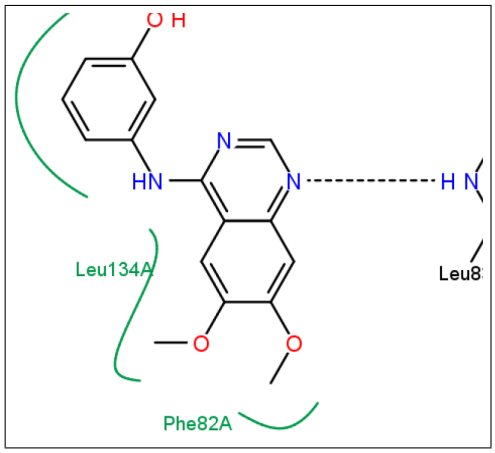
PoseView

2D Sketches of Protein-Ligand Complexes



User Reference
Version 1.1

Katrin Stierand
Marcus Gastreich

Development of *PoseView* has been initiated by Katrin Stierand and Matthias Rarey (ZBH, Univ. Hamburg, Germany). *PoseView* is now available solely through BioSolveIT, St. Augustin, Germany.

Phone ++49-2241-2525-0, support@biosolveit.de

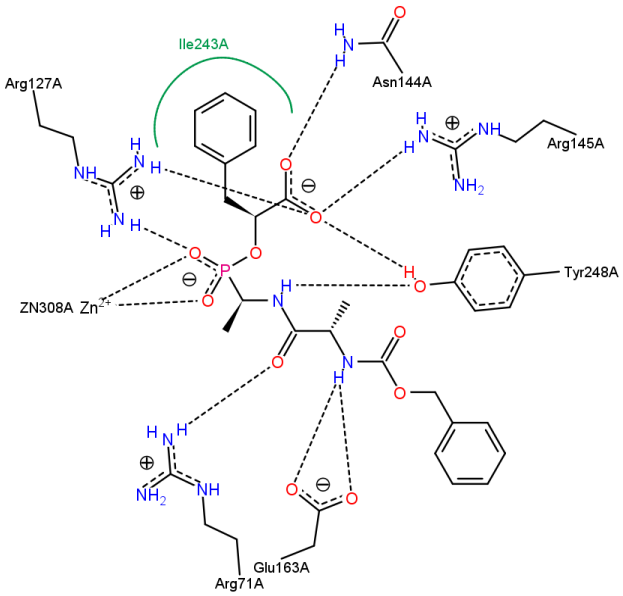
Contents

1 Introduction

1.1 About *PoseView*

PoseView is a computer program which generates 2D pictures of protein-ligand complexes for fast and intuitive perception of interactions; it automatically generates the layouts from given input files.
A diagram contains

- the ligand
  - dashed lines representing hydrogen bonds and metal interactions
- the corresponding residues of the protein
  - green residue labels for amino acids with hydrophobic contacts to the ligand
  - spline segments which highlight the hydrophobic contact areas of the ligand



For details about the implementation and a thorough coverage of *PoseView*'s concept, please see the original publication:

Stierand, K., Rarey, M. (2010)
Drawing the PDB:
Protein-Ligand Complexes in Two Dimensions.
Medicinal Chemistry Letters, Vol. 1, No. 9, pp. 540-545.
http://dx.doi.org/10.1021/ml100164p

Stierand, K., Rarey, M. (2007)
From Modeling to Medicinal Chemistry:
Automatic Generation of Two-Dimensional Complex Diagrams.
ChemMedChem, Vol. 2, No. 6, pp. 853-860.
http://www3.interscience.wiley.com/journal/114210232/abstract

Stierand, K., Maaß, P., Rarey, M. (2006)
Molecular Complexes at a Glance:
Automated Generation of Two-Dimensional Complex Diagrams.
Bioinformatics, Vol. 22, No. 22, pp. 1710-1716.
http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/14/1710
Contact: poseview@biosolveit.de

1.2 Known Bugs and Limitations

1.2.1 Operating System Dependent Issues

- Linux
  PoseView requires the glibc version 2.3 or later. Please consult systems administration if this error (or alike) occurs:

```
/lib/ld-linux.so.2: version 'GLIBC_2.3' not found (required by poseview)
/lib/i686/libpthread.so.0: version 'GLIBC_2.3.2' not found (required by poseview)
/lib/i686/libc.so.6: version 'GLIBC_2.3' not found (required by poseview)
```

1.2.2 Known Bugs

- With this version, interactions of planar H-bond donors or acceptors may not always be fully complete. We are working to resolve this issue with one of the next versions.
- FlexX Interaction Model deactivated. This model is under heavy development and therefore currently deactivated.
- Splash Screen / Help Screens: There are several minor inconsistencies in the splash and help screens which are due to continuing massive development of the associated engines underneath.

