DEVELOMENT OF A PROTEOMICS-BASED SEARCH ENGINE TOOL

HERNANDO GUSTAVO SUAREZ DURAN

EAFIT UNIVERSITY

DEPARTMENT OF SCIENCES AND HUMANITIES

PHYSICS ENGINEERING

MEDELLIN

2012

DEVELOMENT OF A PROTEOMICS-BASED SEARCH ENGINE TOOL

HERNANDO GUSTAVO SUAREZ DURAN

TUTOR:

Dr. OSCAR ALZATE

The University of North Carolina at Chapel Hill (UNC)

CO-TUTOR:

CAROLINA LONDOÑO, M.Sc.

Universidad Pontificia Bolivariana (UPB)

EAFIT UNIVERSITY

DEPARTMENT OF SCIENCES AND HUMANITIES

PHYSICS ENGINEERING

MEDELLIN

2012

ACKNOWLEDGEMENTS

The author expresses his acknowledgements to Dr. Oscar Alzate, Ms. Carolina Londoño and Juan Manuel Jaramillo for their direction and support throughout the project and to EAFIT University for the use of the equipment necessary for the development of this project.

CONTENTS

FIGURES LIST

Page

TABLES LIST

GLOSSARY

ACCESSION NUMBER: A unique identifier given to a protein when is submitted to one of the proteome databases [1]. It may consist of numbers and letters.

ELECTRIC CHARGE: A property possessed by some subatomic particles. Usually associated to the electrical elemental charge of the electrons, which by convention is negative. When an atom has additional electrons, the atom acquires a negative charge, and when it has less than the necessary to be in a neutral state, then the atom acquires a positive charge.

This property is also present in molecules, in which the charge is the sum of the charge of its atoms or functional groups.

ELECTRIC FIELD: A field which surrounds an electric charge. It denotes the space in which an electric charge has influence around it. Electric fields can be created either by an electric charge or a magnetic field in variation. The interaction between electric fields and electric charges is by a force known as electric force.

POSTTRANSLATIONAL MODIFICATION OF PROTEINS: After a protein has been translated, certain enzymes interact with it and modify it to create specific modifications to some amino acid residues [2]. This process is called posttranslational modification (PTM), and it is thanks to it that several forms of the same protein (isoforms) exist within an organism: which in turn provides them with diverse roles according to the organism requirements [3]; i.e., the same protein can have several functions depending on the specific modifications.

At least 700 different types of PTMs have been reported [3], some of which are reversible while others are not. Some of the most common PTMs are phophorylation, glycosylation, ubiquitination, methylation, acetylation, sulfation, sumoylation, myristoylation and palmitoylation [3].

The importance behind the research on PTMs resides mainly on the quest to find how proteins are regulated within an organism and how they perform their functions [3]. This research may help to understand how the mechanisms of disease work [4], [5].

PROTEIN: Proteins are polypeptides of amino acids, which are joined to each other via peptide bonds [2]. They are essential parts of an organism since they participate in virtually every process within the cells, some of which are: catalyzing reactions, transporting molecules, providing movement and others [2].

They have been found to be determinant factors in the appearance of any disease, including Alzheimer's disease [6] and multiple sclerosis [7]. Their study has become of great interest to the scientific community in hopes of understanding human diseases [8], [9].

PROTEOME: The entire set of proteins expressed by the genome of an organism, including all the possible modifications, and the protein-protein interactions [10]. Because proteins change their shape depending on the cellular cycle, external actions and processes like PTM, the proteome is very dynamic and complex [10].

STRUCTURE (in MATLAB): A *variable* type in the MATLAB environment which consists of arrays with named fields and allows storage of data of varying types and sizes within it [11].

TRANSLATION: The process in which a protein is synthesized; this process is divided in two parts. The first part of the process is called transcription, and in this, RNA is created from DNA [2]. This RNA shares some similarities with the DNA it originally comes from, and it also differs in that one of the bases of the DNA, thymine, is changed to another, uracil, in the RNA [2]. The next process is the translation, in which RNA is read and used as a template to assemble the sequence for a protein [2].

TWO-DIMENSIONAL DIFFERENTIAL GEL ELECTROPHORESIS (2-D DIGE): A process that separates proteins suspended in a gel for their detection and analysis and generally requires five steps [12], which are the sample preparation, the first separation, an interface, the second separation, and the protein detection. Both separations are performed by applying an electric field to the gel, which in turn makes the proteins move through the gel according to their electrical charge [12].

ABSTRACT

In an effort to facilitate the understanding of the human proteome, two algorithms were developed using MATLAB as the programming language. The main idea was to increase the speed of data analysis when analyzing proteomics data. The algorithms produced an offline proteins database that has been reported as having at least one PTM within the *Homo Sapiens* database. The original information is extracted from two online databases that can be subsequently browsed in search of a specific protein. The resulting program can also be used to filter a number of proteins matching a specific criterion selected by the user.

Keywords: Databases, *Homo Sapiens*, Posttranslational modifications, Proteome, MATLAB.

INTRODUCTION

The understanding of the human genome and the mechanism of human diseases are currently challenges faced by the scientific community [13], and even though new genome-based theories to explain the origin of diseases are still being developed [14], a new approach to tackle these problems has been taken: the study of the proteome.

The proteome includes all possible forms of a protein along with some protein-protein interactions. It has been found that the proteome could be several orders of magnitude more complex than the genome [15]. The genome of an organism is made of several genes, which can create several proteins. The multiple forms that a protein could have, called isoforms, are the consequence of a process called a Post Translational Modification (PTM). PTMs affect the amino acids of a protein, and potentially, all amino acids of a protein can be modified by any type of PTM. There are several PTMs found, all of which can modify any amino acid and create a different isoform. Then, the number of isoforms a protein can have is very high.

The high complexity of the problem makes necessary the participation of other branches of applied science, including bioinformatics and applied mathematics, to facilitate the research on this subject. Mathematics is a useful tool for the development of mathematical models that could predict and analyze the behavior of complex problems, such as the dynamics of the human proteome. This work aims to provide the basis for a future mathematical model of the dynamics of the human proteome starting with an initial analysis of genome and proteome databases.

To sort this information, the scientific has developed different databases such as GenBank [16], RefSeq [17] and UniProt [18], in which all protein information is deposited and where researchers from all over the world can access information about proteins and some of their characteristics.

Some of these databases are curated, while some others are not. A curated database is that one in which a person with authority in the subject, called curator, validates the information that is displayed in the database to ensure its accuracy.

These databases include a large number of entries, each entry representing a different protein, which sometimes makes the consultants' job harder [19], since they cannot read the whole database when looking for the proteins that are important for their research.

Different databases, then, have multiple search criteria which make it easier for the researchers to look for the exact proteins they are interested in without the need to look at all the proteins the database can have.

This project aims to develop a computer tool to facilitate the researchers' job by analyzing and compiling multiple protein databases into a single one which then may be accessed via this computer tool. The goal is to use this tool to sort and classify proteins from the *Homo Sapiens* that have been reported as having a PTM.

1. LITERATURE REVIEW

1.1. PROTEIN DATABASES

A number of publicly available online proteome databases were studied in preparation of this work. They are showed below.

1.1.1. NCBI.

The National Center for Biotechnology Information (NCBI) of USA houses several databases, which include curated genomics and proteomics sequences data that are updated daily [20]. The NCBI's proteomics database is called "Protein", and includes a collection of sequences from several sources, including GenBank, RefSeq, SwissProt, PDB and others [20].

NCBI's Protein is among the largest protein databases. It provides the user with 25 different search criteria [21] and a quick search of all the proteins that belong to the *Homo Sapiens* organism shows 687,810 entries as of November 20, 2012 [22], whereas UniProtKB, shows only 131,321 entries under the same search on the same date [23].

1.1.2. UniProtKB.

This database is a collection of functional information on proteins with a special focus on having additional annotations in each entry [24] such as classifications of proteins, cross-references, in which the entries for the same protein in other databases, and a link to them if available, are displayed, and biological ontologies, in which information like PTMs, and molecular function are displayed. A feature of UniProtKB is that it allows for submissions of non-reviewed protein sequences computationally generated from a section of the database called "UniProtKB/TrEMBL" [24]. On the other hand, another section of UniprotKB,

“UniprotKB/SwissProt”, only contains high quality manually annotated and non-redundant protein sequences [24].

An important advantage of this database over the others is its large number of search criteria. It provides the user with 32 different criteria for searching proteins in the database [25].

1.1.3. PhosphoSitePlus.

This database is one of the biggest collections of proteins with PTMs [3]. It is a curated proteome library that focuses on proteins with a PTM. It was reengineered from PhosphoSite [26], and unlike its predecessor, it’s not limited to phosphorylation.

This database provides the user with 7 different search criteria [27], and as of November 20, 2012, it has at least 18,755 proteins. An important feature of this database for the purpose of this project is that all of the entries include the protein’s molecular weight and isoelectric point.

1.1.4. PHOSIDA.

This database was initially a database for phosphorylated proteins [3], however it now contains entries associated with other PTMs such as acetylation and N-glycosylation [28]. It is regularly updated using data from other databases such as UniProt/SwissProt [3].

PHOSIDA provides the user with 3 different proteins search criteria [29] and as of November 20, 2012, it houses more than 25,000 protein entries [30].

