Available online at www.sciencedirect.com

ScienceDirect

Journal of Molecular Graphics and Modelling 25 (2007) 831–836

Journal of Molecular Graphics and Modelling

www.elsevier.com/locate/JMGM

High throughput processing of the structural information in the protein data bank

Zoltan Szabadka, Vince Grolmusz *

Department of Computer Science, Eötvös University, H-1117 Budapest, Hungary

Received 18 May 2006; received in revised form 19 July 2006; accepted 12 August 2006
Available online 12 September 2006

Abstract

The protein data bank (PDB) is the largest, most comprehensive, freely available depository of protein structural information, containing more than 37 500 deposited structures. On one hand, the form and the organization of the PDB seems to be perfectly adequate for gathering information from specific protein structures, by using the bibliographic references and the informative remark fields. On the other hand, however, it seems to be impossible to automatically review remark fields and journal references for processing hundreds or thousands of PDB files.

We present here a family of combinatorial algorithms to solve some of these problems. Our algorithms are capable to automatically analyze PDB structural information, identify missing atoms, repair chain ID information, and most importantly, the algorithms are capable of identifying ligands with their respective binding sites.

Keywords: Protein Data Bank; Biochemical databases; Database cleaning protein–ligand complex identification; Test databases; Protein–ligand docking

1. Introduction

The wealth of structural information stored in the protein data bank (PDB) [1] is the result of the work of thousands of researchers and millions of work-hours. By properly exploiting this wealth of information mankind may get solutions for a wide spectrum of health-related problems and illnesses, debilitating or killing hundreds of millions every year.

Widely available computational techniques, such as well developed data structures and algorithms, database applications together with the low-cost, reliable and high-power computer hardware would clearly imply the existence of a plethora of (fully automated) algorithmic solutions for handling the PDB.

Unfortunately, this does not hold. Most possible this discrepancy may be due to the fact that the PDB started to function as the depository of the crystallographic data, complementing journal publications: Researchers solved the structure of a protein, wrote a paper on the result, and deposited the data of the solution in the publicly available PDB.

The irregularities of the structure deposited (such as lacking atomic coordinates, broken chains, unidentified substructures) are mostly remarked in the cited publication and also in the remark-fields of the PDB file. The textual annotations, however, make the automatic processing of the protein-structures difficult.

1.1. In silico docking and the PDB

In silico docking studies are increasingly important in the search of new lead molecules in pharmacology. For testing any new docking method one needs a large library of crystallographically verified protein–ligand complexes.

The most well known such collection is the CCDC/ASTEX test set [2], which is hand-made, and contains 305 protein–ligand pairs.

For more reliable testing results researchers may need much larger sets, consisting of thousands of verified protein–ligand pairs. Such data-sets can only be made by algorithmic methods.

However, it is a surprisingly hard task to provide an automated method that reliably decides if a given structure contains a complex of a protein with its ligand.

This statement may be a little bit confusing, since atoms, carrying the HET label are not supposed to be in the peptide-chain,

* Corresponding author. Fax: +36 1 381 2156.
E-mail addresses: sinus@cs.elte.hu (Z. Szabadka), grolmusz@cs.elte.hu (V. Grolmusz).

1093-3263/$ – see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmgm.2006.08.004

so those structures that contains HET atoms other than the oxygen of the water would qualify for being a complex. Unfortunately, this is not the case. Metal ions, modified residues, and small molecules added in the crystallization all contain hetero-atoms, and they are not considered to be ligands.

We review several results from the literature here. Note, that even the numbers of complexes found have a large deviation in what follows.

The highly acclaimed pictorial database of the PDB, the PDBsum [3,4] contains 6498 ligands, bond to 10 564 proteins or RNA/DNA molecules.

Kinoshita and Nakamura [5] reviewed binding sites of heteroatoms, except for metal, PO_4, SO_4, modified residues, and covalently attached HET atoms from the PDB X-ray crystallographic database. After filtering out low resolution structures, they reported 26 359 binding sites on 14,330 PDB entries from the PDB [6].

