doi:10.1016/j.jmb.2004.04.047

J. Mol. Biol. (2004) **340**, 179–190

JMB

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

LGL: Creating a Map of Protein Function with an Algorithm for Visualizing Very Large Biological Networks

Alex T. Adai[1], Shailesh V. Date[1], Shannon Wieland[1] and Edward M. Marcotte[1,2]*

[1]*Center for Systems and Synthetic Biology, and Institute for Cellular and Molecular Biology, 1 University Avenue University of Texas, Austin TX 78712-1095, USA*

[2]*Department of Chemistry and Biochemistry, Institute for Cellular and Molecular Biology 1 University Avenue University of Texas, Austin TX 78712-1095, USA*

**Corresponding author*

Networks are proving to be central to the study of gene function, protein–protein interaction, and biochemical pathway data. Visualization of networks is important for their study, but visualization tools are often inadequate for working with very large biological networks. Here, we present an algorithm, called large graph layout (LGL), which can be used to dynamically visualize large networks on the order of hundreds of thousands of vertices and millions of edges. LGL applies a force-directed iterative layout guided by a minimal spanning tree of the network in order to generate coordinates for the vertices in two or three dimensions, which are subsequently visualized and interactively navigated with companion programs. We demonstrate the use of LGL in visualizing an extensive protein map summarizing the results of ~21 billion sequence comparisons between 145,579 proteins from 50 genomes. Proteins are positioned in the map according to sequence homology and gene fusions, with the map ultimately serving as a theoretical framework that integrates inferences about gene function derived from sequence homology, remote homology, gene fusions, and higher-order fusions. We confirm that protein neighbors in the resulting map are functionally related, and that distinct map regions correspond to distinct cellular systems, enabling a computational strategy for discovering proteins' functions on the basis of the proteins' map positions. Using the map produced by LGL, we infer general functions for 23 uncharacterized protein families. LGL is freely available (at http://bioinformatics.icmb.utexas.edu/lgl).

Keywords: network; visualization; protein function; protein map; bioinformatics

Introduction

About half of the roughly 40,000 genes encoded by the human genome,[1,2] as in most sequenced genomes, are completely uncharacterized and of unknown function. There is a broad need for methods to discover the functions of these thousands of uncharacterized genes and how they interact with each other. An important method of function discovery is the study of genes and their products as components of networks, rather than studying genes in isolation or in linear pathways. Networks are already being used to model gene function, protein–protein interaction,[3,4] and biochemical pathway[5–7] data, often on a genome-wide scale. This has brought forth the need for

Supplementary data associated with this article can be found at doi: 10.1016/j.jmb.2004.04.047

Present addresses: A. Adai, 600 16th St., Genentech Hall, University of California, San Francisco, San Francisco, CA 94143-2280, USA; S. Date, 1423 Blockley Hall, 423 Guardian Drive, University of Pennsylvania, Philadelphia, PA 19104, USA; S. Wieland, 77 Massachusetts Avenue, Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA.

Abbreviations used: LGL, large graph layout; MST, minimum spanning tree; COGS, clusters of orthologous groups; VRML, virtual reality modeling language.

E-mail address of the corresponding author: marcotte@icmb.utexas.edu tools for visualizing and exploring biological networks. Network-drawing programs are quite prevalent, such as AT&T GraphViz among others,[8] and more recently, network visualization tools have been developed specifically for the biological community.[9–11] Such algorithms are typically intended for small networks. However, algorithms are required to dynamically, clearly, and interactively visualize large networks on the order of millions of edges and hundreds of thousands of vertices such as are generated by large-scale protein and DNA sequence comparisons. Here, we first introduce a method for dynamically visualizing and exploring such large networks in two (2D) or three dimensions (3D), implemented as a open source suite of programs named large graph layout (LGL), then apply LGL to visualize a complete global protein homology network.

When confronted with the thousands of genes from genome sequencing projects, one might first compare the gene sequences with each other to identify gene families. The results of such a direct sequence comparison are informative,[12,13] but additional information emerges by examining results from many such sequence comparisons.[14–17] This notion is formalized in a protein homology network, which is a network whose vertices represent proteins and whose edges represent significant amino acid sequence similarity relationships between pairs of the proteins.[9,18,19] Such a network captures much of the history of gene evolution, including evidence of gene duplications,[20,21] deletions,[21] fusions,[22–24] and fissions,[24] that is preserved in the sequences of current day genes.

We expect such networks to be highly structured. Intuitively, proteins in a sequence family should be more similar to each other than to unrelated proteins. Likewise, a fusion protein will often exhibit similarity to each of its components' respective protein families, even though the families show no similarity to each other. In such cases, the fusion may represent the rare merger of disparate sequences, or the more common tendency for domains to rearrange or swap ("domain promiscuity"[22]). Proteins grouped by their similarities should therefore cluster into sequence families linked occasionally by fusion proteins. These fusion proteins, termed Rosetta Stone proteins,[22] tend primarily to link proteins of related function,[22,23,25,26] so by a simple extension of this Rosetta Stone principle, we might expect that organizing proteins by their sequence similarities would simultaneously organize them according to cellular pathways and functions. Therefore, a map in which proteins are spatially positioned according to their sequence similarities should directly reveal protein function.

As projective methods,[27] clustering algorithms,[18,19,28] and distance-preserving algorithms[29,30] do not preserve the organization induced by fusion proteins, we have instead opted for a network visualization approach to create the map. Portions of such homology networks have been visualized,[9] but the homology networks' large scale makes them difficult to visualize in their entirety. The LGL algorithm was developed to make visualization of such large biological networks tractable, as well as aesthetically pleasing and informative for networks with complex internal structures. We apply LGL to create a map for the complete set of 145,579 proteins from 50 genomes and demonstrate that the map effectively captures protein function information, and that genes' functions can be directly discovered from a map in which genes are organized according to historic genetic events of gene duplications, deletions, fusions, and fissions.

Results

The LGL network layout algorithm

LGL is based on a mass-spring algorithm where edges play the role of springs pulling together vertices, treated as masses, into highly connected clusters. For any given set of data, LGL works in two distinct phases. The first phase of LGL generates 2D (or 3D) coordinates of each vertex in space in a process known as the network layout. The layout consists of three stages: (1) separation of the original network into connected sets (sets of vertices that are reachable by each other by traversing the edges connecting them); (2) generation of spatial coordinates for each vertex in each connected set (laying out each connected set independently); and (3) integration of the connected sets into one coordinate system.

A side effect of using springs to simply pull together masses is that masses can stack on top of each other unless a repulsive force, such as another spring term, forces the vertices apart. One may imagine the repulsive term as radially directed springs attached to vertices to push away any proximal vertices (see Methods). To calculate the repulsive term efficiently, the vertices are placed in a grid where the voxels are only slightly larger than the repulsive spring lengths. A spatially localized repulsion term such as a spring only requires a local inspection of each vertex, which is equivalent to inspecting only neighboring voxels. The computational complexity depends on the grid size and the density of the vertices in the grid, but ranges from an upper limit of $O(n^2)$ down to $O(n)$, representing considerable computational savings over the exhaustive inspection of all vertices for proximity.