1.1.5. Phospho.ELM.

This is a curated database of eukaryotic proteins [31]. All the entries contain a vast variety of information and it has the largest collection of proteins displaying at least one phosphorylation [3].

This database provides the user with 7 search criteria [32].

2. MATERIALS AND METHODS

2.1. NCBI / REFSEQ JOINT DATABASE

2.1.1. Gene Database. The NCBI provides a database called Gene (http://www.ncbi.nlm.nih.gov/gene), which is easily searchable, focuses on genomes that have been completely sequenced, provides enough pertinent information in an organized format and is updated daily (http://www.ncbi.nlm.nih.gov/books/NBK3841/#EntrezGene.Quick_Start).

Gene presents the data on a table-based format with rows representing entries, in this case different genes, and columns representing different types of information. **Table 1** shows an example of data containing three genes. **Table 2** shows the meaning of each column in this database.

Table 1. Three entries of the Gene database are shown. **(a)** Shows columns 1-5, **(b)** shows columns 6-9, **(c)** shows columns 10-13, **(d)** shows columns 14-15.

Taxonomy ID	GeneID	Default Symbol	LocusTag	Synonyms
9606	1	A1BG	-	A1B|ABG|DKFZp686F0970|GAB|HYST2477
9606	2	A2M	-	A2MD|CPAMD5|DKFZp779B086|FWP007|S863-7
9606	3	A2MP1	-	A2MP

(a)

Other Databases [xdatabase:value]	Chromosome	Map Location	Description
HGNC:5|MIM:138670|Ensembl:ENSG00000121410|HPRD:00726	19	19q13.4	alpha-1-B glycoprotein
HGNC:7|MIM:103950|Ensembl:ENSG00000175899|HPRD:00072	12	12p13.31	alpha-2-macroglobulin

HGNC:8	12	12p13.3-p12.3	alpha-2-macroglobulin pseudogene 1

(b)

Type of Gene	Official Symbol	Full Name from Nomenclature Authority	Nom. Status
protein-coding	A1BG	alpha-1-B glycoprotein	O
protein-coding	A2M	alpha-2-macroglobulin	O
Pseudo	A2MP1	alpha-2-macroglobulin pseudogene 1	O

(c)

Other Designation	Modification Date
alpha-1B-glycoprotein	20110820
C3 and PZP-like alpha-2-macroglobulin domain-containing protein 5|OTTHUMP00000196829|alpha-2-M	20110819
-	20110803

(d)

Table 2. Explanation of each column in the Gene database.

Taxonomy ID	The unique identifier provided by NCBI Taxonomy for the species or strain/isolate
GeneID	The unique identifier for a gene
Symbol	The default symbol for the gene
LocusTag	The LocusTag value
Synonyms	Set of unofficial symbols for the gene
Other Databases	Identifiers in other databases for this gene
Chromosome	The chromosome on which this gene is placed. For mitochondrial genomes, the value 'MT' is used
Map location	The map location for this gene
Description	A descriptive name for this gene
Type of Gene	The type assigned to the gene
Official Symbol	When not '-', indicates that this symbol is from nomenclature authority
Full Name From Nomenclature Authority	When not '-', indicates that this full name is from a nomenclature authority

Nomenclature Status	When not '-', indicates the status of the name from the nomenclature authority (O for official, I for interim)
Other Designations	Some alternate descriptions that have been assigned to a GeneID
Modification Date	The last date a gene record was updated, in YYYYMMDD format

The Gene database provides genetic information for a wide range of species, making it a very large database (132MB as of to date). Since this study is focused on human genes, only human entries were used. Gene allows downloading complete databases, and also smaller ones consisting of certain classes, such as mammals, or species, such as the homo-sapiens. These files are smaller (2.2MB as of to date) and easier to manage. For this work we downloaded the homo sapiens-associated gene database (ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/Homo_sapiens.gene_info.gz).

2.1.2. Protein Database. Two options were considered for protein databases, GenBank (ftp://ftp.ncbi.nlm.nih.gov/genbank/) and RefSeq (ftp://ftp.ncbi.nih.gov/refseq/). Both databases provide the information using very different formats and different amounts of information, while RefSeq is more compact, it only provides four types of data for each entry including the protein's sequence, whereas GenBank provides more information per entry resulting in a bigger database and therefore, a larger file (83.1MB). Both databases differ also in the way they work, for example RefSeq is curated and revised by the NCBI [x2] and limits its entries only to model organisms, while GenBank is not curated, any author can submit and revise data and doesn't have a limit to species included [x3].

In the early stages of the work, both databases were used, however the large files from GenBank did not allow a fluid and easy work; therefore RefSeq was selected as the only protein database for this study.

RefSeq database's format is the FASTA format, which is a widely used text-based format for nucleotides and peptide sequences. In the RefSeq database, each entry is spread in at least two rows, and one column of text. **Table 3** shows two proteins listed from the RefSeq database in FASTA format. The first row starts with the character (>), which denotes the start of a new protein. This row also contains different types of information and their value, which are separated with the vertical bar (|). In this case “gi” is followed by a separator and the protein’s unique identifier, while “ref” is followed by a separator and the protein’s accession.version. The information after the last separator is the protein’s name and, between brackets, the species. The following rows contain the amino acids sequence of the protein. **Table 4** shows the meaning of each character of the protein sequence in FASTA format.

Table 3. Two entries of the RefSeq database in FASTA format.

<table>
<tr><td>>gi|157412240|ref|NP_001094800.1| C1GALT1-specific chaperone 1-like [xHomo sapiens]</td></tr>
<tr><td>MVSASGTSFFKGMLLGSISWVLITMFGQIHIRHRGQTQDHEHHHLRPPNRNDFLNTSKVILLELSKSIRVFCIIFGESED</td></tr>
<tr><td>ESYWAVLKETWTKHCDKAELYDTKNDNLFNIESNDRWVQMRTAYKYVFEKYGDNYNWFFLALPTTFAVIENLKYLLFTRD</td></tr>
<tr><td>ASQPFYLGHTVIFGDLEYVTVEGGIVLSRELMKRLNRLLDNSETCADQSVIWKLSEDKQLAICLKYAGVHAENAEDYEGR</td></tr>
<tr><td>DVFNTKPIAQLIEEALSNNPQQVVEGCCSDMAITFNGLTPQKMEVMMYGLYRLRAFGHYFNDTLVFLPPVGSEND</td></tr>
<tr><td>>gi|221136769|ref|NP_001137536.1| hypothetical protein LOC100129239 [xHomo sapiens]</td></tr>
<tr><td>MAKVTSEPQKPNEDVDEQTPSTSSTKGRKKGKTPRQRRSRSGVKGLKTTRKAKRPLRGSSSQKAGETNTPAGKPKKARGP</td></tr>
<tr><td>ILRGRYHRLKEKMKKEEADKEQSETSVL</td></tr>
</table>

Table 4. Meaning of each character in a protein sequence in FASTA format.

Amino Acid Code	Meaning
A	Alanine
C	Cysteine
D	Aspartic acid
E	Glutamic acid
F	Phenylalanine
G	Glycine
H	Histidine
I	Isoleucine
K	Lysine

L	Leucine
M	Methionine
N	Asparagine
O	Pyrrolysine
P	Proline
Q	Glutamine
R	Arginine
S	Serine
T	Threonine
U	Selenocysteine
V	Valine
W	Tryptophan
Y	Tyrosine
X	any
*	translation stop
-	gap of indeterminate length

As in the Gene database, the NCBI databases web page provides a protein database containing information for homo-sapiens only (ftp://ftp.ncbi.nih.gov/refseq/H_sapiens/mRNA_Prot/human.protein.faa.gz) in a manageable file size (8.8MB).

2.2. VARIABLES OF STUDY

Because many PTMs alter either the isoelectric point and/or the molecular weight of the protein being modified, the algorithm developed here includes the isoelectric point and molecular weight as variables to develop the algorithm. Due to their importance in identifying the protein, accession number, name and protein sequences were also chosen as desired information in the database.

The following is a list with the information that each protein entry has in the database:

- Accession number.
- Definition/Name.
- Protein Sequence.
- PTM information (as reported in NCBI's Protein).
- Molecular weight.
- Basal isoelectric point.
- Link to the protein's entry in the NCBI' database.
- Link to the protein's entry in the PhosphoSitePlus' database.
- Link to the protein's entry in the UniProtKB' database.

2.3. DATABASES

Due to its large number of entries and the large amount of information for each entry, NCBI's Protein was chosen as the primary database for this work. As a secondary database, PhosphoSitePlus was chosen because it was curated, has a large number of entries and each of them contains the desired data.

Both databases are easily accessible online using a web browser with internet connection and both offer clean interfaces with multiple search criteria for the user [21], [27].

2.4. ENVIRONMENT

Due to its powerful programming language and the author's knowledge and proficiency on the software, the computing environment used for this study was MATLAB®. All of the following results were compiled in MATLAB 7.10.0 (R2010a).

3. ALGORITHMS DEVELOPMENT

3.1. READING INTERNET DATA

MATLAB contains a "built-in" tool called "urlread" which downloads URL content to a "string" [33], which is a variable-type in the MATLAB environment used to store text as an array of characters [34]. To use the "urlread" command, the user needs to provide a working URL and a variable name to store the data in. The data retrieved by the "urlread" command are the source code for the URL in HTML format.

