# A system for automated validation of GPCRs homology models against mutational data

**Stefan Mordalski[1*], Tomasz Kosciolek[1], Kurt Kristiansen[2], Ingebrigt Sylte[2], Zdzisław Chilmończyk[3], Andrzej J Bojarski[1]**

[1]Department of Medicinal Chemistry, Institute of Pharmacology Polish Academy of Sciences, 12 Smetna St, 31-343 Krakow, Poland
[2]Medical Pharmacology and Toxicology, Department of Medical Biology, Faculty of Health Science, University of Tromsø, N-9037 Tromsø, Norway
[3]National Medicines Institute, 30/34 Chełmska Street, 00-725 Warszawa
e-mail: stefanm@if-pan.krakow.pl

## Introduction

Homology modeling of G-Protein Coupled Receptors is a very challenging task. Little number of crystal templates and low sequence similarity (especially for non class A GPCRs) between target and template render the sequence alignment ambiguous in most cases. For the above reasons data acquired from site-directed mutagenesis become a vital aspect of homology modeling, as they can be used to evaluate created models.

In this study we present a tool for automated checking homology models against the mutational data collected within tinyGRAP (1) database. tinyGRAP database is querried for the investigated sequence and its close homologs (i.e. group members), and substitution mutations are retrieved. Query results are then checked whether appropriate residues face inside of the receptor (with some margin), and the tool produces report in PyMol .pse file pointing amino acids both violating and meeting the mutational „constrains".

## Database preparation

The first step in setting up the program was to prepare tinyGRAP database. Since the dataset is a collection of files containing mutational data, the apropriate parser was prepared to convert it into form suitable for RDBM engine. An SQLAlchemy python module was used to provide Object Relational Model (ORM) for the data, providing convenient mapping database table onto actual python class. Such approach allows picking from wide choice of database engines from SQLite to Oracle without any changes in the script. In this study the SQLite engine was selected due to its simplicity.

In addition, we plan to adapt the GRAP database in the same fashion as the above. Original GRAP database provides additional information about the changes in affinity caused by the mutation. It would be additional hint to assuming the mutation and model sensibility.

## Model verification workflow

Verification of the input model goes in several steps (Fig. 1). First, the chain sequences retrieved from the model are located within full SwissProt sequence of the protein and residue numbers from SwissProt record are assigned to coresponding ones in the model (**A**). In the next step, tinyGRAP database is querried against desired protein and resulting amino acids are localised within the model structure (**A**, residue in red).

For the residues found in the structure a geometric centre of the sidechain (**B**, green dot) is allocated and its distance from central axis (**B**, blue dot) of the protein is calculated. Analogous range is computed for the alpha carbon (**B**, red dot) of selected amino acid. Residue is then labeled as 'good' if sidechain centre is closer to the main axis than the alpha carbon, and 'bad' flag is turned on in the oposite.

Finally the report is produced (**C**). It is a PyMol session file containing input model along with the textual description of the evaluation (reidue number, it's verification status and authors and title of the publication containing the point mutation used.

## Conclusions and future plans

The program appears to be a very convenient tool for automating model checking against the mutational data. In rare peculiar cases it can make mistakes in flaging the amino acids, and thus visual inspection of the report is essential, but anyways the tool reduces the amount of work required for locating and checking the mutations for the given homology model.

In the nearest future we plan to use the original GRAP database, containing more information than the 'tiny' version, and provide both GUI and web service versions of the tool.

