

# Fastest Searching Mechanism of Bio-Medical Database

<sup>1</sup>K. Subba Rao, <sup>2</sup>M. Raja Babu

<sup>1,2</sup>Dept. of CSE, Aditya Engineering College, Surampalem, Kakinada, AP, India

## Abstract

Biologists, chemists, medical and health scientists are used to searching their domain literature – such as PubMed – using a keyword search interface. Currently, in an exploratory scenario where the user tries to find citations relevant to her line of research and hence not known a priori, she submits an initially broad keyword-based query that typically returns a large number of results.

We demonstrate the BioNav system, a novel search interface for biomedical databases, such as PubMed. BioNav enables users to navigate large number of query results by categorizing them using MeSH; a comprehensive concept hierarchy used by PubMed. Once the query results are organized into a navigation tree, BioNav reveals only a small subset of the concept nodes at each step, selected such that the expected user navigation cost is minimized. In contrast, previous works expand the hierarchy in a predefined static manner, without navigation cost modeling. BioNav is available at <http://db.cse.buffalo.edu/bionav>

## Keywords

Interactive Data Exploration And Discovery, Search Process, Graphical User Interfaces, Interaction Styles.

## 1. Introduction

We focus on one of the most important and largest collections of biomedical data freely available on the Internet, that of the National Center for Biotechnology Information (NCBI). The NCBI maintains over 30 public databases containing biomedical information of various types, such as published medical documents (PubMed), gene listings (Entrez Gene), protein listings (Entrez Protein), and DNA sequence information (Entrez Sequence). It also stores and manages pairwise associations between records in the databases according to the various types of content.

For example, a particular document *d* listed in PubMed might be associated with all genes *G* from Entrez Gene that are mentioned in *d*. *d* may also have associations with other PubMed documents that cite *d* as a reference, as well as associations to the PubMed documents that *d* itself cites. Furthermore, each gene *g* 2 *G* could have associations with the proteins for which *g* codes, or the DNA sequences in which *g*'s code appears. Usually, the various types of records in these databases also have many attributes associated with them. For example, PubMed documents might be annotated with the date of publication, authors, and general topics, while gene records could be annotated with the relevant species, location on chromosome, or function. This rich space of record attributes is key in aiding understanding of the data.

Given the huge amount of data at NCBI, and the large number of databases, myriad variations of these associations are possible. To organize this data in a way useful for knowledge exploration, note that NCBI's multiple databases can be abstracted as a massive entity-relation graph. In this graph, nodes correspond to individual knowledge points or database records, such as documents, genes, proteins, and other object types. Associations between database objects can then be modeled as directed or undirected links in the graph, connecting related nodes. The entity-graph model has already been applied to various document collections, including some in the biomedical domain, and much research has dealt with

providing a broad overview of research publications and trends by visualizing the graph, typically using a force-directed node layout scheme [7], or other schemes such as circular [2], matrix-based [6], hierarchical [1], or layered [4] node layouts. These types of top-down visualizations simplify the identification of concepts like research fronts [2].

However, our motivation lies not in discovering overall trends, but rather in accomplishing the everyday technical tasks of knowledge exploration and discovery undertaken by biomedical scientists and researchers. Scientists researching a particular gene, protein, or topic want to find specific and relevant information that will aid in their research. As a result, when using NCBI's databases, they begin with a specific query or set of queries, and explore outward from the initial query result. They might also cross-reference records from multiple databases. Our visualization tools are designed to aid this query-specific exploration.

The last decade has been marked by unprecedented growth in both the production of biomedical data and the amount of published literature discussing it. The MEDLINE database, on which the PubMed search engine operates, contains over 18 million citations, and the database is growing at the rate of 500,000 new citations each year [7]. Keyword search queries on these databases return a large results set from which only a small portion is relevant for the user. Many solutions have been proposed to address this problem – commonly referred to as information-overload [2,3]. These approaches can be broadly classified into two classes: ranking and categorization, which can also be combined.

BioNav belongs primarily to the categorization class, which is ideal for this domain given the rich concept hierarchies available for biomedical data, such as MeSH [5]. Each citation in MEDLINE is associated with several MeSH concepts in two ways: (i) by being explicitly annotated with them, and (ii) by mentioning them in their text. Since these associations are provided by PubMed, a relatively straightforward interface to navigate the query result would first attach the citations to the corresponding MeSH concept nodes and then let the user navigate the concept hierarchy.

