Biomine: A Network-Structured Resource of Biological Entities for Link Prediction

Lauri Eronen, Petteri Hintsanen, and Hannu Toivonen

Department of Computer Science and HIIT, University of Helsinki, Finland
firstname.lastname@cs.helsinki.fi

Abstract. Biomine is a biological graph database constructed from public databases. Its entities (vertices) include biological concepts (such as genes, proteins, tissues, processes and phenotypes, as well as scientific articles) and relations (edges) between these entities correspond to real-world phenomena such as "a gene codes for a protein" or "an article refers to a phenotype". Biomine also provides tools for querying the graph for connections and visualizing them interactively.

We describe the Biomine graph database. We also discuss link discovery in such biological graphs and review possible link prediction measures. Biomine currently contains over 1 million entities and over 8 million relations between them, with focus on human genetics. It is available on-line[1] and can be queried for connecting subgraphs between biological entities.

1 Introduction

Biomine is a large biological graph (or BisoNet [1]) whose entities (vertices) include concrete biological concepts such as genes, proteins and tissues, but also abstract concepts such as biological processes, phenotypes and scientific articles. Relations (edges) between these entities correspond to real-world phenomena such as "a gene codes for a protein" or "an article refers to a phenotype". We are motivated by link discovery in such biological graphs with the primary aim of prioritising putative disease-susceptibility genes.

A generic goal of Biomine is to help users discover and understand relations between biological entities, such as indirect connections between a gene and a disease. In the context of bisociative or creative information exploration [2], our aim is to facilitate discovery of bisociations between biological entities that are not connected within a single existing database. As a concrete and motivating example, consider a gene mapping process for a disease (or other phenotype). Current genome-wide analysis methods produce a large number of candidate genes, i.e., putative disease-susceptibility genes for the disease. A question then is how to prioritize these genes so that further efforts can be focused on the most promising candidates. One approach is to look at what is already known about the putative disease genes and see how they relate to each other and

[1] http://biomine.cs.helsinki.fi

M.R. Berthold (Ed.): Bisociative Knowledge Discovery, LNAI 7250, pp. 364–378, 2012.

to the phenotype under study. This might reveal evidence for the hypothesised association or facilitate a more detailed hypothesis about the mechanisms of the relationship. Due to the lack of automated methods the work is mostly done by manually browsing the databases. This is a slow and laborious process which necessarily limits the extent and coverage of the search. In this chapter we describe a database and methods for (partial) automation of the prioritising task.

Biological graphs can be built from publicly available biological databases. Converting (relational) biological knowledge to a graph form is conceptually simple though not straightforward. For instance, how to map different biological concepts and their attributes into the graph and how to weight edges is non-trivial. In Sections 2 and 3 we consider these issues in the context of Biomine, a relatively large biological graph. In Section 4 we then review some proposed link goodness measures and consider the evaluation of link significance. We briefly review related work in Section 5 and conclude in Section 6.

2 Biomine Database

We now describe in more detail *Biomine*, a large index of various interlinked public biological databases. Biomine offers a uniform view to these databases by representing their contents as a large, heterogeneous graph, with probabilistic edges. Vertices in this graph represent entities (records) in the original databases and edges represent their annotated relationships (cross-references between records). Edges have weights that are interpreted as probabilities. A preliminary version of Biomine has been described by Sevon et al [3]. In this section we take a brief look at the core components of Biomine: its data model and source databases. Edge weighting is considered separately in Section 3.

2.1 Data Model

The choice of *data representation*, or *data model*, is important in link mining [4]. To facilitate wide applicability, the core Biomine data model is deliberately simple: all source database records are represented as vertices in an undirected, labelled and weighted multigraph $G = (V, E)$. The elements of the vertex set V are biological entities such as genes, proteins and biological processes as well as more general objects like article abstracts. They are labelled by a type, such as *gene* or *protein*, from set T_v. We denote the vertex type mapping by $t_v : V \mapsto T_v$.

Edge multiset $E \subset [V]^2$ consists of unordered vertex pairs $\{u, v\}$. As with vertices, edges have labels from edge type set T_e and we denote this mapping by $t_e : E \mapsto T_e$. Edge types depict annotated relations between vertices, such as *codes for* (e.g., gene *codes for* protein) or *refers to* (e.g., article *refers to* gene).

Each edge has a source database where the corresponding relation resides. We denote this source database mapping by $s : E \mapsto \mathcal{D}$ where $\mathcal{D}$ is the set of source databases. For a given graph $G = (V, E)$, we refer to its vertex set V by $V(G)$

and its edge set E by $E(G)$. Finally, we denote the set of neighbouring vertices of v by $N(v) = \{u \in V : \{v, u\} \in E\}$.

