IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning

Kaspar Riesen and Horst Bunke

Institute of Computer Science and Applied Mathematics, University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{riesen,bunke}@iam.unibe.ch

Abstract. In recent years the use of graph based representation has gained popularity in pattern recognition and machine learning. As a matter of fact, object representation by means of graphs has a number of advantages over feature vectors. Therefore, various algorithms for graph based machine learning have been proposed in the literature. However, in contrast with the emerging interest in graph based representation, a lack of standardized graph data sets for benchmarking can be observed. Common practice is that researchers use their own data sets, and this behavior cumbers the objective evaluation of the proposed methods. In order to make the different approaches in graph based machine learning better comparable, the present paper aims at introducing a repository of graph data sets and corresponding benchmarks, covering a wide spectrum of different applications.

1 Introduction

The first step in any system in pattern recognition, machine learning, data mining, and related fields consists in representing objects by adequate data structures. In the statistical approach the data structure is given by an n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$, where each of the n dimensions represents the value of a specific feature. In recent years a huge amount of algorithms for classification, clustering, and analysis of objects given in terms of feature vectors have been developed [1].

Yet, the use of feature vectors implicates two limitations. First, as vectors represent a predefined set of features, all vectors of a set have to preserve the same length regardless of the size or complexity of the corresponding objects. Furthermore, there is no direct possibility to describe binary relationships among different parts of an object. It is well known that both constraints can be overcome by graph based representation [2]. That is, graphs allow us to adapt their size to the complexity of the underlying objects, and moreover, graphs offer a convenient possibility to describe relationships among different parts of an object.

Due to the ability of graphs to represent properties of entities and binary relations at the same time, a growing interest in graph-based object representation

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 287–297, 2008.

in pattern analysis can be observed [2]. That is, graphs found widespread applications in science and engineering. In the fields of bioinformatics and chemoinformatics, for instance, graph based representations have been intensively used [3, 4, 5]. Another field of research where graphs have been studied with emerging interest is that of web content mining [6]. Image classification is a further area of research where graph based representation draws the attention [7, 8, 9, 10]. Finally, we like to mention computer network analysis, where graphs have been used to detect network anomalies and predict abnormal events [11].

In statistical machine learning, the UCI Machine Learning Repository [12] is well established and widely used for benchmarking different algorithms. On the other hand a lack of standardized data sets for benchmarks in graph based machine learning can be observed. For an early discussion of benchmarking graph matching algorithms see [13]. As of today, however, there is only one standard set of graphs adequate for graph matching tasks publicly available to the knowledge of the authors, viz. the TC-15 graph database [14]. However, this data set consists of synthetically generated graphs only. Furthermore the graphs are particularly generated for exact graph matching algorithms rather than general matching tasks. In [15] benchmarks for graph problems are available. These benchmarks, however, are defined for special problems from graph theory, such as the maximum clique or vertex coloring problem, and are not related to pattern recognition and machine learning. The present paper aims at making a first step towards creating a graph repository that is suitable for a wide spectrum of tasks in pattern recognition and machine learning. These graph data sets emerged in the context of the autors' recent work on graph kernels [16] and graph embedding [17]. All graph data sets discussed in the present paper are publicly available or will be made available in the near future[1].

2 The Graph Set Repository

An attributed graph g, or graph for short, is defined as a four-tuple $g = (V, E, \mu, \nu)$, where V is the finite set of nodes, $E \subseteq V \times V$ is the set of edges, $\mu : V \to L$ is the node labeling function, and $\nu : E \to L$ is the edge labeling function. This definition allows us to handle arbitrary graphs with unconstrained labeling functions. For example, the labels can be given by the set of integers, the vector space $\mathbb{R}^n$, or a set of symbolic labels $L = \{\alpha, \beta, \gamma, \ldots\}$. Moreover, unlabeled graphs are obtained by assigning the same label l to all nodes and edges. Edges are given by pairs of nodes (u, v), where $u \in V$ denotes the source node and $v \in V$ the target node of a directed edge. Undirected graphs can be modeled by inserting a reverse edge $(v, u) \in E$ for each edge $(u, v) \in E$ with $\nu(u, v) = \nu(v, u)$.

