단백질 상호작용 네트워크의 허브노드 중심의 고속 시각화

Hub-Seeded Fast Visualization for Protein-Protein Interaction Networks

방선이, SunLee Bang, 최재훈, JaeHun Choi, 박종민, JongMin Park, 박수준, SooJun Park,
한국전자통신연구원

요약 단백질 상호작용 네트워크의 데이터의 양이 증가함에 따라 이를 보다 쉽게 분석하기 위해 고속으로 시각화 방법이 요구되고 있다. 본 논문은 방대한 단백질 상호작용 네트워크에서 물리적 관계도가 높은 단백질을 중심으로 다단계에 걸쳐 스프링-포스(spring force)레이아웃 기법을 적용하여 그래프를 시각화하는 방법을 제안한다. 본 논문에 따른 단백질 상호작용 네트워크에서 시각화하는 방법은 물리적 관계도가 높은 단백질을 선정하는 단계, 선정된 단백질을 기반으로 네트워크를 합병하는 단계 및 합병된 노드들을 확장하는 단계를 거쳐 시각화하는 것을 특징으로 한다. 이에 따라, 단백질 상호작용 네트워크를 균형 상태의 그래프로 표현하고 고속으로 시각화할 수 있다는 장점이 있다.

핵심어: 허브노드, Force-Directed 알고리즘, 단백질 상호작용, 네트워크 시각화

1. 서론

일반적으로 하나의 단백질은 고유한 기능을 가지고 있지만, 생체 내에서 특정한 생물학적 역할을 하기 위해서 여러 다른 단백질들과 다양한 상호작용을 한다. 현재, 대부분의 단백질간의 상호작용 데이터는 'Yeast Two-Hybrid' 와 'co-AP/MS'라는 생물학적 실험을 통해 빠르게 추출 되고 있으며, 추출된 데이터는 BIND[1], DIP[2], MINT[3], BioGRID[4] 등과 같이 데이터베이스에 체계적으로 관리되고 있다[5,6].

이러한 단백질들간의 상호작용 데이터를 단백질은 노드로 이들 사이의 상호작용을 에지로 표현하면 네트워크로 나타낼 수 있다[3]. 방대한 단백질들 사이의 복잡한 관계들로부터 특정 단백질이 아닌 전체적인 생체 메커니즘을 효과적으로 분석하기 위해 네트워크를 빠르면서 균형적인 그래프로 시각화 방법이 요구된다.

단백질 상호작용(PPI: protein-protein interaction) 네트워크를 시각화하기 위해 'Force-Directed Placement'(FDP)[7] 알고리즘이 많이 사용되고 있다[8]. 이 알고리즘은 노드와 에지의 집합에 대해 force를 지정하여 균형 상태를 이루도록 레이아웃한다. 에지들이 겹쳐져 레이아웃되는 것을 방지하기 위해, 인접노드들간의 에지는 서로 당기는 local force로 보며, 비인접노드들은 서로 밀어내는 global force로 본다. 이 알고리즘은 융통성 있고 구현하기 쉬우며 드로잉 결과도 양호하기 때문에 많이 사용되지만, 대용량의 데이터에 대해서는 느리다는 단점이 있다. 속도향상을 위해 노드집합에 대해 다단계로 클러스터를 이룬 후, 이를 확장하는 과정에서 FDP를 적용하는 'Multilevel for Force-Directed Placement' (MFDP) 알고리즘이 있다[9]. 그러나, 단계별로 시작노드를 무작위로 설정하며 각 반복 단계에서 시작노드의 인접노드 중 한 노드만을 선택해 합병을 수행하므로 허브노드와 같이 한 노드에 이웃노드들을 많이 가지고 있는 경우는 이를 처리하는데 시간이 많이 소요된다. 또한, 성장하는 네트워크에서 새로운 노드가 추가될 때 새로운 노드들은 허브에 링크를 향하는 "선호적 연결" 특징을 가지고 있으며[10], 이러한 특징은 PPI 네트워크에도 적용된다. 이에 본 논문에서는 물리적 관계도가 높은 노드를 중심으로 다단계 시각화 방법을 제안한다.

