From Sequence to Structure To Literature : The Protocol Approach to BioInformation

O. P Wu owu@bic.nus.edu.sg
BioInformatics Center, National University of Singapore, Singapore 119260
Institute of System Science, Singapore 119597

K.T Seow mcbskt@nus.sg
Institute of Molecular & Cell Biology, National University of Singapore, Singapore 119260

L. Wong limsoon@iss.nus.sg
BioInformatics Center, National University of Singapore, Singapore 119260
Institute of System Science, Singapore 119597
Institute of Molecular & Cell Biology, National University of Singapore, Singapore 119260

S. Y Chung suchung@zen2.usuf2.usuhs.mil
Dept. of Biochemistry, Uniformed Services University of Health Science, Bethesda, MD, USA

S. Subbiah subbiah@subbiah.wistar.upenn.edu
Wistar Institute, 3601 Spruce St. Philadelphia, PA 19104, USA
BioInformatics Center, National University of Singapore, Singapore 119260

Abstract:

Until the recent advent of high-throughput experimental data-acquisition in biology, the computational analysis of the biological data was predominantly on an *ad hoc* basis - i.e. the application of a given piece of software on the biological data depended on the need of the moment. This "functional approach" often resulted in piecemeal computational analysis with large amount of intervening "dead-time". The present high-throughput availability of experimental biological data requires a more streamlined and integrated "protocol approach". In this work, we illustrate such a user-friendly protocol using a common question frequently faced by a wet-lab bench-biologist - "Now that I have a DNA or protein sequence, what can I do with it using a computer ?". As phrased, this question is steeped in the functional approach. In contrast, the protocol approach would re-phrase the same question as "Now that I have a DNA or protein sequence, what can a computer do for me ?". Our integrating tool can start with a sequence and build a substantial custom data-warehouse of computationally derived sequence information, structure information and relevant published literature, that is continually updated.

Section 1. Introduction

With the advent of many high-throughput experimental methodologies (e.g. sequencing, combinatorial chemistry) in modern biology, there has been an explosion of biological data. Like the earlier data obtained by more laborious and slower means, the newer data too is eminently computable, using the many hundreds of public and commercial computer algorithms that have been developed over the last few decades. While applying well-known predictive computer algorithms one at a time on experimental data was adequate for earlier times and reflects a "functional approach" to computational analysis, the automated nature of experimental data acquisition in the current era, begs the development of a more streamlined approach. The present work explores a "protocol approach"[1] for theoretical analysis, that is not unlike the protocol based high-throughput, experimental data acquisition process itself. Unlike the functional approach which left the bench-biologist to painfully apply *ad hoc* one

computer software at a time, with much lost time in between trying to recall/or get advice on what piece of software to use next, the "protocol approach" guides (i.e. prompts) the same bench-biologist through a natural path of successively relevant computer software.

In other words, given a set of disparate computer software, one assesses the typical order in which a bench-biologist uses the different software, given a particular starting point, e.g. a newly sequenced piece of DNA. Based on this user-input information, a meaningful order of programs - that is, a protocol - can be stitched together to form the basis of a user-friendly, end-user application, preferably one that is deployed on the ubiquitous web-enabled PC that sits on every wet-lab bench.
This protocol approach then, requires:

1. Finding typical end-user biologists and ascertaining what software programs they normally use (e.g. GCG, BLAST etc.)
2. Surveying the same biologists, and determining the typical order in which they use this disparate software given different starting points. (e.g. I have a DNA sequence or I have a protein sequence and what can a computer do for me?)
3. The third step is to integrate the various software, in the user-preferred order,to create a natural work-flow, with a user-friendly interface.

In the present work, we talked to our end-user, wet-lab biologists in the Institute of Molecular & Cell Biology (in particular K.T. S) to achieve the first and second steps. For the third step, we used the highly sophisticated general data integration system Kleisli/CPL[2] that has been under development in-house at our BioInformatics Center.

What is Kleisli/CPL

The Kleisli/CPL system is a very powerful data integration system that can seamlessly integrate/query a diverse array of seemingly incompatible data sources on-the-fly. The data sources can be varied and complicated remote databases, relational like *Oracle, etc.* or object oriented or even simply unstructured flat files, as well as predictive computer algorithms that generate a lot of data and thus are de facto virtual databases.

