

README

1 aRMSD – minimalWindowsSupport

1.1 background

aRMSD (<https://github.com/nbehrnd/aRMSD>),¹ a program to compare molecular structures with each other, works sufficiently well in an ecosystem of Debian, or Xubuntu. At present (December 2018), I faced difficulties to work with the program in Windows.

*This project aims to provide an elementary support for aRMSD in Windows, allowing to use the program at all by help of WinPython² – even if some of the functions present in program’s source are not supported by the Python version provided. It is meant as a *ad hoc* and temporary fix only.*

In contrast to other distributions of Python (e.g., from www.python.org, or compilations like Anaconda / Miniconda) providing a permanent installation into your system, WinPython provides a *self-contained* functional Python for the Windows system. In other words, you basically just decompress an archive – if needed, you may add additional Python modules (e.g., as a wheel found on Gohlke’s directory³, or via pip) – and are ready to go. Literally, you may carry this one with you on a USB thumb-drive. Hence, you do not need administrator privileges to use initiate and use this Python. In addition, it provides it’s own separate and independent system-independent interfaces (CLI, IDLE, etc.); so you may use it side-by-side with any other preexisting installation(s) of Python, either by source (e.g., Anaconda, Python, WinPython), release (e.g., an already installed Python 2.7.15 and an other Python 3.6.7) or version (32 bit / 64 bit, if supported by the OS).⁴

The following outlines the essential steps to use aRMSD in a 64 bit system (and Python 3.6.5), or in a 32 bit system (and Python 2.7.13) allowing at least a *basic use* of the program, the scrutiny of models in the *.xyz format.

¹Forked from the original branch <https://github.com/armsd/aRMSD>.

²<https://winpython.github.io/>

³Unofficial Windows Binaries for Python Extension Packages, <https://www.lfd.uci.edu/~gohlke/pythonlibs/>, accessed in December 2018.

⁴The WinPython distribution includes a package manager allowing you to anchor the WinPython’s Python *currently* used into the system’s Path variable, and consequently, to then become the one system-wide recognized default. This will over-write previous Python-related settings, though. If desired, launch the package manager, and explicitly choose the “register” option.

1.2 Example Windows 7 Pro, 64 bit (Python 3.6.5)

The footprint of this *temporary fix* is quite large (about 2 GB permanent hard disk space is needed), since WinPython will provide you many more packages than required to run aRMSD.

WinPython (version WinPython64-3.6.7.0Qt5)² provides Python 3.6 altogether with some of the dependencies outlined by Arne Wagner: matplotlib, uncertainties, and performance related cython. With 488 MB size (as compressed archive), it is too large to be mirrored here below, but beside on the project page itself you find it here on a dedicated page on GitHub.⁵

The other *essential* dependency of aRMSD is the vtk-rendering engine.⁶ The wheel-directory maintained by Christoph Gohlke³ provides one vtk-wheel suitable here. It is worth about 27 MB and is mirrored in *this* repository (folder cp36-win_amd64).

1.2.1 files needed

The two are characterized by these md5sums:

```
2254b65e50a8c1834d10d253e243d23a  VTK-7.1.1-cp36-cp36m-win_amd64.whl
72b0612de9fdc341e87f01d9ca7b230f  WinPython64-3.6.7.0Qt5.exe
```

1.2.2 Install process

It is mandatory that the hosting Windows operating system is 64 bit. The system's PATH variable of the system is not touched, and no administrator privilege is required. However, anticipate about 2 GB disk space to be used since WinPython will provide you with *many* more packages than only the ones needed to run aRMSD.

Download the two files into a directory easy accessible for you. A mouse double-click on the WinPython executable will extract an archive. After about one or two minutes, the newly generated folder WPy-3670 equally contains an entry

WinPython Control Panel.exe

which basically is the package manager of WinPython.

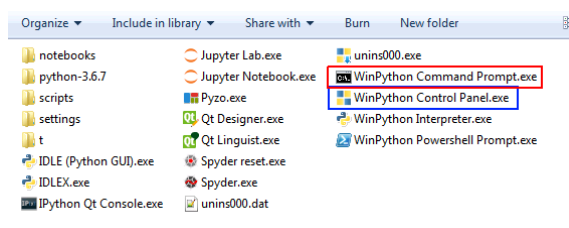


Figure 1: Automatically extracted folder by WinPython. Note the entry about the package manager (Control Panel, blue rectangle), and the CLI (red rectangle).

⁵Project site's entry: <https://github.com/winpython/winpython/releases/tag/1.11.20181031>, download link: <https://github.com/winpython/winpython/releases/download/1.11.20181031/Winpython32-3.6.7.0Qt5.exe>

⁶See <https://www.vtk.org/> and <https://en.wikipedia.org/wiki/VTK>.

A double-click on this opens this. Choose now the index card Install / Upgrade Packages. In the bottom left corner of this display, you find a button to point this manager to the location of the vtk-wheel provided. After a few moments, this selection will show up in the main menu of the manager, too. Subsequently, push the button in the bottom right corner to add the wheel to the packages considered by WinPython eventually managed within the WPy-3670 folder. Again after a few moments (about 10 sec), the manager will install the wheel.

