QUANTUM FUTURE
Scientific Software

Web: QFSciences.com
Support: Support@qfsciences.com
CEO: Laszlo@qfsciences.com
Phone: 505-570-1150

QuantumFuture Software on Azure

(Easy to Use, Secure, Efficient, Affordable)

This Azure program bundle from QF has four command line applications: qfazureconfig, qfazurelaunch, qfgetresultsfromazure and qfdeleteazurestorage. The names of the applications are hopefully descriptive enough and the functionalities can be described in some short sentences. All four applications are offered for Linux operating systems where the file extensions of the applications are .x and for MS Windows where the file extensions are .exe as usual. All our MS Windows applications are digitally signed for security purposes.

The qfazureconfig application must be run first and this application must be executed only once after the users set up their own Azure account. This application asks the exact name of the azure resources and the exact names of the azure batch accounts that the users want to use for QF calculations and then it creates a special configuration text file that all QF azure applications use. This configuration file, together with the QuantumFuture.lic license file and the input file (SomeProjectName.tar.gz format) always need to be in the working directory. The qfazureconfig application also checks if Azure CLI, Azure batch and python3 are installed and it offers installation options if one or more are missing. All our Azure software depends only on bash, Azure CLI, Azure Batch and python3.

After the QF Azure configuration step is done the rest three applications can be used as command line applications with convenient command line interfaces and with Graphical User Interface (GUI) as well. Either way, thousands of *ab initio* calculations, QM based conformation searches and semiempirical QM calculations can be launched on Azure as easily as a short one-line command or with a few clicks with the mouse.

Using qfazurelaunch.x Application with Command Line Interface

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 a ProjectName.tar.gz file. 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 and the QF Azure configuration 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.

Details

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

1. User provides credentials. 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, with the CLI command:

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

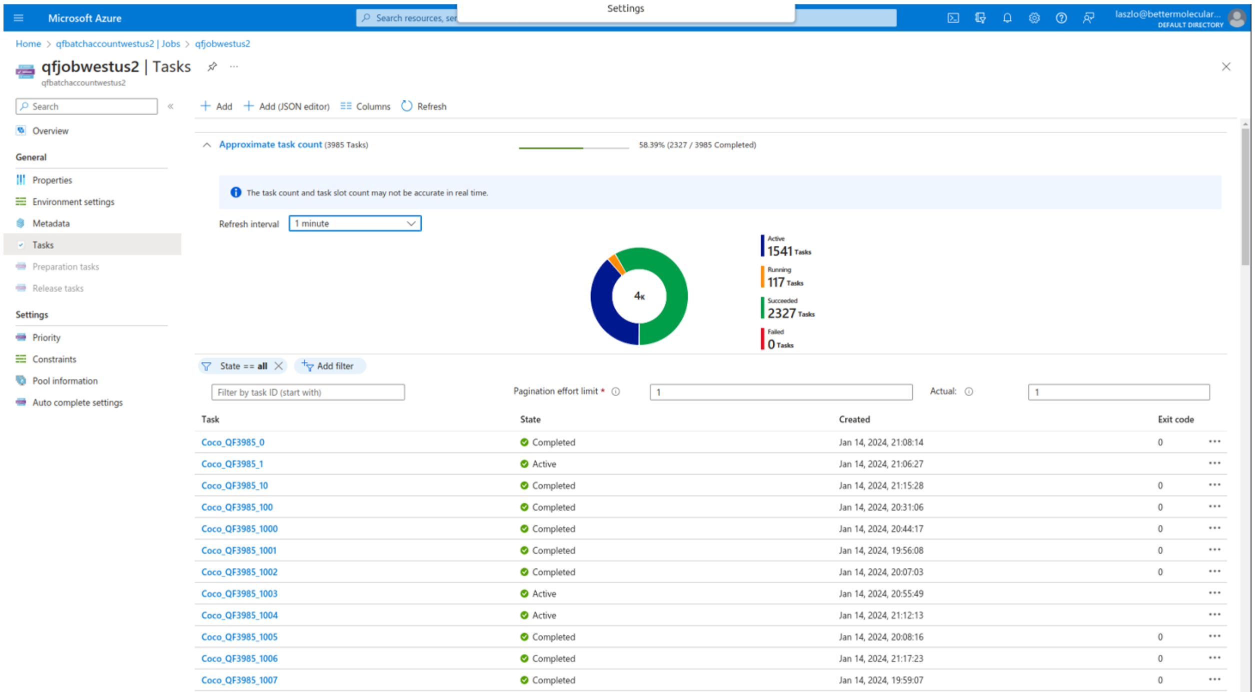
   The output will contain secrets that the users should keep for themselves. An example is:

