BioLog: A Browser Based Collaboration and Resource Navigation Assistant for BioMedical Researchers

P Singh[1] , R. Bhimavarapu[1], H Davulcu[1], C Baral[1,] S Kim[2],
H Liu[1], M Bittner[2], IV Ramakrishnan[3]

[1] Dept of CSE, Arizona State University, Tempe AZ 85287
[2] Translational Genomics Research Institute, Phoenix AZ 85004
[3] CS Dept, Stony Brook University,Stony Brook NY 11794
{prabhdeep.singh, ravi, hdavulcu, Chitta, skim, hliu}@asu.edu
mbittner@tgen.org, ram@cs.sunysb.edu

Abstract. We often realize that communicating with other colleagues who are studying similar topics helps to identify information relevant to our area of study, which otherwise may not have been found. We wish to accelerate acquisition of collective knowledge in a defined area by identifying specific spheres of inquiry. Such spheres correspond to groups of people who are experts in a field. In this paper we provide a systematic way to gain knowledge from their online search activity, and enable them to organize and share their search findings for further analysis. We have built a prototype system, BioLog, to help biomedical researchers share this implicit knowledge among their peers and store their access patterns into a central system for reuse. BioLog has been deployed it in two labs within TGen as a pilot study. The data has been gathered and analyzed by preliminary text-mining and collaborative filtering methods.

1 Introduction

We often realize that communicating with other colleagues who are studying similar topics helps to identify information relevant to our area of study, which otherwise may not have been found. Hence, there have been many organizational efforts and a variety of tools produced to support sharing of knowledge, as well as data, within communities of shared research areas. The collective knowledge of sets of experts is different from the massive, general, text archives of information that we typically rely on since it is limited to a particular realm of findings. It is further different in that it reflects the experts' current models of what that field suggests and it is dynamic, and constantly changing as a result of researchers search activity. While data sharing among experts is improving constantly, model sharing has not improved. We wish to accelerate acquisition of collective knowledge in well defined areas by identifying specific spheres of inquiry and corresponding groups of people. We also provide a systematic way to gain knowledge from their online search activity, and enable them to organize and share their findings for further analysis

One place where experts' models are evident for further analysis and inferencing is their interaction logs with archived information sources. For example, PubMed [1] is

a well-known repository of biological literature and serves as an invaluable biological repository. It is frequently used as a first stage tool in creating and refining new hypotheses. An expert's prior understanding of the biological relationships and their emerging models will be implicit in their search patterns of PubMed and other such biomedical resources.

Biologists go to PubMed when they have a model with some supporting evidence but want to seek further support. It is also sought when they have a incomplete model with some missing elements or a fragmented model with missing relationships. They type in keywords and PubMed retrieves a list of keyword matching abstracts. Researchers scan through the list and identify a subset of abstracts that might be relevant to their model -- most likely based on the titles and the authors of articles. Once they identify the subset of articles, they follow-up on those articles and read the corresponding abstracts. Sometimes, they home in on their by iteratively narrowing down their keyword searches. However, they find it more informative to talk to their expert colleagues, who are studying similar subjects, to obtain recommendations and leads about other relevant articles that might contain missing links, as illustrated in Figure 1.

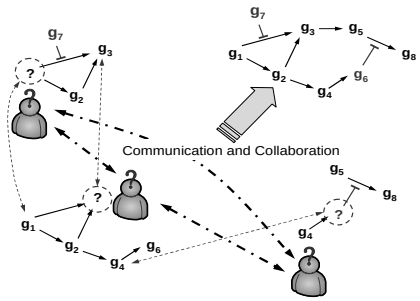


Figure 1: Communication and collaboration among biologists to combine knowledge of missing links.

One problem is that researchers often do not know whom to talk to. It could be someone in their lab or someone at another institute. A precursor to collaboration is to first find whom to work with or ask for help.

