BIOINFORMATICS

Vol. 19 Suppl. 1 2003, pages i241–i249
DOI: 10.1093/bioinformatics/btg1033

Processing of natural language queries to a relational database

M. Samsonova, A. Pisarev and M. Blagov*

St. Petersburg State Polytechnical University, 21, Tikhoretsky ave., St. Petersburg, 194064, Russia

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: A new method is developed to query a relational database in natural language (NL).

Results: The method, based on a semantic approach, interprets grammatical and lexical units of a natural language into concepts of subject domain, which are given in a conceptual scheme. The conceptual scheme is mapped formally onto the logical scheme. We applied the method to query the FlyEx database in natural language. FlyEx contains information on the expression of segmentation genes in *Drosophila melanogaster*. The method allows formulation of queries in various natural languages simultaneously, and is adaptive to changes in the knowledge domain and user's views. It provides optimal transformation of queries from natural language to SQL, as well as visualization of information as a hyperscheme. The method does not require specification of all possible language constructions as well as a standard grammar accuracy in formulation of NL queries.

Availability: http://urchin.spbcas.ru/NLP/NLP.html

Contact: samson@spbcas.ru

Keywords: natural language processing, relational databases, conceptual scheme, segmentation genes, *D.melanogaster*, query optimization.

INTRODUCTION

Natural language processing (NLP) is an emerging field in bioinformatics. The goal of NLP is to develop a new generation of data mining and analysis tools. One of the applications of NLP is the retrieval of information from a database on the basis of its pertinence to user's query.

Currently relational database management systems (RDMS) has become a standard technology to manage large volumes of biomedical information. The information retrieval from most databases consists of a specification of search parameters by filling in a form. When a parameter set is large, a form with many control elements could occupy more than a whole screen. As a result, at each moment a user grasps only part of the information provided by the form. The subdivision of large forms into small parts and their association into a wizard does not solve the problem, since the user quickly gets tired, cannot see next step and thus is not sure what to do.

However, friendly the interface may be, there is always a gap between the user's intention and its realization by means of the interface tools. Therefore an ideal communication between user and computer would be in a natural language (NL). Evidently the user masters her native language much better than any artificial language, no matter which programming or visual interaction language would be used. Substitution of many form fields by a single field for input of the query in NL makes the search convenient as the user is able to see what is selected at each step, as well as to correct her choice made at the previous step by simply editing a phrase in the input field.

However, a single empty input field might discourage the user, if she is not an expert in the subject domain. Indeed, what and how can she ask? Certainly this category of users should be guided, and a role of the guide is to fulfill the conceptual scheme of the subject domain.

The conceptual scheme presents basic concepts in the domain, machine-interpretable definitions of basic concepts, as well as relations among them graphically. Usually a conceptual scheme of the domain is not a goal itself. Developing the scheme is akin to the definiton of a set of data and their relations for use in other programs.

In this paper we develop the conceptual scheme of the information on expression of segmentation genes in the fruit fly *Drosophila melanogaster* aimed to assist in formulation of queries in NL to the database. The conceptual scheme will help a user, who must learn what terms in the domain mean, by making explicit definitions of these terms. Another reason for development of the conceptual scheme is the complexity of biomedical language, which is rife with jargon, synonyms and equivocal terms. The conceptual scheme assists in processing queries in NL by inclusion of these terms as concepts.

It is important to mention that the term 'natural language' is quite ambiguous. Its original interpretation (traditional for artificial intelligence) implied a substantial di-

*To whom correspondence should be addressed.

alogue freedom between user and computer. It was assumed that this way of interaction would become quite common in time. However, a practice of construction of traditional NL systems (as a rule, based on syntactic approach) led to revision of the initial interpretation to the almost opposite one: often NL is now considered as a syntactically correct and rather constrained language genre.

In our implementation the term NL has intermediate meaning in comparison with the two extreme interpretations described above. From one side, total syntactical correctness of a query is not required and in many cases it may have the same telegraphic style as used by Internet search engines. From the other side we use a query understanding technology, which allows the maximum exploitation of pragmatic, semantic and lexical information from the query in order to specify its meaning.

SYSTEM AND METHODS

The subject domain and the database

At present, observational information about gene expression patterns is becoming available in unprecedented amounts. Scientific progress in developmental biology as well as improved understanding of genetically based diseases depends on the timely and full exploitation of this data.

Currently, the most detailed information with regard to time and space of gene expression in *D.melanogaster* is available for genes controlling segmentation (Akam, 1987; Ingham, 1988). Immediately following fertilization and egg deposition, the newly formed zygotic nucleus undergoes a series of rapid and synchronous nuclear divisions. By the ninth such division the nuclei have migrated to the cortex (outside) of the egg, and the embryo begins a stage of development called the syncytial blastoderm. This stage (also denoted as embryo stage 4) lasts from 90 to 130 minutes after fertilization and consists of three cleavage cycles from 10 to 13. The embryonic stage 5 lasts from 130 to 180 minutes after fertilization. Of particular importance is the time from the end of the nuclear division 13 to the gastrulation start, which is named as cleavage cycle 14A. At this time the segments are determined and the invagination of membranes and the cellularization of cells happens (Foe and Alberts, 1983).

