Central European Science Journals

w w w . c e s j . c o m

Central European Journal of Biology

Central European Science Journals

DOI: 10.2478/s11535-006-0003-9

Research article

CEJB 1(1) 2006 29–42

Search for proteins with similarity to the CFTR R domain using an optimized RDBMS solution, mBioSQL

Tamás Hegedűs*, John R. Riordan†

Department of Biochemistry & Biophysics and Cystic Fibrosis T&R Center, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599, USA

Received 13 October 2005; accepted 10 January 2006

Abstract: The cystic fibrosis transmembrane conductance regulator (CFTR) comprises ATP binding and transmembrane domains, and a unique regulatory (R) domain not found in other ATP binding cassette proteins. Phosphorylation of the R domain at different sites by PKA and PKC is obligatory for the chloride channel function of CFTR. Sequence similarity searches on the R domain were uninformative. Furthermore, R domains from different species show low sequence similarity. Since these R domains resemble each other only in the location of the phosphorylation sites, we generated different R domain patterns masking amino acids between these sites. Because of the high number of the generated patterns we expected a large number of matches from the UniProt database. Therefore, a relational database management system (RDBMS) was set up to handle the results. During the software development our system grew into a general package which we term Modular BioSQL (mBioSQL). It has higher performance than other solutions and presents a generalized method for the storage of biological result-sets in RDBMS allowing convenient further analysis. Application of this approach revealed that the R domain phosphorylation pattern is most similar to those in nuclear proteins, including transcription and splicing factors.

Keywords: Cystic fibrosis, CFTR, regulatory domain, phosphorylation, relational database management system

* E-mail: hegedus@med.unc.edu, Corresponding autor

† E-mail: john_riordan@med.unc.edu

1 Introduction

The cystic fibrosis transmembrane conductance regulator (CFTR) is unique among the ABC (ATP binding cassette) proteins as an ion channel [1, 2]. This function necessitates an extremely tight control of the opening of its passive permeability pathway. This is provided by a central domain not present in other ABC proteins, termed the R domain. Under normal conditions, several sites within this domain must be phosphorylated by PKA for activation [3–5]. Removal of these phosphoryl groups by serine protein phosphatases rapidly terminates gating. Although it has been suggested previously that this phosphorylation promotes ATP hydrolysis by the nucleotide binding domains of CFTR [6, 7], this seems not to be the case. Although direct evidence is not yet available, it now seems more likely that R domain phosphorylation enables conformational coupling between the impact of ATP binding at two sites formed by the nucleotide binding domains and the channel gate [8]. There is little insight into the mechanism whereby this occurs beyond the fact that the introduction of extra negative charges plays a role [9] and promotes interaction of the R domain with other parts of the protein [10]. The R domain has a low proportion of defined secondary structural elements and its sequence is much less conserved among species than that of the rest of CFTR [11–13]. Common tools for sequence similarity searches, like blast and querying protein databases with a matrix profile built from the R domains, have not returned proteins with significant similarity to this domain ([11] and unpublished results).

The most conserved and distinguishing feature of the R domain is the positioning of ∼9 consensus sites for phosphorylation by PKA. The spacing of these sites in the primary structure is identical or very similar among the species with known CFTR sequence [11]. Therefore, we have utilized this conserved pattern of sites to attempt to identify other R domain-related proteins which might provide additional clues to its mechanism of action. To facilitate this approach with a large number of patterns, we set up a relational database management system (RDBMS). During the software development, the system grew into a general package which we term Modular BioSQL (mBioSQL) which performs favorably compared to available systems. It required some optimization to increase database performance, and now profits from its ability to load the analysis results back to the RDBMS for further analysis by the Structured Query Language (SQL). Based on the application of our approach the R domain phosphorylation pattern shows similarity to those in nuclear proteins, including transcription and splicing factors.

2 System and methods

The local warehouse was implemented on a DELL Precision Workstation with Pentium 4 (3.00 GHz), 1 GB RAM, and 80 GB IDE hard drive (7200 rpm). The operating system was RedHat Enterprise Linux WS 3. Python 2.3.3 version was used with BioPython 1.30 (http://www.python.org; http://www.biopython.org).

PostgreSQL 7.4.2 relational database management system (http://www.postgresql.org)

was compiled, while MySQL 4.0.20 (http://www.mysql.com) was not according to the developers instructions. MySQLdb, PyPgSQL, and psycopg database adaptors for Python were tested and used.

For benchmarks, SwissProt 43.3 was used with BioSQL schema 1.26 (http://obda.open-bio.org), and EMBOSS 2.8. (http://emboss.sourceforge.net). The BioSQL schema was populated by load_seqdatabase.pl of BioPerl [14]. The final pattern search was done on UniProt 4.4 (http://www.uniprot.org; ([15]).