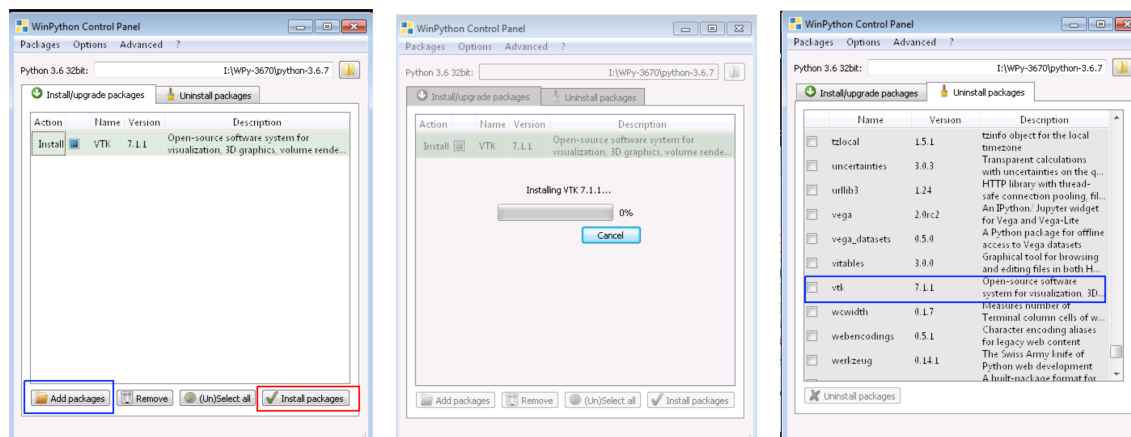


Figure 2: Stages of the installation process. Left figure: the manager with a wheel already loaded (blue rectangle) *prior* to launch the installation (red rectangle) on the index card "Install / upgrade packages". Middle figure: processing the wheel-installation. Right figure: confirmation of installation (index card "Uninstall packages")

Once the intermediate installation notifier clears up, you may close the manager entirely.

1.3 Example Windows 7 Pro, 32 bit (Python 2.7.13)

This approach departs from a minimal Python environment requiring a few steps more than the one in the other section. Installing only the essential modules to work with aRMSD leaves you with a considerably smaller footprint, too; only about 450 MB in comparison to about 2 GB of the former.

1.3.1 files needed

Download both the executable as well as the wheels³ into a directory easily accessible for you. They are all mirrored into this project's folder cp27-win32. Their md5sums are:

d02cb43ee04f977b5e017470c67d4de3	Cython-0.28.6-cp27-cp27m-win32.whl
e23338f500c076e0703d01109ddf09b6	future-0.17.0-py2-none-any.whl
07b9f3f2dcbb050486975e87d8d83a98	matplotlib-2.2.3-cp27-cp27m-win32.whl
bdf843a0fc9289169b1285fbfa19175e	mk1_service-1.1.2-cp27-cp27m-win32.whl
80f2fb38172ef2b3fb9281310ea2416b	numpy-1.15.4+mk1-cp27-cp27m-win32.whl

```
6203552bd9effc238a37cc0cd30436f3 openbabel-2.4.1-cp27-cp27m-win32.whl
8344450ccfb5864bb487c9a1a731c263 PyQt4-4.11.4-cp27-cp27m-win32.whl
211afcecc308b06a18d114789a6053c9 uncertainties-3.0.3-py2-none-any.whl
fddde0173dcd04015b77591e9f778746 VTK-6.3.0-cp27-cp27m-win32.whl
492a27996270cc4ecc0eacf5701fec5f WinPython-32bit-2.7.13.0Zero.exe
```

1.3.2 installation process

Decompress the archive by mouse double-click on the exe, which will generate again a separate folder within the current directory.

This will provide you a Python environment this slim (remember "zero" is part of the file name), that the GUI of the package manager won't work yet. With the CLI within this folder (WinPython Command Prompt.exe), enter the directory where the Qt-wheel is located. As example,⁷ if the wheel was copied into d:\toto, then cd into this directory, and launch the installation from the CLI by

```
wppm -i d:\toto\PyQt4-4.11.4-cp27-cp27m-win32.whl
```

The CLI / WinPython Command Prompt.exe will inform you explicitly when this process accomplished. Subsequently, you may close *this* interface.

Now you are able to launch the GUI of the package manager. Drag-and-drop all the other wheels provided, and to launch their installation. Again, the package manager progressively informs you about the process, and may be closed once the task is completed.

1.3.3 note about this installation

Gohlke's directory³ equally offers VTK-7.1.1-cp27-cp27m-win32.whl not included here. While your system may differ from the one accessible to mine, updating to this version *broke* aRMSD.

1.4 Known limitations of this fix

This *temporary fix* allows you to align and scrutinize model data – at least in the most elementary *.xyz format – from the CLI of WinPython by

```
python armsd/aRMSD.py
```

to generate statistics plots and a permanent record.

Depending on the installation, aRMSD may inform you about the missing link to openbabel.⁸ (Since the Path variable is not touched, it will not recognize openbabel even if it were installed on the hosting computer, either.) Consider this freeware if you need to convert models provided in a file format different than *.xyz.

Some options vtk may offer (e.g., anaglyph representation) may be unavailable here. Saving the renderings by vtk as *.png (key stroke s) however is supported.

◇

⁷The source for this tip is derived from <https://github.com/winpython/winpython/issues/397>.

⁸Open Babel: The Open Source Chemistry Toolbox, http://openbabel.org/wiki/Main_Page