edges는 당연히 $p \cdot N(N-1)/2$가 된다.

별 의미없이 보이는 이 random graph의 가장 중요한 특징은 criticality[13]를 가진다는 것이다. 즉 p값이 변함에 따라서 어떤 특정한 값 근처에서 성질이 급격하게 변화하는 현상을 보여준다. 흥미로운 예를 들어보자. 이 그래프$G(N, p)$에서, p값이 증가함에 따라서 전체 그래프가 하나의 연결된 그래프로 될 확률 $C(p)$는 특정점에서 급격히 올라가는 특성을 보인다. 인간사회나 집단에서도 한 집단의 개별 친밀도가 일정 이상을 넘어가면 전체가 하나로 연결, 즉 소문이 전체로 퍼질 가능성이 급격히 올라가는 급변현상(critical transition)을 볼 수 있다. 이 급변현상은 현대 과학의 가장 중요한 문제중 하나이다. 즉 급변이 일어나기 전까지는 어떤 외형적 조짐도 나타나지 않다가 아주 미급변점(critical point)근처에서의 아주 세한 변화에 따라 상황은 급변하게 된다. 급변현상이 일반변화에 비하여 보여주는 또 다른 특성은 급변된 상황이 이전 상황으로 되돌아오는 것은 매우 어렵다는 것이다.[14]

Random graph 모형은 전염병의 전파를 예방하거나 설명하는데 매우 적절하다. 예를 들어 각 사람들끼리의 상호작용이 일정 이하가 되면 전염병은 특정지역에서 발생하여도 잠시 창궐해도 더 이상 퍼지지 않고 쉽게 사그라든다. 이이 비해서 사람들끼리의 접촉의 정도가 critical point보다 조금만 더 높으면, 즉 그래프로 보았을 때 평균 degree수가 p_c보다 조금만 더 높으면 병은 삽시간에 전 지역으로 빠르게 퍼지게 된다. 문제는 이

[12] G의 complement graph G^c는 G에 존재하는 edge 는 없어지고 G 에 없는 edge가 추가되는 그래프이다.

[13] 물이 어는 과정도 여기에 포함된다. 순수한 물은 0도에서 갑자기 그 상태(phase)가 변화하여 액체에서 고체, 또는 고체에서 액체로 바뀐다. -5도에서 물이 90% 얼고 +2도에서 5%가 얼지않는다. 0 도에서 모든 분자가 갑자기(critically) 어는 것이다.

[14] 경로가 매우 긴 비가역적 반응이라고 한다. 예를 들어 싸운 사람이 화해하여 싸우기 이전으로 돌아가는 것은 전형적인 비가역반응이다.

현상에 criticality가 존재한다는 것이다. 즉 전역적으로 전염병이 퍼지기 바로 직전까지도 아무런 조짐이 나타나지 않다가 바로 그 다음에 삽시간에 전역으로 퍼지는 현상을 보인다는 것이다. 따라서 이런 전염병이나 컴퓨터 바이러스, 악성 소문, 각종 반사회적 개체는 critical point에 이르기 전에 적극적으로 예방을 해야만 비상상황을 막을 수 있다. Random graph 는 바로 이런 급변현상을 진단하고 예방하는데 매우 중요한 모형으로 활용되고 있다.

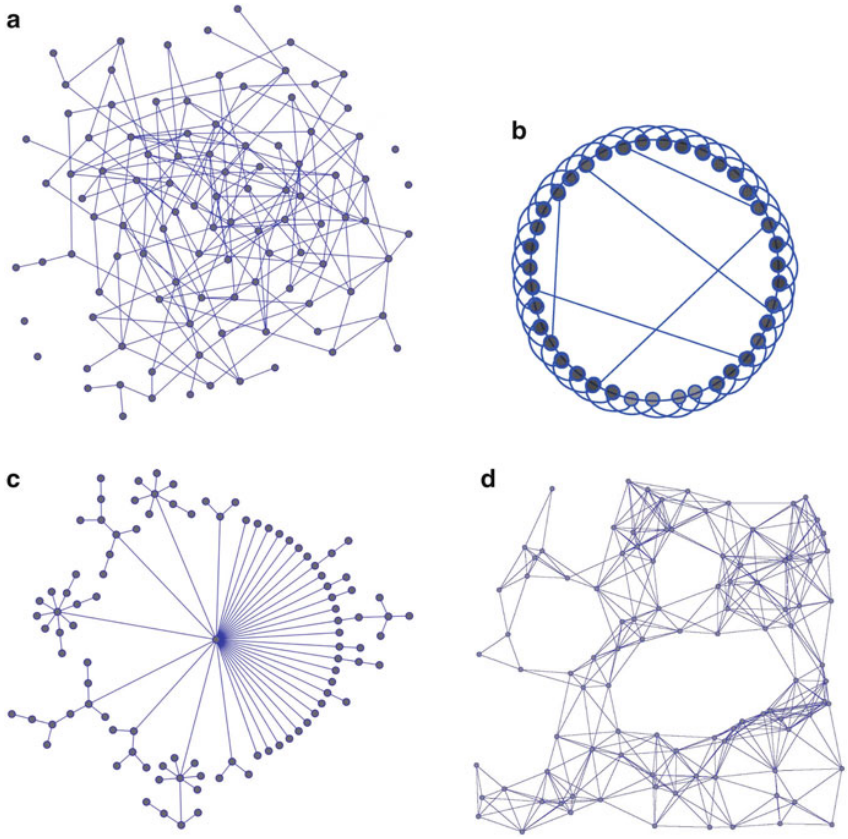


Figure 5: 가장 대표적인 4가지 그래프 모형. a)는 random graph(정확하게는 Erdös-Rényi Random graph model) 이다. 어떤 edge는 거리에 상관없이 임의로 선택된 2 개의 노드를 연결한다. b) small-world graph 이다. c) 는 hub node가 존재하는 scale-free graph, 또는 power-law graph이다. 그리고 마지막 d) 그래프는 geometric graph 이다. 즉 두 개저의 정점은 그 둘의 거리가 L_0이하가 되면 항상 연결된다.

그 다음은 small-world 모형이 있는데 Watts-Strogatz(WS) random graph라고 불린다. 이 모형에는 3개의 매개변수가 존재하여 $WS(n, k, p)$로 표현된다. n은 전체 vertex의 수를 나타내며 다음과 같이 구성된다. 일단 n개의 노드를 원주 위에 놓는다. 그리고

각 노드에서 왼쪽으로 원주상에서 가장 가까운$k/2$개, 오른쪽으로 가장 가까운 $k/2$를 모두 연결한다. 즉 $k = 2$가 되면 양쪽으로 인접한 각 하나씩의 노드를 연결한다. 그 다음 임의의 두 노드를 확률 p를 잡아서 연결한다. 이것을 small-world graph라고 부르는 것은 각각의 노드는 물리적으로 가까운 k명의 친구와 연결되어 있으며(실제 사회생활과 빅슷하게), 그리고 간혹 먼 개체와는 낮은 p의 확률로 연결되어 있는 것과 닮아있기 때문이다. 이렇게 구성된 WS 그래프는 그 자체적으로 매우 흥미로운 특징을 잘 보여주고 있다. 예를 들어 현대 사회에서 알음알음으로 6번 정도만 건너가면 거의 모든 사람들끼리 연결이 된다는 "작은 세상(small world)" 현상은 이 그래프의 전형적인 특성이다. 즉 이 그래프에서 임의의 두 노드상의 최단거리는 상상외로 매우 짧다는 특징을 가지고 있다.

그 다음 중요한 또 다른 모형은 그림-5-(c)에 있는 Scale-free network이다. 이 네트워크는 A.L. Barabasi와 그의 지도학생인 Reka Albert가 고안한 모형으로 자연계에서 나타나는 많은 네트워크를 설명하는데 유용한 모형이다. 이 모형이 Erdös-Rényi Random graph model과 다른 것은 edge 가 생기는 과정이 각각 독립적으로 만들어지는 것이 아니라 현재의 상황에 종속적이라는 것이다. Erdös-Rényi 모형은 임의의 두 노드를 연결하는 edge iid분포확률에 의해서 독립적이지만, 이 모형에서 새로운 edge는 이미 구성된 그래프의 특성에 따라서 달라진다. 쉽게 표현하자면 새로운 edge는 이미 많은 이웃을 가진 노드(degree가 높은 노드)에 더 많이 붙으려고 한다는 것이다. 일종의 부익부 빈익빈[15] 현상으로도 설명할 수 있다.

이 모형에서 새로운 edge가 추가되는 과정은 다음과 같다. 전체 노드 중에서 새로운 edge가 선택할 vertex는 현재의 vertex degree에 비례하여 확률적으로 선택된다. 예를 들어 어떤 노드 x의 이웃이 10 명이고 다른 노드 y의 이웃이 1이라면[16] 다음 단계에서 x에 새로운 edge가 생길 가능성은 y에 비해서 10배나 높아진다다.

이런 과정으로 네트워크를 구성하다보면 degree가 높은 vertex에는 더 많은 edge가 붙어있게 되고, 에지가 적은 노드들은 갈수록 edgr가 추가되는 과정에서 도태되게 된다. 이렇게 구성된 그래프의 degree 분포를 계산해보면 그 분포는 지수분포를 보임이 증명되었다. 즉 degree가 k[17]인 노드가 생길 확률은 $(1/k)^{\alpha}$로 표현된다. 다르게 말하면

[15] The rich get richer and the poor get poorer
[16] 앞으로 각 vertex x의 degree는 $\rho(x)$로 표시한다
[17] 이웃한 노드의 갯수가 k개

차수가 높은 노드가 생길 가능성은 그 차수의 거듭제곱에 반비례한다는 것이다. 이 경우 α를 scaling factor라고 부르고 이 값은 지수분포 네트워크를 구별하는 중요한 지표, 매개변수가 된다. 지수분포 네트워크의 가장 큰 특성은 매우 높은 차수를 가진 허브(Hub)노드가 반드시 존재한다는 것이다.

실제 자연계나 사회에 존재하는 대부분은 네트워크들, 예를 들어 World Wide Web 그래프나, PPI 그래프, 각 논문들의 citation 그래프[18] 역시 그러하다. 경제학에서도 이 법칙이 존재하는데 Pareto의 법칙이 이것을 설명하고 있다. 부자는 가진 자산을 활용해서 더 큰 부자가 되지만 가난한 사람들은 투자할 돈이 없어 돈을 벌지 못하고 이 때문에 돈은 더욱 줄어들어 갈수록 더 궁핍한 생활을 하게되어 결국 가난의 굴레에서 빠져나오지 못하는 과정이 나타난다.

실제 재산의 정도에 따른 인구수를 나열해보면 대부분의 사회에서 정확하게 지수분포임을 확인할 수 있다. 이 지수분포 그래프를 Log-Log Chart로 그려보면 직선으로 나타나는데, 고 그때의 기울기가 바로 Scaling factor가 됨을 알 수 있다. 아래 그림-6는 전형적인 power-law를 따르는 사회현상의 한 예이다. 이 그래프는 Log-Log chart인데 가로축은 어떤 회사에 속한 사원의 수를 나타낸다. 가장 많은 사원의 수는 표에 나타난바와 같이 10^6이다. 그리고 세로축은 그런 회사의 전체에서의 비율을 나타낸다. 아래로 갈수록 비율이 적음을 나타낸다. 그림과 같이 그 분포는 log-log chart 에서 매우 정확한 직선을 나타내고 있다. 인간사회의 거의 모든 집단적 특성은 이 power-law를 따르며 그것이 바로 전형적인 사회현상의 한 특징이다. 다시 말해서 사회 특정 집단의 특정 property (attribute)가 만일 power-law 분포를 따르지 않는다고 하면 그 자체가 매우 중요한 연구대상이 될 수 있다.

이러한 power-law법칙을 따르는 네트워크를 scale-free network이라고도 불리는데 그 뜻은 전체가 지수분포를 따르기 때문에 척도를 알아낼 수 없기 때문이다.[19]

[18]어떤 논문 X가 다른 논문 Y를 참고했으면 edge (X, Y)를 추가한다. 이 과정은 부익부 빈익빈 과정의 대표적인 예가 된다. Citation이 많이 되었기 때문에 더 유명해지고, 더 유명해지기 때문에 많은 사람들이 인용을 하게 되는 선순환 구조를 가진다. 역으로 다른 수많은 논문들은 도태 과정을 거치는데, 초기에 인용이 잘 안되기 때문에 다른 논문에서 인용을 잘 하지 않고, 그것 때문에 다시 중요도를 낮게 평가받아 결국 사라지게 된다.

[19]만일 어떤 집단의 소득수준이 정확하게 지수분포를 따른다고 한다면, 그 중에서 상위 10%를 뽑아도 그 안에서는 다시 상위, 하부 분포가 전체의 분포대로 나타나고, 하위 10%를 선별해서 다시 그래프로 그려도 그 분포는 같은 scale factor 의 지수분포를 하기 때문에 samping된 집단안에서 분포는 항상 동일하기 때문에 그 집단이 전체에서 어디에 위치하고 있는지를 알 수 없다는 말이다. 만일 우리나라 국민들의

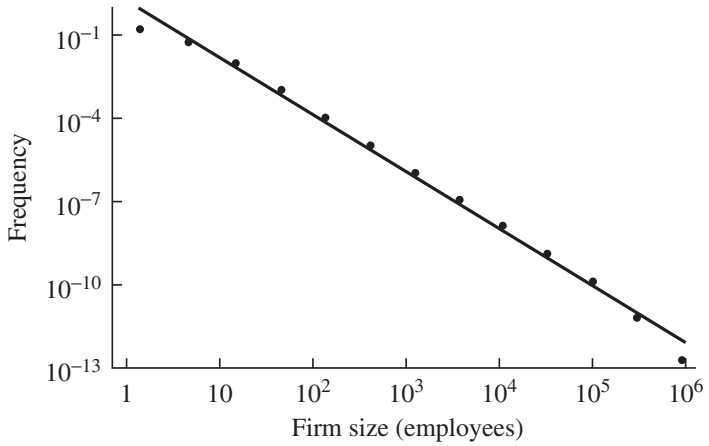


Figure 6: 가로축은 특정 업체의 종업원의 수가 Log scale로 표시된 것이며, 세로축은 그런 특성을 가진 개체의 빈도(frequency)가 로그 scale로 표시된 것이다. 이 분포는 놀라울 정도로 정확하게 지수분포를 나타내고 있다.

최근 생물 네트워크 연구자들이 주목하고 있는 것은 아래 그림-7에 표시된 것과 같은 기하 그래프(Geometric graph) 모형 이다. 기하 그래프는 각 개체의 기하학적인 위치로부터 생성된다. 쉽게 표현하자면 공간상에 점들이 분포하고 그 점들이 일정한 거리보다 가까울 경우 서로 연결된다. 즉 edge를 가지게 된다. 기하 개체는 반드시 점(point set)일 필요는 없으며 넓이를 가진 원이나 사각형, 또는 3차원 일반적인 입체도 가능하다. 이 그래프의 특징은 공간적으로 가까운 거리에 있는 개체들끼리 뭉친 형태를 보이는 것이다. 이 그래프의 최대, 최소 차수(degree)라든지, 지름(diameter)[20]에 대한 확률적인 특성은 매우 흥미롭다. 특히 기하 그래프 모형는 최근의 이동통신, sensor network design에서도 잘 활용되고 있다. 기하 그래프의 역사와 특성, 통계적인 특성, 다양한 응용의 예는 Mathew Penrose의 역작에 잘 기술되어 있으므로 참고하면 될 것이다[5].

그런데 생물 네트워크 중에 가장 전형적인 예라고 할 수 있는 단백질 상호작용 네트워크(Protein-Protein Interaction Network)의 특성이 기하 그래프의 특성과 유사하다는

소둑이 power-law를 따른다면 500대 재벌집단만을 선별하여 그린 지수분포 그래프 $k^{-\alpha}$나 하위 계층의 소득분포가 보여주는 분포는 완전히 동일하게 나타난다는 것이다. 이것은 결국 부분집단의 scale을 sampling만으로는 알 수 없다는 것을 말해주므로 이것은 scale-free라고 부른다. 또 다른 예로 어떤 험한 산에 있는 돌들의 크기가 지수분포를 가진다면 우리가 개미만큼 작아졌을 때에도 주위에서 보이는 돌들의 크기 분포나, 우리가 거대한 공룡만큼 커진 상황에서 보이는 돌들의 크기분포는 동일하게 느껴질 수 밖에 없다.

[20] 그래프 모든 점들끼리의 최단거리 중에서 가장 긴 거리

새로운 사실이 속속 밝혀져 주목을 받도 있다 [6, 7]. 만일 이것이 사실이라면 PPI연구는 새로운 전기를 맞이하게 되는데, 이전 연구자들 대부분은 PPI 네트워크가 일반적인 power-law network이라고 가정하에서 연구를 전개했기 때문이다. 왜 기하학적인 모양과 아무런 상관이 없는 PPI 네트워크가 기하 그래프적 특성을 가지고 있는지에 대해서는 아직 연구 중이지만 N. Pržulj 연구팀은 이 이유를 이렇게 설명한다.

일반적으로 유전자, 세부적으로는 각 단백질들은 진화과정에서 자기복제 (self-replication) 와 변이 (mutation) 을 하게 된다. 또 다른 단백질과 결합하여 결합한 두 개의 단백질과 유사한 새로운 단백질 서열을 만들어 내는데, 그렇게 유사한 단백질들간 상호작용은 다른 쌍들에 비해서 더 활발하다는 것이다. 그래서 상호작용이 높은 이들 쌍은 화학구조적 특성으로나 서열상으로 상당히 닮은 형상을 가지고 있을 수 밖에 없다. 따라서 이들을 유사한 것들끼리 기하공간에 가까이 배치하고 그 중 상호작용을 하는 것끼리 연결을 하면 단백질 장용 그래프는 결국 기하그래프의 구조적 특성과 유사한 연결특성을 가지는 것이 아닐까 한다[8].

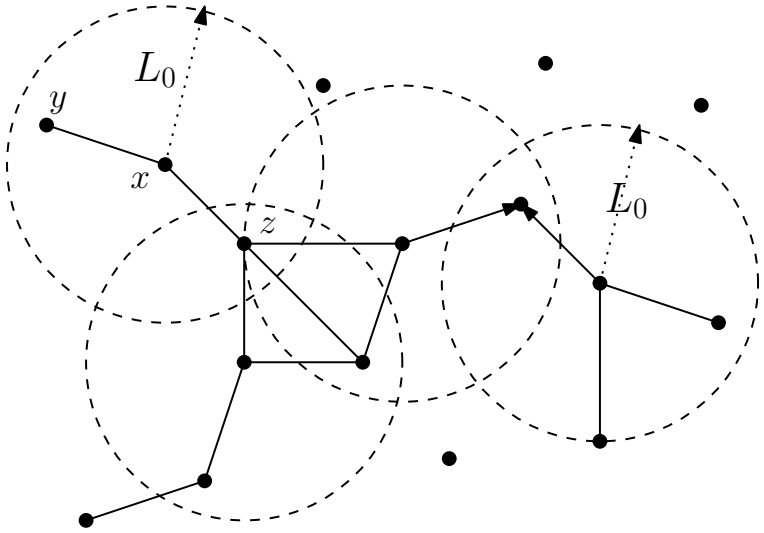


Figure 7: Geometric graph의 예. geometric graph에서 공간상의 점 (개체) 간은 그들의 거리가 어떤 정해진 문턱값 (threshold) L_0 보다 작으면 edge로 연결된다. 따라서 거리가 멀리 떨어져 있는 점은 edge로 연결되지 않아 전체적으로 disconnected component도 될 수 있다.

앞서 설명한 모형 외에도 기본 모형에서 변형된 여러 모형이 존재한다. 그리고 각각은 적절한 연구목적에 따라서 다양하게 활용되고 있다. 자세한 내용은 요즘 연구되는 각종 네트워크 관련 모형을 잘 설명한 Newman의 역작을 살펴보면 될 것이다[9].

정리해서 말하자면 생물에서 이론적으로 가능한 모든 실험을 다 해볼 수 없고 또한 그런 과정은 필연적으로 많은 오류와 잡음을 포함할 수 밖에 없기 때문에 이렇게 빠진 부분은 이론적 연구를 통하여 메꿔나갈 수 밖에 없다. 그런데 이 과정에서 우리가 어떤 그래프, 네트워크 모형을 가정하고 시작하는가에 따라서 결론은 매우 달라지고 또한 심한 경우 같은 자료로 부터 서로 상반된 결론에 도달할 수 도 있다. 따라서 생물 네트워크 연구시작 전에 어떤 네트워크 모형을 가정하는지에 대한 관점을 분명히 해야할 것이다. 그렇게 해야만 상반된 또는 모순된 결론을 최종적으로 피할 수 있기 때문이다.

다음은 이 보고서의 중심문제인 그래프 매칭, 그래프 동일성 검사 문제를 다루고자 한다.

2.2 그래프 동일성(Isomorphism) 판별문제

두 그래프의 유사도를 정량적으로 계산하는 문제는 여러 분야에서 활용되고 있다. 사회연결망 분석, 컴퓨터 비전에서 두 물체의 동일성을 판단하는 응용에 있어서 그래프의 유사도 계산은 매우 중요한 역할을 한다. 두 그래프 사이의 다양한 측도의 유사도(similarity)를 계산하는 방법은 알고리즘의 유형에 따라서 크게 세가지로 구분된다. 첫번째로는 각 vertex나 edge의 mapping 함수, 즉 isomorphic function을 직접 구하는 방법인 가장 정통적인 접근법이 있다. 이 방법은 매우 전형적인 접근법이긴하지만 그래프 노드수가 많아지면 계산이 시간, 공간상으로 불가능한 단점이 있다.

제한없이 주어진 두 그래프의 isomorphism을 찾아내는 것은 오래전부터 연구된 매우 유명한 계산문제이다. 크기가 같은 두 그래프가 서로 isomorphic한지를 판단하는 문제는 NP-complete인지 아니면 polynomial time algorithm 이 존재하는지 조차 규명되지 못한 상태이다. 같은 상황에 있는 문제[21]로 Gary and Johnson이 밝힌 12개의 문제가 추가로 존재한다. 그러나 subgraph isomorphism 문제는 NP-complete 문제임이 밝혀졌다[22].

아래 그림-8에서 (a)는 전체 그래프의 isomorphism 문제의 예를 보여주는 것이고 (b)

[21]아직 효율적인 polynomial time algorithm이 존재하지도 않으면서 동시에 이것이 NP-complete문제인지도 밝혀지지 않은, 아주 어중간한 상태에 있는 문제

[22]이 증명은 비교적 쉽다. 어떤 그래프에서 hamiltonian cycle이 존재하는지를 검사하는 문제는 이 그래프에서 같은 크기의 Cycle을 subgraph로 가지는지를 검사하는 것과 동일하다. 즉 subgraph isomorphism 문제는 hamiltonian cycle 문제보다 쉽지 않다. 따라서 NP-complete문제인 hamiltonian 문제로의 reducing이 가능하므로 제한없는 일반 그래프에서의 subgraph isomorphism문제는 NP-complete 에 속하게 된다.

는 subgraph isomorphism문제의 한 예이이다. 전체 isomorphism문제가 요구하는 것은 는 제시된 두 그래프가 vertex와 edge에서 동일하도록 되도록(만일 동일하다면), G_1의 모든 vertex를 어떻게G_2의 vertex로 mapping하는지 그 mapping함수, 또는 isomorphism을 찾는 것이다. 그러나 (a)와 달리 (b)는 G_3가 G_4의 어떤 부분과 완전히 일치하는지 그 부분을 찾는 것이 문제가 된다. 따라서 G_4의 vertex 중에서는 매핑에 들어가지 않는 vertex들이 반드시 존재해야 한다. 그런데 전체 isomorphism 문제에서는 모든 vertex는 반드시 하나씩의 짝으로 매핑되어야 한다.

즉 아래 그림-8에서 볼 때 G_1과 G_2은 위상적으로 완전히 동일하다. 이 두 그래프에서 isomorphism은 그것을 유지하는 mapping function 함수(isomorphism) 는 $f()$는 다음과 같다.

$$f(1,2,3,4,5,6,7,8) = (b,c,g,f,a,d,h,e)$$

이와 유사한 subgraph isomorphism 문제는 G_3과 같은 위상구조를 가진 subgraph를 G_4에서 찾는 것이다. 이 subgraph isomorphims문제는 위 두 그래프의 isomorphism을 찾는 문제보다 훨씬 더 어렵다. 그림-8에서 볼 때 우리는 이것이 쉽지 않다는 것을 바로 인지할 수 있다. 왜냐하면 graph isomorphism 문제는 두 그래프에서 특성이 다른 하나의 vertex나 또는 subgraph만 찾아도 isomorphism이 없다는 것이 확인이 되지만 subgraph isomorphism문제는 그런 식으로 그것의 존재가 부정이 되지 못하기 때문이다. 간단한 예를 들어 $|G_a| = 10$, $|G_b| = 11$와 같이 vertex의 수가 다르면 두 그래프는 절대 isomorphic할 수가 없다. 또한 maximum degree라든지 minimum degree가 달라도 isomorphic하지 않는 것은 쉽게 알 수 있다.

그런데 대상 그래프의 구성요건이 제한적이라고 해서 이 문제가 쉽게 풀리는 것은 아니다. 특별한 class중에서 polynomial time에 풀리는 경우는 아래에서 설명한 tree 클래스에서는 가능하고 planar graph에서도 가능하다. 그리고 대부분의 perfect graph class인 interval graph, permutation graph등에서도 isomorphism 문제는 다항시간에 해결이 가능하지만 그 외 대부분의 그래프에서는 아직도 해결되지 못한 영역으로 남아있다.

graph isomorphism문제를 해결하는 알고리즘 중에서 그나마 현실적인 solution은 2008년도 제시된 $O(2^{\sqrt{n \log n}})$ 알고리즘이 가장 나은 복잡도를 보이지만 실제 구현상으

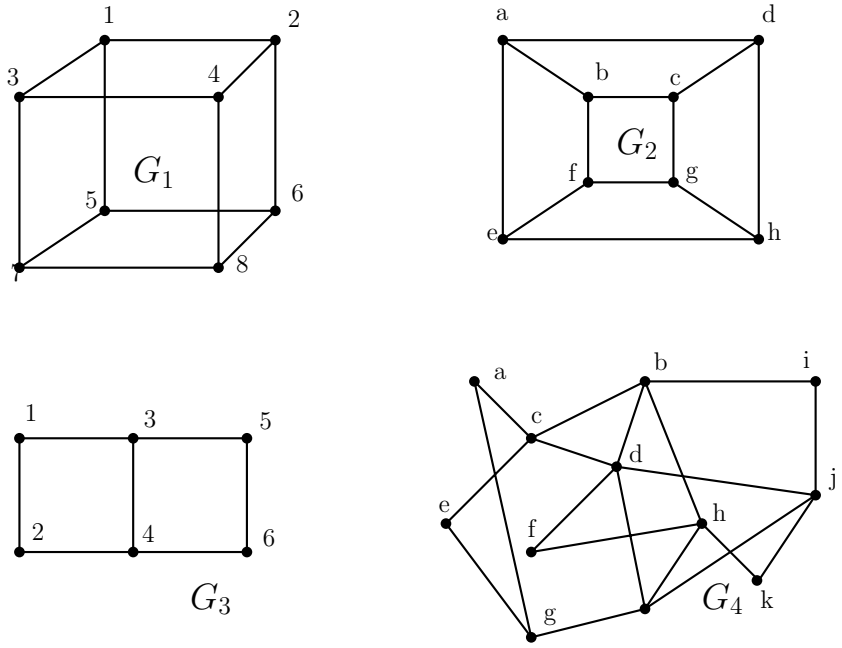


Figure 8: G_1과 G_3이 같음을 vertex mapping으로 찾아보자. 그리고 G_4안에 과연 G_3가 존재(embedding)하는지를 찾아보자. 이 두번째 문제가 subgraph isomorphism 문제이다.

로는 매우 어렵다. 이 정도 시간 복잡도라면 대략 노드의 갯수가 1000개 이상인 경우 즉 $|G| > 1000$ 인 문제는 적절한 시간에 답을 기대할 수 없다. 그러나 트리에서 그래프 동일성 문제, 정확하게 트리 동일성 문제는 선형시간에 쉽게 풀릴 수 있다. 방법은 간단하다. 두 트리 T_1 와 T_2 에서 먼저 center node[23]를 찾아서 이것을 root로 해서 rooted ordered tree를 만든다. 그리고 각 subtree를 왼쪽에서 오른쪽으로 정렬할 때, subtree의 갯수가 많은 쪽은 선호해서 정렬한다. 만일 같은 갯수라면 다시 그 subtree의 subtree의 ordering으로 순서를 정한다. 이렇게 정리된 트리를 표준형(Canonical form)으로 정리된 표준형 트리(canonical tree)라고 하는데 이 트리를 Preorder 방식으로 traversal하면 선형시간에 두 트리가 동일한지를 쉽게 판단할 수 있다.[24]

[23]트리의 Center node는 가장 점 terminal node까지의 거리가 최소가 되는 노드이다. 트리에서 이 center(정확하게는 vertex center) 노드는 최대 2개까지 존재하면 만일 2개가 존재할 경우에는 반드시 인접한다.

[24]각 트리의 center node각 최대 2개씩 있을 수 있으므로 정확하게는 각 4 번의 작업, 즉 각 트리의 center node등의 모든 쌍에 대해서 검증작업을 하면 된다.

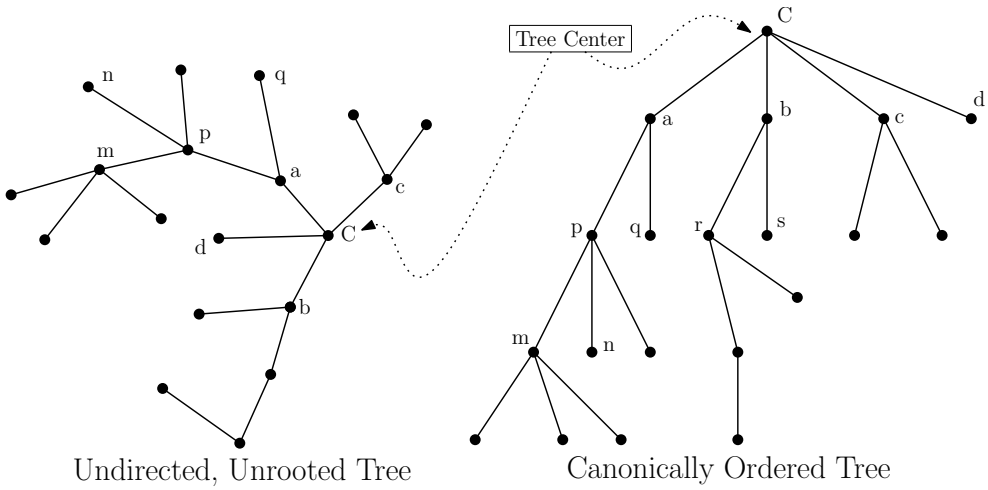


Figure 9: 트리의 경우에는 표준순서화를 통하여 쉽세 두 트리가 동일한지를 검사할 수 있다. 트리의 중심(Center of tree)를 잡아서 그 subtree들을 왼쪽부터 순서대로 배열한다. 그 순서는 subtree의 노드 수가 많은 순서대로 배치를 고 이것을 모든 subtree에 대하여 recursive하게 적용하여 모든 중간 노드에서 그 노드의 subtree들은 노드의 갯수에 따라서 왼쪽에서 오른쪽 순서대로 배치한다. 만일 그 subtree의 노드 수가 같으면 그 subtree의 subtree 중에서 가장 큰 것을 기준으로 tie breaking을 한다.

2.3 그래프의 전역특성을 이용한 유사도 측정

계산적으로 불가능한 isomorphism 검사를 대신하는 방법으로는 그래프 자체를 비교하는 것이 아니라 그래프의 주요한 특성치를 비교하는 것이다. 이러한 접근이 전역특성을 이용한 유사성 검사 접근법이다. 예를 들어 두 그래프를 비교하는 가장 간단한 방법은 그래프 $G(V, E)$의 노드 수 ($|V|$)와 edge 수를 비교하는 것이다. 물론 그래프의 노드수와 에지수가 같다고해서 두 그래프가 우연히 같을 가능성은 거의 없지만 일단 이 둘이 같으면 그나마 두개의 그래프가 같을 가능성이 있다는 점에서 검사에 관한 약간의 정보를 받을 수 있다. 다른 방법은 그래프의 특성 지표를 만들어 그래프 대신 이 지표를 비교하는 것이다. 즉 어떤 그래프 $G_a(V_a, E_a)$를 두 개의 component를 가진 vector인 $(|V_a|, |E_a|)$로 표시하여 이 vector를 대신 비교하는 것이다. 즉 두 그래프 vector간의 거리로 두 그래프의 위상적 유사도를 짐작하는 방밥이다.

이 방법이 가진 한가지 장점, 즉 그래프의 vector를 그래프 fingerprint로 이용해서 비교하는 방법은 비교대상 그래프의 크기가 아무리 커도 변환된 fingerprint 벡터의 크기는 항상 일정하다는 젬에서 계산상 유리하다[25]. 이제 남아있는 문제는 이 그래프 fingerprint

[25]아무리 그래프가 커다고 해도 fingerpint의 크기는 항상 일정하기 때문에 memory space면에서 유리

vector에 어떤 요소를 집어 넣을 것인가, 또는 그러한 vector component 를 몇 개나 넣을 것인가이다. 당연히 다양한 vector component가 많을수록 비교는 더 정교해질 수 있지만 항상 그런 것은 아니다.

이 방법은 그래프의 크기가 매우 커서 전체의 구조를 명시적으로(explicitly) 확보할 수 없는 경우에 활용하기 좋다. 예를 들어 World Wide Web은 그 전체를 탐색하는 것조차 불가능하기 때문에 이것들 간의 isomorphism을 찾는 것은 이론적으로나 실제적으로 불가능하다. 가령 예를 들어 독일 전체의 internet interconnection 과 우리나라의 interconnection을 비교하는 문제를 생각해보자. 이 경우에는 vertex나 edge 자체가 생겼다가 없어지기도 하기 때문에 graph mapping을 이용하기는 매우 어렵다. 이 경우에 비교할 그래프에 hub site가 존재하는지, 또는 일정 이상 크기의 hub이 몇개나 존재하는지 또는 복수 개의 high-degree 노드 사이를 연결하는 shortest path의 길이가 서로 비슷한지 등을 비교하는 것으로 전체 그래프의 유사성을 대신 짐작할 수는 있을 것이다.

그래프의 전역특성(global property) 지표 중 가장 단순한 지표는 노드의 수와 에지(edge) 수이다. 일단 에지의 수가 많으면 그래프가 조밀(dense)하다고 짐작할 수 있지만 그것이 complete graph와 가깝지 않는 한 에지 수는 가장 단순한 지표이다. 이보다 좀 더 나은 지표는 네트워크의 차수를 정렬한 그래픽 순서(graphic sequence)이다. 즉 전체 노드의 차수를 내림차순으로 정리한 vector를 해당 그래프 G의 그래픽순서라고 부른다. 즉 정렬된 차수는 $graphics(G) = (d_1, d_2 \ldots, d_{n-1}, d_n)$, 단 $d_i \geq d_{i+1}$를 만족해야 한다. 다음 그림-10은 그래프 G_x와 그 그래픽 순서 $graphic(G_x)$를 보여주고 있다.

그런데 그래픽 순서가 같아도 두 그래프는 전혀 다를 수가 있다는 점이다. 아래 그림-11는 같은 그래픽 순서를 가지는 다른 모양의 그래프의 예를 보여준다. 이것으로 볼 때 단순한 차수의 나열 정도인 그래픽 벡터는 비교할 그래프의 특성을 나타내기에 부족함을 알 수 있다.

생물 네트워크의 초기 연구에는 네트위크의 차수분포가 가장 중요한 특성으로 받아들여졌다. 그래프의 차수분포만 확실해지면 그것이 기반하여 전체 네트워크에서의 최단거리, 평균거리, 연결도(connectivity)[26]를 유추하여 계산할 수 있었기 때문이다. 그런데

하다.

[26] 어떤 그래프를 분리하기 위하여 제거해야할 최소의 vertex(edge) 갯수는 vertex(edge) connectivity가 된다.

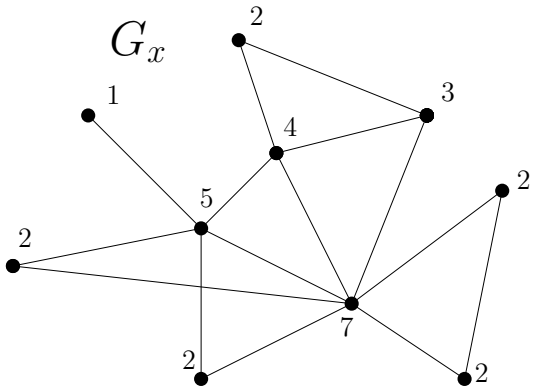


Figure 10: 주어진 그래프와 그 그래픽 순서(graphic sequence). 그래프 노드에 표시된 숫자는 해당 노드의 차수(the number of incident neighboring vertices)를 나타낸다.