All the documentation, scripts, database schema, and sample result sets are available at http://mbiosql.biohegedus.org.

3 Results

3.1 Generation of patterns with phosphorylation sites

Although R domain sequences are not highly conserved among different species, the distances between phosphorylation sites are preserved strikingly [11]. We chose the nine most important dibasic PKA sites involved in PKA mediated regulation to generate an amino acid pattern corresponding to the phosphorylation sites and any amino acids with appropriate numbers to define the distances between the phosphorylation sites (Table 1). Since this pattern is very strict other variants with decreased numbers of sites were assigned with increasing penalty scores. The maximum penalty score was defined as six. The resultant 340 basic patterns were used to generate three different classes of regular expressions allowing ±1, ±2, ±4 amino acid wobble between the phosphorylation sites. From the extensive pattern search with 1,020 motifs in the UniProt 4.4 database (SwissProt+trEmbl) containing 1,825,667 proteins, a high number of matches were expected and therefore a local warehouse with RDBMS was implemented.

3.2 Setting up mBioSQL and the pattern search as an application example

Since we could not find a system with acceptable performance and design for this project, we created one. The first optimization task was the population of the database from UniProt [15] containing a large amount of data (Figure 1, arrow a). From the benchmarks (Table 2) it is clear that the bottleneck is not the flat file parsing, but the insertion of large numbers of rows into the database. In order to enhance the performance of the inserts, most of the indexes were switched off during the data load and created after the population process. This resulted in approximately a tenfold decrease in time (7 h 23 min compared to 41 min) compared to the situation when there are existing indexes during the inserts.

Alignment of C-terminal part of R domains from different species

```
                    720|        730|        740|        750|        760|        770|        780|        790|
HUMAN   RKFSIVQKTPLQMNGIEEDSDEPLE--RRLSLVPDSEQGEAILPRISVISTGPTLQARRRQSVLNLMTHS-VNQGQNIHRKTTASTRKVS
BOVIN   RKFSVVQKTsLQMNGIEgaADaPLE--RRLSLVPhSEpGEgILPRsNaVNSGPTflggRRQSVLNLMTgSsVNQGQSIHRKTatSTRKMS
SHEEP   RKFSVVQKTsLQMNGIDgaSDEPLE--RRLSLVPhSEpGEgILPRsNaVNSGPTflggRRQSVLNLMTcSsVNQGQSIHRKTatSTRKMS
MOUSE   RKiSIVQKTPLcID---gESDDlqE--kRLSLVPDSEQGEAaLPRsNMIATGPTfpgRRQSVLDLMTfT-pNsGsSnlQRTrtSiRKIS
XENLA   RKFSLMQKSqpQMSGIEEEdmpaeQgeRKLSLVPESEQGEAsLPRsNfLNTGPTfQgRRQSVLNLMTrTsISQGsNafatrnASvRKMS
SQUAC   KKFSLVQtamsypqtngmEdatsepgeRhfSLIPENElGEptkPRsNIfkSelpfQAhRRQSVLaLMTHS-stspnkIHaRrSA-vRKMS
Cons    RKFSIVQKTxlqMNgiexesdxxxExxRRLSLVPeSExGExxLPRSNxInTGPTFxgxRRQSVLNLMTxSxvNqGxSihrKtSxSxRKMS
```

The basic R domain pattern (P:=[RK][RK]X[ST])

P_{660}	X(9)	P_{670}	X(15)	P_{686}	X(13)	P_{700}	X(11)	P_{712}	X(24)	P_{737}	X(30)	P_{768}	X(26)	P_{795}	X(17)	P_{813}

Decreasing the length

P_{660}	X(9)	P_{670}	X(15)	P_{686}	X(13)	P_{700}	X(11)	P_{712}	X(24)	P_{737}	X(30)	P_{768}	X(26)	P_{795}		
		P_{670}	X(15)	P_{686}	X(13)	P_{700}	X(11)	P_{712}	X(24)	P_{737}	X(30)	P_{768}	X(26)	P_{795}	X(17)	P_{813}
P_{660}	X(9)	P_{670}	X(15)	P_{686}	X(13)	P_{700}	X(11)	P_{712}	X(24)	P_{737}	X(30)	P_{768}				

Decreasing the P-sites (E: any four amino acids) — Classes by wobble (±1, ±2, ±4)

P_{660}	X(9)	E	X(15)	P_{686}	X(13)	P_{700}	...	P_{660}	X(8,10)	P_{670}	X(14,16)	P_{686}	...	
P_{660}	X(9)	P_{670}	X(15)	E	X(13)	P_{700}	...	P_{660}	X(7,11)	P_{670}	X(13,17)	P_{686}	...	
P_{660}	X(9)	E	X(15)	E	X(13)	P_{700}	...	P_{660}	X(5,13)	P_{670}	X(11,19)	P_{686}	...	

