# A User-centric Framework for Accessing Biological Sources and Tools[*]

Sarah Cohen-Boulakia[1], Susan Davidson[2] and Christine Froidevaux[1]

[1] LRI, CNRS UMR 8023, Université Paris-Sud,
Orsay, France
`cohen,chris@lri.fr`
[2] Department of Computer and Information Science
University of Pennsylvania, USA
`susan@cis.upenn.edu`

**Abstract.** Biologists face two problems in interpreting their experiments: the integration of their data with information from multiple heterogeneous sources and data analysis with bioinformatics tools. It is difficult for scientists to choose between the numerous sources and tools without assistance. Following a thorough analysis of scientists' needs during the querying process, we found that biologists express *preferences* concerning the sources to be queried and the tools to be used. Interviews also showed that the querying process itself – the *strategy* followed – differs between scientists. In response to these findings, we have introduced a user-centric framework allowing to specify various querying processes. Then we have developed the BioGuide system which helps the scientists to choose suitable sources and tools, find complementary information in sources, and deal with divergent data. It is generic in that it can be adapted by each user to provide answers respecting his/her *preferences*, and obtained following his/her *strategies*.
**Availability:** http://www.lri.fr/∼cohen/bioguide/bioguide.html.

## 1 Introduction

Life sciences are continuously evolving so that the number and size of new sources providing specialized information in biological sciences have increased exponentially in the last few years,[3] as well as the number of tools required to carry out bioinformatics tasks. Scientists are therefore frequently faced with the problem of selecting sources and tools when interpreting their data. The diversity of sources and tools available makes it increasingly difficult to make this selection without assistance.

We firstly introduce a framework allowing to specify various querying processes. Our work was developed following a thorough study of scientists' needs during

---

[3] See the annual Nucleic Acids Research database issue (January).

querying and data management. After interviewing scientists working in various domains, we found that they expressed *preferences* concerning the sources queried and the tools used. Moreover, this study emphasized the fact that the process of querying itself – the *strategy* – varies from one scientist to another. We have then designed the BioGuide system, which provides scientists with support during the querying process. BioGuide assists the scientist with data searches within sources, providing information concerning the sequences of sources to be consulted and the tools to be used: the *paths* between sources to be followed.

We first describe the method used to assess scientists' requirements, and present the needs identified (section 2). We then describe the notion of strategy (section 3) and the way in which we propose to manage preferences (section 4). Section 5 introduces the formal framework and presents the general architecture of BioGuide, explaining how it provides support for the querying process. The biological significance of the results obtained will be presented in section 6. Section 7 compares our work to previous work and concludes the paper.

## 2 User Requirements

### 2.1 Process: Interviews and Questionnaire

We started with a thorough study of user requirements (cf. BioGuide site). We investigated the way in which scientists query sources and perform bioinformatics tasks (in the spirit of [18] and [6]), paying particular attention to determining why biologists query one source rather than another (*preferences*) and identifying the steps of their querying process (*strategies*).

A questionnaire was developed based on lists of user requirements in three kinds of documents: (i) survey articles [11] and reports of workshops on biological source querying (ii) studies on data quality [14], [4], [15] and (iii) studies on user guidance during the querying process, involving BioMediator [12], BioNavigation [9] and DSS [2]. The questionnaire comprised 28 questions and was constructed according to standard guidelines. As an illustration, four questions are provided:

- Choose a particular context from your own area of study and list some biological queries that you frequently make.
- If several sources yield answers for your query, do you access all of them or only few? If you query only a few, how do you proceed?
- In your mind, what is a "high-quality" source/tool?
- When you look for data related to two linked entities (e.g. a gene and the protein it encodes), how do you proceed (sources accessed, way of correlating information, etc.)?

After collecting responses to the questionnaire, we conducted *interviews* according to classical techniques. We sent questionnaires to 20 individuals, including both biologists and bioinformatics specialists. Their research interests fell into three main domains: *studies of diseases*, *functional* and *structural genomics*.

From the questionnaire, we identified 156 common queries. Some had almost identical structures (e.g. the search for genes involved in *breast* or in *bladder* cancer) and we grouped them together, giving a total of 119 distinct queries.

## 2.2 Transparent queries and Traceability

In most cases, neither the sources to access nor the tools to be used were specified by the biologists in their queries. Instead, their queries involved only biological **entities** and **relationships** between entities. An example of such queries is *"Return all contigs that map 'close' to marker M on chromosome 19"* which includes the biological entities CONTIG, MARKER and CHROMOSOME and includes the relationships "maps close to" and "(located) on". We conclude that scientists find it very useful not to have to specify the sources and tools that is, to make **transparent** queries [10].

Follow-up interviews showed that scientists want to ask transparent queries while being aware of the **origin of the answers** obtained. They want to know the *why-provenance* [1] that is, which sources and/or which tools have been used to calculate the data they obtain. Traceability is particularly important for verifying results, drawing conclusions and testing biological hypotheses [19].

## 2.3 Source and tool requirements

A more complex step in the querying process is the **assembly** of information between entities. From the sample queries, we observed that relationships between entities are either explicitly **stored** in the sources or **calculated** by a bioinformatics tool. For example, in the query *"Return all contigs that map 'close' to marker M on chromosome 19"*, the fact that Marker M is on chromosome 19 must be *stored* in the data sources queried by the biologist. Conversely, the relationship of "close mapping" can be *calculated* (e.g. using *Blastn*). For each calculated relationship between entities, we also determined which tools were used to achieve it (e.g. *Blastn*) based on the interview information.