2. HubSeededMFDP

MFDP 알고리즘은 무작위로 한 노드를 선정한 다음, 연결된 다른 한 노드와의 합병 여부를 계산한 후, FDP 알고리즘으로 확장하여 시각화한다. 노드들을 한꺼번에 합병하기 위해 기존 MFDP알고리즘에 대해 피벗노드 선정단계, 합병단계, 확장단계를 다음과 같이 개선한다.

2.1 피벗노드 선정

합병과정을 수행하기 위해 기준이 되는 피벗노드는 다음과 같이 순위화된다. 네트워크상에서 다른 단백질들과 많은 관계를 가지고 있는 단백질일수록 허브노드이므로 연결된 그래프에서 인접노드가 가장 많은 노드를 피벗노드로 선정한다. 예를 들어 그림 1의 그래프에서 모든 노드들이 합병노드가 아닌 경우, 이들을 인접노드 순으로 정렬하면 List1과 같이 정렬된다.

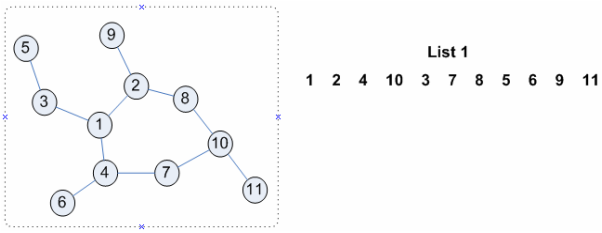

그림 1 피벗노드 선정의 예

여기서, 1, 2, 4, 10번 노드들은 모두 인접노드의 수가 3으로 동일하다. 만약, 그림 1이 합병과정을 몇 번 수행한 후, 1, 2, 4, 10번 노드들이 합병노드이고 각 합병노드의 합병된 노드의 수(nested degree)가 각각 5, 4, 3, 2로 다르다고 가정하자. 단계별 합병과정에서 한 노드를 기준으로 계속 합병되는 것을 방지하기 위해, 인접노드수가 같다면, nested degree와의 합이 가장 작은 노드에 우선순위를 두어 정렬한다. 이에 따라, List1은 10, 4, 2, 1, 3, 7, 8, 5, 6, 9, 11 순으로 정렬된다. 정렬된 리스트에 대해 방문하지 않은 노드들 중 순위가 제일 높은 노드를 피벗노드로 선정한다. 피벗노드 선정을 위한 노드의 정렬된 리스트를 구하는 때는, 합병과정에서 정렬된 리스트를 모두 방문한 경우를 합병과정의 한 단계로 보고, 이 단계가 끝날 때마다 합병그래프에 대한 피벗노드 선정을 위한 노드에 대한 정렬된 리스트를 구한다.

2.2 합병과정(Coarsening Step)

각 합병 단계별로 그래프의 피벗노드를 중심으로 연결된 모든 노드들을 리스트에 저장하여 합병된 새로운 노드로 대치하여 최종그래프의 크기가 특정수일 때까지 합병과정을 반복하여 최종 합병 그래프를 만든다.

합병과정 시, 특정노드위주로 합병되는 경우, 확장과정 단계를 많이 거치게 되므로 이를 방지하기 위해 cutValue를 설정한다. nested degree가 높은 노드들끼리 합병되지 않기 위한 cutValue 선정은 다음과 같다. 노드 리스트의 각 노드의 nested degree에 대해 내림차순 정렬을 한 후, “상위 20%에 해당하는 nested degree” 이면서 “nested degree의 평균보다 큰 값” 에 해당하는 최소 nested degree를 cutValue로 설정한다.