Table 1 Derivation of CFTR R domain phosphorylation patterns.

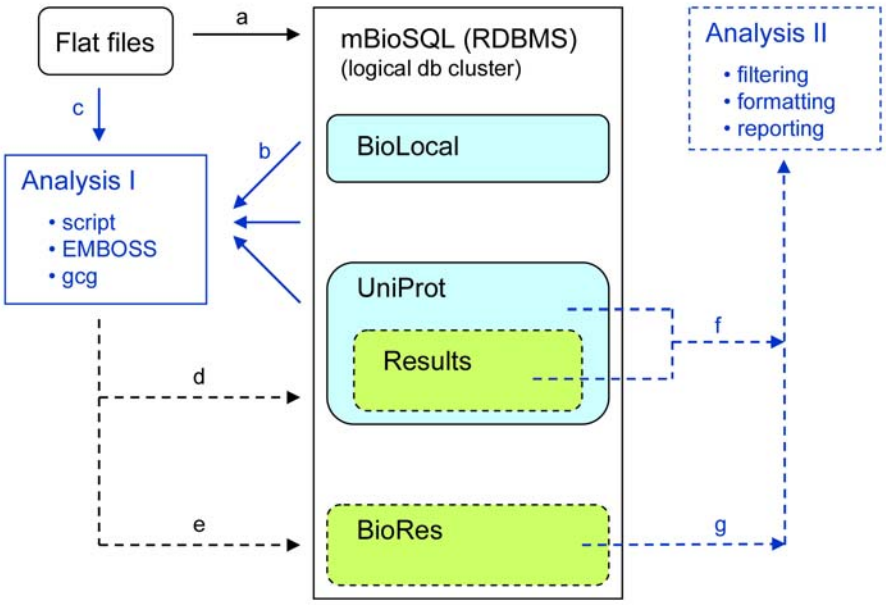

Fig. 1 Schematic representation of mBioSQL concepts. Data from flat files were parsed by BioPython parsers and loaded into RDBMS (arrow a). Coding of the controlled vocabularies and optimization of the index usage resulted in higher database performance. Sequences for level I analysis can be retrieved from either relational database (arrow b) or flat files (arrow c) depending on the application. According to the type of the result-set and the next analysis steps, the result can be stored either in a sequence database (arrow d) or in a separate 'Result' database (arrow e). By the powerful SQL merging data from 'data' tables and 'result' tables (arrow f), filtering, and formatting can be easily performed in the last analysis process (arrow f, g). See http://mbiosql.biohegedus.org for detailed database schema and data processing workflow.

In the case of large databases the searching also needs special optimization. We achieved increased speed of queries by coding the 'concepts', controlled vocabularies, like the keyword or database list of UniProt. These concepts can be stored in separate tables referencing their entries by integers from other tables that significantly decrease the size of the database (Table 2; see web page for detailed schema). For example, referencing the strings of the keywords with 2 byte long integers results in a decrease from 13 MB to 3 MB. This gain is not a significant saving of disk space, but a huge saving of memory. Application of this approach decreases both the data and the index size allowing retention of more of them in the memory. This results in a decreased number of disk accesses, which is the biggest limitation in RDBMS performance, thereby speeding up queries with index usage or resource consumption joins.

When retrieving specific entries from an RDBMS, SQL can be easily used to formulate quite complicated questions. However, because of the intensive internal administration the data access via RDBMS/SQL is not always justified (Figure 1, arrow b). For example, if access to all the sequence data from UniProt is required, the sequential reading and parsing of a file will always be much faster (Figure 1, arrow c; Table 2). If a specific subset

Loading data into RDBMS:			
Parsing the SwissProt flat file by BioPython:			6 min
System	*Data*	*Driver*	*Time*
BioSQL	SwissProt	psycopg	7h 23 min
mBioSQL	SwissProt	MySQLdb	48 min
mBioSQL	SwissProt	psycopg	41 min
Database size in the file system:			
System	*Data*		*Size*
BioSQL	SwissProt	Data+Index	1,126 M
mBioSQL	SwissProt	Data+Index	706 M
Flat file	UniProt	Data	3,225 M
		Index	766 M
		Total	3,991 M
mBioSQL	UniProt	Data	3,041 M
		Index	780 M
		Total	3,812 M
Reading random sequences:			
System	*Data*	*Driver*	*n(sequences)/sec*
Flat file**	UniProt		1,111,111
BioSQL	SwissProt	psycopg	7,142
mBioSQL	SwissProt	psycopg	14,556
mBioSQL	SwissProt	PyPgSQL	3,600

*Please note: not only the speed and size of the database defines its quality.
**Using EMBOSS tools.

