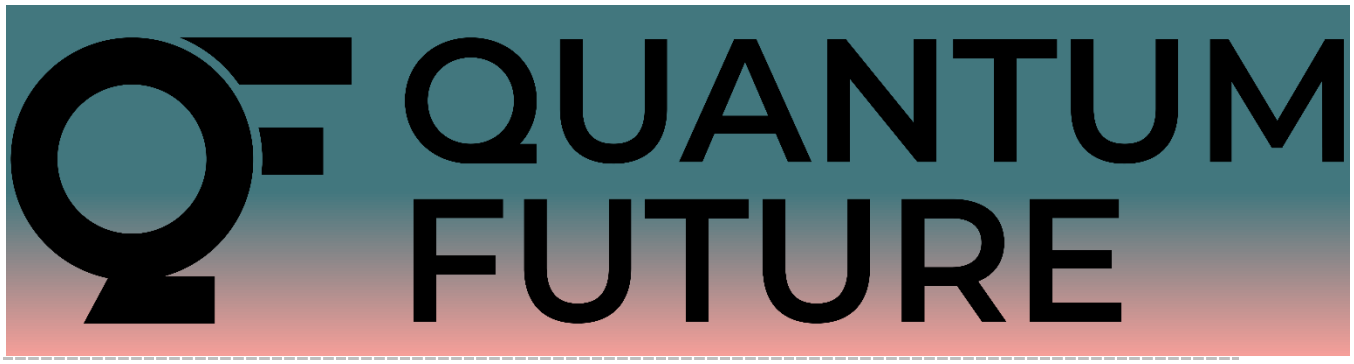


Release Notes for the QF2024 Software

Laszlo Fusti-Molnar, Owner, and CEO

QuantumFuture Scientific Software LLC

E: laszlo@qfsciences.com

Vision

To express my vision clearly, I aim to avoid creating yet another general-purpose *ab initio* sandbox. I don't want to burden users with reading manuals, deciphering intricate input file formats, or delving into the complexities of various functional and basis set specifications. Instead, my goal is to deliver a product that effortlessly conducts precise dispersion corrected, modern DFT calculations. Users should require minimal knowledge, and the process should remain user-friendly.

Moreover, I am committed to ensuring exceptional performance, which significantly accelerates research projects and leads to substantial savings in computational time, energy, and operating costs. I sincerely appreciate your investment of time and trust in our software. It is my hope that you'll discover this investment to be one of the most rewarding decisions you've made.

You are the ultimate judge of how well our program aligns with the vision outlined above and how effectively it supports your day-to-day research and potential future projects. Please feel free to share any feedback you may have and thank you for choosing QF software!

Applications in this Release

This software release incorporates several significant scientific applications: qfdft.x, qfconfsearchDFT.x, qftorsionscan.x, and qfLowerLevel.x. Brief descriptions of each are provided below. Additionally, we offer a user-friendly GUI program that enables you to execute qfdft.x, qfconfsearchDFT.x, and qfLowerLevel.x calculations effortlessly within a graphical environment, eliminating the need for manual command input. Furthermore, the qfazurelaunch.x application simplifies the submission and execution of large projects on the Azure cloud with just one straightforward command. Importantly, this solution on Azure is not only entirely automated but also exceptionally cost-effective, as it exclusively leverages heavily discounted spot instances while efficiently managing their limitations. The following section offers concise introductions to all the mentioned applications.

QFDFT is an exceptionally fast DFT program, which harnesses the power of very sophisticated Fourier transformation and other numerical grid-based algorithms using all-electron Gaussian basis sets to deliver unparalleled computational speed and accuracy for drug and material design. This revolutionary application performs *ab initio* DFT-D4 energy calculations, analytical force calculations, geometry optimizations and thermochemistry calculations with exceptional efficiency, while preserving the full precision of traditional non-relativistic all-electron ground state DFT calculations using Gaussian basis sets. Additionally, it incorporates modern D4 VDW corrections, optimizing D4 parameters for every supported basis set and functional combination, further enhancing accuracy without incurring additional computational costs. Notably, QFDFT supports the utilization of original D4 parameters developed by Professor Grimme's group where available. Please note that QFDFT is not designed for quantum chemists seeking a broad range of DFT calculations options, or a sandbox of unlimited stable/unstable tested/untested features as there are already ample programs available for such purposes. Instead, our primary focus is to provide the drug design and material design communities with an exceptionally efficient tool, enabling them to obtain highly precise dispersion-corrected DFT-D4 energies, forces, optimized geometries and thermodynamic properties effortlessly, without requiring in-depth knowledge of DFT functionals and basis sets.

QFConfsearchDFT is a robust application that seeks low energy conformations at the quantum mechanical semi-empirical level of theory, followed by performing accurate DFT-D4 energy calculations and sorting the final structures based on DFT-D4 energies. Recently we were testing this application in detail by using 150 FDA approved drug molecules having 25 examples for each 2-7 rotational bond categories. We have found extraordinary accuracy with this application! See all the details below. This advanced tool not only supports vacuum calculations but also includes several important solvents. The resulting low energy conformations, sorted based on their DFT-D4 energies, are outputted to an SDF file. The application utilizes RDKit C++ APIs and the XTB program suite as third-party tools in the conformation search process, leveraging their useful functionalities. Prior to obtaining quantum mechanical optimized geometries using the GFN-XTB method, users have the flexibility to choose between MMFF, UFF, and GFNFF force fields for conformation enumeration. It is worth noting that the results can be influenced by the force field used in the enumeration process, as different force fields may yield varying numbers of local minima at different geometries. To ensure robustness, the default behavior is to employ all three force fields in the conformation search process and combine the results before performing QM calculations. For users seeking the most rigorous approach, an even more robust solution exists by using *ab initio* DFT-D4 geometry optimizations instead of semi-empirical QM optimizations. To do so set the --stopAfterFF option to true. With this setting, the application collects all reasonable geometries generated by the different force fields and terminates. The resulting output SDF file can be divided into individual molecules, which can then undergo separate DFT-D4 geometry optimizations using our qfdft.x application. Although this process requires a few additional steps and is not integrated within this application, it is worth noting that DFT-D4 geometry optimizations, even with our highly efficient DFT-D4 program, typically entail calculation times at least an order or two orders of magnitude longer than those of this application. Thus, distributing

the workload across multiple computational nodes proves to be a more practical approach in this case. An automated solution for this option is currently under development on Azure and will be available soon.

QFTorsionScan is a powerful software combines quantum mechanical semi-empirical calculations with sophisticated conformational search algorithms to generate low-energy conformations for each sampled torsion value, as well as construct highly informative torsional energy profiles. The application employs a two-step approach to accurately explore the conformational landscape. Initially, it performs conformational sampling at the quantum mechanical semi-empirical level, ensuring an appropriate representation of molecular flexibility. Subsequently, geometry optimizations are conducted by freezing the sampled torsion angles for all conformations. This enables thorough exploration of the potential energy surface, resulting in refined structures and precise energy evaluations. The conformations are then sorted based on their energies, allowing for efficient identification of the lowest energy conformations for each torsion scan. To accomplish these tasks, the application leverages the RDKit C++ APIs, and it incorporates the XTB program suite as well.

QFLowerLevel is an advanced and comprehensive application specifically designed to facilitate non *ab initio* computational methods for a wide range of computational chemistry calculations. This versatile application combines the powerful capabilities of three third-party programs, XTB, MOPAC and RDKit, while also incorporating important additional features developed at QF in-house. The application seamlessly integrates the GFN-XTB quantum mechanical semi-empirical approach from XTB, PM6-D3H4X and PM7 methods from MOPAC, GFN-FF force filed from XTB, MMFF94 and UFF force fields from RDKit with our QF programs. The application offers the same capabilities as QFDFT does i.e. energy calculations, geometry optimizations with or without constraints and thermochemistry calculations. In terms of geometry optimizations, the application utilizes the native optimizer from the XTB package in case of no constraints while using our QF optimizer for PM6-D3H4X and PM7 methods. When the user request constraints of any kind in geometry optimizations, for example we have one keyword to request optimizations of all Hydrogen positions in the molecule while keeping all non-hydrogen atoms fixed, then QF optimizers are used for all semi-empirical QM and all force field-based methods. We obtain or calculate the hessian of a given method by numerical differentiations and then utilize our QF programs to calculate the thermodynamic quantities for all supported methods in the case of thermochemistry calculations.

All major applications use command line input parameters, no input files are needed to define any calculation options, and the only input files required by them are the molecular structures themselves. To get started, please run all three applications with the "--help" option to get familiar with the current options and capabilities. It probably takes no more than 2-3 minutes of reading per application. This release comes with examples as well as some small scientific studies for which all outputs and necessary scripts are included, so you might be able to reproduce all the calculations and get help in starting similar projects.

In the next chapters of this document, I will provide some examples for the qfdft.x application. I believe that the usage of the other applications is so trivial that additional explanations than those produced by the "--help" option are not needed, so instead of focusing how to use them I included some results of an interesting small scientific study which explain their usefulness for your own projects.

In addition, we have the first version of a Graphical User Interface QFGUICalculationsLauncher application that allows to run all major scientific applications in GUI environment without typing any linux command. This application allows one QF application at the time on the local node for now and it will be extended to run any number of calculations on HPC clusters and on public or private clouds soon.

The qfazurelaunch.x application offers the capability to effortlessly initiate extensive projects on Azure with just a single command, with all necessary operations handled automatically. At the project's conclusion, results can be conveniently downloaded using an additional single command. Users are required to possess and appropriately configure their own Azure account, as all calculations are executed independently under their account. There is no need for any prior knowledge of cloud computing; the entire project can be completed using just these two straightforward commands.

In addition to its remarkable simplicity, our approach exclusively employs heavily discounted spot instances, which offer discounts of up to 90% compared to on-demand instances. We have also mitigated the potential drawbacks associated with such highly discounted computational resources. Rest assured, it's not too good to be true! See the example below for a demonstration.

Installation

Hardware requirements: Any X86-64 (https://en.wikipedia.org/wiki/X86-64) compatible processor capable of AVX512 or at least AVX2 vector instructions.

Supported operating systems: The "native" package has undergone rigorous testing on Ubuntu 20.04. Furthermore, we offer a publicly available Docker image based on Ubuntu 20.04, which is compatible with virtually all Linux distributions.

It's important to note that our Docker image has also been tested on Windows 10 and later versions. However, we have observed a significant performance slowdown when running our Docker containers on Windows, resulting in a 30-50% increase in computational expenses. Clearly, this is not an acceptable outcome, and we are actively planning to provide a native Windows version in future releases. On the other hand, running our Docker containers on Linux introduces only a minor performance impact, approximately 5%. Unfortunately, we do not support macOS since Apple's new CPUs lack X86-64 compatibility.

While we do not officially test and support the native package on all Linux distributions, users who choose this option may require some guidance. In terms of dependencies, this distribution includes an essential "libs" directory that contains the necessary libraries. Beyond these libraries, we needed to install a few additional packages when building the Docker containers from standard Linux distributions. Here are the commands for several major Linux versions:

Ubuntu20.04, Ubuntu22.04

```
apt update
apt install --assume-yes build-essential
apt-get install --assume-yes libcairo2-dev
apt-get install -y libunwind-dev
```

Rockylinux8 (Redhat 8 compatible) and Rockylinux9 (Redhat 9 compatible)

yum -y update

yum -y groupinstall 'Development Tools'

yum -y install cairo-devel

yum install epel-release -y

yum -y install libunwind-devel

Other relatively recent Linux versions would probably work successfully as well. Older Linux versions with older and incompatible system libraries will not work in terms of native installation and our docker image can be used in those situations.

Installation steps:

1. Download the installer application from

   https://bettermolecularmodelling.com/qffileexchange/QF23Beta/QFInstaller

   and run the installer.

2. Please DO NOT PROCEED IF YOU DO NOT ACCEPT THE TERMS IN THE LICENSE AGREEMENT!

3. Please send an email to support@qfsciences.com to request a license file from us. Please indicate either in the subject or in the body of the email that you accept the license agreement!

4. When you receive your license files, please copy them to the directory where the executable programs are installed. Note, that in the docker version, the license files must be in the same directory as the input molecular structure files.

5. Using the native version:

   a. Set a few necessary environmental variables. Here are the commands in bash for instance:

```
export QuantumFuture=$HOME/QF23Beta
export LD_LIBRARY_PATH=$QuantumFuture/libs
export qfsciences_LICENSE=$QuantumFuture
export PATH=$QuantumFuture:$PATH
export RLM_LICENSE_PASSWORD=YourPasswordHere
export LD_LIBRARY_PATH=$QuantumFuture/Qt5/lib:$LD_LIBRARY_PATH
export QT_PLUGIN_PATH=$QuantumFuture/Qt5/plugins
```

   In my case the installation directory was $HOME/QF23Beta. If that is different for you, then be sure to modify that line accordingly. Your password is included in the license file that we sent you.

   b. If your CPU is equipped with AVX512 vector instructions, then run one of the **SetLinksToAVX512.sh** shell script. Older processors have less sophisticated AVX2 vector instructions. If this is the case with your CPU then run one of the **SetLinksToAVX2.sh** scripts. Super old CPUs without the AVX2 vector instructions are not supported. *Please do not do step B before step A! The scripts in step B are using the **$QuantumFuture** environmental variable that the previous step sets!*

   c. Sit back and enjoy the ride...

We highly recommend running all major programs with the --help argument and reading the short help (about a 1-2 minute read each) in order to have a quick understanding about the current capabilities and about the meaning of the input parameters. **Give us your feedback in as many details as possible!**

6. Using the docker version:

   a. Install docker under your desired operating system. All necessary information about how to do so can be found online. You do not have to learn the docker's syntax to use our software, although we encourage you to do so. Following this short section below and making aliases you will be able to run our programs without any additional knowledge of docker.

   b. To get the current version just use the following command:

```
docker pull quantumfuture/2024:latest
```

   if you are not member of the docker group on linux then add sudo in front:

```
sudo docker pull quantumfuture/2024:latest
```

   Otherwise just neglect the sudo in all commands below.

   c. Copy the license file to the directory where your input files are located. The license file needs to remain in the directory where the calculations are running. Note that the license file can also be in the directory where the executables are in the native version without docker, but since the license file is not part of the docker distribution it is not possible to do so in the docker version. In principle it is possible to make a private docker image with the license file in it but if you do so be very careful! Docker hub makes it super easy to change a private image to public with just one click and a small confirmation step. After that everybody can use that image together with that given license and perform unlimited calculations until the license

counter goes to zero. In any such cases and in general if you feel that your license has been potentially stolen or compromised in any way, please let me know immediately!

d. Usage: Our preference is that we can use the programs the same way as in the native version without docker, so we have tried to organize everything accordingly. For instance, here is the command for performing a DFT-D4 calculation.

```
qfdft_AVX512.x --Input mymolecule.sdf --CalcType 3 >& mymolecule.log
```

We want the usage of the Docker program to be similar, so if we make an alias such as

```
alias dockerQFDFT="sudo docker run --rm --name qfdft --env
RLM_LICENSE_PASSWORD=YourPasswordHere  --workdir=/qftmp  -v
$(pwd):/qftmp quantumfuture/2024  qfdft_AVX512.x”
```

then we can use the program in a similar way:

```
dockerQFDFT --Input mymolecule.sdf --CalcType 3 >& mymolecule.log
```

This way only the executable name being different indicates that we are using the docker version. All outputs go to the current working directory where the inputs are located just like how the program works natively without docker. One small remark: On windows the $(pwd) needs to be replaced with ${pwd}. With that little change it runs fine on windows 10, besides the unfortunate slow down compared to native Linux runs.
All other major applications work the same way.

e. Note that you can only use one docker container on a given node, since we gave the qfdft name to the container and the name of the container must be unique. If you want to run many containers within one node just skip the --name qfdft part and docker will assign a unique name for your containers. So, in this general case the alias would look like:

```
alias dockerQFDFT="sudo docker run --rm --env
RLM_LICENSE_PASSWORD=YourPasswordHere --workdir=/qftmp  -v
$(pwd):/qftmp quantumfuture/2024  qfdft_AVX512.x”
```

This is true for QFDFT and similar for the other applications. In this case you do not have the qfdft name for your containers, but you can use more than one at the same time.

Licensing scheme, license options and prices

Our licensing vision is ambitious, aiming to provide a user-friendly scheme that seamlessly adapts to various computational environments. Whether you're using a personal laptop, desktop, private computer cluster, or cloud infrastructure, our licensing scheme is designed to work effortlessly. We've achieved this by leveraging Reprise licensing alongside our custom implementation using Reprise APIs.

An integral aspect of our licensing scheme is the "pay as you go" feature. We charge license fees based on actual usage, following a model like public cloud providers when provisioning computational resources. Our licensed applications are categorized into three different license versions: "Large," "Medium," and "Small."

- The "Large" license version is intended for use on large computers with up to 64 OMP threads, with a licensing charge of $1/hour per instance.
- The "Medium" versions are designed for smaller workstations with up to 8 OMP threads, with a licensing charge of $0.5/hour per instance.
- The "Small" versions are suitable for smaller workstations with up to 4 OMP threads, with a licensing charge of $0.33/hour per instance.

Our applications automatically detect the number of OMP threads in use and apply one of the three licensing models accordingly. It's worth noting that renting a 16-physical Intel core node through AWS (referred to as a 32 vCPU node) typically costs around $1/hour per node. The minimum cost for all three licensing models is 10 cents, with an additional charge of 10 cents every 6, 12, or 20 minutes as the applications periodically contact our license server.

We offer these prices uniformly for every computational unit, whether it's a laptop or a very large computational node. The sole technical requirement is a stable internet connection since the applications periodically communicate with the license server.

We provide two payment options:

1. **Prepay Option**: Purchase computational credits on our website at qfsciences.com before starting projects. You can continue using our software until the licensing credits are exhausted or refilled. No service agreement is required for this option.
2. **Monthly Billing Option**: For trusted customers, we offer computational credits (and refills if needed) as a service without an initial investment. Monthly billing is based on actual usage, but it necessitates signing our service agreement to authorize us to bill monthly.

Beta testers and potential new customers can obtain licenses for evaluation purposes at no charge. We are generous in offering hundreds of node-hours of free licenses for this purpose.