The genetic network which controls segmentation in *D.melanogaster* is one of the few genetic networks fully characterized at a genetic and functional level (Wieschaus *et al.*, 1984; Nusslein-Volhard *et al.*, 1984). The initial determination of the segments is a consequence of the expression of 16 genes which are mainly transcription factors. Several of these genes, (named as maternal coordinate genes), are expressed from the maternal genome to provide asymmetric initial conditions. The others are zygotic and are expressed in patterns that become more spatially refined over time. Of particular importance are members of the 'gap','pair-rule' and 'segment-polarity' classes of segmentation genes (Akam, 1987; Ingham, 1988).

Different in situ hybridization methods were applied to study the level of RNA and protein encoded by segmentation genes. Recently we have introduced a new method to measure quantitatively a level of gene expression at cellular resolution (Myasnikova *et al.*, 2001). This method monitors the expression of segmentation genes at protein level and consists of several steps, as will be described below.

Gene expression was measured by confocal scanning microscope using fluorescence tagged antibodies. Each gene is detected in a single channel of a confocal microscope. Each embryo is scanned for the expression of three genes at a time. Three embryo images are combined and the resultant image is used to construct a binary nuclear mask. The nuclear mask is applied to reduce the embryo image to a table containing quantitative data on gene expression in each nucleus of the embryo. All these operations are performed using Khoros, which is a standard image processing package (2001).

We acquired gene expression data from fixed embryos for which a precise developmental time was not known. Thus the temporal dynamics must be reconstructed from many samples, each at a different stage of development. A fundamental step in such reconstruction is to determine the developmental age of each embryo. The cleavage cycle was used to stage embryos prior to cycle 14A, as each cleavage cycle from 9 to 13 is 12 minutes or less in duration. However cleavage cycle 14A is about 50 minutes long and therefore during this cycle other markers were used for staging embryos. First, all embryos were subdivided by visual inspection of expression patterns of pair-rule genes into 8 temporal equivalence classes. Next, the precise developmental age from the onset of cleavage cycle 14A was experimentally determined for 103 embryos using the standard curve which gives membrane invagination as a function of developmental time (Merrill *et al.*, 1988). Third, the developmental age of all other embryos was predicted using the pattern recognition method (Myasnikova *et al.*, 2002).

The quantitative gene expression data are further processed to yield the averaged data on expression of each segmentation gene, which were named as integrated data. The expression of segmentation genes is largely a function of position along the anterior-posterior (A-P) axis of the embryo body. This means that to characterize and to analyze the expression of segmentation genes it is sufficient to consider data in one dimension. The one-dimensional quantitative gene expression data were extracted from the central 10 % strip in the A-P direction.

The steps of processing of quantitative gene expression

data include data normalization, registration and averaging. During the data normalization step, the quantitative gene expression data were rescaled in order to get rid of distortions caused by the presence of a background signal.

To eliminate small individual differences between embryos, the quantitative gene expression data were subjected to registration. Two registration methods were used (Myasnikova *et al.*, 1999; Kozlov *et al.*, 2000). Both methods are based on the extraction of characteristic features in each image (which are named Ground Control Points, GCPs), and application of a coordinate transformation to make the corresponding GCPs in different images coincide as closely as possible.

At the data averaging step the registered and normalized data for a given gene, time class and embryo set were averaged and mapped onto a set of nuclei of the averaged embryo (Kozlov *et al.*, 2002). This operation yields the integrated data on expression of all segmentation genes at cellular resolution and at a given time interval. Two sets of integrated data were generated: for the 10% strip along the A-P axis and for the whole embryo image.

The application of the method described above results in acquisition of a large number of digital images of segmentation gene expression patterns together with quantitative data on the expression of genes in the segmentation genetic network at cellular resolution. This information is currently stored in FlyEx database (http://urchin.spbcas.ru/flyex; Samsonova *et al.*, 2000).

FlyEx is developed using IBM DB2 v7.2 as a DBMS. It contains images of 14 segmentation gene (*bcd, cad, hb, Kr, gt, kni, eve, ftz, h, run, odd, prd, slp, tll*) expression patterns obtained from 809 embryos, as well as quantitative and processed data on expression of these genes in each nucleus of each individual embryo image. The integrated data are available for 9 segmentation genes at 8 time points.

The conceptual scheme

The conceptual scheme is an oriented graph, in which nodes are concepts of the knowledge domain, and edges define relations between the concepts.