Different kinds of **links** between sources may therefore be distinguished: *internal links* (within the same source), *cross-references* (between different sources) and *tool-links*. *Internal links* may be seen as a way of obtaining information on one entity from another entity within the same source. *Cross-references* are hypertext links from an entity in one source to complementary information in another source, and are not necessarily symmetric (e.g. there are an increasing number of specialized sources which crossreference GenBank but are not referenced in return). Finally, *tool-links* are services provided by a source, yielding links with entities in other sources. Each source may provide several different services achieving a given relationship. For example, GenBank provides different tools (e.g. *Blastx, tBlastn*) to enable users to carrying out "similarity searches" between the genes of GenBank and proteins of various sources.

It is also clear from interviews that scientists have **preferences** concerning entities in sources and tools. One of the key issues facing bioinformaticians is therefore to help the scientists to evaluate their confidence in sources and tools, and to make use of this confidence in a semi-automatic querying process. We return to this in section 4.

## 3 Strategies

Interviews revealed that each scientist followed paths between sources and queried the sources by first considering each entity for which information was sought and then by linking information about entities by means of cross-references or tools. Since information is collected entity by entity, each entity is treated exactly once. However, the scientists differed considerably in other aspects of querying, in particular whether or not (i) they followed an order on the entities, (ii) they were willing to explore other entities, and (iii) they were willing to visit a source more than once. We term these query criteria *Ord* (*Ordered*), *OnlyGE* (*Only-GivenEntities*) and *SourceOFA* (*SourceOnceForAll*), respectively, and call the combination of criteria the query **strategy**.

### 3.1 Querying entities by following an order

The first criterion, *Ord*, determines whether the entities of interest are searched in the given order or whether all orderings of the entities are considered. It is typically chosen when the scientists know that the desired information is provided by the given ordering, as opposed to when they want to get as much information as possible[4]. For example, if the scientists search for the chromosomal location of the sequence of a given BAC (Bacterial Artificial Chromosome), they may access a few sources containing BAC information and may follow cross-references to sources providing information about chromosomal location. In this situation, the scientists order the entities so as to start from the known entity and end with the entity sought; only links from BAC to CHROMOSOME are followed. However, if the information sought is not available in the data sources, the biologists may browse the sources to obtain as much information as possible. The two entities are therefore also considered in reverse order (from CHROMOSOME to BAC). Thus, they consider all the permutations between entities (from BAC to CHROMOSOME and from CHROMOSOME to BAC).

### 3.2 Querying Only Given Entities

The second criterion, *OnlyGE*, determines whether the scientists are interested in finding information using only the given entities, or whether they are willing to explore *additional* entities that are *biologically linked* to the entities explicitly sought. As an illustration, consider the previous example of scientists interested in finding data on the chromosomal location of a given BAC b. If the scientists do not find any information about the BAC b by querying sources for entities BAC and CHROMOSOME, they may consult sources providing information on other entities, such as GENE, and try to determine the location of genes known to be present on b. This makes it possible to determine the location of the BAC b.

---

[4] Note that if the entities are not ordered the non-symmetric aspect of links between sources can be resolved.

### 3.3 Querying a source once for all

The third strategy criterion, *SourceOFA*, determines whether or not a given source can be visited more than once. The second approach is primarily adopted by scientists who wish to validate information already obtained. Visiting a given source multiple times makes it possible for the biologist to check whether the information obtained - and to which further information has been added via the browsing of several sources - has remained coherent. This process is particularly important when the data reflects expertise, as experts may disagree, resulting in divergent data. Continuing with our example, the scientists may query the source MapView to obtain data about a given BAC and follow a cross-reference to GenBank to find the chromosomal location of that BAC. GenBank is queried here because it contains all the available genomic data. However, GenBank is a large public data repository, containing information originating from many different laboratories; therefore, some of the data it contains may be erroneous. The biologists then follow links from localization information in GenBank to the same kind of information in MapView to compare the results.

### 3.4 Combining the criteria

Interestingly, criteria may be combined, generating a wide variety of querying processes. Scientists typically adopt the simple strategy where the criteria *Ord*, *OnlyGE*, *SourceOFA* are chosen. If the results obtained are not satisfactory, the scientists may then drop one of these criteria, e.g. allow the entities to be queried in any order. Section 6 shows how following strategies allows the scientists to find complementary data and to deal with divergent data. We will also see how allowing them to choose his/her **strategy** represents a real challenge in the development of systems providing support for the querying process.

## 4 Management of Preferences

Our goal is to get as much information as possible from the sources using alternative paths that follow the chosen strategy. Unfortunately, the number of alternative paths may be very large. BioGuide therefore allows users to state **preferences** to filter and rank the paths considered.

### 4.1 Initializing preferences

Responses to our questionnaire showed that the reason why a source or tool is preferred varies between scientists. Interviews revealed that about 30 criteria determine preferences (e.g. reliability, completeness and ease of use), mainly in association with entities in sources and links between them. Some users even base their preferences for tool-links on the sources which provide them. We thus asked and helped the users to quantify the confidence that they have in the components of each path, i.e. entities in sources and links between them.