In most cases a biologist has some ‘handles’ (such as a set of nucleotide sequences or gene names) and he or she searches the repositories using those handles. For example, a biologist trying to figure out (parts of) a pathway that explains a particular phenomena may start with a list of gene and protein names as handles. Starting with one of those names, when one searches a repository like PubMed, it is likely that a large number of matches will be found. For example, the search term `g-protein' leads to 51,286 matches in PubMed. The researcher is then faced with the problem of narrowing down the articles that are relevant to his topic of investigation by adding additional keywords or trying alternative keywords. The time it takes to find the right matches plays a huge role in the overall timely success of the quest. A biomedical researcher would benefit tremendously if the various resources would rank the links in a way that matches her own priority. The situation here is closer to recommender systems such as the ones used in Netflix.com or Amazon.com where the system recommends movies and books respectively based on the users’ past interaction with the system, the users’ feedback (in terms of ratings in case of Netflix.com) and the global

knowledge extracted from the web log of all the users as well as the corresponding web content.

We have built a helper application, named BioLog, to archive scientists' access pattern of PubMed of NIH/NCBI as well as the client software that allows users to browse through group specific archives. The system logs the user identity, search keywords used, list of matching articles, set of followed articles, and the amount of time spent on each abstract. We also extract list of gene names using a state-of-the art gene/protein extractor, the Abner [17] system, from each abstract. We developed preliminary recommender algorithms based on gene-to-gene, abstract-to-abstract and user-to-user relevance networks by using a combination of collaborative filtering and content-based filtering techniques. BioLog system automatically recommends alternative lists of genes, articles and other researchers upon each keyword search.

In this paper we propose a recommendation algorithm based upon a clustering technique. Clustering is a technique to group items or data points that are similar in a given context. It has been widely used for many quantitative studies, including gene expression data analysis [9,10]. This is a natural choice of approach to find relevant or similar set of articles or genes given co-observations of genes and articles. A similar set of articles may represent a specific research subject, and a similar set of genes may indicate members of a regulatory network. However, in the context of high dimensional datasets such as those relating PubMed articles, genes, and users, where the datasets are wide and sparse, with many irrelevant dimensions, it is difficult to find relationships that exist in subspace of the dataset. *Subspace clustering* [11] is a form of unsupervised machine learning that seeks to uncover groups of objects that are related in terms of only a subset of the attributes (dimensions) in the dataset. In our effort to identify similar articles or genes, when the number of genes runs over tens of thousands, the number of users in tens of thousands and the number of articles in millions, but the number of users in a group who access articles being relatively rather small, we demonstrate that subspace clustering is useful and effective.

The rest of the paper is structured as follows. Section 2 presents the related work. Section 3 is the system flow. Section 4 is the system design. Section 5 presents relevance networks. Section 6 presents the BioLog's recommendation algorithm. Finally, Section 7 presents our preliminary pilot studies.

2. RELATED WORK

Collaborative filtering (or recommender systems) predicts products or topics a new user might like by using a database about other users past preferences. These systems are popular for their use on e-commerce web sites, where the systems use input about a customer's interests to generate a list of recommended items.

In Memory Based Algorithms [2] the task of collaborative filtering is to predict the votes/interests of the active user from a database of user votes from a sample or population of other users. The strategies mentioned in the memory based algorithms can be used in our current problem of recommending abstracts and users. The user database,

which is the log of browsing history in our case, contains information of the various abstracts accessed by the users in the system. We can construct a user-abstract preference/access table, which is analogous to the user-item information mentioned earlier. Based on this information, we could compute the similarity between pairs of users. Based on the similarity, other un-accessed abstracts could be recommended. Using either of the similarity based metrics, similar users can be recommended too. The user-abstract table/matrix constructed from the log would be very sparse since each user would have accessed an insignificant percentage of the total number of abstracts (from PubMed). The Pearson's correlation based or the vector based similarity [3] would not yield good measures if there are very few abstracts in common between two users. Another major pitfall of this approach is in regard with its scalability. Recommendations at runtime for the active user would require the system to scan over the complete database to compute the similarity metrics between the active user and the other set of users and then uses the weights over the common set of abstracts for the selected users.