새로운 분자생물학 실험기기가 속속 등장하고 대규모 실험이 가능해짐에 따라서 생겨난 많은 새로운 데이터들은 이전에 생물 네트워크의 구성 모형인 power-law, scale-free network의 특성과는 상당한 거리가 있음으로 보여주고 있다. 그 이유는 이전 실험에서 잡음을 충분히 제거하지 못했고 그것을 power-law 그래프로 over-fiiting한 결과라고 지적하는 연구자도 있다. 특히 N. Pržulj의 최신 결과에 따르면 PPI 네트워크는 scale-free라기보다는 geometric graph에 훨씬 더 가깝다는 것이다[6, 7]. 최근의 이러한 새로운 결과는 생물 네트워크 연구의 새로운 방향을 제시하고 있다. 기하 그래프의 차수분포는 포아송 분포(Poisson Distribution)를 나타내고 있음이 잘 알려져 있다.

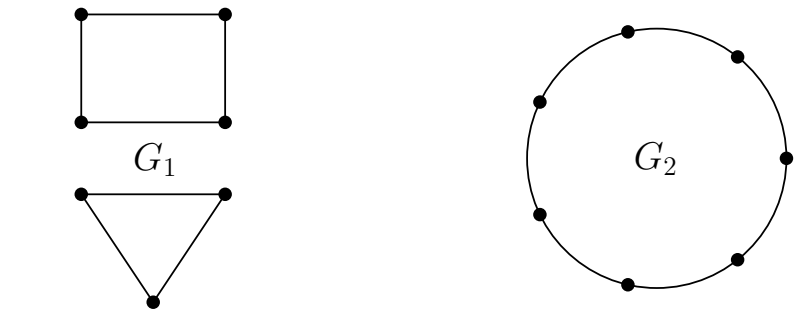


Figure 11: 같은 그래픽 순서를 가지고 있지만 서로 다른 그래프의 예. 왼쪽 그래프는 disconnected graph 이지만 오른쪽 그래프는 하나의 연결 component로 구성되어 있다.

어떤 개체의 전체적인 특성을 나타내는 가장 일반적인 방법은 특성의 분포(distribution)를 보여주는 것이다. 예를 들어 정규분포에서의 평균과 분산은 전체 집단의

특성을 보여주는 가장 대표적인 지표값이듯이 그래프의 차수 분포(degree distribution), 군집지수(clustering coefficient), 군집 스펙트럼 (clustering spectrum), 네트워크의 지름 (network diameter), 최단거리 스펙트럼은 그래프의 상세특징을 보여주는 실용적인 지표로 쓰이고 있다.

앞서 설명한 바와 같이 차수의 분포는 그래프 모형을 결정하는데 중요한 역할을 한다. 차수의 분포가 지수분포를 따를 경우 해당 그래프는 연결차수가 매우 높은 hub node가 존재하는 power-law graph, 또는 scale-free network가 된다. 만일 그 구성이 scale-free network이라고 한다면 전체의 모양을 쉽게 짐작할 수 있다. 예를 들어 degree 1인 노드의 수가 10000개 이고 degree 2인 노드가 5000 라면 비율은 1/2로 줄어들기 때문에 그 scale factor는 $\log_2(1/2) = -1$이 된다. 따라서 차수가 3인 노드의 수는 2500개 정도, 4인 노드는 1250개 정도가 될 것이라고 추측할 수 있다.

각 노드의 차수가 같으면 우리는 이것을 정규 그래프(regular graph)라고 부르는데 이같이 차수의 분포가 극단적으로 하나에 몰려있는 경우에도 그래프의 특성은 쉽게 파악된다. 정규분포에 가까운 그래프 역시 전체 특성을 쉽게 확인할 수 있는데 $k-regular$ 그래프의 diameter는 $O(\log_k |G|)$ 임을 쉽게 알 수 있다. 이보다 좀 더 좋은 특성값은 군집지수 분포이다. 어떤 노드 v의 군집지수는 v의 이웃 노드($N(v)$)의 집합간에 연결된 edge 의 수로 결정된다. 만일 모두 연결되어 있다면 지수는 1이 되고, 만일 아무것도 연결되어있지 않다면 그 값은 0이 된다. 아래 그림-12은 군집지수의 예를 보여주고 있다. 군집지수는 v의 이웃노드간 연결된 edge를 가장 많을 경우와 비교해서 비율로 나타낸 것이다. 즉 이웃들끼리 연결된 edge의 수가 E개라면 그 값은 $E/\binom{|N(v)|}{2}$ 로 정의된다. 만일 아래 그림-12의 (c)와 같이 이웃의 갯수가 5개이고 그들간 모두 연결되는 경우에는 $10/\binom{5}{2} = 1$ 이 된다. 전체 그래프의 군집지수는 각 노드 군집지수의 평균으로 계산된다. 그래프의 군집 스펙트럼(clustering coefficient)는 전체 노드의 평균 군집지수를 degree 별로 표시한 vector이다.

그래프나 네트워크에서 두 노드 x, y의 거리(distance), $dist(x, y)$는 두 점을 연결하는 최단거리로 정의되는데 이 거리의 분포 정보도 그래프의 전역 특성을 나타내는 중요한 지표로 쓰인다. 그래프 G의 지름(diameter) 인 $diameter(G)$는 $\max_{u,v \in G}\{dist(u, v)\}$ 으로 정의된다. 그래프의 지름 모든 쌍간의 최던거리 중에서 최대값으로 정의되어 있는데

이 지름은 그래프 노드들이 얼마나 가까이 연결되어있는지를 설명해주는 좋은 지표가 된다.

어떤 생물 네트워크의 군집지수가 a이거나 또는 지름이 d라고 밝혀진 경우 이 결과를 어떻게 활용할 것인지는 아주 중요한 문제다. 그런데 그 값 자체만으로는 의미를 줄 수 없기 때문에 우리는 random graph를 만들어서 그 그래프의 군집지수나 지름을 랜덤 그래프의 그것과 비교하여 새로운 의미를 찾아낸다. 일반적으로 random graph의 지름은 $O(\log n)$ 임이 잘 밝혀져 있는데 만일 어떤 PPI의 지름이 이보다 훨씬 더 커거나 또는 더 짧을 경우에는 그 생물 네트워크 또는 PPI는 다른 어떤 의미있는 특성을 가지고 있을 것이라고 추측할 수 있다. 일반적으로 PPI와 같은 생물 네트워크들의 군집지수는 random graph 와 비교해서 훨씬 높음을 알 수 있는데 이런 사실을 이용할 수 있다. 그러나 네트워크의 전역적인 특성만으로 네트워크를 구분하기에는 아직 부족한 면이 많다는 것이 여러 논문에서 지적되고 있다. 예를 들어 전역특성은 거의 비슷하지만 전체적인 특성은 크게 다른 다양한 네트워크들이 이미 다른 연구에서 지적되고 있다[3].

요약하자면 전역특성은 비슷한 생물 네트워크를 분류하는데는 도움이 되지만 그것만으로는 부족한 것이 사실이다. 전역특성이 도움이 되는 경우는 서로 다른 네트워크를 분별하는데에는 유용하게 사용될 수 있다. 예를 들어 어떤 두 네트워크가 전역적 특성이 다를 경우라면 일단 이 두 네트워크의 특성은 다르다고 봐야 한다. 전역 특성은 서로 다름(difference)를 확인하는데에는 중요한 지표가 될 수 있지만 서로 유사한 특성을 가지고 있음으로 보여주기에는 초기 preprocessing step에서 사용될 수 있는 정도라고 할 수 있다. 예를 들어 어떤 두 미생물의 pathway network을 비교해서 한 개체가 다른 개체의 모델(reference model)이 될 수 있는지를 살펴보고자 할 때, 만일 두 network의 전역특성이 다르다면 일찍 비교대상에서 제외할 수 있어 전체 작업 일정을 단축시킬 수 있다. 게다가 대부분 우리가 얻을 수 있는 생물 네트워크 자체는 그 자체로 불완전한, 또는 실험상의 잡음이 많이 포함된 결과이기 때문에 전역특성을 이용한 네트워크의 유사성 비교는 아직 부족한 면에 많다. Web graph라든지 비생물적 개체에서 전역특성, 예를 들어 그래프 adjacency matrix의 eigenvalue, eigenvectors(특성값, 특성벡터)는 유용한 도구이지만 이런 static한 그래프가 아닌 생물 그래프에서 전역특성은 신뢰할만한 지표는 되지 못하고 있다.

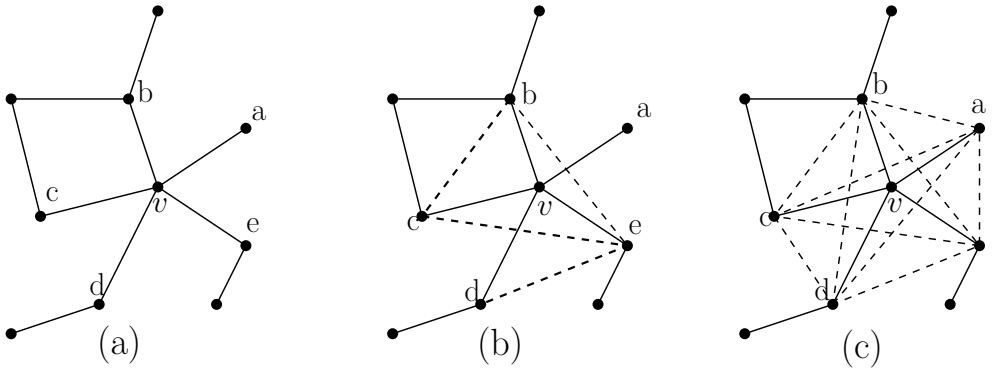


Figure 12: 노드 v의 군집지수를 계산한 3가지 예. v의 이웃노드인 {a, b, c, d, e}끼리의 연결정도가 군집지수 계산에 사용된다. (a)의 경우에 하나의 연결 edge가 없으므로 군집지수는 0/10=0.0, (b) 의 경우는 4개(점선으로 표시)가 존재하므로 4/10=0.25, (c)의 경우에는 모든 가능한 쌍에 대하여 존재하므로 10/10=1.0 이 됨을 알 수 있다.

2.4 그래프의 지역특성을 이용한 유사도 측정

네트워크의 지역특성(local property)은 매우 중요한 개념이다. 생태계가 모두 연결되어 있고 인간의 두뇌 뉴런들이 조밀하게 연결되어 있지만 실제 신경세포의 예와 같이 구조를 본다면 각 단위체들은 모두 주위의 몇 이웃들과만 연결되어 있다. 이들이 전일적으로 기작을 하여 전체가 하나의 제어구조상에서 움직이는 것 같이 관찰되지만 실제로는 bottom-up 구조로 연결되어있고 그 방식으로 동작을 하고 있다. 따라서 각 개체의 local 특성을 파악하는 것은 생물 네트워크 이해의 중요한 모멘텀이 될 수 있다. 실제 분자생물학 기준으로 볼 때 어떤 기능이 개체의 전신에서 동시에 일어나는 것은 거의 없기 때문이다.

지역특성을 이용하여 두 그래프를 비교하는 기본적인 방법론은 기본 building block을 먼저 정의한 뒤에 component를 이용하여 그 기본 component가 얼마나 존재하는지를 비교하는 것이다. 가장 대표적인 방법이 graphlet 을 이용하는 방법이다. 이러한 방법을 composition analysis 방법이라고 부르는데, DNA 를 비교할 때 k-mer의 구성성분을 비교하는 것이다. 즉 어떤 긴 길이의 DNA 를 유사성을 직접 string alignment 를 이용해서 비교하는 것이 아니라 2-mer 즉 AA, AC, AG, AT, GA, GG, GT, GC, TA,TG,TC,TT,CA, CG,CC, CT가 나타난 비율을 비교하는 방식이다. 이 방식은 결국 위에서 설명한 특성비교법과도 유사한 면을 가지고 있다. 이 방식을 Network Query에 기반한 방법이라고도 말한다. 즉 Query graph(graphlet) g_Q가 대상 그래프인 G_a, G_b

에서 얼마나 자주 나타나는가를 비교함으로서 유사성을 짐작하는 것이다[10]. 비유하자면 두 개의 긴 문서를 비교할 때 몇 개의 단어를 던져서 그 단어가 두 개의 문서에서 나타난 비율을 조사해서 유사도를 비교하는 방법과 유사하다. 문서비교에서 이러한 방법은 TF-IDF(term frequency, Inverse-Document-frequency)과도 본질적으로 같은 것이라고 할 수 있다.

그 중에서 가장 중요한 개념은 biological motif(모티브)이다Motif[11]. 모티브란 작은 subgraph의 일종인데 전체적으로 그 발현빈도가 무작위 그래프에서의 평균 발생빈도이상으로 자주 나타나는 생물학적 구조체이다. 평균이상으로 자주 나타나는 것은 대부분 그 안에서 어떤 특별한 기능과 의미를 가지고 있다. 예를 들어 한글을 글자단위로 분석해볼 때 글자 "은", "는", "이", "가"는 다른 random한 단위 문자의 출현빈도이 비해서 유의미하게 나타남을 확인할 수 있다. 이 비율의 의미가 말하는 것은 다른 random한 한글 한자에 비해서 "은", "는", "이", " 가" 글자들이 뭔가 다른 기능을 한다는 것을 짐작할 수 있게 해준다. 누군가 영어를 전혀 모르는 외계인이 영어를 분석해본다면 "a", "an", "the"가 훨씬 자주 나타남을 확인할 수 있을 것이다. 따라서 이 단어는 다른 단어와 비교해서 뭔가 다른 기능을 할 것임을 예상할 수 있고 그러한 방향으로 연구를 할 수 있듯이 뭔가 자주 나타나는 개체(frequent item)는 미지의 데이터 분석에서 아주 중요한 시발점이 된다.

생물학적 motif[27] 라고 부른다. 그리고 이 motif를 찾아내는 일은 모든 분자생물학에서 매우 중요한 초기 작업이 되고 있다. 그런데 여기에서 한가지 고려해야할 point가 있다. 우리가 자주 만나는 subgraph의 빈도를 random한 그래프와 비교를 해야하는데 그 random 그래프가 어떤 모양인가에 따라서 전혀 다른 의미로 해석할 수 있다. 이런 이유 때문에 생물 네트워크 연구에서 가장 중요한 것은 우리가 연구하려는 네트워크의 모형이 어디에 기초하고 있는가를 확인하는 일은 가장 선행되어야 하는 작업이다.

두 그래프를 지역적 특성에 기초하여 비교하는 최신 방법으로는 graphlet기법이

[27]생물학적 motif는 정의가 명확하지 않는 buzz word이다. 서열연구에서도 일단 자주 나타나는 비슷한 substring을 찾는데 이것은 유전자의 기능을 조절하는 부분으로 유전자의 서열과 기능은 달라고 이 부분은 모두 비슷하다. 따라서 sequence에서 motif를 찾아내는 것은 유전자 사냥의 첫단계가 된다. 다른 비유를 하자면 file system에서 볼 때 file의 내용을 사용자의 의도에 따라서 제각각 이지만 file head 부분은 대부분 비슷한 구조를 가지고 있다. UNIX에서 i-node구조를 motif와 개념적으로 유사하다고 생각하면 된다. 따라서 미지의 disk에서 뭔가 의미있는 binary data를 찾아내려고 한다면 이러한 i-node motif부터 찾아야 할 것이다.

있다[12]. 이 방법은 graphlet이라는 단위 그래프가 두 그래프에서 출현하는 빈도를 기준하고 그 빈도 graphlet vector로 그래프를 대신 비교한다. Graphlet은 작은 몇 개의 노드로 구성된 모든 가능한 subgraph의 집합이다. 그림에는 5개 노드로 구성할 수 있는 30 개의 서로 다른 Graphlet이 있다. 우리는 이것을 이용해서 어떤 두 network N_a와 N_b에서 선택한 두 노드 x와 y가 지역적(locally)으로 얼마나 유사한지를 밝히고자 한다[12]. 여러 연구에서 이 방법이 활용되고 있다[13, 14].

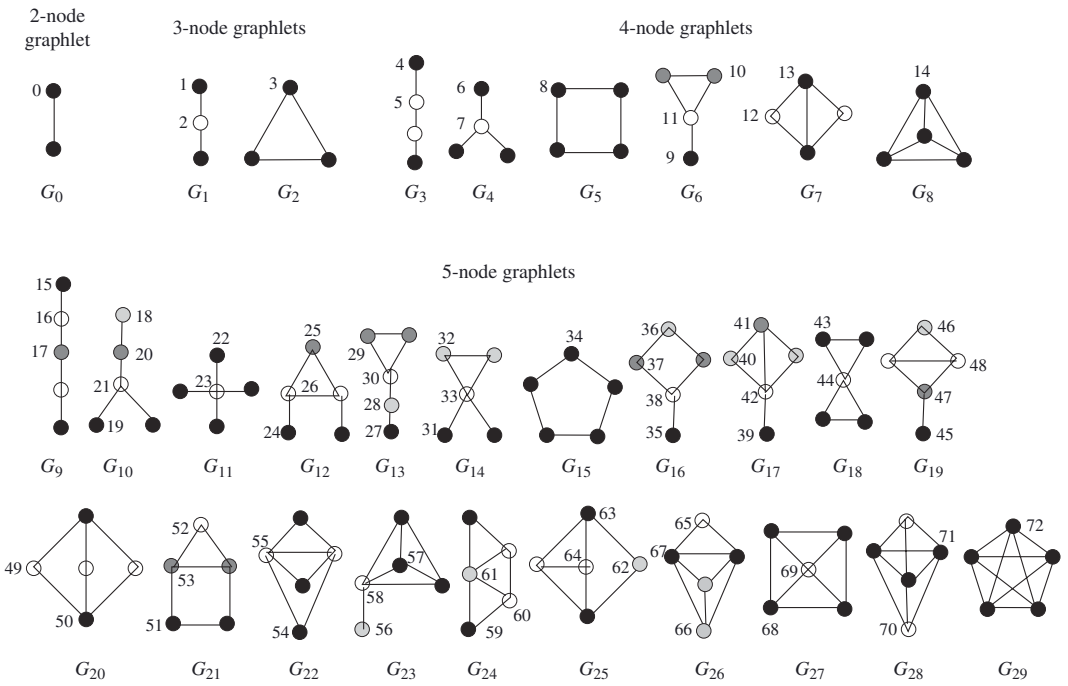


Figure 13: 5개의 노드를 가진 모든 가능한 connected subgraph들의 graphlet이 된다. 단 disconnected graphlet을 고려하면 갯수가 너무 많아지므로 graphlet은 연결된 subgraph만 고려한다. 그림에서 노드에 붙은 번호는 unlabelled graph에서 노드들의 서로 다른 위상 일변번호를 나타내고 있다. 이것을 automorphism orbit이라고 한다. 예를 들어 5개의 노드가 하나의 Cycle 로 이루어진 graphlet의 경우 각 노드는 모든 위치에서 위상이 같기 때문에 서로 다른 orbit은 단 하나 뿐임을 알 수 있이다.

두 그래프의 각 두 vertex u, v 에서의 지역적 유사성을 비교하기 위해서는 해당 위치에서 일정부분에 포함된 subgraph를 모두 찾아내서 그것의 출현빈도 분포로 graph similarity를 계산하는 것이다. 그런데 이 접근에는 2가지 문제가 있다. 하나는 아래 그림과 같이 d_G의 범위를 정하는 것이고 다른 하나는 그렇게 잘라낸 local subgraph을 어떻게 비교할 것인가이다. Graphlet에서 택한 방법은 이 u, v에 걸쳐있는 graphlet의

종류를 비교함으로서 그 노드들의 지역적 유사성을 이들의 graphlet 빈도로 짐작하고자 하는 것이다.

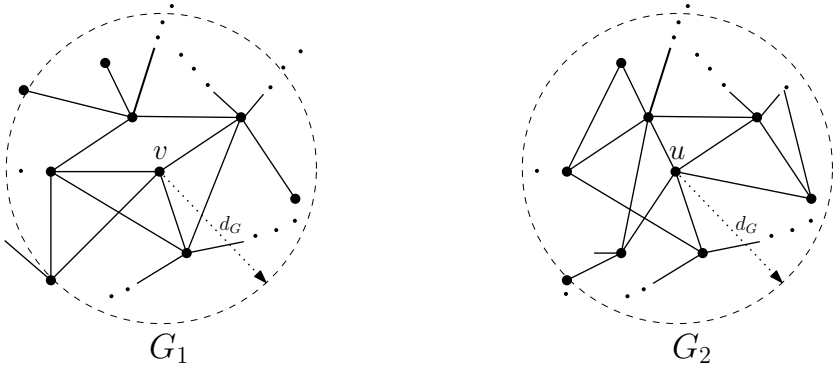


Figure 14: 두 그래프 G_1, G_2를 각각의 정점 v, u에서의 지역적 유사성을 비교 한다. 첫번째 문제는 얼마나 넓은 지역(그림에서 d_G로 표시)을 선택할 것인가이고, 또 다른 문제는 그 기본성분 그래프의 출현빈도를 어떻게 비교하는가이다.

graphlet을 이용하여 전체 네트워크를 bottom-up 상향식으로 분석하는 시스템으로는 GraphCrunch과 GRAAL이 있다. 이 도구는 생물학 그래프나 일반 위상적 그래프에 상관없이 Graphlet에 기반하여 그래프의 유사성을 비교해준다는 면에서 볼 때 범용성이 있다고 할 수있다[15, 10].

그림-15에 나타는 예제 그래프의 각 orbit별 나타난 Graphlet의 갯수를 발췌하여 부분적으로 나타내면 다음 표와 같다. 이 73 원소를 가진 Graphlet vector, $GVD(v)$와 $GVD(u)$ 사이의 공간거리를 이용하면 두 노드의 지역 유사성(local similarity)를 계산할 수 있고, 모든 노드에서 나타나는 Graphlet의 빈도를 표시한 벡터 $GVD(G_a)$, $GVD(G_b)$로 만들어 두 벡터의 거리를 계산하면 이것은 전체 그래프의 유사성으로 사용할 수 있다. graphlet 기반의 GRAAL 시스템은 이후 6장에서 자세히 설명할 예정이다.

orbit	0	1	2	3	4	5	6	7	...	21	26	30	33	53
GDV(v)	5	2	8	2	0	5	0	4	...	2	2	2	4	1

2.5 네트워크 중심성(Centrality) 기반 분석

네트워크의 특성 지표 중에는 나타내는데 중심성(Centrality)도 있다. 즉 네트워크에서 "중심"이 어디에 있는가를 판별하는 것으로서, 이것 자체로 두 네트위크의 동일성을 검사하기에는 크게 부족하지만 네트워크 분석의 초기단계에서 활용될 수 있다. 두

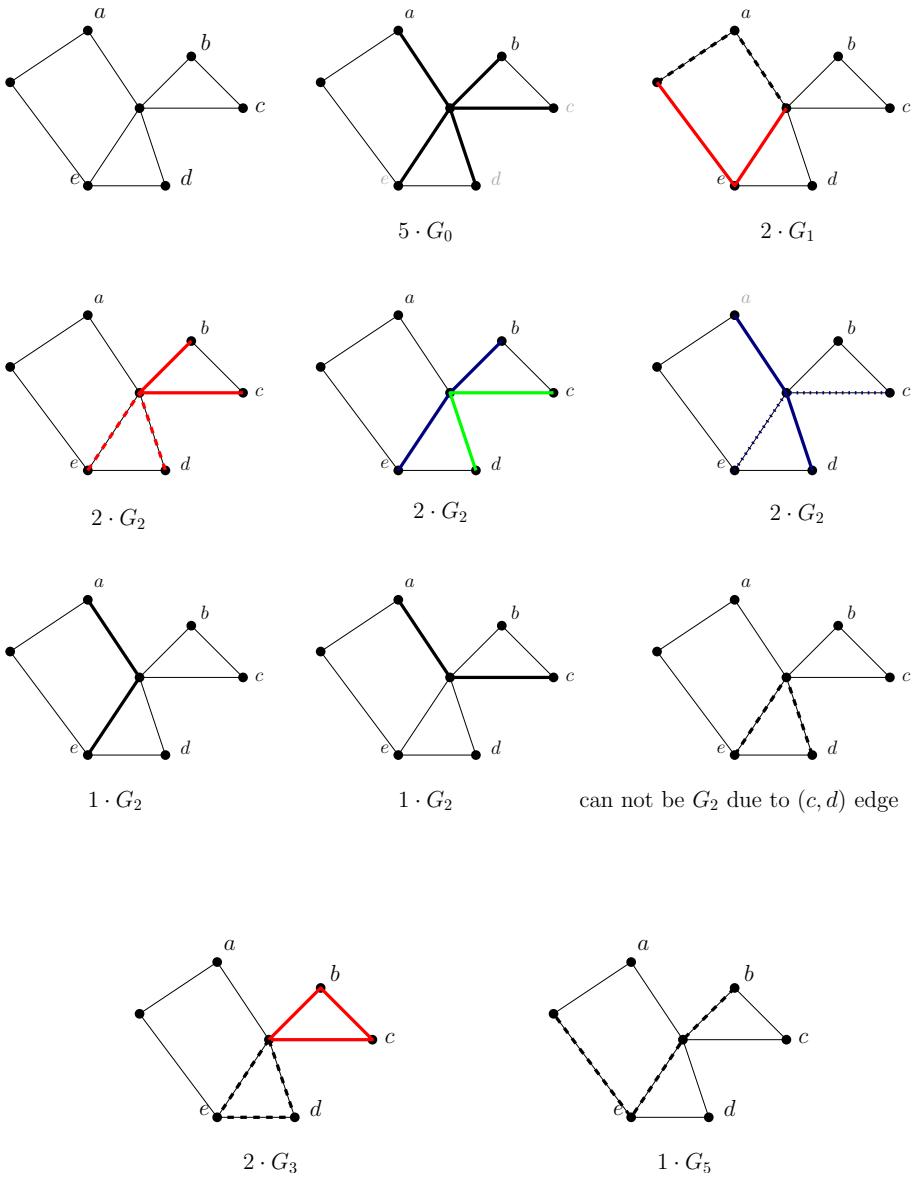


Figure 15: Sample Graph의 노드 v에 포함된 다양한 Graphlet. G_0는 하나의 edge인데 이런 것이 모두 5개 존재한다. G_1은 path의 길이가 2인 것으로 한 끝에 붙어 있는데 vertex induced subgraph중에 1번 orbit을 포함하고있는 graphlet은 모두 2개 뿐임을 알 수 있다. 같은 방식으로 살펴 볼 때 orbit 2번이 v에 접한 경우는 모두 8개임을 알 수 있다.

네트워크의 중심을 먼저 매칭(정렬)시킨 후 나머지를 그리디(greedy)하게 정렬하는 방법을 대부분 정렬 알고리즘이 사용하고 있는데, 이 경우 첫 정렬 시작점을 찾을 작업에 네트워크 중심을 활용할 수 있다. 그런데 이 그래프 중심은 우리가 어떻게 정의하는가에 따라서 달라진다.

가장 단순한 중심을 차수중심(degree centrality)이다. 이것은 전체 노드 중에서 가장 차수가 높은 노드를 중심으로 보는 관점이다. 그 다음 근접중심(closeness centrality)는 다른 전체 노드와의 거리의 합(또는 최대값)이 가장 작은 지점을 찾는 방식으로 결정된다. 이것은 그래프 이론에서 말하는 센터 노드(center node)이다. 물론 center node는 한개인 경우가 대부분 이지만 문제에 따라서는 복수개인 k-center를 찾아야 하는 경우도 있다. scale-free network에서 차수중심 노드와 근접중심 노드는 같아질 가능성이 높지만 일반적인 그래프에서 이 두 중심은 아무런 관계가 없다. 예를 들어 도시(city)로 비유할 때, 도로가 많이 연결된 신흥개발지는 연결차수가 높은 반면 지리적 중심은 아닌 경우가 많다. 유럽 대부분 도시에서 근접중심은 이전 도시가 생길 때 만들어진 구도시(Old town)에 존재한다.

또 다른 중심성은 사이중심(betweenness centrality)으로 그래프에서 임의의 두 지점을 연결하는 최단거리가 얼마나 그 지점을 많이 지나가는지의 정도로 결정된다. 즉 어떤 노드 x를 거의 모든 쌍의 최단 경로 $path(u, v), u, v \in G(V)$가 지난다면 이 지점은 사이중심이 된다. 지리적으로 표현하자면 교통의 중심지가 이 사이중심 노드에 해당될 것이다. 그 외에도 다양한 중심노드에 대한 정의가 있으므로 자세한 내용은 Ref-[9]을 참조하면 될 것이다. 이 네트워크 중심지 비교는 두 네트워크가 유사하다는 것을 밝히는데에는 도움이 크게 되지 않지만 두 네트워크가 상이하다는 것을 빠르게 확인시켜주는데에는 매우 효율적인 지표가 된다. 따라서 네트워크 DB에서 탐색작업을 할 때 candidate space를 줄여주는 효과가 있다. 즉 두 네트워크의 "중심"의 위치나 모양이 크게 다르다면 이 네트워크는 매우 상이한 구조라는 것을 알 수 있기 때문에 고려대상에서 바로 제외시킬 수 있다.

2.6 생물 네트워크 연구의 최근 이슈

최근의 새로운 방법 중에서 인공지능이나 자연어 분석, 자동추론(automated deduction)에 사용되는 belief propagation network 모형을 이용하는 방법이 제안되었다. 일반적인 그래프 매칭 알고리즘의 휴리스틱 버전들이 대부분 local similarity를 이용해서 유사한 지역을 확장해자는 것에 착안하여 이런 과정은 Loopy Belief Propagation(LBP)과 유사한 것에 착안하여 LBP를 수정하여 그래프 매칭에 이용하였다. 또한 이들은 spectral graph

이론과 주성분 분석(Principle Component Analysis, PCA)를 결합하여 부그래프 매칭 문제 풀이법도 제시했다. 그 부산물로서 periodic pattern이나 평균출현 횟수에 비하여 출현빈도가 훨씬 떨어지는 infrequent pattern을 찾아낸다는 점에서 의미를 가진다고 보인다[16].

그러나 필자의 관점으로 볼 때 이런 복잡한 heuristics이 실제 전통적인 graph 이론 기반의 combinatorial 해법보다 더 나은지에 대해서는 회의적이다. 이러한 방법들은 특정 데이터에 대해서 overfitting되기가 쉽고 그 내부 성능을 조정할 수 있는 많은 주절변수가 있기 때문에 만일 처음 수행했을 때 만족할만한 성능이 나온다면 몰라도, 그것을 조정해서 우리가 원하는 쪽으로 성능을 개선한다든지 또는 새로운 기능이나 관찰지표를 추가하는 것은 상당히 어려운 일이라고 생각된다. 참고문헌[16]에서 실험에 사용한 데이터의 크기도 그닥 큰 편이 아니라 이 방법의 현실적 실효성에 대해서는 의문이 남아있다. 하지만 다양한 그래프 알고리즘을 응용해서 그래프 유사도 문제에 접근하려는 시도의 일환으로의 작은 가치는 있다고 하겠다.

향후 기존의 AI나 machine learning 기법을 응용한 네트워크 분석이 앞으로 많이 제시될 것으로 예상된다. 그 이유로는 이전보다 computing power가 매우 높아졌고 다양한 parallel 계산이 가능해졌기 때문에 Support Vector Machine이나 Deep Learning과 같이 이전에는 단순히 이론적인 모형으로만 제시된 알고리즘을 실제 거대 생물네트워크에 적용할 수 있게 되었기 때문이다. 그러나 방법이 아무리 새롭다고 해도 그 성능이 수준에 미치지 못한다면 그 방법만으로의 가치는 떨어진다고 할 것이다. 성능의 불안정성, 사용에서의 불편함, 직관적인 사용자의 간섭이 어려운 것은 AI 기반의 네트워크 정렬 도구들이 공통적으로 가지는 단점이라고 보인다.

앞에서 설명한대로 생물 네트워크가 어떤 그래프 모형을 따르는가를 확인하는 것은 매우 중요한 문제이다. 왜냐하면 그러한 근본적인 가정이 없으면 어떤 과학적인 모형이나 추론이 불가능하기 때문이다. 특히 생물학의 특성상 모든 가능한 실험을 할 수 없기 때문에 네트워크의 특성을 수준별[28]로 확정하기 위한 많은 노력이 생물학 뿐만 아니라 물리학, 전산학에서도 이루지고 있다. 초기에는 자연계 대부분은 네트워크는 Random graph중에서 power-law를 따르는 scale-free graph라는 이론이 대세를 이루었지만 추후

[28] 분자단위인지, 세포단위인지, 기관단위인지 아니면 개체나 생태계 수준의 거대 단위인지를 미리 결정해야 한다.

새로운 실험의 결과는 이런 모형과는 다른 결론을 보여주었다. 가장 대표적인 결과는 PPI는 geometric graph에 훨씬 가깝다는 연구[6]와 같이 생물 네트워크는 우리가 어떤 관점으로 보는가에 따라서 전혀 다른 모형으로 될 수 있다. 따라서 네트워크 자체에 어떤 불변의 특징이 존재하는 것이 아나라는 식의 배반된 결과가 속속 확인되고 있는 상황이다[17, 18, 19]. 앞으로도 이 논란은 최신의 실험기기에 의한 실험결과가 만들어짐에 따라서 계속 논란이 될 것으로 예상된다.

3 생물 네트워크 연구 개요

3.1 생물 네트워크 유사도 연구의 목적

Next Generation Sequencing 기술과 같은 새로운 장비에 의해서 엄청나게 빠른 속도로 새로운 실험결과가 쏟아져 나오고 있다. 그 결과 새로운 생물학적 사실들도 속속 알려지고 있는데 특히 이전의 sequence 연구에서 확장된 생물학적 네트워크 모형이 최근의 주요 연구주제로 떠오르고 있다. 그러나 이러한 생물 네트워크(Biological Network) 연구는 아직도 초보적 수준에 머물러 있다.

어떤 생물학적인 연결 관계(interconnection relation)도 이를 추상화하면 결국은 그래프 이론적 모형으로 나타낼 수 있다. 그래프 이론은 매우 다양한 위상적 관계를 나타내기 아주 적합한 도구이기 때문에 그래프 이론적 모형과 그 알고리즘을 연구하는 것은 생물 네트워크의 특성과 숨어있는 사실을 찾아주는데 중요한 도구가 된다.특히 그러한 상호작용 그래프의 크기가 커짐에 따라서 큰 크기의 그래프를 다루는데 필요한 알고리즘의 성능은 갈수록 중요해지고 있다.

생물 네트워크 모형은 어떤 현상을 다루는가, 그 기본 개체가 무엇인가에 따라서 매우 다양하게 분류될 수 있다. 생물 네트워크를 나타낸 그래프에서 각 노드, 또는 vertex는 특정 유전자(gene), 단백질, metabolite가 될 수 있으며 또한 그들의 결합체로도 될 수 있다. 그리고 그 그래프에서 edge 는 기능적 상호작용(화학적 상호작용)을 나타낼 수 있다. 예를 들어 PPI(protein-protein interaction) 네트워크에서 node는 하나의 단백질이 될 수 있으며 그들 간의 edge (x, y)는 두 단백질 x와 y가 결합(physically bind)할 수 있을 때 주어진다. 만일 세포 안에 존재하는 모든 단백질에 대하여 PPI 네트워크를 실험적으로 구성한다면 그 크기는 엄청나게 커질 수 있다. 특히 그 그래프에서 edge의 개수는 노드수의 제곱으로 커지기 때문에 전체 가능한 모든 단백질 쌍에 대하여 실험하는 것은 불가능하다. 또한 같은 단백질을 node로 구성하더라고 다른 기능, 예를 들어 transcriptional regulation, cell signalling, 유전자들 사이의 기능적 연관(예를 들어 synthetic lethality), metabolism(대사과정), neuronal synaptic connection에 기초하면 우리는 새로운 네트워크를 구성할 수 있다.

대사과정(metabolism)도 그래프로 묘사될 수 있는 대표적인 생물 네트워크이다.

이 대사 네트워크에는 2종류가 존재하는데 한 종류의 metabolic network에서 node는 chemical compound를 나타내고, edge 는 두 화합물이 동일한 화학적 반응에 동시에 관여할 때 설정된다. 즉 이 그래프에서 $C \in (x, y)$ 라는 edge가 의미하는 바는 화합물 x 와 y가 C라는 특정 화학반응에 동원된다는 것을 의미한다. 이와 다른 metabolic network 에서 node는 특정 효소 E_i가 담당하는 각각의 화학반응 r_i을 나타낸다. 이 경우 edge (r_i, r_j)는 두 화학반응 r_i, r_j가 적어도 하나 이상의 반응 화합물, 또는 기질(substrate), 또는 생성물(product)을 공유한다는 것을 의미한다. 또한 신호전달, 유전자 조절(gene regulation), lethal interaction(치사 상호작용)[29]의 동작과정도 그래프로 모형화가 가능하다. 그리고 세포 신호전달 체계는 가장 대표적인 생물학적 통신 네트워크라고 할 수 있다. 이 신호체계는 세포가 스트레스와 같은 외부 영향에 어떻게 대처할 것인지를 결정하는 매우 중요한 네트워크이다. 이 네트워크에서 노드는 bio-molecule (또는 그 복합물)이 되고, edge는 각 개체간의 전달신호의 통로를 나타낸다.