This posed the first challenge for the development of the algorithm: looking for the URL containing the data that would be transferred into the database that the algorithm was expected to build. The finding of these URLs is described below.

3.1.1. NCBI. NCBI's Protein main search tool was analyzed first. The URL pathway for this tool is,

"http://www.ncbi.nlm.nih.gov/protein/?term="

By copying and pasting this URL into any web browser (without the quotation marks), and adding a search query in the end, after the equals sign, the browser opens up the results corresponding to this query as if you have had entered the NCBI's main webpage and performed the search manually.

This list of results is presented as a list of protein names that act as links which direct you to each protein's entry within the database, and all the entries' data are actually displayed.

By exploring the web further, it was found that NCBI's Protein was also searchable with another search tool built in NCBI's search engine. It's called "*Sequence Viewer: Protein*", and the URL pathway is, “http://www.ncbi.nlm.nih.gov/sviewer/batchseq.cgi?db=protein&query_key=1&term=”

By copying and pasting this string and adding the search query to the end, the browser opens the exact same list of results as the main search tool with one important difference: all the entries that match the user’s search query are shown in their entirety, including all the data for each entry, instead of just links as the main search tool results.

By using a small algorithm that read the results of both search tools with the command “urlread”, it was evident that using the second search tool was more adequate for this project, since it allowed the algorithm to read more entries per page read. This is because the results page of the *Sequence Viewer* displays the information of a number of proteins immediately, whereas using the main search tool would require entering each entry in the database separately to download the corresponding information.

Since this work is centered around *Homo Sapiens* and PTMs, the search query input was the “homo sapiens[organism]” and “ptm”. The instruction to perform this search is,

“http://www.ncbi.nlm.nih.gov/sviewer/batchseq.cgi?db=protein&query_key=1&term=homo%20sapiens%5BOrganism%5D%20and%20ptm”

The output from this URL search using the strings presented above shows ten results per page, and we are expected to read a couple of thousands of entries, so to speed-up the URL reading process, additional specifications were added to the string to display 500 entries per page. The instructions to perform the search with all the specifications described above is,

“http://www.ncbi.nlm.nih.gov/sviewer/batchseq.cgi?db=protein&query_key=1&term=homo%20sapiens%5BOrganism%5D%20and%20ptm&page_size=500&fmt_mask=0&report=genpept&page=”

By copying and pasting this string and, an internet browser displays 500 proteins that have been reported as being of the *Homo Sapiens* organism and have at least one PTM. After the equals sign at the end of the string, a number can be added, which indicates which page of the results to show; adding no number displays the same results as adding the number “1”, which shows first 500 proteins to match the query, adding “2” shows proteins 501- 1000, and so on.

This is the string that the algorithm uses to download the first part of the data to build the database.

3.1.2. PhosphoSitePlus. Unlike NCBI's Protein, PhosPhositePlus does not have the option to search in the database without selecting a specific search criterion first. Since from the first database search, the accession number for each protein was stored, using the accession number search criterion was adequate for this.

A string to explore the URL that performed an accession number search within PhosphoSitePlus is,

"http://www.phosphosite.org/proteinSearchSubmitAction.do?accessionIds="

By copying and pasting this string and adding an accession number at the end in an internet browser shows the protein that matches the accession number entered; however, it does not show the information and characteristics of the protein. Instead, the page displays a number of links, each representing the same protein for different organisms.

By following the link for the human protein, the protein's entry in the database is shown, displaying the data that the PhosphoSitePlus database contains for such protein. The string format for each protein's entry is then,

"http://www.phosphosite.org/proteinAction.do?id="

By copying and pasting this string and adding at the end, protein's ID, in an internet browser, the entry for the protein initially searched for is displayed. The protein's ID number is fetched from the previous URL by locating the link the *Homo Sapiens* protein and copying it from there.

These last two strings are the ones used by the algorithm. The first one to locate the protein's ID, and the second to retrieve the second part of data to populate the database.

3.2. LOCATING DATA

Having located the strings that once placed in an internet browser would show the information needed to build the database, it is important to know where exactly in the HTML source code such data are located. This process helps eliminating data that are not necessary for this analysis.

To do this, the HTML source code for several entries was analyzed with an internet browser until specific “tags” were identified. These tags are pieces of code within the HTML source code for each entry page and indicate where each piece of data starts and where it ends. They are later used to instruct the algorithm where to start reading and where to stop. These tags are shown in **Table 5**.

Table 5. Tags indicating where each piece of data begins and end within the HTML source code.

	Database	**Beginning Tag**	**Ending Tag**
Accession number	NCBI's Protein	ACCESSION	VERSION
Definition/Name		DEFINITION	ACCESSION
Sequence		</a> 1	//</pre>
PTM information		[PTM]	[
Molecular weight	PhosphoSitePlus	Molecular weight:&nbsp;</span>	&nbsp;Da
Basal Isoelectric point		Basal Isoelectric point:&nbsp;</span>	&nbsp;

4. RESULTS AND DISCUSSION

4.1. NCBI / REFSEQ JOINT DATABASE

Two algorithms were written and compiled in MATLAB to work one after the other. A summary of the whole process is displayed in **Fig. 1**.

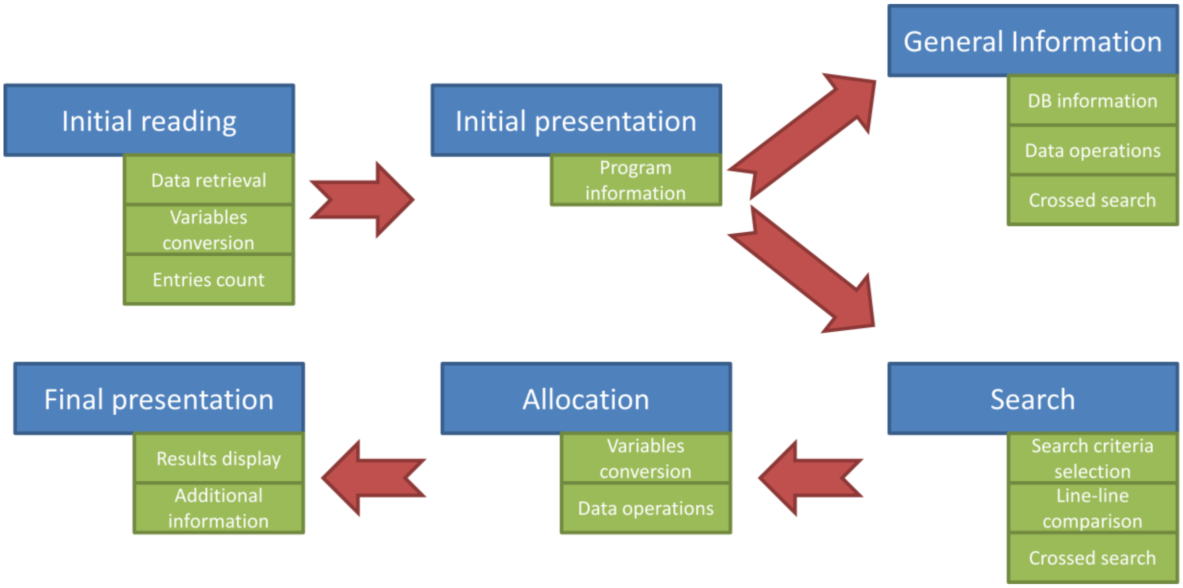


Fig. 1. Graphical representation of the working algorithm.

A detailed explanation of each step is now presented.

4.1.1. Data Retrieval. The database files were imported directly into MATLAB's workspace with an algorithm using MATLAB's "fastaread" command for RefSeq and the command "Importdata" for Gene and assigned into a single structure for easy reading. The data are imported directly into the workspace to avoid loading the databases each time the main algorithm runs, and thus, saving time for the user.

4.1.2. Conversion of Variables. Initially both databases were imported using a variable type called “structure”. This variable has a matrix-like conformation which in turn may contain other matrixes inside. After loading, most data go through several conversions to facilitate working their use. **Fig. 2** shows the main conversions performed by the algorithm.

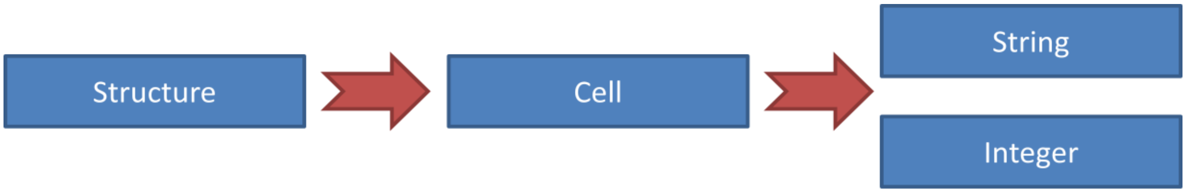


Fig. 2 Databases are imported into a variable type called “structure” which are like special matrixes that can contain other matrixes. Each structure is converted into arrays of cells, which are more like simple matrixes. If the cell contains numeric data, the algorithm converts into an integer, which is simple numeric variable. If the cell contains text data, the algorithm converts it into strings, which are character-based variables that may contain anything from a single letter to several words.

