Serdica J. Computing **12** (2018), No 3, 175–190

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

OBJECT-ORIENTED ANALYSIS FOR DESIGN OF PROTEIN FINGERPRINTS

Monica Dimitrova, Dimitar Vassilev

ABSTRACT. A new approach, based on object-oriented programming, for search and classification of protein patterns in a protein database is developed. Aiming at an improvement of the analysis of protein sequences and structures, certain basic patterns – such as motifs and fingerprints – are used. An unknown protein sequence can be classified by searching for those fingerprints from the database, assessing matches by estimating the statistical significance. The current implementation of the model algorithm for searching is a powerful tool that uses the PRINTS database as an example, however it does not support the option of adding of new features due to the conservative design of the program and the lack of publicly available code. A new version of the PRINTS database has recently been developed and this will require adding new features in the future. A novel object-oriented model for the implementation of the algorithm is proposed. This model is used to build a web application prototype, written in Python—the most widely used programming language in bioinformatics at present. The result of this study is a

ACM Computing Classification System (1998): J.3.

Key words: object-oriented analysis, bioinformatics, databases, protein fingerprints, pattern recognition.

maintainable software with open source code that can easily be extended with new functionalities.

1. Introduction. Nowadays data generating research technologies applied in biology and medicine are developing at a very quick pace and influence a wide array of scientific disciplines and in particular bioinformatics. The amount of data retrieved from various sequencing projects has been explored thoroughly in the last two decades. These data have produced new challenges for processing, storage and visualization in a rational and knowledge-generating manner. The key instrument in achieving this goal is the continuous improvement of the employed computational models and related software solutions, as well as the development of new approaches. One of the most common problems in bioinformatics applied to proteomics (a domain which deals with analysis, storage, visualisation of proteins and amino acids) is understanding the relationship between amino acid sequences and three-dimensional structures of the proteins in terms of their structure and function prediction. Much progress has been made in classification of proteins based on their sequences, and this knowledge is thoroughly used in protein modeling.

The models used at present for *in silico* inference of gene/protein function rely mostly on the identification of relationships between novel sequences and those of known function. The similarity found at the sequence level is assumed to be reflected by similarity at the levels of structure and function. The analysis of uncharacterized proteins is usually based on scanning and mapping full amino-acid sequence datasets against one or more publicly available databases [1]. Databases are divided into two categories—primary data sources, e. g., SWISS-PROT [2], OWL [3, 4], and secondary data sources [5] that condense the information from the primary databases into more and different potent identifiers (motifs, profiles, etc.) of evolutionary relationships, such as PROSITE [6], BLOCKS [7] and PRINTS [8, 5]. Such databases store reduced descriptions of protein families and can be used in practice for predicting the functions and structures of novel proteins [1].

1.1. Objective and tasks of the study. The major objective of the study is to develop and test a new implementation of the method for searching of protein fingerprints in unknown protein sequences for using them in

PRINTS database—a widest applicable bioinformatics resource, compendium of protein **fingerprints.** A *fingerprint* is a group of conserved motifs (short sequences of amino acids) used to characterise a protein family. Fingerprints can encode protein folds and functionalities more flexibly and powerfully than single motifs can.

A new object-oriented model for the implementation of the algorithm for fingerprints searching is developed and used to build a web app prototype which can be easily extended to meet the features of the PRINTS database. By developing such a tool with an open source code that can be easily maintained will be possible to redesign and reengineer PRINTS alongside modern IT technologies. In order to achieve the above stated goal, the following tasks are completed:

- An overview on the existing structure of PRINTS protein fingerprint compendium-database and the most used algorithm for searching of protein fingerprints. Specific features of bioinformatics basics will be given and discussed.
- Introducing a novel object-oriented model used to describe the records in the database as well as the algorithm for searching.
- Developing a web application prototype that finds the best matching fingerprints in the PRINTS database on a given amino acid sequence with a potential to be extended with new modules and features.