Table 1 lists the vertex types used in Biomine; similarly, Table 2 lists the edge types. Some representative examples of typed edges are given in Table 3. All tables refer to the Biomine database built at 4th June 2010.

Table 1. Biomine vertex types T_v, primary source database for each type, and the total amount and mean degrees of corresponding vertices

Type	Primary source database	Amount	Mean degree
Active site	InterPro	89	95.82
Allelic variant	OMIM	19,455	1.44
Article	PubMed	532,675	3.98
Binding site	InterPro	62	111.18
Biological process	GO	19,539	32.47
Cellular component	GO	2,856	122.18
Compound	KEGG	15,879	0.55
Conserved site	InterPro	575	58.29
Domain	InterPro	5,515	69.55
Drug	KEGG	8,846	0.69
Enzyme	KEGG	5,095	10.15
Family	InterPro	12,718	10.61
Gene	Entrez Gene	192,893	18.60
Gene/Phenotype	OMIM	343	82.35
Genomic context	Entrez Gene	11,825	18.68
Glycan	KEGG	2,519	0.92
Homolog group	HomoloGene	25,780	3.18
Molecular function	GO	9,529	49.07
Ortholog group	KEGG	13,067	3.81
Pathway	UniProt	1,875	37.10
Phenotype	OMIM	6,559	16.95
PTM	InterPro	16	82.88
Protein	UniProt	275,292	29.58
Region	InterPro	1,441	20.14
Repeat	InterPro	255	94.84
Tissue	UniProt	1,317	189.10
		total 1,166,020	14.84

2.2 Source Databases

Biomine essentially is an index to several interlinked, publicly available *source databases*. Each database provides different kinds of entities and relations to Biomine, some overlapping. We briefly review the main features of the source databases below.

NCBI's *Entrez Gene* [5,6] provides gene entries for different organisms. Currently, Biomine contains five model organisms: human, mouse, rat, fruit fly and

Table 2. Biomine edge types T_e and amount of edges of each type

Type	Source databases	Amount
affects	Entrez Gene	5,077
belongs to	Entrez Gene, HomoloGene, KEGG, STRING, SwissProt, TrEMBL	689,026
codes for	Entrez Gene, KEGG, STRING	174,480
contains	SwissProt, TrEMBL	454,553
functionally associated to	STRING	2,916,286
has	Entrez Gene, InterPro, KEGG, OMIM, SwissProt, TrEMBL	464,369
has synonym	Entrez Gene	1,666
interacts with	Entrez Gene, SwissProt, TrEMBL	97,361
is a	GO, InterPro, KEGG	51,483
is expressed in	SwissProt, TrEMBL	234,153
is found in	Entrez Gene, InterPro, KEGG, SwissProt, TrEMBL	337,542
is homologous to	HomoloGene	259,390
is located in	Entrez Gene, OMIM	144,495
is part of	GO, InterPro, OMIM	54,196
is related to	GO, HomoloGene, KEGG, OMIM, SwissProt, TrEMBL	25,414
overlaps	OMIM	8,199
participates in	Entrez Gene, InterPro, KEGG, SwissProt, TrEMBL, UniProt	605,237
refers to	Entrez Gene, KEGG, OMIM, SwissProt, TrEMBL	2,216,614
subsumes	Entrez Gene, KEGG, STRING, SwissProt, TrEMBL	140,555
targets	KEGG	4,885
		total 8,884,981

Table 3. Some examples of Biomine edge types, their source databases and the amount of corresponding edges. Observe that a sequence of such edges would constitute a gene–gene path in the graph.

Edge	Source database	Amount
Gene *codes for* Protein	STRING	5,948
Protein *belongs to* Family	SwissProt	30,651
Family *participates in* Biological process	InterPro	5,274
Biological process *is related to* Tissue	GO	13,103
Protein *is expressed in* Tissue	SwissProt	176,034
Enzyme *subsumes* Protein	TrEMBL	7,907
Gene *codes for* Enzyme	KEGG	14,195

nematode (*Caenorhabditis elegans*). Genes are connected to their protein products and other homologous genes (similar genes in different organisms). Homology relations come from an another Entrez database *HomoloGene* [6]

UniProt [7] is the main source of protein-related information. Its core elements are proteins, pathways and tissues. These elements form vertices in the graph. Manually annotated and reviewed proteins are in *Swiss-Prot* subdatabase, while *TrEBML* subdatabase contains automatically annotated and nonreviewed proteins. UniProt contains many relations, such as protein interactions and expressions, and classifications into protein families and pathways.