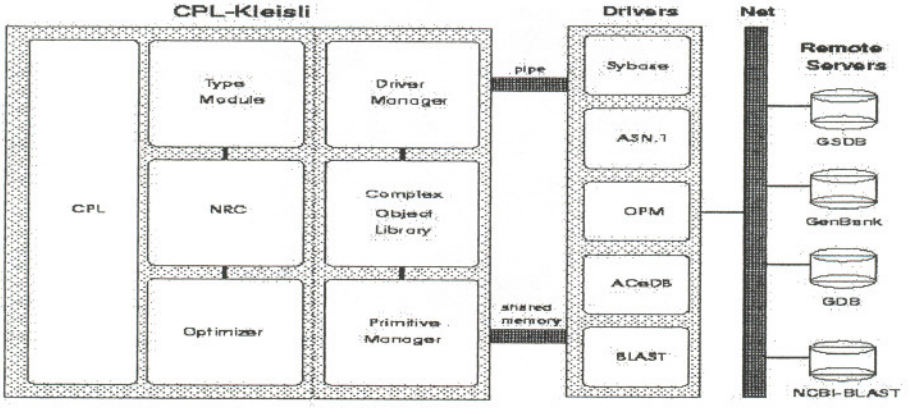


Figure 1. Structure of Kleisli/CPL

Kleisli, together with the Collection Programming Language (CPL), which is a high-level query language that has been implemented for it, is a powerful open query system for broad-scale integration of heterogeneous distributed databanks. It is particularly suitable for dealing with the complexity of the many existing biological databanks. It offers high-level flexible access to human genome and other biological sources that are highly heterogeneous, geographically scattered, highly complex, constantly evolving and high in volume. By its nature, it generates data warehouses of integrated results that are constantly updated as the data sources themselves are updated.

The openness of Kleisli allows the easy introduction of new primitives, optimization rules, cost functions, data scanners, and data writes. Queries that need to freely combine external data from different sources are readily expressed in short Kleisli/CPL scripts. While the Kleisli/CPL system is described in detail in more than 30 papers ([2][3][4]...) Figure 1 illustrates its general layout and component parts. CPL, the high-level query language, sits atop various components of Kleisli like the query optimizer, drive manager, primitive manager and the ***NRC*** (nested relational calculus) module. These in turn communicate with the various extend biological data sources, like GDB[5], Genbank[6] and popular computer programs like BLAST[7], via their associated specific drivers that can comprehend the underlying data schema/structure. The current version of Kleisli/CPL has built-in drivers for the data sources used in the present work - Prosite[8], Genbank, etc.

As described in [2][3][4], the Kleisli/CPL system comes equipped with a large number of drivers that handle data communication with a wide array of popular biological databases, e.g. *ACeDB, Genbank*, and standard software that embed predictive computer algorithms such as ClustalW[9] multiple sequence alignment, BLAST and etc.

The high-level query language CPL makes it relatively simple to implement queries that involve multiple sources. As an example, here is a CPL script for extracting the summary, sequence, and MedLine abstracts of denguevirus sequences having helicase motifs.

```
{ (#summary: x,  #seq: y,  #abstracts: Z)
| \x ← "(organism denguevirus)".amino-summaries,
  \r ← x.#uid.amino-report,
  \Z == { z | \m ← r.muids, \z ← m.medline-abstracts},
  \y ← r.seq,
  \W == { w | \w ← y.prosite-scan, w.#title islike "%helicase%" },
 not (W = {})}
```

Let us briefly explain the CPL script. The expression *"(organism denguevirus)".amino-summaries* causes the amino-summaries function to be executed, which accesses Entrez[10] and returns summaries of all denguevirus sequences. For each such summary x, the expression *x.#uid.amino-report* passes x's unique identifier (*x.#uid*) to the function amino-report, which accesses Entrez and returns x's full report. For each such report r, we then build the set Z containing the abstracts of all MedLine articles referenced by r. This is accomplished by first

running the function *muids* to extract MedLine identifiers in *r*; then running the function *medline-abstracts* on each such identifier *m*. For each report *r*, the expression *r.seq* invokes the function *seq* to extract the sequence in *r*. The above gives us the summary, sequence, and abstracts of all denguevirus sequences. We still need to determine which ones have the helicase motif. For each sequence *y*, the expression *y.prosite-scan* invokes the Prosite-scan software to return known motifs in *y*. For each motif *w*, we check if its title contains the word "helicase". If so we save such *w*s into a set *W*. Clearly, *y* has a helicase motif if *W* is non-empty.

In this work, using the **Kleisli/CPL** system, with relative ease we have integrated almost 20 commonly available programs, databases and other bioinformatics tools into a single cohesive protocol that takes a "wet-lab" PC-empowered biologist from his/her recently discovered gene sequence to various biology information all the way to structure information.

Section 2. Methods and Data sources

2.1 Program tools

- BLAST (Basic Local Alignment Search Tool):
  This is a popular software for identifying homologs of a query sequence from a database. It is available from NCBI and comes in various versions:

  ⇒ blastp which compares an amino acid query sequence against a protein sequence database
  ⇒ blastx which compares a nucleotide query sequence translated in all reading frames against a protein sequence database.
  ⇒ blastn, tblastp, etc.