Short history with some theoretical comments about QFDFT

The development of QFDFT started on January 1st, 2018, after I founded QuantumFuture Scientific Software LLC. Since I had left the *ab initio* methodological development field for computational drug design around 2006, I started to refresh my memory by reading many articles including my own scientific papers about the Fourier Transform Coulomb (FTC) method. It is a method that I had developed about 20 years ago for fast linear scaling evaluation of the Coulomb matrix elements in DFT. There were, however, many unsolved problems in that method such as the need of analytical two electron integrals which dominated the computational costs for certain class of integrals which FTC method could not handle accurately. Through sheer dedication, I solved every single one of these problems without compromising anything in terms of accuracy, resulting in a fantastic solution for the Coulomb matrices that sacrifices neither performance nor accuracy. I call this technology QFC, which stands for the QuantumFuture's Coulomb algorithm. Another integral part of the DFT calculation is the computation of the exchange-correlation matrix elements. Together with my coworkers at QCHEM, we had developed the MRXC method, which shared similarities with FTC method to accelerate the computational speed for the exchange-correlation matrix elements. I have developed a new, more advanced version of it, and the resulting software was already more than efficient enough to provide preliminary results and to hold its own in the application for a phase I NIH SBIR grant to implement analytical atomic gradients of the DFT energy. Later, I worked out a completely new numerical grid technology that is based on a completely different idea, which turned out to be much simpler than MRXC, very accurate, and performant, making it clearly superior to any version of my MRXC based implementation. I call this method QFXC, and this is the technique which is used in QFDFT. The third important aspect of DFT programs is the computational expenses of the linear algebra routines. There are many dense matrix-matrix multiplications involved in the DIIS based SCF iterative process and they scale cubically with the number of basis functions as well as the diagonalization of the KS matrix. The good news is that for drug-size molecules, even with accurate basis sets which use polarization and diffuse functions, the computational cost of the linear algebra evaluations is usually low and only gains any significance after about 5000-6000 basis functions in QFDFT. Thus, linear algebra cost is not an issue in any DFT calculations for typical drug-size molecules or clusters, while it makes still impossible to perform DFT calculations with good quality basis sets for entire proteins or protein-ligand systems. This issue is high on the list of priorities to try to solve soon by making use of either the Divide and Conquer or the FMO algorithms.

The development of QFDFT commenced on January 1st, 2018, coinciding with the founding of QuantumFuture Scientific Software LLC. After transitioning from the computational drug design field around 2006, I embarked on refreshing my knowledge by extensively reviewing scientific articles, including my own papers, pertaining to the Fourier Transform Coulomb (FTC) method. This method, which I had developed approximately two decades ago, was designed for the rapid linear scaling evaluation of Coulomb matrix elements in Density Functional Theory (DFT). However, it had certain unresolved challenges, notably the necessity for some certain categories of analytical two-electron integrals involving two or more

core-type basis functions, which posed computational bottlenecks for specific types of integrals. Through relentless dedication, I successfully addressed each of these challenges, achieving a remarkable solution for Coulomb matrices that neither compromises accuracy nor performance. This innovation is referred to as QFC, denoting QuantumFuture's Coulomb algorithm.

Another crucial component of DFT calculations involves the computation of exchange-correlation matrix elements. Collaborating with my colleagues at QCHEM, we had previously developed the MRXC method, which shared similarities with the FTC method to enhance computational speed for exchange-correlation matrix element calculations. Subsequently, I created an advanced version of MRXC, resulting in software efficient enough to provide preliminary results and secure an application for a Phase I NIH SBIR grant aimed at implementing analytical atomic gradients for DFT energy calculations.

Later, I introduced a novel numerical grid technology based on an entirely different concept, which proved to be simpler, highly accurate, and efficient, surpassing any previous MRXC-based implementations. This method, known as QFXC, is the foundation of QFDFT.

The third critical aspect of DFT programs pertains to the computational expenses associated with linear algebra routines. Numerous dense matrix-matrix multiplications are inherent in the DIIS-based SCF iterative process, and their computational cost scales cubically with the number of basis functions, as well as with the diagonalization of the KS matrix. Fortunately, for molecules of typical drug-size, even when employing accurate basis sets with polarization and diffuse functions, the computational cost of linear algebra evaluations remains relatively low. It only becomes significant when dealing with systems containing approximately 5000-6000 basis functions within QFDFT.

As a result, linear algebra cost is not a concern for typical DFT calculations involving drug-sized molecules or clusters. However, it still poses challenges for performing DFT calculations with high-quality basis sets for entire proteins or protein-ligand systems. Addressing this challenge is a top priority, and we aim to explore potential solutions such as the Divide and Conquer or the Fragment Molecular Orbital (FMO) algorithms in the near future.

Capabilities

This version of the software is exclusively designed for conducting DFT closed-shell singlet ground state calculations using non-hybrid functionals. In addition to performing modern dispersion corrected DFT energy calculations, the software excels at analytic atomic force calculations, geometry optimizations, constrained geometry optimizations, and statistical thermodynamic calculations. We have seamlessly integrated the D4 dispersion correction scheme developed by Professor Grimme's group, along with its analytical gradients. Consequently, the resulting DFT-D4 energies and forces exhibit remarkable accuracy when coupled with precise functional and basis set selections.

To enhance the accuracy of dispersion corrections and molecular interactions, we have devoted additional effort. The D4 parameters are unique for different functionals and are meticulously fitted by Professor Grimme's group using large basis sets that approach the DFT basis set limits and are free from Basis Set Superposition Errors (BSSE). The question arises as to whether these same dispersion correction parameters should be used when employing smaller and more practical basis sets that accurately describe molecular interactions but fall short of the basis set limit with some BSSE. The answer is clear: for practical basis sets, it is more beneficial to fit D4 parameters for each (basis set – functional) pair. While this approach may be less practical for general-purpose *ab initio* packages that support numerous functionals and basis sets, our focus on a select few high-quality options allows us to determine optimal parameters for each (basis set—functional) pair. For in-depth details, please refer to our scientific study below.

Let me take a moment to underscore why this validation study holds immense value and what it signifies for our users. Our aim in designing and building this product is to make it immensely useful and practical in critical domains of computational drug design. This includes the precise determination of conformations, geometries, and strain energies of drug-like molecules, as well as the accurate calculation of interaction energies between drug molecules or lead molecules and proteins, and the faithful representation of organic compound interactions in molecular crystals, among other applications. An accurate and validated intermolecular interaction model stands as a fundamental pillar in achieving these objectives. Initially, this might seem counterintuitive, especially in the context of conformations. However, the accuracy of nonbonded interactions is intricately linked to the precision of intermolecular interactions.

The study has led to three major conclusions:

1. The revSCAN and revTPSS functionals demonstrate exceptional accuracy across a wide range of tested basis sets. Considering that the computational costs of these functionals are on par with other meta-GGA functionals and only slightly higher than GGA functionals, there are few scientific reasons to opt for different functionals in typical drug design projects. We intend to expand this study to cover additional standard benchmark datasets for conformational relative energies to verify the universality of this finding.
2. The 6-311G++(df,pd) basis set essentially approaches the basis set limit, devoid of BSSE, and delivers accuracy comparable to much larger basis sets such as pc2 or def2-TZVPPD. Our default basis set choice for energy and force calculations is def2-TZVP, considering its optimal balance

between accuracy and computational speed, as corroborated by the results. The 6-311++G** basis set is also a very reasonable and practical choice.

3. When employing smaller and faster basis sets, the DFT-D4 interaction energies exhibit significant errors when using Grimme's parametrization for D4 VDW corrections. For such basis sets, our proprietary D4 parametrization significantly enhances the accuracy of calculated energies. Furthermore, reasonable results have been obtained using the smallest supported 6-311G** basis set with the revSCAN, revTPSS, and revM06L functionals, despite the absence of diffuse and additional polarization basis functions.

It's essential to note that this software does not support hybrid functionals due to the associated significant increase in computational costs stemming from the calculation of exchange matrix elements. In my view, hybrid functional support is unnecessary, given that we can achieve remarkable accuracy without resorting to hybrid functionals in contemporary DFT applications. Nevertheless, QFDFT does have certain limitations; it does not support g-type basis functions or f-type core-like basis functions. To address these limitations, I made minor modifications to a small number of basis sets, primarily for transition metals and non-production larger basis sets. These modifications are not expected to significantly impact any computed physical or chemical properties.

As of now, QFDFT exclusively employs Cartesian basis sets, although there is consideration for potentially offering spherical basis sets in the future. This aspect may be less crucial today compared to three decades ago. For drug-sized molecules, Cartesian basis sets offer greater flexibility and include additional d and f-type basis functions, which can contribute to lowering energies compared to their spherical counterparts, at least for practical basis sets. The consideration of spherical basis sets may primarily serve to expedite computational times in matrix-matrix multiplications and KS matrix diagonalizations.

An additional technical consideration relates to the use of large Gaussian basis sets for larger molecules. This can potentially introduce linear dependence problems and hinder SCF convergence. While many quantum chemistry programs adopt techniques to mitigate linear dependence issues, I have chosen not to employ such an approach at present. My decision is rooted in the fact that I have not encountered SCF convergence problems with the supported production basis sets, even for sizable systems. SCF convergence issues typically arise from unfavorable geometry or small HOMO-LUMO gaps. Employing linear dependence reduction techniques effectively reduces the basis set size and, in turn, increases energy values—strictly speaking, this is not ideal. Moreover, it can introduce discontinuity problems in potential energy surfaces. Thus, if SCF convergence remains unaffected by using the full basis set it is preferable to refrain from employing such basis set reduction techniques.

In terms of electrostatic properties, this release includes support for a limited number of properties, including molecular dipoles, atomic Mulliken and EEQ charges, electrostatic potentials at each atomic site, atomic multipoles using Distributed Multipole Analysis, HOMO and LUMO energies, and more. We anticipate the forthcoming implementation of additional valuable properties such as electron densities, electrostatic potential, HOMO and LUMO orbitals on a grid for research and visualization purposes.

We incorporate continuum solvation models into our software using the xtb semi-empirical package. It's important to note that continuum solvation models are approximations, and some opinions suggest that they may introduce larger errors compared to certain force field calculations. Specifically, continuum models struggle to accurately represent the first and second hydration shells in systems involving ligands and proteins in a water environment. However, it's worth emphasizing that the difference between implementing a continuum solvation model within a semiempirical method versus integrating it into an *ab initio* DFT code is expected to be significantly smaller in magnitude than the errors introduced by the inherent approximations of continuum solvent models. When using solvation models at the semi-empirical level as opposed to the *ab initio* level, the impact on solvation effects is minimal, while simultaneously offering substantial reductions in computational time for DFT calculations and streamlining development efforts.

Accuracy tests

This version includes a directory containing DFT results for small molecules, along with two subdirectories featuring the same DFT calculations conducted using the GAMESS-US and NWCHEM software. The selection of small molecules for these fundamental tests includes H2O, Alanine, Histidine-dimer, and a molecule named XXVI, which is a drug-like molecule from CCDC and holds significance due to our successful prediction of its crystal structure in the 2016 blind challenge.

For these tests, two basis sets, namely 6-311G** and def2-svpd, were chosen, along with PBE and TPSS functionals. All possible combinations of functionals and basis sets were employed for each molecule using all three *ab initio* quantum chemistry programs. To ensure reproducibility, the input files for GAMESS and NWCHEM calculations are provided, and QFDFT results can be replicated by specifying the functional and basis set in the command line.

Upon careful examination of the results, it becomes evident that the agreement among all three programs is quite satisfactory for smaller molecules. However, for larger examples, an interesting observation emerges: NWCHEM appears to yield significantly higher energies compared to QFDFT and GAMESS. This discrepancy might be attributed to excessive basis set reduction due to linear dependence in NWCHEM. Unfortunately, I could not identify a straightforward input option to enhance accuracy. QFDFT, on the other hand, appears to maintain a high level of accuracy.

If you possess insights on how to enhance the accuracy of NWCHEM calculations, we encourage you to replicate these calculations and share your findings with us. Additionally, for an in-depth exploration of the accuracy of DFT-D4 VDW dispersion corrections, please refer to our concise study in the subsequent chapter.

Practical VDW-D4 parameterizations for DFT-D4 calculations by using optimal parameters for functional-basis set pairs

(Presenting examples of good, bad, and ugly results)

Introductions, motivations

The most modern fourth generation DFT dispersion corrections (D4 VDW corrections) have been implemented in our software based on the article of [*Caldeweyher E, Ehlert S, Hansen A, Neugebauer H, Spicher S, Bannwarth C, Grimme S. A generally applicable atomic-charge dependent London dispersion correction. The Journal of Chemical Physics. 2019;150(15):154122*].

The two-body dispersion energies are approximated with the

$$E_{disp}^{(6,8)} = \sum_{A,B} \sum_{n=6,8} Sn \frac{C_{(n)}^{AB}}{R_{AB}^{(n)}} f_{damp}^{(n)}(R_{AB})$$

equation. We cannot go into every detail here, but we note that S_n are optimizable parameters where S_6 is usually kept one for most functionals and S_8 are optimized for each supported functionals. The most frequently used "damping" function has the form of

$$f_{damp}^{(n)}(R_{AB}) = \frac{R_{AB}^{(n)}}{R_{AB}^{(n)} + (a_1 R_0^{AB} + a_2)^{(n)}}$$

where the a_1 and the a_2 are also optimizable parameters. The value of the damping function is 1 at the asymptotically large atomic separation where the simple dispersion energy equation above is valid and smoothly goes to 0 with decreasing the atomic distance where the DFT functional takes over the description of the electron correlations. Thus, a_1 and a_2 are also functional dependent parameters. Traditionally all four parameters are determined by DFT calculations for popular functionals using large basis sets close to the basis set limits and fitted to very high quality CCSD(T)-CBS standard intermolecular interaction benchmark data sets. We discovered, however, that the assumption of these parameters to be basis set independent is an inaccurate approximation in some cases and significant accuracy gains can be obtained for medium size and very practical basis sets by optimizing S_8, a_1, $a_2$parameters for functional/basis set combinations. For instance, the RMSD error of the DFT-D4 energies compared to the accurate CCSD(T)-CBS energies in the S66x8 standard intermolecular benchmark set using revTPSS functional and def2-svpd basis set is 1.6 Kcal/mol using the published D4 parameters. This error is rather large and even good

quality force fields can provide more accurate results. After optimization of the S_8, a_1, $a_2$parameters using our private CCSD(T)-CBS data set we have repeated the same DFT-D4 calculations for the S66x8 standard set by using it as the test set. We noticed that the RMSD error of the interaction energies was reduced from 1.6 Kcal/mol to 0.62 Kcal/mol. This is a very significant improvement in accuracy. It is easy to understand the reason why such improvements are possible. Medium size basis sets like the def2-SVPD have some significant BSSE (Basis Set Superposition Error) which make intermolecular interaction artificially too strong while using very large basis sets close to the basis set limits the BSSE is negligible. When the VDW dispersion corrections are optimized using very large basis sets the resulting corrected DFT-D4 interactions energies are close to the reference energies (CCSD(T)-CBS) but obviously if we add the same VDW D4 corrections to the DFT energies obtained with medium basis set with some significant BSSE then we overestimate the reference interaction energies due to the BSSE. It is therefore much more beneficial to have a different VDW correction which goes down to zero more rapidly with the decrease of atom-atom distances than the original VDW correction does. Since medium size basis sets are the most important in practical computational drug design projects this research and developments offer significant impact in improving the accuracy of practical VDW corrected DFT-D4 calculations and this is the primary motivation of this work. Our QFDFT software can now offer not only exceptional calculation speed but also superior accuracy upon the re-parameterization of S_8, a_1 and a_2 parameters.

There are two important notes to highlight before we jump into the details. First, note that we do not look for general conclusions for all quantum chemistry problems. We want to focus on problems that are dominating the computational drug design field i.e., we would like to have accurate intermolecular interaction energies and accurate geometries since this is essential for a large range of problems from protein ligand interactions and interaction with solvent molecules, through the important non bonded interactions in conformational and strain energies to organic crystal structure and solubility predictions. Second, note that we do not want to develop a much more sophisticated method that can deal with all deficiencies coming from some basis sets. If we obtain three new D4 parameters for our supported basis set/functional combinations, then the calculation expenses do not change at all and implementing three new parameters for each functional-basis set pairs requires practically negligible development time (which is ideal for our startup company).

At the moment QFDFT supports PBE, BP86, TPSS, revTPSS, RGE2, revSCAN, R2SCAN, revM06_L functionals and 6-311G**, 6-311G++**, 6-311G(df,pd), 6-311G++(df,pd), def2-SVP, def2-SVPD, def2-TZVP basis sets for production calculations and additional pc2 and def2-TZVPPD large basis sets for special tests purposes. We have optimized the D4 parameters for all functional and production basis set pairs. revM06_L functional does not need VDW corrections of course. The training set utilizes our private CCSD(T)-CBS dimer set with nearly 200 dimers having 10+ points each along a given direction of interaction. This training set has some minimal overlap with the S66x8 benchmark set which we used as a test set. Note, that the S66x8 set was part of the training set in the parameter optimizations of Professor Grimme's research group and therefore we expect that at large basis sets using Grimme's parameters

provide slightly more accurate results for the S66x8 set but it does not necessary mean that it is more accurate in general since the optimum values are different for our training set for instance. Nevertheless, the differences are very small for large basis sets as we will show below. QFDFT program automatically utilizes the optimal QF VDW D4 parameters for all production calculations and with a simple command line option one can request to use Grimme's parameters if it is desired by the user and if it is available. The revSCAN and the RGE2 funtionals, for instance, do not have optimized Grimme's D4 parameters as far as we know. In addition, our parameterization is not considering exclusively the energy values of the dimer's data set and we have added two more quantities both in the training phase to obtain optimized D4 parameters and we calculate those quantities in the testing phase as well. First, we have determined the minimum locations and the minimum energies for all dimers based on the CCSD(T)-CBS and the actual model energy curves by fitting a simple quadratic function at their minimum and we considered the RMSD of the minimum locations as well as the RMSD of the minimum energies in our parameter optimization process. In this analysis we have determined the same quantities for the S66x8 test set and we tabulated the RMSD and the MD (mean deviation) of all energy points compared to the CCSD(T)-CBS indicated as (A) in the tables, the RMSD and the MD of the minima locations indicated as (ML) in the tables and RMSD and MD for the minima energies indicated as (MV) that stands for minima values. There are some rare cases when the given model energy curve is repulsive. We simply excluded those dimers in the statistics of the minimum locations and minimum values. All energies are in Kcal/mol and the geometry stats are based on using Angstrom.