각 유전자들 간의 상호 조절작용을 나타내는 유전자 조절 네트워크(Gene Regulatory Network, GRN)도 대표적인 생물 네트워크이다. 어떤 유전자 a가 일정 수준이상으로 발현되면 그 영향으로 다른 유전자 b 를 up-regulation 시키거나 또는 down-regulation 시키는 역할을 하는데 이것을 directed graph로 모형화 한 것인 GRN이다. 그런데 GRN은 단순한 그래프 모형에 dynamics까지 더해진 셈이 된다. 특정 유전자가 발현되는 정도에 따라서 다른 유전자에게도 영향을 미치고, 그것이 결국 자신에게 다시 feedback 되기 때문에 GRN을 simulation 하는 일은 복잡한 닫힌계에서의 미분방정식을 푸는 문제로 결국 수렴하게 된다. GRN을 이해하고 예측하는 것은 결국 모든 생물현상을 환원적 레벨에서 이해하는 것이 되기 때문에 가장 근원적인 문제가 되고 있지만 실험 데이터의 오차나 오류, 모형의 정교성에서 미진하기 때문에 현실적 실용성은 미진한 편이다. 그러나 부작용이 최소화된 신약개발이나 유전자 치료에 이 GRN은 궁극의 해답을 주기 때문에 GRN의 분석에 많은 투자가 되고 있다.

이상 살펴본 바와 같이 생물학의 다양한 레벨에서 거의 모든 생명현상은 그래프로 묘사될 수 있기 때문에 다양한 종류의 생물 네트워크를 탐구하는 것은 궁극적인 연구주제가 된다. 학문적으로 볼 때에도 네트워크 연구는 계산생물학(computational biology)

[29] Synthetic lethality arises when a combination of mutations in two or more genes leads to cell death, whereas a mutation in only one of these genes does not, and by itself is said to be viable.

나 시스템 생물학(systems biology)의 최종 도달점이 될 것이다.

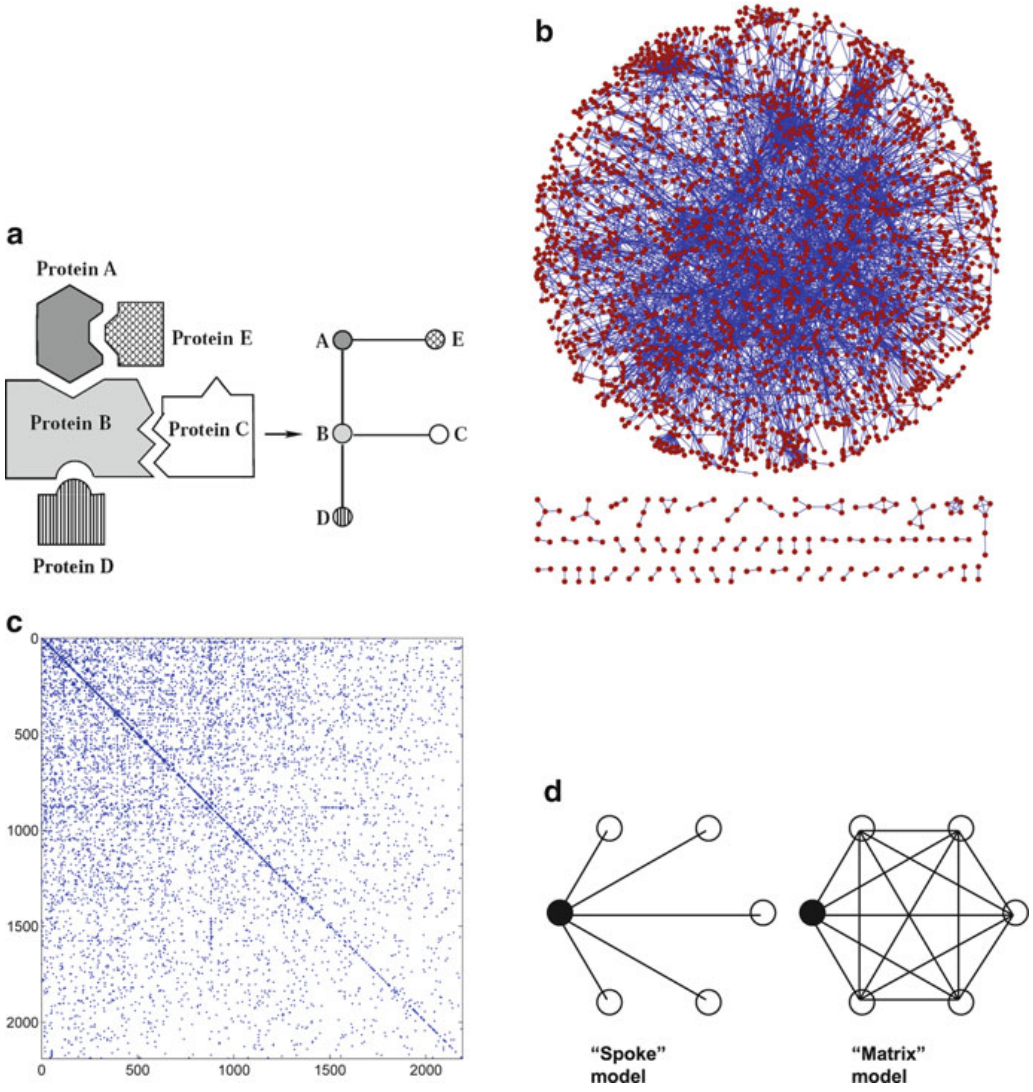


Figure 16: 전형적인 Biological Network의 예 [8]. 그림 a)는 PPI 네트워크의 도식이다. 각각 상호작용하는 단백질들끼리 Edge로 연결된 그림이 Tree형태로 오른쪽에 있다. 그림 b) 는 Baker가 구현한 PPI 네트워크 모형이다. 이 그림은 Database of Interaction Protein (DIP)에서 다운받을 수 있다. 그림 c)는 b) 그래프의 연결관계를 adjacency 관계로 나타낸 것이다. 그림의 각 entry에 있는 점은 두 개 단백질이 상호작용을 함을 나타낸다. 그림에 나타나있듯이 하나의 단백질과 상호작용을 하는 단백질의 갯수는 대략 power-law 분포를 따르고 있음을 알 수 있다. 즉 많은 것은 아주 많고 sparse한 것은 상당히 sparse하다. 이 현상은 PPI network이 전형적인 Complex Network의 일종임을 잘 보여주고 있다. 그림 d)는 상호작용 단백질 관계를 spoke형과 행렬형으로 표시한 것이다.

현대 생물학은 sequence에서 network으로 이동하고 있다고 말할 수 있다. 따라서 충분한 실험과 빠른 실험기구가 만들어낸 네트워크 데이터만 있으면 생물학적 모든 사실이 밝혀질 것인가에 대한 기대가 있을 수 있다. 그러나 그런 희망에는 여러 장애가 있다. 가장 큰 장애요소는 실험자료 안에 수많은 실험 오류와 인간의 힘으로 제어할 수 없는 본원적인 잡음(noisy)의 존재다. 따라서 실제 현상을 그대로 묘사하는 네트워크를 인간이 구할 수 있는 것은 거의 불가능한 일이라고 여겨지므로 실험결과에서 빠진 부분은 실제 실험이 아닌 추론이나 예측으로 메꿔야 한다. 이 때문에 동일한 실험으로 부터 만들어진 생물 네트워크의 분석결과끼리 서로 상반되는 결론에 도달하는 경우도 드물지 않게 나타나고 있다.

생물 네트워크 분석의 또 다른 장애는 앞에서 말한 바와 같이 이용가능성이 높은 그래프 문제는 대부분 NP-complete류의 intractable problem이기 때문에 아무리 계산력이 좋거나 시간이 많아도 최적의 결과를 얻을 수 없다는 것이다. 그러나 한가지 유념해야 할 것은 생물학적 분석에서는 최적의 결과를 얻은 알고리즘을 개발하는데 최종 목적을 두어서는 안된다는 것이다. 이론전산학에서 추구하는 엄밀한 알고리즘보다 유연한 다양한 휴리스틱이 생물 네트워크 연구에서 더 유용하기 때문에 하나의 최적결과 생산에만 매몰되오서는 좋은 시스템을 개발할 수 없다. 특히 본 보고서에서 소개하는 네트워크 정렬(Biological Network Alignment, BNA)의 다양한 알고리즘이 다소 ad hoc하게 보인다고 할지라도 그것을 실험의 유용성면으로 의미가 있으면 충분히 그 가치를 인정받을 수 있다.

3.1.1 생물 모듈의 기능예측을 위한 네트워크 비교

생물 네트워크는 그 대상이 되는 생물체의 종과 구성된 생물학적 단위가 어떤 수준인지에 따라서 달라진다. 예를 들어 최상위인 생태계(ecology) 수준인지 아니면 아주 낮은 분자 단위인지, 아니면 그 중간 상위 단계인 세포 내에서의 신호전달 단계, 또는 유전자간의 상호작용인지에 따라서 구성되는 네트워크는 달라질 수 있다.

만일 하나의 model 생물에 대해서는 우리가 그 내부 작용에 대하여 잘 알고 있다고 하자. 예를 들어 우리는 전형적인 모델생물인 애기장대나 대장균에 대해서의 거의 모든 수준에서의 생물학적 동작을 우리가 잘 알고 있다. 이것이 가능한 이유는 이런

reference model organism에 대해서는 다양한 방법의 실험이 가능하기 때문이다. 즉 특정 유전자의 기능이 어떤 것인지, 만일 그 유전자의 기능을 knock-out 시켰을 때 전체에 어떤 영향을 주는 것인지를 실험으로 확인할 수 있다. 그러나 그 보다 상위 생물체인 포유류의 경우에는 이런 조작적 실험이 불가능하다. 왜냐하면 상위의 고등동물일수록 유전자 수준의 조작은 생물의 생존자체를 불가능하게 만들기 때문에 분자유전학적 조작 실험은 할 수가 없다. 특히 인간의 경우에는 현실적으로도 이런 실험이 불가능하다. 배아 상태에서 유전자조작을 하면 거의 대부분의 사산이나 유산이 되기 때문에 실험을 지속할 수가 없으며, 또한 윤리적으로 인간을 대상체로 삼는 실험은 가능하지도 않고, 이런 일은 법으로 엄격하게 제한을 받고 있기 때문이다. 이런 경우 미지의 생물 단위, 예를들어 인간 유전자의 기능을 알아내려면 어떻게 해야 할 것인지가 현대 생물학의 과제이다. 이 경우 인간 유전자 네트워크를 다른 모델 생물의 유전자 네트워크와 비교해서, 그 연결구조가 비슷한 네트워크를 찾으면 그 mapping되는 구조를 이용해서 기능이 알려지지 않은 유전자의 기능을 짐작할 수 있다. 이 과정을 Projecting Functional Annotation[30]이라고 부른다.

이렇게 두 종의 네트워크를 비교하면 기능상으로 비슷한 역할을 하는 것으로 생각되는 개체를 발견할 수 있다. 이런 관계를 homologous하다고 표현한다. 예를 들어 사람의 유전자 $g_5 6$과 고릴라의 유전자 $G_x 110$이 homologous하다고 표현하는 것은 그것의 자체 서열구조상으로도 유사하고 세포내에서도 비슷한 기능을 하는 유전자라는 것을 의미한다. 특히 homologous한 관계는 서로 다른 종의 단백질 서열에서도 확인이 가능하다. 만일 homologous한 두 종의 단백질이나 유전자가 진화과정에서 공통 조상에서 갈라져 나온 것이라고 믿어지면 두 개체는 서로 orthologous하다고 표현한다. 그리고 또 다른 개념인 paralogous도 중요한 개념이다. paralogous는 하나의 종에서 하나의 유전자가 자체 내에서 복제(duplication)을 통하여 디른 곳에 새롭게 나타난 것을 말한다. 비유를 들어 설명하자면 최초의 실용적 tablet computer를 iPad라고 한다면 이후 이것을 응용해서 만들어진 삼성제품이나 소니 제품은 모두 iPad와 기능적으로 homologous라고 할 수 있을 것다. 그리고 삼성의 갤럭시 제품군에서 서로 크기가 조금씩 다른 제품은 삼성이라는 동일종 내에서의 paralogous관계에 있다고 말할 수 있을 것이다. 분자생물학

[30]분자생물학에서 annotation이란 알려지지 않는 특성을 밝혀서 그에 관련된 정보를 추가하는 작업을 말한다. annotated gene이란 대략의 기능이 알려진 유전자를 말한다. annotation이 안된 유전자란 서열상으로만 보았을 때 유전자로 판단된 DNA seuence를 말한다.

연구, 유전체 연구를 좁혀 표현하라고 한다면 결국 homologous한 gene들을 모든 종에서 찾아내는 작업이라고 할 수 있다.

이미 생물학에서는 각종 종의 유전자들에 대해서 이들간 orthologous한 그룹을 연구한 결과를 제공하고 있다. 그것은 Clusters of Orthologous Group(COG) 라고 불리는 데이터베이스이다[20]. 그리고 이와 유사한 진핵생물체의 orthologous그룹에 대한 대형 DB인 Inparanoid도 이 목적으로 제공되고 있다[21]. 특히 COG DB는 고등동물의 유전학 연구에서 새로운 유전자의 기능을 찾아내는데 매우 중요한 역할을 하고 있다. COG는 서열 탐색도구인 BLAST를 이용해서 특정 단백질(실제로는 유전자의 일부)과 가장 matching이 잘되는 최상의 3개의 서열을 찾아서 이것을 확장시켜 만들어진 DB이다. 그리고 새로운 단백질 서열이 제시되면 그것을 BLAST로 확인하여 이미 구성된 COG DB에서 가장 matching이 높은 3 개의 COG node 를 찾아서 그들과 새로운 연결을 만들어 COG DB에 추가하는 식으로 DB가 확장되고 있다.

3.1.2 생물학적 핵심기능과 보존 지역(Conserved Region)

앞에서도 잠시 설명했지만 생물 시스템은 다양한 하위 모듈로 분해(decompose) 할 수 있는 것이 특징이다. 즉 다단계의 계층구조 개체, 기관(organ), 세포, 단백질등의 하위 기능모듈(functional module)로 분해하여 접근할 수 있다. 세포내 상호작용체(cellular interactome)인 단백질 작용 네트워크(PPI), metabolic pathway, metabolic network, 신호전달 경로(signal transduction network)은 대표적인 세부 네트워크이다. 그런데 우리는 다른 종간의 각종 생물 네트워크가 완전히 다르지 않음을 알고 있다. 예를 들어 사람이나 대부분 유인원류의 유전자 네트워크를 보면 상당히 일치하는 부분이 많음을 볼 수 있다. 그래프 이론적으로 볼 때 두 그래프 또는 복수개의 그래프에서 어떤 특정한 subgraph G_s가 공통적으로 존재한다면 이 subgraph가 대표하는 기능은 매우 중요한 역할을 하고 있음으로 예상할 수 있다. 이렇게 종간의 서로 다른 특성을 무시하고 공통적으로 존재하는 어떤 기능은 그 종들에서 가장 핵심적인 기능을 담당하는 경우가 많다. 어떤 종이 진화과정에서 갈라져 나올 때 가장 중요한 기능은 그대로 전달되어야만 새로 변화된 개체가 생명을 유지할 수 있기 때문에 이렇게 보존되는(Conserved) 기능을 찾는 일은 핵심기능을 찾는 일과 같아진다[10]. 따라서 우리가 관심 가지고 있는 Network Local Alignment가 찾아주는 공통 모듈(Common subgraph)을 찾는 일을 매우 중요한

의미를 가진다고 할 수 있다.

예를 들어 뒤에 소개할 네트워크 정렬도구인 pathBLAST[22]는 이스트(S. Cirevisiae)와 장염의 원인균인 헬리코박터 파이로리(H. phylori)균의 pathway를 상호 비교하여 놀라울 정도로 비슷한 pathway를 발견하는데 성공했다. 두 종의진화적 거리가 가깝지 않음에도 불구하고 이런 공통의 pathway가 보존되고 있다는 사실은 여러 생물학자들의 관심을 끌기에 충분했다. 이를 이용하면 각 종에서 알려지지 않는 유전자의 기능을 추론하는 것은 이전에 비해서 훨씬 용이하게 되었다. 설사 그 추론된 기능이 정확하게 일치하지는 않는다고 하더라고 예측되는 기능이 몇 가지로 정리되면, 이것을 실험으로 확인하는데 드는 비용을 획기적으로 줄일 수 있다. 예를 들어 네트워크 정렬도구인 MaWASh와 NetworkBLAST-M, Graemlin[23]은 이미 분석이 완료된(annotated) 특정 종의 네트워크를 미지의 네트워크와 정렬(alignment)하여 미지의 기능을 추측하는데 유용하게 사용되고 있다.

3.1.3 진화과정(Evolutionary Process) 추론

진화는 유전자 수준에서 볼 때 도태, 변이(mutation), 전달(transfer)의 과정이며 특히 mutation은 종 차원의 진화를 발생하는 가장 중요한 진화 event이다. mutation은 단백질 상호작용 규칙을 변화시키고, 대사 반응(metabolic reaction), 유전자 수준의 반응(genetic reaction)의 변화를 발생시킨다. 이런 변화의 전과정을 추적하는 과정을 진화적 계통분류학 (evolutionary phylogenetics)이라고 할 수 있다. 우리는 수억년 전의 진화과정을 볼 수도 없고 또한 재현할 수도 없기 때문에 그것을 유추하는 방법은 유전자, 또는 DNA, protein 수준에서 일어난 변화를 바탕으로 계통분석(phylogenetics)이라는 수학적 과정으로 이것을 유추한다. 이 작업에서 가장 선행되어야 하는 연구는 두 종간의 단백질 네트워크를 비교하는 것이다. 특히 단백질 네트워크의 정렬(alignment)를 통하여 빠진 모듈, 새로 추가된 모듈, 바뀐 모듈을 찾아내는 것은 진화적 관계를 밝혀주는데 핵심역할을 한다. 가장 낮은 수준에서 단백질의 작은 변화는 구조의 작은 변화를 만들어주고, 구조의 변화는 결국 DNA 서열의 변이(mutation)를 통하여 cellular process의 변화를 만들어준다. 따라서 단백질 구성(서열변화, 구조변화, 상호작용)의 변화를 역추적하면 단계별 진화과정을 유추할 수 있고 이들을 직렬로 쌓아보면 전체적인 진화의 과정을 역으로 구성할 수 있다. 특히 이 작업이 중요한 것은 단순히 단백질 구조의 단위별

변화에 기반한 homologous 연구로는 밝힐 수 없는 모호성을 해결해준다. 예를 들어 p_a가 p_b, p_c와 homologous 값으로는 유사하다고 했을 때 p_a와 반응하는 다른 단백질들과의 그래프적 이웃관계를 이용한 진화분석을 통하면 그 모호성을 해결하는데 큰 도움을 줄 수 있다. 특히 세포사멸(cell death)에 관한 사실은 단순히 서열분석으로는 알아낼 수 없고, 다양한 진화적 과정 분석을 이용해야만 제대로 알 수 있다는 것이 최근 연구에서 속속 밝혀지고 있다. 예를 들어 2007년 Wagner는 어떤 서로 다른 생물학적 모듈간에 높게 연결된 노드들의 진화 속도가 한 모듈 내에서 같은 정도로 연결된 다른 노드들간의 진화 속도보다 훨씬 빠름을 진화분석 과정에서 밝혀냈다[24]. 또한 각 단백질은 개체별 진화를 하는 것이 아니라 어떤 상호작용을 집중적으로 하는 단백질 클러스터(cluster)를 중심으로 동시다발적으로 진화를 한다는 사실이 Yosef과 그의 동료들에에 의해서 새롭게 밝혀졌다[25].

네트워크 비교의 의미는 다른 데에도 잇다. 포유류와 같이 겉모양 확인이 가능한 고등생물들은 진화과정을 모양만으로(phenotype)도 추측이 가능하지만 세균이나 바이러스와 같이 기능적 모양이 의미가 없을 경우에 그들간의 진화적 관계를 밝히는 것은 유전자 수준으로 내려가지 않으면 불가능하다. 예를 들어 대장균의 변종을 보면 인간에게 치명적으로 작용하는 O-157(오 일오칠)등을 포함하여 매우 많은데 이를 구분하여 그들간의 진화과정을 밝히는 일은 DNA수준에서만 분석이 가능하다. 따라서 우리가 살펴볼 네트워크 정렬 및 비교 도구는 이런 극소 생물체들의 진화트리(evolutionary tree), 계통트리(phylogenetic tree)를 구성하는데 필수적으로 활용되고 있다. 요약하자면 생물체들의 진화를 추적하는 것은 현대생물학의 가장 중요한 연구주제인데, 이 과정에 가장 중요한 기법은 각 개체의 생물학적 네트워크를 정렬하고 비교하여 같고 다른 모듈을 찾아내는 것이라고 할 수있다.

3.1.4 네트워크 비교분석을 통한 질병 분석

인간의 질병은 암, 자기면역 메커니즘의 고장, 호르몬 조절 과정의 고장, 선천적 유전적 질환, 외부 감염, 신경생리학적 혼란, 정신질환 등으로 분류될 수 있는데 이들은 모두 유전자의 구조적 결함 또는 대사작용 메카니즘의 결함으로 발생하는 것이다. 그것의 근원은 결국 질병원인성(pathogen)에 의한 감염이나 유전자적 수준에서의 변이(mutation), 손실(missing), 불필요한 확장(extra copying)에 있기 때문에 질병의 원인 분석은 분자

생물학적 수준의 분석, 즉 유전자와 관련된 단백질 상호작용으로 부터 시작해야 한다.

근대 이전에는 질병 원인을 외부 요인을 찾아내려고 했는데 비해서 현대 의학은 이미 공개된 전 유전체(Whole genome), 단백질 네트워크, 단백질체학(Proteomics), 대사적용 결과물[31], 외형상의 변화를 종합해서 찾아낸다. 현대적 관점으로 볼 때 모든 질병의 원인은 유전자 기능의 변화에 의한 것인데, 그것은 결국 단백질의 서열적 변화가 동반하는 상호작용의 변화로 설명될 수 있다. 즉 신체내에서 일어나는 질병현상은 결국 단백잘 상호작용의 결과이다. 예를 들면 대부분의 암은 특정 유전자의 이상적 변이에 의해서 발병하는 것인데, 문제는 변이가 일어나면 다시 이전 상태로 돌아가지 못하는 상황(세포사멸 과정이 암에는 없기 때문에)이 암(cancer)의 치료를 어렵게 한다. 따라서 병의 원인을 분석하는 가장 기초적인 작업은 질병 개체의 단백질 상호작용 네트워크을 정상적인 개체의 그것과 비교하는 것이라고 할 수 있다.

앞서 설명한 것과 같이 미지의 모듈(단백질, 유전자, 세포 구성요소)의 기능을 예측(projection)하는 방법은 이미 알려진 네트워크, 예를 들면 COG나 KEGG의 구조와 그래프 비교를 통하여 알려진 annotation을 이용하는 것이다. 즉 이미 annoatation된 질병의 여러 네트워크을 분석할 질병의 네트워크과 상호비교하면 질병 원인과 관련된 유전자를 찾아낼 수 있다. 즉 일반적인 정상 세포의 모듈기능을 예측하는 것과 동일한 과정으로 문제의 유전자나 관련된 모듈을 찾아낼 수 있다. 또한 계통분석을 통해서도 질병분석이 가능하다. 계통트리(phylogentic tree)는 특히 바이러스 계열의 진화를 규명하는데 유용한데, 개에서 고양이로 옮아갈 수 있는 바이러스가 어떤 것인지 그것이 어떻게 가능한지를 밝히는데 유용하게 활용되고 있다. 그리고 종간 전염이 가능한 바이러스의 일반적인 메카니즘을 연구하면 이전 문제가 된 조류독감이나 구제역 바이러스가 사람에게 옮기는 과정을 탐구하는데에도 매우 유용하게 활용될 수 있다. 이외에도 계통분석(phylogenetics)는 종간의 질병이동 연구에 매우 중요한 역할을 하게 될 것이다.

정상적인 세포를 질병 세포로 바꾸는 병원성(pathogen) 요인과 그 과정을 이해하는 것도 질병연구의 중요한 주제중 하나 인데 여기에도 네트워크 비교는 중요하다. 예를 들어 일년에 100만명 이상의 사망자를 발생시키는 열대성 말라리아 기생충(P. faliciparum)을 방지하는 것은 매우 중요한 전지구적 보건문제이다. 문제는 왜 열대성 말라리아

[31] 일반적인 종합병원 임상병리과에서 검사하는 대부분의 검사결과 소변검사, 피검사 등이 여기에 해당한다.

기생충이 각종 새로운 약에 빠르게 면역성을 보이는가를 규명하는 것이었다. 그런데 말라리아 기생충 스스로가 사람의 몸 안에서 살아 남아야하기 때문에 인간의 정상적인 pathway를 방해하지 않는 선에서 기생충 스스로의 기능 pathway를 따로 운영해야만 지금과 같은 상황이 된다. 따라서 이 기생충의 대사작용 pathway와 인간의 pathway를 비교하여 그 차이를 규명할 수 있다면 이 기생충의 어떤 기능을 공격해야 면역성을 가진 기생충까지 퇴치시킬 수 있는지 알수 있고, 그로부터 기생충의 면역기능을 회피하여 공격하는 궁극적인 약을 만들 수 있다. 즉 인간과 말라리아 기생충에 공통으로 보존되는 pathway을 명백히 찾아낼 수 있으면 말라리아 치료제에 새로운 장을 열어줄 것인데, 여기에서도 가장 중요한 기능은 두 종간의 생물 네트워크를 비교하는 것이라 하겠다.

실제 이러한 접근을 시도한 연구자가 있다. Suthram[26] 은 말라리아 병원충의 네트워크와 이스트, 꼬마선충(C. elegans), 초파리(D. megalogaster), H.pylori의 네트워크를 pathBLAST를 이용하여 분석하였다. 그 결과에 의하면 그 병원충과 효모 사이에는 3개의 공통 네트워크가 존재함을 확인할 수 있었는데 비해서, 다른 개체인 선충, 초파리와는 공유하고 있는 subnetwork이 하나도 없음을 밝혔다. 그런데 다른 3 개체인 꼬마선충, 효모, 초파리 간에는 공통적으로 존재하는 네트워크가 매우 많음을 발견했다. 이 연구가 기여한 바는 다음과 같다. 말라리아 기생충인 P.faliciparum 병원충의 모델 생물로는 초파리나 파일로리균, 꼬마선충보다는 효모가 더 적절하다는 것이다. 이 결과는 2005년 네이처에 발표되었다[26]. 이 결과가 제약업계에 의미하는 바는 매우 심대하였을 것이다. 말라리아 치료제로 고려할 수 있는 성분은 매우 많은데 이 모두를 시험약으로 만들어서 임상실험을 하기에는 엄청난 재원이 들지만, 그 가능성을 모델 생물인 효모를 통해서 실험을 한다면 좀 더 빠른 시간에, 좀 더 가능성이 높은 말라리아 병원충을 제어할 수 있는 약효 성분을 찾아낼 수 있다는 것이다. 그것이 Suthram의 네트워크 비교연구가 현실에 기여한 부분이다.

또 다른 생물 네트워크 비교분석의 활용 사례가 있다. 조절효소는 세포내 대사작용에서 가장 중요한 역할을 하는데 그 중에서 인산화반응(phosphorylation)은 대사조절 작용에서 가장 중요한 과정이다. 그런데 DNA에서 인산화 반응 진화과정에서 강력하게 보존되는 단백질 서열과는 달리 매우 쉽게 변하는 특징이 있다. 따라서 대사과정이 빠진 서열분석만으로 이 지역을 찾아내는 것은 매우 어려운 일이었다. 그런데 이것을 인간, 이스트, 선충, 초파리 등 종간의 네트워크 비교를 통하여 6292개의 사이트를 24000여개의

인산화 사이트 중에서 찾아내는데 성공했다[27]. 그 중에서 479 개의 위치는 다른 비교종과 공유하고 있음을 밝혔다. 그런데 인산화 사이트는 위치적으로 불안하기 때문에 인산화 반응 지역이 진화적으로 보존되는지의 여부를 서열 수준에서는 알아낼 수 없다. 따라서 Hung은 키나제 기질(kinase substrate)[32] 지역을 종간 네트워크 정렬방법을 활용해서 conserved 인산화 사이트를 추론하는데 성공했다. 결론적으로 이 방법을 통해서 인간 유전자 중에서 인산-단백질을 코딩하는 유전자와 암관련 유전자 중에 많은 것이 서로 겹친다는 새로운 사실을 밝혀내는데 성공했다.

3.2 이종(hetrogeneous) 생물 네트워크 간의 유사성 비교

일반적으로 생물 네트워크 연구는 유사한 기능의 네트워크들기리의 비교연구가 주를 이루었다. 예를 들어 사람과 초파리의 소화관련 네트워크를 비교한다든지, 이스트와 대장균의 유전자 네트워크를 비교한다든지이다. 그런데 서로 다른 범주간 비교연구도 시도되고 있다. Wu[28]의 연구결과는 독특한데 유전병의 외적증상을 그래프로 만든 네트워크와 인간 단백질 반응 네트워크의 구조적 유사성을 비교한 것이다. 즉 인간질병의 표현체(Phenome)[33]와 유전형(phenotypic)[34] 네트워크의 공통점을 찾아냈다. 그 결과 유사한 증상(표현형)들과 관계된 단백질들이 밀집된 형태로 PPI subnetwork로 존재함을 확인할 수 있었다. 이 말은 어떤 유전자는 하나가 아닌 여러 질병에 공통적으로 작용하여 유사한 증상을 일으킨다고 볼 수 있다. 즉 우리가 질병 이름을 붙일 때에는 각각 편의대로 붙이지만(역사적으로 볼 때 병이 알려지는 순서대로 적절히 붙인다.) 그들이 유전적 수준에서 얼마나 공통성이 있는지에 대해서는 별 관심을 가지지 않았다. 따라서 이 연구가 잘 정리되면 한 질병의 치료제가 다른 질병의 치료제로 될 가능성을 타진하는 효율적인 방법이 될 수도 있을 것으로 보인다. 예를 들어 전립선 비대증 치료제가 발모 치료제로 우연한 기회에 알려졌는데, 이런 상황은 Wu[28] 등이 제시한 연구 방법론이 실제적으로 충분히 활용될 수 있는 가능성을 보인 것이라 판단된다.

이질 네트워크에 대한 또 다른 재미있는 연구가 있다. Goh[29] 등은 3개의 이질적인

[32] 키나제는 인산화 반응은 촉매제가 된다

[33] 표현형, phenotype을 나타내는 기본 개체 유전적 특성이 외부로 관찰되는 현상, 예를 들어 키 피부색, 눈동자 색상, 머리카락의 곱슬함 등은 표현체의 일종이다.

[34] 유전자 수준에서 구분되는 개체. 유전형이 다르다고 항상 그것이 표현형으로 드러나는 것은 아니다. 반대로 표현형으로 다른 것이 항상 유전형으로도 다르다고는 할 수 없다. 즉 다른 표현형에 대응하는 유전형을 우리가 항상 확인할 수는 없다.

네트워크, 즉 질병유전자 네트워크 N_g, 질병(질환) 네트워크 N_d, 그리고 마지막으로는 질병체인 디지좀(diseasome) 네트워크 $N_{d,g}$을 구성했다. 질병유전자 네트워크의 노드는 유전자 g_i이며 두 노드를 연결하는 edge (g_i, g_j)는 두 유전자가 하나의 질환(disorder) d_k에 공통으로 관여할 때 주어진다. 그 다음 N_d의 node 는 각 질병 d_i가 되며 (d_i, d_j)는 서로 다른 질환 또는 질병 d_i, d_j에 관련된 유전자 중에 공통된 유전자가 존재할 때 주어진다. 그래프 이론으로 본다면 이것은 N_g의 line graph $L(N_g)$[35]가 된다. 그리고 $N_{d,g}$의 vertex는 질환과 질환 유전자 모두가 표현된다. 그리고 edge 는 특정 질환 d_s와 그 원인 유전자 g_s를 이용해서 하나의 에지, (d_s, g_s)로 생성된다. 이 3개 그래프의 비교결과 서열수준으로 유사한 질환 유전자가 병 증세로도 유사한 질환을 일으킴을 확인할 수 있음을 확인할 수 있었다[29]. 앞으로 이러한 이질적인 생물 네트워크간의 비교연구는 상당히 흥미로운 결과를 줄 수 있을 것이다.

3.3 다양한 네트워크 정렬방법론의 분류(Taxonomy)

정렬이란 어떤 같은 범주의 서로 다른 개체들간 mapping을 통하여 주어진 목적함수를 최대 또는 최소로 만드는 최적화 문제의 일종이다. 이를 위한 정렬 알고리즘(alignment)은 정렬할 대상이 무엇인가에 따라서 선형정렬, 그래프 정렬, 행렬 정렬로 구분된다. 선형정렬은 DNA, RNA, protein sequence와 같이 선형적으로 저장된 개체를 정렬하는 것이며 그래프 정렬은 본 보고서에서 다루는 것과 같이 일반적인 위상구조를 가진 그래프끼리 서로 유사성을 찾아내는 작업이다. 그리고 많이 연구가 되고 있지는 않지만 크기가 서로 다른 행렬끼리 정렬하는 문제도 있지만 이 문제는 그 복잡도 조차 잘 알려져있기 않기 때문에 중요한 연구는 아직 이루어지지 못하고 있다. 물론 선형 개체와 일반 그래프 개체의 중간 복잡도를 가진 트리(Tree)에 대한 정렬문제도 생각해 볼 수 있다.

정렬은 정렬 알고리즘의 최종 결과물의 성격에 따라 지역 정렬(Local alignment)와 전역 정렬(Global alignment)로 나누어져 있다. 이 둘의 중간 쯤 되는 Semi-global 정렬도 있다. 지역 정렬은 두 개체간 가장 비슷한, 즉 목적함수 또는 유사도 평가함수값이 가장 높은 **일부분**을 찾아내는 것이다. 이에 반해서 전역 정렬은 개체의 모든 원소를

[35] 어떤 그래프 G의 line graph $L(g)$는 다음과 같이 구성된다. G의 모든 vertex v 는 $L(g)$의 edge가 되고 edge는 $L(G)$의 vertex가 된다. G의 edge가 서로 연결된 경우, 즉 edge (x, y)와 (x, z)이 두 edge는 x에서 만나므로 이 edge의 $L(G)$에서는 edge 를 가진다. 일종의 dual construction이다.

정렬에 포함시켜 전체적으로 가장 유사한 형상(configuration)을 발견하는 과정이다. 이 둘은 사용목적에 따라서 적절하게 선택해야 한다. 만일 두 DNA sequence에서 보존지역(conserved region)이 무엇인지 알고싶다면 지역정렬을 해야 할 것이고, 두 DNA 서열이 어떻게 진화, 변화하였는지를 알고싶다면 전역 정렬을 사용해야 한다. 전역정렬은 두 개체의 유전적 거리가 멀다면 별 의미없는 결과를 보여준다. 예를 들어 사람과 그와 유전적으로 가까운 침팬지의 특정 염색체를 비교하는 것은 전역정렬을 통해서 가까운 진화적 거리를 확인할 수 있다. 이 전역정렬은 사람과 침팬지의 염색체에서 사라지고 새로 추가된 DNA 부분, 또는 그러한 유전자를 확인할 수 있게 해준다. 이와 다르게 만일 사람과 특정 식물의 염색체를 전역정렬로 비교한다면 서로 상이한 부분이 너무 많아서 그 결과가 우리에게 주는 의미는 크지 않을 것이다. 이 경우에 지역정렬을 사용하면 사람과 식물이 생물체로서 가지고 있는 보존 구역 또는 필수 유전자의 대략적인 모양을 찾아낼 수 있기 때문에 의미가 있다.

다음으로는, 정렬할 개체의 갯수에 따라서 쌍정렬(pairwise alignment)와 다중정렬(multiple alignment)로 나뉜다. multiple alignment의 좋은 예는 영장류인 사람, 침팬지, 오랑우탄, 보노보, 고릴자 5종의 유전체에서 공통적으로 존재하는 어떤 서열을 찾는 문제다. 보통 pairwise alignment와 달리 multiple alignment는 그 갯수에 따라서 공간 시간 복잡도가 급격히 증가하는 특징을 가지고 있다. 예를 들어 길이가 각각 $n_1, n_2, \ldots n_k$인 k종의 서열을 다중정렬을 한다면 그 시간 복잡도는 $O(n_1 n_2 \ldots n_k)$가 되고 만일 거의 같은 N 길이를 가진 서열 kr 개를 다중정렬로 분석하려면 그 공간, 시간 복잡도는 $O(N^k)$가 되어 $k > 50$ 이상만 되면 거의 현실에서는 풀 수 없는 문제가 되고 있는 것이 현실이다. 따라서 다양한 휴리스틱이 동원될 수 밖에 없고 현실도 그러하다.

