BMC Bioinformatics

BioMed Central

Software

Open Access

# MolTalk – a programming library for protein structures and structure analysis

Alexander V Diemand*[1] and Holger Scheib[2]

Address: [1]University of Lausanne and Swiss Institute of Bioinformatics, 155, chemin de Boveresses, 1066 Epalinges s/Lausanne, Switzerland and [2]University of Geneva and Swiss Institute of Bioinformatics, Centre Médicale Universitaire, 1, rue Michel-Servet, 1211 Geneva 4, Switzerland

Email: Alexander V Diemand* - alexander.diemand@isb-sib.ch; Holger Scheib - holger.scheib@isb-sib.ch

* Corresponding author

Published: 19 April 2004

*BMC Bioinformatics* 2004, **5**:39

This article is available from: http://www.biomedcentral.com/1471-2105/5/39

Received: 27 October 2003
Accepted: 19 April 2004

© 2004 Diemand and Scheib; licensee BioMed Central Ltd. This is an Open Access article: verbatim copying and redistribution of this article are permitted in all media for any purpose, provided this notice is preserved along with the article's original URL.

## Abstract

**Background:** Two of the mostly unsolved but increasingly urgent problems for modern biologists are a) to quickly and easily analyse protein structures and b) to comprehensively mine the wealth of information, which is distributed along with the 3D co-ordinates by the Protein Data Bank (PDB). Tools which address this issue need to be highly flexible and powerful but at the same time must be freely available and easy to learn.

**Results:** We present MolTalk, an elaborate programming language, which consists of the programming library *libmoltalk* implemented in Objective-C and the Smalltalk-based interpreter MolTalk. MolTalk combines the advantages of an easy to learn and programmable procedural scripting with the flexibility and power of a full programming language.

An overview of currently available applications of MolTalk is given and with PDBChainSaw one such application is described in more detail. PDBChainSaw is a MolTalk-based parser and information extraction utility of PDB files. Weekly updates of the PDB are synchronised with PDBChainSaw and are available for free download from the MolTalk project page http://www.moltalk.org following the link to PDBChainSaw. For each chain in a protein structure, PDBChainSaw extracts the sequence from its co-ordinates and provides additional information from the PDB-file header section, such as scientific organism, compound name, and EC code.

**Conclusion:** MolTalk provides a rich set of methods to analyse and even modify experimentally determined or modelled protein structures. These methods vary in complexity and are thus suitable for beginners and advanced programmers alike. We envision MolTalk to be most valuable in the following applications:

1) To analyse protein structures repetitively in large-scale, i.e. to benchmark protein structure prediction methods or to evaluate structural models. The quality of the resulting 3D-models can be assessed by e.g. calculating a Ramachandran-Sasisekharan plot.

2) To quickly retrieve information for (a limited number of) macro-molecular structures, i.e. H-bonds, salt bridges, contacts between amino acids and ligands or at the interface between two chains.

3) To programme more complex structural bioinformatics software and to implement demanding algorithms through its portability to Objective-C, e.g. iMolTalk.

4) To be used as a front end to databases, e.g. PDBChainSaw.

## Background

The major demand from Life Sciences towards bioinformatics today is to combine the often heterogeneous information available and make it easily accessible to a broad range of users. In the past, these efforts concentrated on coping with the overwhelming amount of data that entered and still enter nucleotide and protein sequence databases [1,2]. Today, other information sources, such as protein structures, subsequently come under the spotlight of a broader scientific community.

In contrast to the sequence world, only one central data resource exists for protein structures, the Protein Data Bank (PDB) [3]. Despite the undisputed advantage of having all structural data available from one source in a common file format, protein structures impose a new level of complexity. They carry information about where in space the adjacent residues of a protein sequence are located. Furthermore, protein structures provide insights into the spatial environment of an amino acid, which is different from its sequence neighbourhood, as well as into its interactions with other residues or heterogeneous ligands. This wealth of information contains answers to questions as diverse as to how proteins function or what compounds may interact with a given protein. However, these answers often remain inaccessible to a broader scientific community.

