Application of highly efficient database systems in virtual screening protocol

Stefan Mordalski[(1)], **Tomasz Kościółek**[(1)], **Ingebrigt Sylte**[(2)], **Andrzej J. Bojarski**[(1)]

(1) Polish Academy of Sciences, Institute of Pharmacology, Krakow, Poland; stefanm@if-pan.krakow.pl
(2) Medicinal Pharmacology and Toxicology, Department of Medicinal Biology, Faculty of Health Science, University of Tromsø, N-9037 Tromsø, Norway

Introduction

Increasing power of modern multicore CPUs caused computational power to be relatively cheap nowadays and so processor time greed of Virtual Screening (VS) protocol is no longer an issue. However, the amount of output produced and size of compound databases used for VS brought up new challenges, such as ways of sharing large amounts of data and flexible and rapid results analysis. In this poster we present application of MySQL cluster database engine supporting results analysis and filtering.

Cluster database engine divides the data into smaller chunks and spreads them across the cluster nodes. This approach allows all the dataset to be held in RAM and so provides instant access to data and speeds up multitable queries by several magnitudes. In addition several applications were written for batch processing VS outputs and ease up visual inspection stage of virtual screening protocol.

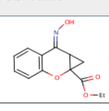
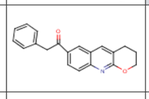
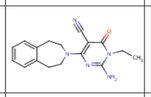
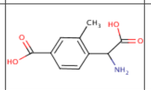
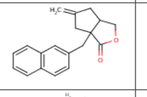
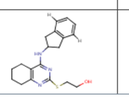
Application

There are several steps of our Virtual Screening Workflow[(1)] where usage of Database Engine may simplify and automate data collection and analysis. The docking results are parsed, and identificators of ligand and receptor along with docking score, ligand efficiency, etc of every complex are submitted into table. Data collected are then processed in various ways, i.e. number of docked ligands per receptor, best poses - any kind of statistics can be calculated with a single query.
A script supporting visual inspection stage allows fetching selected ligand-receptor poses and displaying them in PyMol.
Our efforts in improving virtual screening analysis resulted in another tool providing automated generating and storing of Structural Interaction Fingerprints (SIFts) – based on stored fingerprints for docking poses, a SIFt pattern, an averaged fingerprint describing binding-site can be created and visualized.

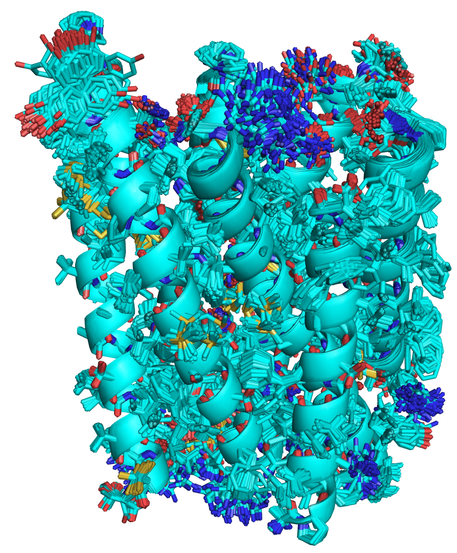
Summary

Implementing databases into various steps of Virtual Screening Workflow significantly improves their performance. In particular, data analyzing process, when using SQL, becomes indeed rapid and flexible process. Moreover, centralized data storing gives access to all interested users over the network, and allows simultaneous modification of data due to transaction system implemented into database engine.

A

CdId	INFO.Structure	INFO.MolWe...	INFO.Formula	ID	INFO.Traditi...	INFO.IUPAC...	INFO.Smiles	INFO.SALT_...	INFO.AVIAL...	INFO.DATE
3		247,25	C13H13NO4	1	ethyl (6E)-6-(hydroxyimino)-2-oxatricyclo[5.4.0.0^{3,5}]undeca-	ethyl (6E)-6-(hydroxyimino)-2-oxatricyclo[5.4.0.0^{3,5}]undeca-	CCOC(=O)C12CC1\C(=N/O)c1ccccc1O2			2008-03-28
4		303,35	C20H17NO2	2	2-phenyl-1-(2H,3H,4H-pyrano[2,3-b]quinolin-7-yl)ethan-1-one	2-phenyl-1-(2H,3H,4H-pyrano[2,3-b]quinolin-7-yl)ethanone	O=C(Cc1ccccc1)c1ccc2nc3OCCCc3cc2c1			2008-03-28
5		309,37	C17H19N5O	3	2-amino-1-ethyl-6-oxo-4-(2,3,4,5-tetrahydro-1H-3-benzazepin-	2-amino-1-ethyl-6-oxo-4-(2,3,4,5-tetrahydro-1H-3-benzazepin-	CCn1c(N)nc(N2CCc3ccccc3CC2)c(C#N)c1=O			2008-03-28
6		209,20	C10H11NO4	4	4-[amino(carboxy)methyl]-3-methylbenzoic acid	4-[amino(carboxy)methyl]-3-methylbenzoic acid	Cc1cc(ccc1C(N)C(O)=O)C(O)=O			2008-03-28
7		278,35	C19H18O2	5	5-methylidene-6a-(naphthalen-2-ylmethyl)-hexahydro-1H-cyclopenta[c]furan-	5-methylidene-6a-(naphthalen-2-ylmethyl)-hexahydro-1H-cyclopenta[c]furan-	C=C1CC2CO C(=O) C2(C1) Cc1ccc2ccccc2c1			2008-03-28
8		341,47	C19H23N3OS	6	2-({4-(2,3-dihydro-1H-inden-2-ylamino)-5,6,7,8-tetrahydroqu	2-({4-(2,3-dihydro-1H-inden-2-ylamino)-5,6,7,8-tetrahydroqu	OCCSc1nc2CCCCc2c(NC2Cc3ccccc3C2)n1			2008-03-28

...