Paul et al. [7] created a database from the PDB, containing in separate directories binding sites, ligands, and complexes for 4223 PDB entries. The database was selected also from the X-ray crystallographic data of the PDB, by the following method: First, a textual search was performed for any of the words ‘‘complex,’’ ‘‘inhibitor,’’ and ‘‘with,’’ and the resulting files were saved. Then low resolution structures, superseded entries, entries with high-molecule weight ligands, unwanted macromolecules, co-factors were thrown out. Then, using both the HET and the SEQRES fields, the nature of the ligands were identified, also using a list of PDB-ligands compiled in [3].

These results show that counting the protein–ligand complexes in the PDB is not a straightforward task.

2. Methodology

In this section, we describe our analytic method for deriving reliable information from the sometimes unreliable PDB files.

The advantage of our method relative to the earlier ones:

- protein–ligand complexes are identified reliably,
- missing residues and atoms in chains are handled properly, that is, even if several atoms are missing from a chain our algorithm will still not recognize the parts as distinct chains,
- moreover, placeholders are inserted into chains for missing residues/atoms, denoting that the objects were not measured crystallographically, but—according to the more reliable sequence information—they should be there: This way our algorithm ‘‘repairs’’ faulty PDB’s, or recognizes that flexible chain sequences are present. Note, that we do not even try to predict the atomic coordinates of the missing residues/atoms: It would be unrealistic and misleading ‘‘to freeze’’ highly flexible and fast-moving regions to any arbitrary or even computed position.
- Ligands are identified without using the HET-atom labels, properly handling modified residues and small artifacts, due to crystallization protocols. We collected a—surprisingly long—list of modified residues (see Table 2).

2.1. Defining a graph

For any given PDB entry a graph is defined where the atoms are the vertices, and the covalent bonds between atoms are the edges. More precisely, we add an edge between two atoms if their distance is less than 1.25 times the sum of their covalent radii. (The rate 1.25 was not chosen randomly. We screened all those covalently bound atom-pairs, which were marked in some connection table of the PDB Chemical Component Dictionary (formerly PDB HET group dictionary, http://deposit.pdb.org/het_dictionary.txt), or known to be bound in a polypeptide chain, and found that less than 1/10 000 fraction of those are farther than 1.25 times the sum of their respective covalent radii).

Now, if we are given this graph, how can we distinguish the protein chains from their ligands?

Our first idea was to take the connected components of this bond-graph, and define the protein chains as the relatively large, and the ligands as the relatively small connected components. But this approach turned out to be inadequate for several reasons:

- First, the disulphide bridges are covalent bonds, that can connect two different protein chains. One can easily deal with this case, for these bonds occur only between the sulphur atoms of two cysteine residues.
- A more serious problem is that there are several PDB entries, where atomic coordinates for entire amino acids are missing. If this occurs in the middle of a protein chain, then the component detecting algorithm will identify this as two or more different chains, or if one of the parts is too small, it will think of it as being a ligand.
- Another problem is that there are ligands that can bind covalently to a protein chain, so the above algorithm will not find them.

Consequently, only the coordinates of each atom in a PDB entry may not be enough to properly decompose a complex to chains and ligands.

However, the PDB files contains also the amino acid sequence information (SEQRES) of proteins, and this information describes the covalent structure of the protein chain without any doubt.

First we consider the chain-identifiers from the SEQRES records. The small peptides and nucleotides consisting of less than 10 residues will be considered as ligands.

The remaining chains with at least 10 residues will be reviewed next.

For a given chain-identifier i we compile a sequence of residues R_i from the atomic coordinate section of the PDB file as follows:

- for the fixed chain-identifier i we will look for the residues between the first occurrence of the identifier of the chain and either a TER record or the first occurrence of another chain identifier;
- between the just defined limits,

if the sequence of the residues with three-letter codes are known—allowing even unknown labels (UNK)—this sequence will be copied to R_i.

if the sequence of the chain is unknown, then each residue in the above described part of the entry will be included in R_i.