The good results

The table below shows our best results using revSCAN, revTPSS and R2SCAN functionals and def2-TZVP, 6-311++G**, 6-311++G*(df,pd) basis sets. Any combination of these basis sets and functionals, regardless of whether we use QF or Grimme's D4 parameters, provide accurate results with only some minor differences here and there. The revSCAN functional with QF parameters seems to be the most accurate one. We have not found optimized D4 parameters yet from Grimme's group for the revSCAN functional. Both QF and Grimme's D4 parameters are available for revTPSS and R2SCAN functionals and can be chosen with a simple command line option in all our QF applications. The default is QF D4.

Overall statistics of accuracy of dispersion corrected DFT (DFT-D4) calculations for S66x8 intermolecular interaction sets using different basis sets, functionals and VDW D4 parametrizations. (A) → All points, (ML) → Minima Locations, (MV) → Minima Values. All energies are in Kcal/mol, distances are in Angstrom.

Functional	Basis Set, Origin of D4 Parameters	RMSD(A)	RMSD(ML)	RMSD(MV)
RevSCAN	**def2-TZVP, QF D4**	**0.414**	**0.0133**	**0.489**
RevSCAN	6-311G++(df,pd), QF D4	0.434	0.0128	0.521
RevSCAN	6-311G++**, QF D4	0.451	0.0140	0.534
RevTPSS	def2-TZVP, QF D4	0.544	0.0114	0.640
RevTPSS	def2-TZVP, Grimme's D4	0.439	0.0130	0.513
RevTPSS	6-311G++(df,pd), QF D4	0.547	0.0125	0.634
RevTPSS	6-311G++(df,pd), Grimme's D4	0.721	0.0150	0.833
RevTPSS	6-311G++**, QF D4	0.558	0.0117	0.656
RevTPSS	6-311G++**, Grimme's D4	0.653	0.0127	0.762
R2SCAN	def2-TZVP, QF D4	0.572	0.0176	0.657
R2SCAN	def2-TZVP, Grimme's D4	0.529	0.0138	0.651
R2SCAN	6-311G++(df,pd), QF D4	0.534	0.0162	0.631
R2SCAN	6-311G++(df,pd), Grimme's D4	0.740	0.0133	0.920
R2SCAN	6-311G++**, QF D4	0.585	0.0177	0.686
R2SCAN	6-311G++**, Grimme's D4	0.666	0.0132	0.828

The bad results

Overall statistics of accuracy of dispersion corrected DFT (DFT-D4) calculations for S66x8 intermolecular interaction sets using different basis sets, functionals and VDW D4 parameterizations. (A) → All points, (ML) → Minima Locations, (MV) → Minima Values. All energies are in Kcal/mol, distances are in Angstrom.

Functional	Basis Set, Origin of D4 Parameters	RMSD(A)	RMSD(ML)	RMSD(MV)
RevTPSS	def2-SVPD, QF D4	0.573	0.0160	0.651
RevTPSS	def2-SVPD, Grimme's D4	1.607	0.0290	1.948
R2SCAN	def2-SVPD, QF D4	0.731	0.0175	0.919
R2SCAN	def2-SVPD, Grimme's D4	1.573	0.0246	1.946
PBE	def2-SVPD, QF D4	0.747	0.0387	0.757
PBE	def2-SVPD, Grimme's D4	1.744	0.0263	2.125

All results are very inaccurate and basically below force filed quality by using D4 parameters from Grimme's group which were optimized using a very large basis set. This statement is true for all functionals that we have tested so far. The QF optimized D4 parameters make the def2-SVPD basis set much more reasonable for DFT-D4 calculations. Having said that the computational costs with the def2-SVPD basis set are very similar, almost the same as with def2-TZVP using QFDFT and the later basis set looks to be more accurate and therefore choosing def2-TZVP or even 6-311++G** basis set is recommended.

The ugly results

Overall statistics of accuracy of dispersion corrected DFT (DFT-D4) calculations for S66x8 intermolecular interaction sets using different basis sets and VDW D4 parametrizations for the ***BP86 functional****. (A) → All points, (ML) → Minima Locations, (MV) → Minima Values.* All energies are in Kcal/mol, distances are in Angstrom.

Basis Set, Origin of D4 Parameters	RMSD(A)	RMSD(ML)	RMSD(MV)
def2-TZVP, QF D4	0.706	0.0244	0.755
def2-TZVP, Grimme's D4	1.606	0.0137	1.989
6-311G++(df,pd), QF D4	0.624	0.0189	0.712
6-311G++(df,pd), Grimme's D4	1.732	0.0171	2.134
6-311G++**, QF D4	0.688	0.0246	0.765
6-311G++**, Grimme's D4	1.664	0.0163	2.055
def2-TZVPPD, Grimme's D4	1.352	0.0121	1.693

The results above clearly indicate that something could be wrong with Grimme's D4 parameter for BP86 functional because regardless of the choice of the basis set, we obtained extremely inaccurate and below force field quality results. After triple checking our implementation, we have contacted Professor Grimme's research group, and we have received the following reply:

"Hi Laszlo,

interesting question. I investigated a bit and found out that the BP functional as implemented in Turbomole is using a different LDA correlation functional than Orca. If I recall correctly I performed the BP calculations with Turbomole back then.

Whether this actually has an impact on the D4 parameters needs checking, I haven't recalculated the BP interactions with Orca yet to redo the fit and see whether this might be the cause. On the other hand it might just be a suboptimal fit for the BP functional with D4.

That's all I have at the moment."

Based on this reply it seems to us that perhaps not the correct BP86 functional has been used during the D4 parameterizations at Professor Grimme's research group. We hope that the situation is not the same for the previous generation D3 parameterizations because almost countless scientific papers, proposals, reports have been using DFT with dispersion corrected BP86 functional over the last decades or so, and having such blow for the accuracy of all such published results would not look good for the community. Note also that the results could be much more accurate by using the same incorrectly implemented BP86 functional which had been used during the parameterizations. Our applications obviously do not support Grimme's D4 parameters for DFT-D4 calculations with the BP86 functional while the QF optimized D4 parameters provides reasonably accurate results. All raw results for all functionals and basis sets that we have tested are tabulated in the appendix.

Performance

The computational performance of QFDFT, especially for typical drug-sized molecules with accurate basis sets, stands out as exceptionally superior compared to NWCHEM, GAMESS-US, and PSI4. To provide a sense of its computational efficiency, I've included a small comparative study below, using progressively larger drug molecules. The table presented below illustrates the computational costs for fully self-consistent energy calculations and analytical atomic force evaluations.

One striking highlight from the table is that QFDFT is more than 43 times faster than both other programs when considering a moderately sized molecule like remdesivir (77 atoms). As molecular size increases, the speedups become even more pronounced due to the improved scaling properties of our algorithms. For larger molecules with over 200 atoms, the calculation becomes over 180 times faster. Importantly, these efficiency gains and speedups are achieved without the need for new hardware architectures like GPUs. Instead, they result from the inherent lower scaling of our algorithm, coupled with efficient programming techniques, all executed on the same Intel-based CPU.

To provide context for our results, it's worth noting that program development in both GAMESS and NWCHEM has been supported for decades through academic and government research grants, as well as direct backing from national laboratories. Achieving such substantial computational speed improvements, while maintaining accuracy on the same architecture as these widely used and developed programs, represents a significant accomplishment. Our software has the potential to open new avenues in computational drug design by enabling faster and more cost-effective development of new medicines and materials.

Computational times (in seconds) for six selected molecules with different molecular sizes using def2-SVPD basis set and TPSS functional. All calculations were performed on a single socket computer using an 18 core Intel i9 processor1.

Number of atoms (Molecule's name, number of Cartesian basis functions)	Energy calculations			Gradients	
	QFDFT[3]	GAMESS	NWCHEM[2]	**QFDFT**	GAMESS
21 (Aspirin, 362)	**17**	129	240	**4**	27
77 (Remdesivir, 1232)	**150**	7213	6534	**28**	1104
139 (Bacteriopheophytin B, 2040)	**324**	18588	42673	**52**	5044
175 (Griselimycin, 2542)	**587**	61490	104402	**110**	9924
196 (Cyclosporin, 2794)	**822**	109263	120779	**166**	13792
245 (Mersacidin, 3801)	**1237**	232893	461391	**252**	31586

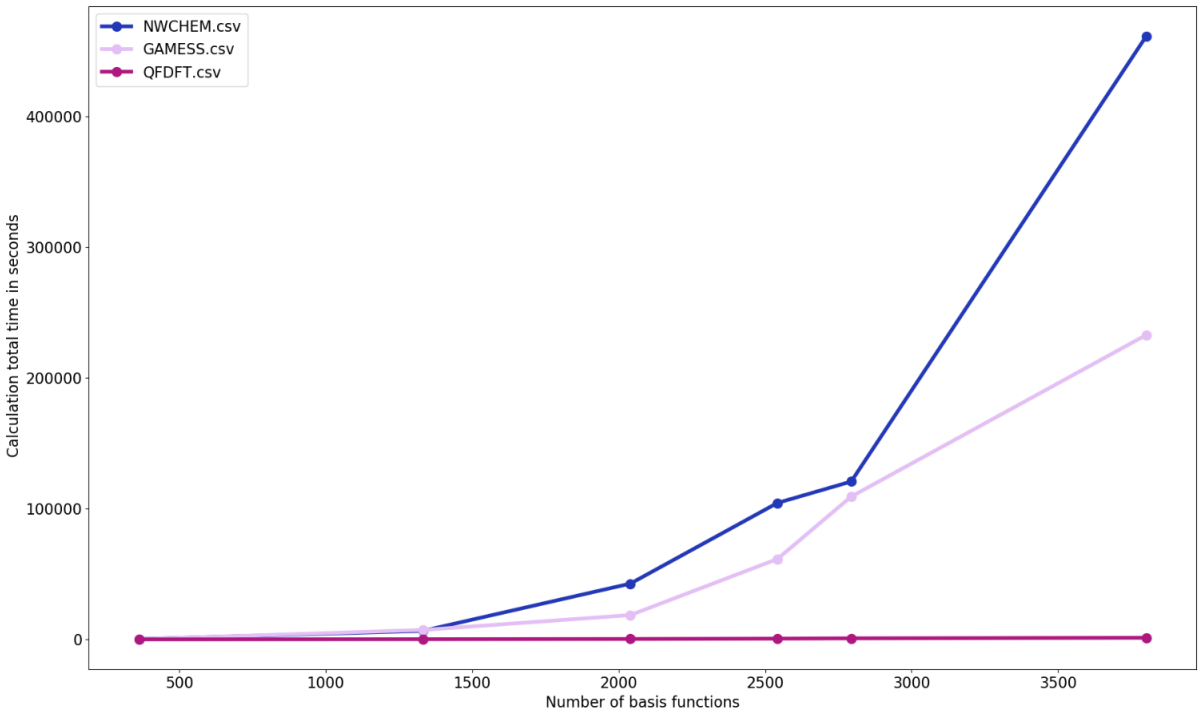
[1]*To be consistent with the accuracy validations the same (75,590) atomic grids were used in both GAMESS and NWCHEM calculations and the same adaptive atomic grids were used in all our QFDFT calculations. Integral thresholds were kept at their default values in all three packages. The "delta Fock" option was turned off in GAMESS calculations because of frequent convergence problems otherwise and QFDFT did not use delta Fock.*

[2]*NWCHEM calculations were not converged completely within 50 SCF cycles for Bacteriopheophytin B and for Mersacidin; the computational timings for 50 iterations are shown. Griselimycin, which is a larger molecule than Bacteriopheophytin B, did converge successfully in 49 SCF iterations. The reason for SCF convergence problems is probably the diffuse basis set.*

[3]*There have been some additional speed up improvements in our software and the results are obtained by using our latest alpha QFDFT version.*

As some say "a picture worth a thousand words" so I plot the calculation costs of the DFT energies from the above table and show it below without any additional comments.

Computational times (in seconds) for six selected molecules with different molecular sizes using def2-SVPD basis set and TPSS functional. All calculations were performed on a single socket computer using an 18 core Intel i9 processor.



Usage

QFDFT is a command line application, and it does not require the preparation of any input file whatsoever. It does not require any learning as to how to use it, so there is no manual detailing its usage with no plans for one. If I must write a manual about how to use the product then I would have failed to achieve my vision of simple and effortless usage for everyone. It has a minimal number of command line options that are necessary. The order of the command line arguments does not matter. Any time when the argument list contains the **–help** keyword, the help will show regardless of any other arguments. The help is short, being only about a few pages now. So, start with the

```
qfdft.x –help
```

command, please read the short help about the current calculation options (2-3 minutes read) and start using it. Please let me know how this ultra-simplistic approach works for you!

This release comes with a few simple python scripts as well. The qfdft.py script does exactly what the qfdft.x does and the only advantage is that this little python script can obviously be expanded or put into an existing python project to make the project capable of using QFDFT. The qfdft_all_sdf.py script performs DFT calculation for all sdf files in the given directory one by one. This is obviously very useful for mass calculations. I will show some very simple shell scripts below with similar functionality, but one advantage of this python script is that one can choose any command line option as command line arguments of this python script while the options are specified in the shell script and cannot be changed without editing the shell scripts.

Examples

This release comes with the directory named QF22BetaExamples. This directory contains a few subdirectories, one is the accuracy tests that I have mentioned above. Another directory is MMFF10 where I have chosen 10 molecules from the MMFF set and performed several different calculation types:

1. **DFT energy calculation,**
2. **DFT energy calculation plus forces,**
3. **Geometry optimizations with semiempirical QM followed by DFT energy calculation,**

4 Geometry optimization with DFT followed by DFT energy calculation,

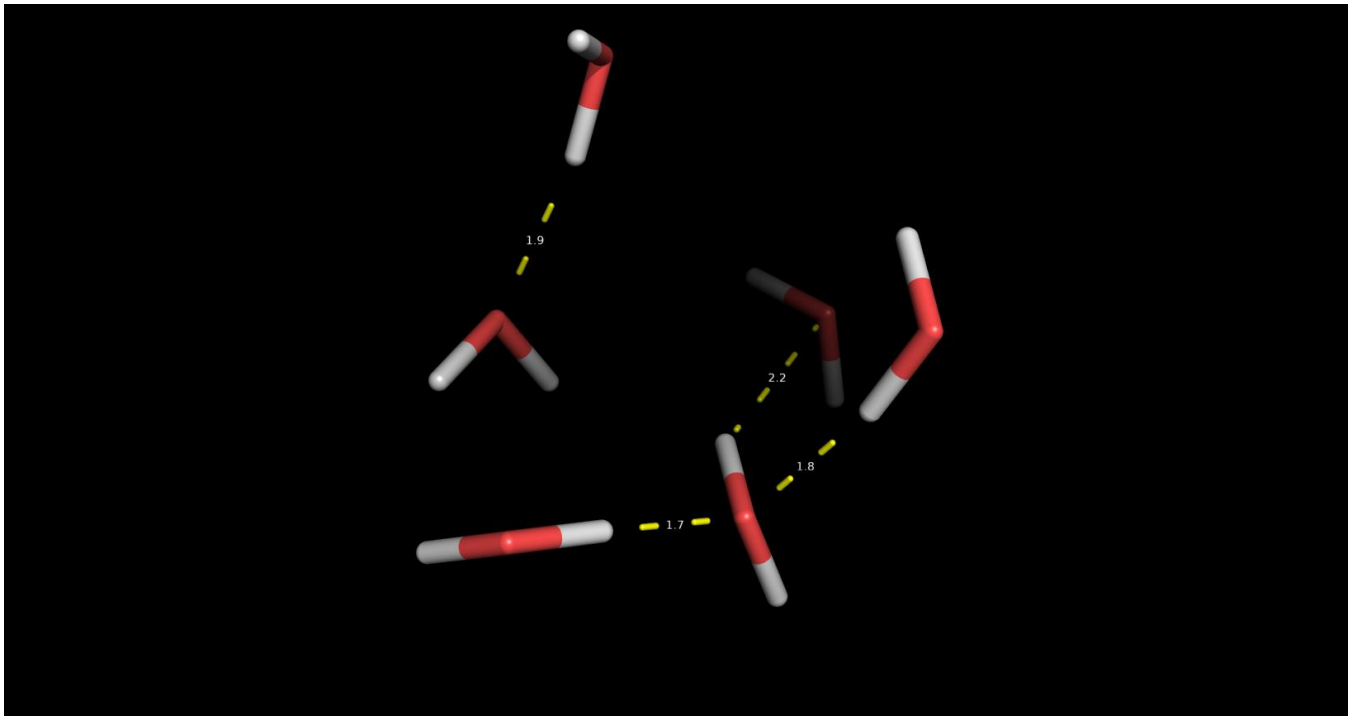
5 Geometry optimizations with semiempirical QM followed by DFT geometry optimization followed by DFT energy calculation

This example also demonstrates how to set the molecular charges from file in case that the input molecular structures are coming from an xyz file where there is no information about the total charge of the molecule. (Note, that if the input files are sdf then the total charge of the molecule is automatically determined from the sdf file.). I provide two ways to do so. One way is using the command line option of **–Charge arg** which has a default 0 value, so it is not needed to use this in the case of a neutral molecule. This scheme is, however, hard to automate when we have many xyz files with different molecules and different total charges. For this reason I provide the **–ChargeFileName arg** second option in order to specify the total charge of the molecule. This file should contain only one signed integer number. Here is the simple shell script:

```
#!/bin/csh -f

foreach calctype (1 2)
  foreach molecule (MMFF94_?.xyz MMFF94_??.xyz)
    set chargefile = `echo $molecule | sed 's/\.xyz/\.chr/'`
    set optfile = `echo $molecule | sed 's/\.xyz/_Optimized\.xyz/'`
    echo "Processing " $molecule " calctype=" $calctype
    qfdft.x --Input $molecule --ChargeFileName $chargefile --CalcType $calctype >&
$molecule.$calctype.log
  end
end

foreach calctype (3 4 5)
  foreach molecule (MMFF94_?.xyz MMFF94_??.xyz)
    set chargefile = `echo $molecule | sed 's/\.xyz/\.chr/'`
    set optfile = `echo $molecule | sed 's/\.xyz/_Optimized\.xyz/'`
    echo "Processing " $molecule " calctype=" $calctype
    qfdft.x --Input $molecule --ChargeFileName $chargefile --CalcType $calctype >&
$molecule.$calctype.log
    mv $optfile $molecule.$calctype.xyz
  end
end

exit
```

Another directory is for testing geometry optimizations for a six-member water cluster. I got the input structure from the GEOMETRIC geometry optimization project, and it is shown below. I have indicated all hydrogen bonds and the distances between the donor and the acceptor atoms.