Each of the data sets presented in the next subsections is divided into three disjoint subsets, which can be used for training, validation, and testing novel learning algorithms. Hence, the benchmarks are primarily designed for supervised learning tasks. Note, however, that the test set can be also used for benchmarking unsupervised learning algorithms. If appropriate, all three or two out of

[1] `http://www.iam.unibe.ch/fki/databases/iam-graph-database`

the three subsets can be merged. For each data set the classification result of a k-nearest neighbor classifier (k-NN) in conjunction with graph edit distance [18] on the test set is given. These results can serve as a first reference system to compare other algorithms with. Note that the meta parameters for graph edit distance (node and edge insertion/deletion cost) and the number k of graphs considered by the classifier are determined on the independent validation sets. A summary of the graph data sets together with some characteristic properties appears in Table 1. In the following subsections, each data set will be described in greater detail.

2.1 Letter Database

The first graph data set involves graphs that represent distorted letter drawings. We consider the 15 capital letters of the Roman alphabet that consist of straight lines only (*A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z)*. For each class, a prototype line drawing is manually constructed. These prototype drawings are then converted into prototype graphs by representing lines by undirected edges and ending points of lines by nodes. Each node is labeled with a two-dimensional attribute giving its position relative to a reference coordinate system. Edges are unlabeled. The graph database consists of a training set, a validation set, and a test set of size 750 each. The graphs are uniformly distributed over the 15 classes. In order to test classifiers under different conditions, distortions are applied on the prototype graphs with three different levels of strength, viz. *low*, *medium* and *high*. Hence, our experimental data set comprises 6,750 graphs altogether. In Fig. 1 the prototype graph and a graph instance for each distortion level representing the letter A are illustrated. The classification rates achieved on this data set are 99.6% (*low*), 94.0% (*medium*), and 90.0% (*high*).

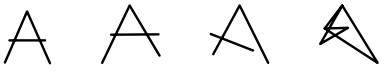


Fig. 1. Instances of letter A: Original and distortion levels *low*, *medium* and *high* (from left to right)

2.2 Digit Database

The digit data set consists of graphs representing handwritten digits [19]. The original version of this database includes 10,992 handwritten digits from classes 0 to 9. For our data set a randomly selected subset of totally 3,500 digits is used. This set is split into traininig set of size 1,000, a validation set of size 500, and a test set of size 2,000. The digit graphs are uniformly distributed over the 10 classes. During the recording of the digits, the position of the pen was determined with constant time intervals. The resulting sequences of (x, y)-coordinates were converted into graphs by inserting nodes in regular intervals between the starting

and ending points of a line. Successive nodes are connected by undirected edges. Each node is labeled with a two-dimensional attribute giving its position relative to a reference coordinate system. The edges are attributed with an angle denoting the orientation of the edge with respect to the horizontal direction. In Fig. 2 one particular graph instance for each digit class is illustrated. The classification rate achieved on this data set is 91.0%.

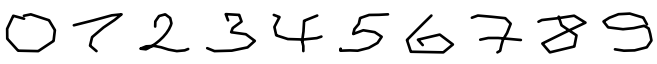


Fig. 2. A graph instance of each of the ten digit classes

2.3 GREC Database

The GREC data set consists of graphs representing symbols from architectural and electronic drawings. The images occur at five different distortion levels. In Fig. 3 for each distortion level one example of a drawing is given. Depending on the distortion level, either erosion, dilation, or other morphological operations are applied. The result is thinned to obtain lines of one pixel width. Finally, graphs are extracted from the resulting denoised images by tracing the lines from end to end and detecting intersections as well as corners. Ending points, corners, intersections and circles are represented by nodes and labeled with a two-dimensional attribute giving their position. The nodes are connected by undirected edges which are labeled as *line* or *arc*. An additional attribute specifies the angle with respect to the horizontal direction or the diameter in case of arcs. From the original GREC database [20], 22 classes are considered. For an adequately sized set, all graphs are distorted nine times to obtain a data set containing 1,100 graphs uniformely distributed over the 22 classes. The resulting set is split into a traininig and a validation set of size 286 each, and a test set of size 528. The classification rate achieved on this data set is 95.5%.