예를 들어 그림 2의 초기 그래프에 대해 1번 노드가 피벗노드이므로 인접노드인 2, 3, 4번 노드를 합병하여 합병노드 c1을, 다음 피벗노드 10번 노드의 인접노드인 7, 8, 11번 노드를 합병하여 합병노드 c2로 대치한다. 다음 피벗노드 5번 노드에 대해 인접노드인 c1의 nested degree가 cutValue와 동일하므로 합병을 하지 않는다. 6, 9번 노드도 인접노드가 c1이므로 합병을 하지 않고 합병 1단계를 마친다. 다음 합병단계에서 피벗노드 c1노드의 인접노드 5, 6, 9, c2노드에 대해 nested degree가 cutValue와 동일한 c2노드를 제외한 노드들을 합병하여 c3노드로 대치한다. 다음 피벗노드인 c2노드의 인접노드 c3이 nested degree는 cutValue인 7이므로 합병 2단계를 마친다. 합병된 그래프의 사이즈가 2이므로 모든 합병과정을 마친다.

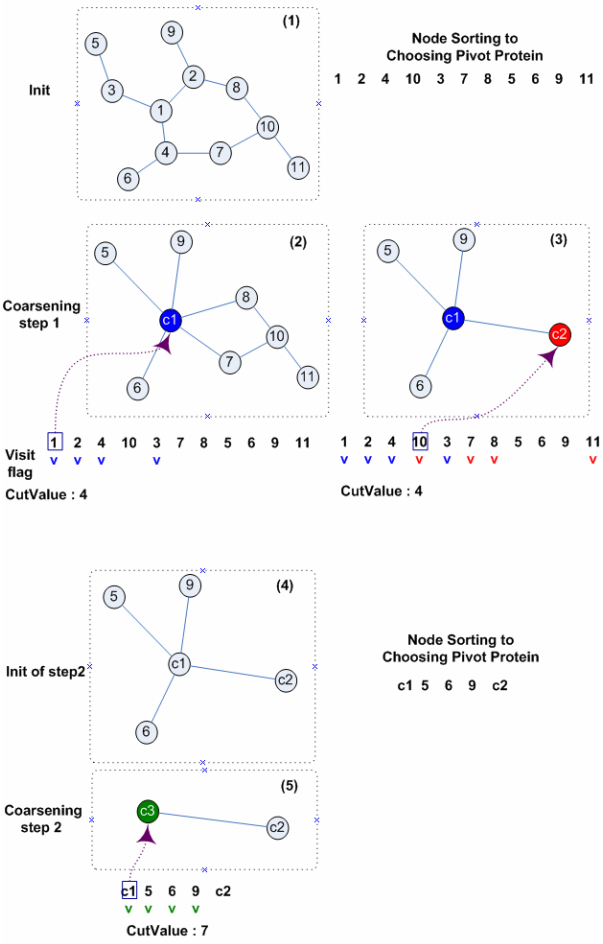

그림 2 합병과정 예

2.3 확장과정(Expansion Step)

확장과정에서 FDP과정을 조금 수행할수록 레이아웃 속도가 향상되므로 최대한 FDP의 최종 상태에 가깝도록 nested node들을 위치시키도록 한다. 이를 위해 먼저, 피벗노드를 중심으로 합병된 노드들을 자연거리상수

(natural spring length)를 지름으로 하는 원주 위에 균일하게 위치하도록 확장한다. 균일하게 위치 선정은 각 합병노드들의 nested degree로 원주를 분할하여 분할 포인트(partition point)를 설정한다. 균형상태의 그래프는 노드들간의 관계 정보를 이용하므로 이전 확장단계에서 위치가 확정된 노드와 관계가 있는 노드는 대표위치(main position)를 구한 후, 대표위치와 가까운 분할 포인트에 해당 노드를 위치시킨다. 합병그래프의 모든 합병노드들이 각기 한번씩 확장을 거친 경우를 확장과정의 한 단계로 보고, 모든 합병노드가 확장될 때까지 이 단계를 반복한다.

그림 2에 대한 최종 합병 그래프인 그림3-(1)의 1단계 확장과정은 그림 3과 같다.