2. Protein fingerprints. The two categories of methods for identifying proteins use either profiles or motifs. The former approach is based on the compiling of a familial discriminator that contains both conserved and non-conserved regions of a multiple sequence alignment. In comparison, the motif approaches extract only the most conserved regions and can be divided into those that use a single motif and those that use multiple motifs. The single-motif searching methods, however, do not offer a biological context since only one conserved region is not enough for a match and might miss distant relatives that contain a vague version of the pattern [1].

The PRINTS protein fingerprint compendium-database uses multiple conserved motifs in order to create signatures that correspond to family

memberships [9]. It is usual to find more than one motif belonging to a protein family within a multiple sequence alignment and as more motifs are used, the matching with their natural neighbors increases (Fig. 1). A set of such motifs is defined as a fingerprint and is highly informative for the identification of distant relatives in a database search—mismatches are tolerated both at individual residues level and at motif level [10]. Usually, the motifs are separated along the sequence and do not overlap, however they may be contiguous in 3D-space [5].

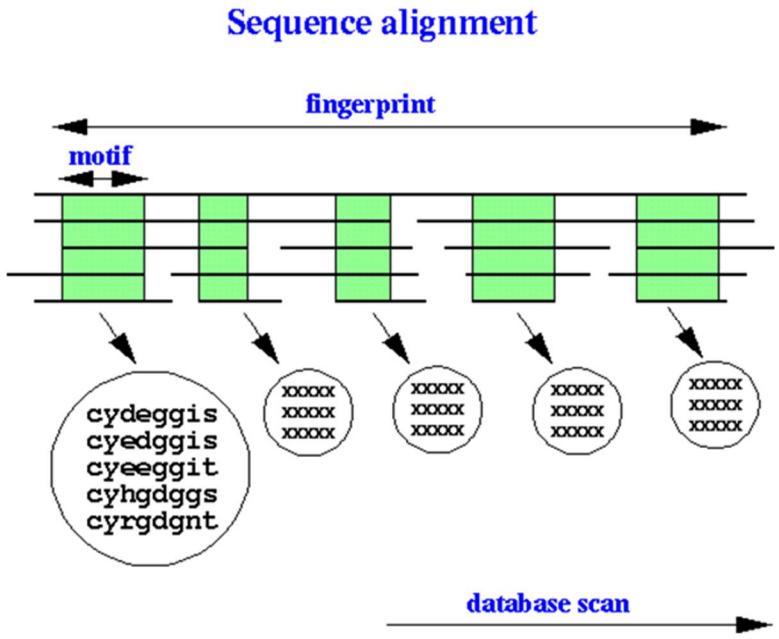

Fig. 1. Generation of protein fingerprints with their corresponding motifs

2.1. PRINTS database structure and functionality. PRINTS protein fingerprint compendium is a highly applicable information resource in bioinformatics that was created at the end of the last century and is constantly being updated (under development). It offers a range of new analysis and tools used in the annotation process and contributes unique functionality to InterPro, a freely available resource used to classify sequences into protein families [11]. PRINTS services are able to improve the quality of sequencing because the novel proteins can be compared and mapped against the whole

database or particular sequences in order to examine their structures and functionalities. With this knowledge protein (fingerprint) familial hierarchies can be made explicit and associations can be traced from subfamily, through family, to superfamily relations [8].

There are two kinds of fingerprints presented in the database depending on their complexity—simple and composite. Simple fingerprints are essentially single motifs, while composite fingerprints encode multiple motifs [12]. The majority of the database records are of the second type because the possibility of differentiation is greater in a search for many components and the results are easier to interpret.

The evolution and development of PRINTS makes sense and is possible thanks to the cooperation with other databases and projects. One of the most recent and notable projects is the integration of PRINTS with InterPro [8] and the resolving of protein family memberships as effective as possible in order to help InterPro's automated sequence analysis. Another successful collaboration was the European Kidney and Urine Proteomics project (EuroKUP) [8] in which a range of medically relevant protein families were studied to build hierarchical fingerprints for families in order to gain a better understanding of specific sequence properties that might cause chronic kidney diseases.