To guide the user, initial confidence values for components of a path can be automatically generated using information such as the *average speed* of a tool, or the *source-entity cardinality* (i.e. an estimate of the number of instances of an entity in a source) [9]. These initial values may then be improved, adjusted or rectified by comparing the values obtained for all the source-entities related to a given set of entities and/or to a given set of sources. BioGuide provides a user-friendly interface (Fig. 1) through which the user can adjust the improved initial values.
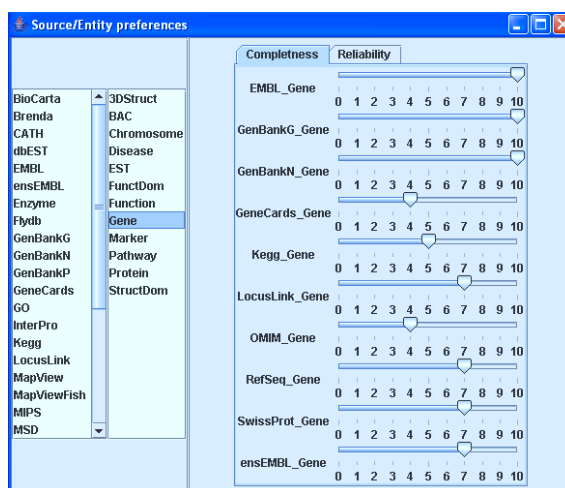


**Fig. 1.** Initializing Preferences

### 4.2 Using values of preferences

Firstly, we introduce the notion of **level of filter preference** and distinguish three different levels: (i) *global*, (ii) *intermediate* and (iii) *local*. The *global level* corresponds to a filter on a path, i.e. on the sequence of sources and links taken as a whole. Filters at the *intermediate level* focus on a given entity or relationship. At the *local level*, filters relate to a given source or a given link, allowing the biologist to name the source/tool to use. Section 6 will illustrate this notion.

If the number of alternative paths is still too large, we can **sort** them according to the biologist's preferences [2], [9]. To do this, we must associate a value with each path. The way in which the global value of a path is computed from the confidence assigned to its components (source-entities and links), i.e. the *sort-operation* used (e.g. the *weighted sum*), can vary (cf. BioGuide site).

# 5 BioGuide: querying according to strategies

In this section we introduce the architecture of BioGuide (see Fig. 2) and then describe more precisely its two main modules: *EntityPathsGenerator*(EPG) and *SourceEntityPathTranslator* (SEPT).

## 5.1 Architecture

From a query expressed in natural language ($Q_{nat}$), the scientist first has to extract the underlying biological entities and the relationships between them ($Q_{entRel}$). In BioGuide, this pre-process is performed by the user, but could easily be automated, as described by [16]. BioGuide supports biologists in this task by providing a graph of entities (described in the next subsection).

The steps (I) to (IV) of the BioGuide process are shown in Fig. 2. (I) The *initial user's query Q* consists of (i) $Q_{entRel}$, the entities and relationships underlying the user's query; and (ii) the choice of the user concerning entity related strategy criteria (*Ord* and *OnlyGE*). (II) From $Q$, the *EPG* module yields $P_e$, the set of paths in the graph of entities generated according to the entity related strategy criteria. (III) The *extended user's query* $Q_{se}$ consists of (a) $P_e$, the output of the *EPG* module, (b) the choice of the user concerning the strategy criterion *SourceOFA*, and (c) the user's preferences. (IV) Using $Q_{se}$ and the source-entities graph, the *SEPT* module generates the list $L_{pse}$ of paths between source-entities that can be used to retrieve the data.
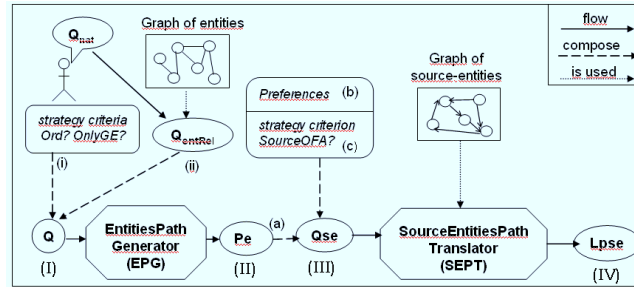


**Fig. 2.** BioGuide architecture

## 5.2 *EntityPathsGenerator*: Transparency and strategies

We now present how the *EPG* module processes and we describe its components.

**Graph of entities:** We extracted entities and relationships from the collected queries and used the answers given during interviews to build the *graph of entities*. The nodes are the biological entities and the edges are the biological relationships between them (see Fig. 3). This graph expresses biological knowledge