4.1.3. Entries Count. The number of entries on both databases needs to be counted from the beginning. For the Gene database this is simple because each row represents a unique entry; however for the RefSeq database this is not the case because the number of rows occupied by a single entry depends on the length of its protein’s sequence. MATLAB has built-in commands that extract information from files in FASTA format, and one of them counts the number of entries in the database by counting the number of appearances of the character (>).

4.1.4. Program information. Since the algorithm has two main functions, a branch is presented depending on the user’s objectives. The most important and complex function is to search through the databases, compare one to the other and present relevant information. The other function presents general information on both databases..

4.1.5. Databases information. The total number of genes and proteins in both databases are presented, along with the last modifications dates.

4.1.6. Data operations. The number of protein-coding genes in the database was counted. The number of total amino acids per type in the database was counted and displayed along with the total number of amino acids of all types. In principle, it is expected that these numbers represent the total number of amino acids in a homo-sapiens. **Table 5** reports these results.

Table 6. Analysis of the databases.

Total genes	42158
Protein-Coding genes	20462
Total proteins	32378
Total amino acids	18273382
Alanine	1272404
Cysteine	409000
Aspartate / Aspartic acid	879051
Glutamate / Glutamic acid	1312069
Phenylalanine	662270
Glycine	1191963
Histidine	477079
Isoleucine	799230
Lysine	1057326
Leucine	1801166
Methionine	394491
Asparagine	666210
Proline	1147232
Glutamine	872776
Arginine	1031411
Serine	1517963
Threonine	977104
Selenocysteine	75

Valine	1096548
Tryptophan	220738
Tyrosine	487194
Any	82

Although the main function of genes is protein-coding, a large number of genes do not have lost this protein-coding ability or are no longer expressed. These genes are called pseudogenes and make up what is known as the noncoding DNA. The Gene database lists these genes as well and the number of how many protein-coding and noncoding genes are present is counted as well, which is shown as well in **Table 5.**

Some predicted proteins which haven't been fully sequenced are usually presented in FASTA format with several appearances of the character (X), which in FASTA format represent that this position can be occupied by any other amino acid. In the RefSeq database 82 appearances of the character (X) were found, all of them in sequences of predicted and hypothetical proteins such as LOC642424.

4.1.7. Search criteria selection. The user selects from five search criteria which in turn, leads the program to decide which database to search on first. **Fig. 3** shows each search criteria and its corresponding database.

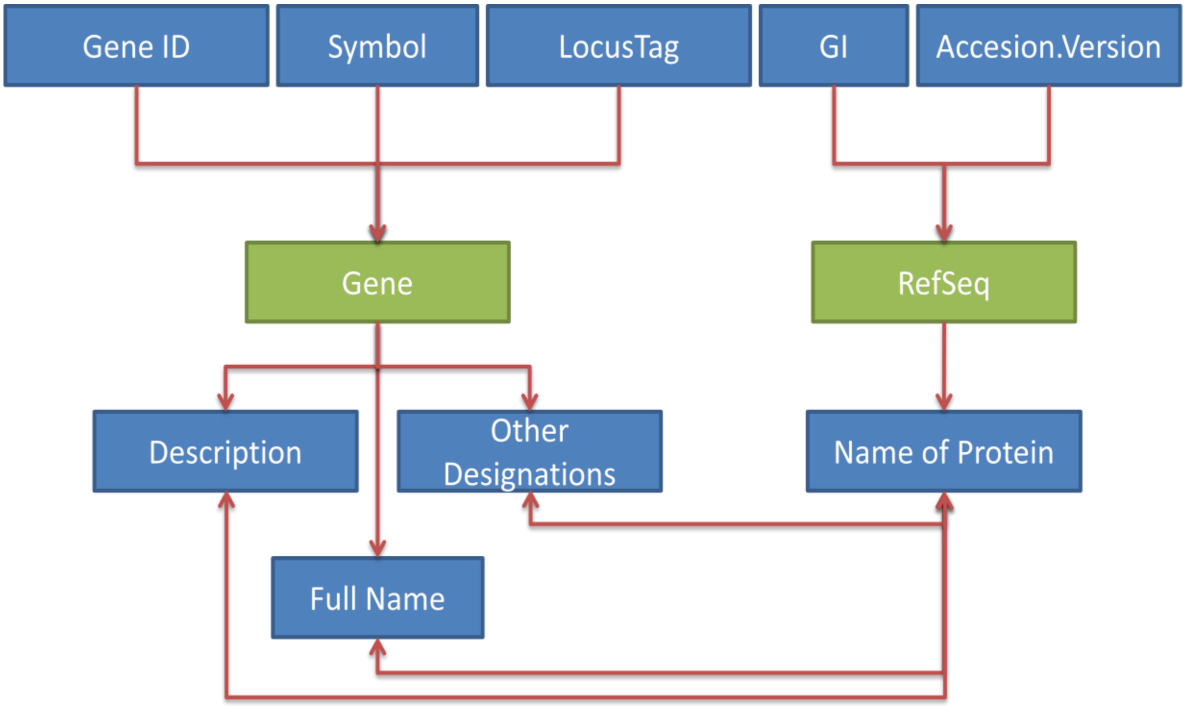


Fig. 3. The five boxes on top represent the five search criteria while the two middle ones the databases they lead to. The four boxes below the databases represent the information extracted from each database that can be used to cross-search and compare both databases.

4.1.8. Line-Line Comparison. The search method is a simple line-line comparison among the user's input and each entry of the selected database. The user's input is assigned in one variable and, depending of the search criteria selected, the program row of the corresponding column and compares the both. Numeric data is compared directly with conditionals, however comparing text data, strings, a two step process is involved: first size is compared, which accounts for the number of characters the input has, if equal, then both texts are compared to see if they match. This two step process is justified in the fact that MATLAB isn't capable of comparing two strings of different size.

4.1.9. Crossed Search. Connecting Gene to RefSeq to determine which gene coded for a specific protein is essential for the future development of a model of the proteome. The information provided in both databases that can be used for this is, in RefSeq, the name of the protein, which in Gene may appear either in the "description", the "full name", the "other designations" or the "symbol" fields. **Fig. 3** displays the overall method for cross-search.

If the user searches for a specific protein, the algorithm searches for its corresponding gene after the protein has been found; likewise, if the user searches for a specific gene, the algorithm it searches for its corresponding protein after the gene has been identified.

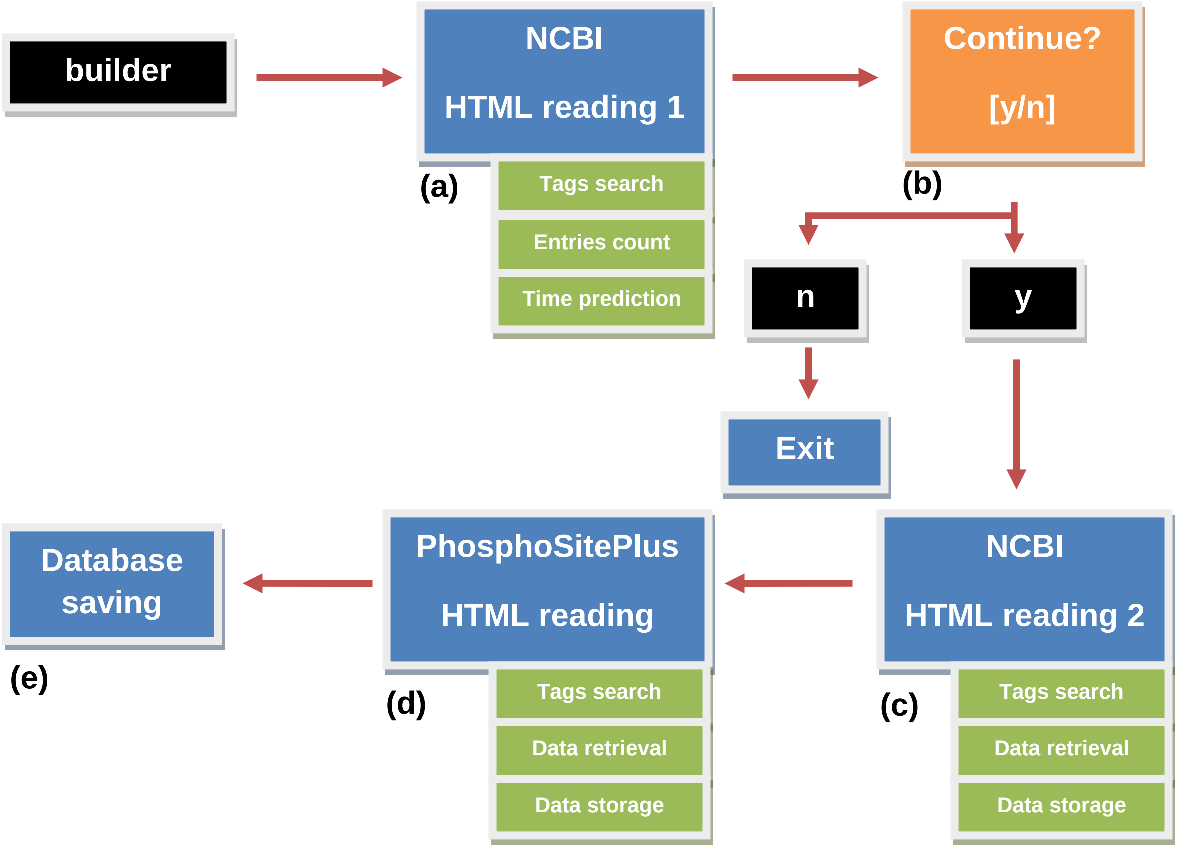
The cross-search component of the algorithm may also be used when the user is only searching for the database information. The algorithm takes every gene in Gene, and searches for a corresponding protein in the RefSeq and counts how many genes are correlated with proteins. This analysis constitutes the most important part of this work because the correlation between genes and expressed proteins can provide information about the behavior of the system; this still a work in progress and the aim for future developments.