4 선형 서열 정렬문제

이 장에서는 일반적인 위상구조를 가진 그래프 정렬문제를 살펴보기 전에 우리는 그보다 단순한 1차원 구조인 서열정렬의 문제를 먼저 살펴보고자 한다. 다차원 네트워크 정렬 문제의 다양한 변형이나 해결 방법론은 1차 서열 정렬문제를 확장하면 쉽게 이해할 수 있다.

4.1 선형서열 (Linear Sequence) 정렬과 비용함수

길이가 다른 두 서열 aactggca 와 gaaagaaca을 정렬 (alignment) 한다는 것은 두 서열에 gap 기호 '-'를 포함해서 길이가 같도록 만드는 것이다. 가장 간단한 방법은 이것을 주어진 순서대로 정렬하고 갈이를 같게 만들기 위해서 두번째 서열의 마지막에 gap 기호를 붙이는 것으로 다음과 같다.

```
aactggca-
gaaagaaca
```

그런데 정렬의 목적이 유사한 것을 확인하려는 것이므로, 정렬하는 과정에서 가능하면 column별로 matching의 갯수가 많도록, 즉 높은 정렬값을 가지도록 조작하고자 한다. 즉 다음과 같이 만들면 둘 사이의 유사한 부분을 좀 더 쉽게 확인할 수 있다.

```
-aactgg--ca-
gaa-tggccgaa
*++++++**#+*
```

위의 도표에서 '+'는 matching, '*'는 gap, '#'는 mismatch의 위치를 각각 나타낸다. 만일 match, mismatch, gap의 비용값을 각각 +1, -1, -2 이라고 한다면 위 전역 정렬에 만들어주는 결과값은 $6(+1)+6(-1)+1(-2)$ 이므로 그 최종 정렬값은 -2가 된다. 문제는 이 두 서열을 적절하게 정렬해서 가장 높은 값이 나오도록 짜맞추는 것이 정렬문제의 최종 목적이다. 가장 결과가 trivial한 정렬 중 하나는 양쪽을 모두 gap으로 채운 다음 각각을 그대로 앞 뒤로 배치하는 것으로 아래와 같다.

```
--------aactggca
gaaagaaca-------
```

정렬에서 가장 중요한 것은 정렬된 상태를 평가하는 비용함수이다. 비용함수에 따라

서 정렬의 최적모양이 달라질 수 있기 때문이다. 그리고 원하는 결과를 보기 위해서는 적절한 또는 올바른 비용함수를 사용하는 것이 매우 중요하다. 일반적인 서열 정렬에서 같은 위치에 같은 DNA base가 있을 것, 즉 일치하는 것을 선호하기 때문에 두 DNA base가 같으면 match로 판단하여 "+" 점수를 준다. 만일 정렬된 상황에서 같은 칼럼(column)에 서로 다른 base 가 있는 것은 원하는 결과가 아니기 때문에 이것은 mismatch로 판단하여 음수값을 지정한다. 보통 음수값은 이런 현상을 방지한다는 뜻에서 penalty cost, 또는 벌칙함수라고도 부른다. 그리고 gap을 사용하는 것도 원하는 일치 관계가 아니기 때문에 이것에도 음수값을 지정한다. match에 얼마나 큰 양수(match) 값을 줄 것인가, mismatch나 gap에 얼마나 큰 penalty를 줄 것인지는 응용분야나 저렬의 목적에 따라서 달라질 수 있다. 일반적인 비용행렬은 아래와 같다. 만일 gap을 더 억제하고 싶다면 gap penalty 를 더 올리면 된다.

	a	g	t	c	-
a	+1	-1	-1	-1	-2
g	-1	+1	-1	-1	-2
t	-1	-1	+1	-1	-2
c	-1	-1	-1	+1	-2
-	-2	-2	-2	-2	N

4.1.1 서열(Sequence)의 전역정렬과 지역정렬

이 장에서는 앞서 설명한 local, global alignment를 비교해 본다. 우리는 다음과 같은 2개의 서열을 가지고 있다.

S_a=attgaaacagatgaca, S_b=aggtcaacagatggtt

위 두 서열을 적절하게 배치하여 높은 정렬 값이 나오도록 전역정렬(global alignment)을 수행하면 다음과 같이 만들 수 있다.

```
---attgaa-acagatttgaca
gccattgaatacagat--gact
```

이것은 모든 base가 정렬에 반드시 포함되어야 하는 global alignment의 한 예이다. 즉 전체 DNA base가 모두 alignment에 들어가 있으면 alignment score를 계산할 때 각

column별 점수를 순서대로 구하면 된다. 이와 다르게 local alignment는 정렬된 일부분만 점수계산에 들어간다. 지역정렬을 아래 두 서열 S_x, S_y를 이용해서 설명해보도록 하자.

S_x=ataatagacgtgttgcgcgaaatc, S_y=cgctatggtaccatgaagagagttt

이 두 서열에서 우리가 보고 싶은 것은 전체의 유사성이 아니라 두 서열에서 가장 유사한 부분이 어떤 것인가이다. 이것을 local align하면 아래와 같이 별표로 표시된 부분은 비록 일치하지 않더라도 penalty값을 받지 않는다. 따라서 우리가 선택한 부분에서만 alignment score 를 구하면 된다.

```
**aa-tagacg**********
**********aag-aga-g***
```

양쪽 끝쪽의 alignment에 포함되지 않는 부분을 모두 제외하고 실제 align된 모양만을 나열하면 다음과 같다.

```
aa-tagacg
aag-aga-g
```

이 둘, local alignment와 global alignmet와는 조금 다른 semi-global alignment도 있다. 이것은 하나의 짧은 sequence S_1을 그 보다 훨씬 긴 다른 target sequence S_T 에 매핑할 때 주로 사용한다. semi-global alignment이 앞서 설명한 alignment인 global이나 local과 다른 것은 시작하는 gap sequence나 끝나는 trailing gap sequence는 penalty에 포함시키지 않는 특성이 있다. 단 S_1은 반드시 모두 alignment에 포함되어야 하는 것이다. S_1의 일부분만을 local alignment와 같이 사용하면 안된다. 예를 들어 보자. query sequence S_1와 S_T가 다음과 같다고 할 때 semi-global aligment는 다음과 같이 나타난다.

$$S_1 = \texttt{tgtgaat}, S_T = \texttt{tcgccgtgcgacattcccgctga}$$

```
tcgccgtgcgacattcccgctga
------tgtga-at---------
```

이 경우 아래 쪽의 왼쪽에서 시작하는 6개의 gap과 끝나는(trailing) 9개의 gap은 penalty에 포함하지 않으므로 최종 alignment값은 아래와 같이 계산된다.

```
tgcgacat
tgtga-at
```

따라서 이 semi-global alignment의 값은 6개의 match와 1개의 mismatch, 그리고 한 개의 gap으로 계산되어 $6 - 1 - 2 = +3$이 된다.

4.1.2 정렬 비용함수의 일반화 모형

앞에서 설명한 모든 alignment score 계산방법은 가장 보편적이며 dynamic programming으로 계산이 쉬운 독립된 column별 통합 방식이다. 즉 정렬된 두 서열의 각 칼럼별 점수(페널티 포함)를 옆 칼럼의 값과 상관없이 모두 더하는 방식이다. 이 columnwise addictive 방식을 사용하면 가장 단순한 dynamic programming 기법으로 $O(mn)$ 공간과 시간에 최적의 alignment 결과를 구할 수 있다. (단 m과 n은 비교할 두 서열의 길이다.) local alignment는 좀 더 복잡한 조건을 사용해야 하는데 global alignment 와 크게 다르지 않다. semi-alignment까지도 공간, 시간을 $O(mn)$을 사용해서 해결할 수 있다.

그런데 실제 분자생물학에서 이런 column별 점수계산법을 별로 적합하지 않다고 한다. 즉 실제 optimal alignment를 할 때 gap의 갯수에 비례해서 penalty 점수가 주어지는데, 진화유전학적으로 볼 때 진화거리는 gap과 비례하지 않는다는 것이다. 즉 "----------"이 나타날 가능성과 분리된 "----" 4개 짜리가 다른 위치에서 3번 나타날 가능성은 서로 다르다는 것이다. DNA가 개체별로 유전될 때, 또는 gene이 진화과정에서 변화할 때, 다른 서열이 유전자 속으로 우연히 들어오거나 유전자의 일부 서열이 빠져나갈 수가 있는데 이때 한번에 k 길이의 fragment가 빠져나갈 확률과 $2k$ 길이의 그렇게 될 확률은 길이에 반비례하는 식으로 계산되는 1/2이 아니라고 한다. 따라서 gap이 10개인 penalty는 gap이 2개의 penalty의 5배가 아니라 2개의 gap 보다는 조금 높은 정도라는 것이다. 따라서 gap이 발생할 때의 penalty는 단순히 비례가 아니라 약하게 비례하도록 점수를 계산하는 것이 실제 생물서열의 변화확률과 부합된다는 것이다. 그 약한 비례식을 식으로 표현하면 다음과 같다.

$$penalty(k) = C_0 + C_1 \cdot k$$

여기서 C_0는 open gap penalty로서 gap이 시작되면 주어지는 기본 penalty이다. 그리고 C_1은 1보다 작은 real number가 된다. 만일 C_0=2 이고 C_1=0.1이라고 할 때 gap이 2개인 페날티와 gap이 10개인 페날티를 계산해보면 $penalty(2) = 2.2$ 가 되고 $penalty(10) = 3.0$이 되어 2배가 되지 않는다. 이런 penalty 방식을 affine gap penalty 방식이라고 하고 실제 DNA sequence 정렬에서 많이 사용되고 있다.

즉 두 개의 서열 attagtccgca와 atca를 align할 때 짧게 끊어진 gap이 더 많이 배치된 오른쪽이 더 낮은 정렬값을 반는다. 즉 그렇게 될 가능성이 왼쪽의 경우, 즉 DNA서열 한 군데가 열려서 한 쪽이 연결된 하나의 뭉치 단위로 빠져나가든지 새로 들어오는 경우가 오른쪽과 같이 짧게 여러 군데에서 각각 open될 가능성보다 높다는 것이다.

```
attagtccgca
at-------ca
```

,

```
attagtccgca
---a-t-c--a
```

단백질의 경우에는 20개 아미노산 끼리의 진화적 거리에 따른 PAM(Point Accepted Mutation)과 같은 비용행렬이 존재한다. 그 비용행렬은 비교할 단백질을 가진 개체가 유전적으로 가까운가(예를 들어 인간과 유인원) 또는 멀리 떨어져 있는먼가에 따라서 다르게 사용된다. 예를 들면 인간과 침팬지 단백질 서열을 비교할 때 사용하는 PAM을 원숭이와 물고기 비교에 사용해서는 안된다. 그 경우에는 다른 비용 행렬을 사용해야 한다.

보통의 경우 사용자가 이 비용행렬의 entry값을 적절히 조정해서 원하는 정렬이 나오도록 조정해야 하지만 일반적인 생물연구자들의 경우 보통 default로 세팅된 행렬값을 사용한다. 물론 행렬원소의 값은 항상 정수일 필요는 없다. 필요에 따라서는 floating point로도 얼마든지 설정할 수 있다. 그리고 대부분의 서열정렬 프로그램은 비용함수를 사용자가 설정하도록 열려있다.

그런데 실제 전장 유전체(Whole genome)과 같은 100메가 이상의 길이를 가진 서열은 $O(mn)$의 보통 alignment 알고리즘을 사용해서 수행할 수 없다. 왜냐하면 100 mega × 100 mega 만큼의 공간을 확보하여 수행하기란 거의 모든 시스템에서 불가능하기 때문에 다른 Heuristics를 사용해야 한다. 바로 이런 문제를 위하여 개발된 시스템이 일번 연구자가 서열 탐색에서 가장 많이 사용하는 BLAST: Basic Local Alignment Search Tool 이다. 이 도구는 따로 다운 받아서 local server에 설치해서 사용할 수도 있고, 아니면 NCBI에서 제공해주는 web server를 통해서 온라인으로도 실행해볼 수도 있다.

BLAST는 잘 알려진 DNA, RNA, protein linear sequence를 align하는 도구로서 가장 많이 활용되는 시스템이다. 원래 두 개의 선형 sequence를 비교하는 방법은 Smith-Waterman 방식의 $O(Mn)$ 시간의 dynamic programming이 있지만, 이 방법을 길이가 100 mega이상의 DNA(주로 format은 FASTA)에 적용할 수 없기 때문에 긴 서열의 경우에는 local alignment의 heuristics인 BLAST(Basic Local Alignment Search Tool)`http://www.ncbi.nlm.nih.gov/blast/`를 사용한다.

4.1.3 동적계획법 기반의 서열정렬 알고리즘

앞서 설명한 두 정렬, 지역정렬과 전역정렬은 2차원 동적계획법 (dynamic programming)을 해결된다. 동적계획법이란 어떤 문제의 전단계 모든 부분해(partial solution)를 이용해서 바로 그 다음 단계의 해를 순차적으로 구성하는 알고리즘 구성 기법중의 하나이다. 길이가 n, m인 두 개의 서열이 다음과 같이 표현된다고 하자.

$$V = v_1 v_2 \dots v_n, W = w_1 w_2 \dots w_m$$

일단 두 서열의 최장공통서열(Longest Common Subsequence, LCS)을 구하는 문제부터 살펴보자. LCS란 두 서열의 모든 부서열(subsequence)중에서 가장 긴 서열을 말한다. 주의해야할 사항은 substring이 아니라 subsequence라는 것이다. substring은 연속되어야 한다는 특성이 있지만 subsequence는 그 순서만 유지하면 되므로 선택되는 문자가 건너뛰는 범위에는 제한이 없다. 두 서열이 유사하면 유사할수록 LCS는 길어지게 될 것이다. 예를 들어 보자.

두 서열 A=`attgcgaaga`와 B=`catggcgtaggaaa`의 공통서열은 매우 많다. 예를 A에서 선택된 `a--t-cg--ga=atcgga`는 하나의 공통서열이다. 이것은 B에도 다음과 같이 존재한다. B=`-at--cg---g--a`와 일치한다. 따라서 부서열 중에서 가장 긴 것이 무엇인가 찾아내는 것이 LCS문제를 푸는 것이다. LCS도 두 유전체 서열, 또는 일반적인 서열의 유사도를 짐작하는데 활용되고 있다.

먼저 $s_{i,j}$를 두 서열의 접미서열 $V_i = v_1 v_2 \dots v_i$, $W_j = w_1 w_2 \dots w_j$ 의 LCS를 나타낸다고 하자. 가장 쉽게 알 수 있는 boundary condition은 다음과 같다. 아래에서 입실론ϵ은 길이가 0인 가상의 dummy 문자열을 나타낸다.

$$s_{1,\epsilon} = 0, s_{\epsilon,1} = 0$$

먼저 $s_{1,1}$은 쉽게 계산할 수 있다. 만일 $v_1 = w_1$이면 $s_{1,1} = 1$이고 아닌 모든 경우는 당연히 $s_{1,1} = 0$이다. 이제 이 과정을 이용해서 recursion 식을 만들어 본다. 우리는 $0 \leq p \leq i$, $0 \leq q \leq j$ 에 대하여 $s_{i,j}$를 제외한 모든 p, q 값에 대해서 $s_{p,q}$를 알고 있다고 가정한다. 이제는 이 부분해를 종합하여 $s_{i,j}$를 구해야 한다. 문자열의 제일 마지막에 문자 v_i와 w_j가 어떻게 매칭되는지를 생각해보자. 만일 LCS의 답이 존재한다면 그 경우는 아래 3가지 경우 중 반드시 하나의 경우에 해당된다.

Case1) v_i만 $s_{i,j}$의 LCS에 포함되는 경우. 즉 s_i가 W이 어떤 원소와 매칭이 되어 LCS에 들어가는 경우. Case 2) w_j만 LCS에 들어가는 경우. case 3) v_i와 w_j 모두가 LCS $s_{i,j}$에 들어가는 경우. case 4) v_i와 w_j 모두 LCS에 들어가지 않는 경우. 따라서 우리가 구하려는 것은 LCS이므로 이 4가지 경우의 부분해 중에서 가장 큰 값만을 고르면 된다. 이것을 재귀식으로 기술하면 다음과 같음을 알 수 있다.

$$s_{i,j} = \max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1, \text{if} v_i = w_j \end{cases}$$

전역정렬은 앞서 설명한 LCS 문제에서 mismatch와 gap등의 penalty 값만 아래와 같이 설정해주면 된다. 따라서 정렬할 서열의 두 문자 x, y의 비용함수가 $\delta(x, y)$라고 한다면 두 서열의 전역정렬값 $G_{i,j}$을 구하는 식은 다음과 같이 기술된다.

$$G_{i,j} = \max \begin{cases} G_{i-1,j} + \delta(v_i, -) \\ G_{i,j-1} + \delta(-, w_j) \\ G_{i-1,j-1} + \delta(v_i, w_j) \end{cases}$$

위의 식에서 $\delta(v_i, -)$은 당연히 음수가 될 것이고 $\delta(v_i, w_j)$에서 두 문자가 같으면 match이므로 양수가 된다. 만일 불일치하면 이것을 금지해야하므로 $\delta(v_i, w_j)$는 음수값이 되어야 한다. 물론 이 방식은 각 칼럼별 유사점수를 모두 독립적으로 더하는 방법일

때의 dynamic programming이다. 위의 식이 설명하고 있듯이 이 정렬은 전역정렬로 모든 원소가 정렬에 포함되어 음수 또는 양수의 값을 받는다. 이 문제를 지역정렬 문제로 생각해보자.

전역정렬에서 두 서열의 원소가 모두 다르다면, 즉 어떤 subsequence도 같은 것이 없는, 예를 들어 `aaggaaggaa`와 `ttctccttcctt`이라면 이 둘의 정렬값은 음수가 될 수 밖에 없다. 그러나 이것을 지역정렬로 맞추어 본다면 우리는 가장 유사한 부분이 없는 것을 선택하면 되기 때문에 음수값의 정렬은 항상 피할 수 있다. 이 사실이 의미하는 것은 위 전역정렬 제귀식에서 조금만 수정하면 지역정렬 계산식을 구할 수 있다는 것을 의미한다. 그것은 바로 최대값 max을 구할 때 항상 0을 넣어서 음수값이 되는 것을 막아주면 된다. 따라서 두 서열의 지역정렬값 $L_{i,j}$ 계산은 다음과 같이 표현된다.

$$L_{i,j} = \max \begin{cases} 0 \\ L_{i-1,j} + \delta(v_i, -) \\ L_{i,j-1} + \delta(-, w_j) \\ L_{i-1,j-1} + \delta(v_i, w_j) \end{cases}$$

만일 다른 복잡한 affine gap penalty를 사용한다면 매우 복잡한 dynamic programming 식을 사용해야 할 것이다. 다양한 일반적인 정렬(general alignment)의 유사도 계산방식에 대해서는 이 책을 참고하면 좋다. 이 문헌은 alignment 알고리즘의 다양한 변형에 대하여 가장 잘 기술한 참고문헌이 될 것이다[30]. 좀 더 이론적인 부분은 Pezner의 책을 참고하기 바란다. 이 책을 활용하면 정렬에 관한 보다 다양한 이론을 접할 수 있을 것이다[31].

4.1.4 서열 정렬문제의 알고리즘 관련 이슈

앞의 재귀식들이 의미하듯이 가장 일반적인 정렬 알고리즘은 두 서열의 길이가 n, m일 때 공간도 $O(nm)$ 이 필요하다. 따라서 당연히 이 공간의 모든 cell에 한번 이상의 계산으로 해야하므로 그 시간복잡도 역시 $O(nm)$보다 작을 수는 없다. 그런데 실제 사용하는 생물 유전정보는 수식 수백 mega 베이스 크기이기 때문에 최적을 답을 보장하는 일반적인 quadratic algorirthm은 현실에서 사용할 수 없다.

공간을 좀 교묘하게 사용하면 $O(n+m)$의 시간과 공간에 서열값만을 알아낼 수는

있다. 즉 위의 식에서 현재까지 계산되어 다음 단계로 넘어가기 위하여 필요한 최소한의 $s_{i,j}$ 값만을 지키고 있으면 되므로 $O(n+m)$의 시간과 공간 최적정렬값을 알아낼 수 있다. 그런데 이런 방법은 두 서열의 정렬값만 알려줄 뿐이다. 실제 서열 정렬은 그 값도 중요하지만 그 정렬된 모양을 보는 것이 중요하므로 이것을 알기 위해서는 $O(nm)$ 테이블을 backword로 찾아 가면서 최적의 값을 만드는데 기여한 각각의 match, mismatch, gap insertion을 찾아야 하므로 $O(n+m)$ 시간에 그 정렬된 서열의 형상(configuration)을 만들어 낼 수는 없다.

이후에 교묘한 Divide and Conquer 방식을 이용하여 시간은 $O(nm\log(n+m)$을 사용하지만 공간은 $O(n+m)$ 안에 최적의 지역정렬, 전역정렬의 모양까지를 찾아내는 알고리즘이 만들어졌지만 실용성은 그다지 높지 않다. 대신 많은 생물학자들이 사용하는 방법은 BLAST라는 hesuristics이다. 이 heuristics는 이후 설명할 graph alignment에도 원용되기 때문에 살펴볼 가치가 있다. 이 차선의 방법은 간단하다. 전체의 서열을 한번에 비교하는 것이 아니라 두 서열의 일부분 중 아주 유사한 일부분 substring[36]을 hashing 등으로 먼저 찾는다. 그리고 그 다음에 이 "확실한 지역"을 중심으로 유사한 지역을 양쪽으로 확장하는 그리디 방식이 바로 대안으로 쓰이고 있다.

아래 그림-17는 BLAST 도구가 수행되는 과정을 도식적으로 보여주는 것인데 색이 칠해진 작은 수평 박스가 anchor를 나타낸다. anchor는 완전히 일치 또는 거의 일치하는 짧은 DNA seqgment 인데 보통 10 base에서 50 base 정도 길이의 substring을 사용한다. 이 anchor는 hashing등을 통해서 쉽게 찾아낼 수 있다. 예를 들어 길이가 16인 모든 $16-mer$의 갯수는 4^{16}개가 존재하므로 두 서열에서 일치하는 $16-mer$는 hashing이나 BLAT등의 도구를 이용해서 쉽게 찾아낼 수 있다. 그런 다음에 이 anchor를 중심으로 양쪽의 새로운 string 영역을 살펴가면서 최대로 일치하는 부분지역을 확장해 나가는 작업을 수행한다. 보통 이 작업을 "anchor를 중심으로 양쪽으로 밀어나간다" 라고는 표현한다. 이런 방식이 BLAST-류 정렬 알고리즘의 근간을 이루고 있고 이 방법은 네트워크 정렬에서도 그대로 활용된다.

이렇게 하면 완전 일치된 anchor를 중심으로 비교하는 heuristics를 사용하면 alignment 없이도 유수한 부분 서열을 찾아낼 수 있지만 문제는 아래 그림과 같이 이런 anchor가

[36]이것은 anchor 라고 부른다. 배를 정박시킬 때 쓰이는 닻의 역할과 유사하다고 생각하면 된다.

많을 때 발생한다. 만일 두 개의 전유전체(Whole genome)을 비교할 때 이런 유사한 anchor가 두 유전체에서 각각 10만개, 8만개가 나왔다면 이 모든 쌍에 대해서 위의 작업을 수행해야하므로 그 시간은 엄청나게 걸릴 것이다. 즉 하나의 matched anchor를 밀어가면서 비교하는 것은 별로 어렵지 않으나 이 모든 쌍을 다 살펴보는 것은 상당한 시간이 걸린다. 이 작업을 줄이기 위해서 Suffix tree를 이용하여 빠르게 approximate searching을 하는 Bowtie와 같은 도구가 활용된다. 실제 작업에서는 그 matched anchor pair중에서 일부분, 약 10%만 선택하여 전체 시간을 단축시키기도 편법도 사용된다. 선택되는 비율은 BLAST나 Bowtie 사용자가 조절할 수 있다.

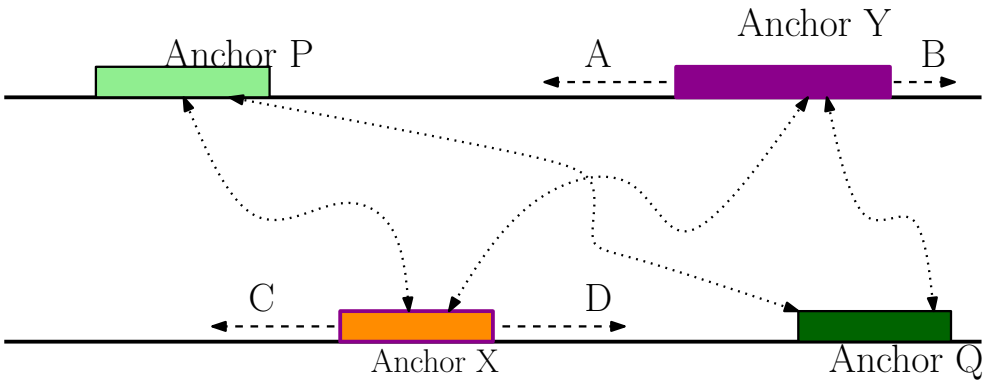


Figure 17: BLAST에서 지역정렬을 찾아는 휴리스틱 방법론. 먼저 거의 완전히 일치하는 짧은 anchor substring을 hashing등을 이용해서 두 대상서열에서 찾는다. 그 다음 그 짝지워진 쌍을 중심으로 양쪽으로 유사한 지역을 끝날때까지 그 영역을 확장시켜 펴나간다. 만일 그 전개과정에서 불일치 정도가 일정이상이면 이 확대작업을 중단하고 그 지역을 local alignment결과로 보고한다.

5 네트워크 정렬문제

이 장에서는 생물학적으로 상이한 복수개의 네트워크를 어떻게 비교하는지, 또한 그 비교가 생물학적으로 어떤 의미를 가지는지, 그런 비교 작업이 함의하고 있는 알고리즘적 문제는 어떤 것인지에 대하여 설명하고자 한다.

5.1 네트워크 정렬의 생물학적 의미

실험 기술의 발달에 의해서 새로운 생물 네트워크가 빠른 속도로 생산되고 있다. 예를 들어 전사 네트워크(transcription network,), 단백질 상호작용 네트워크, 보조 조절 네트워크(co-regulation network), signal transduction network, metabolic network은 대표적인 생물 네트워크이다. 생물 네트워크를 그래프 정렬로 분석하는 중요한 목적은 그 안에서 생물학적인 motif를 찾아내는 일이다. 그리고 종간(across species) 간 생물 네트워크를 비교함으로서 상호 보존된 구역이나 종끼리 서로 다른 기능으로 분화된 지역을 정확하게 발견하기 위한 목적도 있다.

네트워크 정렬 방법에는 크게 통계적 도구나 수식을 이용하는 방법, 또는 그래프 알고리즘을 이용해서 구체적으로 매핑되는 생물 모듈단위를 찾아내는 방법이 있다. 특히 생물학적 motif를 찾아내는 일이 네트워크 비교의 가장 중요한 목적 중의 하나이다. 그것은 motif는 생물체(분자수준)에서 정보처리의 빌딩블록(building block)으로 환원될 수 있기 때문이다. 그런데 생물 네트워크에는 매끈한 수학적 그래프와 달리 많은 잡음이 내재하고 있기 때문에 그대로 비교하기는 어렵고 상당한 오류를 감안한 방식으로 비교해야 한다는 점이 수학적 그래프를 다루는 방법과의 차이점이다. 이런 오류를 포함하는 그래프에서의 매칭 문제는 Error Tolerant Graph Matching(ETGM) 문제로 정의된다. 이 ETGM 문제는 그래프 편집거리를 다루는 장에서 다시 설명할 예정이다.

서로 다른 네트워크를 정렬하는 목적은 기능을 이미 잘 알고 있는 네트워크를 이용해서 잘 모르는 기능의 생물 모듈의 특성을 예측하기 위해서이다. 이것을 구조매핑을 이용한 Network 유사성 분석이라고 한다. 또한 이 유추 방법론은 data mining의 가장 기본적인 방법론이기도 하다. 이런 방법론은 Guilty by Association[37]이라고도 부르는데, 유사한

[37]어떤 미지의 개체 X가 이미 성격이 잘 알려진 이웃 개체들과 "가깝다면", X는 그 이웃개체들과 비슷한 속성을 가지고 있을 것이라는 가정

구조는 그 속성까지 유사할 것이라는 자연계의 기본적인 특성에 기초하고 있다. 아래 그림-18은 그 과정을 잘 보여주는 도식이다.

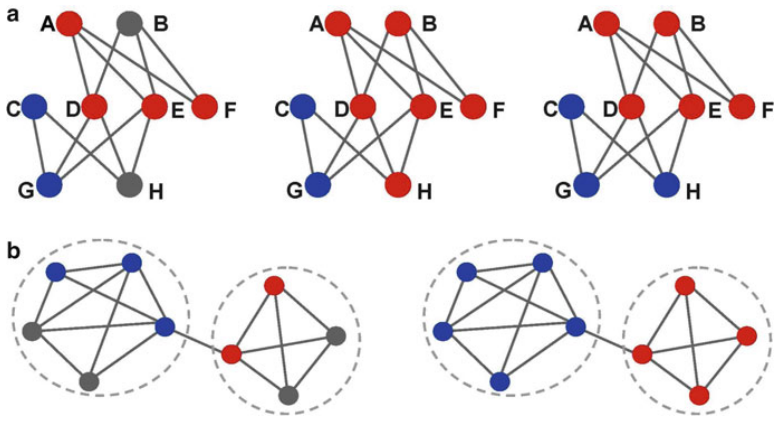


Figure 18: 기능이 알려져있지 않은 노드의 특성을 이미 기능이 알려진 이웃 노드를 활용하여 추론하는 guilty by association 과정. 아래 그램은 군집으로 나눈 다음 같은 군집에 있는 노드들은 같은 기능을 할 것이라는 믿음으로 추론하는 과정.

그림-18에서 붉은색 노드와 파란색 노드는 이미 기능이 알려진 노드를 나타낸다. 그리고 회색으로 표시된 노드는 기능이 알려져있지 않은 노드를 말한다. 편의상 그 기능을 R(ed), B(lue) 라고 하자. 위에 제시된 그래프 (a) 는 다른 그래프와 정렬과정을 통하여 matching이 된 상황을 보여준다. 우리는 그림-(a) 에서 회색 노드인 { B, H }의 기능을 Guilty by Association rule로 예측하고자 한다. 일단 이웃노드의 특성으로부터 유추한다면 가운데 그래프에서 B는 3개의 Red 노드와 이웃하고 있으므로 그 majority는 Red가 되어 B의 기능은 Red로 추측한다. H의 경우에는 Red 이웃이 2개, Blue 이웃이 1개 이므로 이 역시 과반의 이웃은 Red이므로 H역시 Red로 annotation이 된다. 그런데 다른 추론모형을 쓰면 H의 기능은 다르게 유추될 수 있다.

Shared Neighbor Model은 어떤 미지의 노드 x의 기능을 x의 이웃노드 집합 $N(x)$와 같은 이웃을 가진 노드의 기능으로 유추하는 것이다. 예를 들어 $N(H) = \{C, D, E\}$ 인데 같은 노드를 이웃으로 가지는 노드는 G으므로 G와 같은 기능을 할 것이라고 보는 것이다. 즉 $N(G) = N(H) = \{C, D, E\}$이므로 $function(H) \approx function(G)$로 annotation 한다. 이와 유사하게 Cluster 단위로 개체를 구분해서 각 군집별로 미지의 function을 예측하는 방법도 있다.

그림-18(b)가 그 경우인데, 이 네트워크의 노드들을 2개의 cluster로 구분되어 있다(그림에서 점선으로 둘러 쌓인 부분). 이 구분의 기준은 2-connected[38] component이다. 따라서 같은 2-connected component에 속한 미지의 노드(회색노드 4개)들은 각각 자신이 속한 cluster의 기능을 상속받게 되므로 B와 H는 모두 Red로 분류된다. (왜냐하면 둘 모두 Red component에 들어있기 때문이다.) 이외에도 Network flow를 이용하여 미지의 기능을 예측하는 방법도 있다[32]. 이와 같이 association Rule 방법론은 우리가 annotation을 사용하는 domain의 특성에 따라서 잘 선택해서 사용해야 한나. 한가지 방법만을 고정해서 사용하지 말고 정답을 이미 알고 있는 benchmark data를 사용해서 각 방법론을 신중하게 검증해야 한다. 즉 기능이 알려진 몇 개 node의 기능을 의도적으로 지운 뒤에 제시한 추론기법으로 그 기능을 가장 제대로 찾아주는 지를 확인하여 그 기능이 가능 뛰어난 방법을 선택해야 한다. Guilty of association 방법은 domain data에 따라서 성능이 결정되는 것이지 방법 그 자체의 우열로 결정되는 것은 아니다.

이와 같이 전체 유사도가 아닌 특정 노드와 그 이웃 노드들의 유사성에 기초하여 그래프의 전체 유사도를 계산하는 방법과 그것을 실제 그래프 매핑에 응용하는 새로운 방법과 결과가 Zager등의 의해서 잘 정리되어 있다[33].

5.2 그래프 편집거리(edit distance)과 유사도

두 그래프를 비교하고 그 차이를 판별하는 방법 중에서 가장 기본적이고 전통적인 접근법은 그래프의 편집거리(edit distance)를 구하는 것이다. 이 방법은 그래프 이론의 가장 기초적인 내용이다. 다차원 개체인 그래프의 편집거리는 이미 잘 알려진 문자열의 편집거리의 확장판으로 생각하면 된다.

두 그래프의 편집거리(edit distance) $edist(G_1, G_2)$ 는 그래프 G_1을 다른 그래프 G_2로 변환할 때 필요한 최소의 동작횟수를 나타낸다[39]. 즉 그래프 편집은 아래 6개의 기본동작(basic operation)만을 최소한의 횟수로 이용해서 한 그래프를 다른 그래프로 바꾸는 과정이다. 그래프 편집의 기본모형에서 허용되는 기본 동작은 아래 6개 뿐이다.

- 새로운 vertex의 추가(insert), 그 비용은 c_{vi}

[38]어떤 그래프를 분리시키기 위해서 반드시 2개 이상의 vertex를 제거해야만 하는 그래프의 특성, 연결도
[39]단 이 경우 입력 그래프는 모두 vertex edge labelled 그래프이다.

- 새로운 edge의 추가, 그 비용은 c_{ei}
- 특정 vertex의 삭제(delete), 그 비용은 c_{vd}
- 특정 edge의 추가, 비용은 c_{ei}
- 특정 vertex의 레이블을 다른 것으로 변경(substitute), 그 비용은 c_{vs}
- 특정 edge의 레이블을 다른 것으로 변경(substitute), 그 비용은 c_{es}

이 6개의 비용은 하나의 벡터 $C_{etfm} = (c_{vi}, c_{ei}, c_{vd}, c_{ei}, c_{vs}, c_{es})$로 표시된다.

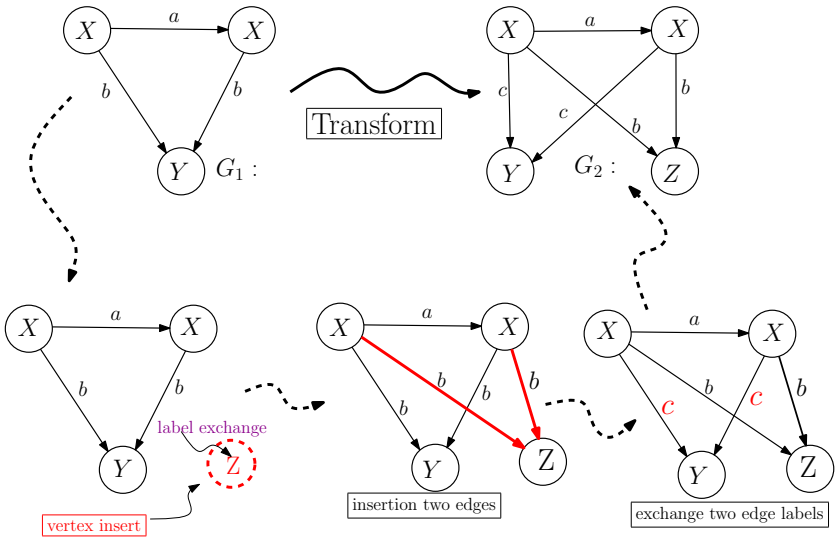


Figure 19: Vertex, edge labelled인 digraph G_1을 허용되는 편집동작만을 이용해서 G_2로 바꾸는 과정. 위의 예라면 새로운 vertex를 추가한 뒤에 새로운 edge를 추가하고 서로 다른 edge의 label을 바꾸면 완성된다. 각 단계별 작업 비용이 1이라고 한다면 이 작업의 편집거리 $gedit(G_1, G_2) = 6$임을 알 수 있다. 그림에서 붉은 색으로 표시된 부분이 이전의 상태의 그래프에서 새로 추가된 그래프 편집동작의 결과를 나타낸다.