The structure of the scheme is defined as follows:
"→is characterized by"("EMBRYO", "DEVELOPMENTAL TIME")
"→is an instance of"("TIME FROM ONSET OF CYCLE 14A","DEVELOPMENTAL TIME")
"→is derived from"("TIME FROM ONSET OF CYCLE 14A","INVAGINATING MEMBRANE")
etc.
Each line in the list above defines a relation between two concepts.

To display the conceptual scheme we developed the algorithm of equirank sets. The conceptual scheme is formally described as a graph. The rank of each node in the graph is defined by its distance from the root node, which rank is equal to zero. The rank of the node is determined recursively - the rank of a parent node exceeds by 1 the maximal rank of its child nodes. The nodes with identical rank are placed in one line. The layout of nodes in the line depends on their interrelations: the algorithm minimizes a sum of edge lengths under constraints on minimal distance between these nodes.

Processing of NL queries

There are semantic and syntactic approaches to automatic analysis and understanding of NL texts. The traditional syntactic approach is based on the analysis of a sentence in order to determine its structure according to grammar. This approach causes several problems both for developers and users. Developers have to specify all possible kinds of queries and user's queries may be misunderstood due to implicit syntactic errors. Moreover, systems based on the syntactic approach depend entirely on a specific language, its syntax and grammar, and it is practically impossible to use them to process any other language. For example, the syntactic approach is implemented in (Microsoft English Query, 1998).

We use language understanding technology based on the semantic approach. This technology was introduced by Narinyany (1979) and developed by Zhigalov (1998). It interprets the grammatical and lexical units of any NL into concepts of a knowledge domain.

In Figure 1, the main steps of processing of a query in NL in our system are presented. Firstly, synonyms and high level concepts are substituted by terms of logical level, which are mapped on the database objects. At the step called 'Search in the Dictionary' an initial chain of semantic components is constructed. The step 'Semantic analysis' converts the initial chain of semantic components into a semantic network, which formally represents a query. At the 'SQL query generation and optimization' step the semantic network is converted into the SQL query to the database. The 'Advanced processing of queries' step performs the subject domain specific processing of queries, e.g. displays data in different views (as a table, graph or image).

All programs are written in Java.

Processing of synonyms and high level concepts. The procedure of processing of NL queries transforms different combinations of word forms to a limited set of terms of the logical level, which are used for a generation of the SQL queries to the database. Besides the logical level terms a query can contain their synonyms or concepts of higher level (hypernyms).

Transformation of synonyms and hypernyms to logical level terms is performed by means of special rules written

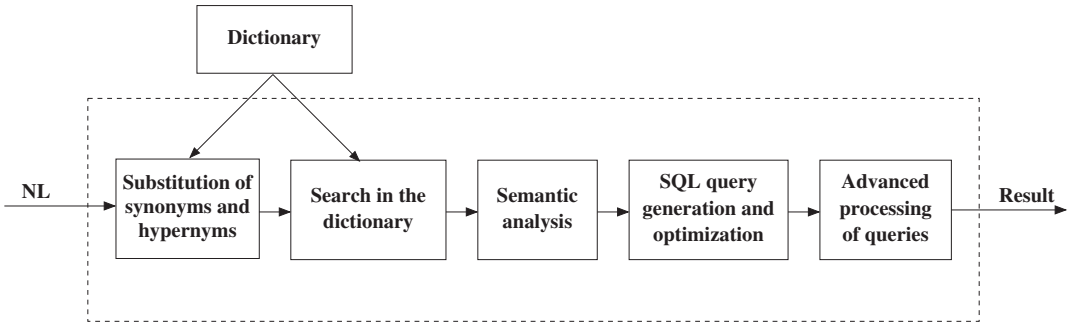

Fig. 1. Main steps of processing of a query in NL.

Table 1. Fragment of the table constructed for transformation of synonyms and hypernyms to logical level terms

	cleavage cycle 11	cleavage cycle 14A	nuclear mask
stage	0.5	0.5	0
4	0.5	0	0
5	0	0.5	0
embryonic	0	0	0.5
syncytium	0	0	0.5

as *'stage 4'* → *'cleavage cycle 11'*, '*stage 5*' → *'cleavage cycle 14A'*, *'embryonic syncytium'* → *'nuclear mask'*, etc. The left part of each rule contains a hypernym or synonym, the right part, a term of the logical level. A special dictionary containing all word forms of synonyms and hypernyms is composed. Synonyms and hypernyms can contain more than one word form (e.g. *stage 4, gene class*), therefore a table is constructed for recognition of these concepts. The table contains conditional probabilities of correspondence of each concept's main word form to the term of logical level. We assume this probability equal to $1/n$, if the concept is formed by n word forms.

Table 1 presents a fragment of the constructed table. The left column of this table contains the dictionary's main word forms, while the terms of the logical level are presented in the first row.