2.2. *Patching chains with residues*

After selecting the residues found in the PDB entry for each chain, we compare this sequence against the list given in the SEQRES records. Next the residues with missing atomic coordinates will be inserted into the sequence. This is done as follows (c.f., Fig. 1):

- First we make chain fragments from the residues with given coordinates.
- Next a graph-edge is added, connecting adjacent residues in the order they appear in the coordinate section, if they are covalently connected, so we get an ordered list of chain fragments.
- We try to match the sequence of these chain fragments against the sequence of the whole chain, given in the SEQRES field, in the correct order.

If one and only one such matching exist between the whole chain and the chain fragments, it means that the place of the chain fragments in the sequence of the whole chain is found, and we can insert the missing residues between the fragments.

In this matching algorithm, an UNK-labeled residue matches any other residue.

After inserting the missing residues we renumber the residues for each chain, starting from 1, to the number of residues. The original insertion codes are removed, and the newly inserted residues are given an insertion code ‘‘M,’’ denoting a missing residue.

2.3. *Patching residues with atoms*

We cannot only add missing residues to the chains, but also missing atoms to the residues. For this, we need the structural information for each residue that can be found in a protein chain.

This might seem to be an easy task, for there are only 20 amino acids, that commonly make up a protein chain.

But, surprisingly, there are several other modified amino acids or other HET groups, that are integrated into the backbone of a peptide chain. We obtained the structural information for these residues from the PDB Chemical Component Dictionary (formerly PDB HET group dictionary. http://deposit.pdb.org/het-dictionary.txt), where the structural information for each HET group found in the PDB is given.

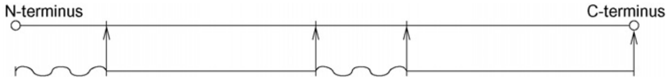

Fig. 1. Inserting missing residues into the sequence. The upper line represents the theoretical sequence of the chain found in the SEQRES records, the two lower straight lines are the found chain fragments, and the curly lines are the inserted residues.

So comparing the theoretical structure of each residue in a chain with the atoms for which coordinate information is found in the entry, we inserted the missing atoms into the residues, marking them with an ‘‘M’’ in the alternate location indicator. (The original alternate location indicators were removed, for we ignored the atoms with other than empty or ‘‘A’’ indicators.) While looking for missing atoms, we ignored the oxygens with atom name OXT, for they are only found in the C-terminus amino acids.

2.4. *Counting missing atoms*

Now that we inserted each missing residue and missing atom, we can answer the important question: How many atoms are missing the coordinate information in a PDB entry. Of course, we do not count hydrogen atoms, for they are usually missing from the PDB file.

This information can be important when we want to select a set of PDB entries, to use it for testing different docking and binding site predicting algorithms: PDB entries with fewer missing atoms can be used for more reliable tests.

2.5. *Detecting flexible loops*

We should remark, that missing atoms are usually a sign of mobile loop or string in the protein-crystal, since flexible atoms will not give usable electron density maps. Consequently, mapping missing atoms this way may help to automatically identify flexible protein parts, and these parts may have biological function (i.e., binding certain ligands).

2.6. *Our definition of ligands*

At this point we have selected the atoms from any given PDB entry that are parts of a protein or DNA chain. The next step is to find the ligands among the remaining atoms. First we select the water molecules—the ones with residue name HOH—and remove them from the set of possible ligand atoms. Then metal and other small ions are selected, that will not be considered as ligands. A complete list of residue names, that were considered as ions can be found in Table 1.

All the remaining atoms will form the set of ligand atoms. Within this set, we can use the above described component detecting algorithm, so a ligand is defined as a connected component of the graph formed by the ligand atoms as vertices and the covalent bonds between the ligand atoms as the edges. The components are determined with a simple breadth first search algorithm, which can also be used to detect the covalent bonds between the ligands and the protein chains, if we formerly build the covalent bonds between the atoms of the ligands and the chains.

