International Journal of Computer Applications Technology and Research
Volume 3– Issue 10, 648 - 653, 2014, ISSN: 2319–8656

DNA Query Language DNAQL: A Novel Approach

Adekunle Y.A
Department of Computer Science
Babcock University,
Ilishan-Remo, Ogun State
Nigeria

Ogunwobi Z.
Department of Mathematical Sciences,
Olabisi Onabanjo University,
Ago-Iwoye, Ogun State,
Nigeria

Alao O.D
Department of Computer Science
Babcock University,
Ilishan-Remo, Ogun State
Nigeria

Efuwape B.T
Department of Mathematical Sciences,
Olabisi Onabanjo University,
Ago-Iwoye, Ogun State,
Nigeria

Ebiesuwa Seun
Department of Computer Science
Babcock University,
Ilishan-Remo, Ogun State
Nigeria

Ainam Jean-Paul
Department of Computer Science
Babcock University,
Ilishan-Remo, Ogun State
Nigeria

Abstract: This paper produces a DNA Query language for querying DNA database in an expressive and concise manner.One of the objectives of this paper was to demonstrate how such a research will be beneficial to biochemistry researchers who are unfamiliar with SQL coding. The paper introduces a new layer between the user application which serves as the interface and the database layer. This layer is then responsible of translating a familiar biochemistry language into a SQL code understandable by the database application. By doing so, the abstraction layer is what is needed to alleviate the use of DNA database by any researcher. Also, a description of common tasks and how they can be translated is given in this paper. Therefore, the novel approach consists of keeping the standard SQL language at the database layer, and yet supplies the same advantages.

Keywords: DNA computing, query language, DNA query language, protein query language and bioinformatics.

1. INTRODUCTION

A Deoxyribonucleic acid (DNA) computer is primarily a nano-computer that makes use of deoxyribonucleic acid to carry out calculations [1].

As a matter of fact, the DNA computers are the next generation microprocessors that make use of the DNA, molecular biology and chemistry instead of the conventional silicon based technologies. DNA computing is a fast developing inter-disciplinary area. The DNA molecules that make up our genes have the ability to perform calculations faster than the world's fastest man-made computers. It has been predicted that DNA might one day be integrated into a computer chip to produce a so-called biochip that will have the capability of pushing the computer even faster.

DNA molecules have already been harnessed to perform complex mathematical calculations. Even while in infancy DNA computers can store billions of time more data than the conventional computers.

A major challenge facing biochemistry and biology researchers is the ability to view relationships among protein data, functions, structures and pathways in a single query or at least in a concise way [2]. For instance, biochemists are performing cutting edge research into carbon-donated hydrogen bonds and their effect on protein structures [3]. In order to do this successfully they need data at the atomic level of the protein to perform calculations. However, no online database is known to exist that supplies experimental data in an easy-to-use format at the atomic level without parsing the data manually nor do tools exist to facilitate the calculations once data is parsed. To bolster their research chemists have been downloading files from the RCSB in Protein Data Bank (.pdb) format, parsing data manually and loading data into spreadsheets to perform calculations. This approach is wearying and potentially error prone and spreadsheet limitations and other limiting factors obviate the need for a more efficient solution.

2. BACKGROUND

A Deoxyribonucleic acid (DNA) computer is primarily a nano-computer that makes use of deoxyribonucleic acid to carry out calculations [1].

As a matter of fact, the DNA computers are the next generation microprocessors that make use of the DNA, molecular biology and chemistry instead of the conventional silicon based technologies. DNA computing is a fast developing inter-disciplinary area. The DNA molecules that make up our genes have the ability to perform calculations faster than the world's fastest man-made computers. It has been predicted that DNA might one day be integrated into a computer chip to produce a so-called biochip that will have the capability of pushing the computer even faster.

DNA molecules have already been harnessed to perform complex mathematical calculations. Even while in infancy

DNA computers can store billions of time more data than the conventional computers.