Probabilistic Cluster Models [4] is a model based method, in which the learning phase can be done offline. Quick recommendations can be given in real time, thereby making the recommendation system scalable. A crucial pitfall in this approach is the Bayesian assumption that the conditional probabilities of the variables given the class are independent. This may well not be the case in our domain. The probabilities of the occurrence of genes given the class, in a given cluster might not be independent with respect to each other. In fact, genes identified in a cluster might be strongly correlated. On the other hand, evaluation results given by the authors for this approach do not seem to be impressive. Other approaches based on correlation outperform this model on most of the datasets.

Clustering is a technique to group items or data points that are similar in a given context. It has been widely used for many quantitative studies, including gene expression data analysis [9,10]. This is a natural choice of approach to find relevant or similar set of abstracts or genes given co-observations of genes and abstracts. A similar set of abstracts may represent a specific research subject, and a similar set of genes may indicate members of a regulatory network.

As datasets become larger and more complex, clustering performance often degrades due to the curse of dimensionality [12, 13]. In high dimensional data, clusters often exist in subspaces [14], and many of the dimensions are often irrelevant. These irrelevant dimensions confuse clustering algorithms by hiding clusters in noisy data. In very high dimensions it is common for all of the instances in a dataset to be nearly equidistant from each other, completely masking the clusters. Feature transformation and feature selection techniques have been used to address the difficulties in clustering high dimensional datasets [11]. However, neither of these techniques is suitable for finding subspace clusters. Feature transformation such as Principle Components Analysis (PCA) attempt to summarize the data by creating new attributes which are combinations of the original attributes in the dataset. Since relative distances are preserved, the effects of the irrelevant dimension remain. Also, the new attributes can

be very difficult to interpret. Feature selection techniques attempt to select the most relevant attributes over the whole dataset. While successful at removing noisy attributes [15], feature selection does not allow us to discover clusters that exist in different subspaces. *Subspace clustering* is a form of unsupervised machine learning approach that we utilize in this paper to uncover groups of objects that are related in terms of only a subset of the attributes (dimensions) in the dataset. In our effort to identify similar abstracts or genes, when the number of genes runs over tens of thousands, the number of users in tens of thousands and the number of articles in millions, but the number of users in a group who access articles being relatively rather small, subspace clustering is useful and effective.

Instead of matching the active user to similar customers, item-to-item based approach matches each of the user's purchased and rated items to similar items, and then combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together. Unlike the traditional collaborative filtering techniques, this algorithm's online computation scales independently of the number of customers and number of items in the product catalog. The above mentioned algorithm can be modified, replacing items with abstracts. This way, we can build up a *similar-abstracts* table by finding abstracts that users tend to look together. As more users tend to access a set of related articles, their pair wise similarity scores go up. Using the similar-abstracts table, related articles can be recommended. As mentioned earlier, this method's online computation scales independently to the number of abstracts and the set the genes, since we would be computing the similarity tables offline. Unlike traditional collaborative filtering techniques, the algorithm also reportedly performs well with limited user data, producing high-quality user data, producing high-quality recommendations. The offline computation of the similarity tables is extremely time intensive, with $O(N^2M)$ as worst case, where N is the number of abstracts/genes and M is the number of users/abstracts respectively for the two above mentioned adaptations to the domain.

3. SYSTEM FLOW

As shown in Figure 5, a biologist initially goes to PubMed types in a keyword search query and PubMed will fetch a list of articles matching the keyword. The biologist scans through the list and identifies a subset of articles that might be relevant to their inquiry, most likely based on the titles and the authors of articles. Once they find the articles of high relevance, they will click on one of the articles and read the abstract to make sure if it is really useful to what they are looking for. Biolog tracks these Web pages in a database log and archives them in a central cache repository with all relevant meta information. Currently we are using a MySQL backend but the module has been built to be database independent. The cached documents are also indexed using a high performance text search engine in order to support keyword searching in the cached documents. Next, gene-to-gene and abstract-to-abstract relevance networks are computed and the recommendation system uses these models.