Table 1
The list and frequencies of the ions in the PDB

HETID	Frequency	HETID	Frequency	HET ID	Frequency	HET ID	Frequency
SO4	2657	CS	19	F	4	PR	1
CA	2182	MO4	18	GD3	4	3MT	1
ZN	1795	OXL	18	MO1	4	CD3	1
MG	1426	ZN3	18	HO	4	NI3	1
CL	1138	ALF	17	PBM	4	CD5	1
NA	815	BCT	17	BA	4	WO5	1
PO4	719	OAA	15	MW2	4	IN	1
MN	573	TL	15	MLI	4	IR3	1
ACT	352	PB	14	CD1	3	IR	1
K	307	MO5	14	HGC	3	SB	1
CU	302	SM	14	AL	3	O4M	1
FE	292	SR	14	PER	3	KO4	1
CD	217	YB	14	LCP	3	LCO	1
HG	198	AU	12	1CU	3	ATH	1
Nl	140	BEF	12	OCL	3	CE	1
CO	117	OF1	11	OS	3	AU3	1
FE2	107	LI	11	TRA	2	MAC	1
CO3	74	MW1	10	GA	2	OC3	1
NO3	64	EMC	10	HAI	2	SE4	1
CAC	62	MLT	10	AG	2	MH2	1
BR	44	6MO	9	3NI	2	OC4	1
PI	43	MOO	8	LA	2	E4N	1
IOD	42	CUA	8	TB	2	MH3	1
AZI	39	ZNO	8	CUZ	2	NA2	1
CYN	37	NET	8	OC2	2	OC5	1
NH4	36	MMC	7	AUC	2	NAO	1
OH	33	MN3	6	DMI	2	SOH	1
IUM	29	RB	6	MW3	2	EU3	1
MO6	29	MOS	6	OC6	2	NAW	1
CU1	29	OC1	6	EU	1	TCN	1
MO3	25	LU	6	BO4	1	BF4	1
ZN2	24	CR	6	FPO	1	OCM	1
UNX	23	2HP	6	OCN	1	THE	1
PO3	23	2MO	5	TMA	1		
MO2	20	NI1	5	PD	1		
VO4	20	OF3	5	2OF	1		
SCN	20	4MO	5	W	1		
SO3	20	CHT	5	3CO	1		
PT	19	YT3	5	NI2	1		
NO2	19	Y1	5	OF2	1		

The ions are identified by their HET I.D. (Available at http://deposit.pdb.org/heLdictionary.txt).

3. Results and discussion

First, from the 26 485 PDB entries[1] those were selected which did not have MODEL/ENDMDL records. Thus we got 23 580 entries. Then for each such entry the number of missing atoms from the protein chains was determined. The result can be seen on Fig. 2.

A remarkable finding is that very few PDB structures have 1–10 missing atoms. This fact can be interpreted as follows: the missing atoms correspond to flexible chain segments, yielding not-evaluable electron density maps. Too short chain segments, however, cannot be flexible at all. If this interpretation of the missing atoms is correct, then Fig. 2 shows that flexible loops are quite common in the PDB structures. This finding may question the correctness of the "rigid protein–flexible ligand" docking methods in the case of about 13 000 PDB structures.

The most important result of our study was the selection of a set of PDB entries, that contain protein–ligand complexes, satisfying the following criteria:

1. The number of atoms in the chains are between 1000 and 10 000. The upper bound is just a technical criterion, excluding too large entries.
2. The number of missing atoms is at most one percent of the number of atoms. We added this criterion since we intended to generate test-sets for rigid protein docking.
3. The ligand has more than 10 and less than 100 atoms, since we are interested mostly in lead-like ligands.
4. The ligand is not bound covalently to the protein, since covalent bounds are usually not favored as leads.

[1] On the RCSB DVD published in Winter, 2004.