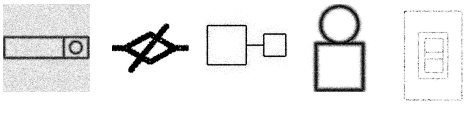
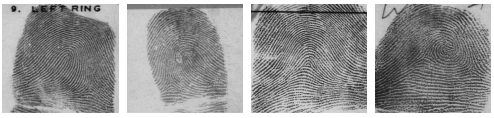


Fig. 3. A sample image of each distortion level

2.4 Fingerprint Database

Fingerprints are converted into graphs by filtering the images and extracting regions that are relevant [21]. In order to obtain graphs from fingerprint images, the relevant regions are binarized and a noise removal and thinning procedure is applied. This results in a skeletonized representation of the extracted regions. Ending points and bifurcation points of the skeletonized regions are represented by nodes. Additional nodes are inserted in regular intervals between ending points and bifurcation points. Finally, undirected edges are inserted to



(a) *Left* (b) *Right* (c) *Arch* (d) *Whorl*

Fig. 4. Fingerprint examples from the four classes

link nodes that are directly connected through a ridge in the skeleton. Each node is labeled with a two-dimensional attribute giving its position. The edges are attributed with an angle denoting the orientation of the edge with respect to the horizontal direction.

The fingerprint database used in our experiments is based on the NIST-4 reference database of fingerprints [22]. It consists of a training set of size 500, a validation set of size 300, and a test set of size 2,000. Thus, there are 2,800 fingerprint images totally out of the four classes *arch*, *left*, *right*, and *whorl* from the Galton-Henry classification system. Note that in our benchmark test only the four-class problem of fingerprint classification is considered, i.e. the fifth class *tented arch* is merged with the class *arch*. Therefore, the first class (arch) consists of about twice as many graphs as the other three classes (left, right, whorl). For examples of these fingerprint classes, see Fig. 4. The classification rate achieved on this data set is 76.6%.

2.5 COIL-RAG Database

The COIL-100 database [23] consists of images of 100 different objects. Images of the objects are taken at pose intervals of 5 degrees. Fig. 5 shows an example image of each class. We first segment images into regions of homogeneous color using a mean shift algorithm [24]. Segmented images are transformed into region adjacency graphs by representing regions by nodes, labeled with attributes

Fig. 5. COIL images of 100 different objects

specifying the color histogram of the corresponding segment, and the adjacency of regions by edges. The edges are labeled with the length, in pixels, of the common border of two adjacent regions. For a more detailed description of the graph extraction process see [7].

The training set is composed of 12 images per object, acquired every 15 degree of rotation. From the remaining images we randomly select five images per object for the validation set, and ten images per object for the test set. This results in a training set of size 2,400, a validation set of size 500, and a test set of size 1,000. The total amount of graphs is uniformely distributed over the 100 classes. The classification rate achieved on this data set is 92.5%.

2.6 COIL-DEL Database

The same images as for the COIL-RAG database described above are used for this data set. However, a different graph extraction procedure is applied [9]. The Harris corner detection algorithm [25] is used to extract corner features from the images. Based on these corner points, a Delaunay triangulation is applied. The result of the triangulation is then converted into a graph by representing lines by undirected edges and ending points of lines by nodes. Each node is labeled with a two-dimensional attribute giving its position, while edges are unlabeled.

For this graph extraction method the training, the validation, and the test sets contain the same images as COIL-RAG. The classification rate achieved on this data set is 93.3%.