To overcome this information gap, we developed MolTalk. MolTalk consists of a programming library implemented in Objective-C [4] that maps PDB structure files to object space as well as of a scripting language based on Smalltalk [5]. Moreover, MolTalk provides numerous methods that enable both the novice as well as the expert structural bioinformatician to rapidly develop software tailored towards their individual needs and to allow for novel insights from protein structure analyses. As an application for MolTalk we describe PDBChainSaw, a mirroring and data extraction routine for PDB files.

## Implementation

MolTalk is composed of two functional parts: (1) the programming library *libmoltalk* and (2) MolTalk, the Smalltalk interpreter. The *libmoltalk* library implements classes (Figure 1) in Objective-C [4] whereas the interpreter MolTalk is based on StepTalk [5], a Smalltalk interpreter for GNUstep [6]. The interpreter interacts with all classes defined in *libmoltalk* and is used as a front end to this library.

The classes implemented in *libmoltalk* are summarised in groups, namely "structural", "mathematics", and "others". Their complexity and flexibility vary as indicated by the labels "Basic" and "Xtra" (Table 1). "Basic" classes can be used by even novice users without special training, whereas classes labelled "Xtra" indicate a higher level of potential difficulty for a user, but allow often, at the same time, a higher degree of flexibility in software development (for details, please refer to the manual pages at http://www.moltalk.org/Manual.html.

Each class consists of a set of methods, which again are labelled either "Basic" or "Xtra". Independent of their class, methods can be organised into (1) "basic features", (2) "extended features", (3) "mathematical functions", and (4) "others". "Basic features" enable mapping into object space and querying. "Extended features" can be further sub-divided into "operations" and "manipulations". "Operations" include e.g. superimposition, structural alignment, and transformation, respectively. With "manipulations" chains, residues or atoms can be added to or removed from a structure. "Mathematical functions" allow the calculation of vectors and matrices to perform spatial transformations. The features summarised in "others" regulate input and output. In Table 1, a list of the potentially most important methods and classes of the group "Structure" is provided.

## Results and Discussion

### *PDBChainSaw*

Extracting and deriving knowledge from PDB files remains a non-standard procedure to date. Therefore, we developed MolTalk to provide and facilitate access to this valuable information. As an example for a possible use of MolTalk, we present PDBChainSaw, a relational database of protein structure chains, which is used in the ModSNP project to model and analyse non-synonymous single nucleotide polymorphisms in human proteins [7]. For PDBChainSaw, PDB files are parsed and a defined set of parameters is extracted and stored in a relational model database (Table 2, Figure 2) [8], which is updated weekly and available from the project site for download http://www.moltalk.org/PDBChainSaw.html. In PDBChainSaw the amino acid sequence of each chain in a protein structure is reconstructed from co-ordinates but also takes into account the information given in the records for modified residues (MODRES in PDB file format [9]). Unfortunately, numerous structure files lack residues, which were not resolved during structure determination but are part of the native protein. These amino acids are responsible for breaks in the main chain of a structure and are included in the structure-derived sequences. Main chain breaks are filled with the unknown residue 'X', one for each missing amino acid using the method getSequence of class Chain (Table 2). This has several advantages. Firstly, missing and chemically modified residues are considered and are included into the sequence. Missing residues can be identified quickly, e.g. when comparing a structure-derived sequence to a native protein as in homology modelling, and the respective regions easily