그런데 위 그림-19에서 보여준 6번의 동작 이하로 이 작업, 즉 G_1에서 G_2로 변환할 수 있다. 아래 그림-20는 그 과정(4번의 작업)을 보여준다.

그래프 유사도 탐색에서 가장 의미있는 결과는 그래프 거리함수 중에서 metric 성질을 만족하는 함수의 발견이다. 어떤 거리함수 $d()$가 metric 성질을 가지려면 3 개의 조건을 만족해야 하는데 그것은 $d(x, y) = d(y, x)$, $d(x, x) = 0$, $d(x, y) \leq d(x, z) + d(z, y)$, 이 3 개의 조건이다. 많은 함수들이 앞의 2조건은 쉽게 만족시키지만, 3번째 조건인 triangle

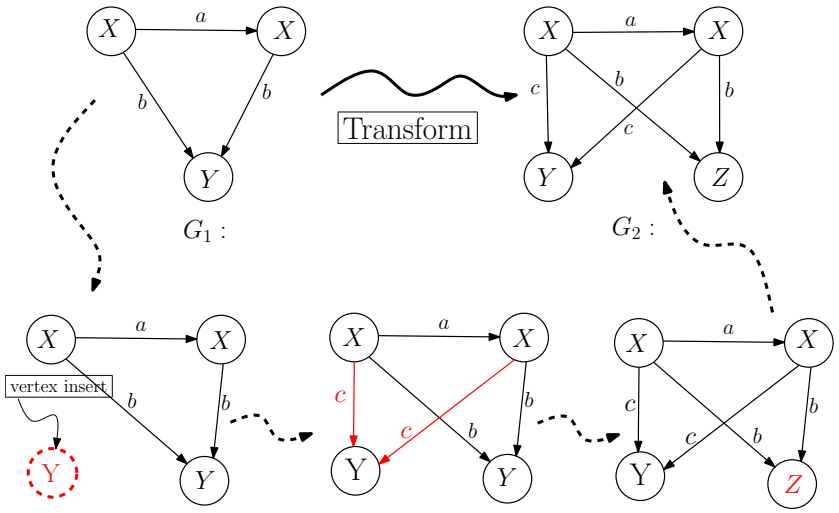


Figure 20: 위에서 표시한 그림-19의 변환을 4번의 편집동작으로 완성한 예. 이 4번은 G_1을 G_2로 바꾸는 최소의 작업횟수이다. 즉 4번이하의 작업으로 G_2를 만드는 방법은 없다. 따라서 $edist(G_1, G_2) = 4$가 된다. 붉은 색으로 표시된 부분은 이전 상태에서 변화된 부분만을 나타낸다.

inequality는 잘 만족시키지 못하고 있다. 그런데 1998년 Bunke와 Shearer에 의해서 그래프 거리함수 중 다음 거리함수가 metric 성질을 만족함이 밝혀졌다[34]. 이 결과는 그래프 매칭에 관한 가장 중요한 성과로 인식되고 있다. Bunke가 제시한 그래프간 거리 공식은 다음과 같다.

$$dist(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{\max\{|G_1|, |G_2|\}}$$

여기에서 $mcs(G_1, G_2)$는 두 그래프에 존재하는 공통 최대 부그래프(maximum common subgraph)를 나타낸다. 즉 $mcs(G_1, G_2) = R$ 이라면 $R \subseteq G_1$, $R \subseteq G_2$이면서 다음의 식 $|R| < |S|$을 만족하는 $mcs(G_1, G_2) = S$가 없다는 것을 말한다. 즉 $mcs(G_1, G_2)$는 두 그래프 G_1, G_2에 공통적으로 존재하는 부그래프 중에서 가장 큰(vertex의 수) 부분 그래프를 말한다. 어떤 거리함수가 metric 성질을 만족하게 되면 개체 탐색에서 매우 유용하게 활용된다. 특히 triangle inequality 조건은 search space를 감소시키는 결정적인 역할을 한다. [40]

[40] 2차원 유클리디안 거리는 대표적인 메트릭 공간이다. 따라서 2차원 공간의 한 지점 x에서 다른 지금

그래프 편집거리를 좀 더 범위를 넓혀서 확장하면 오차허용 그래프 매칭(error tolerant graph matching, etfm) 문제가 된다. 이 문제에 대하여 가장 많은 연구결과를 가진 연구자는 H.Bunke인데 앞서 설명한 $mcs()$ 거리와 edit distance metric과의 연관성에 관한 연구가 대표적인 결과이다[35].

etfm문제는 궁극적으로는 graph edit distance의 모형과 동일하다. 그런데 이 경우 우리가 각 동작의 cost 값을 어떻게 결정하는가에 따라서 최적의 매칭을 달라질 수 있다. 위 그림-19을 활용하여 다시 설명해보고 한다.

그림-21은 G_1과 G_2에 대한 3가지 매칭 방법을 보여준다. 각 그래프 매핑 비용은 앞서 말한 $C_{etfm} = (c_{vi}, c_{ei}, c_{vd}, c_{ei}, c_{vs}, c_{es})$의 값을 각각 해당되는 동작의 횟수만큼 누적으로 더하는 것이다. $C_{etfm} = (1,1,1,1,1,1)$이라고 가정해보자. 이 경우 첫번째 mapping 1은 노드의 레이블 교체 1회, 노드 추가 1회, 두개의 edge 추가 작업을 했으므로 이 mapping 1의 거리값은 4가 된다. 같은 식으로 계산해보면 2번째 매핑의 값은 5가 됨을 알 수 있다. 그리고 3번째 매핑은 노드 한 개의 삭제. 2개 에지의 추가, 두 노드의 추가 그리고 4개 에지의 추가가 필요하므로 etfm 거리값은 9가 된다. 따라서 $C_{etfm} = (1,1,1,1,1,1)$에서 최적의 그래프 매핑은 mapping 1이 된다. 그런데 다른 비용함수 $C_{etfm} = (1,1,3,1,1,1)$을 쓰면 각 매핑의 비용은 달라진다. 이 경우에 각 매핑 1,2,3의 거리값은 각각 6, 5, 9가 되어, 이 경우에는 2번째 mapping이 최적의 매핑(매칭)이 된다. 만일 좀 극단적인 $C_{etfm} = (1,1,1,1,1,7)$을 사용한다면 mapping 3이 최적으로 선택될 것이다. 따라서 최적의 매칭이나 alignment는 universial하게 존재하는 것이 아니고 비용함수에 따라서 결정된다는 것을 항상 명심해야 한다[41] 이 문제는 결국 inverse parametric problem으로 제시된다. inverse parametric problem 문제는 일반적인 매칭과 같이 비용함수를 주고 가장 최적인 그래프 매칭을 찾는 것이 아니라, 그 과정을 거꾸로 하는 것이다. 즉 어떤 매칭결과를 입력으로 주고 이 매칭이 최적의 매칭이라고 판단해주는 cost function을 찾는 문제가 바로 inverse parametric problem이다[42]

---

y로 갈 때 제 3의 지점 z를 거쳐서 가는 것은 항상 거리상으로 손해임을 알고 있기 때문에 우리는 가능한 직선으로 두 점간을 이동하려고 한다. 만일 이런 성질이 2차원 공간에서 만족되지 않는다고 하면 지구상 공간에서의 이동 문제는 대혼란을 겪게 될 것이다.

[41] 이것이 매우 중요한 사실임에도 불구하고 default cost function만을 습관적으로 사용하는 초급 연구자들은 확실하게 이 변화를 인지하지 못하고 있다.

[42] 이 문제는 이론생물학에서 매우 흥미로운 문제임에도 불구하고 명확한 결과가 나온 것이 별로 없다. Biological Network Mining이 본격화되면 결국 모든 정렬문제는 이 inverse parametric 문제로 수렴하게 될 것이다. 우수 연구원 선발문제로 이 inverse parametric 문제를 설명해보자. 일반적으로 우수 연구원

5.3 그래프 매칭 최적(Optimal) 알고리즘

앞서 설명한대로 두 그래프의 매핑문제는 아무런 label이 없을 때에도 NP-complete 이므로 오류를 감안한 각종 그래프 매칭문제는 당연히 NP-complete문제이다. 따라서 최적의 다항시간 알고리즘이 존재하지 않으므로 연구자들은 다양한 휴리스틱 알고리즘을 연구한다. NP-complete류에서 최적의 답을 구하는 전형적인 접근법인 exhaustive search 류의 Branch and Bound 기반의 A^*-알고리즘이 그래프 매칭의 대안으로 오랫동안 사용되었다. 이 방법도 Worst case에는 $O(n!)$의 계산을 피할 수는 없지만 작은 갯수, 대략 30개 안쪽으로 그래프에서 이 방식은 유효하다고 알려져 있다. 그 이후 probabilistic algorithm, neural network 기반의 heuristics, genetic algorithm[36] 등이 제시되었지만 그 어떤 것도 확실한 우위를 차지하고 있지는 못한 상황이다.[43] 이후 실제 수행시간을 빠르게 하기 위하여 preprocessing 작업에 관한 알고리즘이 제시되었다[37]. 이들이 선택한 접근법은 주어진 그래프를 작은 크기의 모듈 subgraph로 쪼갠뒤 이들의 연결관계에 따라 새로운 meta-graph를 만드는 것이다. 일종의 Divide and Conquer, 또는 계층적 level of detail 접근법이라고 할 수 있다. 이 자료구조를 바탕으로 decision tree를 만들면 isomorphic mapping의 가능성이 없은 국부지역을 빠르게 고려대상에서 배제할 수 있기 때문에 전체 매칭 시간을 단축시킬 수 있다. 그러나 이 작업의 수행시간은 빨라도 전처리 작업에는 exponential time을 피할 수 없는 단점은 그대로 남아있다. 따라서 1회성 그래프 비교가 아니라 어떤 graph DB가 있고 질문 그래프 G_q와 일치하는 그래프가 DB에 저장된 그래프 g_i로 존재하는지를 검사하는 DB 탐색모형에서는 유용하게 쓰일 수 있다. 왜냐하면 이미 DB속에서 들어갈 때 한번씩의 preprocessing만을 거치면 되기 때문에 그 시간이 비록 많이 걸릴지라도 전체적으로 보면 효율적이기 때문이다.

그래프 이론적 관점에서 본 그래프 매칭 알고리즘과 수행시간에 대한 분석은 참고문헌 [38]에 자세히 나와있다. 이 문제에 대하여 간단하게 정리된 버전은 참고문헌[39]에

선발은 알려진 parameter에 의해서 평가한다. 예를 들어 논문, 과제, 특허의 점수로 연구원을 정렬하여 그 상위자를 우수연구원으로 결정하는 것이 보통의 방법이다. 그런데 문제는 이 paper, project, patient의 측도에 신뢰성이 과연 있는가 하는 의문을 가질 수 있다. 거꾸로 전원 peer-review(예를 들면 상호평가나 투표)를 통하여 먼저 우수사원을 결정했다고 하자. 이 때 가장 우수한 사원의 3가지 지표로 부터 우리는 새로운 평가를 만들 수 있다. 즉 그 지표로 평가를 하면 해당 사원이 가장 우수하기 때문에 그 지표를 가지고 다른 사원을 평가하는 방법을 고려해 볼 수 있다. 즉 주어진 measure 함수로부터 data를 선별하는 것이 아니라 이미 선별된 data로부터 거꾸로 measure function을 생성하는 것이다. 이런 류의 계산을 모두 inverse parametric problem이라고 한다.

[43] 이들 방법은 대부분 1980년대 초에 제시된 것들이다.

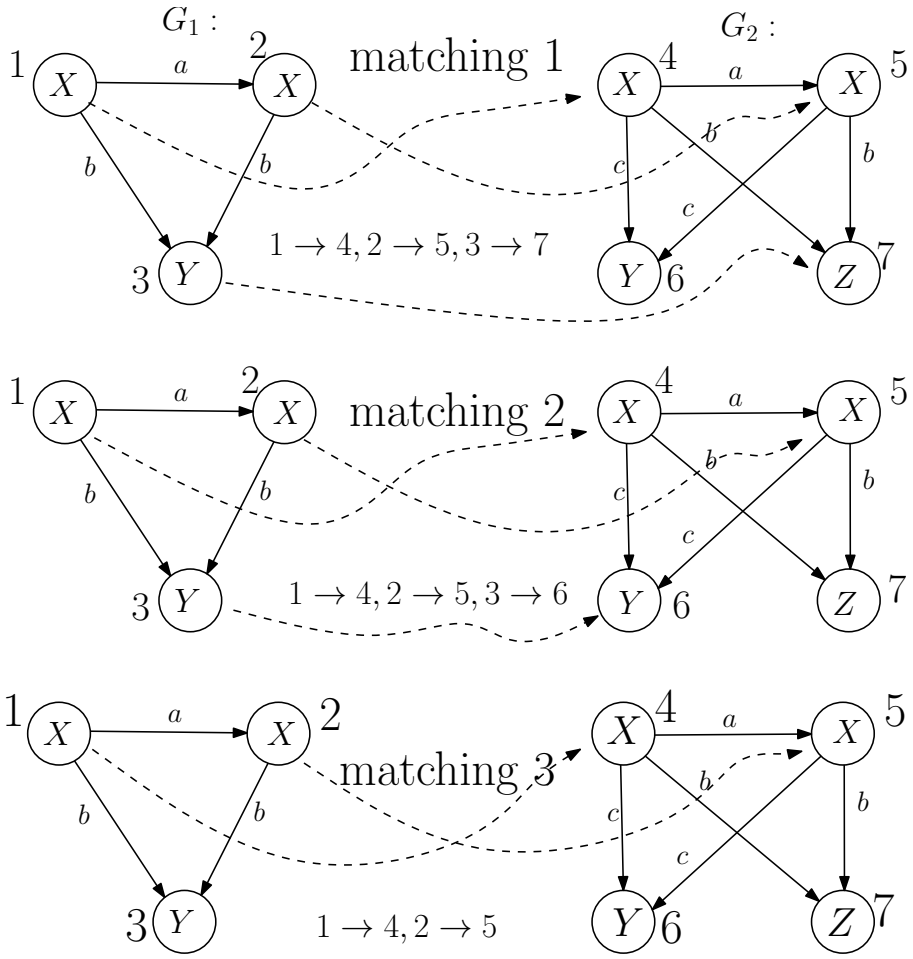


Figure 21: 서로 다른 3가지 그래프 매칭결과. 매칭되는 쌍 그리고 각 동작의 비용함수에 따라서 각 매칭된 그래프의 유사도(편집거리) 값도 달라진다.

있다. 그리고 앞에서 소개한 belief network을 활용한 그래프 매칭 알고리즘이 올해 CMU 연구진에 의해서 제시되었다 [16]. 성능 평가용 데이터로는 이스트의 PPI 네트워크가 사용되었으며, PCA를 이용하여 전처리 작업을 한 것이 이전 연구와 차별되는 부분이다.

5.4 네트워크의 다중정렬 알고리즘

특정 바이오 모듈의 기능을 이미 분석이 끝난 네트워크와의 비교를 통하여 유추하는 쌍정렬(pairwise alignment)과 달리 여러 네트워크를 동시에 고려하는 다중정렬은 여러 종에 공통적으로 존재(conserved)하는 공통성을 찾거나 그 차이를 확인하려는 목적으로 진행된다. 이 작업은 진화 계통분석(phylogenetic analysis)에 특별히 유용하다.

앞서 언급한대로 여러 종의 생물 네트워크에서 공통적으로 존재하는 기능은 생존이라는 **궁국의 공통목적**에 필수불가결한 기능으로 쓰일 수 밖에 없기 때문이다. 매우 중요한 의미를 가진다. 서열정렬에서 언급한대로 다중정렬은 그 비교할 종 갯수의 제곱으로 시간, 공간 복잡도가 늘어나기 때문에 비교종의 수가 많아지면 문제는 현실적으로 해결 불가능한 상태가 되어버린다. 이를 위해서 다양한 휴리스틱이 존재하는데 가장 단순한 방법은 그리디(greedy) 전략을 사용하는 것이다. 만일 방법은 비교할 g개의 네트워크가 $N_1, N_2 \ldots N_g$가 있다고 할 때, 일단 가장 비슷한 두 종의 네트워크를 선택하여 먼저 쌍정렬을 통하여 매칭되는 짝을 결정하는 방법이 가장 단순한 그리디 전략이 된다. 그렇게 선택된 해당 쌍이 N_i, N_j라고 하자. 그러면 이 둘을 제외한 $g-2$개의 네트워크 중에 이미 정렬된 N_i, N_j과 가장 유사성이 높은 네트워크를 추가로 찾아서 이미 정렬된 네트워크에 추가하여 정렬을 한다. 이런 식으로 매번의 단계에서 가장 높은 정렬값을 가진 쌍을 선택하여 정렬하면 전체 다중정렬된 네트워크들은 tree 구조로 연결되게 된다. 추가할 때 제약을 주면 linear한 path 모양으로도 정렬할 수 있다. 다중 정렬의 휴리스틱으로 star alignment도 있다. 이 방법은 다른 $g-1$개와의 유사성이 가장 높은(모두 더한 값으로) 네트워크를 하나 선택해서 이것을 정렬의 중심이 두는 것이다. 그리고 나머지 네트워크들을 이 중심과 star graph 형식으로 연결한다. 이 모두는 최적 정렬을 위한 휴리스틱이기 때문에 실제 데이터의 특성과 제한하는 조건에 따라서 많은 성능 차이를 보인다.

아래 그림-22으로 다중정렬을 설명한다. 그림에 제시된 4개의 그래프 A, B, C, D에서 1,2,4,5 노드로 구성된 subgraph는 4개의 네트워크에 공통으로 존재하는 것임을 알 수 있다[40]. 그러나 그 아래 그림-23에서는 4개의 그래프 모두에 공통으로 존재하는 크기가 4인 subgraph는 없음을 알 수 있다. 그런데 이 4개의 그래프에서 그림과 같은 회색 노드를 인위적으로 추가함으로서 위와 같은 4개 노드로 구성된 공통의 subgraph { 1,2,4,5}를 찾아낼 수 있다. 즉 각 그래프 A, B, C, D에 하나씩의 dummy를 추가함으로서 크기가 4인 공통 그래프를 찾아니는 것이 편집 비용적으로 이득이라고 판단하였기 때문이다. 즉 공통 그래프를 위하여 추가한 dummy node에 따른 비용보다 그것을 넣음으로서 우리가 얻을 수 있는 공통부분의 가치(크기)에 따라서 dummy노드를 추가할지의 여부가 결정된다. 문제는 어디에 얼마만큼의 dummy node를 만들어 넣는가 하는가가 하는 것인데 이는 결국 비용함수에 따라서 결정된다. dummy insertion의 비용이 비싸면 추가되는 일은

최대한 억제될 것으고 공통부분의 크기는 작을 것이다. 이것이 모든 다중정렬 알고리즘의 핵심변수가 된다.

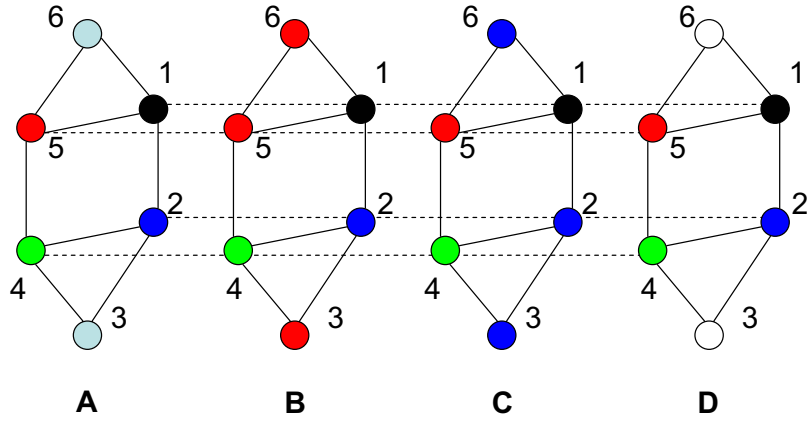


Figure 22: 네트워크 다중정렬의 예[40]. 4개의 네트워크에서 기능과 구조가 보존되는 subnetwork을 찾는 것이 다중정렬의 목적이다. 그래프 노드의 색은 생물학적 기능을 나타낸다. 이 그림에서 {1,2,4,5}로 구성된 subnetwork은 4개 종(그래프) 모두에게서 그대로 보존됨을 볼 수 있다. 따라서 {1,2,4,5} subnetwork은 이 다중정렬의 결과물이다.

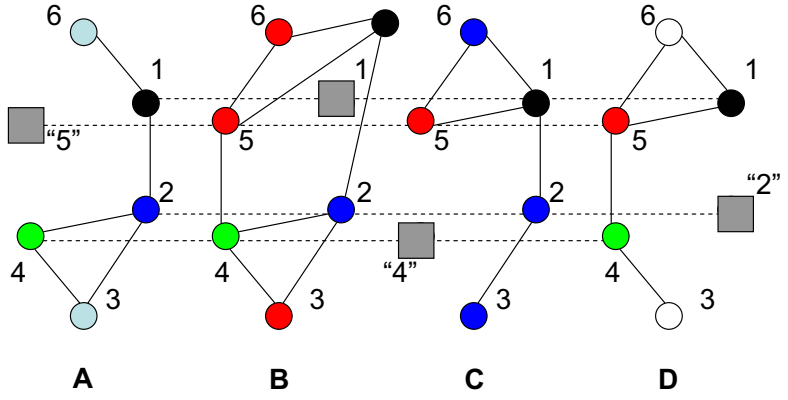


Figure 23: 네트워크 다중정렬에서 잡음과 불일치가 있는 예[40]. 완전히 일치하는 공통의 subnetwork은 이 4개의 네트워크에서 없기 때문에 적절하게 dummy를 추가하여 보존되는 최대 크기의 subnetwork을 찾아내야 한다. 회색 노드가 정렬을 위하여 추가된 dummy node이다.

참고문헌-[40]은 이를 위해서 간결한 다중정렬 알고리즘(일종의 star alignment)을 제시하고 있다. Weskamp의 방법은 G_1부터 G_n까지 n개의 네트워크에서 중심축이

되는 네트워크를 먼저 선택한다. 그것은 다른 모든 그래프와의 유사도가 가장 좋은 것을 선택해도 좋고(평균), 또는 최고 유사도값을 가지는 한 쌍에서 선택해도 좋다. 그것을 G_i라고 하자. 그러면 G_i와 그것과 가장 유사한 다른 G_j를 pairwise align한다. 이때 필요하면 dummy node를 추가한다. 이제 나머지 align 되지 못한 G_k중에서 이미 align된 그래프와 가장 유사한 것을 찾아서 정렬한다. Weskamp의 다중정렬 방식은 간단하지만 중앙 center graph를 어떻게 잘 잡는가에 따라서 성능은 크게 좌우된다. 따라서 몇 개의 다른 seed를 선택하여 위 작업을 반복한 뒤에 적절한 결과를 만들어내는 반복개선 작업이 필수적이다.

요약하자면 시간복잡도나 공간복잡도의 관점에서 압도적으로 우위에 있는 정렬 알고리즘은 없다고 보는 것이 타당할 것이다. 정렬 알고리즘은 전처리(Preprocessing)를 하는지, 한다면 어떻게 하는지, 입력 데이터의 모형을 얼마나 잘 활용하는지[44], 또는 오류의 범위를 어디까지 허용하는지에 따라서 다양한 변형이 나타날 수 있다.

[44]예를 들어 생물 네트워크에서 아주 큰 Degree를 가진 hub가 존재한다면 이것을 중심(center)에 놓고 정렬함으로서 거의 선형시간에 정렬을 마칠 수 있다.

6 네트워크 정렬 시스템의 실제

이 장에서는 위상만 존재하는 그래프와 달리 다른 다양한 생물학적 정보가 추가된 생물 네트워크에서 기본 원소인 개별 node의 유사도를 어떻게 계산하는지에 다하여 먼저 살펴본다. 그리고 정렬 알고리즘의 기본적인 또는 공통적인 구조 원형(prototype)에 대하여 설명한다. 그리고 실제 생물 네트워크 정렬을 위하여 개발된 시스템 중에서 대표적인 것들을 살펴보고 각 시스템이 적용하고 있는 알고리즘의 구조와 특징을 비교한다. 현재까지 공개된 정렬 시스템의 종류는 상당히 많지만 가장 최근, 2010년 이후에 소개된 대표적인 도구나 시스템을 중심으로 소개할 것이다. 이와 더불어 생물 네트워크의 성능평가에서 어떤 지표가 사용되는지, 또 공정한 평가를 위해서 어떤 데이터를 사용하고 있는지, 그 평가과정에서 유의해야할 점에 대해서도 설명한다.

6.1 생물 네트워크 단위노드의 유사도

네트워크 정렬에서 가장 기본적인 연산(basic operation)은 네트워크의 단위 노드간 유사도를 계산하는 것이다. 모든 정렬 알고리즘은 이 단위 노드간의 유사도 계산에서부터 시작한다고 봐도 무방할 것이다. 단위 노드는 PPI와 같이 하나의 단백질[45]이나 유전자, 또는 화학적 복합물 등 여러 단위체가 될 수 있다. 우리는이 문제를 그림-24을 통하여 세 개의 네트워크 N_1, N_2, N_3에서 각 노드 u, v, v'의 유사도를 계산하는 과정으로 설명한다.

생물네트워크 두 노드의 유사도를 구하는 가장 단순한 방법은 주위 위상을 생각하지 않고 두 노드의 독립된 물리적 유사도만을 고려하는 것이다. 예를 들어 단백질이나 유전자라면 BLAST 계산값을 그 유사도로 사용할 수 있다. 그것이 유전자라면 Gene Ontology DB를 이용해서 GO상에서의 유전자 거리로 대신할 수 있다. 이 방법을 사용하면 아래 그림-24에서와 같이 v와 서열상으로 비슷한 v'의 유사함이 u보다 높게 나올 것이다. 그런데 그런 물리적 유사도 외에 그와 연결된 위상까지를 유사도 평가에 고려한다면 이웃의 연결구조가 확연히 다른 v와 v'의 노드 유사도는 현저하게 낮게 나올 것이다. 도리어 연결구조가 유사한 u와 v의 노드 유사도가 더 높게 나올 것이다.

[45] 각각의 선형서열 아미노산 서열로 정확히 표현된다.

그림-24와 같이 N_2와 N_3가 다른 점은 각각 하나의 edge(굵은 점선으로 표시된) 씩만 다를 뿐이다. 따라서 연결 위상만 본다면 N_2의 u는 N_3의 v와 훨씬 더 유사하다. 결국 모든 정렬 알고리즘은 이 두개의 값, 즉 개체간의 물리적 유사도와 이웃을 고려한 위상적 유사도를 어떻게 조합해서 사용하는가에 달려있다. 위상만으로 계산하는 유사도 값을 topological similarity로 하고 단위 노드의 생물학적인 정보상으로 유사도를 평가하는 것을 domain-specific similarity라고 한다[41]. 물론 이 두 관점은 항상 적절한 trade-off를 고려하여 조절되어야 한다. 후자 즉 domain-specific similarity를 사용할 경우에는 모든 쌍간의 생물학적 유사도를 계산한 다음에 이들을 bipartite graph를 만들고 각 쌍들의 matching의 합을 최대로 만드는 optimal assignment problem을 찾는 과정을 거쳐서 가장 유사도가 높은 짝을 찾아야 한다.

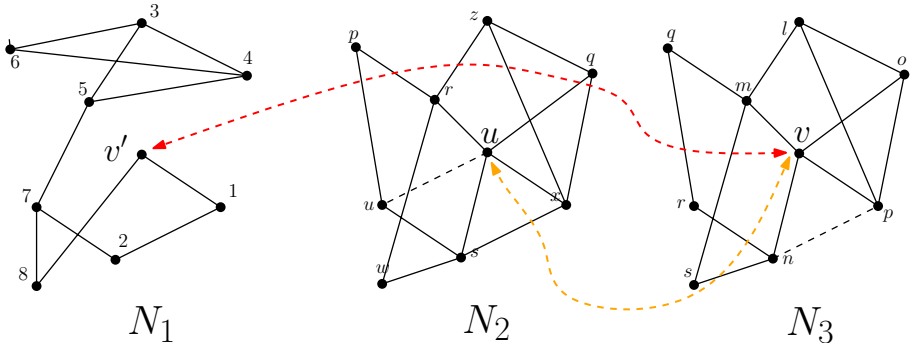


Figure 24: 3개의 생물 네트워크 N_1, N_2, N_3에서 각 단위 노드 u와 v, v' 사이의 생물학적 유사도를 계산한다. 단위노드의 생물 domain 정보만 본다면 서열적으로 유사한 $v \in N_3$가 $v' \in N_1$와 더 유사하게 평가된다. 그러나 위상 관점으로 본다면 주위 이웃과의 연결구조가 비슷한 $v \in N_3$가 $u \in N_2$과 더 유사하다고 계산된다.

6.2 네트워크 정렬 알고리즘의 일반구조

앞서 설명한 바와 같이 위상만 존재하는 수학적 그래프의 isomorphism을 구하는 방법은 생물 네트워크에 그대로 적용할 수 없다. 수많은 실험오류가 존재하는 생물 네트워크에서 isomorphic한 네트워크가 존재할 가능성은 거의 없다, 만일 그런 오유가 없이 완전 isomorphic한 subgraph를 찾는다면 대부분의 경우 trivial한 K_3 정도 크기에 subnetwork만 찾아지기 때문에 현실적인 의미는 없다. 따라서 실제 현장에서의 네트워크 정렬은 일종의 근사 알고리즘(approximate algorithm)으로 접근할 수 밖에 없다. 또한 정렬된

두 네트워크가 얼마나 좋게(goodness) 정렬되었는지를 판단하는 일도 응용 목적에 따라 다르기 때문에 이 두 문제는 생물 네트워크 정렬의 가장 이슈가 되고 있다.

우리가 먼저 결정해야 할 것은 전역 정렬을 할 것인지, 지역 정렬을 할 것인지 이다. 또한 한 쌍의 네트워크만을 정렬할 것인지(pairwise alignment) 인지 아니며 복수개의 네트워크를 정렬해서 그들이 가지고 있는 공통의 모듈을 찾을 것인지도 미리 결정해야 한다. 아래 그림-25은 지역정렬과 전역정렬의 예를 보여주고 있다. 지역 정렬은 최적의 정렬인 경우라도 그림-25의 왼쪽 a와 같이 복수개의 후보 답이 나올 수 있다. 그런데 전역 정렬(그림 b)는 거의 모든 노드가 매칭에 사용되어야 하기 때문에 매칭에 따른 비용함수가 결정되면 최적 전역정열은 유일할 수 밖에 없다. 그림-25 b에 보면 매칭에 들어가지 못한 노드도 있다. 예를 들면 I'나 J'가 바로 그러한 노드들인데 이런 노드는 왼쪽 초록색 그래프로 보면 일종의 gap node가 되어 모두 penalty score를 받게된다.

그런데 생물 네트워크의 각 기본 노드[46]의 영향은 대부분 그 위치의 이웃에 국한되기 때문에 전체를 고려하는 전역정렬은 별로 많이 쓰이지 않는다. 종간의 유전거리가 아주 가까운 경우, 예를 들어 인간과 침팬지의 기능이 매우 유사한 유전자 네트워크에 정역정렬과 같이 특수한 경우만 전역정렬이 사용된다. 이런 이유로 전역정렬에 특화된 알고리즘도 대부분 대부분 지역정렬 기능을 포함하고 있다[42, 43]. 이런 현상은 4장에서 설명한 서열정렬(sequence alignment) 문제에서도 마찬가지다. 공개된 서열 정렬 시스템 대부분은 지역정렬에 특화되어 있다. 요즘의 이슈는 다중정렬 중에서도 1000개 이상의 서열을 처리하는 다중정렬 시스템을 어떻게 개발하고 평가하는가이다. 예를 들어 전유전체 1000개를 동시에 살펴보는 1000 genomes[47]에서 가장 핵심적인 분석 도구는 다중정렬 시스템이 될 것이라고 예상된다.

서열정렬에서 사용되는 정렬은 대부분 지역정렬(local alignment)이 아주 가까운 종, 예를 들어 사람끼리(동양인과 서양인) 비교하여 SNP을[48] 찾아내는 일 정도에 전역정렬이 사용된다. 가장 많은 연구자들이 활용하는 BLAST는 가장 대표적인 지역정렬 도구라고

[46] 유전자 나 단백질, 또는 receptor, cell 등

[47] http://www.1000genomes.org/

[48] Single Nucleotide Polymorphism, 사람의 염색체에 존재하는 개인별 변이. 대략 1000 base마나 하나씩의 DNA가 다르게 표시된다. 이 변이는 인종별, 질병별 등으로 모두 다르게 나타나므로 유전학이나 의학에서 아주 중요한 바이오 marker가 된다. 특히 특정 유전병이나 유전질환 연구의 경우 이 SNP을 찾아내는 것이 가장 핵심의 일이다. 한의학에서도 그 고유의 분류법인 사상체질에 대응하는 SNP이 존재할지의 여부에 대한 연구가 진행된 적이 있지만 뚜렷한 결론은 아직 나오고 있지 못한 상황이다.

할 수 있다.

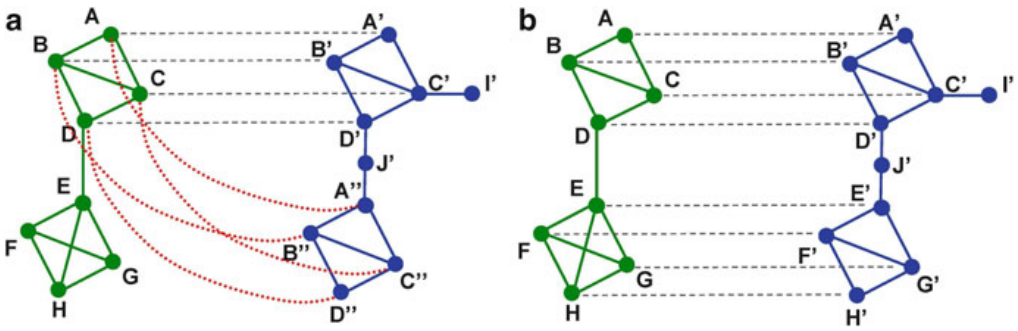


Figure 25: 두 생물 네트워크에서의 local alignment와 global alignmemt의 비교 도식.[8]

이제 네트워크 정렬 알고리즘이 가진 공통의 기본구조를 설명하고 대표적인 몇 시스템을 살펴보고자 한다. 우리는 그래프 정렬이 아닌 방법으로 그래프의 유사성을 비교하는 방법을 앞에서 살펴보았다. 예를 들어 그래프의 spectral Eigenvalue나 degree분포, color 수 등과 같은 특성지표(characteristic parameter)는 단순히 유사한 정도만 알려줄 뿐 어느 부분이 어느 부분과 유사한지에 대해서는 알려주지 못하기 때문에 정렬에 직접 사용될 수는 없다. 단 graph DB에서 특정 그래프와 유사한 또는 alignment가 잘되는 그래프를 선별하는 과정에서 filter의 역할은 할 수 있다. 이 장에서 이 부분, 특성지표로 그래프의 유사성을 판단하는 부분은 생략한다.

거의 모든 네트워크 정렬 알고리즘은 아래와 같은 공통의 골격을 가지고 있다. 우리는 어떤 가상의 유전자 네트워크 G_1과 G_2를 정렬하는 예를 활용하여 이 공통의 구조를 설명한다.

6.2.1 입력 네트워크 전처리 과정

입력 데이터를 그대로 정렬에 사용하기는 쉽지 않다. 일단 데이터의 크기가 너무 큰 경우나 서로 분리되어 있는 경우에는 크기를 작게 만들거나, 분리된 그래프 component 들끼리 따로 따로 정렬해야 하므로 이것을 사전에 확인해야 한다. 특히 잡음이 심한 네트워크의 경우 상당한 수의 노드나 그와 연결된 edge를 미리 걸러내야만 의미있는 결과를 얻을 수 있다. 특히 이미 annotation이 끝난 네트워크를 reference model로 사용할

경우에는 그 annotation의 방법이나 결과가 우리가 align하고자하는 네트워크의 구성 방법과 호응하는지도 확인해야 한다. 즉 서로 전혀 다른 방법으로 구성된 네트워크나 다른 목적으로 개발된 네트워크를 비교하는 일은 전처리를 통해서 걸러져야 한다. 지역정렬이든 전역이든 현실적으로는 2가지 구별되는 방법론이 존재한다. 하나는 그리디(Greedy) 기반의 구성적 알고리즘(constructive algorithm)[49]이며 다른 하나는 하향식 간략화 방법이다.