1.3 License Scheme

Our software is license key protected. Please be aware that you cannot run computations under any circumstances without a valid license. *PoseView* checks out a license as soon as

an actual computation is evoked.

Here are the necessary steps to get it all running:

1. Determine your BioSolveIT-HostID and request a license:

   To obtain a valid license you must determine and send us the host or system ID of your computer:
   `% poseview -i`
   Copy the `BIOSOLVEIT=...` entry.
   Please send the copied BioSolveIT-HostID number to `license@biosolveit.de`, and we will return a license file asap.

   Each line of a license file you will obtain includes the name of the licensed tool or module, and — among other information — the version number and the expiration date of the license.

   `Example`

   ```
   INCREMENT PoseView BIOSOLVE 2.0 28-jun-2015 uncounted \
       HOSTID=BIOSOLVEIT=0BS058665F5FCFFA74EAE1458495A8F65C6 \
       SIGN="0078 AD79 1445 A900 906E 00A4 AB51 F600 AFC8 \
       4394 9356 3C87 EBD2 4A7B E88D"
   ```

2. After you have received your license keys from us, you will need to tell the software where the license is actually located. This depends on whether the license you have requires a license server (we use flexlm) set up or not.

   Typically, purchased licenses require a license server, test and evaluation licenses do not. If in doubt please just ask us. Setting up a license server is described in Section 1.3.1 on page 8.

   Set an environment variable `BIOSOLVE_LICENSE_FILE`
   No license server: pointing to the directory of the license file(s)
   With license server: pointing to `@mylicenseserver` in which `mylicenseserver` is the name/IP of the computer which has the license server installed. The at-sign is indeed needed.

   If you have to define a port, enter the port number before the "@", like this:

   `BIOSOLVE_LICENSE_FILE=portnumber@mylicenseserver`

   - Windows:

     `Control Panel -> System -> Advanced -> Environment Variables`

   - Linux: Use the `setenv` or `export` mechanisms of your shell.

3. A successfully found license can be checked at startup where licenses found will be uttered similar to the very end of this start-up stream:

```
_____________________________________________________________________________
                        ___              __   ___
                       | _ \___ ___ __\ \ / (_)_____ __ __
                       |  _/ _ (_-</ -_) V /| / -_) V  V /
                       |_| \___/__/\___|\_/ |_\___|\_/\_/

ZBH - Center for       Automatic generation of 2D complex diagrams
Bioinformatics
University of Hamburg  Version:   1.1.2       (10.11.11)
Bundesstrasse 43
20146 Hamburg
Germany                Authors:   Katrin Stierand, Matthias Rarey

BioSolveIT GmbH        Copyright: ZBH, University of Hamburg, Germany
An der Ziegelei 79                BioSolveIT GmbH, Sankt Augustin, Germany
53757 St. Augustin     Contact:   poseview@zbh.uni-hamburg.de
Germany                           www.zbh.uni-hamburg.de/poseview
_____________________________________________________________________________

For information about additional contributors and copyright notes
please consult the user guide or type 'help about'.

>> Running on theta (Linuxx86_64 2.6.25.20-0.7-default) with 4 processors.
>> PoseView license check (BioSolveIT keys): succeeded.
>> Licensed modules: PoseView
```

Further details are described below or may be obtained from the BioSolveIT knowledge base page at `http://www.biosolveit.de/faq`.