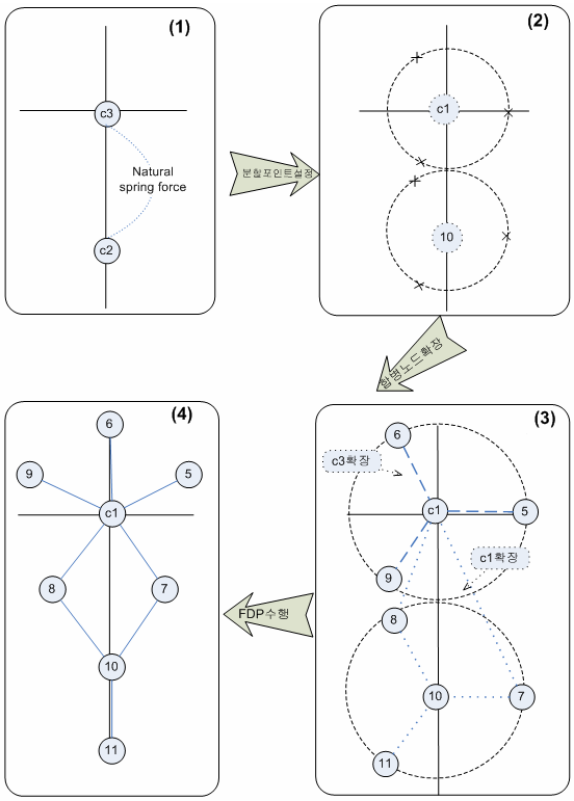

그림 3 1단계 확장과정

합병노드 c3과 c2의 nested node는 각각 {c1, 5, 6, 9}와 {10, 7, 8, 11}이다. 각 합병노드의 피벗노드인 c1과 10을 제외한 각각 나머지 3개의 노드에 대한 분할 포인트를 그림 3-(2)와 같이 설정한다. 확장 1단계이므로 이전 확장단계에서 위치가 확정된 노드가 없으므로 그림 3-(3)과 같이 설정된 분할 포인트에 차례로 nested node를 위치시킨다.

1단계 확장의 최종 그래프 그림 3-(4)에 대한 합병노드 c1의 2단계 확장은 그림 4와 같다.

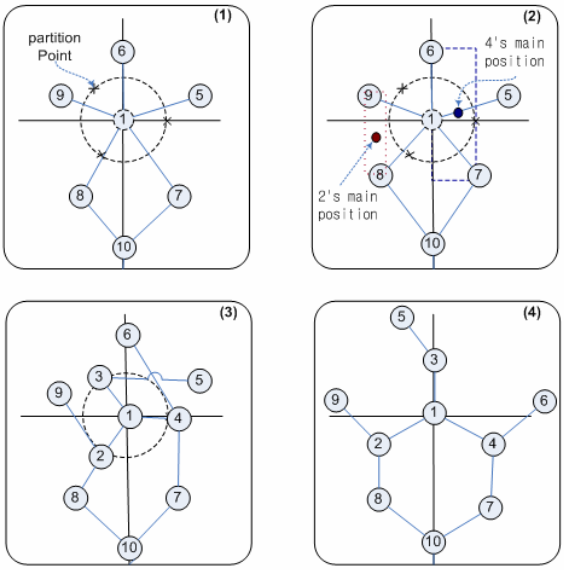

그림 4 2단계 확장과정

합병노드 c1의 nested node는 {1, 2, 3, 4}이므로 피벗노드 1번 노드를 기준으로 분할 포인트를 그림 4-(1)과 같이 설정한다. 분할 포인트의 선택은 이미 이전 확장단계에서 위치를 확정한 노드들에 대해 이웃노드가 많은 노드부터 시작한다. 2번 노드는 8, 9번 노드와 4번 노드는 6, 7번 노드와 3번 노드는 5번 노드와 이웃이므로 2, 4번 노드부터 분할 포인트를 선택한다. 분할 포인트의 선택은 이웃노드와의 관계를 고려한 대표위치를 계산한 후, 이 대표위치와 가까운 분할 포인트에 해당 노드를 위치시킨다. 대표위치는 이웃노드들의 좌표로부터 최소, 최대의 x, y좌표를 구한 후, x, y좌표의 평균값으로 정한다. 예를 들어, 2번 노드의 대표위치의 최소 x좌표는 9번 노드의 x좌표, 최대 x좌표는 8번 노드의 x좌표, 최소 y좌표는 8번 노드의 y좌표, 최대 y좌표는 9번 노드의 y좌표 이므로 이들의 평균을 대표위치의 x, y좌표로 정한다. 이에 따라, 2번 노드와 4번 노드의 대표위치는 그림 4-(2)와 같다. 대표 위치가 선정되면, 그림 4-(3)와 같이 각 노드의 대표위치와 가까운 분할 포인트에 해당 노드를 위치시키고, 나머지 노드인 3번 노드는 빈 분할 포인트에 위치시킨 후 FDP를 수행하면 그림 4-(4)와 같이 최종적으로 그래프를 레이아웃한다.

