VASCo 1.0
User Guide

Acknowledgments

Disclaimer

VASCo: Copyright © 2008, Institute of Molecular Biosciences, Karl-Franzens University Graz (IMB-KFUG). All rights reserved. This software is provided "AS IS". IMB-KFUG makes no warranties, express or implied, including no representation or warranty with respect to the performance of the software and derivatives or their safety, effectiveness, or commercial viability. IMB-KFUG does not warrant the merchantability or fitness of the software and derivatives for any particular purpose, or that they may be exploited without infringing the copyrights, patent rights or property rights of others. This software program may not be sold, leased, transferred, exported or otherwise disclaimed to anyone, in whole or in part, without the prior written consent of IMB-KFUG.

Credits

The VASCo software was written by Georg Steinkellner with scientific advice from Karl Gruber and Christoph Kratky

This manual was written by Georg Steinkellner and Daniel Friedl.

We would also like to thank Michel F. Sanner (MSMS) and Raquel Norel (DelPhi) for allowing us to integrate and distribute there programs along with VASCo.

Manual Version: 3.2009 01 09

Karl-Franzens University Graz

Structural Biology
Institute of Molecular Biosciences
Humboldtstraße 50
8010 Graz
Austria

Phone: +43 (316) 380-1989
Fax: +43 (316) 380-9897
URL: http://strubi.uni-graz.at/

Graz University of Technology

Institute for Genomics and Bioinformatics
Petersgasse 14
8010 Graz
Austria

Phone: +43 (316) 873-5331
Fax: +43 (316) 873-5340
URL: http://genome.tugraz.at

Table of Contents

Introduction

VASCo is a program pipeline for the calculation of protein surface properties and the visualization of annotated surfaces. Special emphasis is laid on protein-protein interactions, which are calculated based on surface point distances. Molecular properties such as electrostatic potential or hydrophobicity are mapped onto these surface points. Molecular surfaces and the corresponding properties are calculated using existing well established programs integrated into the package, as well as custom developed programs. The modular pipeline can easily be extended to include new properties for annotation. The output of the pipeline is most conveniently displayed in PyMOL using a custom-made plug-in.

Installation

The program is mainly written in Python. The modules include also programs and third party software precompiled for different platforms. The software should run on unix based platforms as well as on most windows environments. Please uninstall all previous versions of VASCo-modules.

Windows

The following steps will guide you through the installation process of the windows distribution of VASCo – PPIX – Modules.

Package content

VASCo-Modules-x.win32.exe	Vasco python modules
VASCo.py	main program (command line)
install.pdf	this file
ppixplugin_vx.py	visualization plug-in for PyMOL

(x stands for the version number)

Requirements

1. Python programming package version higher than 2.4.0. (available at www.python.org)
2. PyMOL protein viewer. (available at http://pymol.sourceforge.net)

Setup

This will install the modules into the python site-packages and the visualization plug-in into the PyMOL program

1. Download and install Python version > 2.4.0
2. Download and install PyMOL
3. Execute ***VASCo-Modules-x.win32.exe***
4. Select the Python distribution where you want to install the modules and follow the on screen instructions
5. Run PyMOL and select ***"Plugin" -> "Install Plugin"*** at the drop down menu and select the ***ppixplugin_vx.py***. Restart PyMOL

Linux

The following steps will guide you through the installation process of the unix distribution of VASCo – PPIX – Modules.

Package content

VASCo-Modules-x.zip	Vasco python modules
VASCo.py	main program (command line)
install.pdf	this file
ppixplugin_vx.py	visualization plug-in for PyMOL

(x stands for the version number)

Requirements

1. Python programming package version higher than 2.4.0. (available at www.python.org)
2. PyMOL protein viewer. (available at http://pymol.sourceforge.net)

Setup

This will install the modules into the python site-packages and the visualization plug-in into the PyMOL program

1. Download and install Python version > 2.4.0
2. Download and install PyMOL
3. Unzip ***VASCo-Modules-x.zip***
4. within the unzipped directory ***VASCo-Modules-x*** type

```
python  setup_vasco_x.py  install
```