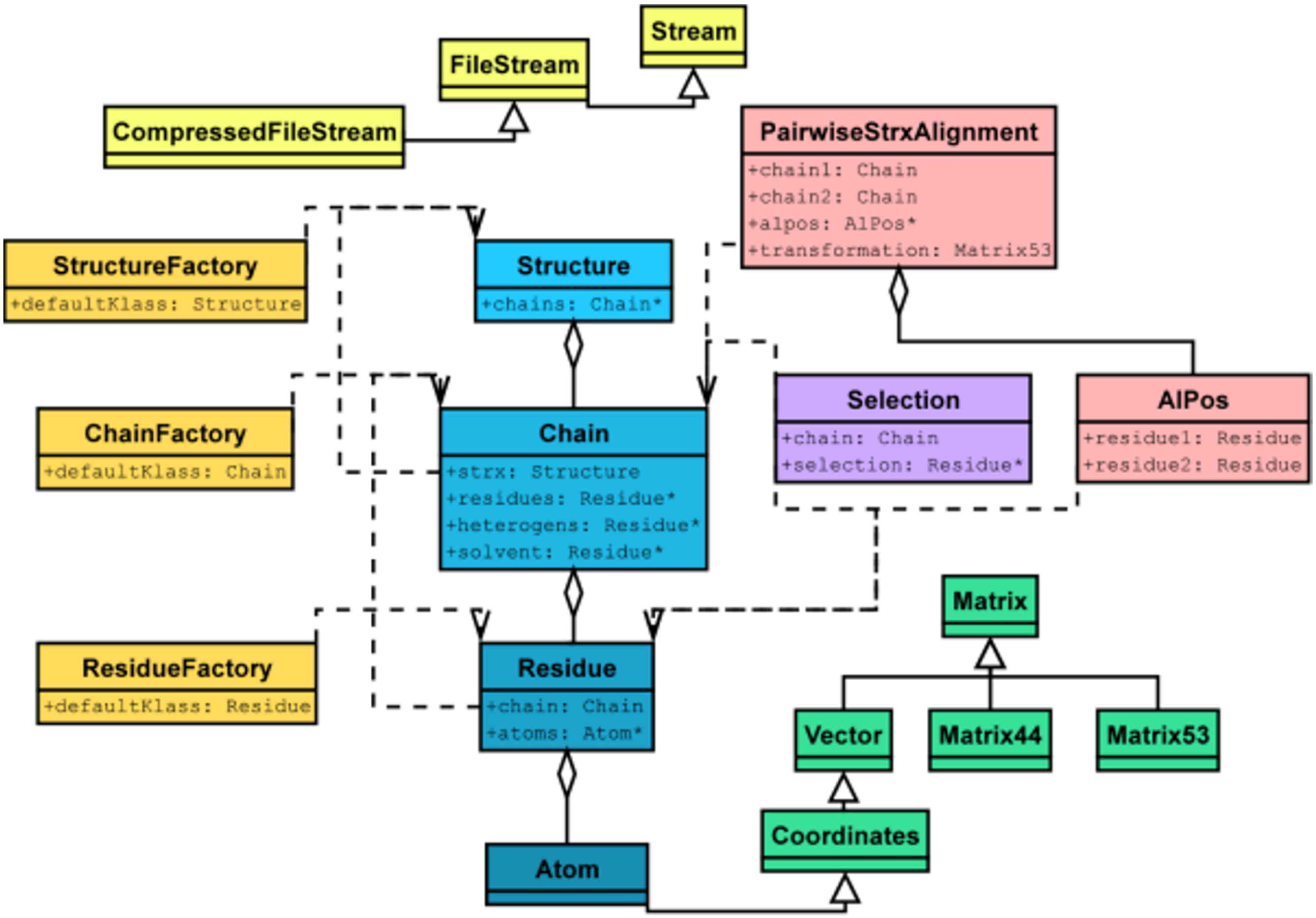

**Figure 1**

**Class diagram of *libmoltalk* showing interdependencies and inheritance.** The coloured rectangles symbolise the classes implemented in *libmoltalk*. The inheritance between classes is shown as lines starting with a triangle at the super-class and ending at the child class(es). Aggregates of instances of classes are marked with a line ending at the container class with a rhomb. Other dependencies between classes are shown as dashed lines. Below the class name in bold, the most important fields of a particular class are given. Classes were grouped according to their function as follows: in yellow classes for input/ output, in blue structural classes, in orange factories, in green mathematical classes, in pink classes representing a structural alignment, and in magenta the selection class.