4. BIOLOG SYSTEM DESIGN

We have built a helper application for Internet Explorer® (IE) to archive scientist's accessing pattern of vast archive of biomedical literatures at PubMed of NIH/NCBI. The archival process consists of a logger, which is responsible for capturing web pages during browsing based on domains which are to be tracked. The capturing of data is in terms of logging Meta information in the database as well as caching of web pages in a central repository.

In Figure 2 below the logger uses browser helper objects (BHO) [5] to store html pages in the file system cache as well as all relevant meta information such as machine name, URL, time-stamp etc to the database.

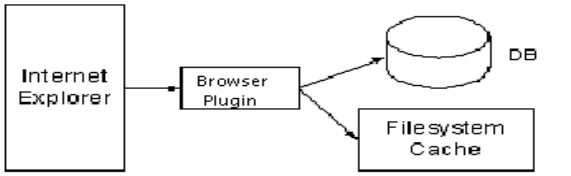


Figure 2: Logger Architecture

Browser Helper Objects are components — specifically, in-process Component Object Model (COM) components — that Internet Explorer will load each time it starts up. Such objects run in the same memory context as the browser and can perform any action on the available windows and modules. Further, a new instance of the BHO is created each time a new browser window is created. In its simplest form, a BHO is a COM in-process server registered under a certain registry's key. Upon start up, Internet Explorer looks up that key and loads all the objects whose CLSID is stored there. Logging of dynamic data on the Web has been a problem. By dynamic data we mean the data input by the user at run time during filling of form elements. We planted our logging module into the IE browser and this architecture can be imported to any other browser with plug-in support. The problem of trapping the dynamic data can be tackled during the pre-navigation step, which is, as soon as the dynamic data is submitted and before the response page is loaded. During navigation, we trap the BeforeNavigation event and at that precise moment we capture a snapshot of the current dynamic page DOM and inspect its form elements for dynamic attribute-value pairs.

The logger, a plug-in program to IE, is activated only when scientists go to PubMed and type in keywords to search through the archive. Then, it records the keywords used, the set of articles displayed, and the set of articles that scientists try to read by clicking on the link to its abstract. It also records the time spent on an abstract as well as other relevant information described above. All the archived information is stored in MySQL database for easy access across many clients.

5. ENTITY-TO-ENTITY RELEVANCE NETWORKS

First, gene-abstract occurrence matrix (GA matrix, **GA**) is constructed for the entries in the log. **GA** matrix is a matrix where its element, ga_{ij}, is 1 if a gene *i* appears in an abstract *j*. Otherwise, it is zero. Similarly, we build user-abstract matrix (UA matrix, **UA**). ua_{ij} is 1 if user *i* read an abstract *j*. Otherwise, it is zero. Based on these matrices, we find gene-gene, abstract-abstract and user-user relevance networks as follows.

5.1 Gene-Gene Relevance Networks

Once GA matrix is constructed, we then compute gene-gene relevance matrix, GG matrix (**GG**), by multiplying **GA** by the transpose of **GA**, and normalizing it by dividing each row of GG by the number of abstracts. gg_{ij} is 0 if genes i and j never appear in an abstract at the same time. gg_{ij} is 1 if genes i and j appear in all of the abstracts looked at. The value obtained will be in the normalized range of [0,1] , 1 indicating that the two genes co-occur all the time and 0 indicating that the two genes never co-occur together. The idea is to assume if two genes are relevant either positively or negatively, they would tend to appear often in same abstract. Often this assumption may not be true; it is not rare to find an abstract to claim two genes are irrelevant in particular context. However, we found that, even with this crude assumption, some of the genes with high relevance could be identified.