A major challenge facing biochemistry and biology researchers is the ability to view relationships among protein data, functions, structures and pathways in a single query or at least in a concise way [2]. For instance, biochemists are performing cutting edge research into carbon-donated hydrogen bonds and their effect on protein structures [3]. In order to do this successfully they need data at the atomic level of the protein to perform calculations. However, no online database is known to exist that supplies experimental data in an easy-to-use format at the atomic level without parsing the data manually nor do tools exist to facilitate the calculations once data is parsed. To bolster their research chemists have been downloading files from the RCSB in Protein Data Bank (.pdb) format, parsing data manually and loading data into spreadsheets to perform calculations. This approach is wearying and potentially error prone and spreadsheet limitations and other limiting factors obviate the need for a more efficient solution

3. RELATED WORKS

I-Min A. et al, worked on Advanced Query Mechanisms for Biological Databases. In their paper, they describe generic tools that provide powerful and flexible support for interactively exploring biological database in a uniform and consistent way that is via common data models, formats and notation in the framework of the Object-Protocol Model (OPM). These tools avoid the restriction imposed by traditional fixed-form query interfaces, while providing users with simple and intuitive facilities [4]

Peter Buneman et al [5], worked on Beyond XML Query Languages where they described challenges for Query Languages, the constraints and optimization. In brief, the paper focused on describing the semantics of the underlying semi structured data model, the basic operations on data, the interaction of these operations with constraints, the nature of updates and the problems of generating XML efficiently from existing sources.

Robert Brijder et al, [6] worked on DNA computing and presented a querying language for databases in DNA. In their paper, a set of formal operations on DNA complexes has been defined, much in the spirit of the operations of the relational algebra in the relational data model. The combination of these operation leads to the query language DNAQL.

Sheriff Elfayoumy et al, [7] introduces a Protein Query Language (PQL) for querying protein structures in an expressive yet concise manner, utilizing the work of Patel [8] and Garcia [9]. One of the objectives of their work was to demonstrate how such a language would be beneficial to protein researchers to obtain in-depth protein data from relational database without extensive SQL knowledge. The PQL was an attempt to provide an intuitive declarative language within query application to researchers who are unfamiliar with SQL coding, but their approach still remains primitive and largely procedural. This limits the ease with which complex queries can be posed and often results in very inefficient query plans.

Sandeep Tata et al, [10] introduced a system called Periscope/SQ based on a well-defined extension of relational algebra – a system that permits declarative and querying on biological sequences. Finally, using a real-world application in eye genetics, they showed how Periscope/SQ can be used to achieve a speedup of two orders of magnitude over existing procedural methods.

Eltabakh et al discussed an extensible database engine for biological databases [12] The proposed engine “extends the functionalities of current DBMSs with (1) annotation and provenance management including storage, indexing, manipulation, and querying of annotation and provenance as first class object in DBMS, (2) local dependency tracking to track the dependencies and derivations among data items, (3) update authorization to support data curation via content-based authorization, in contrast to identity-based authorization, and (4) new access methods and their supporting operators that support pattern matching on various types of compressed biological data type”.

There is a growing and urgent need for declarative and efficient methods for querying biological sequences data [10].

4. METHODOLOGY

4.1 DNA SEQUENCING THEORY

The method of DNA sequencing that will be used in this paper is the enzymatic method which is usually referred to as dideoxy or chain termination sequencing. In this method a short oligonucleotide primer is hybridized to the DNA template that is to be sequenced (Fig.1). A DNA polymerase is then used to initiate DNA synthesis extending from the primer in the 5’ to 3’ direction. The synthesized DNA is complementary to the template strand of DNA. The reaction contains deoxynucleotides (dNTPs: dATP, dCTP, dGTP, TTP) used by the polymerase to extend the chain. However the reaction also contains a small quantity of dideoxynucleotides (ddNTPs) (Fig.1).