(e.g. *proteins are encoded by genes*), bioinformatics knowledge about tools (e.g. *proteins and genes may be similar*) and knowledge about sources (e.g. *information on disease often cross-reference information on 3D-structure*). Labels on the edges specify the kind of semantic relationship between these entities. The users can make use of this graph to build questions by selecting entities and, possibly, relationships between these entities. Moreover, if they do not want to only consider the given entities of their query, they may characterize the *additional entities* and relationships that they would like to consider or to avoid. This can be done by explicitly referring to entities and relationships or by specifying the kind of relationships (e.g. those achieved by tools) used to reach these *additional* entities. We now present more formally the notion of *initial query*.
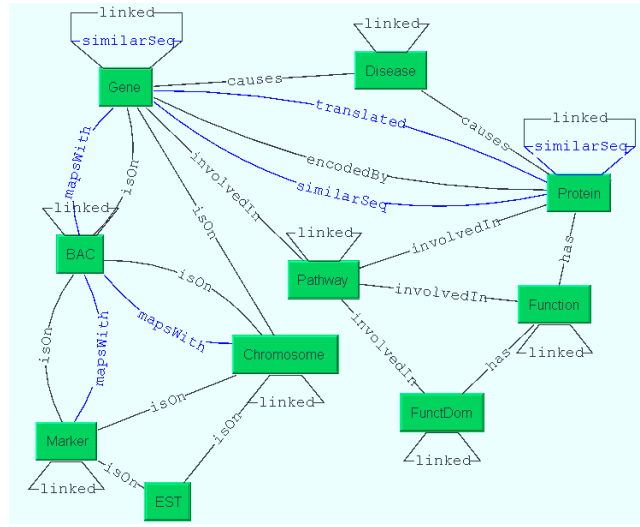


**Fig. 3.** Graph of Entities (Subpart)

**Input of the EPG module: *Q*.** The ***initial user's query*** is Q={L$_{Ent}$, S$_{Rel}$, StrategyE, S$_{noEnt}$, S$_{noRel}$, PropertiesRel} where $L_{Ent}$ and $S_{Rel}$ denote the list of entities and the set of names of relationships (possibly empty), respectively; StrategyE contains the choice of the user concerning the strategy criteria *Ord* and *OnlyGE*; if *OnlyGE* is not chosen by the user then (a) s/he may specify which entities (or relationships) s/he wishes to avoid, by adding them to the set S$_{noEnt}$ (or S$_{noRel}$) and (b) PropertiesRel is a conjunction of properties expressing which *kinds* of relationships can be used to reach *additional* entities.

As an illustration, consider the previous example in which the user wishes to find information connecting a given BAC and its Chromosomal location (L$_{Ent}$=[BAC, CHROMOSOME]) without choosing an order between entities and

considering not only the given entities of his/her query (StrategyE ={}). The user may wish to avoid distant entities such as EST ($S_{noEnt}$={EST}) and may choose to follow only non-tool relationships($S_{noEnt}$={}, $S_{noRel}$={}, Properties-Rel=OnlyNonTool).

The *EPG* module is based on an **algorithm** which aims at calculating from $Q$ the corresponding set of paths in the graph of entities. As an illustration, the following paths are returned by *EPG* from the previous query: BAC isOn CHROMO[5], CHROMO isOn BAC, BAC isOn GENE, GENE isOn CHROMO[6].

**Output of the EPG module: $P_e$.** More formally, the *EPG* module calculates $P_e$, the set of paths in the graph of entities which respect the following four properties. (1) Each path in $P_e$ contains all the entities and relationships specified by the user and visits each entity once only. Moreover, (2) if the user has chosen the strategy criterion *Ord* then the entities in each path must be considered in the order indicated in the list $L_{Ent}$, and (3) if the user has chosen the criterion *OnlyGE* then each entity of each path must belong to $L_{Ent}$. Conversely, (4) if *OnlyGE* has not been chosen, paths may consider *additional* entities and relationships (i.e. not specified in $L_{Ent}$ and $S_{Rel}$). In this case, these entities and relationships must be different from those in $S_{noEnt}$, $S_{noRel}$ and the edges followed must satisfy conditions expressed in PropertiesRel.

The EPG algorithm is **sound and complete** with respect to these properties.

### 5.3 *SourceEntityPathTranslator*: preferences and strategies

The next step involves finding the sources containing entities and the links giving relationships, which is the aim of the SEPT module that we present with its main components here-after.

**The Graph of Source-entities:** After carrying out a thorough study of the sources and tools mentioned in interviews, we designed a graph of source-entities (see Fig. 4). Each node represents an entity in a source. Arrows indicate the links between a given entity in a source and another entity (in the same source or another source). Labels on arrows specify the kind of link. *CrossRef* and *Internal* labels indicate cross-reference and internal links, respectively. Other labels (such as *Blast*) refer to tools.

More formally, let E be the finite set of biological entities (e.g. BAC, GENE), and R be the set of pairs of entities linked by relationships. Let $Lab_r$ be the finite set of labels of relationships between entities (e.g. *SimilarTo*), S be a finite set of data sources (e.g. *GenBank*), N⊆SxE be the set of pairs (source,entity) (e.g. *(GenBank,*GENE*)*), A be the set of directed links (arrows) between (source,entity) pairs, and $Lab_l$ be the finite set of labels of links (e.g. *CrossRef, Blast*) between

---

[5] CHROMO will stand for CHROMOSOME.