5.2 Abstract-Abstract Relevance Networks

Abstract-abstract relevance, AA_G matrix ($\mathbf{AA}_G$), can be built, by multiplying the transpose of **GA** by **GA**, and normalizing it by dividing each row of $\mathbf{AA}_G$ by the number of genes appeared in either abstracts. aa_{ij} is 0 if abstracts *i* and *j* do not have any gene in common. aa_{ij} is 1 if any gene appeared in one abstract appears in the other. This $\mathbf{AA}_G$ matrix corresponds to content-based relevance since the more genes are shared between these two abstracts, the more relevant they are to each other. Another way to define an abstract-abstract relevance matrix is by using the user-abstract access matrix, **UA**. The access matrix **UA** can be multiplied to its transpose to construct another access-based relevance matrix, $\mathbf{AA}_U$. In this preliminary work, we relied on a definition of the abstract-abstract relevance, **AA,** by using a weighted sum of these two different similarity measures $\mathbf{AA}_G$ and $\mathbf{AA}_U$. Similarly User-User relevance matrix can be defined as a weighted sum of commonly accessed gene and abstract based relevance matrices.

6. BIOLOG RECOMMENDATION SYSTEM

Our hybrid recommendation system utilizes a combination of the above relevance networks and a collaborative filtering based approach.

Content based clustering (of genes and abstracts): The log gives us information about the abstracts accessed so far by various users. One can extract the list of genes/proteins from these abstracts. The intention here is to find co-occurring genes based on the abstracts they are present in. Similar logic can be used in finding co-occurring abstracts based on their composition of genes in each abstract.

Algorithm (in finding co-occurring genes)

a. Build the gene-gene relevance network

b. Normalize the obtained GxG matrix using the following formula.

$S_{uv} = \frac{C_{uv}}{C_{uv} + C_{vv} - C_{uv}}$ In this equation, Cxy denotes the un-normalized entries of GxG. Each cell in the matrix is normalized according to the equation shown above. The value obtained will be in the range of [0,1] , 1 indicating that the two genes co-occur all the time and 0 indicating that the two genes never co-occur together.

c. Perform Hierarchical Agglomerative Clustering (HAC) [16] to reach a fixed number of clusters or some termination condition. Genes that co-occur together fall into one cluster.

This way we can identify similar genes. A similar approach can be done on clustering abstracts. Here we build up a normalized AxA matrix from the AxG matrix. Co-occurring abstracts (based on the composition of their genes) fall into one cluster. Therefore, we could find similar abstracts. In fact, this method was used in the preliminary analysis of archives from our pilot studies.

Collaborative Filtering based approach: As contrast to content-based filtering, we can also define the similarity between two abstracts/genes in terms of number of users who have accessed both the abstracts/genes. To recommend similar abstracts, from the log, we build the User by Abstract (UxA) matrix, and compute the AxA normalized co-relational matrix from the UxA matrix. Given any abstract, we could rank the 'k' most similar abstracts based on the correlation similarity measure. Alternatively, User by Abstract (UxA) matrix can be used to find the closest neighbours (similar users), whose preferences can be used to predict the interest/vote on other abstracts. Pearson's correlation co-efficient can be used to find the neighbours, but this strategy would fail if the UxA matrix is sparse.

Hybrid approach - combining content and collaborative filtering based approach: This approach combines a collaborative filtering and a content based mining in finding similar abstracts. Two abstracts are similar:

i) if they have a good set of genes common in them (Content based perspective) and

ii) if many users view both the abstracts (Collaborative Filtering based perspective). In this way, we consider both the content and the user browsing pattern in associating similarity between abstracts. An approach, using weights to combine two different similarity matrices is detailed Figure 3.

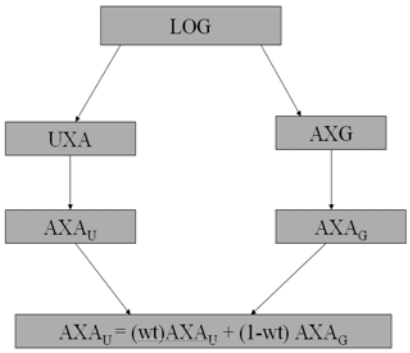


Fig 3. Similarity matrix computation in the hybrid approach using weights.