To illustrate the transformation algorithm let us consider the query 'Which embryos belong to stage 5?'. At first the algorithm checks if any word form of the query is in the dictionary. In our example these word forms are *stage* and *5*, and the query contains main word forms of these terms. Next the vector L is formed from main word forms, which components are equal to 1, if the word form was found in the dictionary, and to 0 otherwise. After that the table is transformed to the matrix A. For word forms presented in Table 1 we have $L = [1\ 0\ 1\ 0\ 0]$ and

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0.5 \end{bmatrix}$$

Next the vector $R = L \cdot A$ is calculated with elements equal to the integrated probability of correspondence of the word forms found in the dictionary to the logical level term. In our example there will be $R = [0.5\ 1\ 0]$. The word forms found in the dictionary are transformed to the logical level term if the condition $r = 1$ is satisfied. In our example integrated probability of correspondence of word forms *stage* and *5* to the logical level term *cleavage cycle 14A* is equal to 1, thus the transformation *stage 5* → *cleavage cycle 14A* is performed.

Search in the dictionary. At the 'Search in the dictionary' step all words of a query are compared with those stored in the knowledge domain dictionary. The words with dictionary entries are transformed to the main word form, and each word is associated with a semantic component of the predefined type (e.g. table, field, value, etc.). The initial chain of the semantic components is constructed in this manner.

As an English dictionary we use the dictionary included in the Linux distributive (English Dictionaries...). This dictionary contains main forms of about 16,000 words. The main form of each word is supplied with a set of flags, which define rules of generation of other word forms. Each flag is denoted by a capital letter and defines what affix to add to the root of the word to generate a specific word form. For example, four word forms will be generated from the main form of the word *mask*, which is supplied with flags $DGRS$:
$D : mask + -ed = masked, G : mask + -ing = masking, R : mask + -er = masker, S : mask + -s = masks$.

The algorithm of the word conversion to the main word form works as follows. At the first step it checks if the word given is in the dictionary, and if so, it means that the main word form was used in the query. Otherwise the main word form is constructed. To do so all rules of generation of word forms are applied in reverse order to try to remove affixes specified by these rules. For example, if the word *masked* is used in the user query the application of rules *G*, *R* and *S* would not give result, while the application of the rule *D* (i.e. removal of the affix *ed*) would result in transformation of this word into the corresponding main word form *mask*. Next the algorithm checks if the initial word form could be generated from the main word form by applying the transforming rule. In the example given above the algorithm checks in the dictionary if the word *mask* is supplied with flag *D* and, hence, the word *masked* can be obtained from the word *mask* by addition of affix *ed*.

To construct the Russian dictionary of the knowledge domain we expand the morphological dictionary by Zaliznyak (1977), containing main word forms, by addition of other word forms. These forms were downloaded from the Russian dictionary available at the site http://www.agama.com, which contains about 1 million words.

The semantic analysis. At this step an initial semantic chain is converted into a semantic network, which represents formally a query meaning. The initial semantic chain is a sequence of elementary semantic components, which do not have any subcomponents. The algorithm for semantic analysis performs the sequential grouping of these semantic components into higher level components, and the semantic network of the query is constructed bottom-up.

To group semantic components the mechanism of production rules is applied. Each rule represents the implication *if <applicability condition> then <operator>*. *<Applicability condition>* defines a certain configuration of the semantic component. The *<operator>* defines a sequence of operations, which will be performed if the production rule is applicable.

Each production rule has its own priority. The rules with higher priority are processed earlier, and if two rules have the same priority level, the sequence of processing is undefined.

All semantic components are characterized by type, description and body. A fixed set of attributes corresponds to each type of semantic components. The description of a semantic component defines specific values of its attributes. The most important attribute of the majority of components is their orientation. The orientation specifies the database object (table, field etc.) on which this semantic component is mapped. The body of a semantic component represents a fragment of the resultant query formulated in any formal language, e.g. in SQL.

There are several types of semantic components. The semantic component of the type *table* is directly mapped on the database table, while the component of the type *field* is mapped on the field of the database table. The component of the type *value* represents the value of the table field. Semantic components of the type *relation* are used to define relation operations ($>$, $<$, $=$, etc.) between semantic components of other types. Components of the type *half interval* and *interval* describe one-side and two-side intervals of the field values correspondingly (e.g. > 5, ≤ 5, > 5 and < 8). Components of the type *aggregate* represent main aggregation operations *(COUNT, SUM, MIN, MAX)*, i.e. execution of arithmetic operations on a set of rows of the database table. The semantic components for logical operations are provided as operands *AND, OR*, and *NOT*.

Semantic components of the type *predicate* and *function* are higher level components used to group the other components. *The predicate* represents a condition of selection of rows in a SQL query. *The function* combines the component of the type *aggregate* with the component of the type *table* (e.g. *COUNT(embryo.*)*) or *field* (e.g. *COUNT(embryo.embrid)*).