Attractive and repulsive forces are calculated for each vertex as described in Methods, and finally summed for all vertices. The positions of the vertices are then updated using the simple relationship: $\vec{x}_{\text{new}} = \vec{x}_{\text{old}} + \vec{F}_{\text{total}}\,\mathrm{d}t$. This relationship is not physical, as $\vec{F}_{\text{total}}$ now has the appearance of velocity and not force when compared to Newton's equations of motion. However, using this relationship precludes any need for an energy dissipation term, and allows the algorithm to avoid tracking vertex velocities. Occasionally, two vertices superimpose, in which case they each receive a small constant force term in a random direction.

Empirical tests of layouts of very large networks based only on summing forces from random initial conditions, even with pre-clustering by connectivity, cannot typically reveal global structure for complex networks, due to the high number of overlaid edges. As biological networks are often quite dense, and layouts based only on the spring algorithm tend to be difficult to interpret, we employ a strategy shown to effectively separate dense networks.[31] During the process of generating spatial coordinates for each vertex, we use a guide tree to determine the order in which vertices are included in the spring force layout calculations. Vertices from a single connected network are laid out iteratively starting with a root vertex and incorporating additional vertices as guided by a minimum spanning tree (MST) of the network. The MST is defined as the minimum set of edges necessary to keep the network connected, where each edge is weighted by its associated BLAST *E*-value, and the sum of all the weights of the edges in the tree are minimized.

Briefly, the MST is generated by ranking the edges by ascending weights and marking each edge if it does not create a cycle with any previously marked edge. When this process is completed, the marked edges form the minimally spanning tree of the network, with a total of $N - 1$ edges connecting N vertices. The MST determines the order of placement of the vertices, giving preference to vertices closer to the root vertex. The root vertex, which can be chosen arbitrarily or based on its centrality in the network, is assigned to level 0. All other vertices are then assigned a level according to their edge-based distance in the MST from the root vertex, setting the order in which they will be incorporated into the layout. Using this guide tree-based layout strategy allows the network layout to come to equilibrium in a manner which preserves the structure of central network components and which reduces cluttering from adjacent vertices.

The layout begins with the root vertex (the level 0 vertex), placed in the center of the grid and flanked radially by all child vertices (level one vertices), as drawn in Figure 1. A sphere (in the general case, for 2D layouts, a circle is used) is generated with the root vertex at the center, and level one vertices are placed in random locations on the surface of the sphere. Network edges are introduced, where appropriate, between vertices present at this iteration (level) of the layout. Note that all network edges are ultimately considered for the layout, not just those in the MST, which serves only to set the order in which vertices are incorporated. The system progresses through time, calculating attractive and repulsive terms until positions of the vertices change negligibly with respect to the positions in the preceding time step, whereupon the next level of vertices specified by the MST are added, and the system is minimized again. This process repeats until the layout is complete, all vertices and edges have been added, and vertices have moved to equilibrium positions.

Following the layout of all connected sets, the individual layouts from disconnected subnetworks are integrated into a single coordinate system *via* a funnel process: the connected sets are sorted in descending size by the number of vertices. The first connected set is placed at the bottom of a potential funnel and other sets are placed one at a time on the rim of the potential funnel and allowed to fall towards the bottom where they are frozen in space upon collision with the previous sets. This process is applied sequentially to each connected set. Since the disconnected networks share no relationship between each other (by definition, they share no edges), the integration step is arbitrary, serving only to provide space between the connected sets.

Applying LGL to visualize a global protein homology map

We have used LGL to visualize the results of approximately 21 billion amino acid sequence comparisons, made using the program BLAST, of 145,579 amino acid sequences from 50 completely sequenced genomes, including ten archaeal, 37 bacterial, and three eukaryotic genomes. The

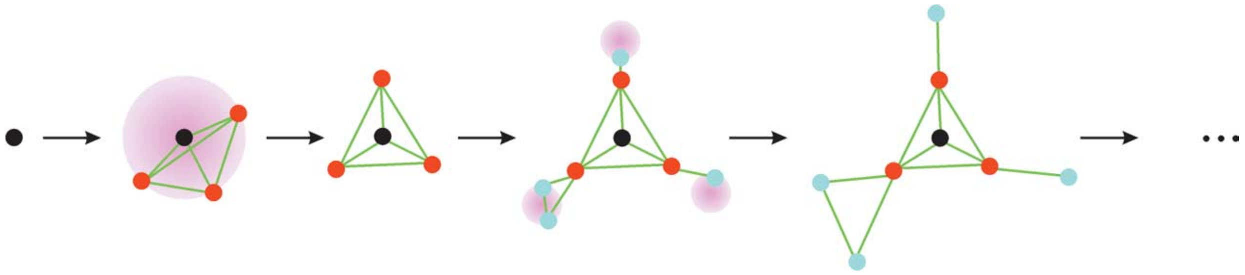

Figure 1. Successive iterations of the layout. Level one vertices (red circles) are placed randomly on a sphere around the root node (black circle). The system is allowed to iterate through time satisfying attractive and repulsive forces until at rest. Level two nodes (blue circles) are placed randomly on spheres directed away from the current layout. Again, the system is allowed to evolve through time till at rest. This process is iterated for the entire graph.

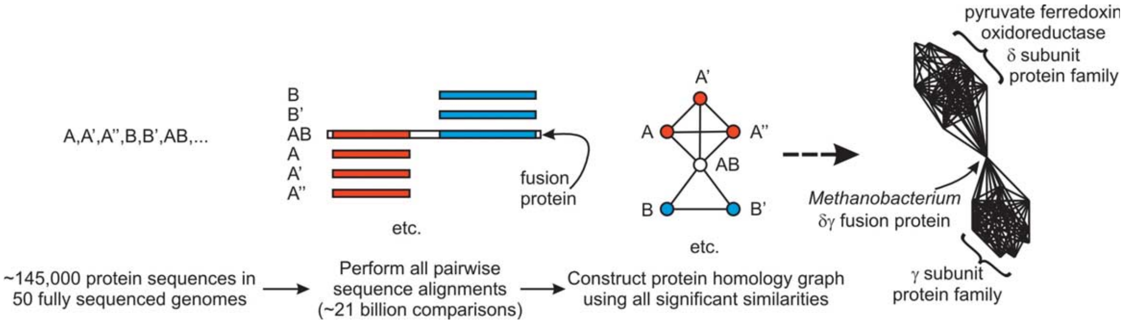
A,A',A",B,B',AB,...
B
B'
AB
A
A'
A"
fusion protein
etc.
A'
A
A"
AB
B
B'
etc.
pyruvate ferredoxin oxidoreductase δ subunit protein family
Methanobacterium δγ fusion protein
γ subunit protein family
~145,000 protein sequences in 50 fully sequenced genomes
Perform all pairwise sequence alignments (~21 billion comparisons)
Construct protein homology graph using all significant similarities