7. SUBSPACE CLUSTERING FOR RECOMMENDATION WITH SPARSE HIGH DIMENSIONAL DATA

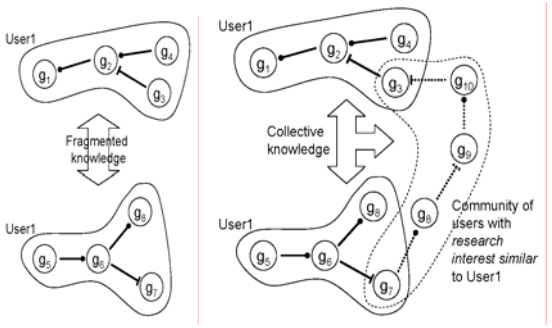
Finding subspace clusters in the gene-abstract occurrence matrix can reveal relationships between genes and abstracts allowing us to recommend relevant subsets of articles for each query. In search of abstracts with shared genes, we can improve efficiency and accuracy by focusing on clusters of abstracts that share relevant genes. On one hand, the number of genes can be as many as fifty thousand and the number of abstracts can be millions; on the other hand, each abstract usually has a small number of genes (from 1 to 6 genes). That is, although the Abstract-Gene matrix has an extremely high dimensionality, clusters of abstracts can only exist in low dimensional subspaces. By finding these low dimensional subspaces, we can achieve the following: (1) given a new set of genes, subspaces defined by associated genes can be quickly identified; (2) clusters of abstracts in these subspaces can be efficiently located; and (3) similar abstracts can then be ranked and recommended as the number of abstracts in the subspaces is significantly smaller than the total number of available abstracts for search.

Given the Abstract-Gene matrix, abstracts are compared using a similarity measure that considers only the positive (non-zero) values in the matrix. This comparison is done first in low dimensional space, revealing those genes that occur frequently together in abstracts. Searches in the low dimensional space allow us to eliminate genes or gene combinations that are not frequent which helps to reduce the search space. The subspaces represent groups of genes that occur often together in abstracts. The clusters represent abstracts that mention many of the same genes. When analyzed, the smaller data set yields 10 clusters in 2-dimension (using only two words as features), 5 clusters in 3-dimension and 1 cluster in 4-dimension. The size of clusters in 2-D ranges from two to 5 abstracts and the cluster found in 4-dimension is composed of 3 abstracts. For the larger dataset, the cluster with the largest dimension was in 12-dimensions with two abstracts belonging in the cluster. There were 4 clusters in 11-D, 9 clusters in 10-D with at least 2 abstracts. In general, more clusters were found in lower dimensional subspaces.

Adding the Abstract-User matrix further improves the utility of the tools, as illustrated in Fig. 5. As hypothesized, dynamic communication with other colleagues studying similar subjects would help locate relevant information for biologists. Let us consider a user (U1) has accessed many abstracts and accumulated knowledge during his/her previous and current querries. The knowledge acquired through a previous query might often be relevant to the current search based on information that has not been realized by the user. If the proposed approach can identify this information by pulling together and analyzing knowledge (abstracts) utilized by other scientists with a similar research interest, such guidance will speed up adopting new knowledge, such as new pathways.

Also, if two different biologists (U1 & U2) may not have a link (common research interest; same gene or transcription factor) to directly connect them even if they might indeed benefit from talking to each other due to some indirect links, the tool might be

able to locate such links by analyzing various links embedded in knowledge access patterns, hence, enable their connection. Synergism resulting from such collaboration would yield much faster knowledge discovery. An illustration similar to Fig. 5, replacing one of User1s with User2 can visualize our approach.