   ```
   {
     "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!



Approximate task count (3985 Tasks) 58.39% (2327 / 3985 Completed)

The task count and task slot count may not be accurate in real time.

Refresh interval: 1 minute

Active 1541 Tasks
Running 117 Tasks
Succeeded 2327 Tasks
Failed 0 Tasks

State == all

Filter by task ID (start with)

Pagination effort limit * 1

Actual: 1

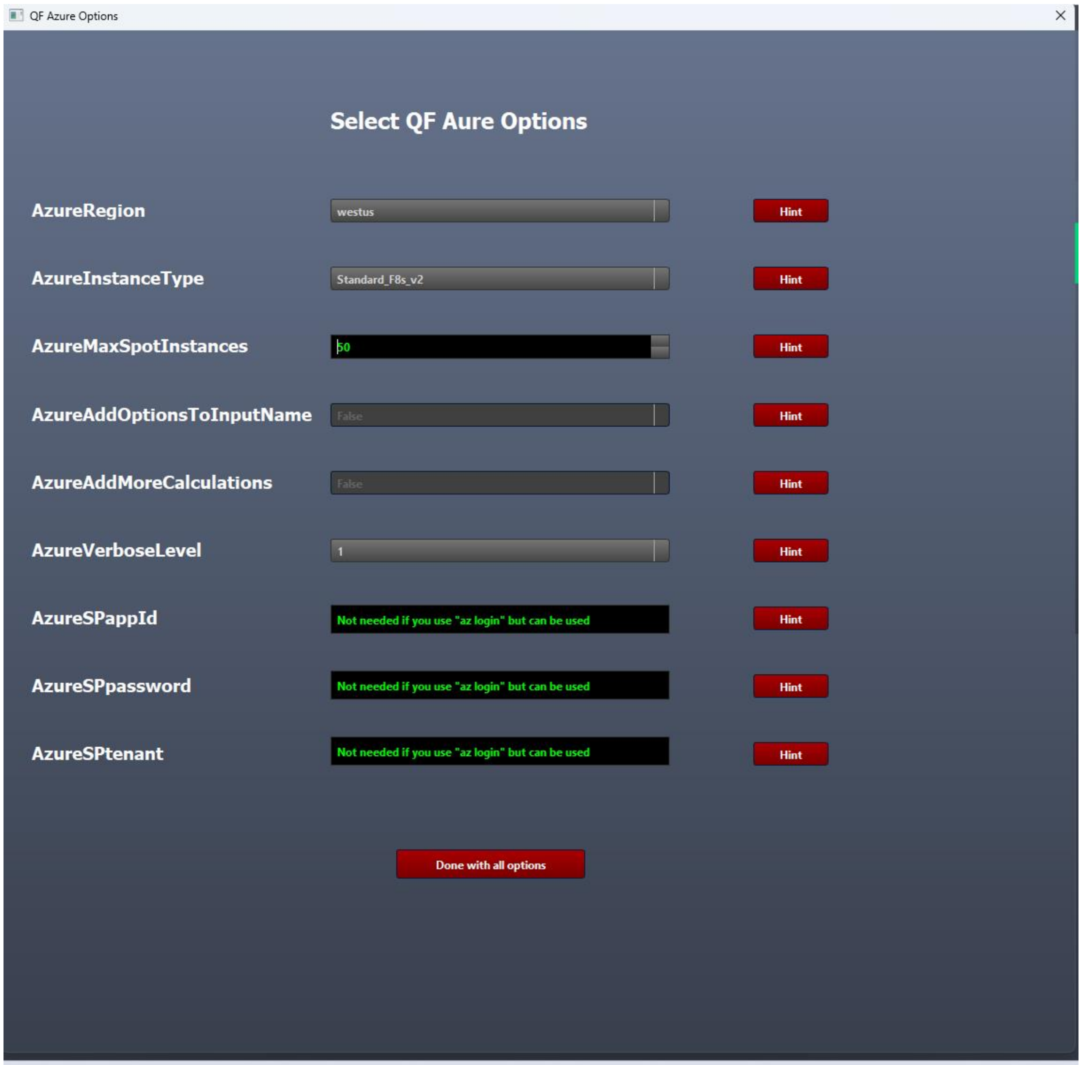
Task	State	Created	Exit code
Coco_QF3985_0	Completed	Jan 14, 2024, 21:08:14	0
Coco_QF3985_1	Active	Jan 14, 2024, 21:06:27	
Coco_QF3985_10	Completed	Jan 14, 2024, 21:15:28	0
Coco_QF3985_100	Completed	Jan 14, 2024, 20:31:06	0
Coco_QF3985_1000	Completed	Jan 14, 2024, 20:44:17	0
Coco_QF3985_1001	Completed	Jan 14, 2024, 19:56:08	0
Coco_QF3985_1002	Completed	Jan 14, 2024, 20:07:03	0
Coco_QF3985_1003	Active	Jan 14, 2024, 20:55:49	
Coco_QF3985_1004	Active	Jan 14, 2024, 21:12:13	
Coco_QF3985_1005	Completed	Jan 14, 2024, 20:08:16	0
Coco_QF3985_1006	Completed	Jan 14, 2024, 21:17:23	0
Coco_QF3985_1007	Completed	Jan 14, 2024, 19:59:07	0

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. These steps can be done with our GUI as well.

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.

Details about the Supported Scientific Applications and GUI Appearance

Azure options are shown on the GUI like this.



QF Azure Options

Select QF Aure Options

AzureRegion	westus
AzureInstanceType	Standard_F8s_v2
AzureMaxSpotInstances	50
AzureAddOptionsToInputName	False