In order to continue developing, PRINTS and the related software should be updated regularly. Originally, PRINTS was built as a single ASCII text file [5]. This type of storing is quite common amongst molecular biology sources created in the past. However, it's not practical anymore because working with such data is unproductive especially when communicating with other databases. Relational databases have become really popular and widely used due to the speed and convenience of adding, deleting and modifying of records. Recently, a relational database has been created from the information in PRINTS in order to facilitate further development of both new tools and the database itself [12]. The information was logically separated while keeping the normalization practices and conventions.

As shown in Fig. 2, the main table in PRINTS is **FINGERPRINT**. It is used to describe a certain fingerprint and the most valuable fields are the ID, title, annotation and set of motifs that belong to the fingerprint. Another highly used table is **MOTIF**—every record is a representation of a motif with

the corresponding title, code, length and position in the fingerprint. Every motif has a set of sequences (variants), obtained in a multiple sequence alignment, described in the table **SEQ**. This table contains a certain sequence, code, start position and interval. These three tables provide an extended view of the fingerprints and contain the information needed to perform a search of an uncharacterized protein against PRINTS.

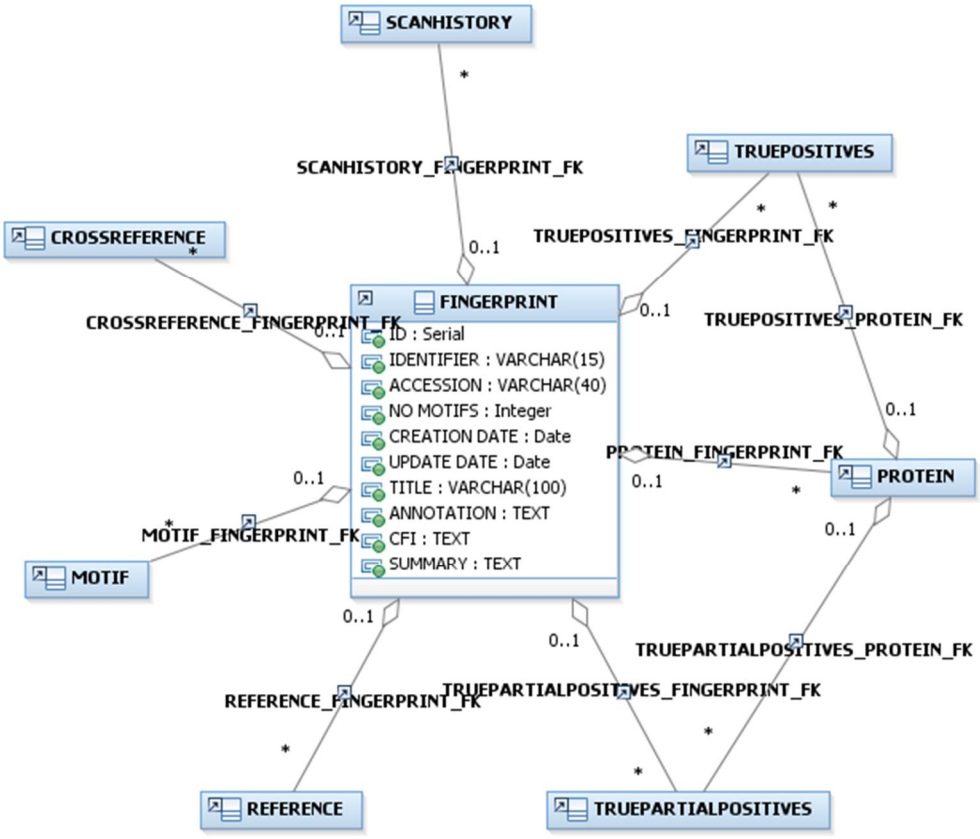

Fig. 2. UML diagram of PRINTS

2.2. Algorithm for searching. As the database has been modernized it is desirable to update the associated tools in order to have a fully up-to-date software package. Currently, searching against fingerprints in order to provide a diagnostic identification is implemented in FingerPRINTScan [8, 13], a web