SQL query generation and optimization. To convert a semantic network into a SQL query the *SELECT* statement is used, given by *SELECT fieldList FROM tableList WHERE condition*, where *fieldList* and *tableList* are lists of the database fields and tables from which the data will be selected and *condition* is the selection criterion.

The algorithm of generation of the SQL query is as follows. The semantic components of types *table*, *field* and *function* represent data requested by user. Thus these components correspond to the *SELECT* clause of the SQL query, while the orientation of initial semantic components defines the data sources and correspond to the *FROM* clause. The semantic component *predicate* describes a selection criterion and corresponds to the *WHERE* clause.

To illustrate the work of our semantic analyzer let us consider an example of a NL query:
'Which embryos are scanned for expression of bcd and belong to late temporal classes?'.
The initial chain of semantic components is written as
embryos (table, embryo, 'embryo') bcd (value, protein.name, 'bicoid') and (logic, -, -) late (halfinterval, embryo.temporal, $\geq$ 5) temporal class (field, embryo.temporal,'embryo.temporal').
Here and below, type, orientation and body of each semantic component are shown in brackets. At the first step the production rule, which groups the components *late* and *temporal class* into the semantic component of a type *predicate*, is applied:
embryos (table, embryo, 'embryo') bcd (value,

protein.name, 'bicoid') and (logic) (predicate, embryo.temporal, embryo.temporal $\geq$ *5).*
Next the component *bcd* is replaced with the predicate:
embryos (table, embryo, 'embryo') (predicate, protein.name, protein.name = 'bicoid') and (logic) (predicate, embryo.temporal, embryo.temporal $\geq$ *5).*
Afterwards, using *AND* operation, two predicates are grouped into the new one:
embryos (table, embryo, 'embryo') (predicate, -, protein.name = 'bicoid' and embryo.temporal $\geq$ *5).*
Finally the resultant semantic chain is obtained, and the SQL query can be generated as follows:
SELECT embryo. FROM embryo, protein, embryochannel WHERE (protein.name = 'bicoid' and embryo.temporal* $\geq$ *5) and (embryo.embrid = embryochannel.embrid and protein.prid = embryochannel.prid).*
The table *embryochannel* joins embryo and protein tables with relation many-to-many, and is included into the SQL query during its generation and optimization, as described below.

The minimization of query execution time is made by applying graph theory methods. The logical scheme is modelled as an undirected, weighted graph, in which nodes represent relational tables and edges correspond to relations between these tables. The weights of edges were obtained experimentally. For each pair of the database tables the *SELECT* query, containing the join of these tables, was executed and the query execution time was estimated. Those joins of tables which take more time will get greater weight values.

The algorithm generates the optimal SQL query by means of construction of a minimal tree for the weighted graph (Figure 2), and can be subdivided into 5 steps:

1. Construction of a matrix of the shortest paths using the Floyd algorithm (1962). The algorithm finds the path of minimum total length between all node pairs in a weighted graph with the arbitrary weights of edges. It results in two matrices P, W of size $|V| \cdot |V|$, where $|V|$ is a graph strength (i.e. number of nodes). The element of the weight matrix W is equal to a length of the shortest path between each pair of nodes, or tends to infinity, if the path between the nodes does not exist. Each element of matrix P defines the shortest path between nodes, e.g. the ij-element is the last but one node in the shortest path between nodes i and j.

2. Search of the shortest path between each pair of nodes, which correspond to database tables requested by a user. All nodes, appearing in these paths, form a set of nodes.

3. Construction of a sub-graph, which contains only the nodes, which appear in the set.

4. Construction of the minimum skeleton tree of the sub-graph. The minimum skeleton tree is the skeleton sub-graph of graph G. The skeleton subgraph of graph $G = \{X, R\}$, where X is a set of nodes and R is a set of edges, is the graph $G' = \{X, R'\}$, which contains the same set of nodes and $R' \subset R$. The minimum skeleton of the sub-graph is obtained by means of the Kruskal algorithm (1956), which is effective enough for graphs with the number of edges only slightly greater than the number of nodes.

5. All terminal nodes, which have only one adjacent edge and were not listed as a requested semantic components, are deleted from the skeleton.

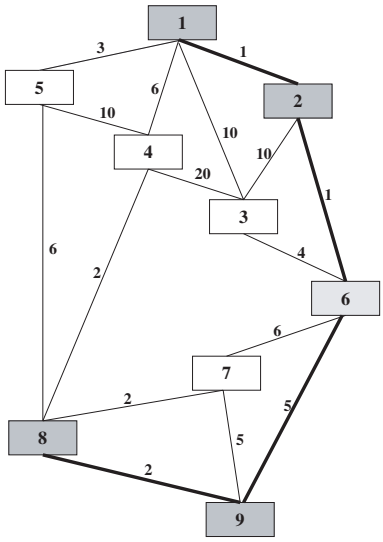

Fig. 2. Construction of the optimal SQL query. Nodes are shown as rectangles. Each edge is supplied with a number equal to its weight. Nodes shown in color (1, 2, 8, 9) correspond to database tables requested by user. The minimal tree which will be used for generation of the SQL query is shown in bold. The table shown as node 6 joins requested tables.