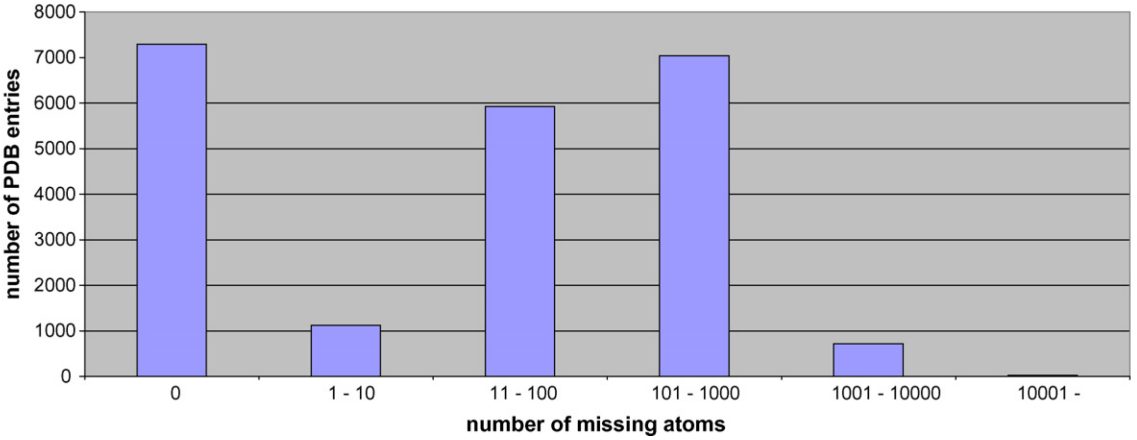

Fig. 2. The distribution of the number of the missing atoms in the PDB files. Note, that typically either no atoms are missing, or more than 10 are missing.