Figure 2. A protein homology map summarizes the results of billions of sequence comparisons by modeling the proteins as vertices in a network, and the statistically significant sequence similarities as edges connecting the relevant proteins. In this manner, proteins within a sequence family (such as A, A′, A″, and AB; or B, B′ and AB) are all or mostly connected to each other, forming a cluster in the map. Fusion proteins (such as AB) serve to connect their component proteins' families. The structure of the resulting map reflects historic genetic events, such as gene fusions, fissions, and duplications, which are responsible for producing the modern-day genes. The map simultaneously represents homology relationships (edges), remote homologies (proteins not directly connected but in the same cluster), and non-homologous functional relationships (adjacent clusters and clusters linked by fusion proteins).

results were interpreted as a large biological network, where each protein was represented as a vertex (as described in Figure 2), and each significant BLAST similarity was represented as an edge connecting the corresponding proteins. The resulting network, seen in Figure 3, has a complex structure, with extensive clustering and interconnections that derive from the diverse evolutionary histories of the proteins. As expected, families of similar proteins form clusters in the map that emerge from the high interconnectivity of the proteins (i.e. not by the results of a clustering algorithm, but by the placement of highly interconnected proteins close to each other in the map). This general approach is known to effectively identify both close and distant sequence homologs.[21,23,32−34] The extensive occurrence of fusion proteins, along with gene duplications, serves to organize the proteins in the protein homology map.

About a third to half of the proteins in the database are linked together by such chains of fusions,

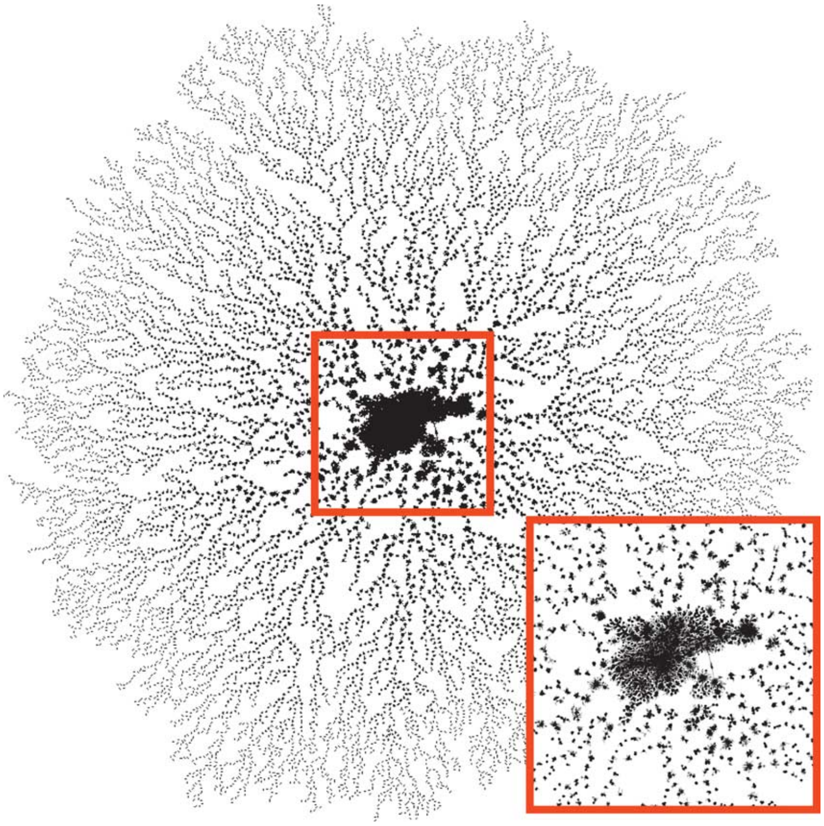

Figure 3. The complete protein homology map. A layout of the entire protein homology map; a total of 11,516 connected sets containing 111,604 proteins (vertices) with 1,912,684 edges. The largest connected set is shown more clearly in the inset and is enlarged further in Figure 4.

depending upon the BLAST threshold used to calculate the protein homology map. For example, at the BLAST threshold of $E < 10^{-6}$, 62,828 proteins are linked together into a single connected subnetwork, indicating that at least 43% of the proteins in the combined proteome are connected by the transitive occurrence of gene fusions. In Figure 3, 30,727 proteins are linked into a single connected set using the BLAST threshold of $E < 10^{-12}$. The remaining proteins are found in one of the other 11,516 connected sets: the next largest connected set at this threshold contains only 973 proteins, and 33,975 of the remaining proteins have no links in the map, corresponding to ORFans[35] without detectable sequence homologs at this BLAST threshold.

Within a connected subnetwork, such as the connected subnetwork of 30,727 proteins enlarged in Figure 4, we expect network organization to be dictated by homology relationships and fusion proteins, resulting in an organization of proteins into sequence families that are themselves organized by function, since protein families linked by such fusion relationships are generally functionally related. We have devised several tests, based upon both extensive manual inspection of the map and quantitative measurement, which confirm that this is indeed the case. First, visual examination of the map revealed many linked, functionally related clusters; several such clusters are expanded and labeled in Figure 5. Adjacent clusters can be seen to have similar function, and proteins' functions change only gradually across the map. Figure 5A shows an example of Rosetta Stone linked proteins such as those diagrammed in Figure 2, in which one fusion protein links the two separate protein families. Figure 5B shows extended Rosetta stone links between functionally related protein clusters. The linked proteins are known to associate with one another to form active pyruvate synthase and α-ketoglutarate:ferredoxin oxidoreductase complexes. Figure 5C shows a more extensive spatial persistence of function: the proteins on the left side of the Figure are involved in acetyl CoA metabolism; towards the right of the Figure, the function is still metabolic but shifts towards related metabolisms, such as that of amino acid residues and carbamoyl phosphate. Examination of other such examples supports the notion that broad regions of the map correspond to general protein functional categories, as labeled in Figure 4.

Second, we compared the functional similarity of pairs of proteins to their spatial separation in the map. For example, examining the subset of *Escherichia coli* proteins in the largest connected subnetwork for which biochemical function is known (*via* assignments in the KEGG pathway database[36]) reveals that, on an average, proteins linked by Rosetta Stone proteins are in the same pathway 75% of the time, compared to 4.7% for random pairs of the same *E. coli* proteins. Thus, we expect functional similarity should persist across space in the map. Employing a set of general protein function annotation, the clusters of orthologous groups (COGS)[17,37] annotations, Figure 6 shows that for pairs of proteins sharing no direct connection in the map, the tendency of the proteins to be in a related pathway extends beyond a typical cluster size. Although these values must be calibrated for each layout, for the map in Figure 4, the correlation length extends well beyond the typical cluster size (defined as the unit distance and corresponding to the spring equilibrium