assessed. Secondly, to store co-ordinate-derived sequences only allowed us to include all PDB structures into the PDBChainSaw database, including older structures for which the native sequence is not given in the PDB file due to missing or incorrect SEQRES lines (e.g. 1KGA, 1PGI, 2PGK). The same may apply to model structures or other structure files which do not comply with all conventions of the PDB file format [9]. It has to be stated, however, that although this approach is convenient to identify positions or regions in proteins for which amino acids could not be resolved experimentally, using such sequences to interpret an alignment quantitatively or in the context of evolutionary similarity may be misleading. Both the alignment score and thus the expectation value would be affected by the non-standard symbol 'X'. For standard amino acids, as present in the SEQRES lines, similarity was defined quantitatively and this similarity is generally applied to construct sequence alignments by using similarity matrices. For non-standard residues, these quantitative relationships do not exist and such sequences can therefore not be used to interpret quantitative relationships between sequences. To address this point and to allow for the highest possible flexibility in protein structure analysis and software development, a method is pro-

**Table 1: The ten classes summarised in group "structure" with labelled difficulty level and a selection of methods available.**

| Class Name | Label | Methods |
|---|---|---|
| Structure | Basic | Returns 4-letter PDB code<br>Returns HEADER, TITLE, REVDAT lines<br>Extracts date from HEADER line<br>Returns type of experimental method<br>Returns resolution as in REMARK2 lines<br>Writes out complete structure to a stream in PDB format<br>Returns enumerator over all chains<br>Returns chain for a given code<br>Removes a chain from structure<br>... |
| Structure Factory | Xtra | Reads structure from directory or file<br>Offers parsing options from directory or file<br>... |
| Chain | Basic | Returns code of this chain (as string/number)<br>Returns chain identifier consisting of PDB and chain code<br>Returns COMPND and SOURCE lines, EC code Transforms all residues/atoms in chain by transformation matrix<br>Returns number of residues (amino acids and nucleic acids), standard amino acids, heterogeneous residues, solvent residues<br>Provides access to residues, heterogeneous residues, solvent residues<br>Adds new residue, heterogen, new solvent molecule to chain<br>Removes a residue, heterogen, solvent molecule from chain<br>Derives amino acid sequence from connected residues<br>Derives amino acid sequence with filled gaps ("X") where missing residues occur<br>Returns amino acid sequence from SEQRES entry<br>Computes geometric hashing table of all residues<br>Finds residues in chain which are close to given co-ordinates based on geometric hashing<br>... |
| Chain Factory | Xtra | Creates a new chain with given code |
| Residue | Basic | Returns the residue name/number<br>Returns the name of the standard residue as the base of this modified residue as in MODRES lines<br>Returns description of residue modification as given in MODRES lines<br>Translates residue name into amino acid one letter code<br>Adds new atom to residue<br>... |
| Residue Factory | Xtra | Creates new residue with number and name<br>... |
| Atom | Basic | Returns atom name/number<br>Returns temperature factor for an atom<br>Returns chemical element<br>Returns partial charge of atom<br>Returns enumerator over all bonded atoms<br>Adds bond from this atom to given atom2<br>Removes all bonds<br>Removes bond to atom2<br>Sets atom to be of chemical type<br>... |
| Coordinates | Basic | Calculates Euclidian distance between two co-ordinates<br>Returns x, y, z from co-ordinates<br>Transforms co-ordinates by transformation matrix<br>... |

**Table 1: The ten classes summarised in group "structure" with labelled difficulty level and a selection of methods available.** *(Continued)*

| | | |
|---|---|---|
| Pairwise Structural Alignment | Basic | Provides access to first/second chain<br>Computes transformation based on superimposed chains<br>Re-computes transformation from selection of residue pairs<br>Calculates RMSD of structural alignment<br>Counts alignment positions in structural alignment<br>Counts aligned pairs only<br>Counts aligned pairs with distance below given cut-off<br>Reads external pairwise alignment from stream in T_Coffee format and re-computes structural alignment from this<br>Writes structural alignment to stream in T_Coffee library format<br>... |
| Selection | Basic | Counts number of residues in this selection<br>Returns enumerator over selected residues<br>Includes/excludes a single residue to/from selection<br>Adds all selected residues from selection2 to this selection<br>Structurally aligns selection1 to selection2 and returns the resulting transformation matrix<br>... |

vided in MolTalk to derive also the sequence of the native protein from the HEADER section (getSEQRES). Table 2 gives an overview of the three types of sequences that can be extracted from PDB files using the MolTalk methods getSequence (co-ordinate derived sequence with 'X' for missing residues), get 3D Sequence (co-ordinate derived sequence), and getSEQRES (sequence of the native protein).