InterPro [8] is another protein-related database. It indexes protein families and structural elements (domains, regions, sites, etc.), and it has hierarchies for these elements. The third protein database, STRING [9], contains known and predicted protein–protein interactions. The interactions include direct (physical) and indirect (functional) associations. STRING also contains clusters of orthologous groups (COGs) and their interactions, with mappings between proteins and COGs.

Gene Ontology (GO) aims to provide a controlled vocabulary for genes and gene products [10]. Its core domains are cellular components, biological processes and molecular functions. The ontology is structured as a directed acyclic graph and each term has defined relationships to one or more other terms in the same domain and sometimes to other domains. This graph is a subgraph of Biomine, and the term vertices are referred to by other databases such as Entrez Gene and UniProt.

Online Mendelian Inheritance in Man (OMIM) is a catalogue of human genes and genetic disorders. It is the main source of phenotype information in Biomine: most of the OMIM entries are *Phenotype* vertices. The database also contains descriptions of allelic variants, gene locations and a large number of references to biomedical literature.

PubMed [6] is a freely accessible online database of biomedical journal citations and abstracts with approximately 20 million entries at the time of writing. Many biological databases (such as UniProt and OMIM) contain references to PubMed entries, for example to index articles where a particular gene or phenotype is mentioned. In Biomine these cross-referenced PubMed entries are *Article* vertices.

Kyoto Encyclopedia of Genes and Genomes (KEGG) is a large, integrated database resource consisting of 16 main databases broadly categorised into systems information, genomic information and chemical information [11]. Biomine uses a subset of KEGG: its pathway, gene, drug, orthology, compound and glycan databases.

Each of the source databases has its own schema for arranging and formatting data. Raw data files are preprocessed into a uniform intermediate format before integration. Intermediate format files are essentially lists of typed edges, vertex attributes and synonym mappings. These files are then imported into a single database to form a large graph. During the importing process synonyms, invalid references and other anomalies are resolved. The complete conversion and importing process is complicated and out of the scope of this chapter.

3 Edge Goodness in Biomine

One of the goals of Biomine is to allow discovery and evaluation of links between vertices specified by the user. To rank paths or assess the significance of a connection between two vertices we need a measure for edge goodness. Edges sometimes have natural weights in the source databases. For example, a homology between two proteins could have a value denoting the degree of sequence similarity. Biomine extends such domain-specific static weighting by considering edge weight, or *goodness*, as a function of three factors:

1. *Reliability.* How confident are we that the relation (and consequently the edge) really exists? How reliable is the data source, how reliable is the method used to produce or predict the edge and how strong or probable is the connection estimated to be in the data source?
2. *Relevance.* How relevant is the edge with respect to the query? We assume that the investigator can give query-specific weights for vertex and/or edge types according to his or her subjective opinions of the importance of each type for the query at hand.
3. *Rarity* of informativeness. How rare and informative is the edge? As an extreme example, an article [12] that refers to over 18,000 human and mouse genes is not likely to be relevant for a specific gene whereas an article that only refers to few genes is much more likely to be informative. In Biomine edge rarity is directly related to the degrees of its incident vertices.

A distinguishing feature of Biomine is the probabilistic interpretation of the above factors: an edge $e \in E$ is considered to be reliable with probability $r(e)$, relevant with probability $q(e)$ and rare (or *informative*) with probability $d(e)$. These factors are combined to a single probability $g(e)$ so that e is an existing and potentially useful relation if e is at the same time reliable, relevant and informative. In other words, edges are random: e "exists" or "is true" with probability $g(e)$, or "does not exist" or "is not true" with probability $1 - g(e)$. With the probabilistic interpretation G is a random graph that naturally models the uncertainty in the source data and the query-specific relevance. We next give definitions for r, q and d, and we combine them into one goodness g.

Reliability $r(e)$ of an edge $e \in E$ is defined as a product of two (independent) reliabilities: a database reliability $r_d : \mathcal{D} \mapsto [0, 1]$ and a relation (edge) reliability $r_r : E \mapsto [0, 1]$. The database reliability r_d is given by the user, and the interpretation of r_d is the degree of belief the user has for a relation being correctly annotated in the corresponding database. For example, the manually curated Swiss-Prot database could be given a perfect reliability by letting $r_d(\text{Swiss-Prot}) = 1.0$, while the computer-annotated TrEMBL database could be assumed to be less precise by letting $r_d(\text{TrEMBL}) = 0.75$. Relation reliability r_r comes from the source database instead: if there is a separate confidence value c associated to e (that reflects similarity or homology score, for example), we let $r_r(e) = c$, where c is scaled between 0 and 1 if needed. Otherwise we let $r_r(e) = 1$. The interpretation of $r_r(e)$ is the confidence of the data source itself on the relation represented by e.