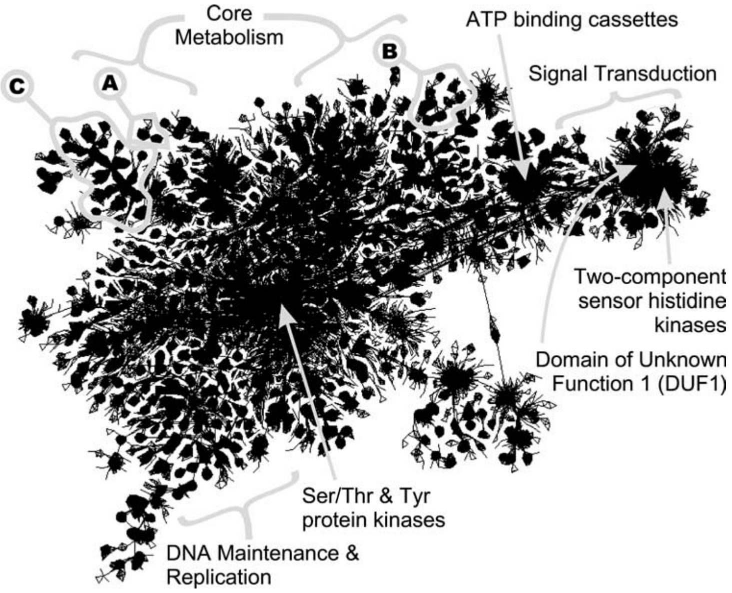
Core Metabolism
ATP binding cassettes
Signal Transduction
C
A
B
Two-component sensor histidine kinases
Domain of Unknown Function 1 (DUF1)
Ser/Thr & Tyr protein kinases
DNA Maintenance & Replication

Figure 4. A map of gene function emerges from ~21 billion gene sequence comparisons. Proteins are drawn as points, with lines connecting proteins with similar sequences, and are arranged so that homologous proteins are adjacent in the Figure. The size of each cluster is proportional to the number of proteins in that sequence family. Fusion proteins force their component proteins' respective families to be close together in the Figure, and thereby serve to organize the proteins in the map according to their functions. The resulting broad trends of protein function are labeled, as are several of the most extensive sequence families. A–C indicate specific regions that are magnified in Figure 5. The general function of uncharacterized proteins can be found from their position in the map; for example, the "domain of unknown function 1" family[38–40] is associated with proteins involved in signal transduction. For clarity, only the greatest connected network component is drawn, containing 30,727 proteins (vertices) and 1,206,654 significant sequence similarities (edges), and representing ~4 billion sequence comparisons.

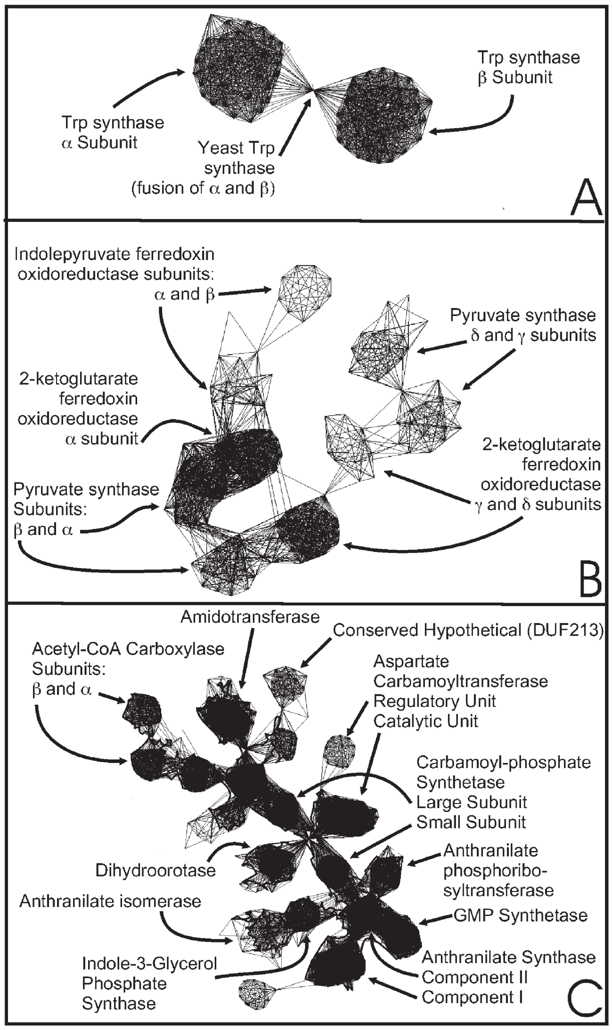

Figure 5. Functionally related gene families form adjacent clusters in the map. Three examples illustrate spatial localization of protein function in the map, specifically A, the linkage of the tryptophan synthase α family to the functionally coupled but non-homologous β family by the yeast tryptophan synthase $\alpha\beta$ fusion protein, B, protein subunits of the pyruvate synthase and alpha-ketoglutarate ferredexin oxidoreductase complexes; and C, metabolic enzymes, particularly those of acetyl CoA and amino acid metabolism.

length, thus a typical grouping of proteins in the map will have a diameter roughly equal to the spring equilibrium length). The functional similarity f decays exponentially with increasing distance d across the map, with $f = f_0\,e^{-kd}$ and $k = 0.26$, $f_0 = 0.68$. The measured decay curve implies that neighboring clusters are $\sim$52% likely to operate in the same broad category of cellular system, consistent with the high degree of functional similarity of proteins linked by gene fusions observed with the KEGG annotation.

The organization of proteins by function forces broad cellular processes to localize in the map. To investigate the conservation of protein function across the network, proteins in the network were labeled in Figure 7 according to the four primary classes of function assigned to the proteins in the COGS database:[17,37] information storage and processing, general cellular processes, metabolism, poorly characterized or unknown functions. Previous analyses have shown that clusters in protein homology networks represent protein families,[18,19] and directly linked proteins (homologs) in our map share one of the four major COGS functional categories 88% of the time. Visual inspection of our map reveals that members of such clusters are typically associated with the same COGS function, as expected. Adjacent clusters tend strongly to be of the same function, as illustrated by chains of functionally related protein families in Figure 7 and shown quantitatively in Figure 6. Empirically, we observe the smaller connected sets (Figure 3) to almost exclusively represent single protein families of similar function.

Components of information storage and metabolism are strongly localized: proteins involved in transcription, translation and DNA replication preferentially occur near the bottom left, signaling systems towards the center and right, and metabolism towards the top. Proteins with unknown functions show the least positional bias and are relatively evenly distributed across the map.

Because proteins are organized by their functions, the function of uncharacterized protein families can be read directly from the map. Examples are shown in Figures 4 and 5. In Figure 5C, the ybgJ family of proteins, also known as the "domain of unknown function 213" family (DUF213), can now be assigned a role in core metabolism, potentially linked to acetyl co-A metabolism. Additionally, the bacterial-specific domain of unknown function 1 family (DUF1) is of unknown function but is implicated in signaling.[38–40] In Figure 4, the DUF1 family is located adjacent to many other proteins of signal transduction, and is especially tightly coupled to two component sensor kinases, implicating these proteins in bacterial signal transduction *via* interactions with sensor kinases. Other examples (not shown) include two families of conserved, uncharacterized proteins, the MJ1359 and rtcB families, which can be assigned a general role in DNA/RNA maintenance based on their global position in the map. The MJ1359 family is linked to DNA repair. The rtcB family, also known as the UPF0027 uncharacterized protein family,[41] is a conserved gene family of unknown function; here, we link these proteins to helicase and DNA polymerase I activity. Predicted functions for 23 uncharacterized protein families are listed in Table 1.