3. 실험 결과

3.1 구현 환경

HubSeededMFDP 알고리즘은 한국전자통신연구원의 바이오인포매틱스팀에서 개발한 바이오 상호관계 네트워크 시스템인 bioINET의 프레임워크 상에 구현하였다. bioINET의 PPI네트워크 레이아웃의 특징은 네트워크의 Connected Graph별로 (0,0)좌표를 기준으로 해당 레이아

웃 알고리즘으로 시각화를 수행 후, 노드수가 많은 그래프부터 우선적으로 배치한다. 각 그래프들이 배치되는 범위는 이전에 배치된 그래프의 최대높이와 최대너비 범위를 기준으로 좌표를 상대적으로 조정하여 배치하도록 한다. FDP 수행 시 파라미터로는 natural length는 50, global force 계산시 사용되는 상수 C는 0.3, FDP수행의 중단을 결정하는데 사용되는 상수 tol은 0.1을 설정하였으며, FDP의 최대 수행 회수에 대한 파라미터를 추가하여 100으로 설정하였다.

bioINET시스템은 자바로 구현되었으며, 데이터베이스는 오라클 XE를 사용하였다. bioINET의 전체적인 인터페이스는 그림 5와 같다.

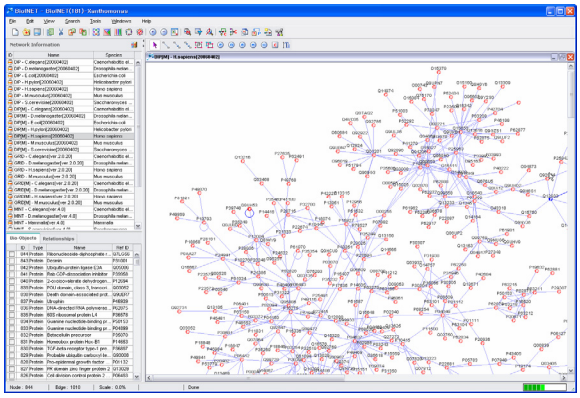

그림 5 bioINET 인터페이스

실험은 CPU가 3.4GHz Pentium4, 메모리는 4GB이며 OS는 Windows XP인 시스템 상에서 수행되었다.

3.2. 성능 평가

PPI네트워크의 공개 DB인 DIP(Database Interacting Proteins)[11]으로부터 데이터를 수집하여 MFDP 알고리즘과 HubSeededMFDP 알고리즘을 비교한 결과는 표 1과 같다.

표 1 MFDP와 Hub-Seed MFDP의 시각화 속도 비교

종	크기	시각화 시간(ms)		(%)
	Node/Edge	MFDP	Hub-Seeded MFDP	향상률
Yeast	4534/16383	363750	294687	19
C. elegans	2353/3334	118766	44172	63
E. Coli	1833/6948	48156	30109	37
D. melanogaster	887/1116	19516	8469	57
Homo Sapiens	846/1012	3875	2328	40

전반적으로 노드의 수가 많은 네트워크일수록 시각화 속도가 향상되었으며, 특히 C. elegans 의 경우 최대 63%까지 향상되었다. 특별히 살펴볼 부분은 Yeast와 E. Coli의 경우, 노드 수에 비해 향상률이 낮은 점이다.

먼저, 이를 위해 표 1의 종별 PPI네트워크의 크기에 대해 노드당 이웃노드의 수를 살펴보면 표 2와 같다.