Input geometry of six-member water cluster



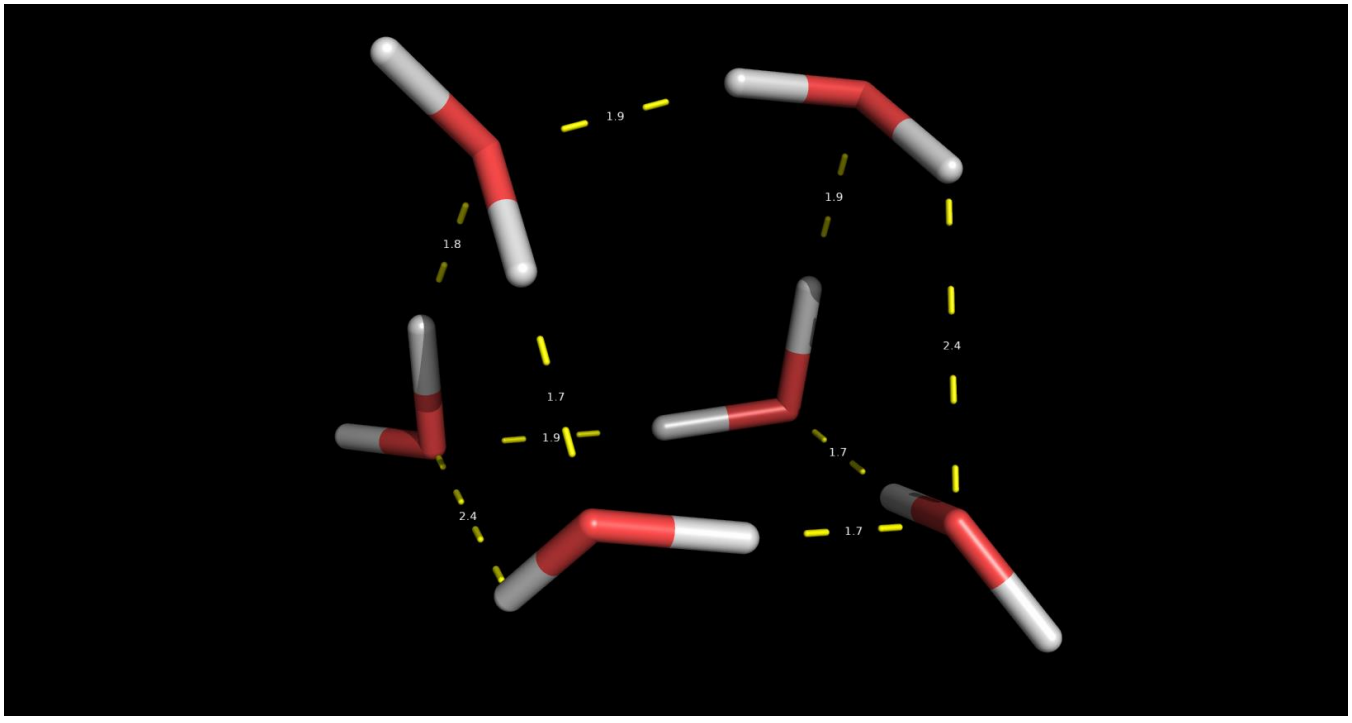
Note that this cluster has a lot of local minima. We are not aiming to determine what is the best structure of this cluster, that calculation would require global optimizations while our geometry optimizations are all local. Our 3 different geometry optimization options are following different paths, and this is a good example to check if their results are reasonable. It is also interesting to check how the solvent effect works, i.e. performing the same geometry optimizations in vacuum and in water. Here are the few lines of shell script to do all of that:

```
#!/bin/csh -f

foreach calctype (3 4 5)
  foreach solvent (vac water)
    echo "Processing solvent=" $solvent " calctype=" $calctype
    qfdft.x --Input  water6.xyz --Solvent $solvent --CalcType $calctype >& water6.$solvent.$calctype.log
    mv water6_Optimized.xyz water6.$solvent.$calctype.xyz
  end
end

exit
```

The results look quite pleasing because all geometry optimization resulted in reasonable structures.

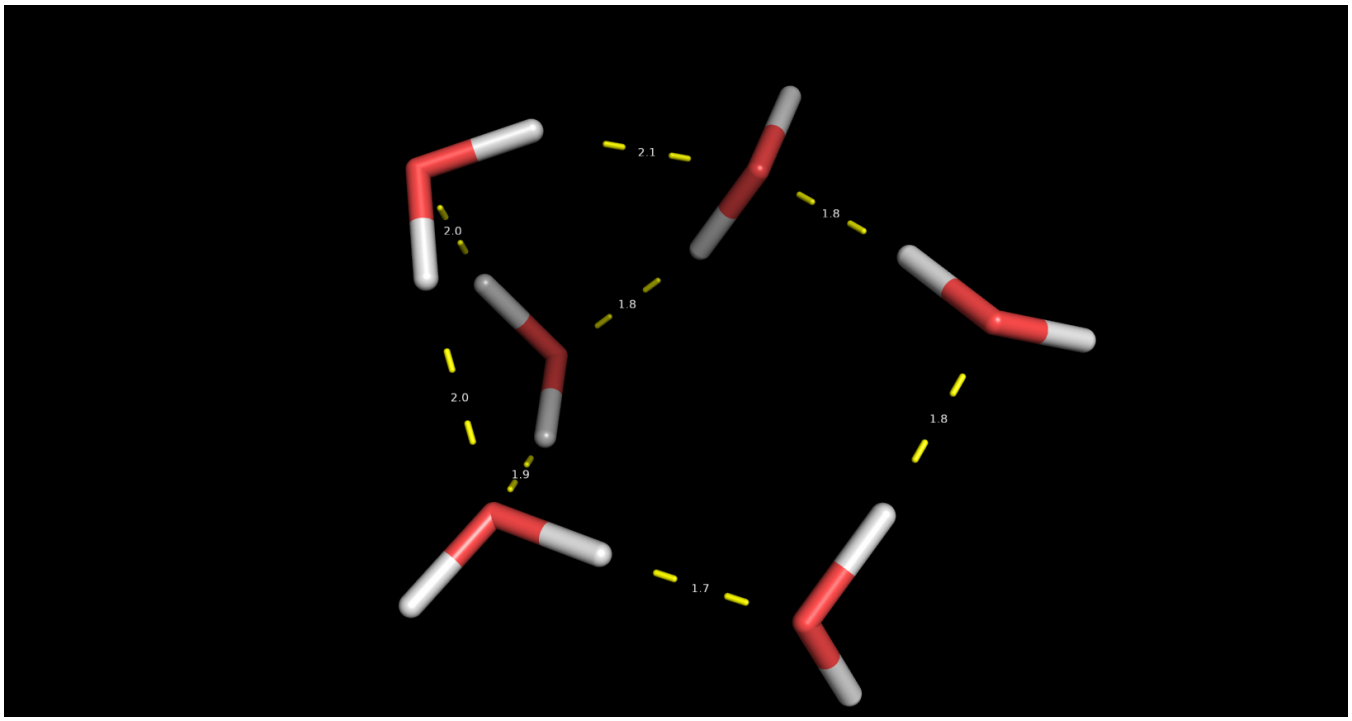
The optimized structure with CalcType=5 in vacuum looks like this:

Geometry optimized structure of six-member water cluster in vacuum (CalcType=5)



There are more hydrogen bonds than in the input structure and there are only 3 hydrogen atoms that are not participating in hydrogen bonds. Interestingly the same geometry optimization in water solution provides a different and still quite reasonable geometry:

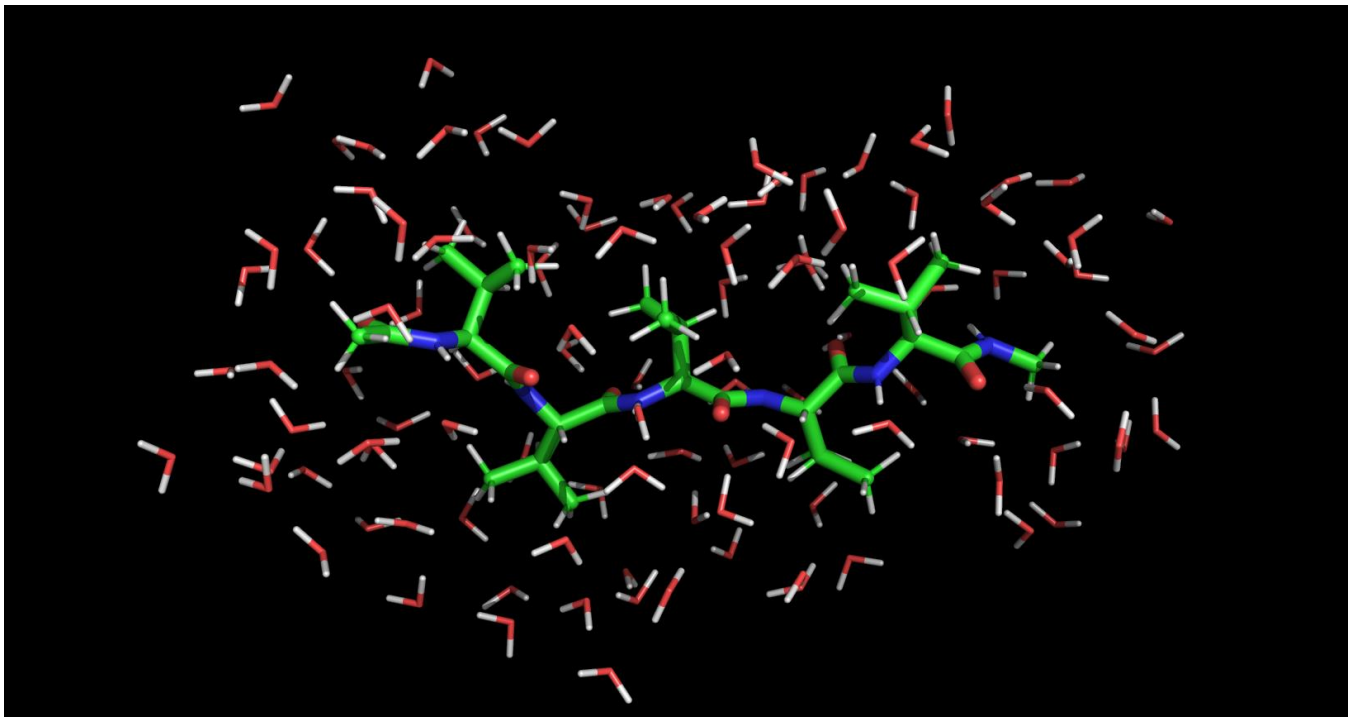
Geometry optimized structure of six-member water cluster in water (CalcType=5)



Interestingly, there are four hydrogen atoms that are not participating in any hydrogen bond in this case.

For our last example I have chosen a hydrated molecule with explicit waters. This molecular cluster was obtained from one of our alpha testers and he is interested in performing very accurate DFT calculations for many such clusters to develop advanced new force fields. The picture of the cluster is shown below.

Hydrated ligand

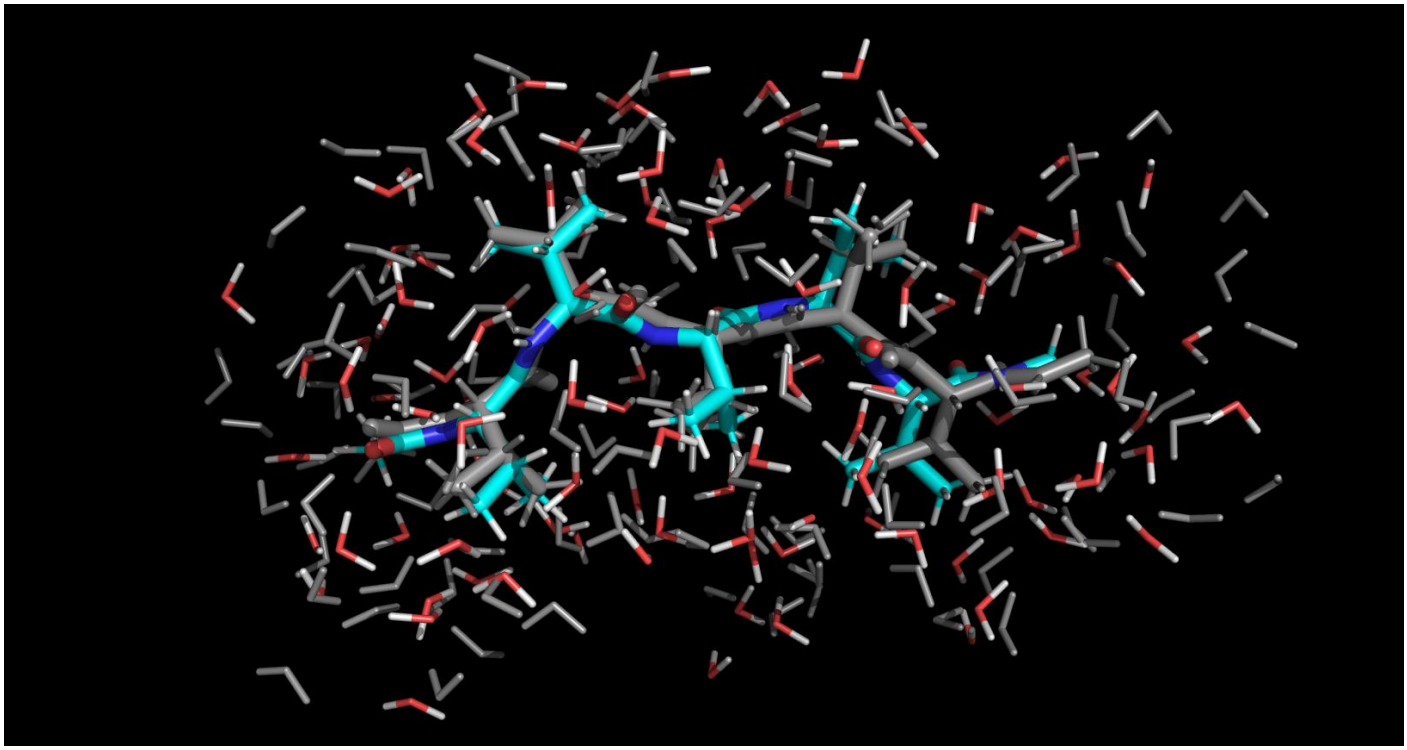


We have performed DFT energy calculations for this cluster using def2-TZVP basis set in vacuum. The calculation cost was about 1000 seconds while using the 6-311G++** default basis set the calculation cost is about 1600 seconds. Either of these basis sets are good choices and provide very accurate results using revTPSS meta functional. Note, that these are quite expensive DFT calculations having 6936 basis functions with def2-TZVP for instance. It is very exciting that we can do such calculations in less than half an hour on a 52 CPU workstation like ours. Since we have a docker container for this distribution it is possible to launch many such calculations using something like docker slurm and finish quite a large project within a day or so using local computer clusters and one can perform practically unlimited such calculations on AZURE by using our qfazurelaunch.x application.

We have repeated the energy calculation with 6-311G++** default basis set in water and using –CalcType 3 option when the geometry was optimized at semiempirical QM level followed by DFT-D4 energy calculation. This job took about 3100 seconds so about half of the time went for geometry optimization at semi-empirical QM level. I found it very interesting that the geometry optimization, again, in water solvent, resulted in about a 568 Kcal/mol lower energy structure with the original structure being a very high energy structure with lots of strain in it. The energy difference between the optimized and the original structure with DFT-D4 in water is even higher, about 615 Kcal/mol. Only less than 10% of the energy gain comes from the solvation contribution. The picture of the original and the optimized cluster is shown

below where I have made the original structure grey. There are some conformation changes in the ligand but the large difference looks to be coming from the rearrangements of the water molecules. All outputs are available in the **HydratedMolecule** directory.

Original and optimized hydrated ligands



Conformation search with qfconfsearchDFT.x

Conformation search represents a cornerstone in computational drug design, spanning a wide spectrum from high-throughput docking to the precise modeling of protein-ligand interactions and even the prediction of organic crystal structures. Although the landscape features a multitude of both commercial and freely available programs for this task, my personal experience underscores the considerable challenge of identifying a conformation search program that strikes the right balance between reliability and robustness.

Typically, mainstream programs prioritize speed, capable of swiftly identifying conformations for drug-sized molecules within seconds to minutes. However, this accelerated pace compromises reliability and robustness, which may not align with the objectives of certain projects. To illustrate, I have been involved in an organic crystal prediction project where, after initially identifying conformations, we expended thousands or even tens of thousands of CPU hours in pursuit of the lowest energy crystal structures, employing these conformations as inputs. This scenario prompts a fundamental question: Is it prudent to employ a program that rapidly identifies conformations, risking the omission of critical ones, and potentially squandering extensive computational resources? Should we not consider an alternative conformation search program that operates at a slower pace, requiring perhaps half an hour or even a few hours to comprehensively identify all significant drug-like molecule conformations, while delivering more dependable results?

In cases such as organic crystal structure prediction and often in molecular dynamics simulations as well, the computational cost of conformation generation becomes inconsequential relative to subsequent modeling steps. Whether it takes seconds or hours to find these conformations, what truly matters is their reliability. Considering this, the goal is to develop a conformation generation code that maximizes robustness. This brings us to the objective of our qfconfsearchDFT.x application, which embarks on its task with thousands of conformations sourced from RDKit (typically 10,000 RDKit conformations by default). The process involves force field-based geometry optimizations, followed by meticulous fine-tuning through quantum mechanical geometry optimizations, employing the xtb semi-empirical method. The journey culminates with DFT energy calculations for the selected QM structures, enabling the reordering of conformations based on accurate DFT-D4 energies.