[6] Relationships between entities are symmetric.
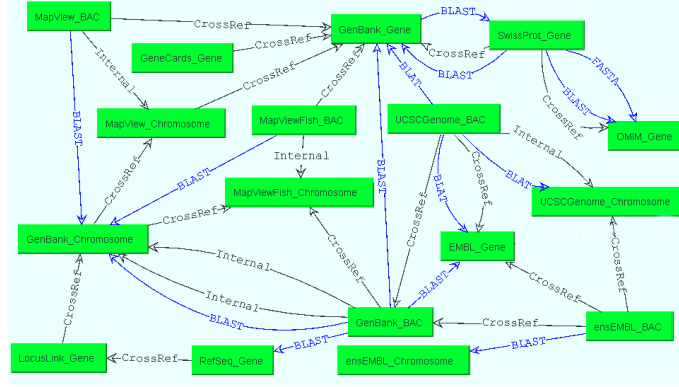
**Fig. 4.** Graph of Source-Entities (Subpart: only source-entities relating to BAC, CHRO-MOSOME and GENE)

(source,entity) pairs. $Lab_l$ contains the names of the links achieving relationships, the names of which are in $Lab_r$. In the rest of the paper we will use the following abbreviations to mention sources: GB, LL, RF, MV, MVF and UG stand for GenBank, RefSeq, LocusLink, MapView, MapViewFish and UCSCGenome.

**Definition 1.** *The **GraphOfSourceEntities** is a directed labelled graph given by the 3-tuple $(N, A, f_{labl})$, where (1) N is the set of nodes given as (source,entity) (2) $A \subseteq N \, x \, N$ is the set of arrows (directed links between nodes) (3) $f_{labl}: A \rightarrow Lab_l$ provides the label of each arrow.*

**Definition 2.** *A **path in GraphOfSourceEntities** is a sequence of pairs of arrows and labels, $(a_1, l_1), (a_2, l_2), ..., (a_k, l_k)$ such that, for $i$ $(1 \leq i \leq k)$, $a_i$ is an arrow from the node $n_{i-1}$ to the node $n_i$ (adjacent arrows) and such that $n_i \neq n_j$ (no cyclic path), for $i \neq j$, $(0 \leq i, j \leq k)$. The **length** of the path is k, the number of arrows.*

The GraphOfSourceEntities is constructed so that: (i) (s,e) is a node if and only if the source s contains the entity e and (ii) a=(s,e) l (s',e') is an arrow if and only if (1) the source s provides a link labelled by l from entity e to entity e' of source s' and (2) there is a relationship r in the graph of entities between e and e' such that l achieves the relation r.

Using the GraphOfSourceEntities the users can specify their filter preferences. In this step, the users may also define their sort preferences and select whether or not they wish to consider each source once for all. We present more formally the notion of *extended query* based on the graph of source-entities.

**Input of the SEPT module: $Q_{se}$.** The **extended query** of the user (cf. Figure 2 step (III)) is $Q_{se} = \{P_e, PrefCond, L_{rank}, Op_{rank}, StrategyS\}$ where $P_e$ is the set of paths in the graph of entities obtained from Q (cf. section 5.1);

PrefCond is a boolean formula expressing filter preferences on paths of source-entities (cf. section 4.1); $L_{rank}$ is a list of pairs (entity, preference criterion) used to rank the paths; $Op_{rank}$ is the *sort-operation* chosen to calculate the value of the preference on each path from the value of preference criteria for its components (pairs of source-entities and links); and StrategyS describes the choice of the user concerning the criterion *SourceOFA* (cf. section 3.3).

The *SEPT* module is based on an algorithm which aims at calculating from $Q_{se}$ the corresponding list of paths in the graph of source-entities, $L_{pse}$. An example of path in $L_{pse}$ is $p_{se}$=(GB,BAC) $\overset{BlastN\_NCBI}{\rightarrow}$ (RS,GENE) $\overset{CrossRef}{\rightarrow}$ (LL, GENE) $\overset{CrossRef}{\rightarrow}$ (GB, CHROMO). Let us mention that this path have been generated using the path $p_e$=BAC *mapsWith* GENE *isOn* CHROMO of $P_e$.

**Definition 3.** *Let us consider* $p_e = e_1 r_1 ... r_{t-1} e_t$ *a path of Pe,* $p_{se} = (s_1, e_1) l_1 (s_2, e_2)$ *...*$l_{n-1}(s_n, e_n)$ *a path of* $P_{se}$ *and m the number of entities in the query.* $p_{se}$ ***corresponds to*** $p_e$ *if and only if (1) the set of entities of* $p_{se}$ *is equal to the set of entities in* $p_e$ *and entities in* $p_{se}$ *appear in the same order as in* $p_e$ *; (2) several source-entities concerning the same entity are possible in* $p_{se}$ *($m \leq n$) but they must be consecutive and linked by cross-references ; (3) let* $(s_i, e_i)$ $l_i$ $(s_{i+1}, e_{i+1})$ *be an arrow of* $p_{se}$ *($1 \leq i \prec n$), if* $e_i$ *and* $e_{i+1}$ *are occurrences of two distinct entities, x and y, there must be an arrow x r y in* $p_e$ *such that* $l_i$ *achieve r ($\exists j, 1 \leq j \prec m$, $x = x_i$, $y = x_{j+1}$ and $r = r_j$).*

Let us return to our example. The path $p_{se}$ corresponds to $p_e$ since: (1) entities are the same and are in the same order; (2) the source-entities related to the GENE entity are consecutive and linked with cross-references; and (3) the BlastN_NCBI tool and a cross-reference achieve the relationships *mapsWith* and *isOn*.