Table 2 Benchmarks*.

of a database is required, other methods used in informatics can be applied: after easy selection of the interesting entries by SQL, the result can be written into a file (Figure 1, arrow a backward), or inserted into a separate table, and then analyzed much faster from the file or from the temporary table.

Employing this approach, we retrieved the sequences of each protein from the SwissProt and TrEMBL flat file for pattern search. R domain patterns with increasing penalty score were searched and the locations of matching proteins were stored in a separate 'result' table in our relational UniProt database (Figure 1, arrow d). The matched part of the sequence was masked, and the search was continued in the same protein. We stored the results in RDBMS since we found the analysis with SQL simple and more flexible than from a file.

The concept of loading analysis results back to the database, making further analysis, and generating statistics from there is generally used in informatics, and has already been applied in some complex, specific bioinformatical applications [16, 17]. This process could

be generalized for simple result-sets, like pattern searching, restriction mapping, etc. The basic idea behind this is the fact that running an analysis program with the less restrictive parameters, and storing and querying of all the results in an RDBMS, is less tedious than running the program several times with different parameters followed by the analysis of a set of files. We present examples of these tasks employing the EMBOSS package. Our scripts load EMBOSS result files into mBioSQL (br_load.py; Figure 1, arrow d, e), and show examples for further, lightweight, level II analysis from the relational system without any SQL programming knowledge (br_anal.py; Figure 1, arrow f, g).

3.3 RDBMS based, level II analysis of the matched proteins

Our phosphorylation patterns matched 5,833 proteins from the total UniProt database (0.3 % of 1,825,667 total), 644 from the SwissProt (0.3 % of 178,022), and 5,189 from the TrEMBL (0.3 % of 1,647,645). We made detailed analysis of matches with maximum penalty score 5 from Class ±1 and ±2, and maximum penalty score 4 from Class ±4, respectively, accounting for 575 proteins (Table 3, bold entries). As mentioned earlier, the id of the matched proteins, the locations of the matches and the penalty scores were loaded into a separate 'result' table in the relational UniProt database. Additional tables, as a subset of UniProt, with the matched proteins above our threshold were created to speed up the analysis. The advantage of this approach is the use of SQL for analysis.

Penalty	class±1	class±2	class±4
0	***15***	**3**	**5**
1	**0**	**4**	**20**
2	**1**	**0**	**43**
3	**0**	**6**	**56**
4	**4**	**29**	**236**
5	**21**	**167**	1032
6	424	1035	2950

*Total number of entries: 1,825,667.

Table 3 Number of matches from the UniProt database*.

By SQL it is possible to generate reports in different flexible ways, merging the protein data with the results into text, html, or other format, narrowing the result-set by applying different filters. Two entries are shown in Figure 2 from a very simple example file, which can be found at the mBioSQL web page containing all of the proteins above the threshold and ordered by penalty scores and classes.

Other features of SQL usage during the analysis of biological results are the simple functions, like COUNT, in combination with different conditions to produce simple statistics. Here we also show an example for analysis of matched proteins from SwissProt. Investigation of individual entries in the report file (Figure 2), containing all the matches from UniProt, showed a high number of matches with proteins with DNA, RNA, and

other nucleic acid associated SwissProt keywords. Based on this, proteins with 'nucleic acid-associated' (NAA) keywords (e.g. DNA condensation, nuclear protein, alternative splicing, etc.) and 'non-nucleic acid-associated' (non-NAA) keywords (e.g. microtubule, flavoprotein, antigen, etc.) were counted and categorized by the 'Rules' section of the SwissProt keyword list file [15]. Our result-set was significantly enriched in proteins with NAA keywords from the "Cellular component", "Complex", and "Sequence diversity" categories compared to the SwissProt database (Table 4). The ratios of proteins with NAA and non-NAA keywords in the "Biological process" category increased to the same extent (34 % and 36 %, respectively), although the initial proportion of NAA proteins was lower (9 % compared to 19 %). Since proteins have keywords from different categories, one entry potentially is counted in several categories, therefore the percentage values are not additive to 100 %. In order to calculate average values for the distribution of NAA and non-NAA proteins, the DISTINCT clause of SQL was used indicating that the previous set of proteins was concentrated from 23 % to 59 %, while the latter collection was depleted from 79 % to 41 % (Table 4).