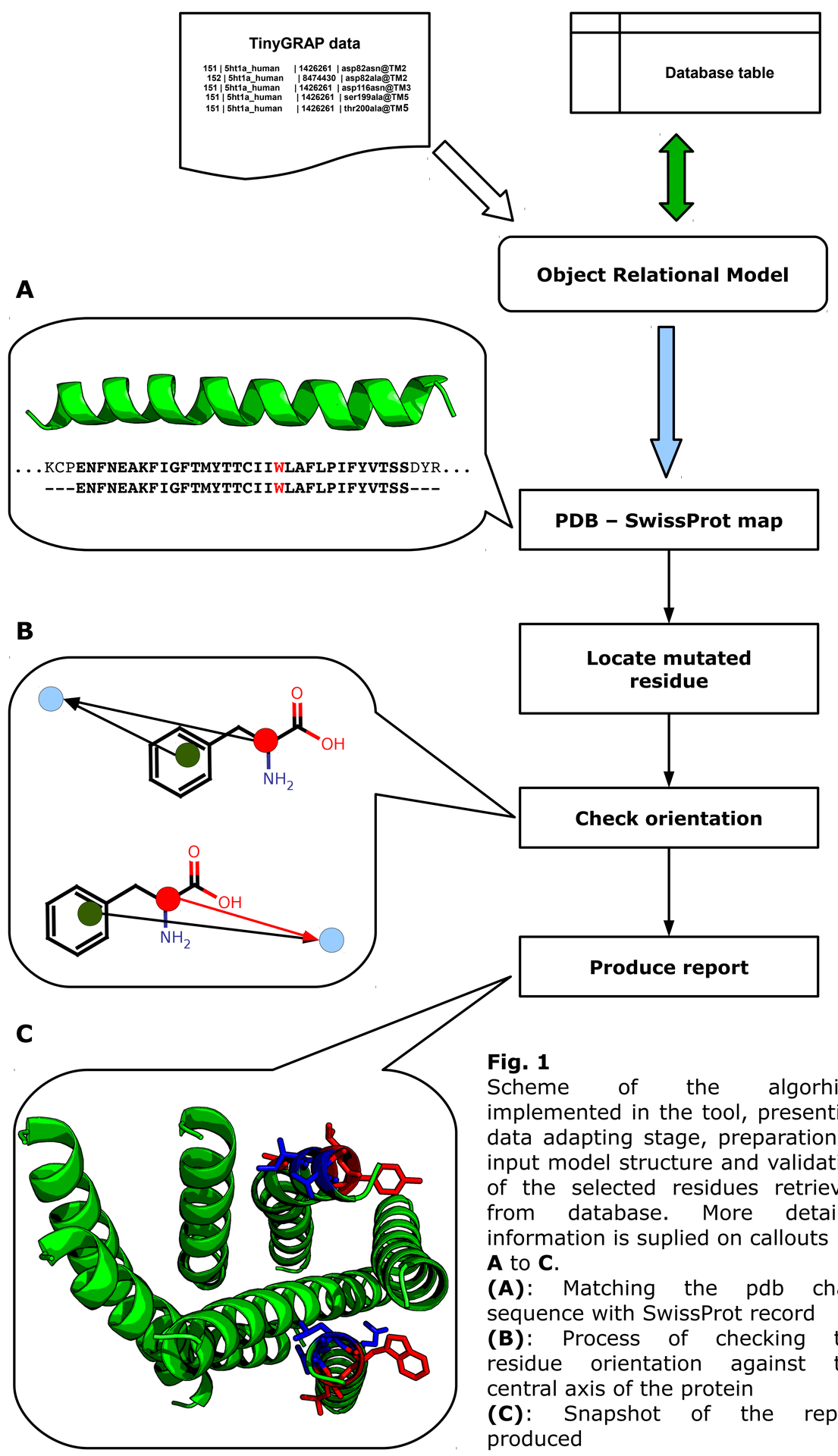


**Fig. 1**
Scheme of the algorhitm implemented in the tool, presenting data adapting stage, preparation of input model structure and validation of the selected residues retrieved from database. More detailed information is suplied on callouts **A** to **C**.
**(A)**: Matching the pdb chain sequence with SwissProt record
**(B)**: Process of checking the residue orientation against the central axis of the protein
**(C)**: Snapshot of the report produced

**Acknowledgments**

The study was partly supported by a grant PNRF-103-AI-1/07 from Norway through the Norwegian Financial Mechanism

**Literature**

1. Beukers MW, Kristiansen K, IJzerman AP, Edvardsen I, **TinyGRAP database: a bioinformatics tool to mine G protein-coupled receptor mutant data.**, Trends Pharmacol Sci. 1999 Dec;**20(12)**:475-7.

Additional python modules used: mmlibs, Bio, SQLAlchemy.
Visualizations were prepared with PyMol

Poster can be downloaded from www.cns-platform.eu