4.1.10. Final Presentation. The algorithm shows pertinent information found in the databases and asks for user input to decide whether or not to show additional information including the protein's amino acid sequence (if it was found), and the latest modification date for such information.

4.2. SUMMARY

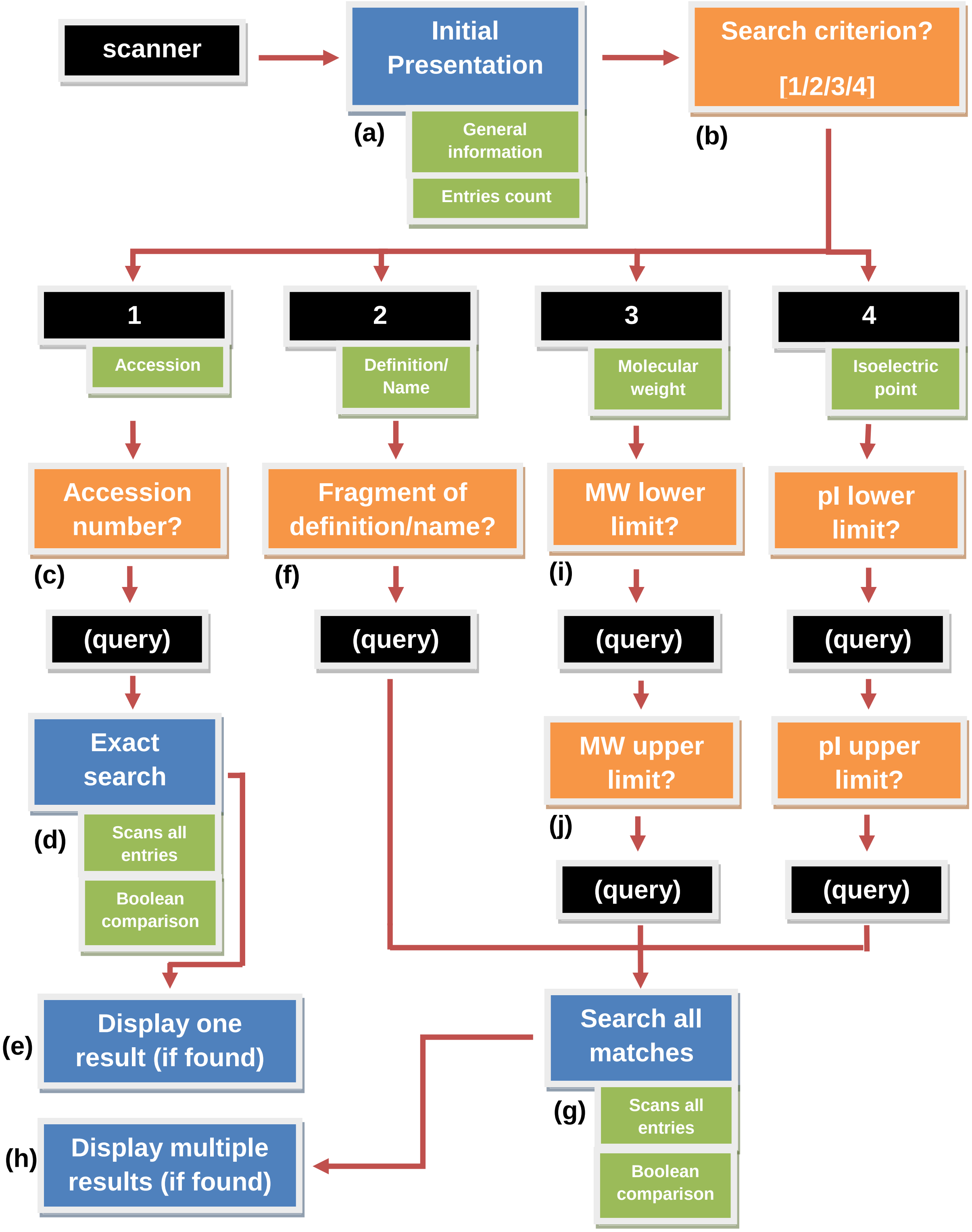
Two algorithms were developed by me in MATLAB's programming language. They are meant to work one after the other; the first one was named "Builder" by the author, and it has the purpose of building the joint database. The second algorithm, named "Scanner", is the one that will work as search engine for the previously built database. A summary of how both algorithms work is displayed in **Figs. 4** and **5**.

Fig. 4. Graphical representation of the "Builder" algorithm. Red arrows indicate the path the algorithm follows, black boxes indicate the input the user have to enter to follow the path, orange boxes indicate questions the algorithm displays when asking for user input, blue boxes indicate processes within the algorithm and green boxes indicate some specifics regarding the closest bigger box.



The user manual for both algorithms is presented in Annex A.

Fig. 5. Graphical representation of the "Scanner" algorithm. Red arrows indicate the path the algorithm follows , black boxes indicate the input the user have to enter to follow the path, orange boxes indicate questions the algorithm displays when asking for user input, blue boxes indicate processes within the algorithm and green boxes indicates some specifics regarding the closest bigger box.



4.3. THE BUILDER ALGORITHM

This algorithm builds the database that we are trying to develop using as input information from publicly available databases. It is necessary to have a working internet connection to run it.

A step by step explanation of the algorithm is presented below.

4.3.1. NCBI HTML Reading 1. In this step, shown in **Fig. 4 (a)**, the algorithm performs a search in the NCBI's Protein database for all the proteins reported belonging to the *Homo Sapiens* and having at least one PTM; The HTML source code of the results page is saved in a local variable. The algorithm then scans the whole HTML code in search of the total number of proteins found by this search.

The algorithm then displays this information and asks if the user wants to continue building the database, as shown in **Fig. 4 (b)**.

4.3.2. NCBI HTML Reading 2. In this step, shown in **Fig. 4 (c)**, the algorithm cycles all the HTML source code in the results page and locates all the beginning and ending tags corresponding to NCBI's Protein data, which are accession number, definition/name, sequence and PTM information. After locating all the tags, the algorithm reads and stores the information in a variable type known as a *structure*.

4.3.3. PhosphoSitePlus HTML Reading. In this step, shown in **Fig. 4 (d)**, the algorithm searches all the accession numbers, which were previously locally stored in a *structure*, in PhosphoSitePlus. After entering the main page for the desired entry in the database, it locates in the HTML source code all the beginning and ending tags corresponding to this database, which are molecular weight and basal isoelectric point. After locating all the tags, the algorithm reads and stores the information in the same *structure* used in the previous step. Now we have a *structure* which has all the information that the database requires.

4.3.4. File Saving. In this step, shown in **Fig. 4 (e)**, the algorithm writes the resulting information into a database in a file with the name “Database.mat”. This file, unlike the *structure* in which the data was being stored in the previous steps, will be available after the algorithm finishes running. The file will be located within the MATLAB folder in our local machine.

4.4. THE SCANNER ALGORITHM

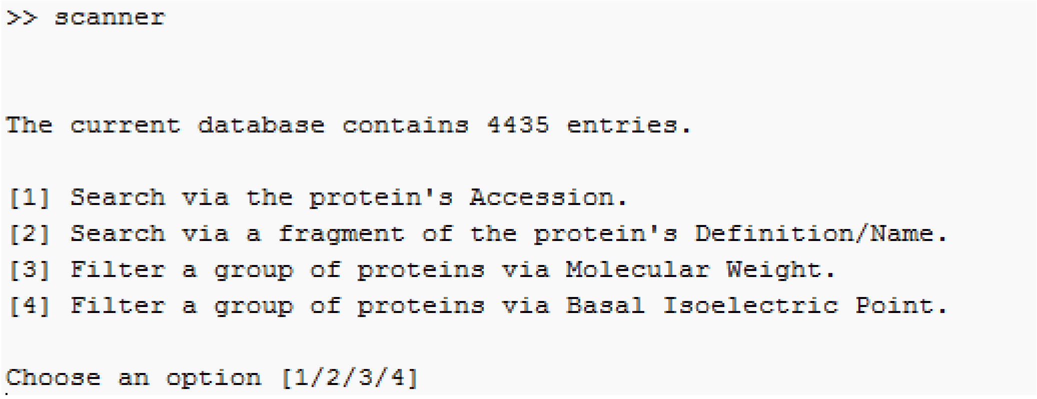
The function of this algorithm is to search within the database that was built by the “Builder” algorithm, according to four different search criteria that the user may select from.