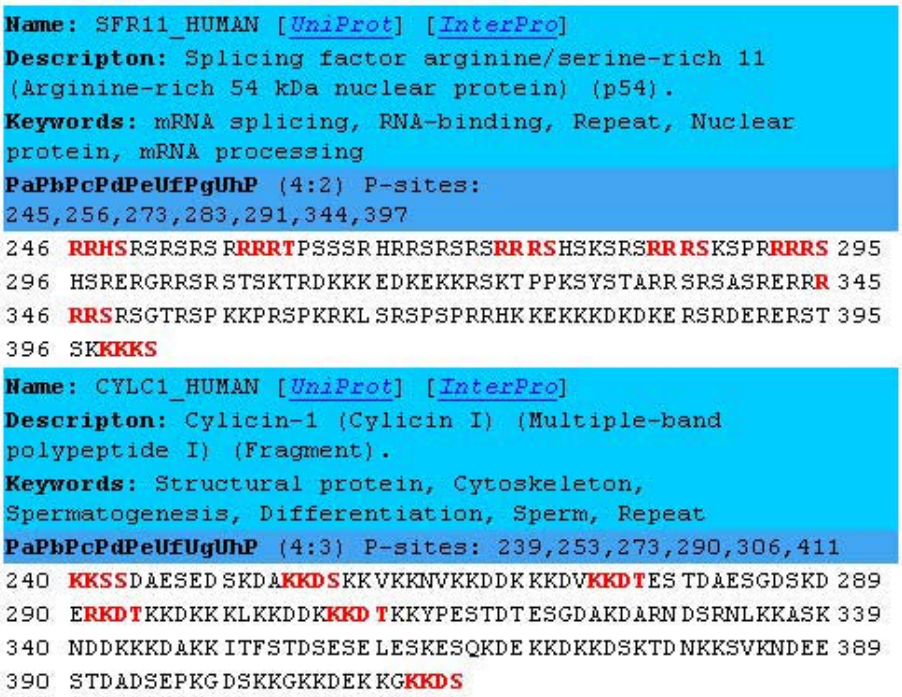

Fig. 2 Two entries from the result file. The result-set of the pattern search were stored in the relational UniProt database. This approach allowed merging of results with data from the sequence database itself into an html file. Two entries from this file with links to the UniProt and InterPro database, with primitive pattern (P: [RK][RK]X[ST] phosphorylation pattern, E: any four amino acids, a-h: distances between the phosphorylation patterns), the class with penalty score, and the matched regions demonstrate how the matched proteins can be visualized and analyzed individually. The full file can be found at the mBioSQL website.

Category; Keywords	N^{+}	$N(SP^{++})$
Cellular component		
Nuclear protein	26* (41 %)	11517 (7 %)
Cell wall, Chloroplast, Cytoskeleton, Membrane, Mitochondrion, Ribosomal protein	9 (14 %)	22391 (13 %)
Complex		
Chromosomal protein, Nucleosome core, Spliceosome	4* (6 %)	1090 (0.6 %)
Microtubule	2 (3 %)	846 (0.5 %)
Sequence diversity		
Alternative splicing	17* (27 %)	6591 (4 %)
Polymorphism	3 (5 %)	3283 (2 %)
Biological process		
DNA condensation, DNA damage, DNA repair, DNA replication, Transcription regulation, mRNA processing, mRNA splicing	22* (34 %)	15756 (9 %)
Other(Pathway, Transport, etc)	23* (36 %)	31962 (19 %)
Ligand, Nucleotide-binding		
DNA-binding, RNA-binding, rRNA-binding	21* (33 %)	19818 (12 %)
ATP binding, FMN, Flavoprotein, GTP-binding	20* (31 %)	22893 (13 %)
Molecular function		
Activator, Initiation factor, Repressor, Ribonucleoprotein, Trans-acting factor	8 (13 %)	15832 (9 %)
Antigen, Developmental protein, Photoreceptor, Structural protein	5 (8 %)	4683 (3 %)
'Nucleic acid associated' matches:	38* (59 %)	40024 (23 %)
'Non-Nucleic acid associated' matches:	26(41 %)	135587 (79 %)

$^{+}$ Matched and filtered proteins from the SwissProt; total: 64
$^{++}$ All proteins with keywords from the SwissProt; total: 170971
* Enrichment was tested by binomial test ($P < 0.001$)

Table 4 Simple statistical analysis of the matched SwissProt entries.

4 Discussion

CFTR is an anion channel that plays a crucial role in secretion and absorption of salt and fluid by epithelial tissues. To be effective in their function, ion channels such as CFTR must be very tightly regulated. The R domain provides this control; phosphorylation of its multiple sites by PKA is obligatory for channel gating. Despite extensive investigations [4, 5, 10, 18–20] the mechanism whereby the unphosphorylated R domain maintains the non-gating state is not understood. As a supplement to the various experimental approaches that have been applied, we have attempted to obtain functional clues by searching for similarities with other proteins.