1.3.1 Running a FlexLM License Server

Instead of specifying a license file directly, it is also possible to serve your licenses from a license server using the FlexLM license management system. Your FlexLM administrator must add the respective license file to the directory from which FlexLM takes its licenses and add the BIOSOLVE vendor daemon to the directory where the `lmgrd` (license manager daemon) resides. A current version of the license manager daemon and the BIOSOLVE vendor daemon can be downloaded from `http://www.biosolveit.de/download?product=bsit_flexlm`. At the console, use the command

```
lmgrd -c <path_to_licensefiles> -l <logfile>
```

to start the license server.

Using FlexLM, it is also possible to use *floating licenses*, i.e., licenses that are hosted by a server and distributed to *PoseView* clients on demand. Floating licenses have a special layout. A floating license is always locked to the name or the IP address of the server and its BioSolveIT-HostID. Server-bound license files will look similar to the example below.

Example

```
SERVER myserver BIOSOLVEIT=0BS058665F5FCFFA74EAE1458495A8F65C6
USE_SERVER
VENDOR BIOSOLVE
INCREMENT PoseView BIOSOLVE 2.0 28-jun-2015 100 SIGN="00C6 1440 7772 \
        E4A8 116C FFFB EDC8 F400 B648 5413 6ECA 8852 4A2E 29B8 E5D6"
```

In this example a site has 100 *PoseView* licenses. Every time a new instance of *PoseView* is launched the server transfers the license to the application. When the computation of *PoseView* finishes or *PoseView* is closed, the license is returned to the server.
Please also refer to our faq page `http://www.biosolveit.de/faq/questions/5` on how to run a FlexLM license server.

2

Commands and Options

2.1 Calling Help

Calling *PoseView* with the flag `-h` starts the splash screen with a help text:

```
                          ___              __   ___
                         | _ \___ ___ __\ \ / (_)_____ __ __
                         |  _/ _ (_-</ -_) V /| / -_) V  V /
                         |_| \___/__/\___|\_/ |_\___|\_/\_/

 ZBH - Center for        Automatic generation of 2D complex diagrams
 Bioinformatics
 University of Hamburg   Version:   1.1.2       (10.11.11)
 Bundesstrasse 43
 20146 Hamburg
 Germany                 Authors:   Katrin Stierand, Matthias Rarey

 BioSolveIT GmbH         Copyright: ZBH, University of Hamburg, Germany
 An der Ziegelei 79                 BioSolveIT GmbH, Sankt Augustin, Germany
 53757 St. Augustin      Contact:   poseview@zbh.uni-hamburg.de
 Germany                            www.zbh.uni-hamburg.de/poseview

 For information about additional contributors and copyright notes
 please consult the user guide or type 'help about'.

>>> SYNOPSIS:
    poseview [OPTIONS]

>>> OPTIONS:
    -c <configfile> : Configuration file (xml format)
    -f <filename>   : Read input data from complex file (.mol2)
    -h              : Printing of this help text
    -i              : Generate processor id for this machine
    -l <ligfile>    : The ligand is read from <ligfile>.
                      Default format is .mol2
    -m <file>       : List of ligand and protein file names.
                      Format: lp <ligfilename> <protfilename>
    -o <outfile>    : Prints the diagram to <outfile>
                      possible file formats: .png, .pdf, .ps, .fig
    -p <protfile>   : The protein is read from <protfile>.
                      Default format is .rdf
    -s <w> <h>      : Size of the .png or .fig output file.
                      png: 100 <= w|h <= 1000, fig 1000 <= w|h <= 50000
```

```
-t <text>       : Info text printed in output file.
-v              : Prints the version number and compilation
                  information.
-w <filename>   : Print complex data to file "filename.mol2"
                  Requires -l and -p or -f
```

Most of the options will be self-explanatory. More detailed option descriptions will follow further below.

2.2 Options and command line arguments

2.2.1 `-f`: Complex files reading

Please see Section 2.3.1 on p.15.

2.2.2 `-i`: License ID (BioSolveIT-HostID) Generation

To be able to send you a license file for *PoseView*, we need to know the so-called *BioSolveIT-HostID*. This ID will be reported upon calling *PoseView* with the `-i` flag. Please see the section on installation and licensing to obtain further details on this.

2.2.3 `-l`: Ligand Input

This reads ligands in mol2 and sdf format. The suffix decides. "SD" or "sd" are not accepted as valid sdf format extensions. The extensions are not case sensitive (whereas the filename is, under Linux). Please see p. 14 for the full usage description.