- ClustalW:
  A popular general purpose multiple alignment program for DNA or protein sequences that allows introducing insertions of gaps.

- Prosite:
  A publicly accessible WWW service which contains a method for determining the function of uncharacterized proteins translated from genomic or DNA sequences. It consists of a database of biologically significant sites, patterns and profiles that help to reliably identify which known family of proteins (if any) a new sequence may belong to.

- Rasmol[11]:
  Molecular graphics software for viewing and manipulating 3-D structures of proteins.

2.2 Data sources:

- **Literature**

  ⇒ **Medline**:
  A bibliographic database produced by the U.S. National Library of Medicine . The database covers worldwide biomedical literature, the citations of which appear in Index Medicus, Index to the Dental Literature and International Nursing Index. We use only the portion of this database which has been incorporated into the popular software Entrez developed by NCBI for automatic hyper-linking of related biomedical information.

- **Protein data**

  ⇒ **swissprot:**
  A very comprehensive compilation of protein sequence maintained in Switzerland.

  ⇒ **pdb:**
  The protein structure database at Brookhaven that contains both protein sequences and 3-dimensional atomic structures

  ⇒ **pir:**
  The Protein Identified Resource - a collection of protein sequence

  ⇒ **nr:**
  This is a non-redundant composite of several well-known biological sequence databases - CDS translations of nucleotide sequence + PDB + SwissProt + PIR

- **DNA databases**

  There are many of these databases and our current set includes **yeast**, **E.coli**, **dbEST**, etc.

Section 3. An example

Given a protein (or DNA) sequence, you want to find all homology of this given sequence. And then you want to perform several operation on some homology, such as Prosite scan, perform multiple alignment on selected homology. Or you want to know the Medline report about some homology. What will you do ?

3.1 Functional approach

As described in ***section 1***(Introduction), using the functional approach may lead to the following separate and often *ad hoc* steps:

- run BLAST on your sequence to get a list of possible homology.
- select a few items from this list.
- perform the fetch program from the GCG Wisconsin suite of sequence program[12] to bring each of their sequence back

- extract these sequences and put them into a file in FASTA format.
- run ClustalW to see the alignment
- ...

3.2 Protocol approach

The streamlined organization of standard bio-computing tools according to their sequential use by wet-lab biologists forms the basis of this approach. With the help of a wet-lab biologist (K.T. Seow), we decided to follow the order as shown in *Figure 4.*

While the system can handle either cDNA sequences or protein sequences as input data, we illustrate the general approach by way of the following protein sequence

SGSFELSVQDLNDLLSDGSGCYSLPSQPCNEVTPRIYVGNASVAQDIPKLQKLGIHVLN AAEGRSFMHVNTNANFYKDSGITYLGIKANDTQEFNLSAYFERAADFIDQLAQKNGR VLVHCREGYSRSPTLVIAYLMMRQKMDVKSALSIVRQNREIGPNDGFLAQLCQLNDR LAKEGKLKP

Step 1. Submit the protein sequence

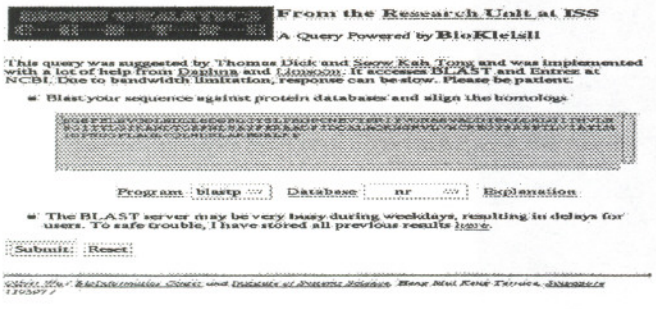


Figure 2. Interface of Protocol Approach

We submit the sequence to our WWW interface[13]. With protein sequence data, the user can choose different blast programs (blastp, blastx) and 2 protein sequence databases (nr, swissprot) as explained in ***section 2***. For this example, we use the **nr** database

Step 2. Perform the BLAST operation

As shown in ***figure 1***, Kleisli/CPL itself carries a driver for BLAST. After getting the data from the remote server located at NCBI, Kleisli/CPL produces a BLAST result - a table of sequences and their entire identifiers from the ***nr*** database (e.g. pdb|2HNP|). The user can directly adjust the level of homology to suit his or her need.

```
Query= gi23
       (183 letters)

Database:  PDB protein sequences
           3114 sequences; 587,048 total letters.
Searching.....................................................done

                                                                  Smalles
                                                                    Sum
                                                          High  Probabil
Sequences producing High-scoring Segment Pairs:           Score  P(N)

 pdb|2HNP|   Protein-Tyrosine Phosphatase 1b (Human) (E.C....    57  0.21
 pdb|1T8Y|   Mol_id: 1; Molecule: Trypsin; Chain: Null; Ec...    56  0.26
 pdb|1YPT|A Protein-Tyrosine Phosphatase (Yersinia) (E.C.3...    52  0.70
 pdb|1DDS|A Molecule: Dihydrofolate Reductase (E.C.1.5.1.3...    49  0.94
```