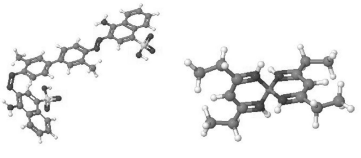
2.7 Web Database

In [6] several methods for creating graphs from web documents are introduced. For the graphs included in this data set, the following method was applied. First, all words occuring in the web document – except for stop words, which contain only little information – are converted into nodes in the resulting web graph. We attribute each node with the corresponding word and its frequency, i.e. even if a word appears more than once in the same web document we create only one unique node for it and store its total frequency as an additional node attribute. Next, different sections of the web document are investigated individually. These sections are *title*, which contains the text related to the document's title, *link*, which is text in a clickable hyperlink, and *text*, which comprises any of the readable text in the web document. If a word w_i immediately precedes word w_{i+1} in any of the sections *title*, *link*, or *text*, a directed edge from the node corresponding to word w_i to the node corresponding to the word w_{i+1} is inserted in our web graph. The resulting edge is attributed with the corresponding section label. Although word w_i might immediately precede word w_{i+1} in more than just one section, only one edge is inserted. That is, an edge is possibly labeled with more than one section label. Finally, only the most frequently used words (nodes) are kept in the graph and the terms are conflated to the most frequently occuring forms.

In our experiments we make use of a data set that consists of 2,340 documents from 20 categories (*Business*, *Health*, *Politics*, *Sports*, *Technology*, *Entertainment*, *Art*, *Cable*, *Culture*, *Film*, *Industry*, *Media*, *Multimedia*, *Music*, *Online*, *People*, *Review*, *Stage*, *Television*, and *Variety*). The last 14 catgories are sub-categories related to entertainment. The number of documents of each category varies from only 24 (Art) up to about 500 (Health). These web documents were originally hosted at Yahoo as news pages (`http://www.yahoo.com`). The database is split into a training, a validation, and a test set of equal size (780). The classification rate achieved on this data set is 80.3%.

2.8 AIDS Database

The AIDS data set consists of graphs representing molecular compounds. We construct graphs from the AIDS Antiviral Screen Database of Active Compounds [26]. This data set consists of two classes (*active, inactive*), which represent molecules with activity against HIV or not. The molecules are converted into graphs in a straightforward manner by representing atoms as nodes and the covalent bonds as edges. Nodes are labeled with the number of the corresponding chemical symbol and edges by the valence of the linkage. In Fig. 6 one molecular compound of both classes is illustrated. Note that different shades of grey represent different chemical symbols, i.e. node labels. We use a training set and a validation set of size 250 each, and a test set of size 1,500. Thus, there are 2,000 elements totally (1,600 inactive elements and 400 active elements). The classification result achieved on this data set is 97.3%.



(a) Active (b) Inactive

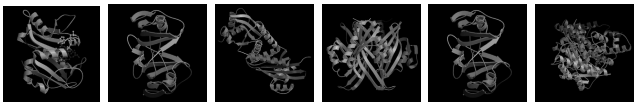
Fig. 6. A molecular compound of both classes

2.9 Mutagenicity Database

Mutagenicity is one of the numerous adverse properties of a compound that hampers its potential to become a marketable drug [27]. In order to convert molecular compounds of the mutagenicity data set into attributed graphs the same procedure as for the AIDS data set is applied. The data set was originally prepared by the authors of [27]. The mutagenicity data set is divided into two classes *mutagen* and *nonmutagen*. We use a training set of size 1,500, a validation set of size 500, and a test set of size 2,337. Thus, there are 4,337 elements totally (2,401 mutagen elements and 1,936 nonmutagen elements). The classification rate achieved on this data set is 71.5%.

2.10 Protein Database

The protein data set consists of graphs representing proteins originally used in [3]. The graphs are constructed from the Protein Data Bank [28] and labeled with their corresponding enzyme class labels from the BRENDA enzyme database [29]. The proteins database consists of six classes (*EC 1, EC 2, EC 3, EC 4, EC 5, EC 6*), which represent proteins out of the six enzyme commission top level hierarchy (EC classes). The proteins are converted into graphs by representing the secondary structure elements of a protein with nodes and edges of an attributed graph. Nodes are labeled with their type (helix, sheet, or loop) and their amino acid sequence (e.g. *TFKEVVRLT*). Every node is connected with an edge to its three nearest neighbors in space. Edges are labeled with their type and the distance they represent in angstroms. In Fig. 7 six images of proteins of all six classes are given.