표 2 종별 PPI네트워크의 노드당 이웃노드의 수 비교

종	노드당 이웃노드의 수
Yeast	3.6133
C.elegans	1.4169
E. Coli	3.79
D. melanogaster	1.2518
Homo Sapiens	1.196

Yeast와 E. Coli의 경우 다른 종에 비해 노드당 이웃노드의 수가 3배 정도 많음을 알 수 있다. 이는 허브노드를 중심으로 다른 이웃노드들간에 서로 복잡한 관계가 많이 이루어졌음을 말한다. 즉, 허브노드만을 중심으로 밀집되어 있을수록 시각화 속도의 향상이 큼을 보인다.

또한, D. melanogaster 와 Homo Sapiens의 경우, PPI 네트워크의 크기는 비슷하지만 속도 향상률에서는 차이가 크게 나는 경우를 살펴보자. 이를 위해, 합병노드를 생성하는 회수 및 FDP 수행 회수를 비교하면 표 3과 같다.

표 3 합병노드 생성 회수 및 FDP 수행회수 비교

Species	합병노드생성 회수 / FDP 수행회수	
	MFDP	Hub-Seeded MFDP
Yeast	4434/52	488/6
C.elegans	2038/ 106	262/29
E. Coli	1443/42	152/12

D. melanogaster	857/42	116 / 4
Homo Sapiens	601/123	133 / 54

이와 같이, HubSeededMFDP의 합병, 확장 과정이 줄어듦에 따라 시각화하는 시간이 줄어든다. 특히, D. melanogaster의 경우, HubSeededMFDP의 합병노드 생성 회수 116회로 MFDP의 합병노드 생성 회수 857회에 비해 많이 줄었음을 알 수 있다. 또한, D. melanogaster의 경우 MFDP에서는 합병을 수행한 회수가 857로 많이 이루어진 반면, Homo Sapiens에서는 601회로 D. melanogaster에 비해 적음을 보이고 있다. 이는 D. melanogaster 가 허브노드를 많이 포함하고 있어 MFDP를 수행할 경우 허브노드를 중심으로 여러 번의 합병노드 생성 과정이 이루어지고 있음을 뜻하고 있으며, 그림 6과 그림 7에서도 그 차이를 알 수 있다.