5. Run PyMOL and select ***"Plugin" -> "Install Plugin"*** at the drop down menu and select the ***ppixplugin_vx.py*** and restart PyMOL. Macintosh users have to follow a different procedure[1]

[1] On Macintosh PyMOL has to be run with the X11/Hybrid mode to install external plug-ins. (http://pymol.org/plugins.html) MacPyMOL for Tiger includes a hybrid X11 mode. Assuming that X11 is already installed, simply duplicate MacPyMOL.app and rename it to "PyMOLX11Hybrid.app". For further information see the PyMOL Wiki Forum http://www.pymolwiki.org/index.php/MAC_Install

Installation remarks:

You need write permission for your Python installation path. If you do not have write permissions please contact your administrator. If you still have permission problems you can also copy the folder ***"ppix_modules"*** within the unzipped VASCo-Modules-x to your working directory but the VASConvert program has to be in the same directory as the ppix_modules folder and the modules are not accessible from other python programs via the import command. (This will also work for windows platforms using the linux “ppix_modules” folder in VASCo-Modules-x.zip, if you encounter any installation problems with the windows installation executable) The script runs different third party programs. Therefore at linux platforms the “PATH” variable has to be extended to "./" (current path) within your .cshrc file (or other config file). e.g. set PATH = ('./' $PATH).

Short user guide

The program VASConvert creates folders and files within your current working directory. The minimum input is a PDB file and a file with standard run parameters (***input.ppix***) located within your working directory which will be created automatically at the first run of the program. The file can be used to set standard parameters which can be overruled by the command line parameters.

Surface Property Calculation

At unix platforms an alias in the ***.cshrc*** file (or any other configuration file) can be set like:

```
alias vasco python <path>/VASConvert-x.py
```

The program can be run within your working directory by typing the created alias with the command line parameters.

```
vasco –in_dir ./ -filename <name>
```

Otherwise the program VASConvert-x.py has to be located in the working directory.

```
python VASConvert-x.py –in_dir ./ -filename <name>
```

where name is the code of the PDB file or the filename (without extension!) of the PDB file which has to be located in the path specified with the ***–in_dir*** parameter. If no ***–in_dir*** parameter is set the file has to be located at the folder ***"<working_dir>/input"***.

Eample filename input:

```
<working_dir>/input/
                         pdb177L.pdb
                         CODE.pdb
                         test.ent
                         something.pdb
```

to run the program for these input files the ***–filename*** parameter has to be set as follows:

```
-filename 177L
-filename CODE
-filename test
-filename something
```

If there are similar filenames like ***pdb177L.pdb*** and ***177L.pdb*** or ***177L.ent*** in your input directory, only the first one which appears in the directory will be used as input file.

The program will create several folders and files during calculation.
The standard files and paths are:

```
<working_dir>/
            output/
                 __analyse/
                      all.log         ........ logfile for all runs
                      all_lock.txt    ........ lock all.log
                 <name>/              ........ dir created for each filename
                      conv_out/       ........ conversion in/out files
                           <name>-<unit>_atoms.csv
                           <name>-<unit>_surface.csv
                           <name>-<unit>_surface.pdb
                           <name>-<unit>.xyzn
                           <name>-<chain>.pdb
                           <name>_surfaceunit.csv
                      delphi_out/     ........ DelPhi and in/out files
                           <name>-<unit>_delphi.csv
                           <name>-<unit>.frc
                           <name>-<unit>.modpdb
                           <name>-<unit>.polH
                           <name>-<unit>.prm
                           <name>-<unit>_surface.pdb
                           <name>-<chain>.polH
                      hyd_out/        ........ HydroCalc in/out files
                           <name>-<unit>_hyd.csv
                      logfile/        ........ log files
                           <name>-<unit>_delphi.log
                           <name>_patch.log
                           PPIX-Convert.log
                      msms_out/       ........ MSMS in/out files
                           <name>-<unit>.area
                           <name>-<unit>.face
                           <name>-<unit>.log
                           <name>-<unit>.vert
                      patch_out/      ........ PatchCalc in/out files
                           <name>-<unit>_patch.csv
                      ppixdb_out/     ........ surface file with properties
                           <name>_db.ppix.gz . file for visualization using
                                                 the PyMOL Plug-in
```

The conversion and calculation output files can be investigated or deleted after the run. An additional run with the same filename will overwrite all created files in the directory. The final file which is used for viewing within PyMOL using the VASCo PyMOL surface viewer Plug-in is the compressed file ***<name>_db.ppix.gz*** located in the ***ppixdb_out*** directory of each filename directory. This file can be read in as it is by loading the file with the ***"Load file"*** menu within the VASCo PyMOL plug-in window.