AzureAddMoreCalculations	False
AzureVerboseLevel	1
AzureSPappId	Not needed if you use "az login" but can be used
AzureSPpassword	Not needed if you use "az login" but can be used
AzureSPtenant	Not needed if you use "az login" but can be used

Three applications from QuantumFuture are supported: QFDFT, QFConfsearchDFT, and QFLowerLevel.

QFDFT

a. Developed anew in C++ from the ground up.

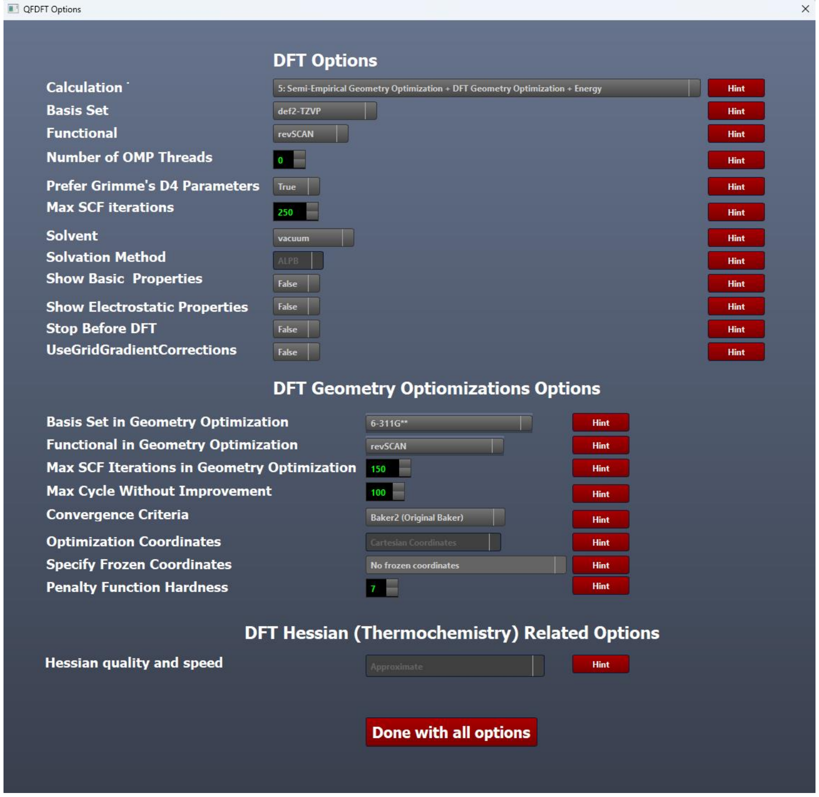
b. Incorporates a unique and very advanced new version of the Fourier Transform Coulomb method (the original developments of the FTC method are described here, here and here) for Coulomb matrix evaluations with linear scaling in molecular size delivering rapid yet precise solutions for DFT energies and atomic forces.

c. Introduces a new innovative atomic grid technology for exchange correlation components, ensuring both accuracy and performance.

d. Enhances accuracy in VDW interactions through novel D4 corrections, meticulously optimized not only for functional considerations but also for functional-basis set pairs.

e. Detailed performance benchmarks are available here, while detailed intermolecular interaction accuracy benchmarks can be accessed here.

f. Current functionalities include ground state closed shell singlet energy calculations, geometry optimizations with or without constraints, and statistical thermochemistry calculations. Additionally, a novel approximation method facilitates Hessian calculations, providing approximately tenfold acceleration compared to traditional approaches. Various fundamental properties are also readily accessible.