“Scanner” starts by accessing the file “Database.mat”, which must be stored in the MATLAB folder in our local machine. This means that the algorithm “Builder” must had been run at least once before, to ensure that a starting database was created. This algorithm can be run offline and is not necessary to run “Builder” every time “Scanner” is going to be run, provided that the database was successfully developed and stored in our local machine.

A step by step explanation of the algorithm is presented below.

4.4.1. Initial Presentation. In this step, shown in **Fig. 5 (a)**, the algorithm reads the file that “Builder” saved in the MATLAB folder (database.mat) and displays the current number of entries in the database and the four options available to the user, each one representing one of the four search criteria. A screenshot of this step can be seen in **Fig. 6**.

Fig. 6. A screenshot of the **Initial Presentation** after “Scanner” has been run.



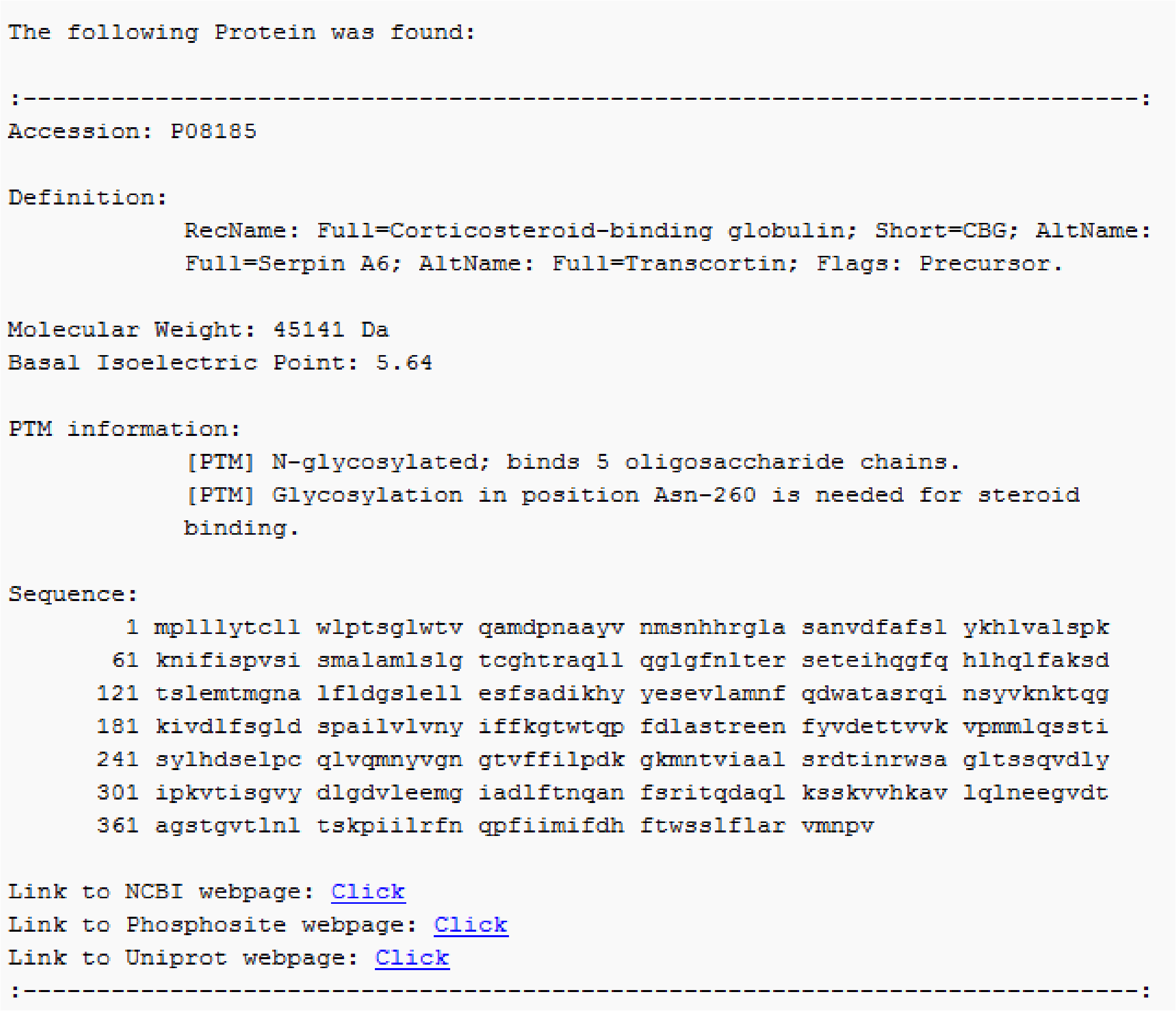
```
>> scanner


The current database contains 4435 entries.

[1] Search via the protein's Accession.
[2] Search via a fragment of the protein's Definition/Name.
[3] Filter a group of proteins via Molecular Weight.
[4] Filter a group of proteins via Basal Isoelectric Point.

Choose an option [1/2/3/4]
```

As shown in **Fig. 5 (b)**, the user is now asked to select one of the four search criteria: number 1 to search by using the accession number of the protein that the user wishes to be displayed; number 2 to search proteins containing certain word or words in their definition or name; number 3 to search all the proteins whose molecular weight is inside a range determined by the user; and number 4 to do the same using the basal isoelectric point of the protein.

4.4.2. Accession number of protein. For option number one the user may search for a specific protein by its accession number. In the step shown in **Fig. 5 (c)** the user must input the accession number of the protein they are searching for. Since the accession number is a unique identifier, the user must write the accession correctly, otherwise the algorithm will not find it in the database.

For this search the algorithm performs a Boolean comparison of the user input with all of the accession numbers, as shown in **Fig. 5 (d)**, in our database. If a result is found, the entry will be displayed, as shown in **Fig. 5 (e)**. **Fig. 7** shows the format of how the entry found by this search criterion is displayed on screen using as accession number P08185.

Fig. 7. A screenshot of the entry found by searching the database for the specific accession number P08185.



```
The following Protein was found:

:------------------------------------------------------------------------------:
Accession: P08185

Definition:
            RecName: Full=Corticosteroid-binding globulin; Short=CBG; AltName:
            Full=Serpin A6; AltName: Full=Transcortin; Flags: Precursor.

Molecular Weight: 45141 Da
Basal Isoelectric Point: 5.64

PTM information:
            [PTM] N-glycosylated; binds 5 oligosaccharide chains.
            [PTM] Glycosylation in position Asn-260 is needed for steroid
            binding.

Sequence:
        1 mplllytcll wlptsglwtv qamdpnaayv nmsnhhrgla sanvdfafsl ykhlvalspk
       61 knifispvsi smalamlslg tcghtraqll qglgfnlter seteihqgfq hlhqlfaksd
      121 tslemtmgna lfldgslell esfsadikhy yesevlamnf qdwatasrqi nsyvknktqg
      181 kivdlfsgld spailvlvny iffkgtwtqp fdlastreen fyvdettvvk vpmmlqssti
      241 sylhdselpc qlvqmnyvgn gtvffilpdk gkmntviaal srdtinrwsa gltssqvdly
      301 ipkvtisgvy dlgdvleemg iadlftnqan fsritqdaql ksskvvhkav lqlneegvdt
      361 agstgvtlnl tskpiilrfn qpfiimifdh ftwsslflar vmnpv

Link to NCBI webpage: Click
Link to Phosphosite webpage: Click
Link to Uniprot webpage: Click
:------------------------------------------------------------------------------:
```

4.4.3. Definition/Name. For option number two, the user may search for all the proteins within the database that contain a word or words within their definition/name according to the NCBI’s Protein database.

As shown in **Fig. 5 (f)**, the algorithm asks the user to input a fragment of the definition/name of the proteins they want to filter. In this case, the algorithm searches for the input in each definition/name in the built database by cycling through all the entries and performing Boolean comparisons, as shown in **Fig. 5 (g)**. If the algorithm finds that the input is in the definition/name of the current entry, then it saves this entry in a separate local variable that will store a list with all the matches.