We define the *edge reliability* $r : E \mapsto [0,1]$ by treating the reliabilities r_d and r_r as probabilities of independent events:

$$r(e) = r_d\big(s(e)\big) \cdot r_r(e) \tag{1}$$

where $s(e)$ is the source database of e. The interpretation of $r(e)$ is that e is reliable if both the database (as a whole) and the annotation are considered reliable.

Relevance $q(e)$ of an edge $e \in E$ is the degree of belief that e represents a relevant connection between vertices u and v with respect to the current query. Edge relevance is analogous to edge reliability r but, in contrary to the static database-related reliability, relevance is query-specific.

Relevance values may be sometimes easier to give in terms of vertex types instead of edge types. Hence Biomine uses two relevance functions: $q_v : T_v \mapsto [0,1]$ for vertex types and $q_e : T_e \mapsto [0,1]$ for edge types. Both q_v and q_e are given by the user. A practical implementation could have a default configuration for both q_v and q_e, so only few adjustments would be needed for a typical query.

As in (1), relevance values q_v and q_e are treated as probabilities of independent events. The *edge relevance* $q : E \mapsto [0,1]$ is

$$q(e) = q_e\big(t_e(e)\big) \cdot \sqrt{q_v\big(t_v(u)\big)} \cdot \sqrt{q_v\big(t_v(v)\big)} \tag{2}$$

where $e = \{u, v\} \in E$. Vertex relevance coefficient $\sqrt{q_v\big(t_v(x)\big)}$ in (2) decomposes the vertex type specific relevance $q_v\big(t_v(x)\big)$ of vertex x for each of its adjacent edges. As path relevance will be later defined as a product of edge relevance values this gives the desired outcome: the relevance of any path visiting a vertex of type τ is multiplied by $q(\tau)$.

We want to give lower scores for paths that visit vertices with high degrees: the higher the degree of vertex $v \in V$ the less likely it is that any two neighbours of v actually have an interesting connection through v. Hence we define *rarity* $d_v : V \mapsto [0,1]$ first for vertices. Rarity $d_v(v)$ represents the probability that any two edges incident on v are related to each other and represent a meaningful path; the higher the rarity, the more informative v is. The following ad hoc formula is used as a basis for rarity:

$$d_v(v) = \frac{1}{(\deg(v) + 1)^{\alpha}} \tag{3}$$

where $0 \leq \alpha \leq 1$ is a *penalising parameter*. It determines how steeply d_v decreases as a function of vertex degree. With $\alpha = 0$ we have $d_v(v) \equiv 1$ so that all vertices are considered equally informative. With $\alpha = 1$ we have $d_v(v) = (deg(v, +)1)^{-1}$ and $d_v(v)$ has the following probabilistic interpretation. Consider a random walker who, at any vertex, is equally likely to follow any edge or stop at the vertex. Given a path $P = (v_1, v_2, \ldots, v_k)$, $v_i \in V$, rarity $d_v(v_i)$ is the probability that the walker who has so far traversed vertices $v_1, \ldots, v_i$ will next stay on the path and visit vertex v_{i+1}.

The simple formula (3) can be too inflexible in practise. Take for example PLA2G7: a widely studied asthma gene that has been referred in 97 articles. Because of these article links (3) would penalise PLA2G7 vertex severely. However, it has only one interaction link and it participates in three biological processes, so PLA2G7 could be informative when the investigator is mostly interested in gene–gene interactions or biological processes. Another issue is that vertex degrees vary wildly between different vertex types (see Table 1) but α is independent of vertex types. This causes unreasonable penalisation for some large-degree vertex types such as GO terms.

To allow more flexibility in degree penalising we replace the single constant α and vertex degree function deg with vertex-type and edge-type specific functions $\alpha : T_v \mapsto [0,1]$ and $\deg : V \times T_e \mapsto \mathcal{N}$ (that is, $deg(v,\tau,)$ denotes the number of edges of type τ adjacent to v). Now the vertex *rarity* $d_v : V \times T_e \mapsto [0,1]$ for vertex $v \in V$ is

$$d_v(v,\tau) = \frac{1}{(\deg(v,\tau)+1)^{\alpha\left(t_v(v)\right)}}. \tag{4}$$

As with relevance (2), the rarity values are decomposed into edge-specific coefficients. The edge *rarity* $d : E \mapsto [0,1]$ becomes

$$d(e) = \sqrt{d_v\big(u,t_e(e)\big)} \cdot \sqrt{d_v\big(v,t_e(e)\big)} = \big[d_v\big(u,t(e)\big) \cdot d_v\big(v,t(e)\big)\big]^{-1/2} \tag{5}$$

where $e = \{u,v\} \in E$.