DFT Options

Calculation	5: Semi-Empirical Geometry Optimization + DFT Geometry Optimization + Energy
Basis Set	def2-TZVP
Functional	revSCAN
Number of OMP Threads	0
Prefer Grimme's D4 Parameters	True
Max SCF iterations	250
Solvent	vacuum
Solvation Method	ALPB
Show Basic Properties	False
Show Electrostatic Properties	False
Stop Before DFT	False
UseGridGradientCorrections	False

DFT Geometry Optiomizations Options

Basis Set in Geometry Optimization	6-311G**
Functional in Geometry Optimization	revSCAN
Max SCF Iterations in Geometry Optimization	150
Max Cycle Without Improvement	100
Convergence Criteria	Baker2 (Original Baker)
Optimization Coordinates	Cartesian Coordinates
Specify Frozen Coordinates	No frozen coordinates
Penalty Function Hardness	7

DFT Hessian (Thermochemistry) Related Options

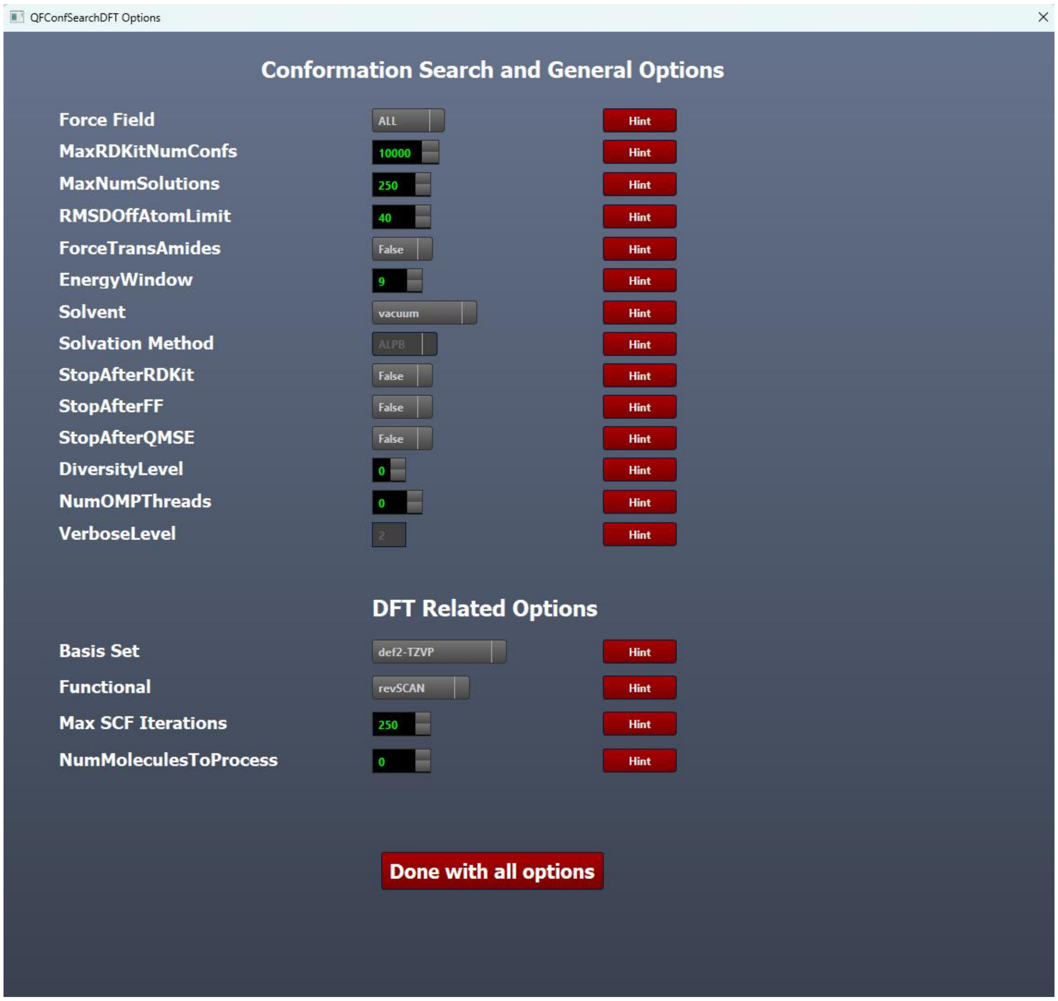
Hessian quality and speed	Approximate

QFConfsearchDFT

a. Diverging from conventional cheminformatics and force field-based methods, this program offers an alternative approach to generating molecular conformations rooted in much more realistic physics principles.

b. The process initiates with the generation of thousands of cheminformatics-based conformations (defaulting to 10,000), followed by local geometry optimizations employing three distinct force fields. Subsequently, QM semi-empirical methods are applied for further local geometry refinements followed by deduplications, culminating in accurate DFT-D4 *ab initio* energy calculations. The final conformational list is then sorted based on very good quality DFT-D4 energies.