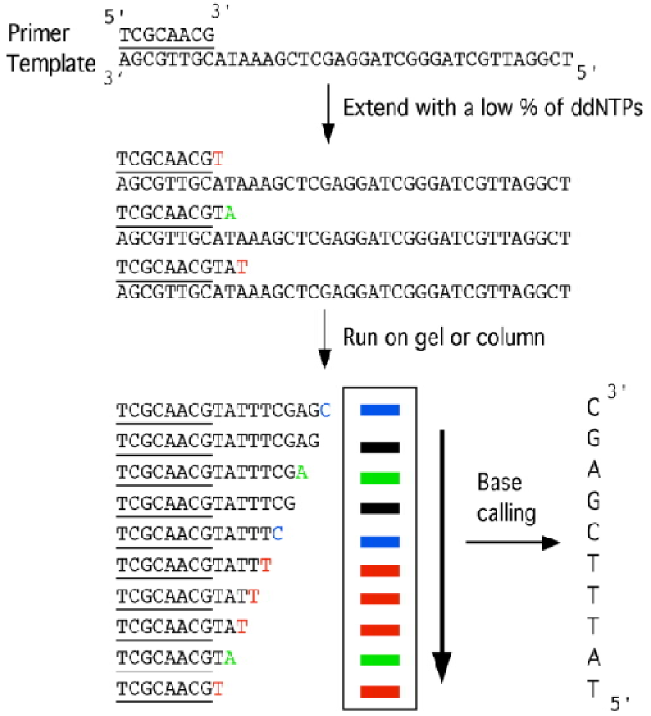

Figure 1: Diagram of DNA sequencing using the chain termination method (Source: Adams G., 2010).

4.2 COMPUTER AS A TOOL FOR DNA SEQUENCE ANALYSIS

In recent time there has been explosion in the number of DNA sequence that has been determined. It is intriguing to know that as DNA sequence information is generated, a problem with the storage and analysis of the vast amounts of information becomes inevitable. This kind of problem is particularly suited to computers. Computers thus serve as tools for handling vast amounts of sequence information produced by molecular biologists.

Computers do not just store sequence information but programs have been written that analyze DNA. For instance, it is important to know where the protein coding sequences are situated on a DNA fragment, what convenient restriction enzyme sites are present in the DNA fragment amongst others. Computers can also be used to determine the similarities between an unknown DNA or protein sequence and a known DNA or protein sequence in the databases.

DNA sequence analysis using computers include searching databases and sequencing databases.

An example of a database entry (dna sequence):

LOCUS AB231879 1383 bp mRNA linear INV 07-JUN-2006
DEFINITION Artemia franciscana mRNA for zinc finger protein Af-Zic, complete cds.
ACCESSION AB231879
VERSION AB231879.1 GI:94966317
KEYWORDS .
SOURCE Artemia franciscana
ORGANISM Artemia franciscana
Eukaryota; Metazoa; Arthropoda; Crustacea; Branchiopoda; Anostraca;
Artemiidae; Artemia.
REFERENCE 1
AUTHORS Aruga,J., Kamiya,A., Takahashi,H., Fujimi,T.J., Shimizu,Y.,
Ohkawa,K., Yazawa,S., Umesono,Y., Noguchi,H., Shimizu,T.,
Saitou,N., Mikoshiba,K., Sakaki,Y., Agata,K. and Toyoda,A.
TITLE A wide-range phylogenetic analysis of Zic proteins: Implications
for correlations between protein structure conservation and body
plan complexity
JOURNAL Genomics 87 (6), 783-792 (2006)
PUBMED 16574373
REFERENCE 2 (bases 1 to 1383)
AUTHORS Aruga,J. and Toyoda,A.
TITLE Direct Submission
JOURNAL Submitted (10-AUG-2005) Jun Aruga, RIKEN Brain Science Institute,
Laboratory for Comparative Neurogenesis; 2-1 Hirosawa, Wako-shi,
Saitama 351-0198, Japan (E-mail:jaruga@brain.riken.jp,
URL:http://www.brain.riken.go.jp/labs/lcn/, Tel:81-48-467-9791,
Fax:81-48-467-9792)
FEATURES Location/Qualifiers
source 1..1383
/organism="Artemia franciscana"
/mol_type="mRNA"
/db_xref="taxon:6661"
gene 1..1383
/gene="Af-Zic"
CDS 1..1383
/gene="Af-Zic"
/codon_start=1
/product="zinc finger protein Af-Zic"
/protein_id="BAE94140.1"
/db_xref="GI:94966318"
/translation="MTASLSASVMNPSFIKRESPASATALFVPN
QFSAVPNFGFHHVP
SACATEQSSEMLNPFVDNHLRLNDQSNFQGYHHPHHG
QIQQHHLGSYAARDFLFRRDM
GLGMGLEAHHTHAAQHHHMFDPSHAAAAAHHAMFT
GFDHNTMRLPTEMYTRDASAA
QQFHQMGSMAPMAHPASAGAFLRYMRTPIKQELHCL
WVDPEQPSPKKTCGKTFGSMHE
IVTHITVEHVGGPECTNHACFWQGCVRNGRAFKAKYK
LVNHIRVHTGEKPFPCPFPGC
GKVFARSENLKIHKRTHTGEKPFKCEFEGCDRRFANSS
DRKKHSHVHTSDKPYNCKVR
GCDKSYTHPSSLRKHMKVHGKSPPPASSGCDSDENESI
ADTNSDSAASPSPSSHDSSQ
VQVNHNRPPNHHNLGLGFTNPGHIGDWYVHQSAPDM
PVPPATEHSPIGPPMHHPPNSL
NYFKTELVQN"
ORIGIN
1 atgactgcta gtttaagtgc aagcgtgatg aatccaagtt ttataaagag ggaaagtcct
61 gcatcggcta cagccctgtt cgtaccaaac caatttagtg cagtgcctaa ttttggattt