Use the -h parameter to get additional input information for the VASCo Module.

python VASCo.py -h

```
Usage:
______
VASConvert-1.0.py -arg1  value1  -arg2  value2 -argX   valueX
arguments:
__________________________________________________________________________________________
-h                                        :this help screen
-opt_file          <ppix option file>     :specify filename of PPIX options input file
                                            if no file exists it will be written,change it
                                            for your own settings and rerun program
-in_dir            <input directory>      :Directory of input pdb files
                                            (pdbCODE.ent   files)
-out_dir           <output directory>     :main dir where output directories
                                            will be created
-filename          <code>                 :PDB code to proceed
                   <code;code;code>       :proceed some PDB CODES (4req;1crw;177l)
                   <PDB_FILE:filename>    :read in PDB-codes from given file
                                            (PDB - codes in first column)
-testrun                                  :runs program with default test directories
                                            and inputfiles
__________________________________________________________________________________________
<variable>        <setting>               :You can set and change every variable if
                                            you want here for available variables and
                                            settings type -hv
-hv                                       :Help Screen for additional Variablesand Settings
```

The standard parameters specified in the file ***input.ppix*** , which will be created automatically at the first run of the program, can be changed and additionally overruled by command line input variables.

For a test run, the ***–testrun*** parameter can be set:

python VASCo.py –testrun

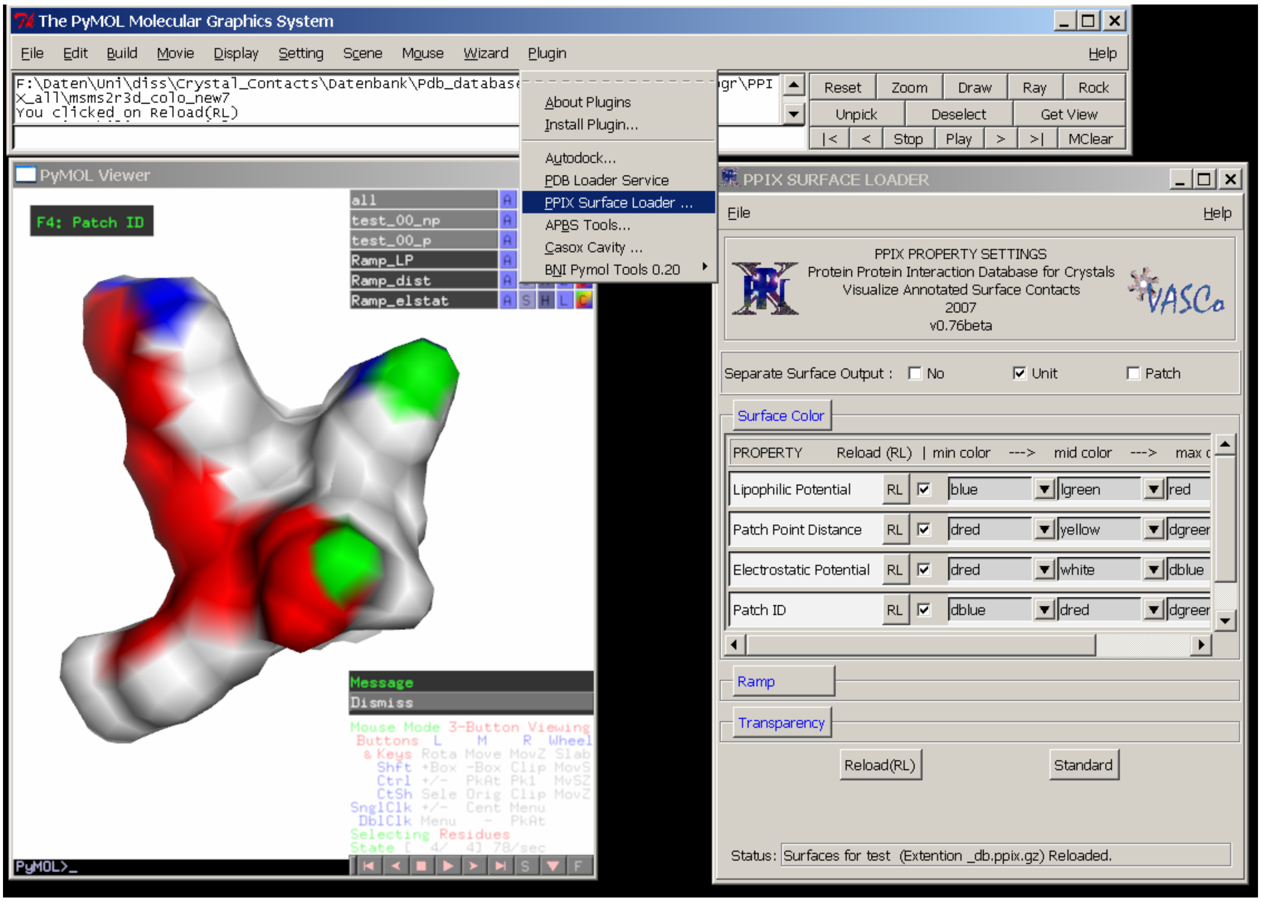
A ./***test_out*** directory will be created with all output directories of a normal run. The ***test_db.ppix.gz*** file located at the ./***test_out/test/ppixdb_out/*** directory can be read into PyMOL using the provided PyMOL PPIX surface loader Plug-in.