tool that searches a given sequence of amino acids against PRINTS to identify the best or closest match and thus be used for indication of the family to which the unknown protein belongs. However, the software has outdated code that is neither open-source nor has a proper documentation.

The original algorithm for searching compares the query sequence against every fingerprint and finds the top results. In order to score a fingerprint all of its motifs should be considered—not every motif needs to be a match but the order must be preserved. Frequency tables and motif profiles are generated for every motif based on their variants and only the motifs with scores higher than the query threshold are reported as matches. This algorithm allows identification of the best matching fingerprint to a query sequence, relying on both scores and biological information [1].

The modernization of the tool proposed in this work provides a completely new code and web interface that follows the latest tendencies. The core of the original algorithm is preserved; however, a new object-oriented approach, presented in Section 3, is used in order to follow good practices in programming such as modularization and composability. The project is open-source and in this way future maintenance and improvements will be easy.

3. Models for object-oriented design. The analysis of the protein fingerprint search algorithm requires it to be divided into separate components as follows:

1. Fetching fingerprints and motifs from the PRINTS database.
2. Scoring of each motif:
    a. building a frequency table and profile of the motif;
    b. dividing the unknown sequence into overlapping motif-sized fragments;
    c. scoring each fragment according to the frequency table and profile;
    d. finding the best scoring sequence fragment for the motif.
3. Scoring of each fingerprint based on their motif scores.
4. Summarizing and displaying best scoring fingerprints.

This approach benefits from the use of an object-oriented model which is based on the concept of objects containing data in the form of fields (attributes). Objects have functions known as methods that can access and modify the data of the object that they are associated with. Each such component is considered as an individual object or a set of objects between which communication is transmitted in the form of messages. Scalar database values are transferred from the database to the non-scalar object-oriented values via object-relational mapping (ORM).

Object-oriented design has many advantages that are applicable in the fingerprint search algorithm. Modularity and encapsulation are easily achieved because objects are self-contained, and each functionality is responsible for a certain task or group of tasks. Code reusability is provided which reduces the duplication of code. Anytime a change needs to be introduced in the code, is it much easier to do it in just one place instead of multiple ones. Object-oriented programming is a natural and pragmatic approach to break down software into smaller problems that can be easily solved, one object at a time.

There are two main types of classes used in the model implementing the fingerpint search algorithm. The first type represents data from the relational PRINTS database via ORM. The second type of classes represents objects that are used as components of the algorithm or presenters of the obtained results.

The relationships between the classes are displayed in an UML diagram in Fig. 3. The DigestSequence class is associated with PositionalAnalysis, FingerprintScore and MotifScore classes. DigestSequence uses PositionalAnalysis instances to calculate scores for motif and a certain fragment of the query sequence. PositionalAnalysis is associated with a certain SubstitutionMatrix instance. The generated scores are represented by MotifScore and FingerprintScore objects. MotifScore is associated with Motif class and FingerprintScore is associated with Fingerprint class. Each Fingerprint has many Motif instances and each Motif has many Seq instances. Below is an abridged explanation of the main groups of classes used in the program.

Fig. 3. UML diagram of object-oriented design

3.1. Fingerprint, Motif, Sequence. These three classes contain the data retrieved from a database record of the corresponding tables in PRINTS. This approach of using ORM allows associating of various methods with the objects, such as calculating the score of a fingerprint against a protein sequence or generating a histogram of amino acids in every position in a motif.