c. Introducing greater scientific rigor into molecular conformation generation holds promise for numerous critical Computer-Aided Drug Design (CADD) projects. We conducted comprehensive benchmarks involving nearly 150 FDA-approved drugs, categorizing them based on the number of rotational bonds (25 compounds for each ranging from 2 to 7 rotatable bonds). Analysis of heavy atom RMSD differences between the closest calculated conformations and experimental geometries revealed that over 63% exhibited differences less than 0.3 Angstrom RMSD, with over 80% demonstrating differences less than 0.5 Angstrom RMSD compared to experimental structures. Further details are available here.



Conformation Search and General Options

Force Field	ALL	Hint
MaxRDKitNumConfs	10000	Hint
MaxNumSolutions	250	Hint
RMSDOffAtomLimit	40	Hint
ForceTransAmides	False	Hint
EnergyWindow	9	Hint
Solvent	vacuum	Hint
Solvation Method	ALPB	Hint
StopAfterRDKit	False	Hint
StopAfterFF	False	Hint
StopAfterQMSE	False	Hint
DiversityLevel	0	Hint
NumOMPThreads	0	Hint
VerboseLevel	2	Hint

DFT Related Options

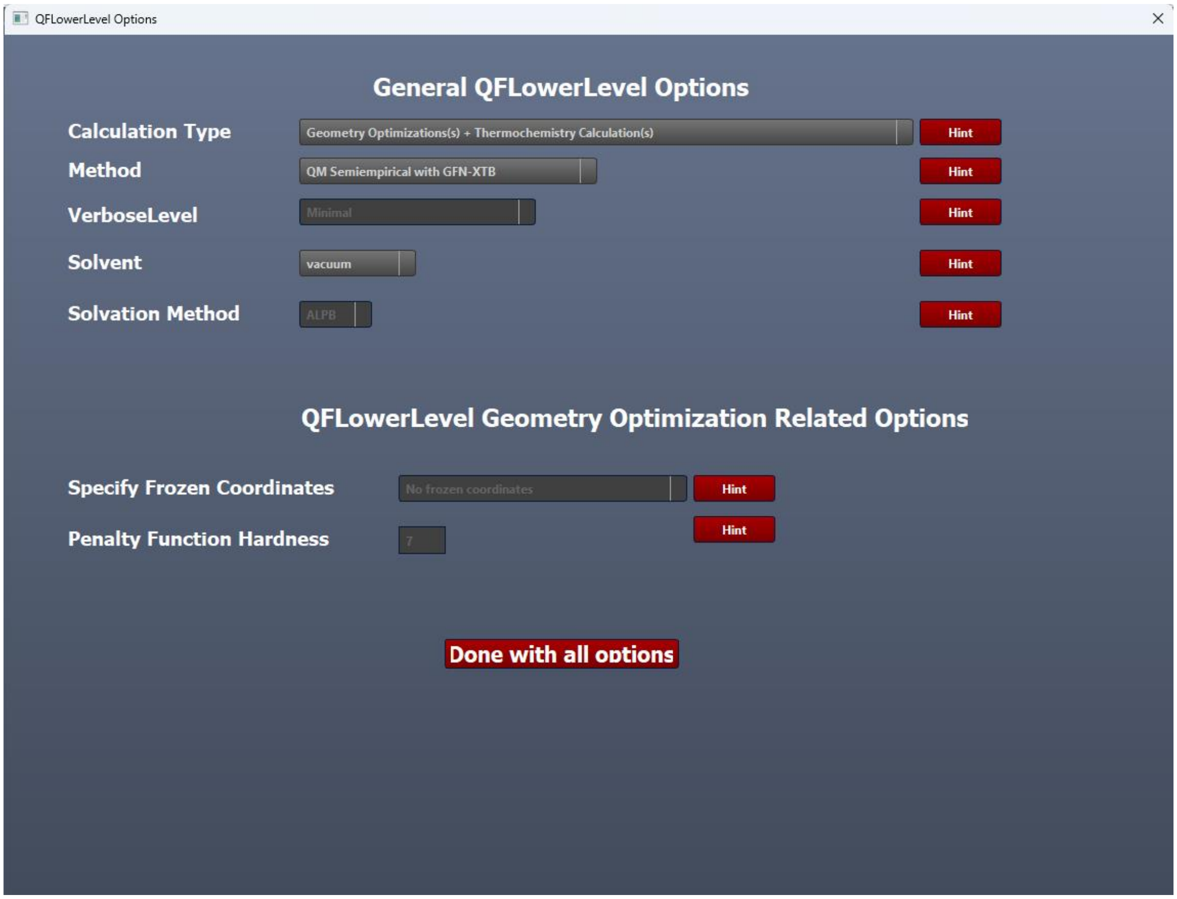
Basis Set	def2-TZVP	Hint
Functional	revSCAN	Hint
Max SCF Iterations	250	Hint
NumMoleculesToProcess	0	Hint

QFLowerLevel