### *What MolTalk can do*

The fact that MolTalk is an elaborate programming language represents also its major advantage over other freely available software with embedded macro languages, such as MolMol [10], SPDBV [11], and Rasmol [12]. GNUstep provides a rich object-oriented Application Programming Interface, API, and services, such as object locaters, calling methods in distant objects. These are all available in both MolTalk, the Smalltalk interpreter, and *libmoltalk*, the Objective-C programming library. A complete and thus powerful programming language, MolTalk can be used for the development of more complex software in structural bioinformatics, as revealed with PDBChainSaw. Other applications could be as diverse as to determine H-bonds and salt bridges in proteins, to measure distances and angles between atoms in order to assess the quality of a structure or to define contacting atoms between amino acids, nucleic acids and a ligand or residues of different chains. Moreover, two structures or a selection of residues therein may be superimposed and the RMSD calculated. Table 1 gives an overview of a selection of the most important methods summarised in group "structure". Several of these applications have been made available via the iMolTalk server http://i.moltalk.org[13].

To achieve a significant gain in execution speed, scripts in MolTalk can be easily ported to Objective-C and the functionality of these programs, compiled and linked with the *libmoltalk* library, remains the same. GNUstep classes and classes defined in MolTalk are still accessible.

MolTalk is particularly aimed at handling large and complex datasets and at implementing demanding algorithms. Short scripts as well as full programs can be written to quickly parse a PDB structure file or to recurrently analyse large-scale and in-depth efforts. Therefore, the target audience can range from a sophisticated software developer to an interested bench biologist. MolTalk positions itself clearly on the side of computational analysis in the emerging new field of structural bioinformatics. The applications provided so far on the MolTalk project page support this standpoint. Other programming libraries that have been developed recently appear to aim at a slightly different target audience and are closely related to experimental structure determination initiatives [14-16].

### *What MolTalk does not aim at*

MolTalk aims at the computational analysis of protein structures rather than protein structure prediction. The former can be performed using existing methods and measuring known parameters, whereas the latter requires the implementation of new methods and algorithms. However, as shown for PDBChainSaw, MolTalk can provide valuable input data for prediction methods. Currently, MolTalk as a purely structure-based programming suite does not include sophisticated methods to examine protein sequences. Neither does it provide a graphical user

**Table 2: A: Information stored in PDBChainSaw for the G chain of bovine mitochondrial F1-ATPase (1OHHG) as an exemplar. In the field "chainid", the PDB four-letter code is followed by the single character chain identifier. The field "chainid2" is the concatenation of the PDB code and the ASCII value of the chain identifier, in this case, "71" corresponding to "G". B: The two other options to return the sequence from a protein structure stored in PDB file format.**

| **A**<br>**Fields** | **Content** |
|---|---|
| pdb | 1OHH |
| code | G |
| chainid | 1OHHG |
| chainid2 | 1OHH71 |
| residues | 94 |
| aminoacids | 94 |
| heterogens | 0 |
| solvent | 0 |
| sequence from co-ordinates (inferred) | ATLKDITRRLKSIKNIQKITKSMKMVAAAKXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX<br>LCGAIHSSVAKQXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX<br>XXX<br>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXTTSE<br>QSAR MTAMDNASKNASEMIDKLTLTFNRTRQAVITKELIEIISGAAAL |
| organism | BOS TAURUS |
| compound | ATP SYNTHASE GAMMA CHAIN, MITOCHONDRIAL SYNONYM: BOVINE MITOCHONDRIAL F1-ATPASE GAMMA SUBUNIT |
| eccode | 3.6.1.34 |
| **B** | |
| sequence from co-ordinates without 'X' | ATLKDITRRLKSIKNIQKITKSMKMVAAAKLCGAIHSSVAKQTTSEQSARMTAMDNASKNASEMIDKLTLT<br>FNRTRQAVITKELIEIISGAAAL |
| sequence from SEQRES | ATLKDITRRLKSIKNIQKITKSMKMVAAAKYARAERELKPARVYGVGSLALYEKADIKTPEDKKKHLIIGVSS<br>DRGLCGAIHSSVAKQMKSEAANLAAAGKEVKIIGVGDKIRSILHRTHSDQFLVTFKEVGRRPPTFGDASVIAL<br>ELLNSGYEFDEGSIIFNRFRSVISYKTEEKPIFSLDTISSAESMSIYDDIDADVLRNYQEYSLANIIYYSLKE<br>STTSEQSARMTAMDNASKNASEMIDKLTLTFNRTRQAVITKELIEIISGAAAL |