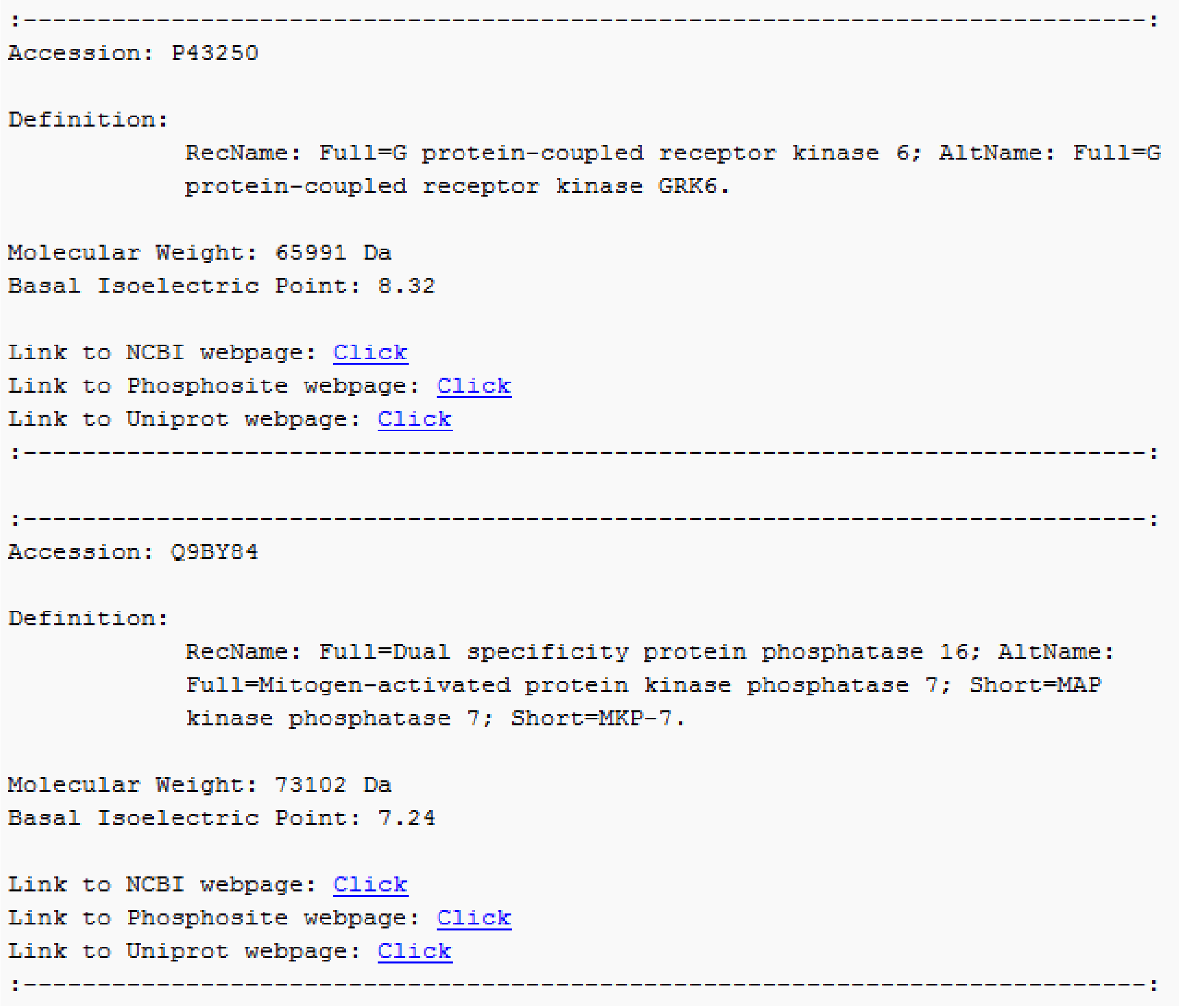
After the algorithm has searched the whole database, it counts the number of entries that were found and displays it, asking the user whether or not to display these results. If the user chooses to display all results, the algorithm displays all the accession numbers, the definition, the molecular weight, the basal isoelectric point, and the links to the online databases for all the proteins satisfying this search criteria, as shown in **Fig. 5 (h)**. Neither PTM information nor sequences are displayed this time to save space in the screen. **Fig. 8** shows a screenshot of the format in which these results are displayed.

4.4.4. Molecular weight and basal isoelectric point. For both of these search criteria the steps are the same: the algorithm asks the user to define a range for the value of these parameters to filter within the database by first entering the lower limit of the range, shown in **Fig. 5 (i)**, and then the upper limit, shown in **Fig. 5 (j)**. If, for example, the user wishes to search all the proteins with a basal isoelectric point between 4 and 4.5, then the user inputs 4 when the algorithm asks for the lower limit and 4.5 when it asks for the upper limit.

The algorithm then cycles through the entire database developed by either MW or pI and makes two Boolean comparisons, as shown in **Fig 5 (g)**. First if the data for the entry is equal or above the lower limit, and then if the data equals or is below the lower limit. If both conditions are met, the algorithm stores this entry in a local variable.

Before displaying the results, the algorithm displays the number of proteins found with the selected search criterion and asks the user whether or not to show them. If the user chooses so, then the algorithm displays all the proteins that matched the query, as shown in **Fig. 5 (h)**, with the same format than it does when filtering by definition/name, which can be seen in **Fig. 8**.

Fig. 8. A screenshot of two entries found by searching the database for the word “kinase” within the proteins’ definition.



```
:-----------------------------------------------------------------------------:
Accession: P43250

Definition:
            RecName: Full=G protein-coupled receptor kinase 6; AltName: Full=G
            protein-coupled receptor kinase GRK6.

Molecular Weight: 65991 Da
Basal Isoelectric Point: 8.32

Link to NCBI webpage: Click
Link to Phosphosite webpage: Click
Link to Uniprot webpage: Click
:-----------------------------------------------------------------------------:

:-----------------------------------------------------------------------------:
Accession: Q9BY84

Definition:
            RecName: Full=Dual specificity protein phosphatase 16; AltName:
            Full=Mitogen-activated protein kinase phosphatase 7; Short=MAP
            kinase phosphatase 7; Short=MKP-7.

Molecular Weight: 73102 Da
Basal Isoelectric Point: 7.24

Link to NCBI webpage: Click
Link to Phosphosite webpage: Click
Link to Uniprot webpage: Click
:-----------------------------------------------------------------------------:
```

5. CONCLUSIONS

Four MATLAB-based algorithms were developed in an effort to facilitate the development of mathematical models explaining the dynamics of the human proteome.

The “Builder” algorithm uses MATLAB programming language to read online databases through their HTML source code and download relevant information from it, building an offline database in a file. This file can then be accessed with the “Scanner” algorithm to perform searches or filter a number of proteins with the help of four criteria, resulting in a user-friendly search engine to interrogate protein databases filtered according to the interest of the researcher.

Although currently the database is only built with proteins that pertain to the *Homo Sapiens* that have at least one PTM according to the NCBI’s Protein database, this algorithm is the foundation for another one using other filtering criteria, such as an another organism, or all the entries within any database.

By doing this, a database which combines the information from all the major online databases could be created, allowing the researcher to have all the information in one place. So far, the only problem that this presents is that building the database might take a long time, and if the user wishes to have an up-to-date database, it must be built again every time. This, however, could be countered by a smarter algorithm which could selectively update only the outdated entries in the database, for example.

Having this bigger, more complete, database in a format easily accessible by MATLAB also would provide the possibility of doing several interesting data operations with the information that could eventually enlighten the current knowledge on the dynamics of the human proteome, such showing the correlation between certain types of proteins, or their functions, or composition with the appearance of certain diseases in the human proteome, for example.

6. FUTURE WORK

Both algorithms are still a work in progress. Current focus is to optimize the code and the database-building method in an attempt to reduce the time that the “Builder” algorithm requires to build a database. It is also necessary to prove the results of this work via experiments in a proteomics laboratory.

REFERENCES

[1] MCENTYRE J., OSTELL J. The NCBI Handbook. Bethesda (MD): National Center for Biotechnology Information, U.S. 2002. Internet: (http://www.ncbi.nlm.nih.gov/books/NBK21106/). Retrieved on Nov 20, 2012.

[2] MATHEWS, Christopher K., VAN HOLDE, Kensal Edward, AHERN, Kevin G. Biochemistry. Redwood City, USA: Benjamin Cummings, 2000. 1186 p.

[3] KAMATH, Karthik S., VASAVADA, Meghana S., SRIVASTAVA, Sanjeeva. Proteomic databases and tools to decipher post-translational modifications. Journal of Proteomics. 2011, vol. 75, p. 127-144.

[4] FUJII, Katsunori, UCHIKAWA, Hideki, TANABE, Yuzo, OMATA, Taku, NONAKA, Ikuya, KOHNO, Yoichi. 14-3-3 Proteins, particularly of the epsilon isoform, are detectable in cerebrospinal fluids of cerebellar diseases in children. Brain and Development. 2012. Internet: (http://www.sciencedirect.com/science/article/pii/S038776041200229X). Retrieved on Nov 20, 2012.

[5] BUÉE, Luc, BUSSIÈRE, Thierry, BUÉE-SCHERRER, Valérie, DELACOURTE, André, HOF, Patrick R. Tau protein isoforms, phosphorylation and role in neurodegenerative disorders. Brain Research Reviews. 2000, vol. 33, issue 1, p. 95-130.

[6] KOKJOHN, Tyler A., ROHER, Alex E. Amyloid precursor protein transgenic mouse models and Alzheimer's disease: Understanding the paradigms, limitations, and contributions. Alzheimer's & Dementia. 2009, vol. 5, issue 4, p. 340-347.

[7] BATES, Ian R., LIBICH, David S., WOOD, D.Denise, MOSCARELLO, Mario A., HARAUZ, George. An Arg/Lys→Gln mutant of recombinant murine myelin basic protein as a mimic of the deiminated form implicated in multiple sclerosis. Protein Expression and Purification. 2002, vol. 25, issue 2, p. 330-341.

[8] CLARK, John I., MUCHOWSKI, Paul J. Small heat-shock proteins and their potential role in human disease. Current Opinion in Structural Biology. 2000, vol. 10, issue 1, p. 52-59.

[9] O'BRIEN, Thomas W. Evolution of a protein-rich mitochondrial ribosome: implications for human genetic disease. Gene. 2002, vol. 286, issue 1, p. 73-79.