As the R domain sequence is not conserved, but the relative distances of the phospho-

rylation sites are preserved in this domain from different species (Table 1), we searched the UniProt protein database for proteins with similar phosphorylation pattern distribution. Simple analysis of the matched entries revealed that the majority of these proteins are nuclear proteins such as transcription factors, splicing factors, and antisigma factors, etc. (Table 4). The common feature of these proteins is that their matched regions are involved in protein-protein interactions highly regulated by phosphorylation. This is conceptually consistent with the idea that CFTR dependent regulation of other transporters might occur through direct protein-protein interaction via the R domain. Our findings may help to interpret the nature of intermolecular interactions such as those, which may occur with calcium-activated chloride channels [21] and SLC (solute carrier) transporters [22], as well as the role of intramolecular interactions of CFTR R domain in the gating mechanism [10].

The SLC proteins are bicarbonate/chloride exchangers that may be important in understanding cystic fibrosis, as impaired bicarbonate secretion in the pancreas and small intestine is one of the most visible phenotypes of patients [23, 24]. SLC26A3 (DRA: Down-Regulated in Adenoma) and SLC26A6 were reported to interact physically with CFTR R domain through their C terminal STAS (sulfate transporter and anti-sigma factor antagonist) domains, influencing both their own function and that of the channel [22]. DRA is believed to be a tumor suppressor. Although, not universally accepted, there are more and more indications that it influences cell growth [25, 26]. It is most likely that the STAS domain with high similarity to anti-sigma factor transcription regulators interferes with signaling pathways. Similar events may account for the high number of published regulatory interactions between CFTR and other proteins [27–29].

The non-nuclear proteins with loose matches include a voltage-gated sodium channel (UniProt:P35498). The function of this protein is modulated by phosphorylation of an intracellular loop [30], which acts as a regulatory domain. This sodium channel, like CFTR, also matures inefficiently during biosynthesis [31]. Only ∼2 % of the protein synthesized reaches the plasma membrane, far less than the ∼25 % of CFTR [32]. It is unknown if the labile nature of these two channels is related to their regulation by phosphorylation.

Since many nuclear proteins contain large numbers of phosphorylation sites, i.e. serine/arginine reach regions, it is also possible that R domain matches with these proteins are coincidental. However, the low ratio of the hits compared to the total number of entries in the UniProt database (0.3 %) and some experimental observations (like CFTR/SLC interactions: [22]) strengthen the validity of the results of the pattern search.

To perform our analysis we set up a local warehouse system including a biological RDBMS (Figure 1). We decided to store our data in an RDBMS, as the expected large result-set would be analyzed more easily with SQL compared to conventional tools, like *grep* and *awk*. Unfortunately most of the biological RDBMS are at the experimental stage with insufficient performance, not yet ready to be used by wet biologists for a specific problem. At the time of development we did not have the opportunity to try the recently published *Atlas* data warehouse [33]. In our package, different biological

databases were implemented in different schema, as we use a small number of different biological databases, and do not intend to deal with the integration problem. We chose the Python scripting language for programming because development in a scripting language is significantly easier and faster compared to other languages like C or Java. Python has a clean object-oriented syntax and runs on most platforms.

The individual implementation of databases permitted population of our databases faster without indexes that were created after the loading was completed. We also could reference certain 'concepts' like database names, feature names, keywords with integers that decreased the database size, and the index size resulting in significant increase in database performance (Table 2). This approach is not used by any of the available public RDBMS systems implementing SwissProt (BioSQL, Prose, solution of [34]). However, it is important to emphasize that a general schema to store heterogeneous biological data (like the BioSQL schema) is needed to help both programmers and users to make development and querying simpler [35–37].

Similar to a productive database environment, we did not use RDBMS exclusively. In some situations it is more reasonable to read data from files, to write the query result into a file, and to analyze it from there. In other cases it is more efficient to select the result into a temporary table. These types of decisions are usually made by experienced programmers, however in some cases automated, intelligent decisions could be planned by programs themselves.

Large results are often stored in an RDBMS system to simplify further analysis and report generation steps. We used this approach to investigate the matched proteins in our pattern search. Furthermore, we generalized this concept and point out that it is worthwhile to load simpler biological results back into RDBMS in order to solve further questions by the powerful SQL (Figure 1). Python scripts are also provided as examples of how to implement this concept without the need of any SQL or programming knowledge and how to integrate RDBMS with analysis packages (like EMBOSS). Several complete software packages with relational database backend appeared recently [38–40], but none could be easily applied to our project. Some of them are gene-centric (e.g. GeneKeyDB), while others need large resources (a 28 CPU Linux cluster is recommended for the *Pegasys* server).