Now that we have defined all the components of edge goodness, the goodness $g : E \mapsto [0,1]$ itself is simply a product of those factors:

$$g(e) = r(e) \cdot q(e) \cdot d(e) \tag{6}$$

where $r(e)$, $q(e)$ and $d(e)$ are the reliability (1), relevance (2) and rarity (5) of an edge $e \in E$. Under the assumptions that $r(e)$, $q(e)$ and $d(e)$ are probabilities for mutually independent necessary conditions for the edge and that edges are independent of each other, the goodness $g(e)$ is the probability that e exists. We remark that these assumptions of independence are strong and in some cases they are arguably unrealistic. However, independence allows us to calculate path and subgraph probabilities easily; we return to these in Section 4.

4 Link Goodness Measures

A link is a more general concept of connection than a simple relation (edge) between two vertices s and t. Links are useful since they can be used to model indirect, weak or otherwise non-trivial connections. A *path* (a sequence of consecutive edges) is probably the simplest link type, but shared neighbourhoods, connected subgraphs and random walks can also be used to represent links. To discover or predict links, assess their strengths or analyse statistical significances of links we need a measure for link goodness in addition to edge goodness.

We next give a short review of some link goodness measures proposed in the literature. They are presented in the order of increasing generality; more general measures utilise more information to determine the strength of a link. The discussion is not restricted to Biomine graphs, so $G = (V, E)$ refers to an arbitrary directed or undirected graph below. See Liben-Nowell and Kleinberg [13] for an experimental evaluation of many of these measures for link prediction.

4.1 Path and Neighbourhood Level

The shortest s–t-path P is a simple but efficient link type. Its length $w(P)$ is a natural measure for link strength:

$$g_s(s,t) = \min_{P \in \mathcal{P}} w(P) = \min_{P \in \mathcal{P}} \sum_{e \in P} w(e) \tag{7}$$

where $w(e)$ is the length (weight) of an edge $e \in E$ and $\mathcal{P}$ is the set of all s–t-paths in G. This measure is easy and efficient to calculate by any shortest path algorithm.

For random graphs where edge "lengths" are probabilities, (7) does not make much sense. However, if edges are independent of each other, like in Biomine graphs, path "length" or goodness follows in a natural way. Let $P = (e_1, \ldots, e_k)$, $e_i \in E$, be a path in G. The *path goodness* $g_p : \mathcal{P} \mapsto [0, 1]$ is

$$g_p(P) = \prod_{e \in P} g(e). \tag{8}$$

With the interpretation that $g(e)$ is the probability that edge e exists (Section 3) the path goodness $g_p(P)$ is the probability that the whole path P exists in a realisation H of G. A *realisation* of G is a non-random subgraph $H \subset G$ where each edge of G has been randomly and independently decided according to the corresponding edge probability.

With path goodness g_p the shortest path corresponds to the most probable, or *best* path. By combining (7) and (8) we get

$$g_b(s,t) = \max_{P \in \mathcal{P}} g_p(P) = \max_{P \in \mathcal{P}} \prod_{e \in P} g(e). \tag{9}$$

Again, any shortest path algorithm can be applied to find most probable paths by using edge weights $w(e) = -\log\big(g(e)\big)$. Let P be the shortest path found with weight $w(P)$. Then

$$w(P) = \sum_{e \in P} -\log\big(g(e)\big) = -\log\big(\prod_{e \in P} g(e)\big) = -\log\big(g_p(P)\big) \tag{10}$$

and since the logarithm function is strictly increasing and $w(P)$ is minimised, $g_p(P)$ is maximised.

Overlapping vertex neighbourhoods may indicate indirect similarity or proximity. The number of overlapping neighbours is the simplest measure in this context:

$$g_n(s,t) = |N(s) \cap N(t)|. \tag{11}$$

This measure has been observed to positively correlate with future collaboration probability in coauthor networks [14]. The normalised form of (11)

$$g_J(s,t) = \frac{|N(s) \cap N(t)|}{|N(s) \cup N(t)|} \tag{12}$$

is the well known *Jaccard index*. Adamic and Adar have proposed [15] a modification of (12) that rewards vertex pairs that share neighbours with low degrees:

$$g_A(s,t) = \sum_{u \in N(s) \cap N(t)} \frac{1}{\log |N(u)|}. \tag{13}$$

4.2 Subgraph Level

The goodness of a single s–t-path, as in (7) and (9), is not necessarily a good measure of the strength of the link between vertices s and t. For example, a link consisting of several parallel paths could be considered to be stronger than a single path even if all of the parallel paths are weak. *Connection subgraphs* take this into account by evaluating connected subgraphs, which can be thought to be a set of paths, containing s and t. Specifically, a connection subgraph between s and t is a connected subgraph $H \subset G$, of a given size, such that $\{s,t\} \subset V(H)$. Subgraph H can be, for example, a set of k shortest paths for some fixed k or it can be chosen to maximise a given connection subgraph goodness function [16].