The **Fingerprint** objects are initialised with fingerprint information such as title, annotation, creation date, update date and relationship with motifs. They also coordinates message passing from the motif objects in order to identify matches.

Motif objects are initialised with motif information such as title, code, relationship with a fingerprint and sequences. They are also responsible for the calculation of motif scores.

Sequence objects are initialised with sequence data such as sequence, pcode and relationship with a motif. They are used in the creation of frequency table and profile for a motif.

3.2. SubstitutionMatrix. It is scientifically proven that certain amino acids can be substituted with others in a particular way [7]. This information is stored in substitution matrices that describe the rate at which one character in a sequence changes to another over time [14]. Rows and columns describe amino acids and cells contain scores that describe the substitution of the two amino acids. Substitution matrices are used in the computing of motif profiles in the protein fingerprint search algorithm. They are stored in YAML files—a human-readable data serialization language, commonly used for configuration files. The **SubstitutionMatrix** class performs the loading of a particular matrix by a given type and retrieving the score for a given pair of amino acids.

3.3. PositionAnalysis. In order to analyze a certain motif, an instance of **PositionAnalysis** class is created. It implements the generating of a histogram (frequency table) of the amino acids in each position in the motif [15]. Another function of this class is the computing of a profile which utilizes both the histogram and the chosen substitution matrix.

3.4. Match, MotifScore and FingerprintScore. To achieve a diagnostic identification, three types of measures are introduced in order to identify a "match" with a fingerprint—amino acid match, motif match and fingerprint match [9]. The amino acid match shows whether the amino acid on position X matches any of the amino acids in the frequency table on that position. If this condition is not met, the score is 0, otherwise the score is equal to the number of times the amino acid is present in the frequency table. The score is summed for every position in the frequency table and motif score is returned. The highest score for every motif represents the fingerprint score. The following three classes implement the evaluation of the results from the search algorithm. Instances of **Match** class contain information about the profile, identity score, statistical significance, the fragment of the query sequence being analyzed and the position in that fragment. Every motif has a **MotifScore** object that builds a list of Match instances and provides statistics such as top match or average score of the matches. **FingeprintScore** instances represent the score of a given fingerprint, providing the number of motifs that score, i.e. meet a certain criteria, and the type of the match—full or partial. Full matches are defined as all motifs in a fingerprint consistently scoring high. Partial matches are matches with less than all of the motifs in a fingerprint [9].

3.5. DigestSequence. This class represents the main part of the algorithm for searching. By given unknown protein sequence and substitution matrix code, it loads fingerprints and motifs from the PRINTS database via the **Seq, Motif** and **Fingerprint** classes. Then it processes every motif by breaking the query sequence into motif-sized fragments, creates **PositionAnalysis** objects for each fragment and finds the best scoring one. Later it generates **MotifScore** and **FingerprintScore** objects for the corresponding motifs and fingerprints and presents the results in JSON format.

4. Web application. The suggested approach is implemented in **Python**, a high-level programming language that is widely used in science and in particular in bioinformatics because of its readability. The web application is created using **Flask**, a lightweight BSD licensed framework for Python [16].

Flask is a micro framework because it does not require additional libraries, however it is highly extendable and various components can be added. The web application has a modern and responsive look, achieved with **Bootstrap**—a popular open source front-end web framework for designing websites [17]. The code of the program is publicly available [18] which will make the future development and maintenance of the project easier. Fig. 4 shows the web page with results from the scan of a protein sequence—the sequence itself is located in the top, followed by a table with fingerprints. Every row describes one motif—fingerprint which includes the motif, number of the motif, identity score, profile score, fragment of the query sequence that has those scores, length of fragment and its position in the query sequence. Motifs are grouped by fingerprints and ordered descending by the score of the fingerprint.