In summary, in order to find proteins with similarity to CFTR R domain a large scale pattern search was implemented based on the easy to use Python, while the analysis was performed through an RDBMS. All the tools and solutions used are freely available and platform-independent. During the rationalization process we faced several informatical challenges and developed solutions that could serve as examples in development of new packages to serve the needs of biologists. Moreover, the results generated by our local warehouse indicate that the R domain of CFTR is similar to regions of various proteins taking part in protein-protein interactions that are highly regulated by phosphorylation. Their mode of regulation and action may provide new insights into structural and functional relationships of the CFTR R domain.

Acknowledgment

Technical help by Seth Kurtzberger and Walter Krafft is greatly appreciated. We are grateful to András Váradi and Stephan Philippi for reviewing the manuscript.

References

[1] J.R. Riordan et al.: "Identification of the cystic fibrosis gene: cloning and characterization of complementary DNA", *Science*, Vol. 245, (1989), pp. 1066–1073.

[2] Y.H. Ko and P.L. Pedersen: "Cystic fibrosis: a brief look at some highlights of a decade of research focused on elucidating and correcting the molecular basis of the disease", *J. Bioenerg. Biomembr.*, Vol. 33, (2001), pp. 513–521.

[3] S.H. Cheng et al.: "Phosphorylation of the R domain by cAMP-dependent protein kinase regulates the CFTR chloride channel", *Cell*, Vol. 66, (1991), pp. 1027–1036.

[4] F.S. Seibert et al.: "Influence of phosphorylation by protein kinase A on CFTR at the cell surface and endoplasmic reticulum", *Biochim. Biophys. Acta*, Vol. 1461, (1999), pp. 275–283.

[5] L. Csanady et al.: "Preferential phosphorylation of R-domain Serine 768 dampens activation of CFTR channels by PKA", *J. Gen. Physiol.*, Vol. 125, (2005), pp. 171–186.

[6] D.C. Gadsby and A.C. Nairn: "Control of CFTR channel gating by phosphorylation and nucleotide hydrolysis", *Physiol. Rev.*, Vol. 79, (1999), pp. S77–S107.

[7] C. Li et al.: "ATPase activity of the cystic fibrosis transmembrane conductance regulator", *J. Biol. Chem.*, Vol. 271, (1996), pp. 28463–28468.

[8] J.R. Riordan: "Assembly of functional CFTR chloride channels", *Annu. Rev. Physiol.*, Vol. 67, (2005), pp. 701–718.

[9] D.P. Rich et al.: "Regulation of the cystic fibrosis transmembrane conductance regulator Cl- channel by negative charge in the R domain", *J. Biol. Chem.*, Vol. 268, (1993), pp. 20259–20267.

[10] V. Chappe et al.: "Phosphorylation of CFTR by PKA promotes binding of the regulatory domain", *Embo. J.*, Vol. 24, (2005), pp. 2730–2740.

[11] A.M. Dulhanty and J.R. Riordan: "A two-domain model for the R domain of the cystic fibrosis transmembrane conductance regulator based on sequence similarities", *FEBS Lett.*, Vol. 343, (1994), pp. 109–114.

[12] A.M. Dulhanty and J.R. Riordan: "Phosphorylation by cAMP-dependent protein kinase causes a conformational change in the R domain of the cystic fibrosis transmembrane conductance regulator", *Biochemistry*, Vol. 33, (1994), pp. 4072–4079.

[13] L.S. Ostedgaard et al.: "A functional R domain from cystic fibrosis transmembrane conductance regulator is predominantly unstructured in solution", *Proc. Natl. Acad. Sci. USA*, Vol. 97, (2000), pp. 5657–5662.

[14] J.E. Stajich et al.: "The Bioperl toolkit: Perl modules for the life sciences", *Genome. Res.*, Vol. 12, (2002), pp. 1611–1618.

[15] A. Bairoch et al.: "The Universal Protein Resource (UniProt)", *Nucleic Acids Res.*, Vol. 33, (2005), pp. D154-D159.

[16] C. del Val et al.: "High-throughput protein analysis integrating bioinformatics and experimental assays", *Nucleic Acids Res.*, Vol. 32, (2004), pp. 742–748.

[17] E.L. Grogan et al.: "Volatility: a new vital sign identified using a novel bedside monitoring strategy", *J. Trauma.*, Vol. 58, (2005), pp. 7–12; discussion 12–14.