Discussion

In summary, we present an algorithm for the effective visualization of very large biological

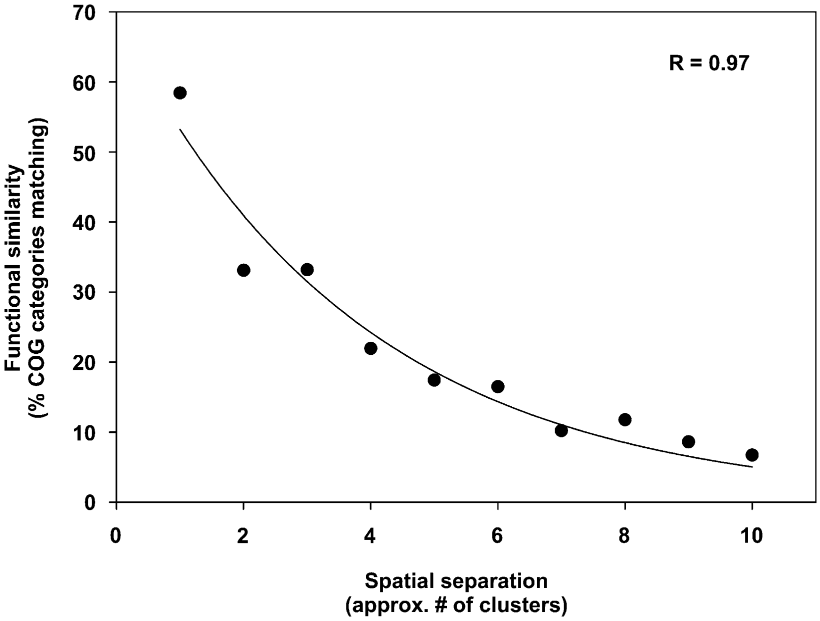

Figure 6. Neighboring proteins tend to be in the same cellular system. The tendency for proteins operate in the same cellular system, as defined by the percentage of matching assignments into the 18 COG database[17,37] pathways, is plotted against the spatial separation in multiples of a typical cluster size. The functional similarity decays exponentially with distance proportional to the function $e^{-0.26d}$ where d is a typical cluster diameter.

networks, based on iterative spring-based layout guided by a MST of the network. We show that this strategy is extremely effective for clearly visualizing the complex internal structure of large networks (Figure 8). We apply the LGL algorithm to explore an approach for studying protein–protein relationships by sequence comparisons, rather than analyzing a few sequences at a time, a

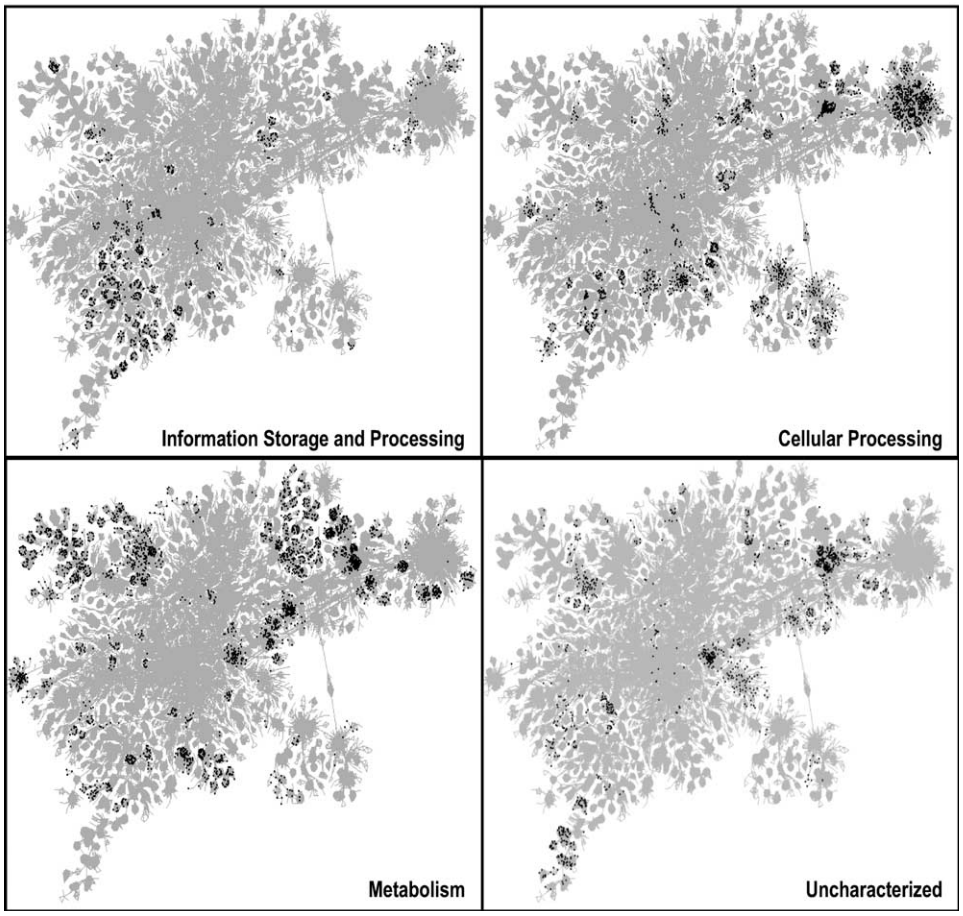

Figure 7. Extended map regions are composed of proteins of related function. The plot shows proteins from the four major functional classes defined in the COG database highlighted (bold vertices) within the protein homology map (gray lines). Adjacent regions of the map tend to include proteins that operate in the same broad cellular processes. Proteins not classified in COGs or defined in multiple COG categories are not highlighted.

Table 1. Functions assigned to uncharacterized protein families on the basis of adjacent characterized protein families observed in the global protein homology map