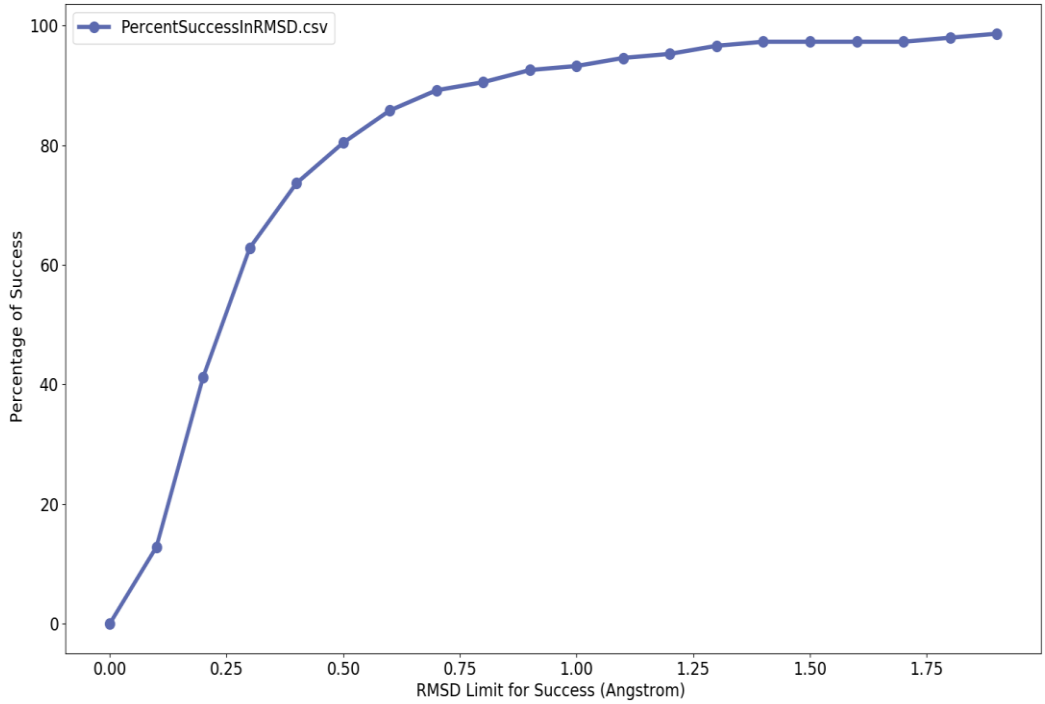
The forthcoming section will feature an in-depth study aimed at validating the performance of the application.

Benchmarking the qfconfsearchDFT.x application

To validate the robustness of our conformation generation code, we conducted an extensive benchmarking exercise utilizing a representative test set comprising 150 FDA-approved drugs. We randomly selected 25 FDA-approved drugs from the Crystallography Open Database (COD) for all six categories defined by the number of rotatable bonds (2, 3, 4, 5, 6, 7). Notably, torsions of methyl groups were excluded in the counts of the rotatable bonds. Employing our innovative qfconfsearchDFT.x program our analysis delved into the frequency of obtaining conformations close to experimental structures, the proximity of our computed conformations to the experimental counterparts, their rankings among generated conformations, and the strain energies. The proximity of molecular conformations to vacuum conformations, measured by the RMSD of non-hydrogen (heavy) atoms, provided a robust benchmark against the highest quality small molecule crystal structures.

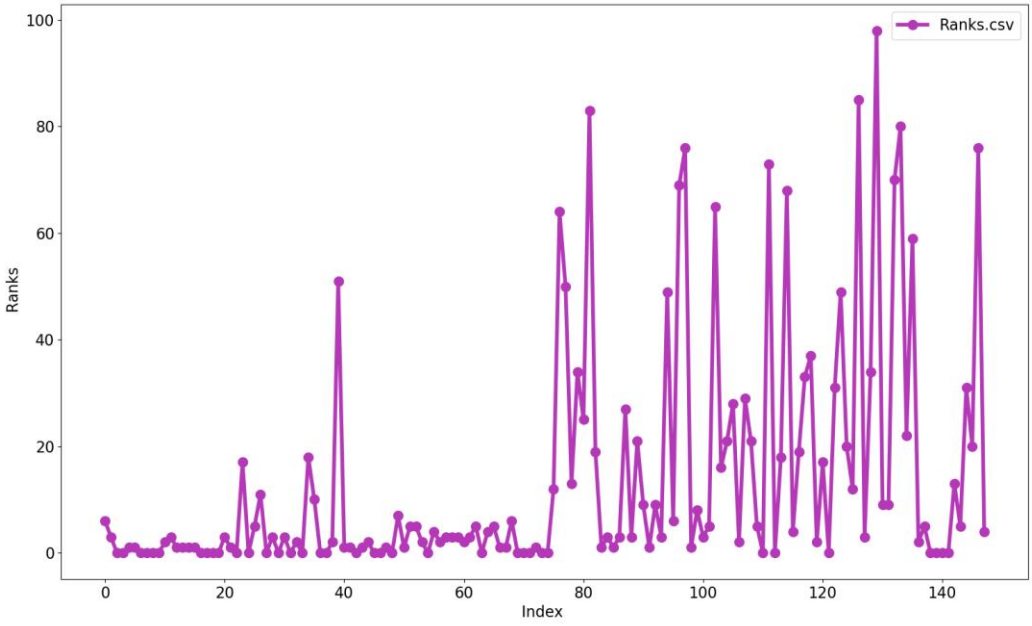
Central to our assessment is the definition of success through heavy atom RMSD values. Success is gauged by the ratio of cases where our found conformations are closer to the experimental structure than a predefined RMSD success limit and then this systematic evaluation allows us to measure the success rate as the function of the RMSD success limits, providing a nuanced understanding of the robustness of our methodology.

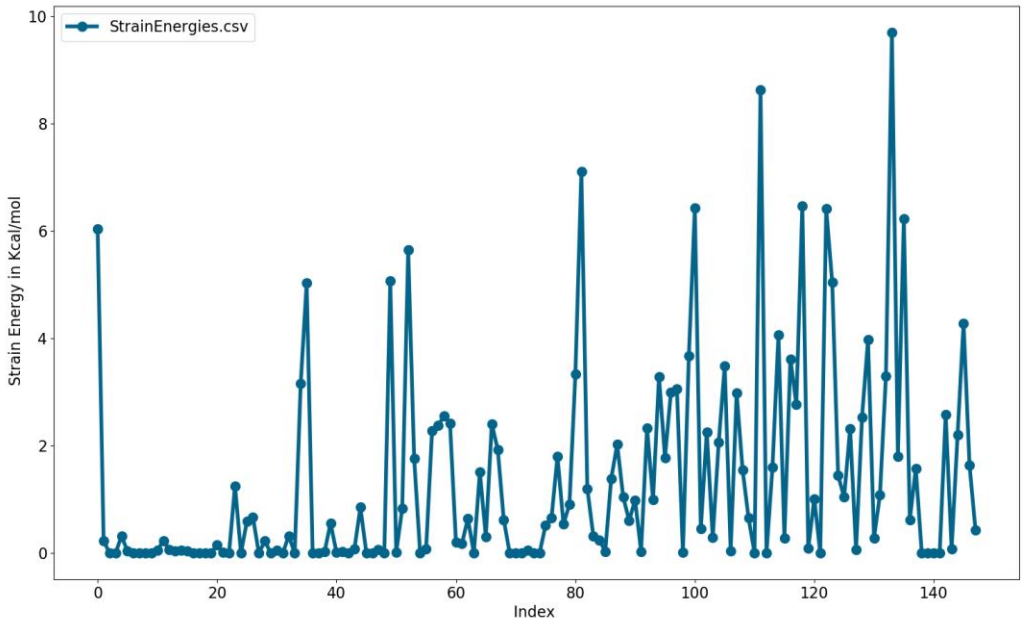
The success rates in percentage are shown in the plot below:



The results show that over 62% of drug molecules exhibit heavy atom RMSD below 0.3 Angstrom, surpassing 80% below 0.5 Angstrom, and an impressive 93% below 1 Angstrom. These figures bear significance when contrasted with traditional cheminformatics and force field-based approaches, often benchmarked the success rates with 1.5 or even at 2 Angstrom RMSD. The pivotal importance of achieving sub 0.5 Angstrom RMSD lies in ensuring accuracy in subsequent predictions, be it in docking, molecular dynamics simulations, or organic crystal structure predictions. Larger difference than 0.5 Angstrom in RMSD almost certainly guarantees to have different heavy atom conformation than the experimental one indicating not as robust scheme in conformation generations as we would like to have which makes the conformation based subsequent projects likely much less accurate.

The DFT-D4 (rev-TPSS, 6-311++G**, QF optimized D4 parameters) energy rankings and the strain energies are shown below.





The index starts with the first 25 drug molecules using 2 rotatable bonds and takes the next 25 batches with 3 rotatable bonds and so on until finishes with the last drug molecule of the batch of 7 rotatable bonds. Unfortunately, there were two drug molecules where the freely available tools were not able to produce valid sdf file from the original cif file and we just dropped those two cases reducing our test set to 148 FDA approved drug structures. Both have 4 rotatable bonds and therefore this category has only 23 instead of 25 molecules.

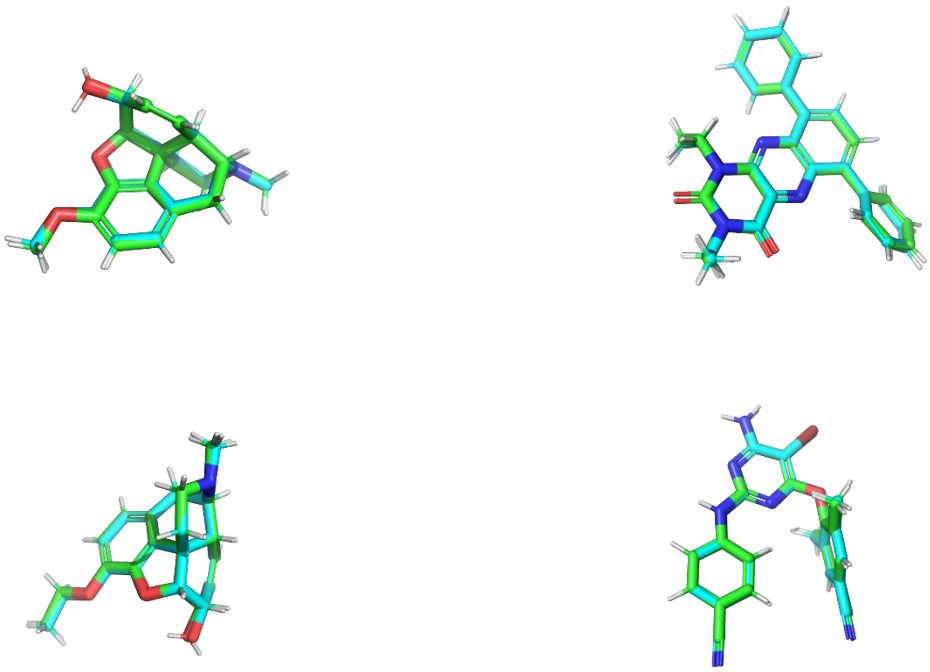
Number of Rotational Bonds	Average RMSD (Angstrom)	Average Rank	Average Strain Energy (Kcal/mol)
2	0.229415	1.64	0.341
3	0.205316	4.72	0.667
4	0.327559	2.43	1.122
5	0.323427	23.24	1.487
6	0.590802	20.24	2.353
7	0.475876	28.24	2.089

Both plots and the table above show that the ranking and the strain energies are usually very low and there are numerous cases when they are exactly zero which means that the lowest energy vacuum conformation is the closest one to the experimental structures. After about 3-4 rotatable bonds the situation gets a bit more complex, and both the rankings and the strain energies are usually larger and more volatile. We found it amazing that even for the drugs with 7 rotatable bonds the lowest energy vacuum conformations are the closest to the experimental structure for numerous examples. Since we know that accurate *ab initio* ranking of the conformations are very important, we have repeated the entire project by using our most accurate revSCAN functional with def2-TZVP basis set.

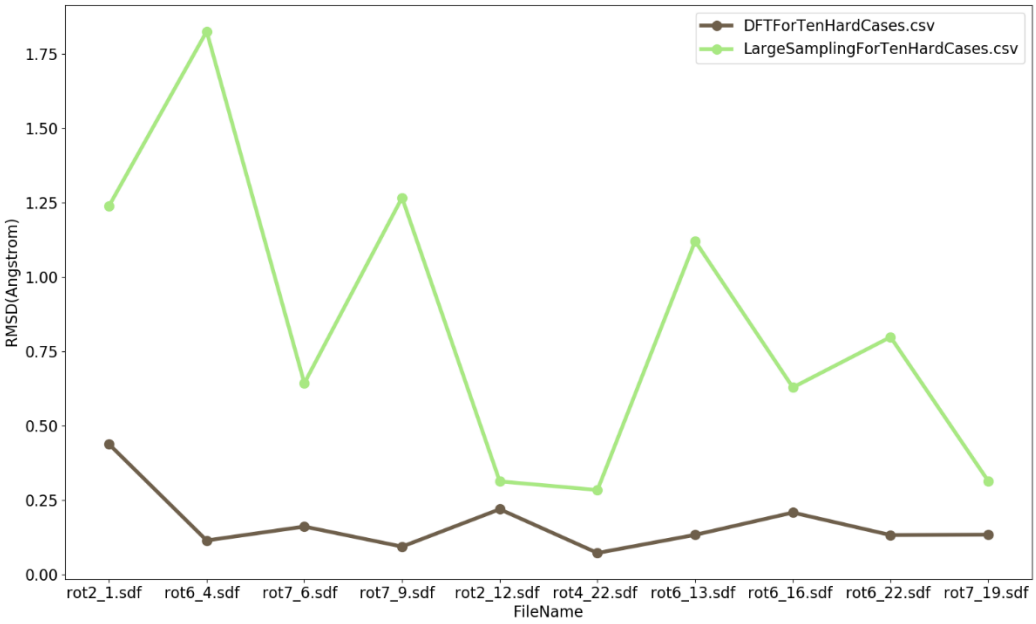
There is no strict rule about that how low the ranking of the vacuum conformations should be with the matching experimental structure but usually lowering the rankings indicates the increase of accuracy at least on average when enough structures are considered. This is exactly what we found here as well. The table below shows the results with revSCAN functional, def2-TZVP basis set and with our new D4 optimized VDW parameters. Perhaps the largest difference is that the average ranking went down from 28 to 22 for the most flexible category having 7 rotatable bonds.

Number of Rotational Bonds	Average RMSD (Angstrom)	Average Rank	Average Strain Energy (Kcal/mol)
2	0.229415	1.40	0.321
3	0.205316	4.12	0.624
4	0.327559	2.22	1.222
5	0.323427	19.96	1.581
6	0.590802	19.36	2.299
7	0.475876	22.00	1.891

A few pictures of amazing overlaps of the experimental and theoretical conformations are show below (just for fun purposes) for Codeine, Ethylmorphine, 1,3-diethyl-6,9-diphenylalloxazine and Etravirine achieving very low 0.0706947, 0.0644524, 0.124212 and 0.118875 Angstrom heavy atom RMSDs.



Acknowledging the pursuit of perfection, we recognize room for improvement in our scheme. Our initial results were achieved using default parameters, with only 10 structures exhibiting RMSD larger than 1.0 Angstrom. Rigorous efforts were invested in increasing sampling size and increasing the number of conformations for reranking conformers with DFT-D4, resulting in some further improvements and bringing the RMSD below 1 Angstrom for 6 out of the 10 problematic cases. We have also made DFT-D4 geometry optimizations for those 10 examples starting the local optimizations from the experimental structures to find out how far one of the nearest DFT-D4 vacuum local minimum geometry is located from the experimental structures. Both sets of results are shown on the plot below.



The plot underscores the remarkable closeness of *ab initio* DFT-D4 vacuum-optimized geometries to experimental structures. While our current scheme balances semi-empirical QM geometries and DFT-D4 re-ranking, the potential for even more accurate geometries and relative energies is apparent through full DFT-D4 geometry optimizations. This solution is a bit too costly on one workstation and therefore it is not available in one shot automatically in our current qfconfsearchDFT application (it can be done with the combination of qfdft and qfconfsearchDFT), however we are putting together a new automatic solution for this purpose on azure soon.

Based on such encouraging results for accurate conformation generations with our software we have decided to build some useful conformational libraries for important drug and drug-like molecules, natural compounds, and fragments. The following section describes the first of such database.

Introducing QF Scientific Database for Drug Repurposing

Part 1: Using FDA Approved Drugs

1. Extensive Conformational Library: Explore low-energy conformations of approximately 1600 FDA-approved drugs. All conformations are ranked with accurate DFT-D4 revSCAN, def2-TZVP *ab initio* calculations.

2. Thermodynamic Insights: Access comprehensive quantum mechanical thermodynamic calculations for each drug molecule's conformations as well as for the conformational ensembles.

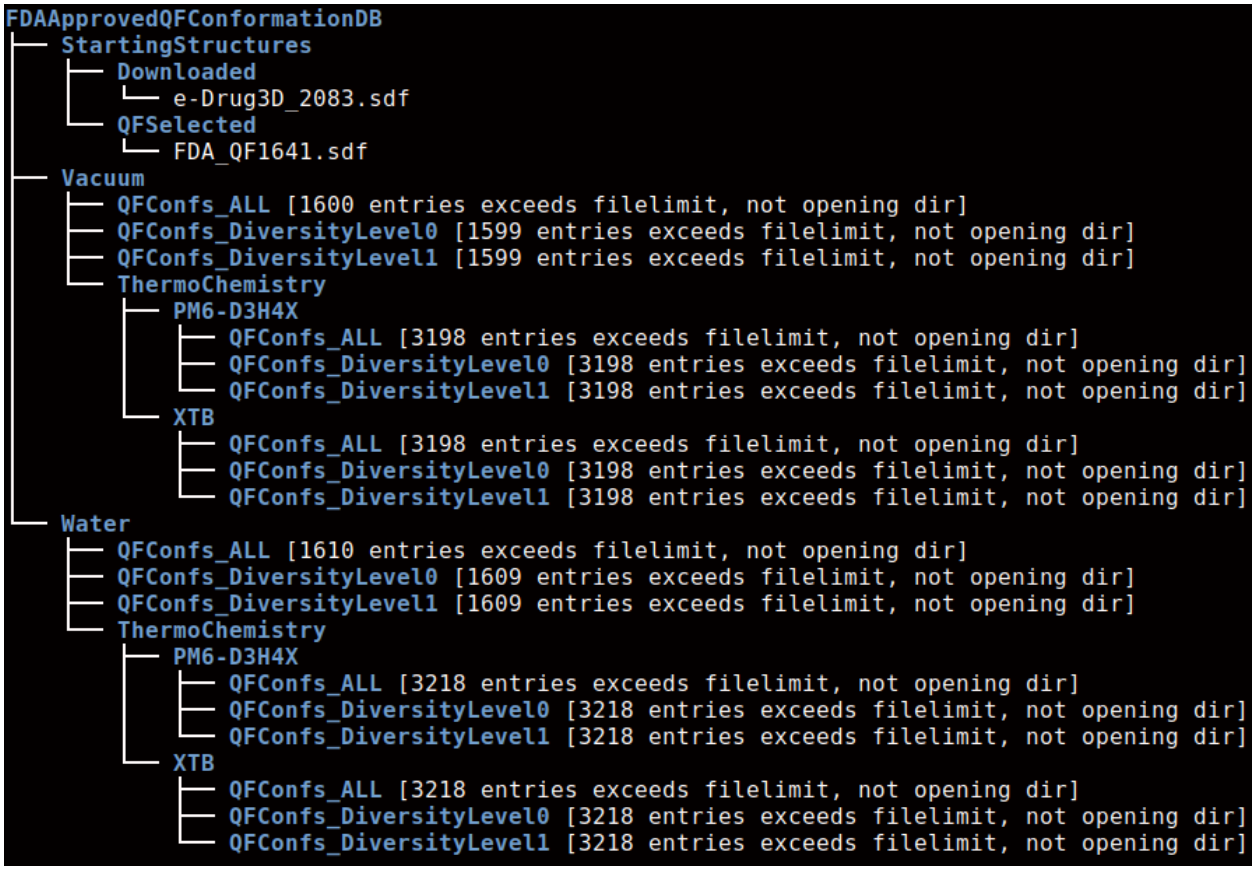
3. Accessibility and Appreciation: Enjoy complimentary access for special QF customers and Microsoft-affiliated research groups, courtesy of Azure credits. A nominal fee is charged for others to help cover our operation expenses.