[18] L.S. Ostedgaard, O. Baldursson and M.J. Welsh: "Regulation of the cystic fibrosis transmembrane conductance regulator Cl- channel by its R domain", *J. Biol. Chem.*, Vol. 276, (2001), pp. 7689–7692.

[19] V. Chappe et al.: "Stimulatory and inhibitory protein kinase C consensus sequences regulate the cystic fibrosis transmembrane conductance regulator", *Proc. Natl. Acad. Sci. USA*, Vol. 101, (2004), pp. 390–395.

[20] L. Csanady et al.: "Functional roles of nonconserved structural segments in CFTR's NH2-terminal nucleotide binding domain", *J. Gen. Physiol.*, Vol. 125, (2005), pp. 43–55.

[21] L. Wei et al.: "The C-terminal part of the R-domain, but not the PDZ binding motif, of CFTR is involved in interaction with Ca(2+)-activated Cl- channels", *Pflugers Arch.*, Vol. 442, (2001), pp. 280–285.

[22] S.B. Ko et al.: "Gating of CFTR by the STAS domain of SLC26 transporters", *Nat. Cell. Biol.*, Vol. 6, (2004), pp. 343–350.

[23] D.B. Mount and M.F. Romero: "The SLC26 gene family of multifunctional anion exchangers", *Pflugers Arch.*, Vol. 447, (2004), pp. 710–721.

[24] M.J. Hug, T. Tamada and R.J. Bridges: "CFTR and bicarbonate secretion by [correction of to] epithelial cells", *News Physiol. Sci.*, Vol. 18, (2003), pp. 38–42.

[25] A. Hemminki et al.: "Intestinal cancer in patients with a germline mutation in the down-regulated in adenoma (DRA) gene", *Oncogene*, Vol. 16, (1998), pp. 681–684.

[26] J.M. Chapman et al.: "The colon anion transporter, down-regulated in adenoma, induces growth suppression that is abrogated by E1A", *Cancer Res.*, Vol. 62, (2002), pp. 5083–5088.

[27] E.M. Schwiebert et al.: "CFTR is a conductance regulator as well as a chloride channel", *Physiol. Rev.*, Vol. 79, (1999), pp. S145–S166.

[28] K. Kunzelmann: "CFTR: interacting with everything?", *News Physiol. Sci.*, Vol. 16, (2001), pp. 167–170.

[29] A.P. Naren et al.: "A macromolecular complex of beta 2 adrenergic receptor, CFTR, and ezrin/radixin/moesin-binding phosphoprotein 50 is regulated by PKA", *Proc. Natl. Acad. Sci. USA*, Vol. 100, (2003), pp. 342–346.

[30] A.R. Cantrell et al.: "Molecular mechanism of convergent regulation of brain Na(+) channels by protein kinase C and protein kinase A anchored to AKAP-15", *Mol. Cell. Neurosci.*, Vol. 21, (2002), pp. 63–80.

[31] W.B. Thornhill and S.R. Levinson: "Biosynthesis of ion channels in cell-free and metabolically labeled cell systems", *Methods Enzymol*, Vol. 207, (1992), pp. 659–670.

[32] S. Pind,, J.R. Riordan and D.B. Williams: "Participation of the endoplasmic reticu-

lum chaperone calnexin (p88, IP90) in the biogenesis of the cystic fibrosis transmembrane conductance regulator", *J. Biol. Chem.*, Vol. 269, (1994), pp. 12784–12788.

[33] S.P. Shah et al.: "Atlas - a data warehouse for integrative bioinformatics", *BMC Bioinformatics*, Vol. 6, (2005), pp. 34.

[34] G. Xie et al.: "Storing biological sequence databases in relational form", *Bioinformatics*, Vol. 16, (2000), pp. 288–289.

[35] J. Kohler, S. Philippi and M. Lange: "SEMEDA: ontology based semantic integration of biological databases", *Bioinformatics*, Vol. 19, (2003), pp. 2420–2427.

[36] S. Philippi: "Light-weight integration of molecular biological databases", *Bioinformatics*, Vol. 20, (2004), pp. 51–57.

[37] S. Stephens: "Data Integration and Knowledge Aggregation in Life Sciences Discovery", *Scientific Comput. Instrum.*, Vol. 21, (2005).

[38] G. Finak et al.: "BIAS: Bioinformatics Integrated Application Software", *Bioinformatics*, Vol. 21, (2005), pp. 1745–1746.

[39] S.A. Kirov et al.: "GeneKeyDB: a lightweight, gene-centric, relational database to support data mining environments", *BMC Bioinformatics*, Vol. 6, (2005), pp. 72.

[40] S.P. Shah et al.: "Pegasys: software for executing and integrating analyses of biological sequences", *BMC Bioinformatics*, Vol. 5, (2004), pp. 40.