Table 2
The list of the 293 modified amino acids present in the PDB

Resid.	PDB	Resid.	PDB	Resid.	PDB	Resid.	PDB	Resid.	PDB	Resid.	PDB	Resid.	PDB	Resid.	PDB
MSE	487d	CXM	1aiq	NEM	1c0f	ASQ	1dc8	ARO	1ffu	BFD	1j97	PR3	3nuc	MCB	1tkq
NH2	487d	PGA	2aig	CSW	1c0t	3AH	1dgh	CSZ	1ffu	CZZ	1j9b	PEC	5nuc	FMA	1tys
ACE	1a0r	MYR	1al2	DLE	1c4d	2PP	1dit	CH2	1fph	TRQ	1jju	HTI	1nwh	LLY	1ucw
NPH	1a18	TRN	1am7	CYQ	1c4w	SCY	1dm3	ALS	1fsu	SEG	1jl0	ALC	1nzq	OPR	1ucy
CME	1a1v	DAS	1an1	GLC	1c58	CSO	1dmp	S1H	1fw3	FOR	1jlx	SMF	1nzq	DSN	1uhg
TYS	1a2c	CBX	1an5	STY	1c51	CMT	1doa	FTR	5fwg	ETA	1jno	MCL	1o5k	NRQ	1uis
PYX	1a2d	ACY	1at5	SEB	1c9m	1LU	1ds2	YOF	3fyg	PHL	1joh	LEF	1ogw	MSO	1uzx
MIS	1a2q	SNN	1at5	CEG	1cap	2LU	1ds3	TRO	1g3p	143	1jvn	DHA	1oln	CLB	1vsb
BUC	1a2u	IAS	1at6	GAL	1cap	CCS	1dss	2MR	1g42	CSR	1jzw	PYT	1oln	OCY	1vsh
TPQ	1a2v	ASX	2atc	GCU	1cap	CSA	1dwq	CRQ	1g7k	PHD	1k68	QUA	1oln	SBD	3vsb
PTR	1a31	FGL	1auk	MAN	1cap	CYF	1dzh	CHG	5gds	PCC	1km8	ROP	1oln	BTR	1wct
5HP	1a39	BHD	1aut	OMT	2cag	ALY	1e6i	HAC	5gds	SIN	1kqe	TSI	1oln	GTH	1wct
T29	1a3b	SBL	1av7	CGN	3cao	GLH	1e79	HMF	5gds	CRG	1kyp	TZB	1oln	PBI	2yfp
T16	1a3e	CLD	1avt	NLE	1cfn	DBY	1eba	NAL	5gds	YCM	1l0q	TZO	1oln		
EFC	1a3t	PVL	1aw8	GLX	2ci2	OAS	1ebv	MME	1gk8	TYT	1lvn	XAA	1oln		
C6C	1a3u	TYI	2axe	GPL	1ckn	6HC	1ec4	SMC	1gk8	LYX	1m1d	XBB	1oln		
C5C	1a3v	CH3	1ay2	GLQ	1cmx	6HG	1ec4	SME	1gkf	VOL	1m24	CYD	1ox4		
CSP	1a5y	SAC	1b0b	GLZ	1cmx	6HT	1ec4	SUI	1gkt	CY4	1m4t	5CS	1ox5		
DPR	1a7y	DPN	1b0q	OCS	1cs8	AEI	4eca	SEC	1gp1	TRW	1mg3	NYC	1oxd		
DTH	1a7y	MEN	1b33	CRF	1cv7	DIV	1ee7	CAY	1gt	AEA	1mhh	4IN	1oxf		
DVA	1a7y	CSB	1b6g	DMT	1cwb	TPL	1ee7	HSO	1h3j	DHN	1mik	5ZA	1oxf		
MVA	1a7y	ABA	1b6j	MNL	1cwc	APP	1efr	PIA	1h6r	NC1	1mws	LCX	1p6b		
PXZ	1a7y	HTR	1b80	DSE	1cwh	BAL	1efr	AGM	1hbm	PG1	1mwt	DOH	1pfx		
SAR	1a7y	SVA	1b8j	MNV	1cwj	CPI	1efr	GL3	1hbm	MC1	1mwu	IIL	1q4v		
CTH	1a7z	CRO	1b9c	MSA	1cwj	TLX	1efr	MGN	1hbm	ORN	1n0x	DPL	1qfi		
H5M	1a7z	BMT	1bck	TBM	1cwj	CYM	1eh7	MHS	1hbm	FGP	1n2k	LYZ	1qgw		
MAA	1a7z	DAL	1bck	TMD	1cwk	BCS	1eh8	PAA	1hbt	TYN	1nbm	NCB	1qmv		
POM	1a7z	MLE	1bck	MGY	1cwl	ASI	1ejc	CSY	1hcj	MN1	1nlo	ASB	1qq6		
LLP	1a8i	HYP	1bdk	MHL	1cwl	MHO	1ek0	CAF	1hyv	MN2	1nlo	TRF	1qs7		
PCA	1a8j	IGL	1bdk	IML	1cwm	CCY	1emk	MLY	1i84	MN7	1nlo	ABU	1qur		
CYG	1a9x	OIC	1bdk	TMB	1cwm	DOA	1eoj	HSL	1idg	MN8	1nlp	CR5	1qyq		
KCX	1aa1	TIH	1bdk	VAD	1cwo	HIC	1eqy	PNL	1ihs	A66	1nr8	LAL	1r1g		
CSE	1aa6	IIC	1bfp	DAR	1czq	STA	3er5	FBE	1iht	APN	1nr8	HLU	1rov		
HMR	1abi	CAS	1bhl	DCY	1czq	MPR	1etl	PTL	1iht	C66	1nr8	4HT	1ru9		
CSD	1acd	CSS	1bi0	DGL	1czq	NEP	1eud	ACA	1ilq	CPN	1nr8	GHG	1ru9		
CGU	1ad7	EHP	1biq	DHI	1czq	PN2	1f80	M3L	1irv	GPN	1nr8	AR4	1s2d		
SCH	1aex	SEP	1bkx	DTR	1czq	CSX	1f8w	MLZ	1iv8	T66	1nr8	CR0	1s6z		
AYA	1ah3	TPO	1bkx	PAssS	1d5w	3PA	1fav	DAH	1ivv	TPN	1nr8	ASE	1sa1		
CEA	2ahj	FME	1bq9	TYQ	1d6u	4BA	1fav	AHP	1j4x	AHB	1nt0	PRS	1sav		
AIB	1ai1	SNC	1buw	TYY	1d6z	GGL	1fav	DAB	1j73	SCS	2nuc	PAQ	1spu		

The modified amino acid is identified by its HET I.D. (available at http://deposit.pdb.org/het-dictionary.txt), right to it with lower case an example PDB I.D. is given, where it occurs.