a. Like QFDFT, this application offers a range of calculation options but employs faster QM semi-empirical methods and force fields. It serves as an ideal tool for conducting numerous preliminary calculations prior to more comprehensive *ab initio* DFT-D4 analyses, or for obtaining approximate molecular geometries and statistical thermodynamic properties.

b. The latest version of the GFN-XTB method from the XTB package, along with the PM6-D3H4X and PM7 methods from MOPAC, are supported, alongside UFF, MMFF94, and GFN-FF force fields. Additional supported methods will be incorporated in future updates.

c. Sharing the same geometry optimization capabilities as QFDFT, QFlowerLevel offers identical constraint options, allowing users to optimize Hydrogen atoms while freezing heavy atoms, for instance. The application shares statistical thermodynamics routines with QFDFT as well.



QFLowerLevel Options

General QFLowerLevel Options

Calculation Type: Geometry Optimizations(s) + Thermochemistry Calculation(s)

Method: QM Semiempirical with GFN-XTB

VerboseLevel: Minimal

Solvent: vacuum

Solvation Method: ALPB

QFLowerLevel Geometry Optimization Related Options

Specify Frozen Coordinates: No frozen coordinates

Penalty Function Hardness: 7

Request a complimentary trial license today and discover the remarkable capabilities of this software firsthand. Seamlessly conduct millions of high-quality QM calculations with ease, ensuring precision and efficiency in your computational chemistry and CADD endeavors. Have FUN exploring the boundless possibilities!