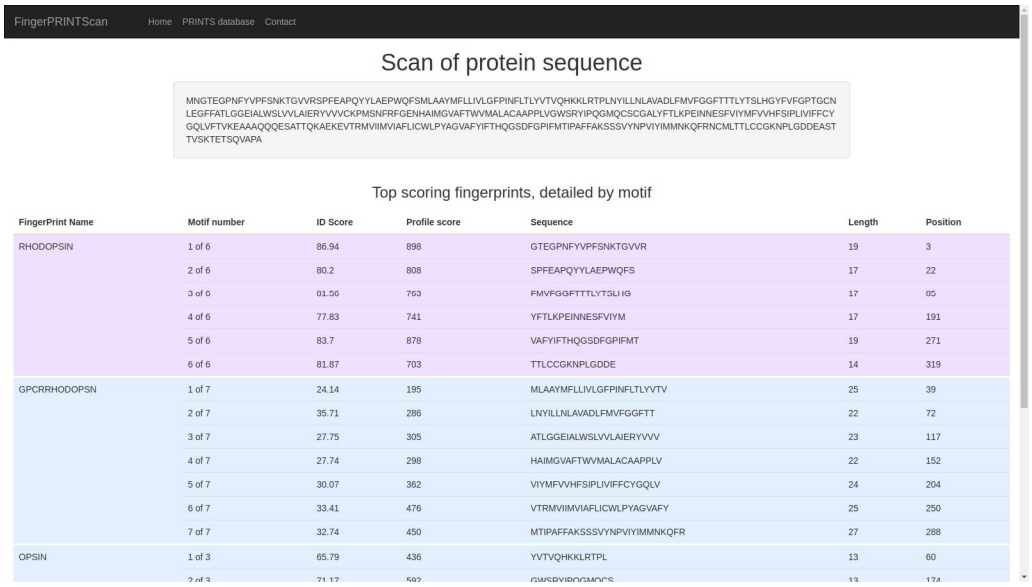

Fig. 4. Screenshot of the web application prototype

5. Summary and further development. Testing was performed with a query amino acid sequence OPSD_SHEEP that is found in Rhodopsin protein (photoreceptor required for image-forming vision) [19]. The program successfully identifies the RHODOPSIN fingerprint as the best match for

OPSD_SHEEP. Moreover, it constructs a detailed list with scores for every motif present in the fingerprint (Fig. 5) as well as length of the motif and position in the fingerprint. However, the calculation of statistical significance, presented as p-values, needs to be improved in order to provide more precise values and thus is not included in the results.

FingerPrint Name	Motif number	ID Score	Profile score	Sequence	Length	Position
RHODOPSIN	1 of 6	86.94	898	GTEGPNFYVPFSNKTGVVR	19	3
	2 of 6	80.2	808	SPFEAPQYYLAEPWQFS	17	22
	3 of 6	81.56	763	FMVFGGFTTTLYTSLHG	17	85
	4 of 6	77.83	741	YFTLKPEINNESFVIYM	17	191
	5 of 6	83.7	878	VAFYIFTHQGSDFGPIFMT	19	271
	6 of 6	81.87	703	TTLCCGKNPLGDDE	14	319

Fig. 5. Detailed results for RHODOPSIN fingerprint, identified as top match for OPSD_SHEEP sequence.

This project can be extended with additional functionalities such as integration with UniProt:Swiss-Prot and UniProt:TrEMBL. This will allow users to submit sequence IDs instead of raw sequence strings as it is possible in the prototype. Another feature that can be implemented is integration with other databases in order to provide PRINTS related information and classification to other tools.

Acknowledgements. The presented work has been funded by the Bulgarian NSF within the "Methods for Data Analysis and Knowledge Discovery in Big Sequencing Datasets" project, Contract DFNI-I02/7 of 12 December 2014, and by the Bulgarian NSF within the "A Model of Integration of Cloud Framework for Hybrid Massive Parallelism and its Application for Analysis and Automated Semantic Enhancement of Big Heterogeneous Data Collections" project, Contract DFNI 02/9 of 17 December 2016.