Domain oriented processing of queries. Besides general semantic analysis of NL queries, there is a necessity of advanced processing of queries in the context of the knowledge domain. For example, a user can request to display the data in different views (as a table, graph or image) or process data by a prescribed method. To implement the knowledge domain oriented semantic analysis we provide semantic components of a special type, in which every component is associated with a certain processing procedure. For example, if the words *quantitative data* and *table* are in a query, the corresponding procedure will be activated, which will execute the SQL query generated at previous step and will present the query result in a table form.

Design of the test system based on Microsoft English Query

To estimate the sensitivity of our system to the accuracy of formulation of the NL queries we have developed the test database, using Microsoft SQL Server 7.0, as a fragment of our database FlyEx. It contains three tables: *embryo*, *protein* and *embryochannel*. The fields of the table *embryo* are *embrid* (primary key), *name* (embryo name), *cleavage* (cleavage cycle) and *temporal* (time class). The table *protein* contains two fields *prid* (primary key) and *name* (gene name). The *embryochannel* is an auxiliary table, containing two fields *embrid* and *prid* and joining *embryo* and *protein* tables with a many-to-many relation. This database was populated with information about several genes and embryos. The test system for processing of queries in NL is designed, using Microsoft English Query, and contains entities *embryo* and *gene* and their relationship.

IMPLEMENTATION

Information retrieval

To formulate and execute queries to the database the HTML form 'Natural Language Interface' (Fig. 3) is to be filled in by a user. The text of a query is entered in the text field QUERY. The list QUERY EXAMPLES contains a set of predefined standard queries for convenience. On selection of a query from the list this query is displayed automatically in the field QUERY and may be edited before execution.

By default a query returns all rows which satisfy the selection criterion. The text field MAX. NUMBER OF ROWS RETURNED allows the user to define a number of rows which will be returned as a result of a query. If this number is smaller than the number of rows indicated in the field, all rows will be returned.

By pressing the button SEND QUERY a query will be executed and after a while the result of the query will appear in a new browser window. In the upper part of this window the query in natural language is displayed, in which words used to retrieve the information from the database are shown in red. Below the query result is displayed as a table. The SQL query, automatically generated by the system, is presented below the result. The SQL query can be edited and returned to the server by pressing the button SEND QUERY.

Selection of the link SWITCH TO RUSSIAN calls the Russian version of the query form. The queries in Russian are submitted and executed similarly to queries in English.

The capabilities of the NL processor

To formulate a NL query a user can use any concept described in the conceptual scheme. Both queries which use higher level or lower level concepts are interpreted by the system equally well. For example, the cleavage cycles from 11 to 13 are parts of developmental stage 4 and the query *'What embryos belong to stage 4?'* returns a list of embryos belonging to these cycles.

To formulate a NL query a user may type the words in any word form (for example, *embryo* or *embryos*, *gene* or *genes*, *mask* or *masked*). The query can be formulated both as a whole phrase or as a list of keywords. For instance, the query *'embryos Kr gt eve'* will return the same result as the query *'Which embryos were scanned for expression of Kruppel, giant and even-skipped?'*

The queries can be formulated using synonyms or even laboratory jargon. For example, a gene name can be introduced both as a full name or a symbol, or even as abbreviation in the form of three capital letters, each of which corresponds to the common notation of the gene (e.g. *'BHE'* - *bcd*, *gt* and *eve*).

The query *'How many ...?'* allows a count of rows satisfying any criterion (e.g. *'How many embryos are scanned for expression of bcd and belong to late temporal classes?'* Besides, it is possible to list these rows on a screen (for example, *'How many genes were scanned for expression in embryo tn1? List these genes.'*)

One of the important features of the system consists in a possibility of arbitrary assignment of conditions posed on values of numerical attributes. The following combinations of semantic constructions are supported: *larger than*, *greater than*, *more than*, $>$, $\geq$, $\leq$, $<$, *less than*, *smaller than*, *from n to m*, *n - m*.

It is possible to combine selection criteria in a query using logical operators *AND, OR, NOT*. For example, the query *'Which embryos were scanned for expression of Kruppel and giant and even-skipped?'* returns a list of embryos, which were scanned for expression of all these genes, while the query *'Which embryos were scanned for expression of Kruppel or giant or even-skipped?'* returns a list of embryos scanned for expression of at least one of these genes.

The query *'Display pattern ...'* returns a pattern of segmentation gene expression. When several patterns in different embryos are requested, an embryo list is displayed, in which embryo names are linked to embryo images. If a pattern in one embryo is requested, it will be displayed at once, instead of the embryo list.