4. 2024 Software Release: Unlock new drug possibilities with our upcoming software, capable of handling similar projects involving tens of thousands or even hundreds of thousands of private drug-like compounds.

5. Project Outsourcing: For those seeking to outsource similar projects, we offer our expertise and capacity to meet your research needs.

Details of the FDAApprovedQFConformationDB database

The summary of the data structure can be shown conveniently with the tree command under Linux by using the –filelimit option which indicates how many files each directory has. The summary is shown below in the picture and a more detailed explanation follows.



1. **Input data preparation steps with python script using RDKit**

   We have downloaded the 2083 starting structures from https://chemoinfo.ipmc.cnrs.fr/MOLDB/index.php site and applied the so-called Weber filter allowing only maximum 10 rotatable bonds and limit the topological polar surface area values below 140 Angstrom^2. This filter has reduced the number of drug molecules to 1642. We ran our qfLowerLevel.x application to optimize the geometries just to make sure that we have valid quantities in all sdf files. (*This step is not strictly needed for the further steps*.) We have filtered out one more structure due to a sdf problem resulting in 1641 drug molecules. These sdf files were the starting inputs for this project.

2. Finding the conformations with the qfconfsearchDFT.x application

We ran our qfconfsearchDFT.x application with default parameters that utilizes revSCAN functional with the new D4 dispersion correction parameters and def2-TZVP basis set in all DFT-D4 calculations. This functional/basis set combination has been found very accurate in both our intermolecular interaction benchmarks using S66x8 standard set as well as in our study to compare calculated vacuum conformations to high quality experimental geometries. For more details about those benchmark studies please look at our corresponding articles on LinkedIn. We have selected 1600 successful results in vacuum and 1610 results in aqueous continuum. The relatively small number of the rest of the calculations have shown some unwanted behaviors where the original bonds from the original sdf file have been changed during the QM semi-empirical geometry optimizations and we did not want to allow such drastic change in the structures since we are looking for conformations of the same molecules in this project and not looking to investigate tautomers or chemical reactions.

The QFConfs_ALL directories have all the results with 1600 sdf files both in vacuum and 1610 sdf files in water continuum. All sdf files have all important conformations that we had found with our application and the DFT-D4 (revSCAN, def2-TZVP) energies as well as the relative energies compared to the lowest energy conformation (strain energies) are attached as standard sdf tags. Note, that the numbering in the file names have no meaning whatsoever, it was just the order of the files that we used during the calculations. The sdf file for a given drug can be located by grep the corresponding name or the CAS number.

3. Further deduplications of the conformations using the qfdeduplicate.x application

We have introduced two new directories under the name of QFConfs_DiversityLevel0 and the QFConfs_DiversityLevel1. Data files in those directories require some explanation. The qfconfsearchDFT.x application uses RMSD pair wise alignment-based deduplications via the C++ APIs of RDKit only for molecules with no more than 40 atoms by default. The problem is that we want to include hydrogen atoms during the RMSD alignments and deduplications because we do not want to lose important conformations where the orientation of the hydrogen atoms is different. This makes the RMSD calculations very expensive scaling very poorly in molecular size. We have some test examples showing that the RMSD based deduplications are more expensive than all *ab initio* DFT-D4 calculations. Therefore, we limit the RMSD based deduplications up to 40 atoms and we have developed an alternative scheme for larger molecules. This alternative scheme is new and very experimental at this point and with the current parameters it is on the conservative side which means that sometimes it keeps structures which are very close to each other. This conservative approach is obviously preferable since we can always eliminate some duplicates later much easier than finding missing conformations that we had eliminated by mistake because that would require repeating some expensive calculations.

Thus, the QF_DiversityLevel0 and the QF_DiversityLevel1 directories have such post processing deduplicated conformations. We used 0.3 Angsrom RMSD limit in the QF_DiversityLevel0 directory and 0.6 Angstrom RMSD in the QF_DiversityLevel1 directory. Note, that if the energies of two conformations differ more than 0.2 Kcal/mol than we keep both regardless of the aligned RMSD difference in the geometries. Obviously, further deduplications or different deduplications starting from the conformations

in the QFConfs_ALL directories are also possible, and users could do that easily by themselves. Unfortunatley one structure failed to be processed so we have 1599 instead of 1600 and 1609 instead of 1610 structures for vacuum and in aqueous continuum respectively.

4. Going beyond *ab initio* DFT-D4 energies and strain energies by using qfLowerLevel.x and qfensemble.x applications.

Besides accurate *ab initio* DFT-D4 energy orders and strain energies of the conformations, this database provides additional quantum mechanical information. We have performed statistical thermodynamic calculations with the GFN-XTB and the PM6-D3H4X quantum mechanical semi-empirical methods for all conformations of all drug molecules by using our qfLowerLevel.x application. Having all necessary thermochemistry quantities for all conformations we have also calculated the Gibbs free energies for the conformational ensemble of all drug molecules using our qfensamble.x program. The Gibbs free energies for the conformational ensembles are written to individual log file for each sdf files by adding a simple .log file extension to each sdf files since this value belongs to the whole conformational ensemble of the given drug molecules. For this reason, the number of files is doubled in all directories of thermochemistry calculations.

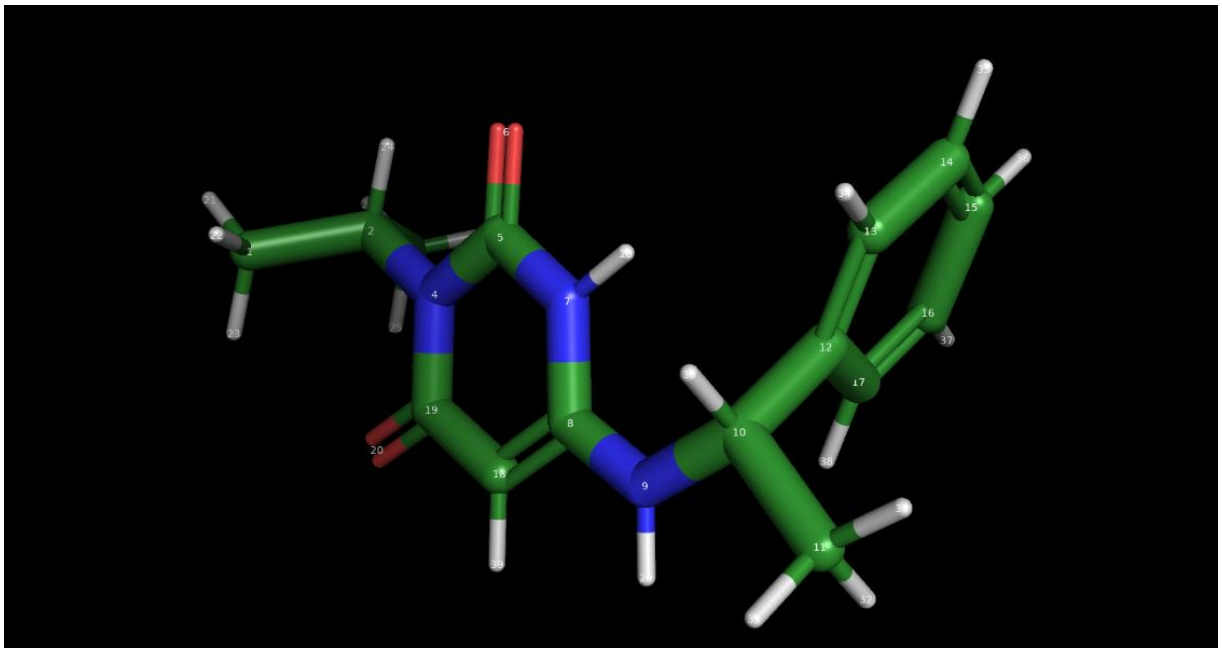
Torsion scan with qftorsionscan.x

Another significant technique closely related to conformation search and widely employed in the drug design community is torsion scans. The qftorsionscan.x application takes four atomic indices that define a specific rotatable bond as inputs. Its primary objective is to identify the lowest energy curve along that rotatable bond. This is achieved through molecular geometry optimizations with constraints applied to numerous potential conformations, ultimately selecting the lowest energy values at each torsion scan point.

While there is an option to utilize force fields such as MMFF and UFF for preliminary scans, it's important to note that the default behavior is to employ QM semi-empirical level, and this is highly recommended based on our extensive experience. Force fields-based torsion scan energy profiles often yield highly inaccurate results, with energy barriers differing significantly from the corresponding QM values. Additionally, the locations of minima can sometimes be entirely different, rendering the results unreliable in our view. We do provide the option to use MMFF and UFF force fields for torsion scans, primarily to allow users to conduct their own experiments and observe the limitations of these approaches.

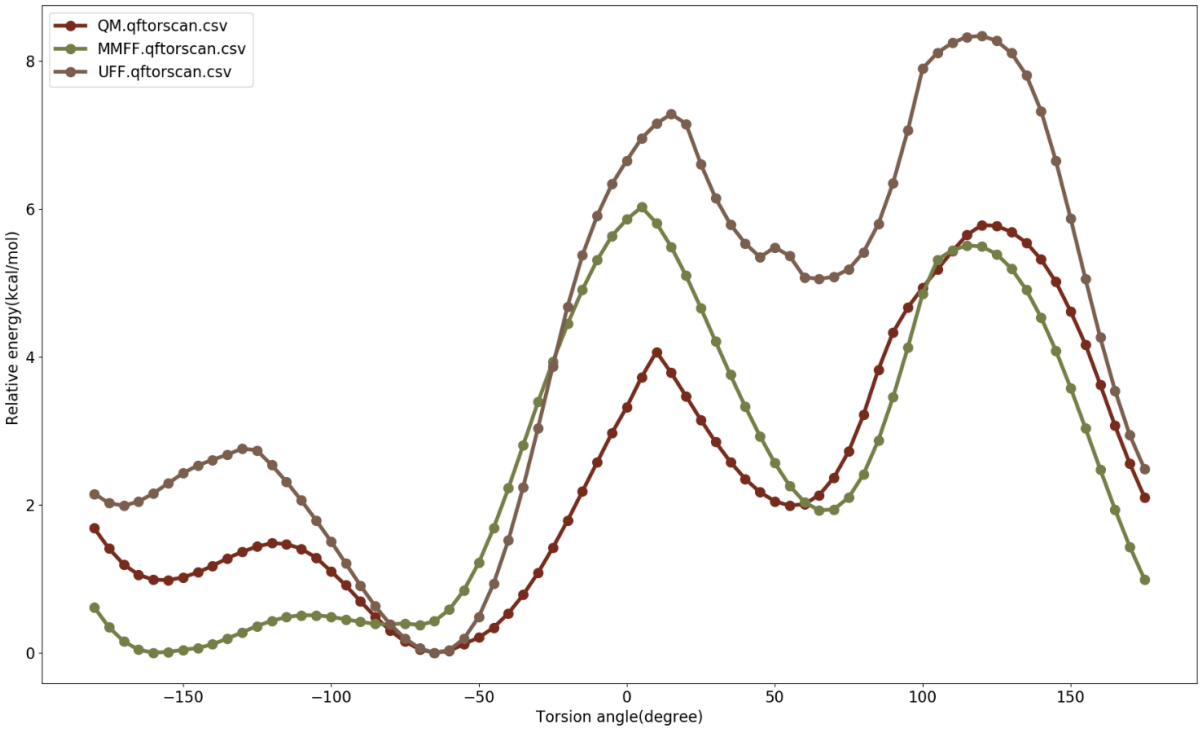
For example, a torsion scan illustration is provided below for Mavacamten along the N-C rotation defined by atomic indexes 8, 9, 10, and 12. In this instance, the MMFF and UFF curves, while not ideal, exhibit some degree of similarity with the QM curve.

Mavacamten with atomic indexes



Torsion scans for Mavacamten using QM semiempirical method, MMFF and UFF force fields.

The N-C rotatable bond is defined by the dihedral angle with the 8,9,10,12 atomic indexes.



Running large project on Azure with qfazurelaunch.x application

This program is designed to facilitate the submission of QF calculations, including a potentially large number of calculations on Azure, with a single command. It currently supports three QF applications: **qfdft.x**, **qfconfsearchDFT.x**, and **qfLowerLevel.x**. Users must specify the desired options for the **qfazurelaunch.x** application, choose one of the three QF applications, provide the corresponding command line options, and specify the project's name. While this may seem complex initially, it's straightforward because default values are available for all required command line options.

Let's illustrate how easy it is to launch thousands or tens of thousands of DFT-D4 energy calculations using the default settings for dispersion-corrected revSCAN functional and def2-TZVP basis set with this simple command:

```
qfazurelaunch.x --azureRegion westus2 --azureInstanceType Standard_F16s_v2 --azureMaxSpotInstances 100 --azureProjectName MyQFProject qfdft.x
```

In this example, we utilize a maximum of 100 16 vCPU nodes in the azure batch pool.

For another example, launching a large project for accurate QM-based conformation searches with a similar command:

```
qfazurelaunch.x --azureRegion westus2 --azureInstanceType Standard_F16s_v2 --azureMaxSpotInstances 100 --azureProjectName MyQFProject qfconfsearchDFT.x
```

Here, default values are used for all **qfdft.x** and **qfconfsearchDFT.x** command line options. Users can choose non-default values in the same way they would on a local Linux node without involving Azure cloud calculations.

Azure Project Name: Structural input files must be tarred and gzip-compressed into ProjectName.tar.gz. These files must be valid structural inputs accepted by the specified QF application. **qfdft.x** accepts xyz

and sdf inputs, while **qfconfsearchDFT.x** and **qfLowerLevel.x** accept sdf and smi inputs. If providing xyz input files for qfdft.x, a charge file with the same filename and a .chg extension must also be provided, containing a single integer to define the molecule's total charge. The QuantumFuture.lic file also needs to be in the same working directory together with the ProjectName.tar.gz file. The remaining Azure command line options are self-explanatory and require no further explanation.

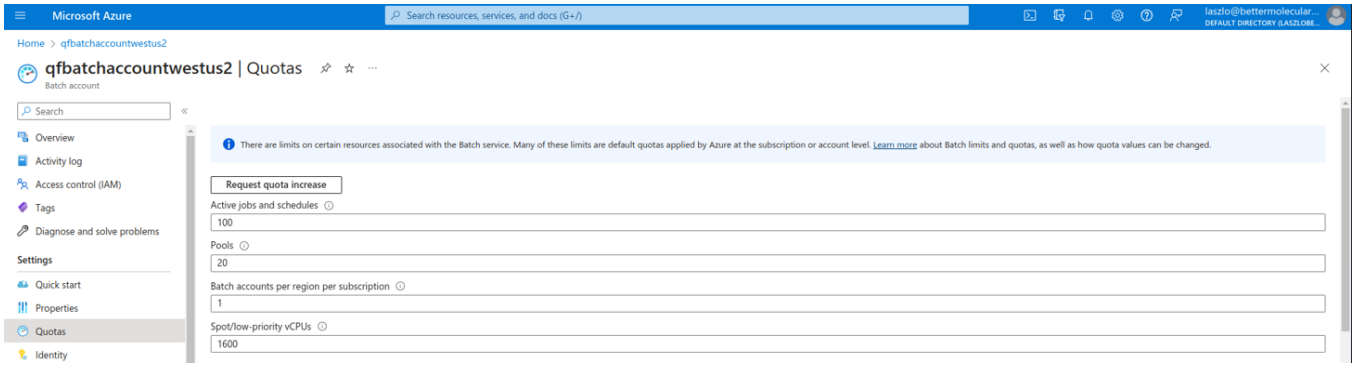
Azure Requirements: All calculations utilize the user's own Azure account, which must be configured as follows. For all supported Azure regions, the user wishes to use, a resource group and a batch account

az group create --name qfbatchresourcewestus2 --location westus2 --tag create-account

az batch account create --resource-group qfbatchresourcewestus2 --name qfbatchaccountwestus2 --location westus2

must be established in that resource group. All calculations will utilize this batch account, so it should be created once, following Azure team's quota limits, and never deleted. The setup of resource groups and batch accounts cannot be automated because the increase of the quota requires assistance from Azure customer service. However, creating them using Azure CLI is straightforward:

After setting up resource groups and batch accounts, users must contact Azure customer service to increase the quota for Spot/low-priority vCPUs to maximize the use of computational nodes. The image below illustrates my setup with a quota of 1600 vCPUs for Spot/low-priority vCPUs.