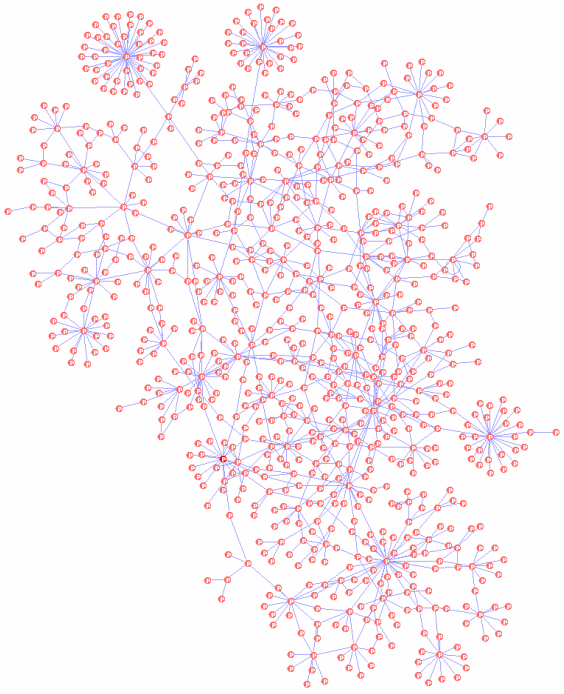

그림 6 D. melanogaster 시각화 예

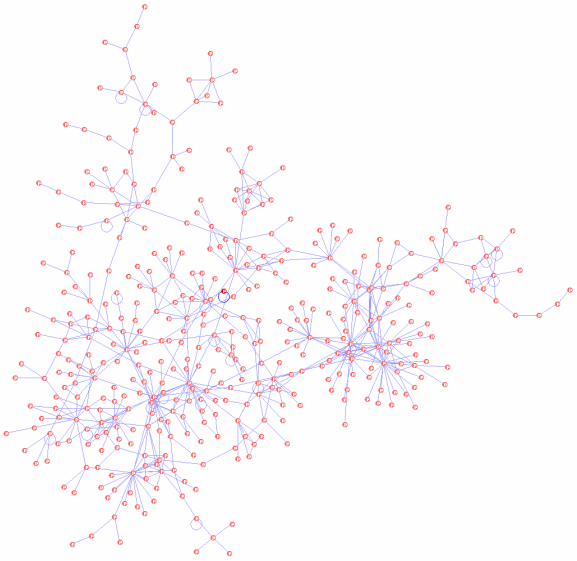

그림 7 Homo Sapiens 시각화 예

따라서, 표 2 및 표 3에 의한 결과 해석을 통해, 물리적 관계도가 높은 노드 정보를 이용하여 시각화하는 HubSeededMFDP 알고리즘이 방대한 단백질 상호작용 네트워크를 고속으로 시각화할 수 있음을 보였다.

4. 결론

단백질 상호작용 네트워크의 데이터 양이 증가함에 따라 분석의 용이성을 위해 이를 고속으로 시각화 하는 작업이 요구되고 있다.

본 논문에서는 물리적 관계도가 높은 노드 중심으로 합병 및 확장의 다단계 시각화를 수행하는 허브노드 중심의 시각화 방법이 기존 그래프 레이아웃 알고리즘에 비해 시각화 속도를 최대 63%까지 향상시켰음을 살펴보았다. 특히, 네트워크가 클수록 또한, 허브노드로 밀집된 노드의 수가 많은 네트워크일수록 더욱 효과적임을 보였다.

향후 과제로는 시각화 속도를 더욱 향상하기 위해 cutValue 선정 방법 및 적합한 파라미터 설정 방법이 요구되며, 신호전달 네트워크의 시각화로 확장하고자 한다.

참고문헌

[1] G. D. Bader, D. Betel and C. W. V. Hogue, “BIND: the Biomolecular Interaction Network Database”, Nucleic Acids Research, Vol. 31, No. 1, pp. 248-250, 2003.

[2] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, “The Database of

Interacting Proteins: 2004 update", Nucleic Acids Research, Vol. 32, Database Issue D449 - D451, 2004.

[3] A. Chatr-aryamontri, A. Ceol, L. M. Palazzi, G. Nardelli, M. V. Schneider, L. Castagnoli, G. Cesareni, "MINT: the Molecular INTeraction database", Nucleic Acids Research, Vol. 35, Database issue D572-D574, 2007.

[4] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: a general repository for interaction datasets", Nucleic Acids Research, Vol. 34, Database Issue D535 - D539, 2006.

[5] P. Uetz and R. L. Jr. Finley, "From protein networks to biological systems" FEBS Letters, Vol. 579, No. 8, pp. 1821~1827, 2005.

[6] C. Mering, R. Krause, M. Cornell, S. Oliver, S. Fields and P. Bork, "Comparative assessment of large-scale data sets of protein-protein inetractions" Nature, Vol. 417, No. 6887, pp. 399~403, 2002.

[7] T. M. J. Fruchterman and E. M. Reingold, "Graph Drawing by Force-Directed Placement", Software-Practice & Experience, Vol. 21, No. 11, pp. 1129~1164, 1991.

[8] P. Uetz, T. Ideker and B. Schwikowski, "Visualization and Integration of Protein-Protein Interactions" E. Golemis, (ed.) Protein-Protein Interactions - A Molecular Cloning Manual. Cold Spring Harbor Laboratory Press, pp. 623-646.

[9] C. Walshaw, "A Multilevel Algorithm for Force-Directed Network-Drawing" Journal of Network Algorithms and Applications, Vol. 7, No. 3, pp. 253~285, 2003.

[10] A. L. 바라바시, "링크-21 세기를 지배하는 네트워크 과학", 동아시아, 한국, 2002.

[11] http://dip.doe-mbi.ucla.edu/