Quantitative and processed expression data can be displayed to a user in different formats - as a table, flat graph or 3D graph. To retrieve this information a user has to specify a desired format of a query , e.g. *'Select as a flat graph a quantitative data on expression of Kruppel in embryos belonging to temporal class 3'*.

Use of conceptual scheme

Conceptual scheme as a guide The conceptual scheme of the information on expression of segmentation genes

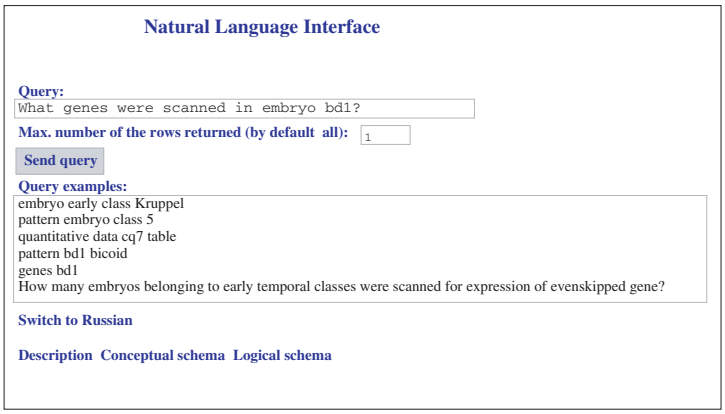

Fig. 3. The HTML form 'Natural Language Interface' for input of NL query.

in *D.melanogaster* presents graphically basic concepts in the domain, as well as relations among them. Due to the large size of the scheme, the semantically related concepts are grouped into eight blocks: TIME, ANATOMY TERM, GENE, ACQUISITION METHOD, GENE EXPRESSION PATTERNS, QUANTITATIVE GENE EXPRESSION DATA, PROCESSED DATA and INTEGRATED DATA. Blocks and concepts are shown as rectangles in different colors, while relations among concepts are displayed as arrows. Each arrow is supplied with a label which explains the nature of the relation.

A scientist can use the conceptual scheme as a guide to clarify the structure of information and the meaning of terms. At first a user is presented with general view of the scheme, which displays blocks of concepts. A given block may be selected by a click with a right mouse button. Selection of a block displays relations between its concepts, as well as relations of these concepts with concepts from other blocks. The later are displayed in different colors. Selection of a concept from another block displays relations between its concepts in a new browser window. A user may return to the general scheme by pressing the MAIN button.

If a concept was selected by clicking with a left mouse button, while pressing the Shift key, a new window comes into view which contains a definition of the term and a query form with a list of predefined NL queries to the database. These queries can be used to retrieve additional information from the database, which will refine the term meaning and make completely explicit the structure and the content of the information.

Interactive formulation of queries to the database. Since all terms of the conceptual scheme are mapped onto the logical one, the conceptual scheme can be used for visual construction of queries to the database. To do so a user selects concepts of interest by double clicking with the left mouse button. Both concepts from one or different blocks can be selected. Upon selection a user submits a query by pressing SEND QUERY button.

'Какие эмбрионы сканировали для выявления экспрессии Kruppel?'

Fig. 4. This query, which is written in Russian, is semantically equivalent to the query 'Which embryos were scanned for expression of Kruppel?'.

Estimation of the sensitivity of the NL processor to accuracy in the formulation of NL queries

To estimate the sensitivity of the developed system to the accuracy in formulation of NL queries we have performed a comparison with the test system, designed on the base of Microsoft English Query. We have tested the interpretation of the query *'Which embryos were scanned for expression of Kruppel?'*, and both systems translated it correctly. However, Microsoft English Query was unable to interpret even slightly modified text of the query as *'Which embryos were scanned for Kruppel?'*, since this query syntax is not equal to the test query. In our system both queries gave identical results as these queries are equivalent semantically. Moreover, in our system it is possible to formulate additional semantically equivalent versions of the query, for example, as a list of keywords *'embryo Kruppel'*, or in Russian (Fig. 4), or in any other language.

CONCLUSIONS

In this paper we present an approach to query a relational database in a natural language. Our method is based on the semantic approach and thus does not require specification of all possible language constructions as well as a standard

grammar accuracy in formulation of NL queries. It allows the user to change arbitrarily the order of words in a sentence, to omit insignificant words and to formulate the query as a complete phrase or as a list of keywords. It may be underlined that our method can interpret and process queries in various languages simultaneously.

The information retrieval from the database proceeds by submission a single phrase in NL instead of tedious selection of many menu elements. Substitution of many form fields with the single field for input of a query in NL simplifies mastering of a database. It also increases the efficiency of access, because the task of understanding the user's informational needs is now carried out by the NLP system.