121 caccatgttc ctagtgcttg tgcaactgag caaagtagtg aaatgctgaa cccttttgtg

4.3 DATABASES

Available databases that can be sequenced are classified into general and organismal specific databases.

General Databases:

- GenBank: DNA sequences (USA database)
- EMBL: DNA sequences (European Molecular Biology Laboratory)
- GenEMBL GenBank and EMBL sequences combined
- DDBJ: DNA sequences (Japan's equivalent of Genbank)
- EST: Expressed Sequence Tags (USA) (DNA sequences)
- STS: Sequence Tagged Sites (USA) (DNA sequences)
- PIR: Protein Identification Resource (protein sequences)
- SwissProt: Protein sequences (Switzerland and EMBL)
- Genpept: Translations of DNA based on authors' information
- PDB: Coordinates for protein 3D structure.

Organismal Specific Databases:

- SGD: Saccharomyces Genomic Database
- YPD: Yeast Protein Database
- WPD: Worm Protein Database
- Wormbase: C. elegans Genome Database
- Sanger: Worm sequence and genomic database
- Flybase: Drosophila sequence and genetic database
- Human: Many

In this paper the GenBank database will be used. The GenBank sequence database is an open access, annotated collection of all publicly available nucleotide sequences and their protein translations. This database is produced and maintained by the National Center for Biotechnology Information (NCBI) as part of the International Nucleotide Sequence Database Collaboration (INSDC). The National Center for Biotechnology Information is a part of the National Institutes of Health in the United States. GenBank and its collaborators receive sequences produced in laboratories throughout the world from more than 100,000 distinct organisms. In the more than 30 years since its establishment, GenBank has become the most important and most influential database for research in almost all biological fields, whose data are accessed and cited by millions of researchers around the world. GenBank continues to grow at an exponential rate, doubling every 18 months. Release 194, produced in February 2013, contained over 150 billion nucleotide bases in more than 162 million sequences. GenBank is built by direct submissions from individual laboratories, as well as from bulk submissions from large-scale sequencing centers.

4.4 RELATIONAL SCHEMA FOR GenBank

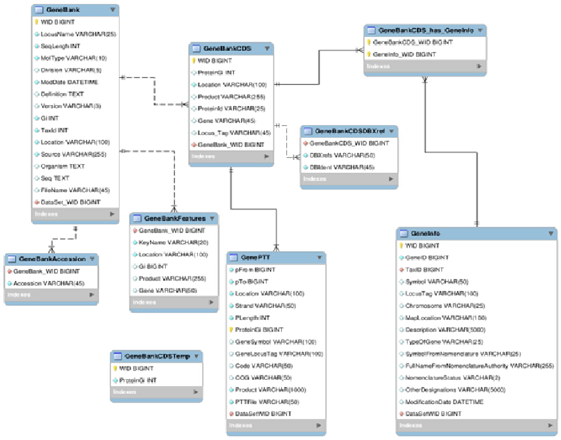

Figure 2: A Framework for Biological Data Integration (Source: JBioWH, 2013)

4.5 DATASET REPOSITORIES

Research chemists around the world do have access to various public DNA data sources, but the access is not designed to support processing and retrieval at the atomic level. Online 'database' supporting biochemistry research include Genbank, EMBL Data Library in the UK, the DNA Bank of Japan (DDBJ), and COLUMBIA [11]. In essence, the only known public access to these databases is via a supplied front-end, and the returned data is formatted for user reading than for storing the data into a databases for further processing and analysis.