(a) *EC 1* (b) *EC 2* (c) *EC 3* (d) *EC 4* (e) *EC 5* (f) *EC 6*

Fig. 7. Protein examples of all top level classes

There are 600 proteins totally, 100 per class. We use a training, validation and test set of equal size (200). The classification task on this data set consists in predicting the enzyme class membership. We achieve a classification rate of 65.5% on the test set.

3 Conclusions

In the present paper ten graph sets with quite different characteristics are presented. They represent line drawings, gray scale and color images, HTML webpages, molecular compounds, and proteins. In Table 1 a summary of all graph datasets and their corresponding characteristics is provided. In addition to a description of the data sets, classification results achieved with a simple reference system based on a nearest-neighbor classifier are given. All data sets are publicly available or will be made available soon. We expect that the graph repository introduced in this paper provides a major contribution towards promoting the use of graph based representations and making graph based pattern recognition and machine learning algorithms better comparable against each other.

In future work we want to further expand the graph set repository. Towards this end, we highly encourage the community not only to use the available sets for developing and testing their algorithms, but also to integrate their own graph sets into our repository.

Table 1. Summary of graph data set characteristics, viz. the size of the training (*tr*), the validation (*va*) and the test set (*te*), the number of classes (#classes), the label alphabet of both nodes and edges, the average and maximum number of nodes and edges (∅/max nodes/edges), whether the graphs are uniformely distributed over the classes or not (balanced), and the recognition rate of the k-NN classifier (RR)

Database	size (*tr*, *va*, *te*)	#classes	node labels	edge labels	∅ nodes	∅ edges	max nodes	max edges	balanced	RR
Letter (*low*)	750, 750, 750	15	x, y coordinates	none	4.7	3.1	8	6	Y	99.6%
Letter (*medium*)	750, 750, 750	15	x, y coordinates	none	4.7	3.2	9	7	Y	94.0%
Letter (*high*)	750, 750, 750	15	x, y coordinates	none	4.7	4.5	9	9	Y	90.0%
Digit	1,000, 500, 2,000	10	x, y coordinates	Angle	11.8	13.1	32	30	Y	91.0%
GREC	286, 286, 528	22	x, y coordinates	Line type	11.5	12.2	25	30	Y	95.5%
Fingerprint	500, 300, 2,000	4	x, y coordinates	Angle	5.42	4.42	26	24	N	76.6%
COIL-RAG	2,400, 500, 1,000	100	RGB histogram	Boundary length	3.0	3.0	11	13	Y	92.5%
COIL-DEL	2,400, 500, 1,000	100	x, y coordinates	none	21.5	54.2	77	222	Y	93.3%
Web	780, 780, 780	20	Word and its frequency	Section(s) type	186.1	104.6	834	596	N	80.3%
AIDS	250, 250, 1,500	2	Chemical symbol	Valence	15.7	16.2	95	103	N	97.3%
Mutagenicity	1,500, 500, 2,337	2	Chemical symbol	Valence	30.3	30.8	417	112	N	71.5%
Protein	200, 200, 200	6	Type and aa-sequence	Type and distance	32.6	62.1	126	149	Y	65.5%

Acknowledgements

This work has been supported by the Swiss National Science Foundation (Project 200021-113198/1). For making their programs for graph extraction of the COIL images available to us, we would like to thank D. Emms (Delaunay graphs) and B. Le Saux (RAG graphs). The webgraphs and the protein graphs are due to A. Schenker and K. Borgwardt, respectively. We are very grateful to both authors.