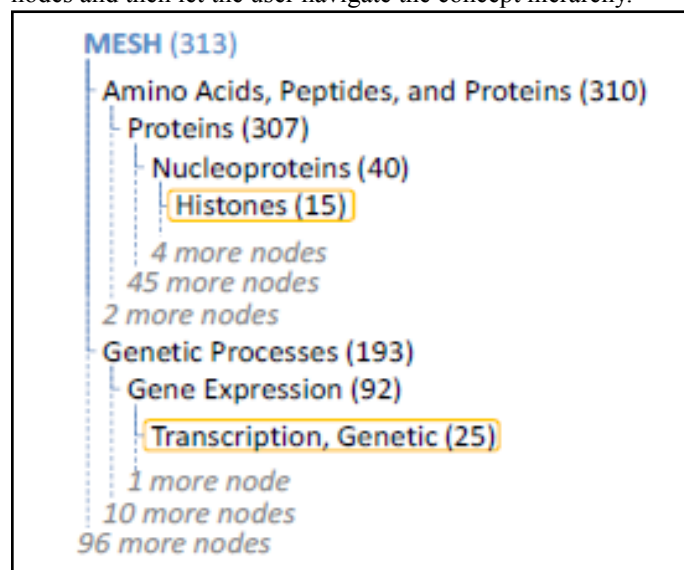


Fig. 1: Static Navigation on the Mesh Concept Hierarchy

Fig. 1, displays a snapshot of such an interface where shown next to each node label is the count of distinct citations in the subtree rooted at that node. For this example, we assume that the user queries MEDLINE for the nucleoprotein “prothymosin” and his personal interests are reflected in the two indicated concepts, corresponding to two independent lines of research related to prothymosin. A typical navigation starts with revealing the children of the root ranked by their citation count, and is continued with expanding one or more of them, revealing their ranked children and so on. Further, the user may click on a concept and inspect the attached citations. A similar interface and navigation method is used by GoPubMed [6] and e-commerce sites, such as Amazon and eBay.

The above static navigation method—same for every query result—is problematic when the MeSH hierarchy (or one with similar properties) is used for categorization for the following reasons. The massive size of the MeSH hierarchy (with 48,441 concept nodes) makes it challenging for the users to effectively navigate to the desired concepts and browse the associated citations. A substantial number of duplicate citations are introduced in the navigation tree of fig. 1, since each one of the 313 distinct citations is associated with several concepts. Specifically, the total count of citations in fig. 1, is 40,195.

BioNav, first proposed in [1], introduces a dynamic navigation method that depends on the particular query result at hand. The query results are attached to the corresponding MeSH concept nodes as in Figure 1, but then the navigation proceeds differently. The key action on the interface is the expansion of a node that selectively reveals a ranked list of descendant (not necessarily children) concepts, instead of simply showing all its children.



Fig. 2: BioNav Interface After Querying for "Prothymosin" and Its Associated Subtree Information Window

BioNav Interface. Fig. 2 shows the state of the BioNav interface after querying for “prothymosin”. The root of the MeSH tree can be seen on the left pane. The right pane shows the results under the current node of the navigation tree of the left pane. The user can also view more information about a subtree rooted at a given concept node by clicking on the icons that appear next to each concept label. The table of the pop-up window in Figure 2 shows various characteristics of the current subtree, including the fact that the 313 citations in the query result are spread over 3940 concept nodes.

BioNav Navigation. Figure 3a shows the initial expansion of the root node where only 8 (highlighted) descendants are revealed compared to 98 children shown in Figure 1. The concepts are ranked by their relevance to the user query and the number of them revealed depends on the characteristics of the query results. Next, assuming the user is interested in the “Amino Acids...” node and judging that the 310 attached citations is still a big number, she expands it by clicking on the “>>>” hyperlink next to it in

Figure 3b. The user inspects the 6 concepts revealed and decides that she is not interested in any of them. Hence, she expands the “Amino Acids...” node one more time in Fig. 3(c), revealing 4 additional concepts. Note that “Nucleoproteins” is an example of a descendant node being revealed, since its parent node “Proteins” (shown in fig. 1) is not revealed in Fig. 3(c). In Fig. 3(d), the user expands the “Nucleoproteins” node and reveals “Histones”, one of the two key concepts for the query. Note that to reach “Histones” using the BioNav navigation method only 23 concepts are revealed, after 4 node expansions, compared to 152 concepts, also after 4 expansions, with the static navigation method of fig. 1.



Fig. 4: BioNav Navigation

## II. Motivation

- Exploratory queries are increasingly becoming a common phenomenon in life sciences. e.g., search for citations on a given keyword on PubMed.
- These queries return too-many results, but only a small fraction is relevant. The user ends up examining all or most of the result tuples to find the interesting ones.
- Can happen when the user is unsure about what is relevant. e.g., user is looking for articles on a broad topic: ‘cancer’ . . . , query returns over 2 million citations on PubMed.

This phenomenon is commonly referred to as information overloded

### III. Existing System

Existing search operation Information overload is a major problem when searching Biomedical databases such as PubMed, where typically a large number of citations are returned, of which only a small subset is relevant to the user.

### IV. Proposed System

The MeSH concept hierarchy is a labeled tree [5], where the label of a child concept node is more specific than the one of its parent. Once the user issues a keyword query, PubMed-BioNav uses the Entrez Programming Utilities (eUtils) [4] –returns a list of citations, each associated with several MeSH concepts. BioNav constructs a navigation tree by attaching to each concept node of the MeSH concept hierarchy a list of its associated citations and removing all nodes with no citations, while preserving the ancestor-descendent relationship. The navigation tree  $T(V, E, r)$  is the maximum embedding of an initial navigation tree  $TI(VI, EI, r)$  such that no node  $n \in V$  is labeled with an empty results list  $L(n)$ , excluding the root (in order to maintain the tree structure and avoid the creation of a forest).