[10] CASTELLANOS, Lila, GONZÁLEZ, Luis Javier, PADRÓN, Gabriel. Proteómica. In: Combinatoria molecular. La Habana: Elfos Scientae, 2004, p. 367-404.

[11] THE MATHWORKS, INC. Documentation Center - Structures - MATLAB & Simulink. Internet: (http://www.mathworks.com/help/matlab/structures.html). Retrieved on Nov 20, 2012.

[12] RABILLOUD, Thierry, LELONG, Cécile. Two-dimensional gel electrophoresis in proteomics: A tutorial. Journal of Proteomics. 2011, vol. 74, issue 10, p. 1829-1841.

[13] HENG, Henry. Cancer genome sequencing: The challenges ahead. Bioessays. 2007, vol. 29 issue 8, p. 783-794.

[14] HENG, Henry, LIU, Guo, STEVENS, Joshua B., BREMER, Steven W., YE, Karen J., ABDALLAH, Batoul Y., HORNE, Steven D., YE, Christine J. Decoding

the genome beyond sequencing: The new phase of genomic research. Genomics. 2011, vol. 98 issue 4, p. 242-252.

[15] WALSH, Christopher T., GARNEAU-TSODIKOVA, Sylvie, GATTO, Gregory J. Protein Posttranslational Modifications: The Chemistry of Proteome Diversifications. Angew. Chem. Int. Ed. 2005, vol. 44, p. 7342 – 7372.

[16] BURKS, Christian, CINKOSKY, Michael J., GILNA, Paul, HAYDEN, Jamie E.-D., ABE, Yuki, ATENCIO, Edwin J., BARNHOUSE, Steve, BENTON, David, BUENAFE, Connie A., CUMELLA, Karen E., DAVISON, Dan B., EMMERT, David B., FAULKNER, Mary Jo, FICKETT, James W., FISCHER, William M., GOOD, Mark, HORNE, Deborah A., HOUGHTON, F.Kay, KELKAR, Praful M., KELLEY, Tom A., KELLY, Michael, KING, Melinda A., LANGAN, Bernard J., LAURR, Jeffrey T., LOPEZ, Natalie, LYNCH, Conrad, LYNCH, Janet, MARCHI, Janet B., MARR, Thomas G., MARTINEZ, Frances A., MCLEOD, Mia J., MEDVICK, Pat A., MISHRA, Santosh K., MOORE, John, MUNK, Christine A., MONDRAGON, Socorro M., NASSERI, Kevin K., NELSON, Debra, NELSON, Will, NGUYEN, Tan, REISS, Gloria, RICE, John, RYALS, Julie, SALAZAR, Margarita D., STELTS, Stephen R., TRUJILLO, Brian L., TOMLINSON, Laurie J., WEINER, Mark G., WELCH, Frank J., WIIG, Susan E., YUDIN, Katherine, ZINS, Larry B. GenBank: Current status and future directions. Methods in Enzymology, Academic Press. 1990, vol. 183, p. 3-22.

[17] PRUITT, Kim D., KATZ, Kenneth S., SICOTTE, Hugues, MAGLOTT, Donna R., Introducing RefSeq and LocusLink: curated human genome resources at the NCBI. Trends in Genetics. 2000, vol. 16 issue 1, p. 44-47.

[18] SCHNEIDER, Michel, LANE, Lydie, BOUTET, Emmanuel, LIEBERHERR, Damien, BOUGUELERET, Lydie, BAIROCH, Amos. The UniProtKB/Swiss-Prot knowledgebase and its Plant Proteome Annotation Program. Journal of Proteomics. 2009, vol. 72 issue 3, p. 567-573.

[19] ZHOU, Ao, ZHANG, Fan, CHEN, Jake Y. PEPPI: a peptidomic database of human protein isoforms for proteomics experiments. BMC Bioinformatics. 2010, vol 11, suppl 6, S7.

[20] ROMITI, M. Entrez Nucleotide and Entrez Protein FAQs. 2006 (Updated 2010). Internet: (http://www.ncbi.nlm.nih.gov/books/NBK49541/). Retrieved on Nov 20, 2012.

[21] NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION, U.S. Advanced search - Protein – NCBI. Internet: (http://www.ncbi.nlm.nih.gov/protein/advanced). Retrieved on Nov 20, 2012.

[22] NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION, U.S. Homo Sapiens[xorganism] - Protein – NCBI. Internet: (http://www.ncbi.nlm.nih.gov/protein?term=homo%20sapiens%5Borganism%5D). Retrieved on Nov 20, 2012.

[23] UNIPROT CONSORTIUM. Organism:"Homo Sapiens" in UniProtKB. Internet: (http://www.uniprot.org/uniprot/?query=organism%3A%22homo+sapiens%22). Retrieved on Nov 20, 2012.

[24] UNIPROT CONSORTIUM. UniProtKB. Internet: (http://www.uniprot.org/help/uniprotkb). Retrieved on Nov 20, 2012.

[25] UNIPROT CONSORTIUM. UniProt. Internet: (http://www.uniprot.org/). Retrieved on Nov 20, 2012.

[26] CELL SIGNALING TECHNOLOGY. About PhosPhositePlus. Internet: (http://www.phosphosite.org/staticAboutPhosphosite.do). Retrieved on Nov 20, 2012.

[27] CELL SIGNALING TECHNOLOGY. Protein, Sequence, or Reference Search. Internet: (http://www.phosphosite.org/psrSearchAction.do). Retrieved on Nov 20, 2012.

[28] GNAD, Florian, GUNAWARDENA, Jeremy, MANN, Matthias. PHOSIDA 2011: the posttranslational modification database. Nucleic Acids Research. 2010, vol. 39, issue 1, p. 253-260.

[29] MAX PLANCK INSTITUTE OF BIOCHEMISTRY, DEPT. OF PROTEOMICS AND SIGNAL TRANSDUCTION. PHOSIDA. Internet: (http://141.61.102.18/phosida/ptm/eukaryotes/choosesearchoption.aspx?species= homosapiens&). Retrieved on Nov 20, 2012.

[30] MAX PLANCK INSTITUTE OF BIOCHEMISTRY, DEPT. OF PROTEOMICS AND SIGNAL TRANSDUCTION. PHOSIDA. Internet: (http://www.phosida.com/). Retrieved on Nov 20, 2012.

[31] DINKEL, Holger, CHICA, Claudia, VIA, Allegra, GOULD, Cathryn M., JENSEN, Lars J., GIBSON, Toby J., DIELLA, Francesca. Phospho.ELM: a database of phosphorylation sites – updated 2011. Nucleic Acids Research. 2010, vol. 39, issue 1, p. 261-267.

[32] EMBL HEIDELBERG, SCB UNIT. Phospho.ELM Result. Internet: (http://phospho.elm.eu.org/). Retrieved on Nov 20, 2012.

[33] THE MATHWORKS, INC. Download URL content to MATLAB string – MATLAB. Internet: (http://www.mathworks.com/help/matlab/ref/urlread.html). Retrieved on Nov 20, 2012.

[34] THE MATHWORKS, INC. Data Types - MATLAB & Simulink. Internet: (http://www.mathworks.com/help/matlab/data-types_data-types.html). Retrieved on Nov 20, 2012.

ANNEX A. USER MANUAL

MATLAB VERSION

These algorithms have been developed with, and for, MATLAB 7.10.0 (R2010a). Running them with a different MATLAB version may cause the algorithms to not work properly ensuing unexpected results.

PRELIMINARY

Ensure that MATLAB® is correctly installed in your computer. If so, a proper MATLAB folder should have been created in the hard disk. The location of this folder may vary depending on installation configurations or operative system. Typically, for a Windows user, it will be located within the “My Documents” folder.

After the MATLAB folder has been located in the hard disk, access it and place in it both algorithms. These algorithms should come in the form of two different files with the same extension: “.m”.

The files are:

- Builder.m
- Scanner.m

PREPARATIONS

Before running the algorithms make sure that the following conditions are met:

- MATLAB's current folder (which can be located on the upper left part of the screen) shows the two algorithm files: Builder.m and Scanner.m
- If running "Builder", make sure the computer has a working internet connection.
- If running "Scanner", make sure MATLAB's current folder shows a file named "Database.mat", which ensures there is a database to scan.

BUILDER

1. Locate the Command Window.

2. Input "builder" (without the quotation marks) and hit enter.

3. The algorithm will ask you if you wish to proceed to building the database. If you don't, input "n" and hit enter (it will exit the algorithm), if you do, input "y" and hit enter.

4. MATLAB will display a message in the command window when the database building has been finished.

SCANNER

1. Locate the Command Window.

2. Input “scanner” (without the quotation marks) and hit enter.

3. Select a search criterion and input the corresponding number: “1” for accession, “2” for definition/name, “3” for molecular weight and “4” for basal isoelectric point. Hit enter.

4. According to the search criterion you choose, enter the information asked and hit enter.

5. Follow onscreen instructions.

MATLAB TIPS

- To clear the screen, input “clc” in the command window and hit enter.
- To stop any process or algorithm MATLAB is running, press “ctrl” and “c” at the same time in your keyboard while MATLAB is your active window.