References

1. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley Interscience, Hoboken (2000)
2. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Artificial Intelligence 18(3), 265–298 (2004)
3. Borgwardt, K., Ong, C., Schönauer, S., Vishwanathan, S., Smola, A., Kriegel, H.P.: Protein function prediction via graph kernels. Bioinformatics 21(1), 47–56 (2005)
4. Mahé, P., Ueda, N., Akutsu, T.: Graph kernels for molecular structures – activity relationship analysis with support vector machines. Journal of Chemical Information and Modeling 45(4), 939–951 (2005)
5. Ralaivola, L., Swamidass, S., Saigo, H., Baldi, P.: Graph kernels for chemical informatics. Neural Networks 18(8), 1093–1110 (2005)
6. Schenker, A., Bunke, H., Last, M., Kandel, A.: Graph-Theoretic Techniques for Web Content Mining. World Scientific, Singapore (2005)
7. Le Saux, B., Bunke, H.: Feature selection for graph-based image classifiers. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 147–154. Springer, Heidelberg (2005)
8. Harchaoui, Z., Bach, F.: Image classification with segmentation graph kernels. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
9. Luo, B., Wilson, R., Hancock, E.: Spectral embedding of graphs. Pattern Recognition 36(10), 2213–2223 (2003)
10. Neuhaus, M., Bunke, H.: An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 180–189. Springer, Heidelberg (2004)
11. Bunke, H., Dickinson, P., Kraetzl, M., Wallis, W.: A Graph-Theoretic Approach to Enterprise Network Dynamics. Progress in Computer Science and Applied Logic (PCS), vol. 24. Birkhäuser, Basel (2007)
12. Asuncion, A., Newman, D.: (UCI machine learning repository) University of California, Department of Information and Computer Science, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
13. Bunke, H., Vento, M.: Benchmarking of graph matching algorithms. In: Proc. 2nd Int. Workshop on Graph Based Representations in Pattern Recognition, pp. 109–113 (1999)
14. Foggia, P., Sansone, C., Vento, M.: A database of graphs for isomorphism and subgraph isomorphism benchmarking. In: Proc. 3rd Int. Workshop on Graph Based Representations in Pattern Recognition, pp. 176–187 (2001)

15. Xu, K.: Bhoslib: Benchmarks with hidden optimum solutions for graph problems (maximum clique, maximum independent set, minimum vertex cover and vertex coloring), http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm
16. Neuhaus, M., Bunke, H.: Bridging the Gap Between Graph Edit Distance and Kernel Machines. World Scientific, Singapore (2007)
17. Bunke, H., Riesen, K.: A family of novel graph kernels for structural pattern recognition. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 20–31. Springer, Heidelberg (2007)
18. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters 1, 245–253 (1983)
19. Alpaydin, E., Alimoglu, F.: Pen-Based Recognition of Handwritten Digits. Dept. of Computer Engineering, Bogazici University (1998)
20. Dosch, P., Valveny, E.: Report on the second symbol recognition contest. In: Wenyin, L., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 381–397. Springer, Heidelberg (2006)
21. Neuhaus, M., Bunke, H.: A graph matching based approach to fingerprint classification using directional variance. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 191–200. Springer, Heidelberg (2005)
22. Watson, C., Wilson, C.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology (1992)
23. Nene, S., Nayar, S., Murase, H.: Columbia Object Image Library: COIL-100. Technical report, Department of Computer Science, Columbia University, New York (1996)
24. Comaniciu, D., Meer, P.: Robust analysis of feature spaces: Color image segmentation. In: IEEE Conf. on Comp. Vision and Pattern Recognition, pp. 750–755 (1997)
25. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of The Fourth Alvey Vision Conference, pp. 147–151 (1988)
26. DTP, AIDS antiviral screen (2004), http://dtp.nci.nih.gov/docs/aids/aids_data.html
27. Kazius, J., McGuire, R., Bursi, R.: Derivation and validation of toxicophores for mutagenicity prediction. Journal of Medicinal Chemistry 48(1), 312–320 (2005)
28. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shidyalov, I., Bourne, P.: The protein data bank. Nucleic Acids Research 28, 235–242 (2000)
29. Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., abd Schomburg, D.: Brenda, the enzyme database: updates and major new developments. Nucleic Acids Research 32 (2004) Database issue: D431–D433