We model a node expansion at a given navigation step as an EdgeCut in the navigation tree. In Figure 4, the dashed line illustrates the EdgeCut corresponding to the expansion of the node “Amino Acids...”. This expansion reveals the highlighted concepts of fig. 4, which include a subset of the highlighted concepts in fig. 3(c). The EdgeCut consists of the edges (“Proteins”, “Transcription Factors”) and (“Proteins”, “Nucleoproteins”). A valid EdgeCut of a tree  $T(V, E, r)$  is an EdgeCut  $C \subseteq E$  such that no two edges in  $C$  appear in a path from the root to a leaf node. We only consider valid EdgeCuts, because invalid EdgeCuts lead to unintuitive navigations.

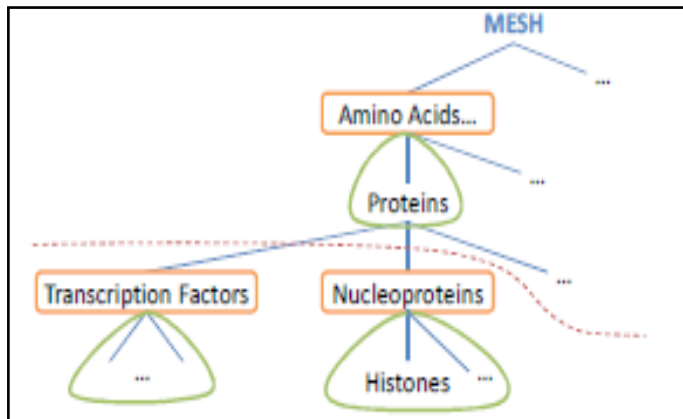


Fig. 3: Navigation Tree Edge Cut and Component Subtrees

Component Subtrees. An EdgeCut causes the creation of two types of component subtrees, a single upper and possibly multiple lower. Fig. 4 shows two lower component subtrees, rooted at “Transcription Factors” and “Nucleoproteins”, and an upper component subtree comprising of the node being expanded “Amino Acids...” and all nodes not in any of the lower component subtrees.

### A. Navigation and Cost Model

BioNav initiates a navigation by constructing the initial results tree and displaying its root to the user. Subsequently, the user navigates the tree by performing one of the following actions on

a given component subtree  $I(n)$  rooted at concept node  $n$ :

1. **EXPAND  $I(n)$** : The user clicks on the “>>>” hyperlink next to node  $n$  and causes an  $\text{EdgeCut}(I(n))$  operation to be performed on it, thus revealing a new set of concept nodes from the set  $I(n)$ .
2. **SHOWRESULTS  $I(n)$** : By performing this action, the user sees the results list  $L(I(n))$  of citations attached to the component subtree  $I(n)$ .
3. **IGNORE  $I(n)$** : The user examines the label of concept node  $n$ , ignores it as unimportant and moves on to the next revealed concept.

#### Algorithm 1 Explore C

```

1: if  $n$  is not a leaf node, then choose one of the following then
2: SHOWRESULTS( $n$ )
3: IGNORE( $n$ )
4: S EXPAND( $n$ )
5: for each  $n_i \in S$  do
6: EXPLORE( $n_i$ )
7: end for
8: else
9: CHOOSE one of the following:
10: a) Examine all tuples in (C)
11: b) IGNORE C
12: end if

```

This navigation process continues until the user finds all the citations she is interested in. The cost of a navigation is computed as follows: We assign (i) cost of 1 to each newly revealed concept node that the user examines after an EXPAND action, (ii) a cost of  $B$  (determined empirically) to each EXPAND action the user executes, and (iii) cost of 1 to each citation displayed after a SHOWRESULTS action. BioNav estimates the navigation cost by taking in to account the probability that the user will execute an EXPLORE or SHOWRESULTS action at each step of the navigation. The EXPLORE probability is proportional to the number of unique results in the corresponding component subtree, whereas normalized entropy of the component subtree is used as the SHOWRESULTS probability.

The BioNav system architecture is shown in fig. 5 and consists of two parts. The off-line components populate the BioNav database with the MeSH concept hierarchy and the associations of the MEDLINE citations with MeSH concepts to decrease the online response time. The on-line components support BioNav’s web interface and the EXPAND/SHOWRESULTS user actions.

### Off-Line Pre-Processing

The BioNav database is first populated with the MeSH hierarchy. Next, the associations of MEDLINE citations and MeSH concepts are populated by issuing a query on PubMed for each concept  $c$ . For each citation  $t_i$  returned by the query, we add the association  $\langle c, t_i \rangle$  in our database.

### On-Line Operation

Upon receiving a keyword query from the user, BioNav executes the same query against the MEDLINE database and retrieves only the IDs (PubMed Identifiers) of the citations in the query result using the ESearch utility [4] and constructs the navigation tree by retrieving the MeSH concepts associated with each citation in the query result. Initially, the root of this navigation tree is shown to the user. Subsequently, when she requests an EXPAND action on the root, the Navigation Subsystem executes a heuristic algorithm to compute the best EdgeCut and the roots of the resulting component subtrees are visualized on the web-interface.