Figure 3. BLAST result

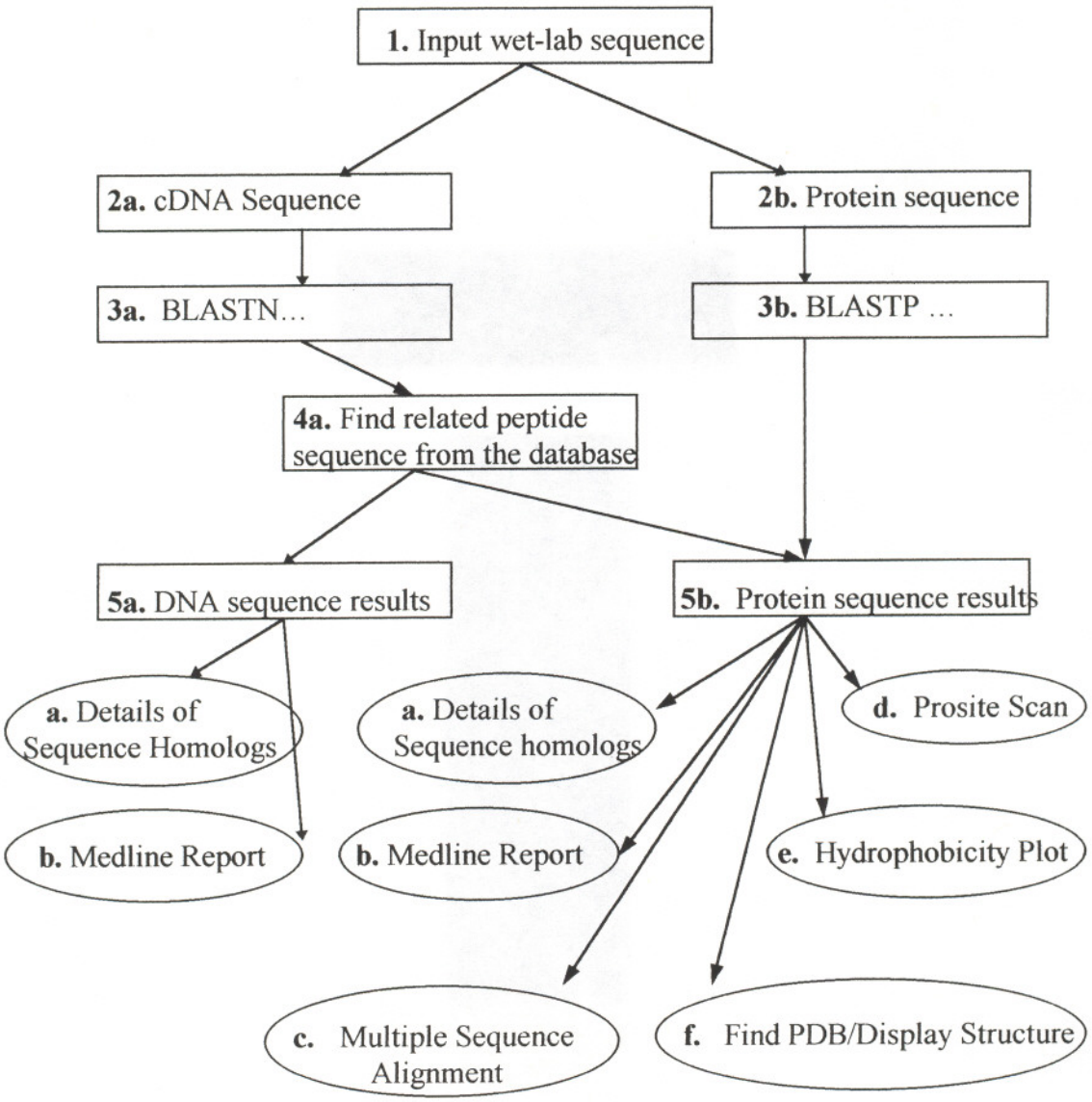


Figure 4. The protocol approach -
A bioinformatics workflow for the wet-lab biologist

Step 3. Further operations on the homologs obtained

3.a Getting the details of sequence homologs

As shown in figure 5, using the links obtained by the Entrez program the detailed information contained with an NR database file (e.g. 2HNP) was accessed by Kleisli/CPL. Various details like the title/file name, the sequence, uid (e.g. 809208) are displayed for the full set (or user selected sub-set) of sequence homologs found by BLAST in step 2.