**Output of the SEPT module: $L_{pse}$.** From $Q_{se}$ the SEPT module yields $L_{pse}$ a list of paths in the graph of source-entities. These paths satisfy the three following properties: (1) Paths of $L_{pse}$ *correspond to* paths of $P_e$ according to the previous definition; (2) each path in $L_{pse}$ satisfies the preference filters; (3) the list of paths in $L_{pse}$ is ranked following sort-preferences specified in $Op_{rank}$ and $L_{rank}$. The SEPT algorithm is **correct and complete** with respect to these properties.

### 5.4   Towards a meaning for source-entities paths

We provide below the meaning of paths between source-entities from a relational database perspective: (i) each node (s,e) in the graph of source-entities is a **view** over the source s of the entity e (represented by a table s_e); and (ii) each link is a kind of *join*. More precisely, tool-links are mapped to a particular conditional join, the **similarity join**, in which data are joined if and only if they are very similar [17]. We considered several similarity functions based on those used by tools (Blast etc.). Furthermore, *internal* and *cross-reference* links are mapped

to a **link-join**. A *link-join* between two tables si_ek and sj_ek' (respectively related to source-entities $(si, ek)$ and $(sj, ek')$), with $id$ as identifier (primary key), is defined by using the table *Link(IdBeg, SourceBeg, IdEnd, SourceEnd)* as follows $si\_ek \bowtie_{(si\_ek.id=Link.idBegin)} Link \bowtie_{(sj\_ek'.id=Link.idEnd)} sj\_ek'$. *Link* contains internal and cross-reference links. A tuple $(i_1, s_1, i_2, s_2)$ is in Link if there is a cross-reference (or internal link) from a biological data identified by $i_1$ in $s_1$ to another data identified by $i_2$ in $s_2$.

Consequently, depending on whether the **Ord** criterion is chosen or not, different paths are generated. Consider two ordered entities $e_1$ and $e_2$: if only one tuple of the form $(i_2, s_2, i_1, s_1)$ concerns $s_1$ and $s_2$ in the *Link* table, then no path between $s_1$ and $s_2$ is generated. Conversely, if *Ord* is dropped then the path $(s_2, e_2) \rightarrow (s_1, e_1)$ is generated. Furthermore, if the criterion **OnlyGE** is dropped, new data may be found due to the ability to introduce new entities. Conversely, if **SourceOFA** is chosen then some links may be missed. With three entities, paths of the form $(s_1, e_1) \rightarrow (s_2, e_2) \rightarrow (s_1, e_3)$ cannot be calculated.

### 5.5 Complexity

The **complexity** of BioGuide is related to the number of source-entities paths generated. The worst case occurs when the graphs of entities and source-entities are complete. Table 1 gives the number of entities paths generated by *EPG* according to the strategy followed. q is the number of entities of the query, n+q is the number of entities in the graph of entities.

In any strategy where *Ord* is dropped (cases b and d), all permutations between the q entities of the user's query are considered. In the case where $OGE$[7] is dropped and *Ord* is taken (c), all the paths with at most i additional entities between q entities are considered (n is the upper bound of i), the first entity and the last one staying fixed. Then, for each entity e, the maximal number of paths of source-entities only focused on e (i.e. each source-entity concerns e) generated is given by the following formula: $\sum_{k=1}^{nbse} \frac{nbse!}{(nbse-k)!}$ where $nbse$ is the number of sources that contain the entity e (k is the number of sources involved in the paths of source-entities).

In the worst case, the time complexity is very high. However, the queries identified in this study consider only a small number of entities at the same time (only 8 % of the queries had more than three entities) and the source-entities paths desired by the user rarely exceed 6 source-entities. Moreover, BioGuide generates paths that are shorter than 15 source-entities long in less than 1 second.

**Table 1.** Number of paths depending on the criteria combination

| a. {OGE, Ord} | b. {OGE} | c. {Ord} | d. *no criteria* |
|---|---|---|---|
| 1 | q! | $\sum_{i=0}^{n} \frac{(i+q-2)!}{(q-2)!}$ | $q(q-1)\sum_{i=0}^{n}(i+q-2)!$ |

---

[7] OGE stands for OnlyGE.

# 6 Results

## 6.1 Using Strategies

The ability to use different strategies and alternative ways of retrieving data across sources, combined with the ability to use tools and take user preferences into account, was considered very useful by the biologists interviewed. A knowledge of which **tools** may be used for a particular bioinformatics task was considered important in a variety of domains, such as the annotation of newly acquired genome with sequence similarity search and 3D-structure analysis with structure comparison. Moreover, all of the biologists questioned used strategies where they do not **limit them to query the entities** of their query. For example, in cancer studies knowledge about PROTEINs and PATHWAYs is obtained using FUNCTION as an additional entity. In protein-protein docking studies, biologists may use STRUCTURALDOMAINs to link PROTEIN and 3D-STRUCTURE. In annotation projects, the CHROMOSOMAL location of INTRONs is found using data about ESTs. Furthermore, more than half the interviewees frequently adopted strategies where no **order** is fixed between entities. Only when the goal of the search was to find very high-quality data did biologists adopt strategies with a fix order between entities. This is the case when searching for samples for expensive experiments (e.g. crystallization of PROTEINs). Finally, strategies where a **source is queried once for all** are adopted by biologists for only a very small number of sources in which they have a high level of confidence. In most cases, strategies where sources are queried several times are adopted to ensure that the results obtained are reliable.