(a) (b)

Figure 4. Subspace clustering finds closely related genes based on user's access patterns of articles. Each cluster indicates that the genes grouped together appeared many times in the set of articles accessed by the user. The set of articles in which the clustered genes appear together can be pulled from each cluster as knowledge support. (a) The knowledge of User1 is fragmented due to the lack of relevant knowledge (links) in individual access patterns. (b) Collective knowledge helps User1 realize two pathways are connected.

The Figure 4 above exhibits how subspace clustering can be applied effectively to discover implicit knowledge for a researcher. Figure 4 (a) shows that two subspaces exist for User1 alone where a subspace represents a set of genes occurring together. Here, User1 thinks that genes 1,2,3,4 are linked to each other and genes 5,6,7,8 are linked with each other independently with no connection between the subspaces. Figure 4 (b) shows that there exists a subspace generated from all users where the subspace suggests that there is a link between gene3 and gene7. Notice that User1 did not realize or was not aware of the connection between the two genes but by using the knowledge from the community of users, User1 can be given such knowledge. This kind of knowledge could be very useful for User1 because if he was working independently, it might have taken him a longer period of time or in the worst case the user might not have been aware of this knowledge at all. Preliminary experimental results of subspace clustering on large Web logs indicate that such knowledge can be effectively discovered from the data.

8. PILOT STUDY IN TWO TGEN LABS

Two biology labs at TGen [7] were selected to perform pilot study with BioLog. Both labs are part of the Neurogenomics program at TGen. We set up two central servers to archive their access patterns on PubMed separately.

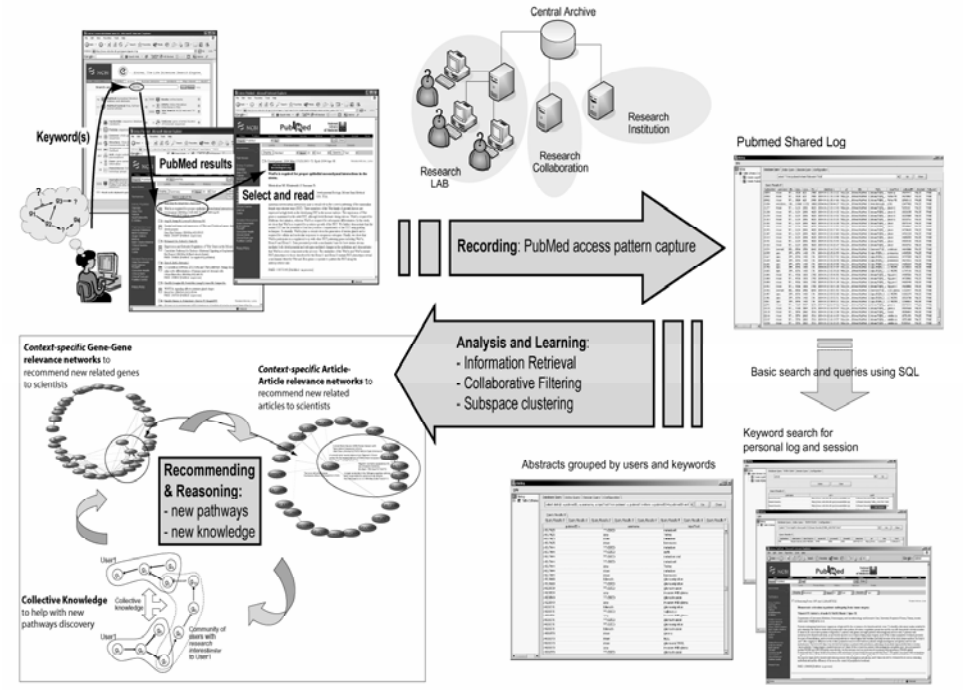


Figure 5. BioLog: PubMed Recording, Reasoning and Recommending (R3) Navigation Assistant Pilot Study

Since both study brain-related diseases, they could share some commonality. However, they are two different labs studying different specific diseases; therefore, they would differ significantly in accessing literatures in PubMed. We would like to see if the archives show such difference as well as similarity. During one study 25 abstracts accessed, while the other archive returned 253 abstracts accessed.