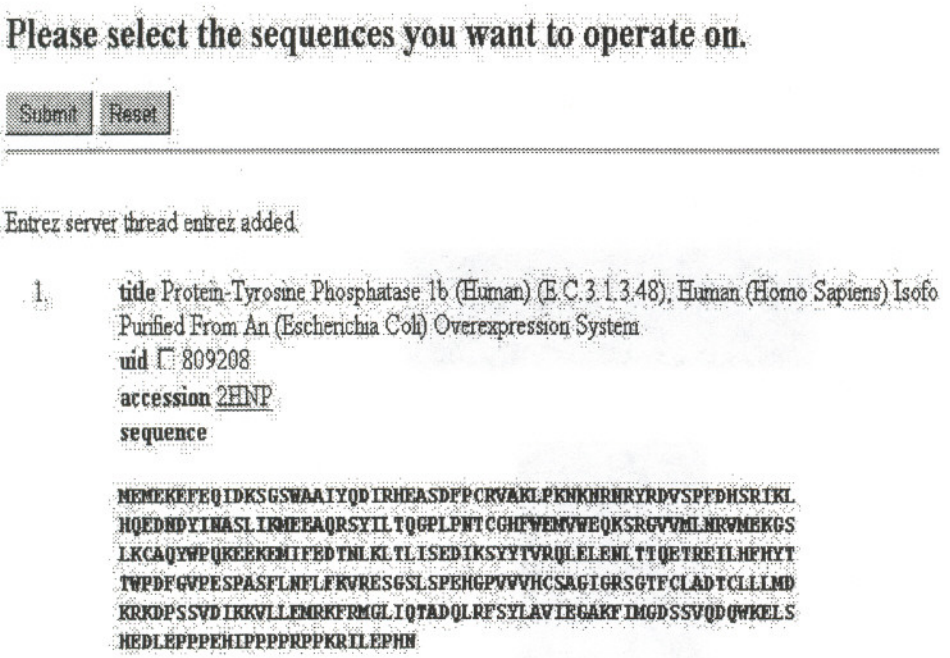


Figure 5. Homologs obtained

3.b Get the Medline report

For each homology, using the muid(e.g.94174273) obtained but not displayed in step 3.a. Kleisli/CPL gathers all the related medline abstracts, figure 6. Further, while not shown, it is possible to click on the muid location (e.g. 94174273) to access/display Entrez's links to the medline abstracts most-related to those initially obtained. Obviously this process can be repetitively iterated with the help of Entrez.

3.c Perform Multiple Sequence Alignment on selected homologs

Multiple sequence alignment of the homologous protein sequences, labeled by their uids, is automatically conducted using the standard ClustalW program (figure 7) The ClustalW symbol-key - * for identical, . for homologs and - for gaps - is also presented.

3.d Find the protein family for selected homologs - *Prosite Scan*

For both the original sequence input by the user - user sequence - and all its homologs defined by the previous blast step, the prosite library of sequence motifs/signatures is searched for any pattern matches. For each known sequence motif/signature, e.g. the phosophorylase kinase, its occurrence in either the user sequence or any of its homologs is displayed (figure 8), together with additional information such as the title and the actual sequence alignment between the user sequence and the matched sequence segment. Thus the user sequence and its homologs are classified into the well-known prosite protein families (if any).

3.e Protein Sequence Coloring/ Hydrophobicity Profile Generation

This is a handy tool for displaying the user sequence or the homology related /obtained in the BLAST step, using a user-specified color coding scheme based on the amino-acid single-letter code. One built-in color coding scheme, shown in figure 9a, is the color-coding based on standard hydrophobicity rules - where the amino acid are colored in order of increasing hydrophobicity - R,K,D,E,N,Q,H(*blue*), P,Y,W,S,T,G (*green*),A,M,C,F (*yellow*) and L,V,I (*red*),

3.f Finding and displaying (if any) the 3-dimensional structure of the user sequence or its homologs.