## 6.2 Example of CGH analysis

A principal example of the use of BioGuide concerns the task of positioning genomic BAC clones on the draft of the human genome sequence [2]. In CGH (Comparative Genomic Hybridization) array experiments, BACs are used to identify new cancer-related genes and it is of the utmost importance to know the precise position of BACs on the genome sequence. We will study the following query: "Where are **all the BACs** of my CGH array located on the genome sequence?" where the underlying entities are BAC and CHROMOSOME.[8] We initially assumed that the scientist adopted a *simple* strategy choosing all of the criteria (*Ord*, *OnlyGE*, and *SourceOFA*). As for preferences, we assumed that the user indicated the following filters: no source with low completeness whatever the entity is (*global* level), no source providing CHROMOSOME with a medium reliability (*intermediate* level), and the ensEMBL source should not be queried (*local* level). The user also indicated that the results should be sorted by considering *completeness* for BAC and *reliability* for CHROMOSOME. The sort-operation is the weighted sum. Based on these filters and strategy criteria, BioGuide yielded

---

[8] Sources were queried on January 3, 2005; more details on this example are available from the BioGuide web site.

seven source-entities paths. Instantiated data have been got using BioGuide within the HKIS platform[9] [2].

The results given by these paths are complementary, providing information on different instances of BACs. They also give complementary results concerning single instances of BACs. For example, the path (MVF,Bac) $\overset{Internal}{\rightarrow}$(MVF,Chromo) localizes BAC RP11-89F21 on chromosome band 17p11.2 whereas the path (UG,Bac) $\overset{Internal}{\rightarrow}$ (UG,Chromo) is more precise, giving the exact position of this BAC on the chromosome sequence (15,021,683-15,022,225). More globally, these source-entities paths yield the location of about 80% of the BACs.

Let us assume that the user then tries to obtain information about the 20% missing BACs by adopting a more complex strategy without *OnlyGE*. The user also chooses to follow relationships achieved by tools, and not to consider Marker as an additional entity. A new path of entities is generated with Gene as an additional entity. In the corresponding source-entities paths, all the missing BACs can now be located. For example, due to the path (GB,Bac) $\overset{BlastN\_NCBI}{\rightarrow}$ (RS,Gene) $\overset{CrossRef}{\rightarrow}$ (LL,Gene) $\overset{CrossRef}{\rightarrow}$ (GB,Chromo) the chromosomal location of BAC RP11-782H1 was found. More precisely, this BAC (entry AC025749 in GB) mapped with (using the BlastN tool from NCBI) the gene P85B (entry $NM\_005027$ in RS, which cross-refers entry 5296 in LL), which is is located on chromosome 19 (in GB PIK3R2 entry).

Finally, let us assume that the scientist then analyzed the results obtained. Several **divergent** locations were produced by these paths for the BACs CTD-2012D15 and CTD-2008I6. Indeed, BAC CTD-2012D15 may be considered to be located on chromosome X or 11. As sources locating the BAC on chromosome X (GB and MV) are considered less reliable than those locating the BAC on chromosome 11 (UG and MVF), the user is likely to consider it more probable that BAC CTD-2012D15 is located on chromosome 11 [2]. Conversely, the sources involved in the paths which locate the BAC CTD-2008I6 on chromosome 3 or 17 (UG and MVF) are considered to be equally reliable. The biologist must therefore explore new paths to correlate these pieces of information, and does it by adopting a strategy without *SourceOFA* and by considering tools-relationships between Bac and Chromosome. Consequently a new path is generated: (UG,Bac) $\overset{Blat\_UCSC}{\rightarrow}$ (UG,Chromo). The results provided allow the user to conclude that BAC CTD-2008I6 is duplicated in the genome, and is present on both chromosomes 3 and 17.

Due to its multiple-strategies approach, BioGuide enables the users to make the most of the available data and guides them to deal with divergent data.

## 7    Discussion and Conclusion

Based on a thorough study of scientists' needs, we have designed a **user-centric framework** to specify the notions of *queries*, *preferences* and *strategies*. From this framework we have proposed and implemented the BioGuide system which

---

[9] http://www.hkis-project.com

calculates the *paths* between source-entities. Then, we have presented the architecture of BioGuide and have provided a very easy-to-use **implementation**.

Over the last few years, three systems considering paths between sources have been developed: Biomediator [12], Bionavigation [8] [9] and DSS [2]. We sum-up the differences between our approach and these works. Firstly, the underlying query languages of [9] and [12] [13] are formal query languages: a *regular expressions based query language* and an XML-based path language called PQL, respectively. Following our user-centric approach we have proposed a user-friendly **graphical query language**. This language enables to express the strategy criteria which came out of the user requirements. Any query with a strategy combining the presence/absence of the $OnlyGE$ and $Ord$ criteria can be expressed using the query languages of [13] and [9]. Note that writing such queries may be a complex task (e.g. if $Ord$ is dropped then the user has to enumerate all the possible orders between entities of his/her query). Finally, [12] and [9] require the $SourceOFA$ criterion to be present ([12] and [9] do not provide a way of visiting a given source several times in a given path). In DSS, there is only one available strategy where the $OnlyGE$ criterion is present and the other criteria are dropped.