The gene relevance network from smaller archive is shown in Figure 5. The networks are visualized to emphasize the co-occurrence of two genes; if two genes co-occur more often than others, they were put close to each other in the visualizations. Also, the thickness of edge represents the normalized frequency of co-occurrence of the pair; thicker the edge, more often they co-occur. For example, in Figure 5, genes **smn, sma, smn1, smn2,** and **kinase** are very close to each other, indicating they appear in the same abstract often. We also found it interesting that these genes were found in the second network which is constructed from the archive from the other lab. Therefore, this shows that these two labs sometimes study similar genes. This is critical because it might imply that two lab studying similar subjects, brain-related disease in this case, share the genes of their interests, and we might be able to use this clue to find out other group or people that could study some of the subject common to one's research. However, since they do have many other genes that are not in the other's. This could indicate either that one is studying some other subjects that the

other does not (most likely), or that each one is taking a different route to find answers. In the latter case, one might be interested in what other genes the other group is after.

Figure 5 visualizes abstract-abstract relevance network. Interestingly, we have identified a distinct cluster of abstracts in the relevance network from the smaller archive shown in Figure 5, it was related to the cluster of genes identified in the previous section; all describing smn, sma, smn1, or smn2. Such clusters form the basis of BioLog recommendations.

9. FUTURE WORK

The components built as a part of the Biolog system (Figure 5) can also be suitable for domains other than Biology, where a group of people is searching and interacting with a set of entities. Once the recommendation algorithm is embedded into a browser component we plan to perform detailed user evaluations in order to determine the usefulness and validity of BioLog's recommendations as compared to other existing recommender algorithms.

References

[1] Pubmed by NCBI and National Library of Medicine http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

[2] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, New York, ACM, 1994. (175-186).

[3] G. Salton and M. J. McGill. Introduction to Modern Retrieval. McGraw-Hill Book Company, 1983.

[4] Breese, J., Heckerman, D., and Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of the 14 th Conference on Uncertainty in Artificial Intelligence, 1998 (43-52).

[5] AnHai Doan, Robert McCann. Building Data Integration Systems: A Mass Collaboration Approach. IJCAI 03.

[6] Browser Helper Objects: The Browser the Way You Want It. Dino Esposito. Microsoft Corporation. http://msdn.microsoft.com/library/en-us/dnwebgen /html/bho.asp

[7] Legendre, P. & Legendre, L. (1998). Numerical Eoclogy. Second English Edition. Ed. Elsevier.

[8] Translational Genomics Research Institute (TGen) http://www.tgen.org/

[9] Bar-Joseph, Z., et al., K-ary clustering with optimal leaf ordering for gene expression data. Bioinformatics, 2003. **19**(9): p. 1070-8.

[10] Getz, G., et al., Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. Bioinformatics, 2003. **19**(9): p. 1079-89.

[11] Parsons, L., Haque, E., and Liu, H. Subspace clustering for high dimensional data: a review. SIGKDD Explor. Newsl., 1004. Vol. 6, No. 1: p. 90-105.

[12] Devroye, L., Gyorfi, L. and Lugosi, G. A Probabilistic Theory of Pattern Recognition. 1996, Springer-Verlag: New York.
[13] Parsons, L., Haque, E., and Liu, H. Evaluating Subspace Clustering Algorithms. in Workshop on Clustering High Dimensional Data and its Applications, SIAM International Conference on Data Mining (SDM) 2004. 2004.
[14] Friedman, J.H. and Meulman, J.J. Clustering Objects on Subsets of Attributes. Journal of the Royal Statistical Society Series B, Volume 66, Issue 4, page 815, November 2004.
[15] Liu, H. and Yu, L. Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Transaction on Knowledge and Data Engineering, forthcoming.
[16] A.K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.
[17] Burr Settles. Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), Switzerland. 2004.