Faloutsos et al. view G as an electrical network of resistors [16]. They propose an algorithm that extracts a fixed size subgraph H which maximises total delivered current over the subnetwork from s to t when s is assigned a potential of $+1$ volt and t is grounded (0 volts). Total delivered current has a random walk interpretation [17]. At first, let us define transition probabilities

$$p(u,v) = \frac{g(u,v)}{\sum_{w \in N(u)} g(u,w)} \tag{14}$$

for each $(u,v) \in E$. Next, let p_{esc} denote the escape probability according to (14) from s to t; i.e. the probability that a random walker starting from s will reach t before returning to s. The *effective conductance* between s and t is now

$$g_{\text{EC}}(s,t) = \sum_{u \in N(s)} g(s,u) \cdot p_{\text{esc}} \tag{15}$$

which is the expected number of "successful escapes" when the number of escape attempts is $\sum_{u \in N(s)} g(s,u)$ [18].

Effective conductance is an appealing link goodness measure and it has been used to measure centrality in networks [19]. However, it does not penalise uninformative vertices that have large degrees (cf. (4)). Faloutsos et al. dodge this by introducing a global grounded "sink" vertex that is connected to all vertices $v \in V$ with conductance proportional to $\sum_{u \in N(v)} g(v,u)$. As pointed out by Koren et al. [17], this introduces a counterintuitive size bias where the link goodness can decrease if the connection subgraph is enlarged. They propose a modified version of (15) titled cycle-free effective conductance (CFEC):

$$g_{\mathrm{CFEC}}(s,t) = \sum_{u \in N(s)} g(s,u) \cdot p_{\text{cf-esc}}(s,t) = \sum_{u \in N(s)} g(s,u) \cdot \sum_{P \in \mathcal{P}} \Pr(P), \qquad (16)$$

where $p_{\text{cf-esc}}$ is the escape probability restricted to cycle-free random walks (walks that are simple s–t-paths) and $\mathcal{P}$ is the set of all simple s–t-paths in G. CFEC has two desirable properties: it is monotonically increasing as a function of graph size, and a relatively small connection subgraph consisting of the most probable simple s–t-paths is usually enough to approximate $g_{\mathrm{CFEC}}(s,t)$ [17].

4.3 Graph Level

A link goodness measure can utilise the topology of the whole graph G. Most measures on this scale are based on random walks like (15) and (16), although a measure proposed by Katz [20] considers sets of s–t-paths such that

$$g_K(s,t) = \sum_{l=1}^{\infty} \beta^l |\mathcal{P}_l| \qquad (17)$$

where $\mathcal{P}_l$ is the set of all s–t-paths of length l. Parameter $\beta > 0$ controls the effect of longer paths to the goodness.

Random walk models typically consider a single walker w starting from s or two walkers w_1 and w_2 with one starting from s and the other from t. Walkers traverse G randomly with transition probabilities (14). *Hitting time* $H(s,t)$ considers the expected number of steps w has to take to reach t [21]. Its symmetric variant is *commute time* $C(s,t) = H(s,t) + H(t,s)$. Both can be readily used as distance measures, and they have been used as link goodness (proximity) measures as well [13].

SimRank by Jeh and Widow [22] is based on a recursive definition

$$g_{\mathrm{SR}}(s,t) = \begin{cases} 0 & \text{if } N(s) = \emptyset \text{ or } N(t) = \emptyset, \\ 1 & \text{if } s = t, \\ C/(|N(s)||N(t)|) \cdot \sum_{\substack{u \in N(s) \\ v \in N(t)}} s(u,v) & \text{otherwise} \end{cases} \qquad (18)$$

where $C \in [0,1]$ is a constant. SimRank also has a random walk interpretation: value $g_{\mathrm{SR}}(s,t)$ corresponds to the expected value of $C^{\mathbf{t}}$ where $\mathbf{t}$ is the time (number of steps) when walkers w_1 and w_2 first meet [22].

Liben-Nowell and Kleinberg [13] proposed a *rooted PageRank* measure for link goodness based on the well known *PageRank* measure [23]. In rooted PageRank the random walker w returns to s with probability α in every step, or it continues the walk with probability $1 - \alpha$. The measure is the steady state (stationary) probability of t.