B

Poses filtering and selection

SIFts generation and analysis

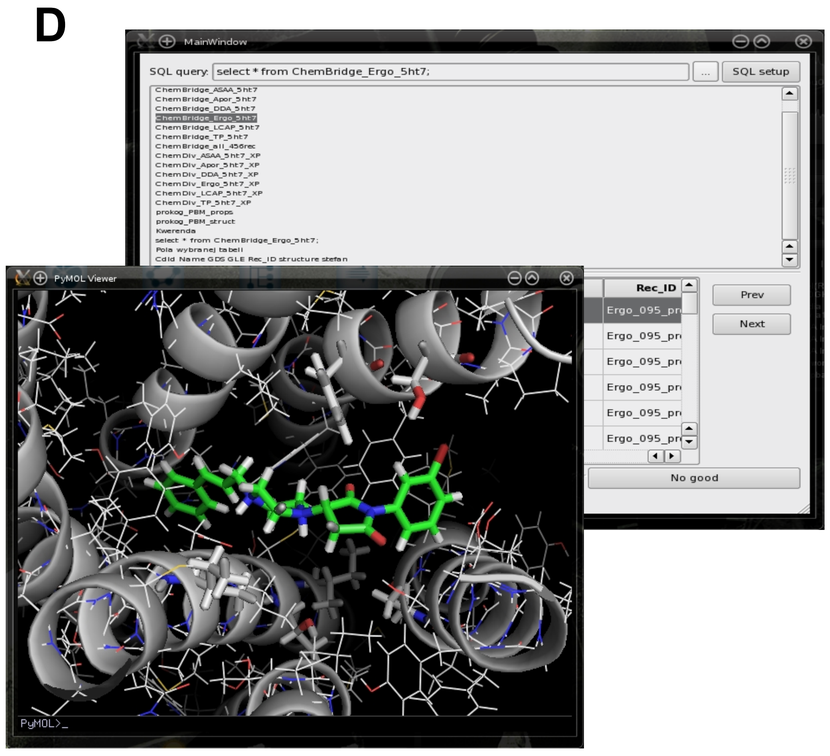
C

```
SELECT ligand_name, receptor_name,
count(ligand_name) FROM my_table GROUP BY
receptor_name ORDER BY count(ligand_name)
DESC;
```

```
+-------------+------------------+--------------------+
| ligand_name | receptor_name    | count(ligand_name) |
+-------------+------------------+--------------------+
| ligand_10   | Q63004.B99990048 |                 18 |
| ligand_19   | Q63004.B99990312 |                 18 |
| ligand_4    | Q63004.B99990199 |                 18 |
| ligand_19   | Q63004.B99990394 |                 17 |
| ligand_16   | Q63004.B99990011 |                 16 |
+-------------+------------------+--------------------+
```

...

D



E

```
...10111001111100000000000000000000000110000000...
...00000000010100000000000000000000000110000000...
...00000000011100000000000000000000000110000000...
...10111001110100000000000000000000000110000000...
...10111001111100000000000000000000000110000000...
```

F

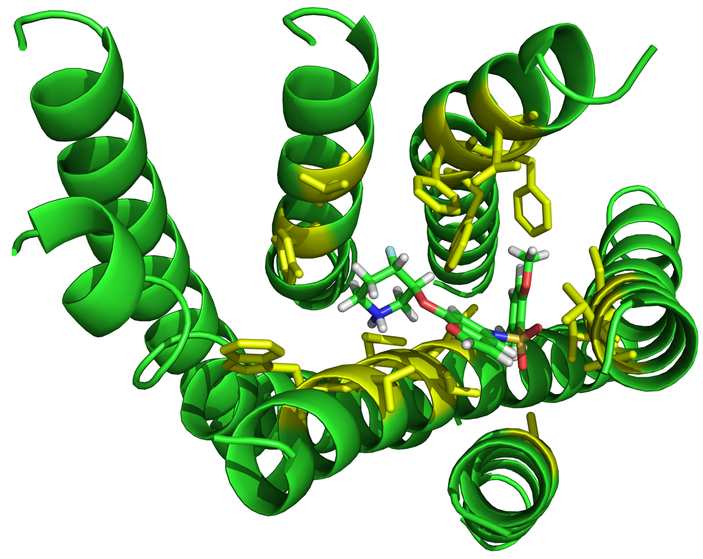


Fig. 1
Results of extensive docking of numerous compounds into several models/targets (**A**) are submitted into database server (**B**). Organisation of stored data allows easy data analysis. Example query (**C**) retrieves number of ligands successfully docked into each receptor. Returned dataset can be then fetched by a script and visualized in PyMol (**D**).
In the separate pathway ligand-receptor poses are used to generate SIFts (**E**). Fingerprints themselves are then analyzed and consensus binding site pattern is produced and visualized (**F**: pattern for all sidechain interactions coloured in yellow)

Acknowledgments

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

Literature

(1) Kurczab R, Nowak M, Chilmonczyk Z, Sylte I, Bojarski AJ *„The development and validation of a novel virtual screening cascade protocol to identify potential serotonin 5-HT(7)R antagonists.”,* Bioorg. Med. Chem. Lett., 2010, 20(8), 2465-8

Protein homology modeling was performed with modeller 9v7, docking and scoring were executed using Glide from Schrodinger Suite 2010. MySQL v5.1.41 was used as database server. Visualisation were prepared with Pymol v1.2 and self scripts.

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