Uncharacterized protein family	Predicted function
MJ1633 family of conserved hypothetical proteins	Function linked to yhfB and MG371 families, inferred connection to nucleotidyl transferase activity (specific links to poly(A) polymerase and tRNA nucleotidyl transferase activities)
YhfB family of conserved hypothetical proteins	Function linked to MJ1633 and MG371 families, inferred connection to nucleotidyl transferase activity (specific links to poly(A) polymerase and tRNA nucleotidyl transferase activities)
MG371 family of conserved hypothetical proteins	Function linked to MJ1633 and yhfB families, inferred connection to nucleotidyl transferase activity (specific links to poly(A) polymerase and tRNA nucleotidyl transferase activities)
HI1730 family of conserved hypothetical proteins	Involved in core metabolism, precise function linked to activity of amidases and three component alpha,beta,biotin carboxylases and DUF213 family
Domain of unknown function DUF213	Involved in core metabolism, precise function linked to activity of amidases and three component alpha,beta,biotin carboxylases and HI1730 family
YbaE family of conserved hypothetical proteins	Metabolism, linked to phosphoglycerate mutase (glycolysis), genes of tryptophan synthesis, periplasmic peptide binding proteins, and PA1729 family
PA1729 family of conserved hypothetical proteins	Metabolism, linked to phosphoglycerate mutase (glycolysis), genes of tryptophan synthesis, periplasmic peptide binding proteins, ybaE family, and mll8746 family
mll8746 family of conserved hypothetical proteins	Metabolism, linked to phosphoglycerate mutase (glycolysis), genes of tryptophan synthesis, periplasmic peptide binding proteins, ybaE family, and PA1729 family
Pfam domain PF01549 family, also called DUF 18 domain of unknown function	Cell-wall/membrane metabolism or degradation, linked to chitinases, lipoproteins and endopeptidases
Rv0867c family of conserved hypothetical proteins	Cell-wall/membrane metabolism or degradation, linked to chitinases and glycosyl hydrolases
Tubby protein family	Involvement in oxidative stress response, with linkages to redoxins
yebA family of conserved hypothetical proteins	Membrane or cell wall associated systems, inferred link to lipoproteins and/or sugar epimerase activities, and uncharacterized protein family UPF0036
Uncharacterized protein family UPF0036	Membrane or cell wall associated systems, inferred link to yebA conserved hypothetical protein family and monooxygenases
Uncharacterized protein family UPF0028	Involved in cell surface and transport, and linked to transmembrane drug efflux proteins and ftsA cell division proteins
Uncharacterized protein family UPF0130	Inferred membrane/lipid-related function, linked to myrosinase-binding protein and jasmonate-inducible protein homologs and Vng1117c family conserved hypothetical proteins
Vng1117c family of conserved hypothetical proteins	Inferred membrane/lipid-related function, linked to myrosinase-binding protein and jasmonate-inducible protein homologs and UPF0130 uncharacterized protein family
Three protein families tightly linked in map: domains of unknown function DUF1 and DUF2, and GGDEF proteins. The proteins group into three tightly overlapping regions in the map	Tightly linked to signal transduction, especially to two component sensor kinases
MJ1359 protein family of conserved hypothetical proteins	General role in DNA/RNA maintenance, linked in map to DNA repair and glucosyl transferases
Pfam domain PF02206 and PF01757 protein families	Membrane-related function, linked to membrane transport proteins, as well as Pfam PF01838 proteins and UPF0051 proteins
Uncharacterized protein family UPF0027 (also known as rtcB family proteins)	General role in DNA/RNA maintenance, function linked to helicases and DNA replication/repair
Uncharacterized protein family UPF0051	Linked to Pfam PF02206, PF01757, and PF01838 protein families

map of proteins is created from billions of sequence comparisons. Although construction of such protein homology networks has been used to identify distant members of protein sequence families,[17,19,28,32–34] when visualized in the manner we describe, the presence of fusion proteins induces a higher order structure to the map and serves as one of the dominant forces dictating the arrangement of the protein families.

Gene fusions have been used to suggest protein interactions,[22,23,25,26] but the extent to which the fusions occur has been difficult to appreciate. The map makes clear the ubiquity of such fusions. Beyond simply summarizing information from fusions alone, the map effectively serves as a single qualitative, unifying theoretical framework for inferring protein function, which incorporates protein sequence homology (through direct map edges), remote sequence homology (as proteins not directly connected but in the same cluster), gene fusions (as linked clusters), and transitive gene fusions (as chains of linked clusters). The absences of detectable homology and fusions are also captured in the network, and these "negative"

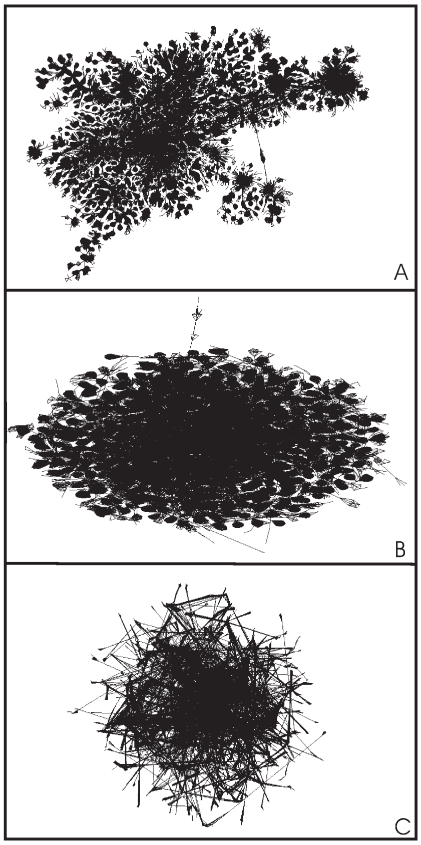

Figure 8. A comparison of LGL with map layouts produced by other algorithms. The layout of the protein homology map by LGL (A) is contrasted with the layout of the same network by the spring-force algorithm only, lacking the minimal spanning tree calculation and iterative layout procedure (B), and with the layout by the approach of InterViewer.[43] Interviewer collapses equivalent nodes into single nodes, thereby simplifying the graph, and is one of the few available graph layout programs that scales to such large networks. The layout from LGL reveals more of the internal graph structure than the other approaches tested.

trends also influence the layout. The spatial layout of this map represents the simultaneous satisfaction of all of these constraints, each constraint providing functional information about the proteins. Possible functions of uncharacterized proteins may then be suggested from their positions in the resulting map. Systematic analysis of this sort may indicate new connections between cellular systems, aid discovery of protein functions, as well as visually illustrate the complex evolutionary histories of genes and proteins.

An interesting question is how promiscuous domains,[22] which are essentially domains found fused to numerous other domains, affect this approach. Such domains should have the effect of organizing many proteins around them. It is likely that this trend accounts for the central location of the protein kinases in Figure 4. We suspect that predicting function with the protein homology map may be more appropriately applied to non-promiscuous domains, although it is possible that as the sequence similarity threshold used to create the map becomes more stringent, such promiscuous domain families may be broken up into subfamilies, and the effects of the promiscuity on the layout consequently reduced. In this context, it may prove interesting to compare protein homology maps generated only from eukaryotic genomes to maps generated from prokaryotes to see how the enhanced eukaryotic propensity for multidomain proteins affects the map structure.

The algorithm used to visualize such a large graph clearly affects the functional inferences drawn. Beyond the choice of algorithm, the dimensionality of the map is important as well, as the protein homology network itself is actually a high dimensional mathematical object, likely to be substantially distorted when visualized in only two or three dimensions. Although we have chosen two dimensions for the work presented here, it is reasonable to expect that higher dimensional spatial layouts could perform better for preserving the relationships in the data and therefore for predicting protein function, although the visual interpretation of such maps would clearly be much more difficult. We have chosen the above approaches largely to illustrate the concept; the choice of dimensionality and layout algorithm parameters can be optimized in the future for better functional prediction. However, the analysis presented here demonstrates the potential of LGL for exploring sequence and functional relationships among extremely large sets of proteins. Other applications include visualizing genome or EST sequence assemblies, protein interaction, metabolic or regulatory networks; and other applications that would benefit by visualizations of large graphs. Due to its scalability, LGL has recently also been applied by the Opte Project to visualize the structure of the Internet†.