6.2.2 방법론 1 : 그리디 알고리즘 기반의 시작점 설정

어떤 네트워크 정렬 알고리즘도 반드시 시작하는 노드나 지역을 가지고 있다. 이것은 1차원 BLAST에서 anchor region과 같은 역할을 한다. 항상 이 anchor지역으로 쓰일 유사한 ㅈ역(부분)을 찾아서 확장해나가는 것이 이 계열 정렬 알고리즘의 공통적인 구조이다. 그런데 이 시작점으로는 하나의 노드도 가능하고 그림-26과 같이 작은 subgraph도 선택될 수 있다. 시작점은 가능한 두 네트워크에서 유일한 모양이 되는 것이라야 전체 계산시간을 줄일 수 있다. 또는 NetAligner와 같이 GO를 통하여 이미 유사성이 밝혀진 노드 쌍이나 subgraph, 특정 주요한 motif에서 시작할 수도 있다. 이 때 두 anchor가 얼마나 유사한지는 GO DB 등 유사한 생물관련 DB를 참조하여 계산한다. 완전 그래프인 K_n을 anchor region으로 사용할 수 있지만 양쪽 모두에 공통적으로 존재하는 K_r을 구하는 것 자체에 $O(|G|^r)$의 시간이 소요되기 때문에 큰 크기의 공통 clique[50]을 찾아내는 것은 부담이 큰 전처리 작업이다.

6.2.3 방법론 2 : 전체 구역의 분할을 통한 네트워크 간략화

위에서 제시한 그리디 방법과 달리 다른 한 방법은 3단계를 거쳐 정렬한다. 첫 단계는 네트워크를 몇 개의 작은 단위로 분할하는 것이고 그 다음 단계는 분할된 각 단위들끼리의 최적쌍을 계산하는 것이다. 그 다음 최적의 쌍으로 고정된 노드에 포함되지 못한 노드들은 고정된(앞 단계에서 anchor로 짝지워진 노드) 지역을 참조해서 지역적으로 유사한 노드들끼리 매칭을 찾는 일이 마지막으로 진행된다. 첫 단계에서 행할 분할의 단위는

[49]이 방식은 여러 단계의 작업을 거쳐 부분해를 계속 확장하여 구성하는 방식이다. Kruskal의 spanning tree 알고리즘이 이 방식이다. 한번 선택되어 결과에 포함된 해는 다시 취소되는 경우가 없다. 만일 이것을 허용하면 구성적 알고리즘과 대비되는 반복 개선 (iterative improvement) 알고리즘이 된다.

[50]어떤 그래프의 subgraph 중에서 complete K_e인 것

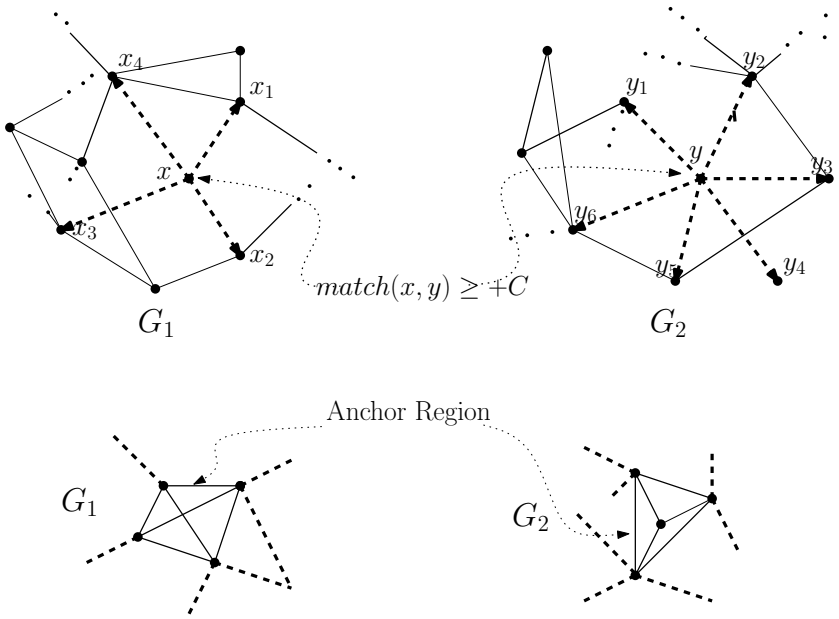


Figure 26: 두 네트워크의 정렬 시작점이 하나의 노드인로 설정된 x, y 인 경우 (위), 시작점이 하나의 노드가 아닌 부분 그래프 (subgraph) 설정된 경우 (아래 그림)

작은 모듈인 K_3 정도 크기거나 annotation 이 분명한 노드들의 구역으로 선택한다. 즉 두 네트워크에서 몇 개의 중요 단위를 선택하여 이들끼리의 domain-specific 유사도를 구한다. 이 작업은 결국 전체 네트워크를 몇 개의 특징지역으로 분할하는 것이다. 그 다음에는 이 분할된 지역끼리 어떻게 짝을 지우면 전체의 유사도가 최대가 되는지를 결정하는 과정인데, 이 과정은 잘 알려진 최적의 배정 (optimal assignment) 문제로 형식화 할 수 있다. 또한 이 문제는 Hungarian algorithm [51]을 이용하면 다항시간에 최적의 값을 찾을 수 있다.

그림-27이 보여주듯이 우리가 선택한 anchor 특징 노드는 각각 4개씩 {1, 2, 3, 4}, {a, b, c, d}이다. 그리고 이 4개 단위노드 (또는 확장하면 지역) 끼리의 생물학적 유사도는 다음 표-1와 같다고 하자. 우리는 이 4개 점들끼리의 짝을 정하고자 하는데 그 짝들의 개별 유사도의 전체 합이 최대가 되도록 짝을 만들고자 한다. 아래 행렬에서 쌍들끼리의 유사도 값이 행렬 원소 $M_{i,j}$ 에 주어져 있다. 이 행렬에서 4개의 짝을 중복없이 선택하여 전체의 합을 취대화는 것이 optimal assignment 문제이다. 이 문제는 그리디 방법으로는

[51] 이 이름은 최초로 이 문제의 알고리즘을 제시한 연구자들의 국적이 모두 헝가리인데서 유래된 것이다. 이 알고리즘의 복잡도는 n^3 인데 지금은 $n^{1.5}$ 까지 개선되어 있다. 이 문제의 optimal algorithm에 대한 연구도 매우 중요한 그래프 알고리즘의 주제이다.

최적을 구할 수 없다. 즉 가장 유사도가 높은 짝부터 먼저 선택하는 방식으로는 최적의 답이 나오지 않는다. 아래 그림에서 원소 $M_{1,c} = 14$로 제일 높은 값인데 이 때문에 제일 먼저 1번과 c를 먼저 짝지우면 결국 최적의 답은 나오지 않는다. 아래 그림으로 굵은 글씨로 표시된 쌍이 최적의 assignment이다. 따라서 매칭의 최적값은 12+11+10+13=46 이다.

	ua	ub	uc	ud
$v1$	11	9	14	**12**
$v2$	6	**11**	8	7
$v3$	**10**	9	5	9
$v4$	7	8	**13**	7

(1)

이렇게 중요한 achnor를 먼전 상위단계에서 짝은 지운 다음, 나머지 노드들을 이 고정점들을 중심으로 적절히 배치하는 것이 하향식 정렬방법의 과정이다. 이 방법의 장점은 몇 구역으로 나누어진 작은 구역끼리의 매칭 취적값을 다항시간에 구할 수 있다는 것이다. 문제는 그 전에 어떻게 전체를 분할하는가, 얼마나 많은 구역으로 분할하는가 (몇 개의 anchor를 마련하는가) 하는 문제이다. 만일 모든 점에 대해서 이렇게 최적쌍을 구하면 전체 topology의 preserving은 전혀 고려되지 않을 것이므로 위상을 고려한 네트워크 정렬이라는 원래 취지는 사라지게 될 것이다. 이 특징점을 하나만 선택하면 위애서 설명한 matching and exptend방식의 그리디 방식과 같아진다.

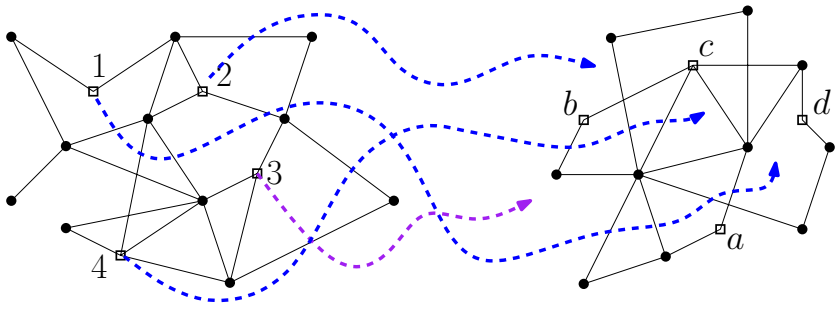


Figure 27: 두 네트워크에서 몇 개의 중요 단위(흰색 박스)를 선택하여 이들끼리의 domain-specific 유사도를 구한 뒤에 가장 좋은 짝을 찾아서 그들끼리 먼저 연결한다 (coarse-level alignment). 그 다음 나머지 노드들은 이미 짝 지워진 anchor node를 중심으로 적절하게 짝을 찾아 확장시켜 나간다. 이 단계를 fine-level 작업이라고 부른다.

6.2.4 부분해의 확장

이제 시작 위치가 정해졌으면 이 지점을 조금씩 넓혀저 유사한 지역을 더 넓히는 과정이 필요하다. 아래 그림-28와 같이 anchor의 비교지역을 얼마나 넓게 보아야 하는지에 따라서 다양한 변형 알고리즘이 생길 수 있다. 가장 단순한 방법은 anchor에 인접한 노드등 중에서 가장 유사성이 높은 쌍만을 선택하는 것이다. 그림-26의 경우라면 $\{x_1, x_2, x_3, x_4\}$와 $\{y_1, y_2, y_3, y_4, y_5, y_6\}$를 모두 비교한다. 즉 4×6개 쌍을 비교하여 그 중에서 matching score가 가장 높은 상위 k개 쌍을 선택한다. 그리고 이 선택된 쌍을 matched region에 포함하고 위 작업을 반복한다.

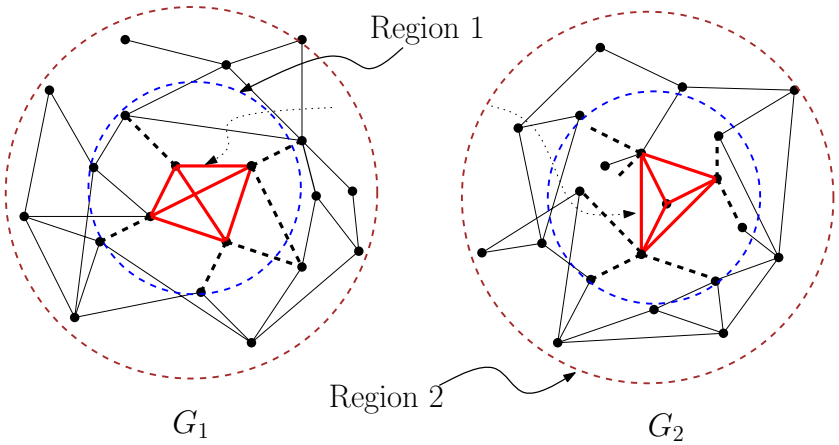


Figure 28: 처음 시작지역이 각각 K_4인 두 네트워크. 시작지역에서부터 어느 범위까리 비교대상에 포함시킬 것인지를 결정해야 한다. 파란색 점선은 anchor(붉은 색으로 표시된 subgraph) 지역에서 거리 1인 구역이고 황동색 점선으로 표시된 구역은 거리 2인 지역이다. 대상 지역을 넓게 보는 것이 더 나은 결과를 줄 가능성은 있지만 그에 따른 계산량은 지수적으로 늘어난다.

아래 그림-29의 경우를 보자. 시작점은 둘 모두 x로서 같다. 각각의 이웃 노드중 중에서 이 상황에서 유사한 쌍은 존재하지 않는다. 만일 거리-1의 이웃공간만 고려한다면 이 정렬은 시작점 x에서 더 확장되지 못하고 멈추게 된다. 그런데 한 단계 더 넓혀 비교영역을 확장하면 매우 높은 상관성을 보이는 m과 m'쌍을 새롭게 찾아낼 수 있다. m이 유전자라면 m과 m'이 서로 orthologous 한 경우라고 가정한다. 이런 연관성은 OrthoDB `http://orthodb.org/orthodb7`를 통하여 확인할 수 있다.

그런데 이렇게 gap node가 aligned subgraph들어갈 것인지의 여부는 앞서 설명했듯이 gap penalty와 matching gain에 의해서 결정된다. 만일 gap penalty가 m과 m'의 매칭값

(gain) 보다 크다면 gap은 허용되지 않을 것이다. 만일 gap node가 들어가면 matching score는 $match(x, x')$ + $match(mm)$ - $penalty(t, gap)$ 이 될 것이다.

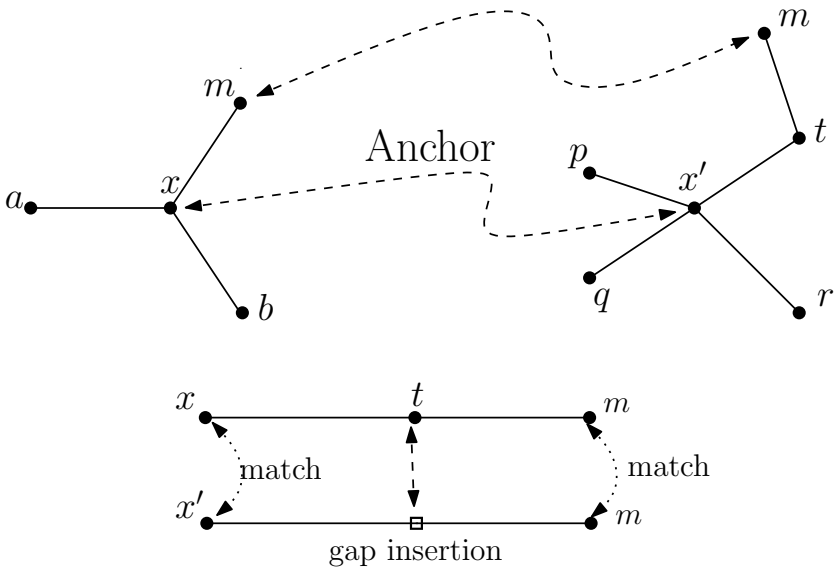


Figure 29: x 와 x' 이 matching anchor인 경우 gap을 허용하는 경우. m 과 m' 의 상관도가 매우 높을 경우 t 을 gap node로 넣고 한 노드를 건너 뛴 다음에 m 과 m' 을 매칭시켜줌으로서 alignment score를 더 높일 수 있다.

비교영역을 확장하는 과정에서 path만 고려한다면 pathBLAST[22] 와 같은 pathway용 alignment tool이 될 것이고, 모든 방향으로 확장해나가면 일반적인 네트워크 alignment tool이 될 것이다. 이웃범위의 확장을 어떻게 결정하는가에 따라 다양한 알고리즘이 생길 수 있다. 이것은 일종의 suboptimal solution을 구하는 문제와 동일하므로 다양한 heuristics이 나올 수 있다. 예를 들어 네트워크의 밀도(얼마나 많은 edge가 존재하는지)에 따라서 각각의 경우에 적합한 알고리즘이 나올 수 있다.

6.2.5 탐색(매칭)작업의 종료여부 판단

중앙 anchor에서 유사한 이웃 찾기위하여 새로운 gap을 넣거나 substitution을 이용해서 확장해 나갈 때 언제까지 이 작업을 해야하는가도 매우 중요한 문제다. 만일 충분한 이웃까지 보아도 더 이상 정렬 score가 증가되지 않으면 local maximum에 도달했다고 생각하여 작업을 중단하고 그것을 하나의 정렬 결과로 report할 수 있다. 이 작업은 다른 network alignment score의 분포를 이용하면 좀 더 도움을 받을 수 있다. 만일 이미 정렬이 끝난 결과를 가지고 있을 경우, 각 aligned subgraph의 매칭점수 분포에 대한

자료를 확보해두으면 이것과 비교함으로서 진행여부에 대한 정보를 얻을 수 있다. 따라서 비슷한 자료를 우리가 정렬할 때 그 값이 locally는 최고이지만 다른 네트워크와 비교해서 작을 경우에는 더 넓은 범위를 추가로 살펴보아야 함을 알수 있다. 이를 위하여 Bayesian 모형을 사용한 다양한 통계적 검증 기법을 활용힐 수 있다. 가장 단순한 방법은 Random Graph끼리의 local, global alignment를 통하여 현재 구한 정렬값의 p-value를 구하여 그것과 지금 정렬중인 네트워크 정렬갑과 비교하여 추가 진행여부를 결정할 수 있다[43].

6.2.6 성능개선을 위한 반복작업

우리는 앞에서 하나의 anchor 지역으로 부터 matching 지역을 확대해가는 방법을 설명했다. 그런데 만일 anchor 지역이 양쪽 network에 각각 p개 q개 존재한다면 BLAST와 같이 이 모든 쌍에 대하여 위의 작업을 반복하여 그 중에서 가장 좋은 결과를 찾아야 한다. 만일 지역정렬이라고 한다면 그 중에서 일정 이상의 값을 보인 matcher pair를 모두 답으로 출력해야 하며, 전역 정렬이라면 그 중에서 제일 높은 하나의 결과를 출력해야 한다.

그런데 anchor가 양쪽에서 단 하나 뿐이라면 이 후보쌍을 찾는 일은 간단해지지만, 그런 후보쌍이 갯수가 많으면 영역확장을 위한 작업의 횟수는 제곱으로 늘어날 수 있다. 만일 단일 노드가 아닌 좀 큰 subgraph, 예를 들면 그림-28와 같이 K_4로부터 시작한다면 그런 후보 쌍의 갯수는 줄일 수 있지만 모든 K_4중에서 matching이 될 수 있는 쌍을 찾는 작업에 $O(n^4 + H_1 \cdot H_2)$[52] 의 시간이 걸리므로 수행 시간상으로 유리하지 않다. 이 문제는 선형 서열의 지역정렬 알고리즘인 BLAST에서도 해결해야하는 공통의 현상이다. 그림-17에 있는 바와 같이 anchor 수를 많이 잡으면 false-negative[53]를 줄일 수 있어 시스템 성능의 sensitivity[54]를 올릴 수 있지만 그만큼 계산량은 급격하게 늘어나게 된다.

[52] 여기에서 H는 각 네트워크에서 선택된 subgraph의 갯수를 나타낸다.

[53] 실제로는 유사한 지역인데 아닌 것으로 판명하는 경우

[54] 쉽게 설명하자면 정답이 되는 것이 후보로 선택한 해(solution) 중에 포함된 비율. 이 값이 높인다는 말은 작은 solution하나라도 놓치지 않겠다는 뜻이다. 설사 true라고 판단했지만 실제 false인 경우가 많아도. 암과 같은 치명적 질환의 진단은 특히 sensitivity가 높아야 한다. 이와 반대의 의미는 specificity가 있다.

6.3 네트워크 정렬 시스템의 성능비교

공개된 여러 정렬 시스템의 정확한 성능평가를 위하여 공통의 데이터를 이용해서 상호비교 평가한 의미있는 연구결과가 있다. 최근 2014년 Clark 연구팀은 기존에 알려진 네트워크 정렬, 비교 및 예측의 성과에 대한 의미있는 논문을 발표했다[41]. 이 평가에 사용된 지표, 사용된 시스템들의 특성, 그리고 평가데이터별 성능의 차이에 대하여 살펴보고자 한다.

Clark은 그리디 방식의 match-and-extend 류의 알고리즘과 다단계 간략화(divide and conquer), 그리고 Graphlet기반의 성분 분석(composition analysis)기반의 시스템을 평가용으로 선택했다. 정렬 시스템은 이 외에도 정보이론을 이용한 방법[55], Machine Learning 기법에서 자주 활용되는 Markov Random Field 기반의 방법[56]도 있지만 본 보고서에 포함하지는 않았다.

6.3.1 정렬시스템 성능 지표

가장 기본적인 측정값은 에지 정확도(Edge Correctness, EC)이다. 즉 mapping한 정점의 에지가 mapped된 이후에도 그대로 유지되는가를 평가하는 것이다. 그림-30의 왼쪽 그래프에서 굵은 선으로 표시된 edge가 N_2에서도 그대로 유지되는지를 조사하여 그 비율을 측정한 것이 EC 값이다.

$$EC = \frac{|f(E(N_1)) \bigcap E(N_2)|}{|E(N_1)|} \tag{2}$$

그림-30의 경우 EC값은 1.0이다. 왜냐하면 모든 N_1의 edge, $E(N_1)$는 N_1의 vertex가 mapped된 이후에도 N_2에서도 그대로 유지되고 있기 때문이다. 그런데 이 EC measure의 한 가지 문제점은 mapping할 네트워크의 크기가 작은 경우나 mapping 시킬 그래프가 sparse할 경우에 EC값에서 에 훨씬 더 유리해진다는 것이다. 따라서 이런 단점을 개선하기 위하여 mapped된 그래프가 가지는 모든 vertex induced subgraph의 모든 edge를 기준으로 평가하는 개선된 기준인 Induced Conserved Structure(ICS)가 실제로는

[55] See 17. Chor, B., Tuller, T.: Biological Networks: Comparison, Conservation, and Evolution via Relative Description Length. Journal of Computational Biology 14(6) (2007) 817–838

[56] Bandyopadhyay, S., Sharan, R., Ideker, T.: Systematic identification of functional orthologs based on protein network comparison. Genome Research 16 (2006) 428–435

더 많이 쓰이고 있다. ICS measure의 formal definition은 다음과 같다. 아래 식에서 $N_{2,[f(V_1)]}$는 N_1의 vertex 중에서 N_2로 mapped된 vertex induced subgraph를 나타낸다.

$$ICS = \frac{|f(E(N_1)) \bigcap E(N_2)|}{|E(N_{2,[f(V_1)]})|} \tag{3}$$

그림-30의 경우에 ICS의 값을 계산해보자. N_1의 굵은 선으로 표시된 subgraph가 N_2에서 매핑된 부분의 vertex induced subgraph $N_{2\{A,B,C,D,E\}}$인데 그안에 포함된 edge는 모두 8 개다. 이 중에서 N_1와 같이 mapped된 edge는 모두 5개이므로 ICS값은 $5/8 = 0.625$가 된다.

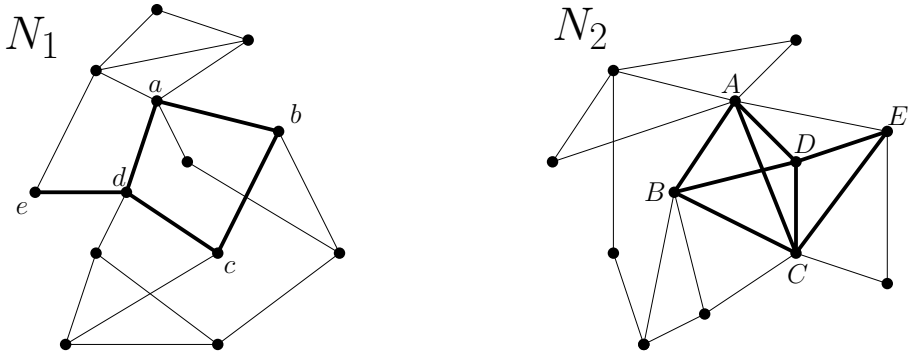
다른 metric으로는 Largest Connected Shared Component(LCSC)가 있다. 이것은 매핑된 개별 에지의 전체합이나 평균을 보는 것이 아니라 가장 큰 공통의 component를 얼마나 잘 찾아주는가를 측정해주는 지표다. 당연히 이 LCSC값이 클수록 더 우수한 정렬 알고리즘이 된다. 관련된 연구에 의하면 가장 매핑된 subgraph중 가장 큰 것은 크기와 매핑된 노드의 갯수에는 유의미한 상관관계가 있다고 한다. 따라서 LCSC와 EC, ICS는 모두 일정한 연관이 있다는 것이다.

앞서 설명한 measure는 위상적 보존성에 대한 측정값이다. 앞서 말한대로 위상적 보존성과 더불어 실제 mapped된 노드 쌍들의 domain-specific similarity도 매우 중요한 평가지표가 될 수 있다. 이것을 위해서 각 match된 쌍의 분자생물학적 또는 유전학적 유사도를 그 평가지표로 사용할 수 있다. 이 중에 GO(gene ontology) DB상에서의 진화적 관계(거리)가 PPI network에서의 정렬결과의 우수성을 평가하는데 사용된다. 즉 쌍으로 짝지워진 두 단백질 거리를 GO DB에 mapping 시켜서 그것이 GO DB상에서 얼마나 가까운(Gene Orthologous Distance)지 그 거리를 살펴보는 것이다. 특정 단백질에 annotation된 GO항목은 복수개가 존재하므로 각 항목당 level-5까지의 term을 모아서 그 term들의 유사성으로 두 단백질 네트워크 쌍 (u, v)의 기능적 유사성 지표로 쓸 수 있다. 아래 식에서 $GO(x)$는 단백질 x에 관련된 5단계[57]까지의 GO Term 의 집합을 나타낸다. GO consistencey라고 정의된 이 평가지표는 다음과 같이 계산된다.

[57]GO는 root 가 있는 트리로 구성되어 있어 해당 Gene root에서 level까지 표현된 모든 term을 더한 것이다.

$$GOC(u,v) = \frac{|GO(u) \bigcap GO(v)|}{|GO(u) \bigcup GO(v)|} \tag{4}$$

이 $GOC(u,v)$ 값을 모든 $u \in N_1$, $v \in N_2$에 대하여 더하거나 각 노드 갯수로 표준화하면 정렬된 네트워크가 얼마나 GO 공간상에서 가까운지를 알 수 있고 이것은 가능성으로 얼마나 잘 정렬되었는지를 가름하는 지수값이 된다.



$$f : N_1 \rightarrow N_2 \quad such\ that \quad f(a,b,c,d,e) = (A,B,C,D,E)$$

Figure 30: 네트워크 N_1을 N_2로 정렬하는 함수 $f()$. 함수$f()$는 N_1에서 굵은 실선으로 표시된 부분을 N_2의 굵은 선으로 표시된 부분으로 매핑되었다. 함수는 $f(a) = A, f(b) = B, f(c) = C, f(d) = D, f(e) = E$로 N_1의 노드를 N_2로 매핑시킨다.

6.3.2 평가용 데이터 선택의 문제

여러 시스템의 성능을 비교평가할 때 가장 중요한 것은 평가용 데이터를 어떻게 준비하는가의 문제이다. 생물 네트워크 평가용 데이터로는 ground truth가 완전히 확보된 시물레이션 데이터와 실제 wet 실험을 통해서 확보된 real network 데이터가 있을 수 있다.

네트워크 정렬 시스템의 성능평가를 위하여 평가용 데이터를 합성해주는 도구로는 NAPA Bench[58]가 있다. 이 사이트를 통해서 우리는 조절변수를 활용하여 실제 데이터와 가장 유사하게 다양한 특성의 생물 네트워크를 인공적으로 만들 수 있다. 네트워크 정렬의 목적이 상이한 네트워크에서 보존된 subnetwork이나 phylogenetic관계를 찾아

[58] http://www.ece.tamu.edu/~bjyoon/NAPAbench/

내는 것이기 때문에 conserved region이 존재하는 복수개의 네트워크도 NAPA Bench를 이용하여 만들 수 있다. 이렇게 정답이 확실한 데이터를 이용하면 객관적인 성능 비교평가가 가능하다. 하지만 실제 wet lab에서 생성된 real 네트워크는 전체 구조나 annotation 내용이 지속적으로 바뀌고(개선되고) 있기 때문에 평가시 어떤 version의 데이터를 이용하는가에 따라서 그 결과는 크게 달라질 수 있기 때문에 어려움이 많다.

Clark의 비교평가에서 실제 Wet lab data로는 IsoBase의 자료를 이용했다. 실제 데이트를 사용해서 평가할 때의 문제는 정렬된 모듈들이 진짜 의미있는 것인지를 확인하기 어렵다는 단점이 있다. 특히 NGS기반의 새로운 실험기술의 등장으로 엄청나게 빠른 속도로 새로운 실험데이터가 쏟아지는 현실에서 실제 생물 데이터의 정답을 확정하기란 거의 불가능한 일이다. 이것이 실제 데이터를 통하여 성능평가를 할 때의 가장 큰 어려움이라고 할 수 있다.

6.4 대표적인 네트워크 지역정렬 시스템

지금부터는 공표된 네트워크 정렬 시스템 중에서 언급할만한 성능의 시스템에 대하여 그들의 주된 방법론과 특성, 차별성에 대하여 지역정렬과 전역정렬로 나누어서 설명하고자 한다. 그리고 최근 들어 주목을 받고 있는 graphlet 기반의 정렬 알고리즘은 따로 구분하여 설명한다.

6.4.1 정렬 시스템의 시초 PathBLAST

pathBLAST는 Biological Network alignment용 실용적 도구로 처음으로 주목을 받은 시스템이다[22]. 이 도구는 이미 많은 연구가 이루어진 박테리아의 pathway 위상정보와 이스트의 pathway에서 유사한 부분을 찾아내기 위해서 사용되었으며 이름이 말해주듯이 서열 정렬도구인 BLAST의 방법을 원용(exploited)한 도구이다. BLAST와 같은 방법의 정렬도구를 The BLAST family라고 부르고 이 안에는 이 pathBLAST와 이것을 확장한 NetworkBLAST, 그리고 이것을 다시 개선한 NetworkBLAST-M이 있다.

서로 다른 두개의 network에서 선택된 2개의 path들의 각 노드들끼리 짝을 지어 match, mismatch, gap을 형성하고 있는 모양을 그림-31에서 볼 수 있다. pathBLAST의 방법은 linear sequence alignment의 일반적인 구조와 본질적으로 동일하다. 짝 지워진

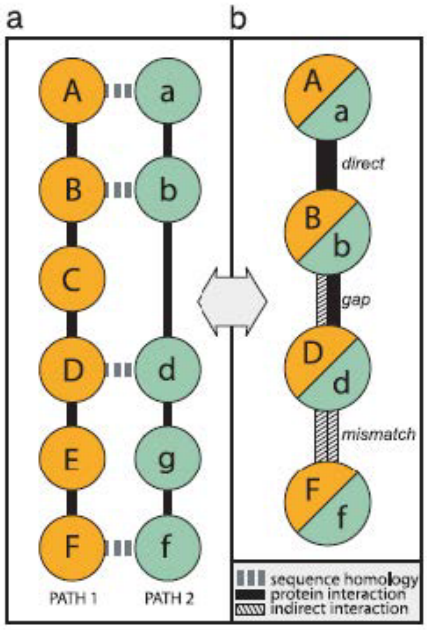


Figure 31: PathBLAST로 두 개의 path가 alig된 예. dummy node가 정렬에 포함되어 있다.

노드의 생물학적인 유사성(homology)에 따라서 pathBLAST의 alignment 모양은 달라진다.

pathBLAST는 두 개 이상의 Biological Network N_a N_b에서 의미가 있는 path를 align 해준다. 이 단순한 방법은 이렇게 시작된다. N_a에서 임의의 한 노드 x_i를 찾고 그와 가장 유사한 노드 y_i를 N_b에서 찾는다. 그 다음 이 출발점 노드에서 연결된 이웃 노드 중에서 서로 가장 유사한 노드를 하나씩 짝을 지워 그 path를 점점 확장해 나간다. 비유하자면 이런 식이다. 서울과 뉴욕을 도로중심으로 pathBLAST로 수행시킨다고 가정해보자. 먼저 두 지역에서 가장 번잡한 출발점을 선택한다. 즉 서울에서는 가장 번잡한 종로가 선택하고 그와 대응되는 뉴욕의 중심거리의 Y가 선택될 것이다. 그 다음 종로와 인전한 지점과 Y와 인접한 지점 중에서 가장 유사한 형상의 거리를 계산해서 찾는다. 이런 방식으로 path를 확장해나간다. 어떤 경우에는 서로 인접한 지점이 일치할 수도 있고(match), 또는 불일치 할 수도 있고(mismatch), 또는 한 지역을 건너뛰어(gap) 나갈 수도 있다. 이 경우 지역, 그러니까 path를 한단계씩 확장해 나갈 때 scoring function를 고려해서 이 과정을 진행한다. 즉 scoring function을 최적화 시키는 방향으로 진행하면 N_a N_b 에서 생물학적으로 유사한 pathway을 pthBLAST가 찾아준다.

$$S(P) = \sum_{v \in P} \log \frac{p(v)}{p_{random}} + \sum_{v \in P} \log \frac{q(v)}{q_{random}}$$

여기에서 $p(e)$는 align된 path상의 protein v가 짝을 이루고 있는 다른 쪽 protein과의 homologous한 정도를 나타내는 유사도이며, $q(e)$는 alignment graph에 포함된 interaction edge(N_a와 N_b를 연결해주는)들의 상호작용의 정도를 나타내는 확률값이다. 이것을 자세히 나타내면 아래와 같다.

$$q(e) = \prod_{i \in e} Pr[i]$$

그 아래 p_{random}은 $p(v), q(e)$의 aligned된 그래프에서의 평균값이다. 이 scoring function은 저자들의 경험적 측정실험에 의해서 결정된 것이다. 가장 의미있는 path들이 서로 대응되는 것 상황을 실험으로 확인하여 결정된 변수값이다. 이 시스템은 `http://www.pathblast.org`에서 다운받아 활용할 수 있다.

6.4.2 그리디 기반의 MaWISh(Maximum Weight Induced SubgrapH)

이 시스템은 네트워크 노드 유사도를 진화거리로 계산하여 그것을 정렬 비용으로 사용한다. 원래 Maximum Weight Induced Subgraph 문제는 Subgraph isomorphism보다 더 어려운 문제이다. 왜냐하면 Maximum Weight Induced Subgraph문제에서 모든 vertex, edge weight가 1일때가 바로 subgraph isomorphism문제가 되기 때문이다. 그들이 제시한 방법은 전형적인 그리디 접근법으로, 하나의 anchor node쌍에서 시작하여 조금씩 유사한 영역을 확장해가며 매칭을 진행한다. 물론 local minima에 빠질 수도 있지만 적절한 randomization을 사용하여 확률적으로 그것을 피해나가는 방법을 택하고 있다. `www.cs.purdue.edu/homes/koyuturk/mule`에 가면 해당 시스템과 사용된 데이터, 분석결과를 모두 확인할 수 있다.

6.4.3 Graemlin

원래 1.0 버전은 다중정렬(multiple network alignment) 전용으로 먼저 개발된 도구이다[23]. 이후 2.0 판에서는 전역정렬도 가능하도록 개선되었다. 기본적인 방법은 각

입력 네트워크의 부분 그래프를 equivalence class로 나눠서 각 class들끼리 matching 시키는 방법을 사용한다. 공통의 조상으로부터 유래된 단백질 집합이나 같은 종내에서의 paralogs가 매칭 단위를 이루는 equivalence class로 정리된다. 하나의 네트워크에서 진화적으로 가장 가까운 종의 내트워크를 순차적으로 정렬해가기 때문에 progressive alignment라고도 불린다. 2.0에서는 hill climbing method를 이용해서 좀 더 빠르게 local minima를 벋어나는 방법이 추가되었다. 그리고 이전 수작업으로 확인된 true set에 가까운 alignment 결과를 user defined 함수로 추가할 수 있도록 허용하여 최종 결과물이 신뢰도를 더 높일 수 있게한다. 사용자의 경험(이미 정리된 정렬결과)를 추가하는 과정에 다양한 machine learning 방법이 사용되고 있다. 관련된 자료와 시스템은 `http://graemlin.stanford.edu/graemlin-2.01.tar.gz`에서 얻을 수 있다.

6.4.4 진화최적화 방법의 GEDEVO

요즘 유행하는 다양한 AI기법이 네트워크 정렬에도 다양하게 응용되고 있다. 그 이유는 정렬 시스템은 결국 NP-complete문제를 푸는 heuristics algorithm이기 될 수 밖에 없기 때문에 optimization에 사용되는 모든 방법, 예를 들어 선형계획(Linear programming), Quadratic Programming, Neural Net, Belief Network[16]와 같은 시도가 모두 활용될 수 있다.

이 시스템은 두 그래프의 편집거리를 구하는 과정을 진화개선 알고리즘으로 접근하고 있다[36]. 대략의 방법은 이런 식이다. 매칭을 시켜야하는 두 그래프를 "짝"을(mating) 지워 그 중간 단계의 다양한 자식 그래프를 만든다. 이 자식 그래프들 중에서 양쪽 부모와 가장 닮은 그래프 후보를 몇 개 골라서 다시 다음 세대 개체군을 형성한다[59]. 이 과정을 반복해서 편집거리가 가까운 그래프를 정리하면 그것이 바로 A 그래프에서 B 그래프로 변화시키는 최단거리, 즉 최단편집의 과정을 보여주는 operation path가 되는 것이다. 저자들의 주장에 의하면 이미 잘 알려진 SPINAL, GHOST, C-GRAAL, M-GRAAL보다 더 나은 성능을 보여준다고 하는데 그 성능 결과와 평가 기준에는 동의하기 힘들 점이 있다. 이들은 단순히 EC measure만을 사용했기 성능을 비교했다. 이 시스템은 노드별 domain-specific knowledge적 관점을 고려하지 않았기 때문에 다른 생물학적 서열유사도나 GO DB유사도를 같이 사용하는 SPINAL, GHOST, C-GRAAL, M-GRAAL와 비교

[59] 전형적인 evolutionary optimization 과정이다

한다면 위상적 유사성, 보존성은 당연히 높을 수 밖에 없다. 이 때문에 비교 지표에서의 공정성에 문제가 있다고 보고자는 판단한다. 해당 시스템은 독일 막스-플랑크 연구소에서 관리하고 있다. http://gedevo.mpi-inf.mpg.de/에서 확인할 수 있는 해당 시스템은 생물 그래프가 아닌 일반적인 그래프에 응용하면 더 나은 결과를 얻을 수 있을 것으로 생각된다.