With random graphs $g_b(s,t)$ is the probability that the best path exists in a realisation of G. A more appropriate measure could be the probability that at least one path exists between s and t. This measure is closely related to the theory of *network reliability* [24], and the desired measure

$$g_R(s,t) = \Pr(H : H \subset G, H \text{ contains an } s\text{–}t\text{-path}), \tag{19}$$

where H is a random instantiation of the uncertain graph G, is the *two-terminal network reliability* of G with terminals s and t. (The connected parties are called terminals in the reliability literature.)

4.4 Estimation of Link Significance

We eventually want to measure how strongly two given vertices s and t are related in graph G. Link goodness measures, such as those discussed in Section 4, allow ranking of links but their values may be difficult to put into perspective. For example, assume we have $f(s,t) = 0.4$ for some goodness measure f. Is this particular value of f high or low? This obviously depends on the data and the specific instances of s and t.

We can estimate the statistical significance of the link by using the goodness value $f(s,t)$ as a test statistic. Returning to the previous example this tells us how likely it is to obtain a link with goodness 0.4 or better by chance. There are multiple meaningful null hypotheses:

N1. Vertices s and t of types $\tau_s \in T_v$ and $\tau_t \in T_v$ are not more strongly connected than randomly chosen vertices s' and t' of types τ_s and τ_t.

N2. Vertex s of type $\tau \in T_v$ is not more strongly connected to vertex t than a randomly chosen vertex s' of type τ.

N3. Vertices s and t are not more strongly connected in the given graph G than in random graph H with edge weights $w' : E(H) \mapsto \mathbb{R}$ generated by model $\mathcal{H}$ similar to the (unknown) model which generated G and w.

The last null hypothesis N3 is clearly the most complicated one: it is not easy to come up with model $\mathcal{H}$ that generates random graphs that are sufficiently similar to the observed graph. The choice from the first two null hypotheses depends on what we are testing. In a symmetrical case, for example when testing the significance of connection between two candidate genes, N1 is appropriate. If the roles of the vertices are asymmetric, as in testing for the connection from a set of candidate genes to a single phenotype, N2 should be used.

Under null hypothesis N1 we can estimate p-value for the test statistic $f(s,t)$ by randomly sampling N pairs of vertices (s',t') from V. Let us denote the sample by $S = \{(s_1,t_1),\ldots,(s_N,t_N)\}$. To obtain an empirical null distribution

we compute the value of test statistic $f(s_i, t_i)$ for each $(s_i, t_i) \in S$, and let $S_+ = \{(s_i, t_i) \in S : f(s_i, t_i) \geq f(s, t)\}$. Then the estimated p-value $\tilde{p}$ is simply

$$\tilde{p} = \frac{|S_+|}{N}. \tag{20}$$

The same procedure can be used under null hypothesis N2 by sampling single vertices $S = \{t_1, \ldots, t_n\}$ and letting $S_+ = \{t_i \in S : f(s, t_i) \geq f(s, t)\}$.

Because vertices of the same type may have wildly varying degrees one should sample vertices s' and t' that have degrees similar to s and t, respectively. If several hypotheses are tested (several candidate genes, for example), the resulting p-values should be adjusted accordingly to account for multiple testing.

5 Related Work

Concurrently with the development of Biomine, several other data integration systems have been proposed in the literature. Of these, most similar to our approach are ONDEX [25] and Biozon [26], which both collect the data from various sources under a single data store. They also use a graph data schema. In both systems, the data model is a graph with typed nodes and edges, allowing for the incorporation of arbitrary data sources. In addition to curated data derived from the source databases, both ONDEX and Biozon include in-house data such as similarity links computed from sequence similarity of proteins and predicted links derived by text mining. Biozon provides several types of queries, most interestingly searching by graph topology and ranking of nodes by importance defined by the graph structure. In ONDEX, the integrated data is accessed by a pipeline, in which individual filtering and graph layout operations may be combined to process the graph in application-specific ways. BioWarehouse [27] aims to provide generic tools for enabling users to build their own combinations of biological data sources. Their data management approach is rather similar to ONDEX and Biozon, but the data is stored in a relational database with a dedicated table for each data type instead of a generic graph structure. This approach allows database access through standard SQL queries, and is not directly suitable for graph-oriented queries.

6 Conclusion

We presented Biomine, a system that integrates data from a number of heterogenous sources into a single, graph-structured index. The current implementation of Biomine contains over 1 million entities and over 8 million relations between them, with focus on human genetics. The index can be queried using a public web interface[2], and results are visualized graphically. Biomine in its current form is a functional proof of concept, covering only part of the available data and with a limited focus on human genetics. Initial experimental results indicate that Biomine and other similar approaches have strong potential for predicting links and annotations.