In this step, using the moid identifier tag, which links protein sequence files from the **nr** database to its related 3-dimensional structure file in the **PDB**[14], we are able to retrieve the 3-dimensional atomic coordinates of the user protein sequence and its homologs. Each 3-D structure can be viewed and manipulated by clicking the model button (as shown in figure 10) which activates the PDB-distributed molecular graphics software, **Rasmol.**

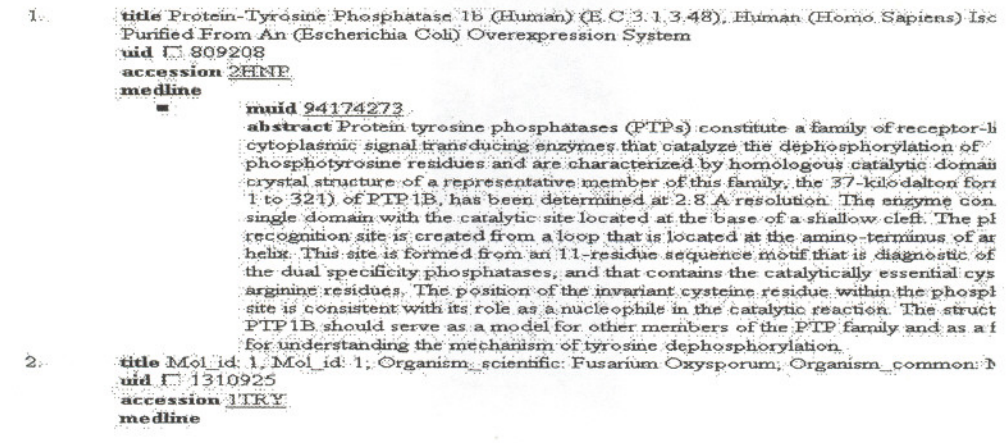
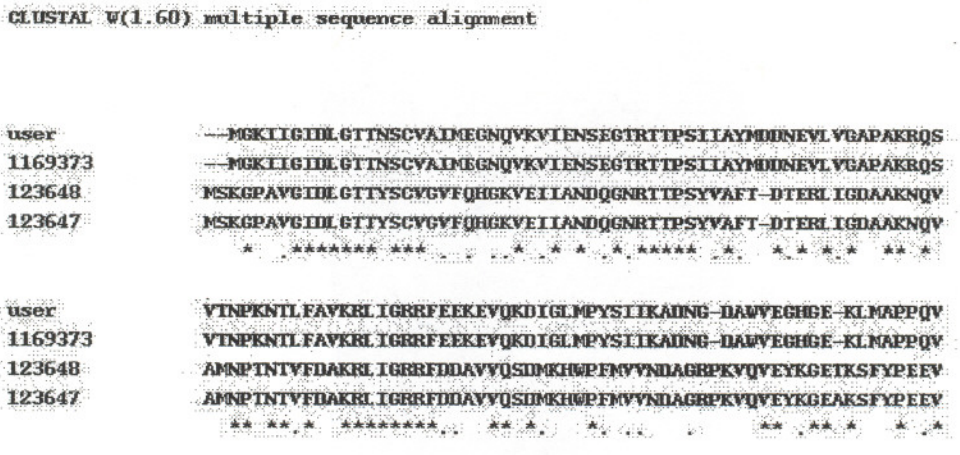

1. **title** Protein-Tyrosine Phosphatase 1b (Human) (E.C.3.1.3.48), Human (Homo Sapiens) Isc
Purified From An (Escherichia Coli) Overexpression System
uid 809208
accession 2HNP
medline
muid 94174273
abstract Protein tyrosine phosphatases (PTPs) constitute a family of receptor-li
cytoplasmic signal transducing enzymes that catalyze the dephosphorylation of
phosphotyrosine residues and are characterized by homologous catalytic domai
crystal structure of a representative member of this family, the 37-kilodalton for
1 to 321) of PTP1B, has been determined at 2.8 A resolution. The enzyme con
single domain with the catalytic site located at the base of a shallow cleft. The pl
recognition site is created from a loop that is located at the amino-terminus of ar
helix. This site is formed from an 11-residue sequence motif that is diagnostic of
the dual specificity phosphatases, and that contains the catalytically essential cys
arginine residues. The position of the invariant cysteine residue within the phospl
site is consistent with its role as a nucleophile in the catalytic reaction. The struct
PTP1B should serve as a model for other members of the PTP family and as a f
for understanding the mechanism of tyrosine dephosphorylation.
2. **title** Mol_id: 1, Mol_id: 1; Organism_scientific: Fusarium Oxysporum; Organism_common: N
uid 1310925
accession 1TRY
medline

Figure 6. Medline report


CLUSTAL W(1.60) multiple sequence alignment

user —MGKTIGIDLGTTNSCVAIMEGNQVKVIENSEGTRTTPSIIAYMDDNEVLVGAPAKRQS
1169373 —MGKTIGIDLGTTNSCVAIMEGNQVKVIENSEGTRTTPSIIAYMDDNEVLVGAPAKRQS
123648 MSKGPAVGIDLGTTYSCVGVFQHGKVEIIANDQGNRTTPSYVAFT-DTERLIGDAAKNQV
123647 MSKGPAVGIDLGTTYSCVGVFQHGKVEIIANDQGNRTTPSYVAFT-DTERLIGDAAKNQV

user VTNPKNTLFAVKRLIGRRFEEKEVQKDIGLMPYSIIKADNG-DAWVEGHGE-KLMAPPQV
1169373 VTNPKNTLFAVKRLIGRRFEEKEVQKDIGLMPYSIIKADNG-DAWVEGHGE-KLMAPPQV
123648 AMNPTNTVFDAKRLIGRRFDDAVVQSDMKHWPFMVVNDAGRPKVQVEYKGETKSFYPEEV
123647 AMNPTNTVFDAKRLIGRRFDDAVVQSDMKHWPFMVVNDAGRPKVQVEYKGEAKSFYPEEV