We have found 8202 such protein–ligand pairs in 3784 PDB entries.

We also compiled the list and frequencies of the ions in the PDB, as seen on Table 1.

The list of modified residues—those that are part of an at lest 10 residue long protein sequence and are not among the 20 natural amino acids—is given in Table 2.

Here, we give a table of the most frequent protein chains. The ‘‘multiplicity’’ means the number of occurrences of the very same chain-sequence under different PDB codes.

Multiplicity	SwissProt	EC	Protein name	Species
165	P00760	3.4.21.4	Trypsin	Bovine
142	P00698	3.2.1.17	Lysozyme	Chicken
125	P00734	3.4.21.5	Thrombin light chain	Human
111	P00734	3.4.21.5	Thrombin heavy chain	Human
92	P06746	2.7.7.7	DNA polymerase beta	Human
84	P69905	–	Hemoglobin A	Human
76	P61823	3.1.27.5	Pancreatic Ribonuclease	Bovine
67	P68871	–	Hemoglobin B	Human
52	P01315	–	Insulin A	Human

Fifteen thousand six hundred eighty four different sequences were found in our study in the PDB; the multiplicities of the sequences (the number of different PDB entries containing them) are shown below:

Multiplicity	No. of chains	Multiplicity	No. of chains
1	11482	27	3
2	2225	28	3
3	806	29	1
4	396	30	2
5	199	31	2
6	138	32	1
7	79	33	1
8	80	37	1
9	53	39	1
10	31	40	1
11	18	41	1
12	19	44	2
13	24	45	1
14	21	47	1
15	9	48	1
16	4	49	1
17	31	52	1
18	10	67	1
19	2	76	1
20	6	84	1
21	4	92	1
22	5	111	1
23	3	125	1
24	3	142	1
26	3	165	1

4. Note on implementation

Our algorithms were written in C++ programming language under Linux operation system. The running time of the processing of the whole PDB was less than 4 h on a low-end workstation (1.2 GHz AMD Athlon processor, 1.5 Gb of memory).

5. Sample output availability

More than 1000 processed PDB files are freely accessible at the site: http://www.math-for-health.com/new_page_8.htm.

References

[1] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, Nucleic Acids Res. 28 (2000) 235–242.

[2] J.W.M. Nissink, C. Murray, M. Hartshorn, M.L. Verdonk, J.C. Cole, R. Taylor, A new test set for validating predictions of protein–ligand interaction, Proteins 49 (4) (2002) 457–471. http://www.ccdc. cam. ac.uk/products/life_sciences/validate/astex/index.php.

[3] R.A. Laskowski, PDBsum: Summaries and analyses of PDB structures, Nucleic Acids Res. 29 (2001) 221–222.

[4] R.A. Laskowski, V.V. Chistyakov, J.M. Thornton, Pdbsum more: New summaries and analyses of the known 3D structures of proteins and nucleic acids, Nucleic Acids Res. (2005).

[5] K. Kinoshita, H. Nakamura, Identification of the ligand binding sites on the molecular surface of proteins, Protein Sci. 14 (2005) 711–718.

[6] K. Kinoshita, J. Furui, H. Nakamura, eF-site and PDBjViewer: Database and viewer for protein functional sites, Bioinformatics 20 (2004) 1329–1330.

[7] N. Paul, E. Kellenberger, G. Bret, P. Muller, D. Rognan, Recovering the true targets of specific ligands by virtual screening of the protein data bank, Proteins: Struct., Funct. Bioinform. 54 (4) (2004) 671–680. http://bioinfo-pharma.u-strasbg.fr/scpdb/scpdb _form.html.