Methods

Calculating the total force on a vertex

The total force produced by the attractive and repulsive forces on any one vertex u for each time step is calculated as:

$$\begin{aligned}\vec{F}_{\mathrm{u,total}} &= \vec{F}_{\mathrm{u,attractive}} + \vec{F}_{\mathrm{u,repulsive}} \\ &= -k_{\mathrm{a}} \sum_{i=1}^{e} (|\vec{x}_i| - a) - k_{\mathrm{r}} \sum_{j=0}^{m} (|\vec{x}_j| - r) \qquad (1)\end{aligned}$$

with $k_{\mathrm{r}} = 0$ for $|x_j| > r$, where k_{a} and k_{r} are the attractive and repulsive spring constants, a is the equilibrium length of the spring connected to adjacent vertex i, $|\vec{x}_j|$ is the Euclidean distance of separation between the two vertices sharing the edge (the spring), e is the number of edges connected to vertex u, $|\vec{x}_j|$ is the distance of separation between the current vertex and a neighboring

† http://www.opte.org

vertex j, and m is the number of localized vertices satisfying $|\vec{x}_j| < r$. The repulsive spring force is only applicable if any two vertices are closer than r, hence its force is always positive (repulsive).

The iterative network layout algorithm

The network layout is guided by the MST, which uses the BLAST E-values as weights. First, the MST is determined for the network using Kruskal's algorithm.[42] The MST is used to guide the progression of the layout starting at the root vertex, v_{root}, which is the first vertex in the network to be assigned coordinates. The root vertex can be arbitrarily selected to emphasize different aspects of the network, or can be chosen depending on the centrality of the vertex in the network. In the latter case, the root vertex is defined by identifying the vertex that minimizes the number of edges that must be traversed in the MST to reach every other vertex. More precisely, $v_{\text{root}} = \min(\sum_{(v,u)\in V} d(v,u))$, where $d(v,u)$ is the minimum number of edges that vertex v must traverse to reach another vertex u. After generating the MST, each vertex is then assigned a level based on the MST, which is simply $d(v_{\text{root}}, u)$. For example, the root vertex is level zero and its adjacent vertices, vertices that it shares an edge with in the MST, are level one; level two vertices would then be adjacent to level one vertices, but not included in a previous level.

Starting with v_{root}, each level is laid out in turn as diagrammed in Figure 1. Each successive iteration proceeds until vertex positions change by less than a distance threshold of approximately 10^{-6} units, with an iteration limit of 150. The next layer of vertices from the MST is placed onto the grid on the surface of a new sphere, $\vec{S}_{\text{child}}$, using the linear combination of two vectors. The first vector is the current layout center of mass, $\vec{M}$, and the other vector is $\vec{P}$, the one separating the parent vertex, $\vec{x}_{\text{parent}}$ at current level -1, and the grandparent, $\vec{x}_{\text{grandparent}}$ at current level -2. The equation to place the next level (children of a given vertex) onto a sphere at position $\vec{S}$ is then proportional to the sum of those two vectors $\vec{M}$ and $\vec{P}$:

$$\vec{S}_{\text{child}} = c\left(\frac{\vec{M}}{|\vec{M}|} + \frac{\vec{P}}{|\vec{P}|}\right) + x_{\text{parent}} \tag{2}$$

$$\vec{M} = \frac{1}{|V_{\text{current}}|}\sum_{v'\in V_{\text{current}}} \vec{x}$$

$$\vec{P} = \vec{x}_{\text{parent}} - \vec{x}_{\text{grandparent}}$$

where c is a constant, $|\vec{M}|$ is the magnitude of $\vec{M}$, $|\vec{P}|$ is the magnitude of $\vec{P}$, V_{current} represents all vertices in the current layout, and $|V_{\text{current}}|$ is the number of vertices in the current layout. Again, edges are introduced between the vertices present at this stage of the layout. This three step process: placing the children of a vertex on a sphere according to equation (2); edge repopulation; and position refinement according to equation (1), continues until all vertices of the original network are at rest on the grid.

Layout visualization

We developed a stand-alone Java program, lglview, to interactively display and explore the resulting 2D networks from a given layout. This program allows searches for vertices, coloring, labeling and zooming into edges and vertices. For 3D layouts, a Perl program, genVrml.pl, was developed to represent networks in virtual reality modeling language (VRML), which is subsequently viewed through one of the many VRML browsers freely available on the Internet†. Combinations of 3D spatial layout and VRML were used for interactively visualizing networks with fewer than 20,000 edges due to high memory allocation of the visualization process. Both the 2D network viewer and the program to prepare coordinates for VRML are freely available for download‡.

Generating the protein homology map

To construct the map, we compared the amino acid sequences of each of the 145,579 known proteins from 50 complete genomes (from the NCBI Entrez Genome web site) with each other using the program BLASTP,[14] using default settings. The results of these ~21 billion comparisons were summarized as described in Figure 2: each protein was represented as a vertex in a network, and each significant BLAST similarity was represented as an edge connecting the corresponding proteins. This produces a directed network, since the edges have direction: the score from protein a to protein b, and the score from protein b to protein a. The network was converted to an undirected network by creating a single edge between two connected proteins and retaining the more significant of the two BLAST E-values as the weight. Layout with LGL required 211 minutes on an Intel Pentium single processor computer.

Several protein homology maps were calculated corresponding to different BLAST score thresholds ranging from 1×10^{-4} to 1×10^{-90}. Our general findings are consistent regardless of threshold; we present results for the map with highly significant BLAST expectation scores of $<1\times10^{-12}$. The list of genomes included in this analysis is included as Supplemental Material (Table 1), while the lists of genes, the protein homology map, and supporting data are available§.

Calculation of functional similarity

Functionally similar proteins were defined as proteins with mutual membership in one of the known 18 COG categories.[17,37] The functional similarity across a map region was calculated as the probability of two proteins belonging to the same COG category as a function of Euclidean distance between proteins, with distance in units of typical cluster size and equal to the spring equilibrium distance of equqtion (1). Proteins that fall within a multiple of the unit distance are binned, so the functional similarity between two proteins as a function of the distance in the map between the proteins, FS(c), is equal to:

$$\text{FS}(c) = \frac{1}{N_c}\sum_{i=1}^{N_c} P_c$$

where c is the binned Euclidean distance between two proteins rounded up to the nearest integer, N_c is the number of COG annotated protein pairs in distance bin c, and P_c is the number of annotated protein pairs with the same COG category in distance bin c.

† http://www.web3d.org

‡ http://bioinformatics.icmb.utexas.edu/lgl

§ http://bioinformatics.icmb.utexas.edu/phg

Acknowledgements

Supported by the Welch Foundation (F-1515), the Texas Advanced Research Program, the National Science Foundation, a Packard Fellowship, and a Camille and Henry Dreyfus New Faculty Award. We thank Hal Burch of Carnegie Mellon University for calculating initial map coordinates prior to the development of LGL, Sarah Archer-Evans for critical comments and discussion, and Reuben Reyes for advice on VRML.