6.4.5 Web 환경을 제공하는 MetNetAligner

제공되는 정렬 시스템은 대부분 stand-alone형이라 설치하는 일이 쉽지 않다. 특히 다른 추가의 library를 깔고 package setting을 일일이 맞춰주는 일은 매우 어려운 일이다. 그래서인지 전산학 중심이 아닌 생물학 연구소에서 개발한 시스템 대부분에서 이런 사용자 중심의 설명에는 꽤 인색하다. 이 시스템과 같은 Web기반의 시스템은 성능을 차치하고서라도 그 사용자 인터페이스 방식에서 편리함을 제공해주기 때문에 충분히 가치를 가지며 주목할만 하다[44]. 이 시스템은 metabolic network 비교에 특화되어 있다. 초기 매칭은 각 효소의 기능상 유사도에 중점을 두고 시작된다. 그리고 효소 metabolic network의 특성상 이 도구에서 사용하는 그래프는 노드 사이에 방향이 있는 directed graph 기반의 네트워크이다. 해당 시스템은 다음에서 접근이 가능하다. http://alla.cs.gsu.edu:8080/MinePW/pages/gmapping/GMMain.html

6.5 대표적인 전역정렬 시스템

6.5.1 그래프 스펙트럴 기반의 GHOST

2012년에 소개된 GHOST(Global network alignment using multiscale spectral signatures) 정렬 시스템은 spectral graph를 활용한 대표적인 정렬시스템이다[45]. 특정 노드 쌍의 유사도는 두 그래프의 라프라시안 행렬의 고유값으로 각각 계산된다. 이 각 노드의 스텍트럴 유사도를 기본으로 이웃 노드의 유사도가 계산되고, 이것이 다음 단계의 노드 유사도로 재귀적으로(recursive) 계산된다. spectral graph 특성을 활용한 정렬 알고리즘은 그 정밀도는 다소 떨어지지만 각 노드의 고유값에 기초한 유사도가 병렬적으로 계산될 수 있다는 점에서 장점을 가진다. GHOST도 가장 유사한 seed node 쌍을 선택하여 이 시작점에서 매칭 지역을 확장시키는 일반적인 그리디 방법을 쓰고 있다. 한 가지

성능상의 특징은 매칭된 지역의 크기가 다른 도구에 비해서 비교적 커다는 것이다. 이 때문인지 즉 앞서 설명한 Largest Connected Shared Component(LCSC) 평가지표에서 가장 우수했다.

6.5.2 조합론적 최적화 방식의 NATALIE

이 시스템은 위상적 유사도와 생물학적 유사도라는 두개의 평가함수를 조합하여 최적화 시키는 정수계획법(Integer Programming)문제로 접근한다[60] 2단계의 라그랑제 relaxation을 거쳐서 정수계획법 문제로 바꾸어 푼다. 이 방식이 다른 정렬 도구와 다른 점은 전역정렬이지만 조건에 따라서 전체 노드 중 일부는 정렬에 포함시키지 않을 수도 있다는 것이다. 어떻게 보면 서열정렬에서 semi-global alignment와 비슷한 모형이라도 볼 수 있다. 그리고 두 노드간의 생물학적 유사도의 threshold를 지정해줄 수 있는(그 이하의 유사도는 고려하지 않음) 기능도 있다. 이 제한조건을 활용하면 noisy가 심한 노드는 계산과정에서 모두 제외할 수 있기 때문에 수행속도를 획기적으로 올릴 수 있다. NATALIE는 unix와 윈도우를 포함해서 다양한 운영체제 모두를 지원하는 점에서 가장 활용성이 높다고 할 것이다.

6.5.3 GO 정보를 활용하는 NETAL

단백질 네트워크의 전역정렬을 위한 시스템이다[46]. 기존의 잘 정렬된 PPI 네트워크에서 참조한 단백질별 매칭값과 단백질의 서열유사도를 종합하여 매칭비용 행렬을 새롭게 구성하여 활용한다. 주된 알고리즘은 Greedy 방법을 사용하는데 유사도가 높은 매칭쌍으로부터 시작해서 매칭구역을 넓혀나가는 전형적인 constructive algorithm frame이 사용된다. 저자들의 주장에 의하면 기존의 전역 정렬 시스템에 비해서 선택된 Edge의 정확도, 가장 큰 공통 부그래프(Largest Common Connected Subgraphs), 그리고 시스템을 통해서 보고된 매칭 단백질을 GO(Gene Ontology)DB로 평가했을 때의 EC measure에서 우월했다고 한다. NETAL의 가장 큰 장점은 속도가 빠르다는 것인데, 이런 이유로 NETAL을 submodule로 사용하면 많은 네트워크를 다중정렬하는데 전처리 과정으로 쓸 수 있는 장점을 가지고 있다. 특히 noisy network에 대하여

[60] 산업공학이나 OR에서 최적화를 해결하는 전형적인 방법과 같다. 속도와 안정성 면에서 뛰어나고 성능의 범위에 대한 사전연구가 잘 되어 있다는 점에서 유리하다.

성능 저하가 심하지 않는(robust) 시스템으로 평가받고 있다. 리눅스용 샐행 모듈은 http://www.bioinf.cs.ipm.ir/software/netal에서 구할 수 있다.

6.5.4 비교평가용 표준 시스템 IsoRank

생물 네트워크 정렬 알고리즘의 성능비교를 위하여 가장 많이 비교되는 시스템으로 전체적으로는 가장 표준적인 알고리즘으로 구성되어 있다[47]. 두 노드의 위상적 유사도는 각 노드의 이웃 이웃들의 유사도로 recursive하게 정의된다. 대부분 새로 개발되는 정렬 시스팀의 성능이나 새로 제안하는 방법론의 우수성은 이 IsoRank와의 성능비교를 통하여 측정된다[48].

6.5.5 SPINAL: scalable protein interaction network alignment

이 시스템은 단백질 네트워크를 2단계 과정(전처리와 반복개선)을 거쳐 수행한다[49]. 첫 단계에서는 각 노드별 이웃에 기초한 국지적 유사도를 계산하여 유사도 행렬을 만든다. 그 다음 단계에서는 첫 단계에서 먼저 매칭된 노드를 중심으로 최적의 이웃을 매칭시키기 위해서 optimal assignment 알고리즘을 이용해서 이웃들을 구체적으로 매칭시킨다.

이스트, 파리, 사람과 선충 등의 단백질 네트워크로 평가를 해보았을 때 기존의 방식보다 나은 결과를 보였다고 저자들은 주장한다. 이 시스템의 장점은 데이터의 크기나 종의 갯수가 늘어남에 대하여 유연한 scalability를 제공한다는 점이다. 그리고 수행시간상으로도 다른 방법과 비교해서 뒤떨어지지 않았다고 한다. 관련 프로그램과 사용한 데이터들은 다음 사이트에서 확인할 수 있다. http://code.google.com/p/spinal/

SPINAL은 이렇게 2단계의 정제 작업을 거치기 때문에 noisy한 네트워크에서 특히 강점을 가진다고 알려져 있으며 GRAAL[13]과의 성능비교에서도 상대적 우수함을 보였다. 또 두 단계의 작업을 사용자가 원하면 분리할 수 있어 1단계인 거친 정렬(coarse-refining)과 2단계 상세 정렬(fine-grained)를 다른 시스팀과 결합하여 사용할 수 있개 해준다. 즉 1단계는 SPINAL에서 돌린 뒤 그 결과를 다른 시스템에 준다든지, 아니면 다른 작업에서 대략 정리된 subnetwork을 SPINAL의 2단계 과정을 넣어 2차적으로 개선하는 것이 가능하다. 개발자들의 주장에 따르면 모든 지표에서 전반적으로 MI-GRAAL보다 더 나았다고 한다.

6.5.6 SA기반의 Net:Coffee

이 시스템의 특징은 전역정렬과 다중정렬을 동시에 제공해주는 기능을 가지고 있다는 것이다[50]. 기초하고 있는 방법은 Optimization 방법중에서 속도는 느리지만 가장 안정적인 답을 제공해주는 Simulated Annealing[61]이다. 다중 서열 시스템인 T-coffee의 triplet 탐색 방법을 일반적인 다차원 그래프로 확장한 버전이라고 보는 것이 적절한 설명이 될 수 있다. 저자들의 주장에 의하면 기존의 여러 시스템에 비해서 우월한 결과를 보여준다고 한다. 관련 시스템은 `https://code.google.com/p/netcoffee/`에서 볼 수 있다.

6.5.7 MAGNA: Maximizing Accuracy in Global Network Alignment

2014년 올해 소개된 이 시스템은 유사한 vertex쌍을 찾아서 그것으로부터 매칭을 확장해 나가는 것이 아니라, **가장 유사한 edge**를 찾아서 그것을 anchor로 삼아 그래프 매칭을 시도한다는 점이 큰 차이점이다[51]. 즉 기존의 시스템이 vertex별로 각각 mapping한 다음에 edge가 얼마나 잘 매핑되었는가를 보지만, 이 시스템은 생물 네트워크 N_a의 edge $(x, y) \in N_a$와 가장 유사한 edge $(p, q) \in N_b$에서 찾는 점이 큰 차이점이다. optimization 방법은 genetic algorithm을 사용했다. 그리고 네트워크 정렬에 특별히 적합한, 새로운 genetic algorithm용 crossover 연산자를 제안한 것이 의미를 가진다. 이 시스템은 `http://nd.edu/~cone/MAGNA`에서 볼 수 있다. 이 논문은 가장 최신의 평가결과를 담고있기 때문에 이 논문부터 살펴보면 최신의 동향을 이해하는데 실제적인 도움이 될 것이다.

6.5.8 Hub 노드 중심의 Hubalign

2014년에 소개된 시스템으로 전역 정렬에서 노드가 위상적으로 더 중요한 정보를 가지는지 또는 단백질 기능적으로 더 중요한 역할을 하는지를 판단하는 minimum degree heuristic을 제안하고 있다[52]. 그것을 허브(hub)라고 간주해서 그 허브 노드를 중심으로 정렬하는 새로운 방법을 발표했다. 개발자의 주장에 따르면 이전에 나

[61] 담금질 기법이라고 불리는데 고체물리에서 열이 높은 상태의 물질이 열이 식어감에 따라서 안정된 상태를 확률적으로 찾아가는 과정을 컴퓨터고 모사한 방법이다. 일명 볼쯔만(Boltzman)기계의 동작원리와 동일하며 반복적 확률과정을 통하여 최적을 찾아가는 방법이다.

온 대부분의 시스템과 비교해서 실제 데이터에서 더 나은 결과를 보여준다고 한다. 실제 human, yeast, worm, mouse, fly의 모든 단백질 네트워크 쌍에 대하여 비교한 결과가 자세히 논문에 나와있다. Ref-[51]와 더불어 가장 최신의 결과가 제시되어 있다. http://ttic.uchicago.edu/~hashemifar/software/HubAlign.zip을 통해서 모든 자료를 다운받을 수 있다.

6.5.9 DualAligner

전역정렬과 지역정렬의 구분되는 단점을 극복하기 위하여 confidence level이 낮은 쪽과 높은 쪽에 각각 다른 cost function을 주는 방법이 개발되었다[53]. 기존의 전역정렬은 억지로 모든 노드가 들어가게 하고, 이에 반해서 지역정렬은 유사도 값이 낮은 부분을 정렬에서 완전히 빼는 단점을 가지고 있다. 그러나 이 시스템에서는 각 노드의 생물학적 중요도, 예를 들면 GO DB등을 통하여 orthologous한 정도에 대한 신뢰도를 따로 산정하여 이 값에 따라서 matching 비용함수를 adaptive하게 조정해주는 기능을 가지고 있어 전역도 지역도 아닌 정렬을 가능하게 한다. 예를 들어 우리는 N_1과 N_2를 정렬하고자 할 때 이 두 네트워크 안에 이미 아주 유사한 subnetwork인 $n_x \in N_1$과 $n_y \in N_2$가 존재한다면 이 둘 n_x, n_y을 하나의 단위로 묶고 이 둘의 유사성 신뢰도를 아주 높게 주는 것이다. 이렇게 설정을 하면 n_x의 노드 대부분은 n_y의 노드와 매칭을 하게 되어 정렬의 결과에서도 n_x와 n_y 서로 align이 되게 될 것이다. 이것은 이미 우리가 알고있는 사실과도 부합되는 것이다. 만일 이런 신뢰성 값 조절이 없다면 n_x의 노드들은 n_y가 아닌 다른 곳에 정렬이 될 수 있고, 이는 우리가 원하는 결과가 아닐 수 있다. 연구자들은 이것을 "쌍대 정렬(Dual Alignment)"이라고 명명하였다. 연구자들의 주장에 의하면 이 방법은 기존의 정렬방법보다 성능면(정확도와 속도)에서 더 우수하다고 한다. 요약하자면 이미 알려진 정보를 활용하여 정렬문제의 탐색 공간을 좀 더 축소시켜서 성능향상을 꾀하는 방법이라고 볼 수 있다. 한 가지 문제는 이것을 위해서 다양한 조절 매개변수(control parameter)들을 사용자가 일일일 결정해야 한다는 점이다. 이 시스템은 다음을 동해서 접근할 수 있다. http://www.cais.ntu.edu.sg/

6.6 Graphlet기반의 네트워크 정렬 시스템

이 장에서는 graphlet 기반의 네트워크 정렬 알고리즘인 GRAAL과 GraphCrunch에 대하여 간략히 설명하고자 한다.

6.6.1 GRAAL (GRAph ALigner)

우리는 앞서 그림-13를 통하여 30개의 graphlet에 대하여 살펴보았다. 다시 정리하면 graphlet은 노드 5개 이하의 모든 subgraph 중에서 연결된(connected) component를 말한다. 그리고 30개의 graphlet에서 각각 서로 다른 73개의 automorphism orbit에 대해서 살펴보았다. $GDV(v)$는 특정 vertex v에 걸쳐있는 73개의 graphlet의 갯수를 표시한 73차원의 vector이다. 이 방법으로 두 그래프 $G(V,E)$와 $H(U,F)$의 정렬이 가능하다. 먼저 어떤 두 노드 $u \in G, v \in H$를 정렬할 graphlet 기반의 비용함수 $C(u,v)$는 다음과 같다.

$$C(u,v) = 2 - (1-\alpha) \cdot \frac{deg(u) + deg(v)}{maxdeg(G) + maxdeg(H)} + \alpha \cdot S(u,v)$$

여기에서 $deg(x)$는 노드 x의 차수(degree)를 나타내고 $maxdeg(G) = \max_{x \in G} deg(x)$를 나타낸다. 그리고 $S(u,v)$는 두 벡터 $GVD(u)$와 $GVD(v)$의 거리를 나타낸다. 여기서 매개변수 α는 두 벡터의 거리 $S(u,v)$와 비교하는 두 정점의 차수 차이 에 따른 상대적 중요도를 조절하는데 사용된다. 즉 만일 이 값이 1이면 우리는 두 정점의 graphlet의 벡터 거리만을 고려하게 되고 $\alpha = 0$이면 graphlet에 대한 고려없이 degree 차이만을 고려하게 된다. 예를 들어 비교대상 정점의 degree가 모두 2이고 두 그래프의 최고 차수가 10이라면 이 두 정점의 중요도는 2/10=1/5=0.2로 계산된다. $C(u,v)$ 값은 0과 1사이로 정규화되어 있다.

GRAAL은 먼저 두 그래프 G와 H에서 특정 두 노드 $u_0 \in G, v_0 \in H$를 시작점으로 선택함으로 시작된다. 보통 이 두 점은 $C(u,v)$ 비용이 가장 작은 쌍이 선택된다. 다르게 표현하자면 가장 graphlet 분포로 볼 때 가장 유사한 쌍이 시작점으로 선택된다. 이 과정에서 tie cost가 많이 발생할 수 있는데, 그때는 랜덤하게 선택한다. 이 tie break 선택과정이 GRAAL 알고리즘의 안정성에 도움을 준다. 그 다음에는 앞서 정렬 알고리즘의 기본구조에서 설명한대로 각 중심에서 거리 r 안쪽에 존재하는 이웃지역(sphere)를 확정한다. 그리고 그 "sphere" 안에 존재하는 모든 쌍의 노드를 비교해서 그 안에서 가장

작은 $C()$ 비용의 새로운 쌍(정렬에 이미 포함되지 않은)을 선정한다. 이 greedy 과정을 일정한 갯수의 노드들이 정렬될 때까지 반복한다. GRAAL은 이 과정에서 처음에는 두 노드의 거리가 1인 쌍들끼리만 비교하다가, 그 다음에는 그 거리를 2, 3으로 늘여나간다. 즉 아래 그림-32과 같이 진행해가면서 상호비교하는 path의 길이를 1,2,3으로 늘려가고 이때 서로 다른 길이의 path가 align될 경우에는 gap node를 새로 추가하여 align한다. 그리고 이 작업을 다른 seed pair(같은 cost값을 가진 다른 pair)에 대해서 반복적으로 수행하여 가장 많은 쌍의 노드가 align되는 결과를 답으로 출력한다.

GRAAL의 성능에 대한 분석은 참고문헌[14]을 보면 좋다. 연구에 의하면 다른 seed로 시작한 정렬의 경우에도 약 60%d 정도는 최종적으로 같은 정렬 결과를 만들었다고 하는데 이 말은 GRAAL의 성능이 상당히 안정적이라는 것이다.

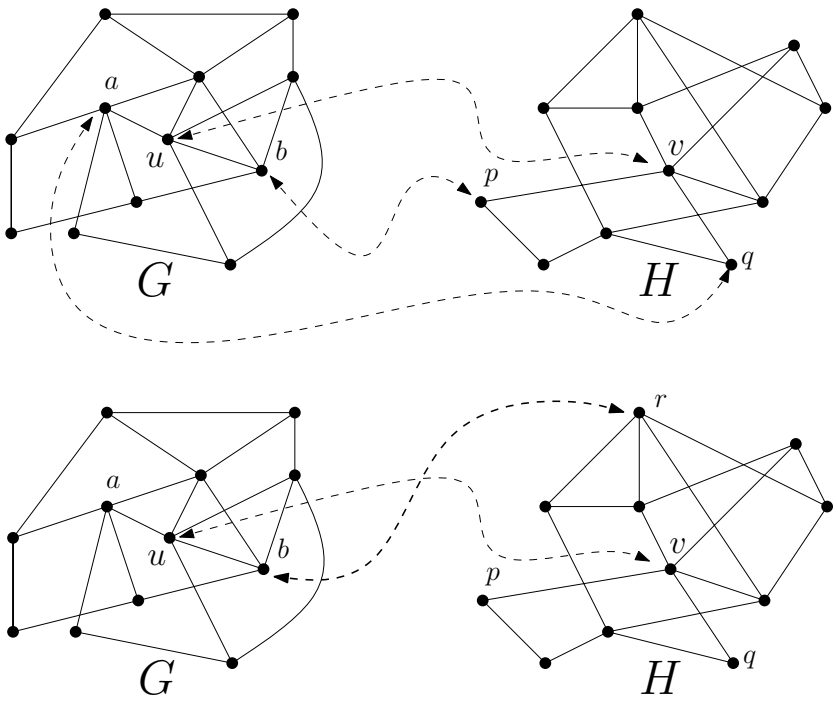


Figure 32: GRAAL에서 중앙 노드에서 길이를 확장하여 서로 다른 길이의 path를 비교하는 과정. 첫번째 그림은 길이가 1인 path(edge)에 대해서만 비교한다. 그 다음은 길이 2 이하의 모든 path 후보에 대하여 비교한다. 즉 align되지 못한 쌍 중에 가장 유사한 쌍을 선택해서 그것과 중앙 seed(여기에서는 u와 v)까지를 거리를 비교해서 그곳에 도달하는 path중에서 가장 유사한 비용점수를 가진 path를 서로 매칭시킨다.

6.6.2 MI-GRAAL

앞서 graphlet 기반의 GRAAL 시스템에서 노드 별 유사도를 비교할 때, 해당 노드에서 일정거리 이하의 subgraph안에서의 위상적 유사도와 서열 유사도를 같이 고려할 수 있도록 확장하였다. 사용자는 위상적 유사도와 서열적 유사도 중에서 그 중요도를 적절히 조절할 수 있다. 정렬 시스템에서 서열 유사도를 완전히 무시하고 위상적으로만 정렬할 수 있는, 즉 완전히 그래프 매칭문제로만 접근이 가능한 최초의 시스템으로 평가받고 있다. 이 때문에 다종의 PPI network에서 그들을 하나의 계통나무(Phylogenetic Tree)로 보여주는 시스템 중에서 가장 안정적인 시스템으로 평가받고 있다.

6.6.3 C-GRAAL

GRAAP family에서 가장 최근에 공개된 나온 시스템이다. MI-GRAAL과는 다르게 서열 유사도와 graphlet vector distance만 정렬에 사용하는 시스템이다. 평가지표 중에서 EC measure로 본다면 MI-GRAAL보다 우수하다. 그 이유는 시작 seed에서 extend가 어렵다고 판단되는 기준을 조절할 수가 있어 만일 그런 경우에는 다른 seed로 옮겨가서 새로 matching 작업을 시도하는 방식을 활용하기 때문이다.

6.6.4 GraphCrunch 2

GraphCrunch 1은 주어진 네트워크를 임의의 Random Network와 비교해서 어떤 특성이 통계적으로 유의미한지를 보여주는 도구이다. 이 GraphCrunch 1 시스템을 확장하여 구현한 GraphCruch 2는 두 그래프의 전역, 자역 정렬까지 가능한 도구이다[15]. 이 도구는 GRAAL의 방법론을 차용하여 두 그래프의 Graphlet signature 유사도를 기반으로 그래프의 유사성을 매칭 노드의 생물학적 유사성에 대한 고려없이 계산해준다. 속도가 빠르고 병렬화 시스템에서도 구동시킬 수 있는 장점이 있다. 이 도구는 생물 네트워크가 아닌 데이터에도 쉽게 활용할 수 있는 범용성을 가진 것이 장점이다.

7 정렬 시스템의 성능 비교평가

이 장에서는 앞서 소개한 다양한 네트워크 정렬 시스템의 성능비교 결과를 요약한다[41]. 비교된 도구는 NETAL, GRAAL, GHOST, PINALOG, C-GRAAL, NetAlign2.0, SPINAL mode1, SPINAL mode2, MI-GRAAL, IsoRank이다. 그리고 끝으로 이러한 연구를 위하여 준비된 다양한 데이터 베이스를 소개한다. 가장 기본적인 데이터베이스인 서열정보기반의 DB를 비롯하여 각종 생물 네트워크, pathway가 관리되고 있는 DB를 소개한다. 그리고 annotation이 잘 정리된 model 생물용으로 활용되는 대표적인 DB 몇 종도 같이 소개한다.

7.1 네트워크 정렬 시스템의 성능 비교

최근 아주 흥미로운 논문이 발표되어 연구자들의 관심을 끌었다[41]. Clark등은 실제 생물 네트워크와 평가용으로 만든 인공 네트워크 데이터를 사용하여 공개된 주요 정렬 알고리즘에 대한 상호 비교하였다. 놀라운 사실은 각 도구가 공개한 결과와 달리 실제 성능의 차이는 심하다는 점이다. 그리고 각 도구의 실제 인터페이스도 사용자 중심이 아니라 개발자 중심으로 불편하게 되어있다는 사실도 지적되었다. 이 연구결과가 흥미로운 이유는 그 이전에는 어떤 연구자도 정렬 시스템에 대한 공개적인 평가를 시도해보지 않았기 때문이다. 지금까지 개발된 시스템관련 논문의 주 내용은 새로 개발된 시스템 성능을 이전의 몇 시스템과 비교하여 그 우수함을 확인하는 것이었다. 이런 논문은 평가의 객관성에 약간의 문제가 있을 수 있다. 예 를 들어 자신이 만든 시스템은 여러 단계의 최적화를 거치는데 비하여 비교 대상인 다른 시스템은 제공된 default값만으로 수행하게 한다든지, 평가용 데이터을 사용하여 약간의 overfitting 사전 작업을 하여 성능을 돋보이게 하는 것이 주로 문제가 될 수 있기 때문이다. 이 때문에 제 3자가 공개적으로 평가한 이런 연구결과가 객관성과 공정성의 면에서 가장 의미있다고 할 것이다.

7.1.1 Benchmark Simulation Data 평가

NEPABench에서 확보한 시뮬레이션 데이터로 앞에서 설명한 도구 NETAL, GRAAL, GHOST, PINALOG, C-GRAAL, NetAlign2.0, SPINAL mode1, SPINAL mode2, MI-GRAAL, IsoRank를 평가한 결과가 있다[41].

단순한 EC measure아 아닌 ICS3 measure로 볼 때 가장 우수한 성능을 보인 것은 NATALIE였으며 그 다음 순위는 GHOST, SPINAL, MI-GRAAL 였다. 그외 C-GRAAL, NETAL, PINALOG는 다들 비슷한 성능을 보여주었다. 그런데 이것은 NATALIE의 align방법이 ICS measure에 유리하기 때문이라고 판단된다. 왜냐하면 NATALIE는 노드 matching에서 어중간한 값들의 쌍은 아예 정렬결과에 포함시키기 않기 때문에 matching pair는 적다. 그리고 이 적은 갯수가 ICS 공식에서 분모로 쓰이게 때문에 적은 수의 확실한 매칭 쌍만을 선로하는 NATALIE는 ICS면에서 보면 유리할 수 밖에 없다.

만일 N_a를 N_b로 정렬할 때 (단 $|N_a| < |N_b|$) N_a의 노드 중에서 align된 N_b의 공통 subgraph에 들어간 노드 비율을 살펴보는 것도 성능평가에 의미가 있다. 물론 이것이 ICS와도 관계가 있지만, 이 비율이 높다는 것은 가장 큰 shared component를 잘 찾아낸다고도 볼 수 있기 때문이다. 이 관점으로 평가결과 MI-GRAAL, SPINAL, GHOST가 큰 덩치의 공통모듈을 찾는데 상대적으로 우수하다고 판명되었다.

생물 네트워크를 정렬하는 가장 중요한 목적은 다른 종간의 orthologous한 유전체 단위를 찾는 것이라고 할 수 있다. 이미 motif가 잘 알려진 네트워크를 그렇지 않은 종의 내트워크와 정렬하여 새로운 morif에 대한 정보를 얻으려는 첫 단계가 바로 네트워크 정렬이기 때문이다. 그런데 시물레이션으로 만들어진 인공 NAPA 데이터는 미리 각 네트워크에 확실한 orthologous쌍을 미리 넣을 수 있기 때문에 성능평가가 쉽고 명확하다. NAPA 데이터 평가결과 NATALIE가 정렬한 쌍은 99%라는 놀라운 정확도로 orthologus한 pair를 모두 찾아냈다. 다르게 말하자면 NATALIE는 sensitivity는 좀 떨어지지만 specificity는 매우 높다고 할 것이다. 그 아래 단계에서는 70%-80%의 정확도의 SPINAL, PINALOG 시스팀이 있다. 한편 IsoRank나 MI-GRAAL은 그보다 한참 떨어진 성능을 보였다. 그 중 GRAAL이 orthologous한 쌍을 찾는데에는 가장 낮은 성능을 보였다[62].

7.1.2 IsoBase 기반의 실제 데이터 평가

인공적인 NAPAbench 데이터가 아닌 실제 wet Lab 데이터의 평가결과는 앞과는 사뭇 달랐다. 평가용 네트워크으로 선택된 것은 꼬마선충, 초파리, 효모 그리고 인간 단백질 네트워크이다. 실제 Isobaase 데이터를 이용한 성능분석이 앞서 시물레이션 데이터의

[62]이것은 graphlet기반의 탐색은 생물학 서열적 우사도를 전혀 고려하지 않기 때문으로 생각된다.

그것과 가장 다른 점은 각 시스템의 성능 범위(span)가 매우 넓게 나왔다는 점이다. 동시에 전체적인 성능이 NAPA data와 비교해서 상당히 떨어진다는 것이다. 그러나 실제 IsoBase의 정답도 확실하지 않는 상황이라 성능평가는 실제 데이터에 존재하는 본원적 잡음과 불완전성을 감안해야 할 것이다.

Isobase를 사용한 평가에서 ICS metric 기준으로 유의미하게 성능 차이를 보인 것은 NETAL이다. 그 다음 순위인 NATALIE나 GRAAL family의 성능은 대부분 비슷했다. 앞의 인공 NAPA 데이터 평가와 달리 GRAAL family가 상당히 나은 성능을 보인 것이 특이한 점이라고 할 것이다. 이것은 실제 네트워크 데이터에서 각 노드의 기능은 주위 이웃 노드의 영향을 직접적으로 받고 있다는 것을 암시해주고 있다.

LCSC(Largest Connected Shared Component)로 평가했을 때에도 NETAL이 가장 우수했다. . 초파리와 인간 네트워크를 NETAL로 정렬했을 때 초파리 네트워크 노드의 90%, 에지의 30%가 인간의 네트워크와 정렬될 수 있음을 보인 점이 주목할만하다. PINALOG나 SPINAL, GRAAL, NATALIE등은 LCSC관점에서 볼 때 각각 떨어져있는 작은 subgraph 조각만을 정렬시켜주는데 그쳤다. 이걸로 볼 때 실제 데이터에서 PINALOG나 SPINAL, GRAAL, NATALIE로는 LCSC를 탐색할 수 없을 것이라고 판단된다. 결론적으로 NETAL은 크기가 큰 subnetwork을 정렬해주는데 가중 우수한 도구라고 보인다. 특히 잡음이 많은 실제 wet lab data에서 NETAL은 강점을 가진다고 판단된다. 정리하자면 인공 데이터와는 달리 실제 IsoBank 데이트의 평가결과 NETAL이 위상에 중점을 둔 정렬에는 가장 신뢰성 높은 도구라고 보인다.

7.2 네트워크 정렬 시스템의 성능평가 종합

가상의 데이터와 실제의 데이트를 사용한 점수를 동시에 고려한 분석을 해본다. 앞서 설명한 GOC 점수 기준으로 보았을 때 PINALOG와 SPINAL이 가장 우수한 반면, ICS에서 우수한 성능을 보인 NETAL의 성능이 최하위임은 흥미로운 결과다. 그리고 GHOST의 성능도 PINALOG나 SPINAL에 비해서 현저하게 낮았다. GO거리는 결국 GO DB를 설계할 때 서열적 유사도와 설험을 통한 연관성을 이미 고려하였기 때문에 위상만을 보는 시스템의 성능은 대부분 GOC 기준으로 나쁘게 나타났다.

한 가지 특이할 점은 꼬마 선충과 다른 종의 네트워크를 비교했을 경우 도구별 차

이없이 대부분의 도구들이 매우 낮은 GOC 점수를 보여주었다는 것이다. 그 원인은 선충의 단백질 네트워크의 밀도가 매우 낮기(sparse) 때문으로 보여진다. 이 사실은 비교할 두 대상의 네트워크 밀도나 특성이 어느 정도 유사해야만 도구별 성능의 차이와 함께 의미있는 결과도 도출된다는 것을 암시해주고 있다. 그런데 GO DB의 설계시 이미 서열의 유사가 고려되었기 때문에 서열의 유사도 정렬을 수행한 결과를 다시 GOC 점수로 평가하는 것에는 이중계산의 문제가 있다. 어떻게 보면 이 일은 새로운 정보의 추가없이 같은 작업을 반복하는 것에 불과하다. 따라서 GO term중에서 실험으로 확인된 GO term만을 추려서 이것들의 거리로만 GOC 점수를 계산하는 것이 더 의미있는 일이라고 판단된다. 이 새로운 GO term 기준으로 평가해 본 결과 거의 모든 도구에서 매우 낮은 GOC 점수가 나옴을 확인할 수 있었다. 이 말은 실제 네트워크에서 위상만으로 기능적 유사도를 추측하는데에는 한계가 았다는 것을 보여주는 것이다.

전체적인 성능을 볼 때 NEPA 데이터에서는 NATALIE가 안정적이며 우수한 성능을 보여주었지만 실제 wet data 인 ISObase 네트워크에서는 그 우위를 보여주지 못했다. 그것은 NEPA 데이터는 edge의 손실없이 매우 깔끔하게 설계된 반면 실제 Isobase 데이터에서는 많은 노이즈가 포함되어 있기 때문이다. 또한 NEPA 네트워크들은 비교 대상들이 모두 유사한 그래프 밀도와 특성을 가지고 있기 때문에 비교 성능이 잘 나온 것으로 보인다. 총론적으로 볼 때 실제 현장에서 사용할만한 수준의 도구는 SPINAL, NATALIE, PINALOG 정도로 판단되며 GHOST도 안정적인 성능을 보여주었지만, 이 도구는 메모리 요구량이 과대하여 자주 중단되는 문제가 있었다.

GRAAL 패밀리 도구들은 각 노드에서 graphlet을 계산하는 과정에서 너무 많은 시간과 resource를 사용하는 것으로 인하여 자주 다운(crash) 되는 현상을 보였다. 그리고 GRAAL 과 IsoRank는 이미 이들의 성능을 능가하는 다른 도구들이 충분히 있다는 것을 확인할 수 있기 때문에 추천할만한 도구는 아니라는 것이 Clark의 결론이다.

7.3 단백질 네트워크 데이터베이스 소개

생물 네트워크의 가장 대표적이며 오랫동안 연구된 네트워크는 단백질 상호작용 네트워크(PPI)이다. 각 종별 다양한 PPI 네트워크가 다양한 실제 실험을 통하여 오랫동안 축적되어왔고 이를 바탕으로 많은 연구가 진행되었다. 그들을 나열하면 다음과 같다.

7.3.1 IsoBase

앞서 설명한 비교평가용 데이터의 출처 데이터 베이스이다[54]. 5종 생물인 이스트, 초파리, 선충, 생쥐, 인간에게서 연구된 단백질 상호작용 네트워크에 대한 가장 최신의 정보가 모두 정리되어 있다. 개발자들은 동원할 수 있는 모든 단백질 orthologus 추측 도구를 이용하여 확인할 수 있는 모든 단백질들간의 관계를 표시해두었다. 특히 단백질들의 기능적 유사성에 중점을 두고 annotation이 정리되어 있다. 그 결과 모두 48,120 종의 단백질을 12,693개의 기능적 클러스터로 구분해두고 있다. 또한 2종 이상의 모델 생물에서 공통적 존재하는 기능이라든지 또한 한 종에만 특별히 존재하는 단백질도 찾아볼 수 있다. 이 DB는 `http://isobase.csail.mit.edu/`를 통해서 다운받을 수 있다.