2.2.4 `-m`: List file processing

This option gives you the possibility of processing multiple protein-ligand complexes at the same time. It works as follows:
The argument to `-m` is an ASCII file one or more lines forming an ASCII "table". In every line, entries are separated by a blank or a tab. The first "column" of the file has a contents identifier, being one out of `p`(rotein), `l`(igand), or `f`(ile). The following column contents is then defined by the preceding identifier. Here's an example:
Contents of the file `mycomplexes.txt` (extension does not matter):

```
pl   myproteinA.pdb   myligandA.mol2
f    complexfile.mol2
```

The file `complexfile.mol2` carries the contents of the entire complex in terms of a mol2 file with a special interaction section (cp. Sec 2.3.1).

2.2.5 `-o`: Outputting images

The default output is the presentation in the integrated GUI browser. The GUI is self-explanatory and gives you plenty of choices to activate or de-activate certain labels, energy values etc.
Possible output file formats are:

- png
- pdf
- ps
- svg
- fig (Xfig program format)

Using Xfig you can change the full appearance of *PoseView* images. Within Xfig, you can change fonts, colors etc.

```
poseview -l <ligand> -p <protein> -o <output> [-t <caption>]
```

The caption is optional and defaults to the complex name.
The output format depends on the suffix of the given filename (.png, .pdf, .ps, .svg, .fig). Other suffixes will be rejected.
To avoid loading a ligand and a protein more than once it is possible to write a complex file as follows:

```
poseview -l <ligand> -p <protein> -w <complexfile>
```

2.2.6 `-p`: Protein file input

Reading proteins can proceed in three ways:

- Straight through a pdb:
  Using the distance/angle based interaction model, this path parses the pdb file and optimizes the hydrogen positions using a fast and new, internal algorithm.
- Using a LeadIT project file (extension `.fxx`)
- *(deprecated)* Via an old FlexX receptor definition file (`.rdf`)

2.2.7 `-s`: Size of the images

With this option, you specify the size of the image to be created in points. For pdf, this defaults to 300x300 pixels. `w|h` stands for width or height. You may specify both, separated by blanks.

2.2.8 `-t`: Text of your choice in output file

Using this option with a string enclosed by quotation marks (") you can add that very string to the end of your output files.

2.2.9 `-v`: Version information

At times, it is useful for us to know the *exact* version with which you work. The output generated with this option is a verbose block of information which we might request from you during support.

2.2.10 `-w`: Complex file write-out

Please see Section 2.3.1 on p.15.

2.3 Basic Usage, Examples

A protein-ligand complex often comes with a separate ligand file (often in SDF format), and a protein file (usually in PDB format). *PoseView* runs in two different modes which is connected to how *PoseView* determines the interactions (cp. the Sec. 3.1 on p. 17).
N.B.:
At this point it is not possible to use a PDB file which contains both the protein and a co-crystallized ligand in the same file and display them in *PoseView*.

2.3.1 Separate files vs. complex Files

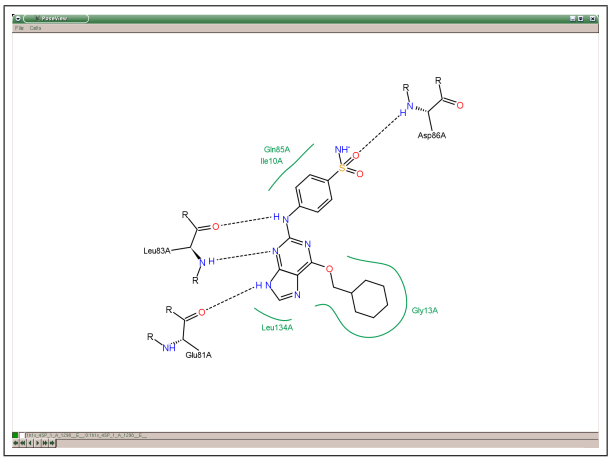
Mode 1: Using separate ligand and protein files

You have two input files, one for the ligand, one for the protein. Here's the quick overview of allowed file formats and some algorithmic details:

- Ligand input file formats: SDF or MOL2
- Receptor input file formats: PDB, MOL2, RDF or FXX (LeadIT project file format)
- Interactions between ligand and protein are estimated based on geometric criteria
- Interacting residues are cut from the receptor by *PoseView*

poseview `-l ligand_file -p protein_file`

A respective call would accordingly look like this:
poseview `-l examples/1h1s.sdf -p examples/1H1S.pdb`



File formats are detected by the file name suffixes (case insensitive). If no suffix is given, ligand file names are extended by `.mol2` and protein file names by `.rdf`.