Visualization in PyMOL

After the installation of the VASCo PyMOL Plug in (***ppixplugin_vx.py***) the surface loader can be accessed by the PyMOL drop down menu ***Plugin ---> PPIX Surface Loader***. To load the output of the VASCo program (***<name>_db.ppix.gz***) located at ***<name>/ppixdb_out/*** after a VASCo run, use ***File --> Open file*** in the PPIX Surface Loader Window. and select the ***test_db.ppix.gz*** file or any other VASCo created surface file. Depending on the size of the surface file and the information provided (and the system specifications) this may take some time.

After loading the file you can select the surface properties of interest and set colour and ramp properties. You can load each property separately by clicking the ***(RL)*** button located at the surface description or select multiple properties and click the ***RELOAD(RL)*** button on the bottom of the main window. As every surface is loaded separately the loading time of the surface representations may take some time. After loading you can easily switch between loaded surface representations with the function keys without a loading delay. The protein structure file which was used to create the surface can be loaded separately into PyMOL using the normal PyMOL file PDB loading commands.

For the testfile surface (***test_db.ppix.gz***) the output should look similar to Figure 1.:



```
F:\Daten\Uni\diss\Crystal_Contacts\Datenbank\Pdb_database                          gr\PPI
X_all\msms2r3d_colo_new7
You clicked on Reload(RL)
```

PPIX PROPERTY SETTINGS
Protein Protein Interaction Database for Crystals
Visualize Annotated Surface Contacts
2007
v0.76beta

Separate Surface Output : □ No ☑ Unit □ Patch

PROPERTY	Reload (RL)		min color	---> mid color	---> max c
Lipophilic Potential	RL	☑	blue	lgreen	red
Patch Point Distance	RL	☑	dred	yellow	dgreer
Electrostatic Potential	RL	☑	dred	white	dblue
Patch ID	RL	☑	dblue	dred	dgreer

Status: Surfaces for test (Extention _db.ppix.gz) Reloaded.

Figure 1: Example of a test_run file (test_db_ppix.gz) loaded into PyMOL with the surface loader Plug-In

For more information about PyMOL commands, type *help* in the command line. For more information about the PyMol command line see: http://pymol.sourceforge.net/newman/user/S0210start_cmds.html