7.3.2 Biomolecular Interaction Network Database (BIND)

단백질 상호작용과, 단백진 분자 복합체, 그리고 그들의 pathway가 가장 완벽한 수준으로 관리되고 있다. 확장된 BIND 2.0는 단백질외 다른 염기서열(segment) 등과의 상호작용에 대한 정보까지 포함하고 있다. 이 데이터 베이스는 다음을 통해서 접근할 수 있다. `http://www.bind.ca/`

7.3.3 Database of Interacting Proteins (DIP)

단백질의 상호작용을 실험적으로 검증하여 기록한 데이터 베이스이다. 중요한 점은 이 정보는 모두 연구자들의 수작업으로 통하여 검증된 것이라는 점이다. 중간과정은 다양한 소프트웨어를 통하여 추출되었지만 최종 등록을 위해서 전문가들의 확인을 일일이 거친 것으로 가장 신뢰성이 높은 상호작용 정보를 가지고 있다고 알려져 있다. 아래를 통하여 접근할 수 있다. `http://dip.doe-mbi.ucla.edu/`

7.3.4 IntAct

여러 일반사용자들이 Web을 통하여 쉽게 단백질 반응 정보를 저장하고 검색할 수 있도록, 사용자별로 특화된 Web Interface를 제공하는 DB이다. 특이할 점은 상호작용 단백질의 내용 정보를 다양한 그래픽을 통하여 가시화(visualization) 해주고 있다는 점이다. 이곳의 Web 인터페이스는 다음을 통하여 볼 수 있다. `http://www.ebi.ac.uk/intact`

7.3.5 Biological General Repository for Interaction Datasets (BioGRID)

전문가 집단에 의해서 실험결과가 정리된(curated) 또 다른 단백질 작용 데이터베이스이다. 단백질간의 상호작용 뿐 아니라 유전자간의 상호작용에 대해서도 같이 기록되어 있다. 대부분의 모형 개체(model organism)에 대한 자료도 관리되고 있으며 같은 결과는 최대한 배제하려는 시도를 통하여 중복 데이터가 많이 정리되어 있다. 자료는 다음을 통하여 접근할 수 있다. http://www.thebiogrid.org/

7.3.6 Molecular INTeraction database (MINT)

이 DB는 실제 실험을 통하여 검증된 결과를 Wet biologist가 정리한 것이 아니라 이미 다른 논문이나 자료집에서 발표된 결과로 부터 유추할 수 있는 단백질 상호작용에 관한 자료를 저장해둔 일종의 meta-data라고 할 수 있다. 다시 말해서 1차 실험결과가 아니라 text-mining을 통하여 정리된 2차 가공 단백질 상호작용 데이터들을 담고 있다. 해당 사이트는 다음에 있다. http://mint.bio.uniroma2.it/mint/

7.3.7 MPact

이 데이터 베이스는 이스트(S. cerevisiae)의 단백질 상호작용에 특화된 데이터베이스이다. 전체 이스트 단백질의 상호작용 결과는 PSI-MI 형식을 통하여 다음 사이트에서 얻을 수 있다. http://mips.gsf.de/genre/proj/mpact

7.3.8 International Molecular Exchange Consortium (IMEx)

이 콘소시움은 단백질 상호작용 연구의 중복과 자원낭비를 위하여 DIP, IntAct, BioGRID, MINT, and MPact을 관리하고 있는 기관들이 연합한 일종의 콘소시움이다. 이 콘소시움을 통하여 실험적 검증에서 중복실험이나 검증작업의 비효율성을 방지한다. 그리고 검증할 단백질 군을 나눠서 각 기관별로 배분하는 역할도 한다.

7.4 경로정보(PathWay) 데이터베이스 소개

특정한 기능의 pathway에 관한 정보만을 따로 관리하는 Database의 종류도 다양하다. 일반적인 다차원의 생물 네트워크에 비해서 선형적인 tree 형식의 기능별 pathway는

실험과 검증이 용이하여 위에서 설명한 네트워크 더 자주 활용된다. 이들 중 활발하게 활용되는 몇 가지를 소개한다.

7.4.1 Kyoto Encyclopedia of Genes and Genomes (KEGG)

KEGG는 pathway관련 연구에서 빠질 수 없는 가장 중요한 정보 소스로서 일본에서 관리하고 있다. 이 온라인 DB는 유전체, 각종 효소들의 pathway, 다양한 생화합물들의 반응 path를 체계적으로 관리하고 있다. 사이트의 위치는 http://www.genome.ad.jp/kegg/ 이며 대사적용, 유전학, 세포내 작용, 인간질병, 신약 개발에 관련된 pathway 정보를 총망라하고 있다는 점에서 아주 의미가 크다. 보통 연구자들이 어떤 pathway에 대한 연구를 시작할 때 가장 먼저 찾아보는 것이 바로 KEBB DB이다. 또 KEGG Orthology(KO)에 대한 표준안을 제정하여 pathway 표현의 국제적 표준형을 관리하고 있다는 점에서 중요한 의미를 가진다. 상당한 전문인력과 노력이 들어간 매우 가치있는 DB라고 할 수 있다.

7.4.2 BioCyc

모델별 pathway 정보를 관리하고 있는 DB로서 그 내부에는 EcoCys과 MetaCys DB로 구성되어 있다. 각각은 앞서 MINT DB와 같이 문헌정보를 이용해서 2차 가공한 자료를 가지고 있다. 위치는 http://biocyc.org/에 있다.

7.4.3 Netpath

이 DB는 인간 세포내에서의 신호 전달(signal transduction)[63] 중에서 대표적인 10개의 면역체계, 암 발병 체계에 관한 pathway 정보를 가지고 있다. 위치는 다음과 같다. http://www.netpath.org/. 각 정보는 PPI, 촉매반응, 전좌(translocation[64], 그리고 특정 리간드에 따라서 다르게 발현되는 유전자 정보를 포함하고 있다.

[63]세포가 외부의 신호를 수용하여 그에 대응하는 세포 기능이 발현되기까지 정보를 전달하는 과정. 세포내 신호 전달 과정에는 다수의 효소가 단계적으로 활성화되면서 정보가 증폭되기도 함.

[64]염색체 일부가 같은 염색체의 다른부분으로 위치가 변화하거나, 또는 다른 염색체 상에 위치로변화하는 염색체의 이상현상. 특히 2개의 비상동 염색체에서 절단이 일어나서 그 부분을 교환하는 것.

7.4.4 Reactome

이 DB는 인간 세포내의 생물학적 전과정에 관한 자료를 관리하고 있다. 각 자료는 여러 연구자들이 이중으로 검증하고 있다. 관련 자료는 http://www.reactome.org/에서 찾아볼 수 있다. 주로 인간에 관한 자료들이 있지만 그외 다른 종에 대한 자료도 많이 포함하고 있다.

7.4.5 NCI-Nature Pathway Interaction Database

이 DB는 인간 세포내의 다양한 신호전달 pathway에 관련된 자료를 가지고 있다. http://pid.nci.nih.gov/는 특히 인간의 암과 그 치료에 관련된 여러 분자단위의 상호작용에 대한 것으로 특화되어 있다.

7.5 유전제 서열정보 데이터베이스 소개

이 장에서는 주요 기본서열 정보용으로 알려진 DB에 대하여 소개한다.

7.5.1 GenBank

미국 NCBI에서 운영하는 유전자 은행이다. 동시에 국제적 연구협의체인 International Nucleotide Sequence Database Collaboration(INSDC)와 공동으로 운영되고 있으며 http://www.ncbi.nlm.nih.gov/genbank/에 있다. 대부분 종의 DNA, 단백질 서열이 망라되어 있다. 또한 새롭게 등록될 새로운 서열을 기존의 서열과 비교하여 가장 최신의 서열정보가 계속 update되고 있다.

7.5.2 UniProt

다양한 연구목적을 위하여 구성된 단백질 DB이다. 스위스 생물정보 연구소(SIB)에서 관리하는 SWISS-Prot과 연계되어 운영되고 있으며 모든 단백질 관련 DB의 중심축 역할을 하고 있다. 이 DB는 다음 사이트를 통해서 사용가능하다. http://www.uniprot.org/.

7.5.3 DNA Data Bank of Japan (DDBJ)

일본 국립 유전학 연구소(Institute of Genetics (NIG) in the Shizuoka)에서 운영하는 생물정보 DB로서 다음 위치에서 접근할 수 있다. http://www.ddbj.nig.ac.jp/

7.5.4 EMBL Nucleotide Sequence Database

유럽 생물정보학 연구소(European Bioinformatics Institute, EBI)에서 관리하는 염기서열 저장소이다. 해당 데이터베이스는 다음을 통해서 접속할 수 있다. ttp://www.ebi.ac.uk/embl 이다.

7.6 주요 모델 생물 유전체 데이터베이스 소개

앞서 설명한 생물 네트워크나 pathway와 같은 고차원의 상호작용 DB와 더불이 가장 저차원의 생물 서열에 관한 DB들도 생물 네트워크 연구에 중요하다. 이들을 나열하면 다음과 같다. 생물 네트워크 정렬 시스템을 새로 만들거나 평가하는데도 가장 중요한 자료는 바로 꼬마선충, 애기장대, 초파리와 같은 모델 생물의 결과와의 비교작업이다. 따라서 모델 생물용 DB는 모든 생물학 연구에서 가장 중심적인 역할을 한다.

7.6.1 Flybase

대표적인 모델 생물인 초파리(D. melanogaster)에 관한 종합정보를 제공하는 DB로서 다음 위치에서 접근이 가능하다. http://flybase.bio.indiana.edu/ 초파리 염색체에 관하여 complete annotation이 관리되고 있으며 또한 초파리 관련 다른 DB, 문헌정보등이 모두 망라되어 있다.

7.6.2 The Arabidopsis Information Resource (TAIR)

식물의 대표적인 모델 생물인 애기장대(Arabidopsis thaliana)의 유전학적 정보가 총정리된 대표적인 단일 종 DB이다. 저장된 것은 annotation된 전유전체(whole genomes) 서열, 유전자 정보, 각 유전자들의 생산물 정보, 유전적 변이, 유전자 expression, 각종 마커

(marker) 정보 등이다. DB는 다음에서 접근이 가능하다. http://www.arabidopsis.org/

7.6.3 Mouse Genome Database (MGD)

생쥐에 관한 모든 유전학적 정보, 개별 분자유전 수준의 정보, 그외 모든 생물학정인 정보가 망라된 DB다. http://www.informatics.jax.org/. 그리고 생쥐의 유전자와 다른 많은 포유류의 유전자 중에서 othologous한 유전자들이 잘 정리되어 있다. 그리고 제공되는 데이터와 관련된 다른 DB 정보가 편리하게 hyperlink로 연결되어 있어서 DNA 서열, 단백질 서열을 GenBank, EMBL, DDBJ, SWISS-PROT 등과 쉽게 결합하여 볼 수 있도록 해주고 있다. 또한 의생물학 문헌 DB인 PubMed와도 잘 연계되어 있다.

7.6.4 Rat Genome Database (RGD)

쥐는 현재까지 인간의 질병을 연구를 위한 대표적인 모델 생물이므로 질병연구나 신약 개발에 매우 중요하게 쓰이고 있다. 이 사이트는 쥐(Rat)의 분자생물학적 모든 정보가 정리된 DB 사이트로서 http://rgd.mcw.edu/를 통해서 접근이 가능하다. 내용은 쥐의 genomic, genetic, functional, physiological, pathway을 포함하고 질병에 관한 정보까지 모두 제공한다.

7.6.5 Saccharomyces Genome Database (SGD)

발효 이스트(효모)의 서열, 유전자, 그리고 각 유전자들의 부산물 등을 포함하고 있는 DB로서 http://www.yeastgenome.org/를 통하여 접근이 가능하다.

8 결론과 제언

우리는 앞에서 서열정렬부터 시작하여 그래프 매칭, 그래프 정렬의 다양한 문제과 관련된 이슈들, 그리고 공개된 여러 시스템에 대해서 살펴보았다. 또 그들의 성능을 가공의 데이터와 실제 데이터를 사용하여 비교평가한 결과도 설명했다. 끝으로 지금까지 설명한 내용을 요약하고 앞으로의 연구방향을 제언하는 것으로 보고서를 마감하고자 한다.

8.1 요약: 생물 네트워크 정렬의 최근 동향

현재까지 진행된 생물 네트워크 관련 연구내용을 요약하면 다음과 같다.

1. 기존의 선형 서열분석으로 파악할 수 없는 새로운 분자생물학적 지식의 탐구에 생물 네트워크 연구는 필수적이다. 미래 생물학의 중심이 될 시스템 생물학(Systems Biology)은 결국은 네트워크 연구로 수렴할 것이다.

2. 현재 생물 네크워크에 관한 기본 데이터가 부족하고, 또한 준비된 데이테에도 많은 오류와 잡음이 섞여있어 이런 문제는 네트워크 분석에 주요한 장애가 되고 있다.

3. 그러나 새로운 분석기술과 대단위 공학적 규모의 실험이 가능해짐에 따라서 생물 네트워크의 양과 질은 상당히 개선될 것이다.

4. 생물 네트워크 정렬은 생물 단위체의 미지의 기능을 탐지하는데 가장 중요한 도구가 되고 있으며 계통분석과 진화연구에 가장 중요한 역할을 하고 있다. 또한 기존의 서열 분석만으로 알아낼 수 없는 고차원적인 지식을 제공하는데 가장 핵심적인 방법론이 될 것이다.

5. 예를 들어 질병분석과 신약개발에 생물 네트워크 분석은 가장 중심적 역할을 하게 될 것이다.

6. 네트워크 분석 문제의 원형은 그래프 동일성 검사(graph isomorphism)라는 전형적인 intractable NP-complete문제이므로 본질적으로 이 문제를 다항시간에 최적으로 풀 수 있는 알고리즘이나 시스템은 존재할 수 없다. 따라서 다양한 응용분야에 적합한 휴리스틱 알고리즘과 시스템의 개발이 현실적인 대안이며 시급히 필요하다.

7. 생물 네트워크 정렬의 두 지표는 각 노드(node, vertex)의 생물학적 유사도와 위상적 유사도이며 모든 알고리즘들이 이 두 지표를 적절히 활용하여 정렬을 시도한다. 그 과정에 다양한 최적화 방법이 시도되는데 우리가 인공지능, 산업공학 등에서 접할 수 있는 거의 모든 방법이 시도되고 있지만 어떤 방법도 확실한 우위를 보이지 못하고 있는 실정이다. 특히 위상적 성능이 우수한 정렬 알고리즘은 생물학적 유사도 매칭에 취약하고, 생물학적 유사성 매칭에 안정적인 시스템은 위상적 비교에 취약하다. 이 둘 모두를 우월하게 보장하는 실용적인 정렬 시스템의 개발이 앞으로의 과제가 될 것이다.

8. 각 시스템간의 정확한 성능 비교가 어려운 것은 성능을 확실히 결정할 수 있는 실제 네트워크 데이타가 부족하기 때문이며, 그 네트워크들간의 보존성이라든지 기능적 유사성에 대한 확실한 Ground Truth 가 없기 때문에 객관성이 담보되지 못하고 있는 것이 현실이다.

9. 새롭게 공개되는 정렬 시스템마다 기존의 시스템의 성능과 비교해서 우수하다는 개별의 주장은 공개시험결과 별로 신뢰할만한 것이 아님이 밝혀졌다. 그 내부 성능을 조절하는 다양한 변수가 너무 많기 때문에 공정학고 객관적인 성능비교는 매우 어려우며 또한 각 성능은 사용한 데이터의 품질에 따라서도 크게 달라지는 본원적 불안정성을 가지고 있기 때문에 최상의 정렬 시스템을 찾으려는 노력은 별 의미가 없다고 보인다.

10. 기존의 정렬 시스템과 성능적으로 크게 다르지 않는 유사한 성능의 “another alignment tool”을 다시 개발하는 것은 큰 가치가 없다고 생각된다. 어떤 확기적인 방법, 예를 들어 spectral graph theory를 사용해도 그 계산괴정이 폰 노이만 계산 모형을 따른다면 NP-complete의 한계를 벗어날 수 없기 때문이다.

11. 성능적으로 더 나은 정렬 시스템을 만들려면 각 네트워크 데이터에 대한 부가적인 정보를 얼마나 더 활용할 수 있는가에 달려있다. 주어진 서열정보와 위상정보, 또는 GO db term만으로는 일정이상의 성능을 가진 정렬 시스템을 만드는 것은 거의 불가능하다고 판단된다. 따라서 기존의 DB정보를 폭넓게 활용하는 DB + alignment + other Bio-Knowledge 형의 지식통합형 정렬도구가 만들어져야 할

것이다. 어떤 architecture로 이런 지식통합형 시스템을 만들어야 할 것인지에 대해서는 아직 특별한 아이디어가 제시되고 있지는 못하지만 이런 방법이 아닌 전통적인 그래프 이론기반이나 AI Heuristics, Combinatorial Optimization[65]로는 더 이상의 성능개선은 힘들 것이다. 결국 biological network alignment는 기존의 잘 정렬된, 전문가가 수작업으로 annotation을 하여 실험을 거쳐 검증된 네트워크의 정보를 어떻게 잘 활용하는가에 달려있기 때문에 정렬과 데이터 마이닝이 결합하는 식의 융합적 연구로 진행되어야 할 것이다.

12. 만일 충분히 많은, 검증된 정렬 네트워크 자료가 모인다면 Big Data 기반의 접근도 가능할 것이다.

13. Clark의 평가논문에서 지적한대로 공개된 시스템의 안정성도 중요한 문제이다. 정렬 성능도 성능이지만 여러 사용자들이 쉽게 활용할 수 있고 튼튼한(robust) 시스템을 만드는 것이 현실적으로 훨씬 가치있는 일이라고 생각된다.

14. 서열정렬에서는 이미 많은 component tool[66]이 개발되고 공개되어 있다. 그러나 아직 네트워크 정렬에서는 대부분 완성된 시스템의 형태로만 공개하고 있다. 따라서 각 기능별 모듈을 독립시켜 다른 연구자들이 쉽게 쓸 수 있는 componentware나 tool box 형식의 보급노력도 의미있는 일이라 하겠다.

8.2 제언: 생물 네트워크 연구의 미래 연구주제 및 준비

8.2.1 정렬 적합도 평가지표 개발

본 연구자의 의견으로 네트워크를 정렬하기 전에 두 네트워크가 정렬하기에 얼마나 적절한 상태에 있는지를 평가하는 방법이 필요하다고 판단된다. 예를 들어 $comparability(N_a, N_b)$가 그 정렬계산의 적정성을 나타내는 어떤 평가값이라고 한다면 정렬의 성능($alignscore(N_a, N_b)$)과 이 값과의 관계, 예를 들어 성능과 이 값의 비율이 어떤 상수에 근접한다는 것을 밝힐 수 있다면 생물 네트워크 정렬 분야에서 새로운 전기를 마련할 것으로 기대된다. 아래

[65] 선형계획, 정수계획, Quadratic Programming, Convex Optimization 등

[66] Bowtie같은 서열 정렬용 중요 module 엔진등이 제공되어 새로운 도구를 개발하는 사람은 기존의 시스템이 사용한 엔진을 이용해서 쉽고 빠르게 개발한다.

제시된 식에서 $alignscore_X(N_a, N_b)$는 alignment 도구 X를 사용하여 두 네크워크를 정렬한 성능값을 나타낸다. 그리고 k^*는 어떤 상수값이다.

$$\frac{comparability(N_a, N_b)}{alignscore(N_a, N_b)} \approx k^*$$

앞의 예와 같이 네트워크가 매우 성긴(sparse) 것과 조밀한 것은 어떤 정렬방법을 사용해서 비교하더라도, 또한 어떤 measure를 이용해서 측정하더라도 그 성능이 낮을 수 밖에 없을 것이다. 단백질 서열의 유사도를 측정할 때 두 개체의 진화적 거리에 따라서 다른 행렬함수를 사용한 것과 같이 종간의 진화적 거리에 따라서 adaptive하게 어떤 조절변수를 사용하여 정렬하는 방법이 개발되어야 할 것이다. 특히 IsoBase에서 선택한 실험용 네트워크와 같이 잡음이 심한 경우 성능이 크게 떨어짐을 확인할 수 있는데, 이런 잡음의 정도까지 미리 평가하여 정렬의 적합도 평가에 고려되어야 할 것이다. 이런 평가작업이 없으면 도구별 우수성이나 결과의 정확성에 대한 정량적인 평가는 불가할 것이다.

8.2.2 네트워크 정렬 결과의 실험적 검증강화

정렬도구는 아무리 잘 만들어도 그것이 어떤 새롭고 구체적인 사실을 보여주지 못한다면 가치를 가지지 못한다. 역으로 아무리 허술한 도구라고 그것을 통하여 어떤 생물학적으로 획기적인 사실을 도출할 수 있으면 그 도구도 같이 주목받을 수 있다. 앞서 Clark의 연구에서 보인바와 같이 다른 연구자들이 만든 네트워크 데이터에서 어떤 conserved region이나 흥미로운 사실을 찾아낸다고 해도 그것에서 어떤 구체적인 사실을 검증하지 못하면 생물학적인 가치는 크게 떨어진다. 따라서 한국적 환경으로 볼 때, 기존의 알려진 네트워클 데이터를 사용하는 것 보다는 작은 모델 생물의 서열분석으로부터 시작하여 그 다음 네트워크를 구성하고 이것을 다른 종의 네트워크와 정렬한 뒤 그로부터 생물학적으로 중요한 사실을 밝혀내는 작업이 필요하다고 생각한다. 즉 제대로 돌아가는 하나의 완성된 생물연구 한 사이클을 만드는 것이 도구의 개발에도 긍정적인 역할을 할 것이다. 다르게 말해서 아무리 중요한 conserved region을 찾아내거나 새로운 phylogenetic analysis 결과를 얻는다고해도 그것이 현장 생물학 연구에 어떤 구체적인 사실로 표현되지 않으면 가치는 크게 떨어지기 때문이다. 따라서 실험생물학자, 분자의학 연구자 등과의

협동 연구가 한국적 상황에서 절실하다고 하겠다.

특히 그래프 마이닝[55]에 특화된 연구에 집중하는 것이 기초연구 역량 강화에 도움이 될 수 있다고 본다. 동시에 다양한 machine learning 기법을 동원하여 국내 연구기관과 공동으로 실제 한국인에 관련된 네트워크 연구로 특화할 필요가 있다. 그 동안 한국에서 특정 종에 대하여 full sequencing과 finishing[67]까지 마친 작은 모형 종에 대하여 완전한 network을 개발하고 이런 "우리 네트워크"를 이미 공개된 다른 네트워크와 비교하는 것이 생물학적으로나 전산학(정렬 알고리즘 개발)적으로 가치있는 일이 될 것이다. 바람직할 것이다.

8.2.3 정렬 시스템 안정성 개선

문제는 상당한 도구들이 별다른 조치없이 다운되는 것인데, 개발된 정렬 시스템의 안정성이 개선되어야 할 것이다. 그리고 모든 도구들이 위상적 유사성과 생물(서열적) 유사도를 기준으로 정렬을 하는데 둘 모두의 지표에서 가장 우수한 도구는 없었다. 즉 각 유사도에 특화된 시스템이 항상 존재한다는 것인데 이 모두의 지표에서 우수한 도구를 개발하는 것이 남은 과제라고 할 것이다. 이를 위해서 먼저 생물 네트워크의 표현에 관한 표준안이 마련되어야 할 것이다. 이미 서열 정보에 대해서는 각 데이터베이스별 형식간의 변환에 대한 많은 도구가 개발되어 있다. 이와 같이 생물 네트워크에 대해서도 표준안이 마련되든지 아니면 각 중요 DB별 자료를 교환해서 사용할 수 있는 변환도구(translator)가 개발되어야 할 것이다. 예를 들면 Graph XML과 같이 어떤 도구에서도 쉽게 사용할 수 있고 그 결과를 교환할 수 있는 표준안이 있으면 네트워크 연구는 한걸음 더 나아갈 수 있을 것이다. 주요 네트워크 정렬 도구별 입출력 형식을 변환해주는 도구개발도 흥미로운 연구주제가 될 것이다.

8.2.4 네트워크 정렬 도구의 Social Network으로의 응용

Social Network을 Biological Network과 비교하면 유사한 점이 많음을 알 수 있다. 각 노드의 생물학적 유사도는 Social Network 개인이 가진 국부적 특성(local property), 예를 들어 나이나 성별, 직업 등이 여기에 포함될 수 있다. 그리고 위상적 특성은 각

[67] 유전체 연구에서 finishing은 전체 서열의 annotation까지 마친 상황을 말한다. 즉 각 gene들의 orthologous 정보까지 총 망라한 상황을 말한다.

개인이 친구를 맺고 있는 다른 이웃들과의 관계 그래프로 표현된다. 만일 생물 네트워크 정렬 시스템을 한국사회에서 구성된 Social Graph와 믹국사회의 Social Graph와 정렬할 수 있다면 매우 흥미로운 결과가 기대된다. 만일 유사한 subnetwork이 있다면 그것을 결정하는 것은 local한 특징인지 아니면 전체적인 사회구조상의 관계망인지 분석해보는 것이다. 이 연구의 장점은 새로운 graph aligner를 개발할 필요없이 이미 잘 구현되어 있는 bio-aligner를 사용하면 된다는 것이다. 그 정렬과정에서 다양한 조절변수를 활용하여 공통 subnetwork이 어떻게 변화하는지를 조사해보는 것은 흥미로운 주제가 된다. 여기에 다중정렬 도구도 의미를 가질 수 있다. 만일 한국, 일본, 중국, 이 대표적인 동양 3국의 Social Grpah에서 공통적으로 존재하는 사회관계망이 존재하는지, 만일 존재한다면 그것의 크기는 얼마인지, 그 속에서 가장 중요한 요소는 어떤 것인지 알아보는 것은 사회학, Computer Science, Biological Network Science를 융합하는 새로운 연구주제가 될 것이다. 또한 Social Graph에서 가장 일반적으로 나타나는 Scale-Free network 전용의 graph aligner를 개발하는 것도 흥미로운 연구주제라고 판단된다. 적용 대상은 인간사회가 아니라 다른 물격인 논문이나 상품 등으로 이루어진 network 모형에서도 앞서의 network aligner를 활용하면 재미있을 것이다. 예를 들어 어떤 유행상품이 퍼지는 속도와 지속성이 그 물건 고유의 내재적 속성인지 아니면 Social relation(Topological relation)인지 그 중요도를 따져보는 연구도 좋은 연구 주제가 될 수 있다. 여기에 graphlet 기반의 정렬 도구는 가장 범용적 도구가 될 것으로 기대된다[15, 10].

8.2.5 가상세포와 네트워크 정렬

최근에는 virtual cell과 같이 특정한 생화학적 물과정을 완전하게 모사하려는 시도가 이루어지고 있다. 그리고 유사한 개념으로 genetic network simulator[68]도 개발되고 있다. 정렬된 결과 network을 중심으로 어떤 interactive한 실험이나 simulation을 할 수 있는 alignment + simulatior가 결합된 연구도 바람직할 것이다.

[68] 예를 들면 http://rusty.fhl.washington.edu/ingeneue/, http://www.cs.tut.fi/~sanchesr/SGN/SGNSim.html과 유사한 것들이 소개되어 있다.

References

[1] F. Chung, *Spectral Graph Theory.* CBMS Regional Conference Series in Mathematics, 1996, no. 92.

[2] R. Bapat, *Graphs and Matrices.* Springer-Verlag, 2010, vol. UniversiTexT.

[3] S. Coulomb, M. Bauer, D. Bernard, and M.-C. Marsolier-Kergoat, "Gene essentiality and the topology of protein interaction networks," *Proceedings of Biological Science*, vol. 272, no. 1573, p. 1721–1725, 2005.

[4] M. Golumbic, *Algorithmic Graph Theory and Perfect Graphs (Second edition, Annals of Discrete Mathematics 57. Elsevier, 2004).* Academic Press, 1980.

[5] M. Penrose, *Random Geometric Graphs.* Oxford University Press, 2006.

[6] T. Milenković, W. L. Ng, W. Hayes, and N. Pržulj, "Geometric evolutionary dynamics of protein interaction networks," *Proceedings of Pacific Symposium on Biocomputing(PSB)*, pp. 178–189, 2010.

[7] N. Pržulj, D.-G. Corneil, and I. Jurisica, "Modeling interactome: scale-free or geometric?" *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[8] T. Milenković and N. Pržulj, "Topological characteristics of molecular networks," *Functional Conference of Molecular Networks in Bioinformatics*, pp. 15–48, 2012.

[9] M. Newman, *Networks : An Introduction.* Oxford Press, 2010.

[10] O. Kuchaiev, T. Milenković, V. Memisević, W. Hayes, and N. Pržulj, "Topological network alignment uncovers biological function and phylogeny," *Journal of The Royal Society Interface*, vol. 34, no. 1, pp. 1–47, 2002.

[11] U. Alon, "Network motifs: theory and experimental approaches," *Nat. Rev. Genet.*, vol. 8, no. 6, pp. 450–461, 2007.

[12] N. Pržulj, D.-G. Corneil, and I. Jurisica, "Efficient estimation of graphlet frequency distributions in protein-protein interaction networks," *Bioinformatics*, vol. 22, no. 8, pp. 974–980, 2006.

[13] T. Milenković, W. L. Ng, W. Hayes, and N. Pržulj, "Optimal network alignment with graphlet degree vectors," *Cancer Informatics*, vol. 9, pp. 121—137, 2008.

[14] N. Pržulj, "Biological comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 8, pp. 177–183, 2007.

[15] O. Kuchaiev, A. Stevanović, W. Hayes, and N. Pržulj, "Graphcrunch 2: Software tool for network modeling, alignment and clustering," *BMC Bioinformatics*, vol. 12, no. 24, pp. 1–13, 2011.

[16] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang, "Algorithms for graph similarity and subgraph matching," *Technical Report of Carnegie-Mellon-University*, 2011.

[17] R. Tanaka, "Scale-rich metabolic networks," *Physical Review Letter*, no. 94, pp. 168 101–168 104, 2005.

[18] E. Feller, "Revisiting scale-free networks," *BioEssay*, no. 27, pp. 11 060–11 068, 2005.

[19] C. Song, S.Havlin, and H. Makse, "Self-similarity of complex networks," *Nature*, no. 433, pp. 392—395, 2005.

[20] R. Tatusov, N. Fedorov, J. Jackson, A. Jacobs, . B. Kiryutin, E. Koonin, and D. K. et al, "The cog database: an updated version includes eukaryotes," *BMC Bioinformatics*, vol. 4, no. 1, 2003.

[21] O.Brien, K. Remm, and M. Sonnhammer, "Inparanoid: a comprehensive database of eukaryotic orthologs," *Nucl. Acids Res*, vol. 33, pp. 476–480, 2005.

[22] T. Milenković and N. Pržulj, "Conserved pathways within bacteria and yeast as revealed by global protein network alignment," *PNAS*, vol. 100, pp. 11 394–11 399, 2003.

[23] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, "Græmlin: General and robust alignment of multiple large interaction networks," *Genome Resarch*, vol. 16, pp. 1169–1181, 2006.

[24] G. Wagner, M. Pavlicev, and J. Cheverud, "The road to modularity," *Nature Genetics*, vol. 8, no. 12, 2007.

[25] N. Yosef, M. Kupiec, F. Ruppin, and R. Sharan, "A complex-centric view of protein network evolution," *Nucl. Acids Res.*, vol. 37, no. 12, 2009.

[26] S. Suthram and T. Sittler, "The plasmodium protein network diverges from those of other eukaryotes," *Nature*, 2005.

[27] C. Tan, B. Bodenmiller, and A. P. et al, "Comparative analysis reveals conserved protein phosphorylation networks implicated in multiple diseases," *Science Signaling*, vol. 2, no. 81, 2009.

[28] X.Wu, Q. Liu, and R. Jiang, "Align human interactome with phenome to identify causative genes and networks underlying disease families," *Bioinformatics*, vol. 25, no. 1, 2009.

[29] K. Goh, M. Cusick, D. Valle, and B. C. at al, "The human disease network," *This*, vol. 104, no. 21, pp. 8685–8690, 2007.

[30] K. Chao and L. Zhang, *Sequence Comparison : Theory and Methods.* Springer-verlag, 2009.

[31] P. Pezner, *Computational Molecular Biology : Algorithmic Approach.* MIT Press, 2000.

[32] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh, "Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps," *Bioinformatics*, vol. 21, pp. 302–310, 2005.

[33] L. Zager and G. Verghese, "Graph similarity scoring and matching," *Applied mathematics Letters*, vol. 21, pp. 86–94, 2008.

[34] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recognition Letters*, vol. 19, pp. 255–259, 1998.

[35] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognition Letters*, vol. 18, pp. 689–694, 1997.

[36] R. Ibragimov, M. Malek, J. Guo, and J. Baumbach, "Gedevo: An evolutionary graph edit distance algorithm for biological alignment," *German Conference on Bioinformatics(GCB13)*, pp. 68–79, 2013.

[37] B. Messmer and H. Bunke, "A new algorithm for error tolerant subgraph isomorphism," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 493–505, 1998.

[38] H. Bunke and X. Jiang, "Graph matching and similarity," *International Series in Intelligent Technologies*, vol. 15, no. 8, pp. 281–304, 2000.

[39] H. Bunke, "Graph matching: Theretical foundations, algorithms, and application," *Proceedings of Vision Interface*, pp. 82–88, 2000.

[40] N. Weskamp, E. Huellermeier, D. Kuhn, and G. Klebe, "Graph alignments : A new concept to detect conserved regions in protein active sites," *Proceedings of German Conference on Bioinformatics*, pp. 131–140, 2004.

[41] C. Clark and J. Kalita, "A comparison of algorithms for the pairwise alignment of biological networks," *Bioinformatics*, vol. 30, no. 16, pp. 2351–2359, 2014.

[42] G. Berg and M. Lassig, "Local graph alignment and motif search in biological networks," *PNAS*, vol. 41, no. 101, pp. 14 689–14 694, 2004.

[43] ——, "Cross-species analysis of biological networks by bayesian alignment," *PNAS*, vol. 29, no. 103, pp. 10 967—10 972, 2004.

[44] Q.Cheng, R. Harrison, and A. Zelikovsky, "MetNetAligner: a web service tool for metabolic network alignments," *Bioinformatics*, vol. 25, no. 15, pp. 1989–1990, 2009.

[45] R. Patro and C. Kingsford, "Global network alignment using multiscale spectral signatures," *Bioinformatics*, vol. 28, no. 23, pp. 3105—3114, 2012.

[46] B. Neyshabur, A. Khadem, S. Hashemifar, and S. Arab, "NETAL: a new graph-based method for global alignment of protein?protein interaction networks," *Bioinformatics*, vol. 29, no. 13, pp. 1654–1662, 2013.

[47] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *PNAS*, vol. 105, no. 2, pp. 12 763—12 768, 2008.

[48] M. Zaslavskiy, F. Bach, and J.-P. Vert, "Global alignment of protein.protein interaction networks by graph matching methods," *Bioinfoprmatics*, vol. 25, pp. 256–267, 2009.

[49] A. Aladağ and C. Erten, "SPINAL: scalable protein interaction network alignment," *Bioinformatics*, vol. 29, no. 7, pp. 917–924, 2013.

[50] J. Hu, B. Kehr, and K. Reinert, "NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks," *Bioinformatics*, vol. 30, no. 4, pp. 540–548, 2014.

[51] V. Saraph and T. Milenković, "MAGNA: Maximizing Accuracy in Global Network Alignment," *Bioinformatics*, vol. 30, no. 20, pp. 2931–2940, 2014.

[52] S. Hashemifar and J. Xu, "HubAlign: an accurate and efficient method for global alignment of protein-protein interaction networks," *Bioinformatics*, vol. 30, no. 17, pp. i438–i444, 2014.

[53] B. Seah, S. S. Bhowmick, and C. F. Dewey, "DualAligner: a dual alignment-based strategy to align protein interaction networks," *Bioinformatics*, vol. 30, no. 18, pp. 2619–2626, 2014.

[54] D. Park, R. Singh, M. Baym, C.-S. Liao, and B. Berger, "Isobase: a database of functionally related proteins across ppi networks," *Nucleic Acids Res.*, vol. 1, no. 39, pp. D295–D300, 2011.

[55] D. Cook and L. Holder, *Mining Graph Data.* Wiley-InterScience, 2007.

[56] J. Hua, D. Koes, and Z. Kou, "Finding motifs in protein-protein interaction networks," *Project Final Report, CMU*, vol. 39, pp. 4760–4768, 2003.

[57] T. Akutsu, "Protein structure alignment using a graph matching technique," *Proc. of Genome Informatics*, vol. 1, pp. 86–94, 1995.

[58] V. D. Blonde, A. Gajardo, M. Heymany, P. Senellar, and P. V. Dooren, "A measure of similarity between graph vertices: Applications to synonym extraction and web searching," *SIAM Review*, vol. 46, no. 4, pp. 647–666, 2004.

[59] S. Q. Le, T. B. Ho, and T. H. Phan, "A novel graph-based similarity measure for 2d chemical structures," *SIAM Review*, vol. 15, no. 2, pp. 82–91, 2004.

[60] T. Milenković and N. Pržulj, "Integrative network alignment reveals large regions of global network similarity in yeast and human," *Bioinformatics*, vol. 27, no. 10, pp. 1390–1396, 2011.

[61] L. Chindelevitch, C.-Y. Ma, C.-S. Liao, and B. Berger, "Optimizing a global alignment of protein interaction networks," *Bioinformatics*, vol. 29, no. 21, pp. 2765–2773, 2013.

[62] R. Milo, S.-Orr, S. Itzkovitz, N. Kashtan, . D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, pp. 824–829, 2002.

[63] D. Higham, M. Rasajski, and N. Pržulj, "Fitting a geometric graph to a protein–protein interaction network," *Bioinformatics*, vol. 24, no. 8, pp. 1093—1099, 2008.

[64] M. Nikolič, "Measuring similarity of graph nodes by neighbor matching," *Intelligent Data Analysis*, 2012.

[65] H. Phan and M. Sternberg, "PINALOG: a novel approach to align protein interaction networks—implications for complex detection and function prediction," *Bioinformatics*, vol. 28, no. 9, pp. 1239—1245, 2012.

[66] M. Koyutürk, Y. Kim, S. Subramaniam, W. Szpankowski, and A. Grama, "Detecting conserved interaction patterns in biological networks," *Journal of Computational Biology*, vol. 6, no. 13, pp. 1299–1322, 2012.

[67] M. El-Kebir, J. Heringa, and G. Klau, "Lagrangian relaxation applied to sparse global network alignment," *Pattern Recognition in Bioinformatics(LNCS)*, vol. 7036, pp. 225—236, 2011.