References

1. Landers, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, L. *et al.* (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.
2. Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G. *et al.* (2001). The sequence of the human genome. *Science*, **291**, 1304–1351.
3. Xenarios, I., Fernandez, E., Salwinski, L., Duan, X. J., Thompson, M. J., Marcotte, E. M. & Eisenberg, D. (2001). DIP: the database of interacting proteins: 2001 update. *Nucl. Acids Res.* **29**, 239–241.
4. Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S.-L. *et al.* (2002). Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, **415**, 180–183.
5. Sirava, M., Schafer, T., Eiglsperger, M., Kaufmann, M., Kohlbacher, O., Bornberg-Bauer, E. & Lenhof, H. P. (2002). BioMiner—modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics*, **18**, S219–S230.
6. Kuffner, R., Zimmer, R. & Lengauer, T. (2000). Pathway analysis in metabolic databases *via* differential metabolic display (DMD). *Bioinformatics*, **16**, 825–836.
7. Goesmann, A., Haubrock, M., Meyer, F., Kalinowski, J. & Giegerich, R. (2002). PathFinder: reconstruction and dynamic visualization of metabolic pathways. *Bioinformatics*, **18**, 124–129.
8. Batagelj, V. & Mrvar, A. (1998). Pajek: a program for large network analysis. *Connections*, **21**, 47–57.
9. Enright, A. J. & Ouzounis, C. A. (2001). BioLayout—an automatic graph layout algorithm for similarity visualization. *Bioinformatics*, **17**, 853–854.
10. Ju, B., Park, B., Park, J. & Han, K. (2003). Visualization and analysis of protein interactions. *Bioinformatics*, **19**, 317–318.
11. Ideker, T., Ozier, O., Schwikowski, B. & Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18**, S233–S240.
12. Chervitz, S. A., Aravind, L., Sherlock, G., Ball, C. A., Koonin, E. V., Dwight, S. S. *et al.* (1998). Comparison of the complete protein sets of worm and yeast: orthology and divergence. *Science*, **282**, 2022–2028.
13. Rubin, G. M., Yandell, M. D., Wortman, J. R., Gabor Miklos, G. L., Nelson, C. R., Hariharan, I. K. *et al.* (2000). Comparative genomics of the eukaryotes. *Science*, **287**, 2204–2215.
14. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.* **25**, 3389–3402.
15. Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S. R. *et al.* (2002). The Pfam protein families database. *Nucl. Acids Res.* **30**, 276–280.
16. Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M. *et al.* (2001). The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucl. Acids Res.* **29**, 37–40.
17. Tatusov, R. L., Koonin, E. V. & Lipman, D. J. (1997). A genomic perspective on protein families. *Science*, **278**, 631–637.
18. Yona, G., Linial, N. & Linial, M. (1999). ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins: Struct. Funct. Genet.* **37**, 360–378.
19. Enright, A. J., Van Dongen, S. & Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucl. Acids Res.* **30**, 1575–1584.
20. Ohno, S. (1970). *Evolution by Gene Duplication*, Springer, Heidelberg, Germany.
21. Lynch, M. & Conery, J. S. (2000). The evolutionary fate and consequences of duplicate genes. *Science*, **290**, 1151–1155.
22. Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O. & Eisenberg, D. (1999). Detecting protein function and protein–protein interactions from genome sequences. *Science*, **285**, 751–753.
23. Enright, A. J., Iliopoulos, I., Kyrpides, N. C. & Ouzounis, C. A. (1999). Protein interaction maps for complete genomes based on gene fusion events. *Nature*, **402**, 86–90.
24. Snel, B., Bork, P. & Huynen, M. (2000). Genome evolution. Gene fusion *versus* gene fission. *Trends Genet.* **16**, 9–11.
25. Yanai, I., Derti, A. & DeLisi, C. (2001). Genes linked by fusion events are generally of the same functional category: a systematic analysis of 30 microbial genomes. *Proc. Natl Acad. Sci. USA*, **98**, 7940–7945.
26. von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P. & Snel, B. (2003). STRING: a database of predicted functional associations between proteins. *Nucl. Acids Res.* **31**, 258–261.
27. Yona, G. & Levitt, M. (2000). Towards a complete map of the protein space based on a unified sequence and structure analysis of all known proteins. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 395–406.
28. Yona, G., Linial, N., Tishby, N. & Linial, M. (1998). A map of the protein space—an automatic hierarchical classification of all protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **6**, 212–221.
29. Linial, M., Linial, N., Tishby, N. & Yona, G. (1997). Global self-organization of all known protein sequences reveals inherent biological signatures. *J. Mol. Biol.* **268**, 539–556.
30. Farnum, M. A., Xu, H. & Agrafiotis, D. K. (2003). Exploring the nonlinear geometry of protein homology. *Protein Sci.* **12**, 1604–1612.
31. Cheswick, B., Burch, H. & Branigan, S. (2000). Mapping and visualizing the internet. *Proc. Usenix Annual Technical Conference*, June 18–23, San Diego, CA.
32. Enright, A. J. & Ouzounis, C. A. (2000). GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics*, **16**, 451–457.
33. Abascal, F. & Valencia, A. (2002). Clustering of proximal sequence space for the identification of protein families. *Bioinformatics*, **18**, 908–921.

34. Sharan, R. & Shamir, R. (2000). CLICK: a clustering algorithm with applications to gene expression analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 307–316.
35. Fischer, D. & Eisenberg, D. (1999). Finding families for genomic ORFans. *Bioinformatics*, **15**, 759–762.
36. Kanehisa, M. & Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucl. Acids Res.* **28**, 27–30.
37. Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S. *et al.* (2001). The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucl. Acids Res.* **29**, 22–28.
38. Hecht, G. B. & Newton, A. (1995). Identification of a novel response regulator required for the swarmer-to-stalked-cell transition in *Caulobacter crescentus*. *J. Bacteriol.* **177**, 6223–6229.
39. Tal, R., Wong, H. C., Calhoon, R., Gelfand, D., Fear, A. L., Volman, G. *et al.* (1998). Three cdg operons control cellular turnover of cyclic di-GMP in *Acetobacter xylinum*: genetic organization and occurrence of conserved domains in isoenzymes. *J. Bacteriol.* **180**, 4416–4425.
40. Schultz, J., Milpetz, F., Bork, P. & Ponting, C. P. (1998). SMART, a simple modular architecture research tool: identification of signaling domains. *Proc. Natl Acad. Sci. USA*, **95**, 5857–5864.
41. Bairoch, A. & Apweiler, R. (2000). The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucl. Acids Res.* **28**, 45–48.
42. Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *In Proceedings of the American Mathematical Society*, **7**, 48–50. 1956.
43. Han, K. & Ju, B. H. (2003). A fast layout algorithm for protein interaction networks. *Bioinformatics*, **19**, 1882–1888.

Edited by F. E. Cohen

(Received 8 December 2003; received in revised form 16 April 2004; accepted 22 April 2004)

Supplementary Material for this paper is available on Science Direct