5. DNA QUERY LANGUAGE: THE NEW APPROACH

The new approach proposed on this paper is declarative in nature. The new approach provides users with the following feature:

- Users may utilize familiar terms when referring to DNA models and other chemistry terms. The underlying relational model is abstracted from user.
- The ability to use mathematical, Boolean and string functions as part of the language. However, constructs such as conditionals and loping are supported at this time.
- The user shall be able to save DNASQL construct for later utilization.

The approach proposed in this paper consists of defining a layer between the user interface and the DNA database. DNASQL: new approach is an attempt to provide an intuitive declarative language within query application to researchers who are unfamiliar with SQL coding. Biochemists can use their familiar language and terms within biochemistry domain to query the database. The idea behind is then to define an abstraction layer in charge of translating the familiar biological language in a language comprehensive by the underlying layer; that is the application layer.

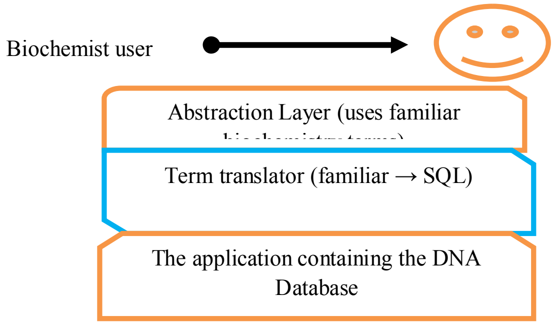
Biochemist user
Abstraction Layer (uses familiar
Term translator (familiar → SQL)
The application containing the DNA Database

Figure 3: The new approach representation

This paper is concerns about layer 2: the layer between the abstraction layer and the database layer. The abstraction layer provides common tools to the biochemist researcher as simple text editor to input the query while the database layer contains the database itself.

Before describing the SQL language proposed in this paper, let us first describe the following requirement and definitions:

Definition 1: the value of a nucleotide can be obtain by giving its position and is defined as:

Strand (i) ***where 0 ≤ i < n with n the length of the DNA strand.***

Definition 2: Get*Strand (char) where char belongs to alphabet {A, C, T, G}*. The function returns an array of indices where '*char'* is found. E.g. for a given strand ACTCAGTA, GetStrand(A) will return Array{0, 4, 7}.

Definition 3: To obtain a value of nucleotide in a given position, we make use of variable within the function. A variable is defined by preceding the name of the variable by '?'. The function is as follow GetStrand(?x, i) and does do the same task as GetStrand(i) function but adds the possibility of getting the variable for further usage within the same query. **GetStrand(?x, i)** where x refers to the variable and i the position of the nucleotide within the strand.

Definition 4: To set a nucleotide value in a given position, user can simply makes use of setStrand(x, i) where x is the value to be inserted and i the position within the strand. This function can easily be used within a update query in a standard SQL language.

Given a table containing record and IDEntry as a primary, the next paragraph describes common tasks used and then translated by the midst layer.

5.1 UPDATE/DELETE/INSERT QUERY

The update statement is used to change **existing** records in a table while delete statement removes one or more records from a table and insert statement adds records to any single table. Either all the rows can be updated, deleted or inserted, or a subset may be chosen using a condition. For the proposed approach, a simple update can be defined as:

Modify table1
SetStrand(A, 2)
Where IDEntry = 1

The SQL equivalent of this query may be:
UPDATE table1
SET "*get the nucleotide at the second position and then set the value at "= A*
Where IDEntry = 1.

In this example, the task of getting the nucleotide at the second position may involve another query resulting in a nested query.