The distinctive feature of the method is in the use of a conceptual scheme of the subject domain as a connecting link between the text of a query and the database scheme. The conceptual scheme assists in processing of queries in natural language. It helps the NL processor to interpret hypernyms and synonyms, as well as equivocal and jargon terms. Moreover, making specifications of domain knowledge explicit, the conceptual scheme guides a user to learn the meaning of each term.

Other important feature of the method for processing queries in NL developed here consists in its flexibility and adaptivity with respect to changes in the knowledge domain and user's views, because any change in the domain knowledge can be easily introduced into the conceptual scheme and subject domain dictionary.

ACKNOWLEDGEMENTS

This work was supported by NIH grant RO1 RR07801, EC grant IST-1999-11009 and GAP awards RBO-1286 and RMO-1323. The authors are thankful to Alexander M. Samsonov for valuable discussions and comments, and Ekaterina Poustelnikova for help in conceptual scheme design.

REFERENCES

Akam,M. (1987) The molecular basis for metameric pattern in the *Drosophila* embryo. *Development*, **101**, 1–22.

English Dictionaries http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html#English-dicts.

Floyd,R.W. (1962) Algorithm 97: shortest path. *Comun. A.C.M.*, 5,6,345.

Foe,V.A. and Alberts,B.M. (1983) Studies of nuclear and cytoplasmic behaviour during the five mitotic cycles that precede gastrulation in *Drosophila* embryogenesis. *Journal of Cell Science*, **61**, 31–70.

Ingham,P.W. (1988) The molecular genetics of embryonic pattern formation in *Drosophila*. *Nature*, **335**, 25–34.

Khoros Pro 2001 Integrated Development Environment (IDE) (2001) http://www.khoros.com.

Kozlov,K., Myasnikova,E., Samsonova,M., Reinitz,J. and Kosman,D. (2000) Method for spatial registration of the expression patterns of *Drosophila* segmentation genes using wavelets. *Computational Technologies*, **5**, 112–119.

Kozlov,K., Myasnikova,E., Pisarev,A., Samsonova,M. and Reinitz,J. (2002) A method for two-dimensional registration and construction of the two-dimensional atlas of gene expression patterns *in situ*. *In Silico Biol.*, **2**, 125–141. http://www.bioinfo.de/isb/2002/02/0011.

Kruskal,Jr,J.B. (1956) On the shortest spanning subtree of a graph and travelling salesman problem. *Proc. Amer. Math. Soc.*, **7**, 48–50.

Merrill,P.T., Sweeton,D. and Wieschaus,E. (1988) Requirements for autosomal gene activity during precellular stages of *Drosophila melanogaster*. *Development*, **104**, 495–509.

Developing with Microsoft English Query in Microsoft SQL Server 7.0 (1998) Microsoft Corporation, http://www.microsoft.com/sql/techinfo/development/70/developingeq.asp.

Myasnikova,E., Kosman,D., Reinitz,J. and Samsonova,M. (1999) Spatio-temporal registration of the expression patterns of *Drosophila* segmentation genes. In *Proceeding of the Seventh International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 195–201.

Myasnikova,E., Samsonova,A., Kozlov,K., Samsonova,M. and Reinitz,J. (2001) Registration of the expression patterns of *Drosophila* segmentation genes by two independent methods. *Bioinformatics*, **17(1)**, 3–12.

Myasnikova,E., Samsonova,A., Samsonova,M. and Reinitz,J. (2002) Support vector regression applied to the determination of the developmental age of a *Drosophila* embryo from its segmentation gene expression patterns. *Bioinformatics*, **18 (Suppl. 1)**, s87–s95.

Nusslein-Volhard,C., Wieschaus,E. and Kluding,H. (1984) Mutations affecting the pattern of the larval cuticle in *Drosophila melanogaster* I. *Zygotic loci* on the second chromosome. *Roux's Archives of Developmental Biology*, **193**, 267–282.

Narinyany,A.S. (1979) Linguistic processors of West Siberia. *Preprint N 199, Computing center of Siberian Branch of Russian Academy of Sciences (in Russian), Novosibirsk, Russia*.

Samsonova,M., Pustel nikova,E., Pisarev,A. and Reinitz,J. (2000) Design of the integrated atlas of segmentation gene expression *in situ*. In *Proceedings of the German Conference on Bioinformatics 5–7 Oct., Heidelberg*. pp. 125–132.

Wieschaus,E., Nusslein-Volhard,C. and Jurgens,G. (1984) Mutations affecting the pattern of the larval cuticle in *Drosophila melanogaster*. *111 Zygotic loci on the X-chromosome and fourth chromosome*, Roux's Archives of Developmental Biology, **193**, pp. 296–307.

Zaliznyak,A.A. (1977) Russian grammar dictionary. Russky yazyk Moscow.

Zhigalov,V.A. (1998) On experience of a design of system for construction of natural language interfaces to databases. In *Proceedings of International Workshop on Computer Linguistic and its Applications Dialog' 98 (in Russian)*. pp. 801–808.