Mode 2: A Complex file; the `-f` command line option

A complex file is a multi mol2 format containing all information for one complex

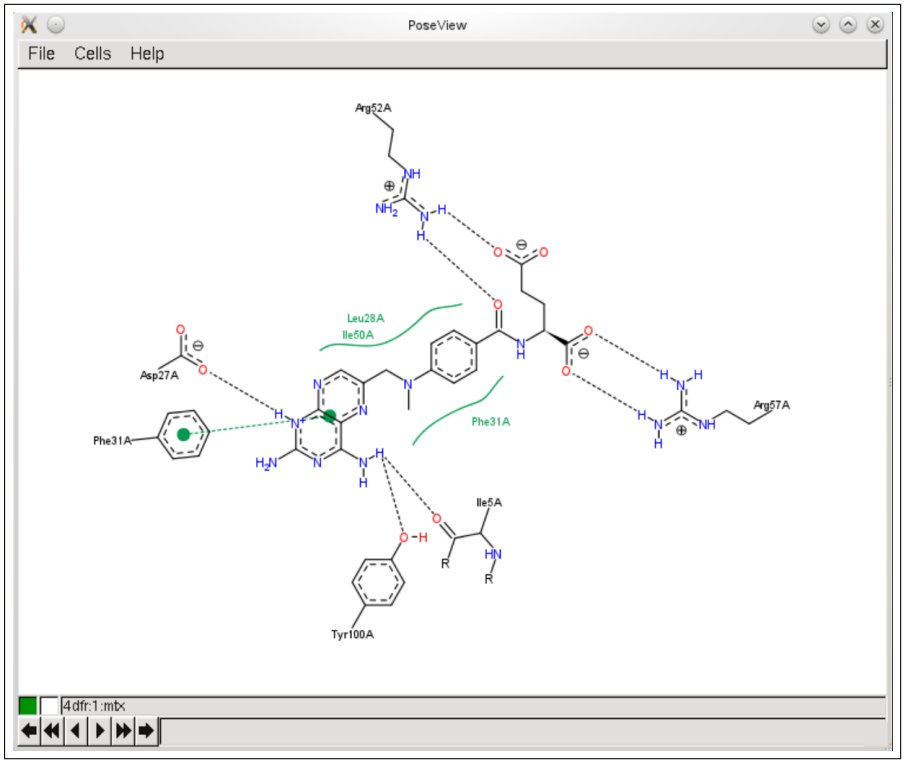
- A comment block contains all needed interaction information (see example below)
- The first molecule has to be the ligand
- The interacting (hydrogen bonds and metal interactions) residues are listed below

poseview `-f multi_mol2_file`

The respective information which *PoseView* needs is written in a special comment section as this header of a file below:

```
@<TRIPOS>COMMENT
ACETYLCHOLINESTERASE (E.C.3.1.1.7) COMPLEXED WITH TACRINE
%AMINO_ACIDS 1
# Format: <mol_id> <mol_nr> <chain_id> <name> <m_id>
2 437 * His440  440
%HYDROGEN_BONDS 1
# Format: <lig_ia_atm> <lig_ia_center> <aa_nr> <aa_ia_atm> <aa_ia_center> <type> <energy>
7 20 437 1 1 0 -4.002900
%HYDROPHOBIC_CONTACTS 2
# Format: <aa_name> <aa_nr> <aa_pdb_nr> <aa_chain_id><nof_cons> <con1> <con2> ... <conn>
PHE 327 330 * 6 1 2 3 4 5 6
TRP 81 84 * 8 3 4 5 6 11 12 13 14
```