Figure 7 Multiple Sequence Alignment

View prosite scan result for the selected homologs

- **title** --------User Sequence--------
  uid user
  accession
  prosite
  - **name** PHOS_KINASE
    prositeid PS00000
    docid PDOC00000
    description Phosphorylase kinase. Taken from METHODS IN ENZYMOLOGY, 200:62-81
    pattern [KR]..S[VI]
    match
    1. 55
       KRQSV
  - **name** PKC
    prositeid PS00000
    docid PDOC00000
    description Protein kinase C. Taken from METHODS IN ENZYMOLOGY, 200:62-81
    pattern [KR].{1,2}[ST].{0,1}[KR]{0,1}
    match
    1. 34
       RTTP

Figure 8. Prosite Scan

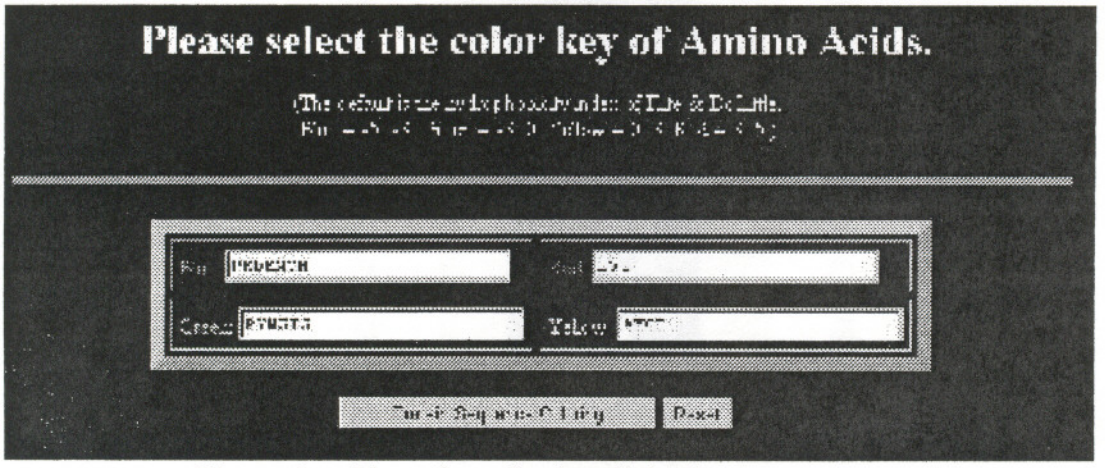


Figure 9a. The color selection dialog box,
with the standard color setting for hydrophobicity profile