REFERENCES

[1] Scordis P., D. R. Flower, T. K. Attwood. FingerPRINTScan: intelligent searching of the PRINTS motif database. *Bioinformatics*, **15** (1999), No 10, 799–806.

[2] Bairoch A., R Apweiler. The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Research*, **27** (1999), No 1, 49–54.

[3] Mahdavi M. A. (Ed.). Bioinformatics—Trends and Methodologies. InTech, Croatia, 2011.

[4] Bleasby A.J., D. Akrigg D., Attwood. T.K. OWL—a non-redundant composite protein sequence database. *Nucleic Acids Research*, **22** (1994), No 17, 3574–3577.

[5] Attwood T. K., M. E. Beck, A. J. Bleasby, D. J. Parry-Smith. PRINTS—a database of protein motif fingerprints. *Nucleic Acids Research,* **22** (1994), No 17, 3590–3596.

[6] Sigrist C. J. A., L. Cerutti, E. De Castro, P. S. Langendijk-Genevaux, V. Bulliard, A. Bairoch, N. Hulo. PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, **38** (2010), D161–D166.

[7] Henikoff S., J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the USA*, **89** (1992), 10915–10919.

[8] Attwood T. K, A. Coletta, G. Muirhead, A. Pavlopoulou, P. B. Philippou, I. Popov, C. Romá-Mateo, A. Theodosiou, A. L. Mitchell. The PRINTS database: a fine-grained protein sequence annotation and analysis resource—its status in 2012. *Database* **2012** (2012), Article ID bas019.

[9] Scordis P. Diagnostic Identification of amino acid sequences using the method of FingerPrinting. MSc thesis, University College London, 1996.

[10] DbBrowser—Protein Families. http://www.bioinf.man.ac.uk/dbbrowser/ember/prototype/CHAPTER03/INFORMATION.shtml, 15 November 2018.

[11] Mitchell A., H. Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, S. Pesseat, A. Sangrador-Vegas, M. Scheremetjew, C. Rato, S. Y. Yong, A. Bateman, M. Punta, T. K. Attwood, C. J. Sigrist, N. Redaschi, C. Rivoire, I. Xenarios, D. Kahn, D. Guyot, P. Bork, I. Letunic, J. Gough, M. Oates, D. Haft, H. Huang, D. A. Natale, C. H. Wu, C. Orengo, I. Sillitoe, H. Mi, P. D. Thomas, R. D. Finn. The InterPro protein families database: the classification resource after 15 years. *Nucleic Acids Research*, **43** (2015), Database issue, D213–221.

[12] Dimitrov A. Structure of the bioinformatic relational database PRINTS and participative functional web applications. MSc Thesis, St. Kliment Ohridski University of Sofia, 2015. (in Bulgarian)

[13] FingerPRINTScan software. http://130.88.97.239/cgi-bin/dbbrowser/fingerPRINTScan/FPScan_fam.cgi, 15 November 2018.

[14] Mount D. W. Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, 2001.

[15] Higgs P. G., T. K. Attwood. Bioinformatics and Molecular Evolution. Blackwell Publishing, 2005.

[16] Flask. http://flask.pocoo.org/, 15 November 2018.

[17] Bootstrap. https://getbootstrap.com/, 15 November 2018.

[18] FingerPRINTScan. https://bitbucket.org/mbdimitrova/fingerprintscan/, 15 November 2018.

[19] UniProtKB—OPSD_SHEEP. http://www.uniprot.org/uniprot/P02700, 15 November 2018.

Monica Dimitrova
e-mail: monicabdimitrova@gmail.com
Dimitar Vassilev
e-mail: dimitar.vassilev@fmi.uni-sofia.bg
Faculty of Mathematics and Informatics
St. Kliment Ohridski University of Sofia
5, James Bourchier Blvd
1164 Sofia, Bulgaria

Received November 10, 2017
Final Accepted November 9, 2018