Same as the update statement, a delete statement can be expressed as:

Delete from table1
Where IDEntry = 1
And a insert as

Insert into table1:
Set setStrand(i)

SELECT QUERY

The select statement returns a result set of records from one or more tables or in other words a select retrieves zero or more rows from one or more database tables or database views. In the context of our proposed model, a select stamen is defined as:

Retrieve from table1
getStrand(?x, i)
where ?x = A

6. CONTRIBUTION TO KNOWLEDGE

This paper introduces a layer between the user and the GenBank database. Unlike the conventional approach where the user accesses the database using Structured Query Language (SQL), this approach proposed in this paper is entirely new and different from all other existing works because it permits the user to make use of the language He or She is well acquainted with and the introduced layer now translates this language into the equivalent SQL query thus carrying out the instruction of the user.

7. SUGGESTION FOR FURTHER RESEARCH

In the future, another researcher can go ahead to implement the proposed approach discussed in this paper which allows the user to use any language He or She is familiar with and this user instruction is now translated into the equivalent SQL query by the introduced layer.

8. CONCLUSION

This paper discussed DNA query language making use of a novel approach. The paper aimed at addressing the problem often times faced by people who want to access databases but lack the requisite understanding of the fundamental SQL which is the language GenBank (the database used in this paper) is based on. The proposed approach makes it possible for users with little or no knowledge of the SQL to issue instructions using any language they are familiar with and these instructions get converted into SQL query by the layer introduced in this paper. Hence, access to databases becomes easier, less strenuous and convenient.

9. REFERENCES

[1]. http://www.webopedia.com/TERM/D/DNA_computer.html Retrieved On Feb 5th, 2010.

[2]. M.I. Jaya, Z. Zainol, and N.H. Malim (2007) "iProt – A Data Warehouse for Protein Database"; International Conference on Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia.

[3]. K. Compaan, R. Vergenz, P. Von Rague Schleyer, and I. Arreguin (2008) "Carbon-donated Hydrogen Bonding: Electrostatics, Frequency Shifts, Directionality, and Bifurcation"; International Journal of Quantum Chemistry, Vol. 108, No. 15, 2914–2923.

[4]. I-Min A. Chen, Anthony S. Kosky, Victor M. Markowitz, Ernest Szeto, and Thodoros Topaloglou, "Advanced Query Mechanisms for Biological Databases", Bioinformatics Systems Division, Gene Logic Inc., 2001.

[5] Peter Buneman, Alin Deutsch, Wenfei Fan, Hartmut Liefke, Arnadu Sahuguet, Wang-chiew Tan, "Beyond XML Query Languages", University of Pennsylvania, November 1998.

[6] Sherif Elfayoumy and Paul Bathen, "Protein Query Language: A Novel Approach", School of Computing, University of North Florida, Jacksonville, FL, USA.

[7] J.M. Patel, D.P. Huddler, and L. Hammel. "Declarative and Efficient Querying on Protein Secondary Structures"; Data Mining in Bioinformatics, pp. 243 – 273, 2005.

[8]M.M. Roldan-Garcia, J.J. Molina-Castro, and J.F. Aldana-Montes. 'ECQ: A Simple Query Language for Semantic Web"; Processing of DEXA, Turin, Italy, 2008.

[9] Sandeep Tata, J.M. Patel, J.S. Friedman, A. Swaroop, "Declarative Querying for Biological Sequence Databases", appears in IEEE ICDE 2006.

[10] S. Trissl, K. Rother, H. Mueller, T. Steinke, I. Koch, R. Preissner, C. Froemmel, and U. Leser. "Columba: an Integrated Database of Proteins, Structures, and Annotations"; BMC Bioinformatics, Vol. 8, No. 81, 2005.

[11]M.Y. Eltabakh, M. Ouzzani, M. Aref, "BDBMS- A Database Management System for Biological Data"; Third Biennial Conference on Innovative Data Systems Research (CIDR), 2007.

[12] M.Y. Eltabakh, M. Ouzzani, M. Aref, A.K. Elmagarmid, Y. Laura-Silva, M.U. Arshad, D. Salt, and I. Baxter. "Managing Biological Data Using BDBMS"; The IEEE 24th International Conference on Data Engineering, 200.