interface nor supply interactive modelling capabilities, where the strengths of MolMol, SPDBV, and VMD [17] clearly lie.

### *Accessibility and effort to learn*

Access to MolTalk software and an extensive tutorial is provided via the MolTalk project page at http://bioinformatics.org/moltalk or http://www.moltalk.org. The tutorial is sub-divided into four parts: (1) MolTalk library, (2) Smalltalk interpreted scripting language, (3) information about the GNUstep Foundation, and (4) a comprehensive index. The MolTalk library section itself is sub-grouped into "Requisites", a "Class diagram", and "Classes". "Requisites" are information about installing a local version of *libmoltalk* and pre-requisites for compiling Objective-C code. The "Class diagram" provides an overview of the dependencies of the classes in MolTalk (Figure 1) together with a hint to the overall difficulty of the methods combined in this class. More detailed information on the attributes and methods of each class is provided in the sub-section "Classes". Again, the methods of each class are labelled either "Basic" or "Xtra" to indicate potential difficulties. Noteworthy, classes flagged as "Basic" can also contain methods of complexity "Xtra" and vice versa. Sub-grouping, together with an index system to highlight potential difficulties, i.e. for novice users, already proved useful internally to enable for straight-forward navigation through the tutorial pages. Therefore, users can easily identify available or desired classes and methods and apply the example scripts provided on the tutorial pages to their own problems. In the future, learning to MolTalk will become even easier through the iMolTalk server [13], where users can upload scripts and retrieve results without being obliged to install MolTalk locally.

## Conclusions

MolTalk is a freely available, well documented and thus easy to learn, robust, and clean implementation of object-oriented mapping of PDB files. The included interpreter for the Smalltalk scripting language allows for rapid software development, while retaining the option to port code to Objective-C, compile, and link with the MolTalk library. This opens the field for MolTalk to become a prominent protein structure analysis suite from small- to large-scale efforts, particularly when large and complex data sets need to be analysed automatically. Moreover, we regard MolTalk as a potential tool for benchmark analysis, i.e. for protein structure prediction methods and model evaluation, respectively. MolTalk may also serve as a data-