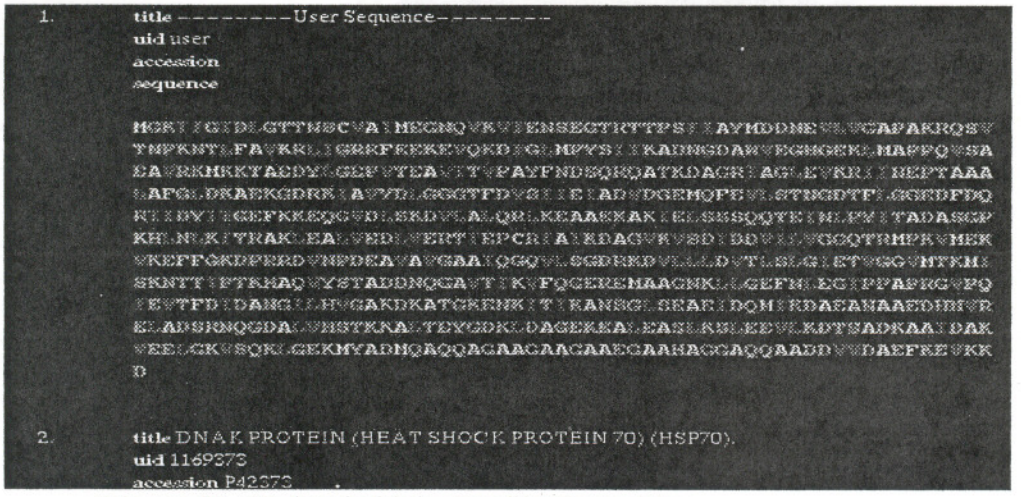


Figure 9b. Hydrophobicity Profile Generation

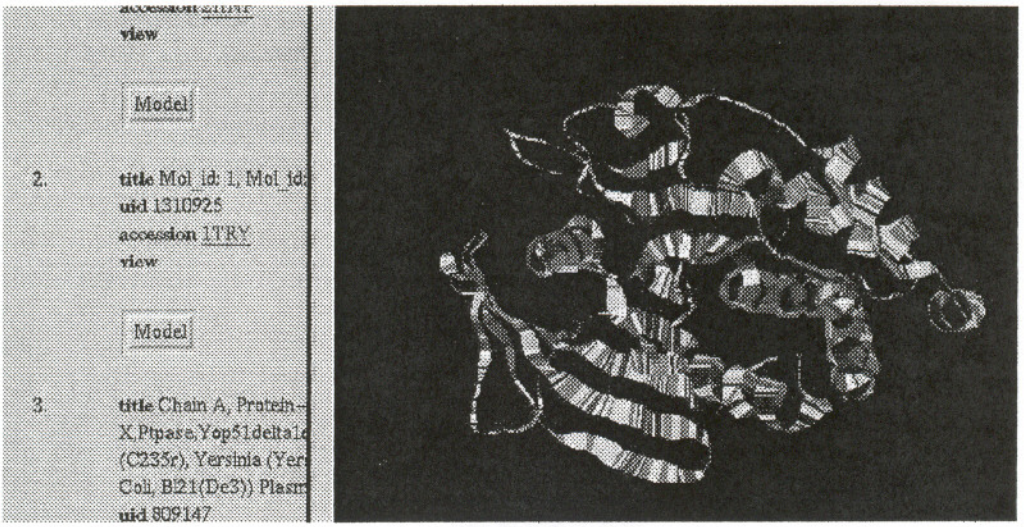


Figure 10. 3-Dimensional Structure

Section 4. Conclusion

In conclusion, we believe that a protocol approach is better suited for analyzing the high-volume bio-information being generated today, than the traditional functional approach to computer software. Using the particular instance where a wet-lab end-user biologist has the routine functionally-oriented question - "Now that I have a DNA or protein sequence, what can I do with it using a computer?" -, the protocol approach re-phrases it to "Now that I have a DNA or protein sequence, what can a computer do for me ?". For this question, in this work, we have illustrated that the computer can indeed do a lot with little expertise and hardly any end-user intervention. Our tool runs many predictive computer programs that can naturally be applied to both the original user-input data and the successive data generated by the protocol itself, searches the published literature, queries the public databases and continually integrates it all into a cohesive and custom sequence-structure-literature data warehouse. This general philosophy can be easily copied for other popular end-user bench-biologist questions.

Acknowledgment:

BioInformatics Center is funded by the Economic development Board of Singapore.

S. Subbiah. would like to acknowledge DOE grant no. DE-FG03-95ER62135 for partial support.

References:

1. Kiong B.K, Tan T.W, Meena K Sakharkar, D Yap (1996) "*BioInformatics and BioComputing on Internet: Treasure trove or Wilderness Wander?*" (poster presentation), BioInformatics and the Effective Use if the Internet for Rapid Drug Discovery, September 19-20, 1996, IBC, Seattle.

2. S. B. Davidson, C. Overton, V. Tannen and L. Wong *BioKleisli: A Digital Library for Biomedical Researchers* Journal of Digital Libraries 1:1, November 1996.
3. K. Hart, L. Wong, C. Overton, P. Buneman. *Using a Query Language to Integrate Biological Data.* Abstracts of 1st Meeting on the Interconnection of Molecular Biology Databases, Stanford, August 1994.
4. P. Buneman, K. Hart, L. Wong. *Answering some ``Unanswerable'' Biological Queries.* Poster presented at ACM Workshop on Information Retrieval and Genomics, Bethesda, May 1994.
5. http://info.gdb.org
6. GenBank Genetic Sequence Data Bank, Bilofsky et al., Nucl. Acids Res. 14(1); 1-4 ,1986
7. http://www.ncbi.nlm.nih.gov/BLAST/
8. http://www.ch.embnet.org/GCGdoc/Data_Files/prosite.html
9. J.D. Thompson, D.G. Higgins and T.J. Gibson (1994) *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice.* Nucleic Acids Research, 22:4673-4680.
10. http://www.ncbi.nlm.nih.gov/Entrez/
11. http://www.pdb.bnl.gov/PPS/rasmol/rasmol.html
12. http://www.gcg.com
13. http://adenine.iss.nus.sg:8080/examples/blast/index.html
14. http://www.pdb.bnl.gov