This mode is mainly used in environments where another BioSolveIT tool or a corporate/third party tool fills the needed information into a multi-mol2 file. Here is an example for a call:
poseview `-f example/mol/4dfr.mol2`


PoseView
File Cells Help
Arg52A
Leu28A
Ile50A
Asp27A
Phe31A
Phe31A
Arg57A
Ile5A
Tyr100A
4dfr:1:mtx

3

Chemistry

3.1 Interaction Models

From the user perspective, *PoseView* has two tasks:

1. determine where interactions are
2. draw the found interactions in 2D

Since there is no straightforward way of a unique definition of interactions, *PoseView* has two modes to determine interactions:

A) Distance & Angle Based: (vdW = van der Waals)
This model uses the following default distance and angle thresholds to determine interactions. Please see Section 3.2 below how to customize these thresholds.

- **hydrogen bonds:** There are two criteria which have to be met: The centers of heavy atoms (donor and acceptor) are closer than 2.4Å and the angle between bond to the donor and the interaction is not smaller than 120°.
- **hydrophobic interactions:** They are drawn if the distance of the centers of a hydrophobic ligand atom and a hydrophobic protein atom is smaller than the sum of vdW-radius(ligand atom), vdW-radius(protein atom), and 0.8Å.
- **metal interactions:** These interactions take metal coordination into account. An interaction is formed if the center of a metal accepting ligand atom is closer than 0.8Å to a coordination surface of a metal ion, i.e., the ligand atom lies not more than 0.8Å away from the possible ligand positions.
- π **interactions:** *PoseView* draws π-π and π-cation interactions. A π-cation interaction is present if the center of an aromatic ring is closer than 4.5Å to a carbon cation. π-π interactions are shown if the centers of two aromatic rings of ligand and protein are closer than 5.0Å.

B) Using your generic description in a so-called *complex-file*:
Using a special section in a preprocessed mol2 file, it is possible to pipe information about where interactions should be drawn to *PoseView* directly. This requires the `-f` option at startup time. There is a commented example of a *complex file* in the `example` directory for your convenience. Typically this would be used if you would like to have your own interaction model interfaced to *PoseView*.

3.2 Parameter Customization

For customization of interaction parameters you can edit the configuration file `settings.pxx`, which resides in the *PoseView* directory. The file format is *xml*. *PoseView* will read the configuration file upon start.
Adjustable parameters are as follows:

HBOND_ATOM_CENTER_DISTANCE The distance below (!) which a H-bond is considered to exist; (cp. below for further constraints on H-bonds) *default:* 2.4Å

HBOND_CONE_ANGLE The angle of a cone within which H-bonds can be formed. *default:* 120.0°

HYDROPHOBIC_ATOM_DISTANCE An additional tolerance distance within which a hydrophobic interaction between two atoms is still considered existent. This value is added to the sum of vdW-radii. *default:* 0.8Å

METAL_TOLERANCE_DISTANCE An additional tolerance distance within which a metal interaction is still considered existent; the distance is computed from the coordination surface of a metal to the center of the respective ligand atom. *default:* 0.8Å

PI_PI_CENTER_DISTANCE The maximum distance between two aromatic rings that form π-*pi*-interactions. *default:* 5.0Å

PI_CATION_CENTER_DISTANCE The maximum distance for an interaction between a π-ring-system and a cation. *default:* 4.5Å

Here is a sample `settings.pxx` file:

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<biosolveit>
  <version major="1" minor="6" patch="0"/>
  <flexdocking>
    <settings>
      <parameters>
        <HBOND_ATOM_CENTER_DISTANCE  value="2.4"   />
        <HBOND_CONE_ANGLE            value="120.0" />
        <HYDROPHOBIC_ATOM_DISTANCE   value="0.8"   />
        <METAL_TOLERANCE_DISTANCE    value="0.8"   />
        <PI_PI_CENTER_DISTANCE       value="5.0"   />
        <PI_CATION_CENTER_DISTANCE   value="4.5"   />
      </parameters>
    </settings>
  </flexdocking>
</biosolveit>
```

4

Availability, Technical Remarks

4.1 Operating Systems

PoseView is available for Linux and Windows. Executables for other operating systems may be made available on request.

The tool's web page is at: `http://www.biosolveit.de/poseview`

Index