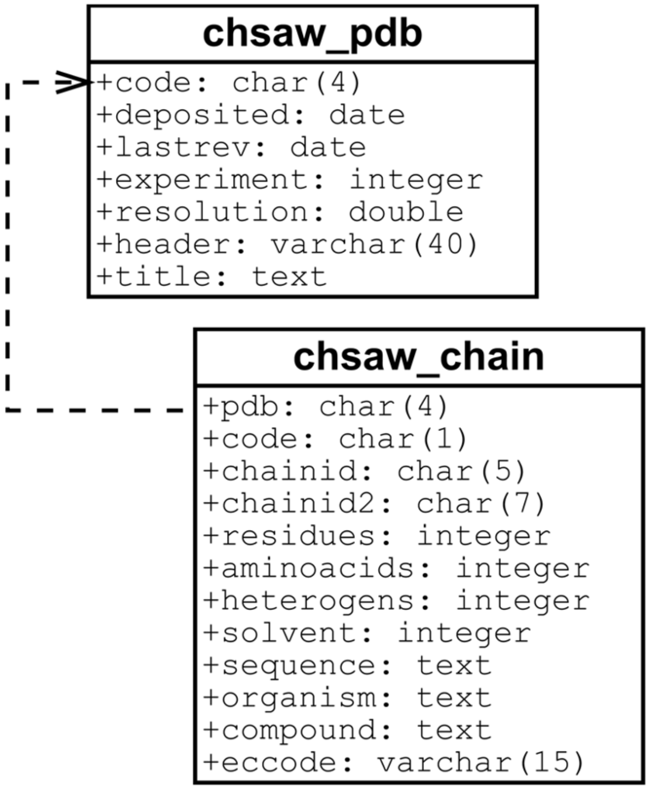

**Figure 2**
**PDBChainSaw database schema.** The database schema of the tables in PDBChainSaw models the one-to-many relation between structures and their chains. Entities in the table "chsaw_pdb" are uniquely identified with the four-letter PDB code. Entities in the table "chsaw_chain" are linked to their parent structure through the PDB code and add their single character chain identifier to their unique identifier field named "chainid". Since chain identifiers may appear to be of lower case or non-alphanumeric, another identifier "chainid2" was set to the numerical value of the chain identifier to allow for unique formatting in FASTA files.

base front end, as demonstrated with PDBChainSaw, to extract information encoded in PDB files, e.g. sequence from co-ordinates.

## Availability and Requirements

***Project name***
MolTalk

***Project homepage***
http://bioinformatics.org/moltalk

http://www.moltalk.org

***Operating system***
Linux and other Unix derivates, Windows and MacOSX

***Other requirements***
GCC 3.x [18], GNUstep [6], StepTalk [5]

***License***
GNU General Public License

***Any restrictions to use by non-academics***
Free of charge as long as GNU GPL is respected

## Authors' contributions

AVD designed and implemented MolTalk. HS evaluated PDBChainSaw and was mainly responsible for manuscript preparation.

## Acknowledgements

AVD was partially supported by GlaxoSmithKline. Both authors are grateful to the Swiss Institute of Bioinformatics.

## References

1. Stoesser G, Baker W, van den Broek A, Garcia-Pastor M, Kanz C, Kulikova T, Leinonen R, Lin Q, Lombard V, Lopez R, Mancuso R, Nardone F, Stoehr P, Tuli MA, Tzouvara K, Vaughan R: **The EMBL Nucleotide Sequence Database: major new developments.** *Nucleic Acids Res* 2003, **31:**17-22.
2. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M: **The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003.** *Nucleic Acids Res* 2003, **31:**365-70.
3. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The Protein Data Bank.** *Nucleic Acids Res* 2000, **28:**235-42.
4. **Objective-C FAQ** [http://www.faqs.org/faqs/computer-lang/Objective-C/faq]
5. **StepTalk – GNUstep scripting framework** [http://steptalk.agentfarms.net]
6. **GNUstep project** [http://www.gnustep.org]
7. Yip LY, Scheib H, Diemand AV, Gattiker A, Famiglietti LM, Gasteiger E, Bairoch A: **The Swiss-Prot Variant Page and the ModSNP Database: A Resource for Sequence and Structure information on Human Protein Variants.** *Hum Mutat* 2004, **23:**464-470.
8. **PostgreSQL relational database system** [http://www.postgresql.org]
9. **PDB file format** [http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html]
10. Koradi R, Billeter M, Wuthrich K: **MOLMOL: a program for display and analysis of macromolecular structures.** *J Mol Graph* 1996, **14:**51-5.
11. Guex N, Peitsch MC: **SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling.** *Electrophoresis* 1997, **18:**2714-23.
12. Sayle RA, Milner-White EJ: **RASMOL: biomolecular graphics for all.** *Trends Biochem Sci* 1995, **20:**374.
13. Diemand AV, Scheib H: **iMolTalk: an interactive, internet-based protein structure analyis server.** *Nucl Acids Res* in press.
14. Painter J, Merritt EA: **mmLib Python toolkit for manipulating annotated structural models of biological macromolecules.** *J Appl Crystal* 2004, **37:**174-178.
15. **Computational Crystallography Toolbox** [http://cctbx.sourceforge.net]
16. **CCP4 software library** [http://www.ccp4.ac.uk]
17. Humphrey W, Dalke A, Schulten K: **VMD: visual molecular dynamics.** *J Mol Graph* 1996, **14:**33-8.
18. **GNU compiler collection GCC** [http://gcc.gnu.org]