[2] `biomine.cs.helsinki.fi`

Acknowledgements. We thank the Biomine team for co-operation. This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland (Grant 118653). We also thank the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract BISON-211898.

References

1. Kötter, T., Berthold, M.R.: From Information Networks to Bisociative Information Networks. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 33–50. Springer, Heidelberg (2012)
2. Dubitzky, W., Kötter, T., Schmidt, O., Berthold, M.R.: Towards Creative Information Exploration Based on Koestler's Concept of Bisociation. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 11–32. Springer, Heidelberg (2012)
3. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: Proceedings of Data Integration in the Life Sciences, Third International Workshop, pp. 35–49 (2006)
4. Getoor, L., Diehl, C.P.: Link mining: A survey. SIGKDD Explorations 7, 3–12 (2005)
5. Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. Nucleic Acids Research 35, D26–D31 (2007)
6. Sayers, E.W., Barrett, T., Benson, D.A., Bolton, E., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Federhen, S., Feolo, M., Geer, L.Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D.J., Lu, Z., Madden, T.L., Madej, T., Maglott, D.R., Marchler-Bauer, A., Miller, V., Mizrachi, I., Ostell, J., Panchenko, A., Pruitt, K.D., Schuler, G.D., Sequeira, E., Sherry, S.T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T.A., Wagner, L., Wang, Y., Wilbur, W.J., Yaschenko, E., Ye, J.: Database resources of the National Center for Biotechnology Information. Nucleic Acids Research 38, 5–16 (2010)
7. The Uniprot Consortium: The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Research 38, D142–D148 (2010)
8. Hunter, S., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Das, U., Daugherty, L., Duquenne, L., Finn, R.D., Gough, J., Haft, D., Hulo, N., Kahn, D., Kelly, E., Laugraud, A., Letunic, I., Lonsdale, D., Lopez, R., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Mulder, N., Natale, D., Orengo, C., Quinn, A.F., Selengut, J.D., Sigrist, C.J.A., Thimma, M., Thomas, P.D., Valentin, F., Wilson, D., Wu, C.H., Yeats, C.: InterPro: the integrative protein signature database. Nucleic Acids Research 37, D211–D215 (2009)
9. Jensen, L.J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P., von Mering, C.: STRING 8—a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Research 37, D412–D416 (2009)

10. The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nature Genetics 25(1), 25–29 (2000)
11. Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M.: KEGG for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Research 38, D355–D360 (January 2010)
12. Gerhard, D.S., et al.: The status, quality, and expansion of the NIH full-length cDNA project: The Mammalian Gene Collection (MGC). Genome Research 14, 2121–2127 (2004), full list of authors http://dx.doi.org/10.1101/gr.2596504
13. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. Journal of the American Society for Information Science and Technology 58(7), 1019–1031 (2007)
14. Newman, M.E.J.: Clustering and preferential attachment in growing networks. Physical Review E 64(2), 025102 (2001)
15. Adamic, L.A., Adar, E.: Friends and neighbors on the Web. Social Networks 25(3), 211–230 (2003)
16. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 118–127 (2004)
17. Koren, Y., North, S.C., Volinsky, C.: Measuring and extracting proximity graphs in networks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–255 (2006)
18. Doyle, P.G., Snell, J.L.: Random walks and electric networks (January 2000), http://arxiv.org/abs/math.PR/0001057
19. Brandes, U., Fleischer, D.: Centrality Measures Based on Current Flow. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 533–544. Springer, Heidelberg (2005)
20. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (1953)
21. Chen, H., Zhang, F.: The expected hitting times for finite Markov chains. Linear Algebra and its Applications 428(11-12), 2730–2749 (2008)
22. Jeh, G., Widom, J.: SimRank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, pp. 538–543. ACM (July 2002)
23. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (1998)
24. Colbourn, C.J.: The Combinatorics of Network Reliability. Oxford University Press (1987)
25. Köhler, J., Baumbach, J., Taubert, J., Specht, M., Skusa, A., Rüegg, A., Rawlings, C., Verrier, P., Philippi, S.: Graph-based analysis and visualization of experimental results with ONDEX. Bioinformatics 22(11), 1383–1390 (2006)
26. Birkland, A., Yona, G.: BIOZON: a system for unification, management and analysis of heterogeneous biological data. BMC Bioinformatics 7(1), 70 (2006)
27. Lee, T., Pouliot, Y., Wagner, V., Gupta, P., Stringer-Calvert, D., Tenenbaum, J., Karp, P.: BioWarehouse: a bioinformatics database warehouse toolkit. BMC Bioinformatics 7(1), 170 (2006)