Furthermore, each of these systems considers user preferences at different levels: [2] considers only global preferences whereas [9] considers both global and intermediate preferences (called meta-data in [9]). Only BioGuide considers all levels of preferences as far as it allows to deal with local preferences (sources can be named) too. Last but not least, BioGuide differs from the previous works in that it is based on labelled-graphs (graphs of entities and source-entities) in which two given entities (resp. source-entities) may be related by several biological relationships (resp. links like cross-references or tools). Therefore BioGuide yields many more alternative paths.

BioGuide thus provides a framework which is general enough to take into account all the abilities (strategies and preferences) of current systems and enables to specify new preferences and strategies. Its implementation allows these abilities to be managed in a simple yet unified and graphical way. We have shown the benefit of BioGuide by highlighting the biological relevance of the alternative paths obtained, through examples in various biological domains. BioGuide has been implemented and is very flexible allowing users to adapt the graphs and the preferences according to his/her needs. It is available for use at http://www.lri.fr/~cohen/bioguide/bioguide.html.

We are currently adding methods to filter and rank the paths in the spirit of [9]. Moreover, as BioGuide is architecture-independent we are studying its use in different integration systems: browsers (SRS [7]) but also mediators (K2 [3]).

---

[10] The exhaustive list of interviewed biologists is available on the Web site.

# References

1. Buneman, P., Khanna, S., Tan, W.: Why and Where: A Characterization of Data Provenance, *Proc. of Int. Conf. on Database Theory (ICDT),316-330*, 2001.
2. Cohen-Boulakia, S., Lair, S., Stransky, N., Graziani, S., Radvanyi, F., Barillot, E., Froidevaux, C.: Selecting biomedical data sources according to user preferences, *Bioinformatics, Proc. ISMB/ECCB04*, **20**, i86-i93, 2004.
3. Davidson, S., Crabtree, J., Brunk, B., Schug, J., Tannen, V., Overton, C., Stoeckert, C.: K2/Kleisli and GUS: Experiments in integrated access to genomic data sources *IBM Systems Journal*, 2001.
4. De Santis, L., Scannapieco, M., Catarci, T.: Trusting Data Quality in Cooperative Information Systems, *Proc. of CoopIS/DOA/ODBASE 2003*, 354-369, 2003.
5. Donelson, L., Tarczy-Hornoch, P., Mork, P., Dolan, C., Mitchell, J., Barrier, M., Mei, H.: The BioMediator System as a Data Integration Tool to Answer Diverse Biologic Queries, *Proc. of MedInfo, IMIA (in CDROM)*, 2004.
6. Ely, J.W., Osheroff, J.A., Gorman, P.N., Ebell, M.H., Chambliss, M.L., Pifer, E.A., Stavri, P.Z.: A taxonomy of generic clinical questions: classification study, *British Medical Journal BMJ* **321 (7258)**, 429-432, 2000.
7. Etzold, T., Ulyanov, A. and Argos, P.: SRS: information retrieval system for molecular biology data banks. *Methods Enzymol*, **266**, 114-128, 1996.
8. Lacroix,Z., Parekh K., Raschid,L., Vidal,M.: Navigating through the Biological Maze, *Proc. Int. IEEE Computational Systems Bioinformatics (CSB)*, 594-595, 2004.
9. Lacroix,Z., Raschid,L., Vidal,M.: Efficient Techniques to Explore and Rank Paths in Life Science Data Sources, *Proc. Data Integration in the Life Sciences*, 187-202, 2004.
10. Levy, A.Y.: Combining Artificial Intelligent and Databases for Data Integration *Artificial Intelligence Today*, 249-268, 1999.
11. Lord, P., Bechhofer, S., Wilkinson, M.D., Schiltz, G., Gessler, D., Hull, D., Goble, C., Stein, L.: Applying Semantic Web Services to Bioinformatics: Experiences Gained, Lessons Learnt *Proc. of Semantic Web Conference (ISWC2004)*, 350-364, 2004.
12. Mork, P., Halevy, A., Tarczy-Hornoch, P.: A model for data integration systems of biomedical data applied to online genetic databases, *AMIA Symp*, 473-477, 2001.
13. Mork, P., Shaker, A., Halevy, A., Tarczy-Hornoch, P.: PQL: A declarative query language over dynamic biological schemata, *Proc. AMIA Symp*, 533-537, 2002.
14. Muller, H., Naumann, F.: Data Quality in Genome Databases, *Proc. Int. Conf. on Information Quality*, 269-284, 2003.
15. Naumann, F., Leser, U., Freytag, J.C.: Quality-driven Integration of Heterogenous Information Systems, *Proc. Int. Conf. Very Large DataBases (VLDB)*, 447-458, 1999.
16. Samsonova, M., Pisarev, A., Blagov, M.: Processing of natural language queries to a relational database, *Bioinformatics*,**19**, i241-i249, 2003.
17. Schallehn, E., Sattler, K-U., Saake, G.: Efficient similarity-based operations for data integration, *Data and Knowledge Engineering*,**48**, 361-387, 2003.
18. Stevens, R.D., Goble, A.C., Baker, P.G., Brass,A.: A classification of tasks in bioinformatics, *Bioinformatics*, **17(1)**, 180-188, 2001.
19. Zhao, J., Wroe, C., Goble, C., Stevens, R., Quan, D. and Greenwood, M.: Using Semantic Web Technologies for Representing e-Science Provenance *Proc of Semantic Web Conference (ISWC2004)*, 92-106, 2004.