The following names must be used for resource groups and batch accounts (note the misspellings; use exactly as listed below):

Azure Region	Resource Group Name	Batch Account Name
westus	qfbatchresourcewestus	qfbatchaccountwestus
westus2	qfbatchresourcewestus2	qfbatchaccountwestus2
westus3	qfbatchresourcewestus3	qfbatchaccountwestus3
centralus	qfbatchresourcescus	qfbatchaccountscus
eastus	qfbatchresourceestus	qfbatchaccounteastus
eastus2	qfbatchresourceestus2	qfbatchaccounteastus2

All users utilize their own Azure accounts, and here are the steps for the project:

1. User provides credentials. On a Linux node, it can be as simple as typing the "az login" command and logging in interactively using the default browser (recommended to use Chrome). After that, the program should work. If the user wants to use our Docker container, they need to set up a so-called Service Principal under their account first to be able to log in non-interactively. For example,

   ```
   az ad sp create-for-rbac --name ServicePrincipalWestus3 --role Contributor --scopes
   /subscriptions/yoursubscription/resourceGroups/qfbatchresourcewestus3
   ```

   with the CLI command:

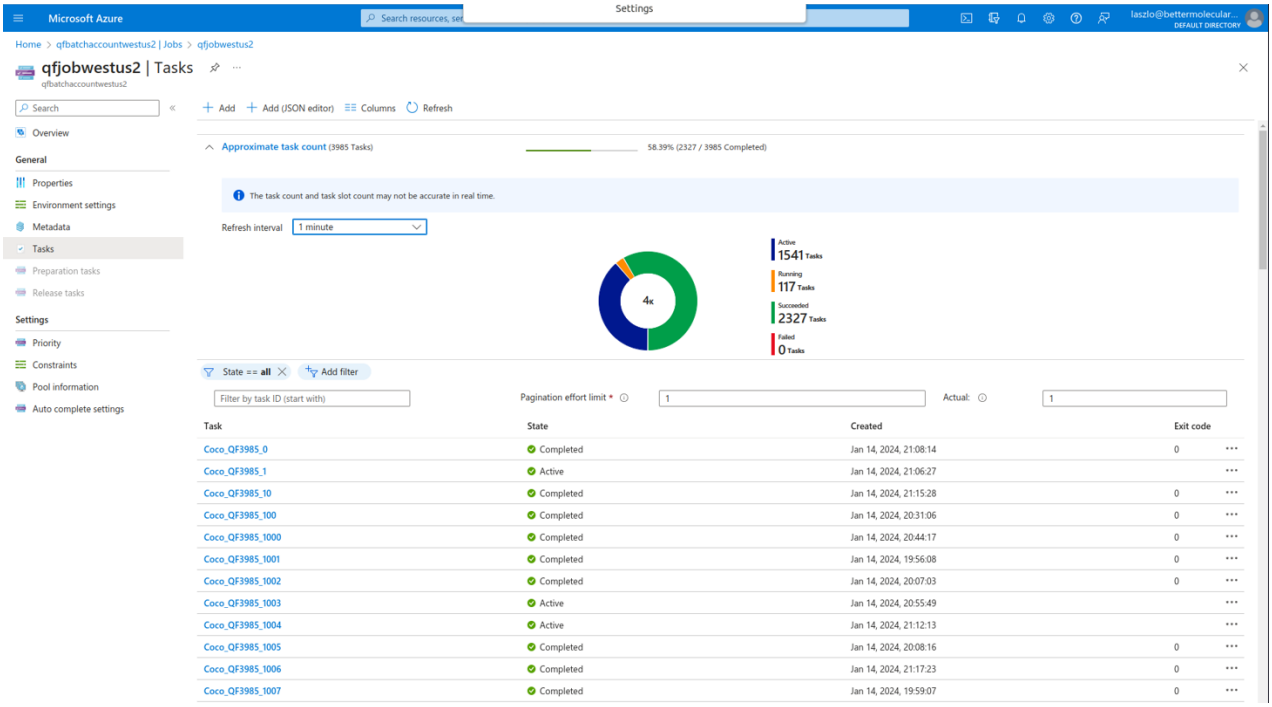
   The output will contain secrets that the users should keep for themselves. An example is:

   ```
   {
     "appId": "some numbers and letters here",
     "displayName":"ServicePrincipalWestus3",
     "password": "more numbers and letters here",
     "tenant": "another set of letters and numbers here"
   }
   ```

2. After successful login, the qfazurelaunch.x application will perform the following steps (there are some flexibilities in the order of the steps, so it is not exactly as listed below):
    a. Download the QF Azure package to the local computer.
    b. Untar the input files.
    c. Create encrypted storage on Azure Blob.
    d. Create user identities.
    e. Upload all QF Azure components and input files to Azure Blob Storage.
    f. Upload some shell scripts.
    g. Create an Azure Batch pool.
    h. Create an Azure Batch job.
    i. Create and submit all Azure tasks (individual calculations in Azure).

    Note that this process may take considerable time for larger projects. Please be patient.

3. The Azure Batch service takes over from here. It scales up the pool based on the number of submitted tasks and the maximum limit of spot instances chosen by users (within the vCPU quota). The computational instances then execute all the tasks. Progress can be conveniently monitored on the Azure portal. An example is shown below with our project for **qfconfsearchDFT.x**, involving 3985 natural compounds. Once the calculations are completed, the Azure Batch service scales down the pool to zero, stopping users from incurring charges for computational nodes. The only remaining cost is for storage. Users must delete both the job and the pool on the Azure portal before starting another project on the same batch account!



4. Users can download the results to their local computer, and the blob storage used for the calculation can be deleted afterward. We provide the **qfgetresultsfromazure.x** application for downloading results and the **qfdeleteazurestorage.x** application to delete the temporary Azure Blob storage used for the project.

Another important topic is the cost of the calculations. The QF license cost is highly affordable, and we offer substantial discounts for large projects. The cost of computational nodes on Azure is minimized by exclusively utilizing spot/low-priority instances, which come with up to a 90% discount compared to on-demand prices. This significantly reduces the overall project cost. Using spot instances, however, comes with some challenges. These cost-effective nodes can be preempted and are typically reclaimed quite frequently, interrupting user calculations. The Azure Batch service effectively manages these interruptions by restarting the same calculations on new nodes when Azure capacity allows. It's important to note that the restarted calculations occur on completely new instances, not the previous ones. For relatively fast calculations, interruption is generally not problematic, as the Batch service keeps track and resumes interrupted calculations on new instances. However, repeating expensive calculations from scratch due to frequent interruptions could lead to higher costs than using on-demand instances, making the project slower or impractical. To address this, we have developed a special restart capability in all three of our supported applications. This technology, combined with our unique shell scripts used on Azure, allows interrupted expensive calculations to resume with minimal repetition once new spot instances become available. This ensures that even expensive jobs can be successfully completed while optimizing the cost of calculations.

QFGUICalculationsLauncher.x

Our first GUI is to perform calculations on a local Linux node without typing any command.

This initial version of our graphical user interface (GUI) supports the qfdft.x, qfconfsearchDFT.x, qfLowerLevel.x, and qftorsionscan.x applications. Users can conveniently select the supported command-line options for each application via the user interface, enabling them to perform calculations as if using commands on a Linux node.

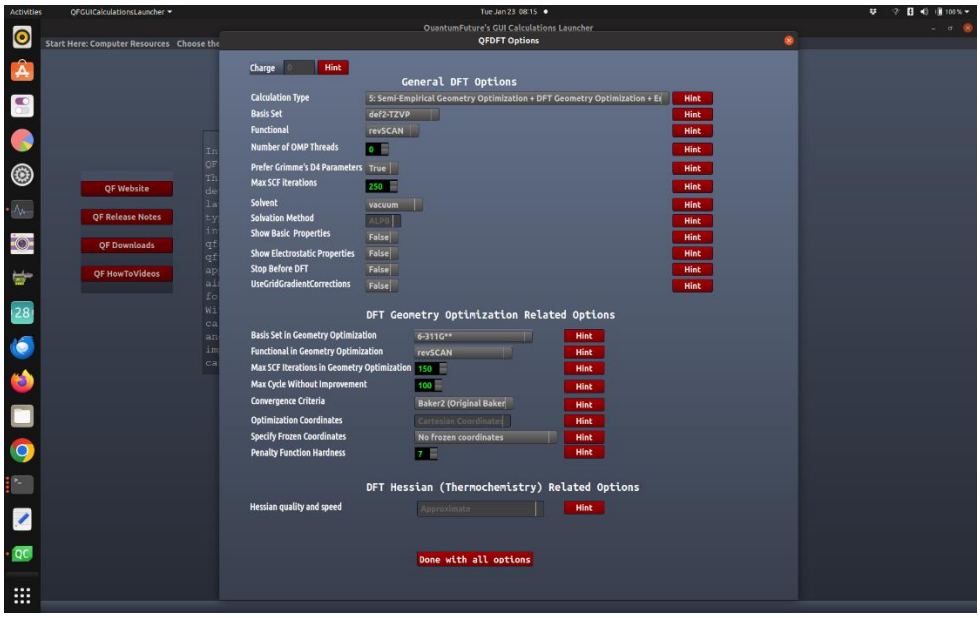
In our upcoming release, likely still in 2024, we plan to provide a similar GUI for launching calculations on Azure. This next version may come in the form of a desktop GUI application, compatible with Linux, Windows, and MAC, or as a unified browser-based solution created with Qt and WebAssembly.

We believe that this GUI is intuitive and straightforward to use, minimizing the need for extensive documentation. To provide you with a visual preview of the interface's functionality, we have included a few screenshots below.

If you have any more text that needs improvement or any other questions, feel free to ask!

Starting page with some introduction

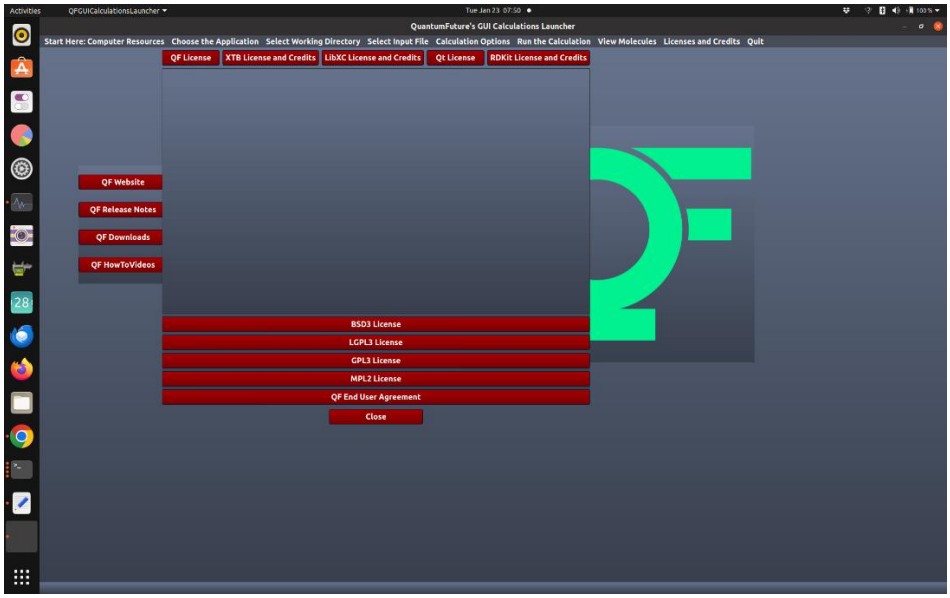
Selecting qfdft.x options



Adding frozen coordinates

Choosing options for qfconfsearchDFT.x

Credits for third party codes



Appendix A: Some additional benchmark data

All tables below: *Overall statistics of accuracy of dispersion corrected DFT (DFT-D4) calculations for S66x8 intermolecular interaction sets using different basis sets, functionals and VDW D4 parametrizations except the revM06_L functional which does not use any VDW corrections.*

(A)→ *All Energy Points,*

(ML)→ Minima Locations,

(MV)→ Minima Energy Values

All energies are in Kcal/mol, distances are in Angstrom.

PBE (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G QF D4**	**1.521**	**0.0437**	**1.730**	**-0.514**	**0.0195**	**-0.664**
6-311G++ QF D4**	**0.682**	**0.0185**	**0.816**	**-0.397**	**0.000716**	**-0.488**
6-311G(df,pd) QF D4	**1.554**	**0.0453**	**1.749**	**-0.456**	**0.0201**	**-0.555**
6-311G++(df,pd) QF D4	**0.663**	**0.0194**	**0.771**	**-0.350**	**0.00347**	**-0.425**
6-311G Grimme D4**	**1.895**	**0.0222**	**2.286**	**-1.454**	**-0.0152**	**-1.927**
6-311G++ Grimme D4**	**0.818**	**0.0162**	**0.985**	**-0.649**	**-0.00270**	**-0.833**
6-311G(df,pd) Grimme D4	**1.932**	**0.0241**	**2.323**	**-1.501**	**-0.0184**	**-1.984**
6-311G++(df,pd) Grimme D4	**0.843**	**0.0157**	**1.010**	**-0.697**	**-0.00521**	**-0.892**

PBE (non-Pople basis sets)		RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP	QF D4	1.938	0.0446	2.266	-0.783	0.0139	-1.042
def2-SVPD	QF D4	0.747	0.0387	0.757	-0.225	0.0194	-0.263
def2-TZVP	QF D4	0.667	0.0197	0.790	-0.351	0.00200	-0.443
def2-SVP	Grimme D4	2.415	0.0290	2.890	-1.805	-0.0223	-2.403
def2-SVPD	Grimme D4	1.744	0.0263	2.125	-1.490	-0.0225	-2.012
def2-TZVP	Grimme D4	0.737	0.0178	0.874	-0.537	0.00102	-0.696
PC2	Grimme D4	0.566	0.0186	0.665	-0.364	0.00341	-0.485
def2-TZVPPD	Grimme D4	0.416	0.0184	0.471	-0.248	0.00471	-0.330

TPSS (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G** QF D4	1.296	0.0371	1.512	-0.600	0.0130	-0.726
6-311G++** QF D4	0.759	0.0175	0.894	-0.599	-0.00208	-0.727
6-311G(df,pd) QF D4	1.304	0.0333	1.534	-0.643	0.00889	-0.778
6-311G++(df,pd) QF D4	0.726	0.0169	0.842	-0.581	-0.00289	-0.696
6-311G** Grimme D4	1.621	0.0239	1.941	-1.254	-0.0145	-1.626
6-311G++** Grimme D4	0.745	0.0203	0.869	-0.565	-0.00170	-0.673
6-311G(df,pd) Grimme D4	1.660	0.0262	1.998	-1.297	-0.0186	-1.700
6-311G++(df,pd) Grimme D4	0.799	0.0193	0.920	-0.614	-0.00503	-0.736

TPSS (non-Pople basis sets)		RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP	QF D4	1.628	0.0611	1.862	-0.440	0.0279	-0.575
def2-SVPD	QF D4	0.734	0.0257	0.845	-0.520	0.00302	-0.619
def2-TZVP	QF D4	0.671	0.0191	0.795	-0.495	-0.000471	-0.610
def2-SVP	Grimme D4	2.088	0.0308	2.551	-1.553	-0.0185	-2.091
def2-SVPD	Grimme D4	1.663	0.0274	1.992	-1.337	-0.0226	-1.773
def2-TZVP	Grimme D4	0.580	0.0256	0.668	-0.378	0.00550	-0.440
PC2	Grimme D4	0.441	0.0261	0.498	-0.199	0.00756	-0.219
def2-TZVPPD	Grimme D4	0.365	0.0262	0.395	-0.0992	0.00960	-0.0884

revTPSS (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G** QF D4	1.161	0.0216	1.376	-0.506	0.00503	-0.592
6-311G++** QF D4	0.558	0.0117	0.656	-0.423	-0.00302	-0.501
6-311G(df,pd) QF D4	1.130	0.0196	1.347	-0.531	0.000650	-0.632
6-311G++(df,pd) QF D4	0.547	0.0125	0.634	-0.433	-0.00713	-0.516
6-311G** Grimme D4	1.501	0.0228	1.802	-1.178	-0.0195	-1.555
6-311G++** Grimme D4	0.653	0.0127	0.762	-0.500	-0.00759	-0.607
6-311G(df,pd) Grimme D4	1.536	0.0261	1.851	-1.216	-0.0230	-1.625
6-311G++(df,pd) Grimme D4	0.721	0.0150	0.833	-0.546	-0.0107	-0.669

revTPSS (non-Pople basis sets)		RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP	QF D4	1.541	0.0511	1.792	-0.255	0.0232	-0.321
def2-SVPD	QF D4	0.573	0.0160	0.651	-0.407	0.000517	-0.462
def2-TZVP	QF D4	0.544	0.0114	0.640	-0.415	-0.00371	-0.503
def2-SVP	Grimme D4	1.903	0.0260	2.301	-1.420	-0.0201	-1.909
def2-SVPD	Grimme D4	1.607	0.0290	1.948	-1.2679	-0.0255	-1.718
def2-TZVP	Grimme D4	0.439	0.0130	0.513	-0.290	-0.000286	-0.323
PC2	Grimme D4	0.295	0.0145	0.346	-0.110	0.00313	-0.101
def2-TZVPPD	Grimme D4	0.256	0.0197	0.294	-0.0172	0.00566	0.0231

revSCAN (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G** QF D4	1.118	0.0253	1.300	-0.017	0.0036	-0.082
6-311G++** QF D4	0.451	0.0140	0.534	-0.182	0.0023	-0.238
6-311G(df,pd) QF D4	1.084	0.0234	1.272	-0.101	-0.0007	-0.198
6-311G++(df,pd) QF D4	0.434	0.0128	0.521	-0.216	0.0004	-0.276

revSCAN (non-Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP QF D4	1.319	0.0262	1.566	-0.184	-0.0033	-0.332
def2-SVPD QF D4	0.598	0.0148	0.739	-0.329	-0.0084	-0.481
def2-TZVP QF D4	0.414	0.0133	0.489	-0.135	0.0003	-0.188

R2SCAN (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G** QF D4	1.213	0.0237	1.451	-0.213	-0.0031	-0.368
6-311G++** QF D4	0.585	0.0177	0.686	-0.020	0.0030	-0.046
6-311G(df,pd) QF D4	1.195	0.0229	1.441	-0.288	-0.0073	-0.476
6-311G++(df,pd) QF D4	0.534	0.0162	0.631	-0.024	0.0009	-0.052
6-311G** Grimme D4	1.414	0.0209	1.740	-1.020	-0.0143	-1.412
6-311G++** Grimme D4	0.666	0.0132	0.828	-0.486	-0.0052	-0.666
6-311G(df,pd) Grimme D4	1.479	0.0228	1.826	-1.095	-0.0179	-1.525
6-311G++(df,pd) Grimme D4	0.740	0.0133	0.920	-0.575	-0.0082	-0.784

R2SCAN (non-Pople basis sets)		RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP	QF D4	1.481	0.0272	1.805	-0.386	-0.0100	-0.634
def2-SVPD	QF D4	0.731	0.0175	0.919	-0.485	-0.0135	-0.721
def2-TZVP	QF D4	0.572	0.0176	0.657	0.019	0.0026	-6.38e-05
def2-SVP	Grimme D4	1.740	0.0270	2.157	-1.193	-0.0210	-1.687
def2-SVPD	Grimme D4	1.573	0.0246	1.946	-1.291	-0.0228	-1.789
def2-TZVP	Grimme D4	0.529	0.0138	0.651	-0.301	-0.0035	-0.427
PC2	Grimme D4	0.449	0.0137	0.544	-0.185	-0.0023	-0.268
def2-TZVPPD	Grimme D4	0.355	0.0124	0.427	-0.122	-0.0017	-0.193

RGE2 (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G** QF D4	1.487	0.0431	1.732	-0.705	0.0146	-0.855
6-311G++** QF D4	0.864	0.0200	1.015	-0.687	-0.00527	-0.834
6-311G(df,pd) QF D4	1.480	0.0378	1.729	-0.768	0.00762	-0.905
6-311G++(df,pd) QF D4	0.822	0.0196	0.957	-0.661	-0.00703	-0.797

RGE2 (non-Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP QF D4	1.828	0.0652	2.094	-0.612	0.0255	-0.750
def2-SVPD QF D4	0.818	0.0299	0.932	-0.582	0.00250	-0.690
def2-TZVP QF D4	0.741	0.0215	0.873	-0.541	-0.00265	-0.660

revM06_L (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G**	0.926	0.0246	1.024	-0.299	0.0189	-0.637
6-311G++**	0.641	0.0301	0.549	0.141	0.0276	-0.0403
6-311G(df,pd)	0.934	0.0229	1.067	-0.389	0.0171	-0.748
6-311G++(df,pd)	0.612	0.0290	0.569	0.103	0.0262	-0.0995

revM06_L (non-Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP	0.998	0.0197	1.136	-0.406	0.0123	-0.770
def2-SVPD	1.051	0.0206	1.417	-0.689	0.0127	-1.122
def2-TZVP	0.580	0.0265	0.516	0.115	0.0240	-0.0791
PC2	0.555	0.0242	0.493	0.153	0.0212	-0.0220
def2-TZVPPD	0.595	0.0277	0.568	0.184	0.0249	0.00152

BP86 (Pople basis sets)	RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
6-311G** QF D4	1.541	0.0419	1.709	-0.557	0.0230	-0.758
6-311G++** QF D4	0.688	0.0246	0.765	-0.311	0.0121	-0.415
6-311G(df,pd) QF D4	1.540	0.0431	1.703	-0.515	0.0235	-0.694
6-311G++(df,pd) QF D4	0.624	0.0189	0.712	-0.361	0.00798	-0.470
6-311G** Grimme D4	2.679	0.0265	3.283	-2.292	-0.0232	-3.127
6-311G++** Grimme D4	1.664	0.0163	2.055	-1.436	-0.0111	-1.944
6-311G(df,pd) Grimme D4	2.726	0.0273	3.348	-2.328	-0.0250	-3.186
6-311G++(df,pd) Grimme D4	1.732	0.0171	2.134	-1.480	-0.0128	-2.002

BP86 (non-Pople basis sets)		RMSD(A)	RMSD(ML)	RMSD(MV)	MD(A)	MD(ML)	MD(MV)
def2-SVP	QF D4	1.903	0.0397	2.183	-0.872	0.0156	-1.170
def2-SVPD	QF D4	0.755	0.0364	0.714	-0.292	0.0220	-0.377
def2-TZVP	QF D4	0.706	0.0244	0.755	-0.351	0.0140	-0.490
def2-SVP	Grimme D4	3.191	0.0339	3.938	-2.654	-0.0317	-3.689
def2-SVPD	Grimme D4	2.799	0.0303	3.477	-2.344	-0.0281	-3.235
def2-TZVP	Grimme D4	1.606	0.0137	1.989	-1.393	-0.00876	-1.896
def2-TZVPPD	Grimme D4	1.352	0.0121	1.693	-1.110	-0.00567	-1.532

Appendix B: Third Party Software Licenses and Credits

RDKit

Some QF products use RDKit with BSD 3-Clause License.

https://opensource.org/license/BSD-3-clause/

Visit https://rdkit.org/ for more info.

XTB

QF products use XTB package via calling the xtb executable program under LGPL3 license.

https://www.gnu.org/licenses/lgpl-3.0.en.html

Visit https://github.com/grimme-lab/xtb/blob/main/COPYING for more info.
Scientfic citation: C. Bannwarth, E. Caldeweyher, S. Ehlert, A. Hansen, P. Pracht, J. Seibert, S. Spicher, S. Grimme WIREs Comput. Mol. Sci., 2020, 11, e01493. DOI: 10.1002/wcms.1493

The source code without any modification from the original version as well as the exact copy of the binary executable that we use can be obtained from our web site at the following location:
https://bettermolecularmodelling.com/qffileexchange/LGPL3Packages/XTB.tar.gz

LIBXC

QF applications are using the LIBXC third party library for evaluations of DFT functional values and their derivatives.

Scientific citation: Susi Lehtola, Conrad Steigemann, Micael J.T. Oliveira, and Miguel A.L. Marques, Recent developments in Libxc - A comprehensive library of functionals for density functional theory, Software X 7, 1 (2018). doi: 10.1016/j.softx.2017.11.002
LIBXC is released under the MPL license (v. 2.0).
License information is at

https://www.mozilla.org/en-US/MPL/2.0/
More about LIBXC is at https://www.tddft.org/programs/Libxc/

Qt

Qt libraries are used under LGPL3 license with our application. All Qt libraries are linked dynamically against our executable and they must be installed separately from our application.
The source code of the Qt5.15 version as well as the compiled libraries can be downloaded from our website at the following location:
https://bettermolecularmodelling.com/qffileexchange/LGPL3Packages/Qt5.tar.gz
The Qt source files have not been modified from their original official version.

License information:

https://www.gnu.org/licenses/lgpl-3.0.en.html

Mopac

The Mopac program are used under LGPL3 license with our applications.

https://www.gnu.org/licenses/lgpl-3.0.en.html

The mopac Linux binary executable is used directly from the downloaded package without any modification of the source code and without even a recompilation.
The mopac package including the source code as well as the executable and the license information are available on our website.
https://bettermolecularmodelling.com/qffileexchange/LGPL3Packages/mopac-22.0.6-linux.tar.gz
Scientific citation:
title: MOPAC
type: software
version: 22.0.6
doi: 10.5281/zenodo.6511958
date-released: 2022-12-18
authors:
- family-names: Stewart
  given-names: "James J. P."
- family-names: Klamt
  given-names: Andreas
- family-names: Thiel
  given-names: Walter
- family-names: Danovich

given-names: David
- family-names: Rocha
given-names: "Gerd B."
- family-names: Gieseking
given-names: "Rebecca L."
- family-names: Moussa
given-names: "Jonathan E."
- family-names: Kurtz
given-names: "Henry A."
- family-names: Korambath
given-names: Prakashan
- family-names: Merz
given-names: "Kenneth M."
name-suffix: Jr.
- family-names: Wang
given-names: Bingze

Appendix C: QF User License Agreement

End User License Agreement
All software from QuantumFuture Scientific Software LLC is licensed to You (End-User) by QuantumFuture Scientific Software LLC, registered in Delaware and operated at 396 Limerick Road, Buda, Texas 78610, United States ("Licensor"), for use only under the terms of this License Agreement. By downloading the Licensed Application form , and any update thereto (as permitted by this License Agreement), You indicate that You agree to be bound by all of the terms and conditions of this License Agreement, and that You accept this License Agreement, referred to in this License Agreement as "Services."
The parties of this License Agreement acknowledge that the Services are not a Party to this License Agreement and are not bound by any provisions or obligations with regard to the Licensed Applications, such as warranty, liability, maintenance and support thereof. QuantumFuture Scientific Software LLC, not the Services, is solely responsible for the Licensed Applications and the content thereof.
This License Agreement may not provide for usage rules for the Licensed Application that are in conflict with the latest ("Usage Rules"). QuantumFuture Scientific Software LLC acknowledges that it had the opportunity to review the Usage Rules and this License Agreement is not conflicting with them.
All software from QuantumFuture Scientific Software LLC when purchased or downloaded through the Services, is licensed to You for use only under the terms of this License Agreement. The Licensor reserves all rights not expressly granted to You. All software from QuantumFuture Scientific Software LLC is to be used on devices that operate with.
1. THE APPLICATIONS
All software from QuantumFuture Scientific Software LLC ("Licensed Applications") are created to help in the general area of scientific research and modeling and they are used to perform scientific research

and modeling in the general areas of quantum chemistry, computational drug and material designs.
The Licensed Applications are not tailored to comply with industry-specific regulations (Health Insurance Portability and Accountability Act (HIPAA), Federal Information Security Management Act (FISMA), etc.), so if your interactions would be subjected to such laws, you may not use any of these Licensed Application. You may not use the Licensed Applications in a way that would violate the Gramm-Leach-Bliley Act (GLBA).
2. SCOPE OF LICENSE
2.1 Violations of the obligations mentioned above, as well as the attempt of such infringement, may be subject to prosecution and damages.
2.2 Licensor reserves the right to modify the terms and conditions of licensing.
2.3 Nothing in this license should be interpreted to restrict third-party terms. When using the Licensed Applications, You must ensure that You comply with applicable third-party terms and conditions.
3. TECHNICAL REQUIREMENTS
3.1 You acknowledge that it is Your responsibility to confirm and determine that the apps end-user device on which You intend to use the Licensed Applications satisfies the required technical specifications.
3.2 Licensor reserves the right to modify the technical specifications as it sees appropriate at any time.
4. MAINTENANCE AND SUPPORT
4.1 The Licensor is solely responsible for providing any maintenance and support services for these Licensed Applications. You can reach the Licensor at the email address listed in the Overview for this Licensed Application.
4.2 QuantumFuture Scientific Software LLC and the End-User acknowledge that the Services have no obligation whatsoever to furnish any maintenance and support services with respect to the Licensed Applications.
5. USER-GENERATED CONTRIBUTIONS
The Licensed Applications may invite you to chat, contribute to, or participate in blogs, message boards, online forums, and other functionality, and may provide you with the opportunity to create, submit, post, display, transmit, perform, publish, distribute, or broadcast content and materials to us including but not limited to computer programs, text, writings, video, audio, photographs, graphics, comments, suggestions, or personal information or other material (collectively, "Contributions"). Contributions may be viewable by other users of the Licensed Application and through third-party websites or applications. As such, any Contributions you transmit may be treated as non-confidential and non-proprietary. When you create or make available any Contributions, you thereby represent and warrant that:
1. The creation, distribution, transmission, public display, or performance, and the accessing, downloading, or copying of your Contributions do not and will not infringe the proprietary rights, including but not limited to the copyright, patent, trademark, trade secret, or moral rights of any third party.
2. You are the creator and owner of or have the necessary licenses, rights, consents, releases, and permissions to use and to authorize us, the Licensed Applications, and other users of the Licensed Applications to use your Contributions in any manner contemplated by the Licensed Applications and this License Agreement.
3. You have the written consent, release, and/or permission of each and every identifiable individual person in your Contributions to use the name or likeness or each and every such identifiable individual person to enable inclusion and use of your Contributions in any manner contemplated by the Licensed

Application and this License Agreement.
4. Your Contributions are not false, inaccurate, or misleading.
5. Your Contributions are not unsolicited or unauthorized advertising, promotional materials, pyramid schemes, chain letters, spam, mass mailings, or other forms of solicitation.
6. Your Contributions are not obscene, lewd, lascivious, filthy, violent, harassing, libelous, slanderous, or otherwise objectionable (as determined by us).
7. Your Contributions do not ridicule, mock, disparage, intimidate, or abuse anyone.
8. Your Contributions are not used to harass or threaten (in the legal sense of those terms) any other person and to promote violence against a specific person or class of people.
9. Your Contributions do not violate any applicable law, regulation, or rule.
10. Your Contributions do not violate the privacy or publicity rights of any third party.
11. Your Contributions do not violate any applicable law concerning child pornography, or otherwise intended to protect the health or well-being of minors.
12. Your Contributions do not include any offensive comments that are connected to race, national origin, gender, sexual preference, or physical handicap.
13. Your Contributions do not otherwise violate, or link to material that violates, any provision of this License Agreement, or any applicable law or regulation.
Any use of the Licensed Application in violation of the foregoing violates this License Agreement and may result in, among other things, termination or suspension of your rights to use the Licensed Applications.
6. CONTRIBUTION LICENSE
By posting your Contributions to any part of the Licensed Applications or making Contributions accessible to the Licensed Applications by linking your account from the Licensed Applications to any of your social networking accounts, you automatically grant, and you represent and warrant that you have the right to grant, to us an unrestricted, unlimited, irrevocable, perpetual, non-exclusive, transferable, royalty-free, fully-paid, worldwide right, and license to host, use copy, reproduce, disclose, sell, resell, publish, broad cast, retitle, archive, store, cache, publicly display, reformat, translate, transmit, excerpt (in whole or in part), and distribute such Contributions (including, without limitation, your image and voice) for any purpose, commercial advertising, or otherwise, and to prepare derivative works of, or incorporate in other works, such as Contributions, and grant and authorize sublicenses of the foregoing. The use and distribution may occur in any media formats and through any media channels.
This license will apply to any form, media, or technology now known or hereafter developed, and includes our use of your name, company name, and franchise name, as applicable, and any of the trademarks, service marks, trade names, logos, and personal and commercial images you provide. You waive all moral rights in your Contributions, and you warrant that moral rights have not otherwise been asserted in your Contributions.
We do not assert any ownership over your Contributions. You retain full ownership of all of your Contributions and any intellectual property rights or other proprietary rights associated with your Contributions. We are not liable for any statements or representations in your Contributions provided by you in any area in the Licensed Application. You are solely responsible for your Contributions to the Licensed Applications and you expressly agree to exonerate us from any and all responsibility and to refrain from any legal action against us regarding your Contributions.
We have the right, in our sole and absolute discretion, (1) to edit, redact, or otherwise change any Contributions; (2) to recategorize any Contributions to place them in more appropriate locations; and (3)

to prescreen or delete any Contributions at any time and for any reason, without notice. We have no obligation to monitor your Contributions.

7. LIABILITY

7.1 LICENSOR IS NOT LIABLE TO LICENSEE FOR ANY DAMAGES, INCLUDING COMPENSATORY, SPECIAL, INCIDENTAL, EXEMPLARY, PUNITIVE, OR CONSEQUENTIAL DAMAGES, CONNECTED WITH OR RESULTING FROM THIS LICENSE AGREEMENT OR LICENSEE'S USE OF THIS SOFTWARE. IN NO EVENT SHALL QUANTUMFUTURE SCIENTIFIC SOFTWARE LLC BE RESPONSIBLE OR LIABLE, WHETHER IN CONTRACT, TORT, WARRANTY OR UNDER ANY STATUTE OR ON ANY OTHER BASIS FOR SPECIAL, INDIRECT, INCIDENTAL, MULTIPLE, PUNITIVE, OR CONSEQUENTIAL DAMAGES, EVEN IF QUANTUMFUTURE SCIENTIFIC SOFTWARE LLC IS ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES, INCLUDING WITHOUT LIMITATION DAMAGES ARISING FROM OR RELATED TO LOSS OF USE, LOSS OF DATA, DOWNTIME, OR FOR LOSS OF REVENUE, PROFITS, GOODWILL OR BUSINESS OR OTHER FINANCIAL LOSS. . IN NO EVENT WILL COMPANY'S AGGREGATE CUMULATIVE LIABILITY FOR ANY CLAIMS ARISING OUT OF OR RELATED TO THIS AGREEMENT EXCEED $10.00 OR 1% OF THE AMOUNT RECIPIENT ACTUALLY PAID COMPANY UNDER THIS AGREEMENT.

7.2 Licensor takes no accountability or responsibility for any damages caused due to a breach of duties according to Section 2 of this License Agreement. To avoid data loss, You are required to make use of backup functions of the Licensed Application to the extent allowed by applicable third-party terms and conditions of use. You are aware that in case of alterations or manipulations of the Licensed Applications, You will not have access to the Licensed Applications.

8. WARRANTY

8.1 ALL SOFTWARE AND DOCUMENTATION FROM QUANTUMFUTURE SCIENTIFIC SOFTWARE LLC ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND COMPANY AND ITS LICENSORS DISCLAIM ALL WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. NO ORAL OR WRITTEN ADVICE OR CONSULTATION GIVEN BY COMPANY, ITS AGENTS OR EMPLOYEES WILL IN ANY WAY GIVE RISE TO A WARRANTY. THE ENTIRE RISK ARISING OUT OF THE USE OF THE SOFTWARE REMAINS WITH RECIPIENT.

8.2 Licensor warrants that the Licensed Applications are free of spyware, trojan horses, viruses, or any other malware at the time of Your download.

9. LEGAL COMPLIANCE

You represent and warrant that You are not located in a country that is subject to a US Government embargo, or that has been designated by the US Government as a "terrorist supporting" country; and that You are not listed on any US Government list of prohibited or restricted parties.

10. CONTACT INFORMATION

For general inquiries, complaints, questions or claims concerning the Licensed Application, please contact:

Support @ QuantumFuture Scientific

396 Limerick Road
Buda, TX 78610
United States
support@qfsciences.com

11. TERMINATION

The license is valid until terminated by QuantumFuture Scientific Software LLC or by You. Your rights under this license will terminate automatically and without notice from QuantumFuture Scientific Software LLC if You fail to adhere to any term(s) of this license. Upon License termination, You shall stop all use of the Licensed Applications, and destroy all copies, full or partial, of the Licensed Applications.

12. THIRD-PARTY TERMS OF AGREEMENTS AND BENEFICIARY

QuantumFuture Scientific Software LLC represents that QuantumFuture Scientific Software LLC will comply with applicable third-party terms of agreement when using Licensed Applications.

In Accordance with Section 9 of the "Instructions for Minimum Terms of Developer's End-User License Agreement," subsidiaries shall be third-party beneficiaries of this End User License Agreement and — upon Your acceptance of the terms and conditions of this License Agreement, will have the right (and will be deemed to have accepted the right) to enforce this End User License Agreement against You as a third-party beneficiary thereof.

13. INTELLECTUAL PROPERTY RIGHTS

QuantumFuture Scientific Software LLC and the End-User acknowledge that, in the event of any third-party claim that the Licensed Applications or the End-User's possession and use of that Licensed Applications infringes on the third party's intellectual property rights, QuantumFuture Scientific Software LLC, and not the Services, will be solely responsible for the investigation, defense, settlement, and discharge or any such intellectual property infringement claims.

14. APPLICABLE LAW

This License Agreement is governed by the laws of the State of Texas excluding its conflicts of law rules.

15. MISCELLANEOUS

15.1 If any of the terms of this agreement should be or become invalid, the validity of the remaining provisions shall not be affected. Invalid terms will be replaced by valid ones formulated in a way that will achieve the primary purpose.

15.2 Collateral agreements, changes and amendments are only valid if laid down in writing. The preceding clause can only be waived in writing.