gqteaWinToga User Manual

Index

  - General Tools
    - 3D Molecular Viewer
    - Energy Plots
    - Molecular Axis Alignment
    - Select Frames
    - Frame Selection by Interatomic Distance Range
    - CPMD Input to XYZ Converter
    - SH Geometry Analyzer
    - Convert cp.x .pos File to trajec.xyz
    - Compute Forces from cp.x .for File
    - Coordinate Converter
    - Convert Coordinates from Angstrom to Bohr
    - Unit Converter
- Configuration
- Practical Examples
  - Example 1: Find a Bond-Length Free-Energy Minimum
  - Example 2: Prepare a CPMD Collision Restart
  - Example 3: Group Surface Hopping Frames by State
  - Example 4: Convert a cp.x Trajectory
  - Example 5: Analyze All Solute Bond Distances
  - Example 6: Analyze All Solute Bond Angles
  - Example 7: Analyze All Solute Dihedral Angles
- Troubleshooting
  - "No file was selected!"
  - "Please input a valid value for ..."
  - "Invalid format for ... Please input exactly two atom labels."
  - "Failed to open file" or "Failed to read file"
  - "The file is empty!"
  - "Invalid line format in TRAJECTORY file"
  - "The stop frame must be less than the total number of frames"
  - ENERGY file format errors
  - Solvent box insertion warnings
  - Missing `SH_STATE.dat` or `TRAJEC.xyz`
  - Molecular viewer does not open or shows no 3D view
  - Packaged Windows executable cannot find GLFW
- Version, Contributors, and License

Overview

gqteaWinToga is a desktop molecular analysis toolkit developed by the gQTEA group. It helps users prepare, inspect, convert, visualize, and analyze molecular simulation data from CPMD, Quantum ESPRESSO `cp.x`, ORCA, surface hopping workflows, and related molecular dynamics tools.

The program is a graphical application built with Python and Toga. Most tools follow the same pattern: open a tool window from the main tabbed interface, browse to one or more input files, enter calculation parameters, then click the tool's action button to generate outputs.

The current application title identifies the toolkit as:

```
gQTEA-0.3.1 Molecular Analysis Toolkit
```

Installation and Requirements

Requirements

Recommended environment:

- Python 3.10 or newer.
- Windows with a working graphical desktop session.
- A Python virtual environment.

Python packages used by the application include:

- `toga-winforms`
- `numpy`
- `scipy`
- `matplotlib`
- `glfw`
- `PyOpenGL`
- `PyOpenGL-accelerate`
- `PyMuPDF`

Some workflows also require files produced by external chemistry programs, such as CPMD, Quantum ESPRESSO `cp.x`, ORCA, Gaussian, or Vanderbilt `runatom.x`. gqteaWinToga prepares and analyzes files for these programs, but it does not replace the external simulation engines.

Install from Source

From PowerShell:

```
git clone https://github.com/AdemirCamargo57/gqteaWinToga.git
cd gqteaWinToga
python -m venv venv
.\venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
python -m pip install toga-winforms numpy scipy matplotlib glfw
PyOpenGL PyOpenGL-accelerate PyMuPDF
```

If you already have the source tree and an active virtual environment, install the packages in that environment.

Start the Program

Run the launcher from the source directory:

```
python gqteaWinToga.py
```

The main window opens with five tabs:

- **Geometry**
- **Inputs**
- **Structural**
- **Thermo**
- **Tools**

Choose a tab, click a tool, complete the fields in the tool window, and run the calculation or conversion.

Input File Basics

XYZ Files

Many tools read XYZ geometry or trajectory files. A standard single-frame XYZ file has:

```
5
Comment line
P  0.112  -0.092   0.064
O  0.112   1.475  -0.490
O  1.470  -0.876  -0.490
O -1.246  -0.876  -0.490
O  0.112  -0.092   1.727
```

The first line is the number of atoms. The second line is a comment. Each remaining line contains an atomic symbol and Cartesian coordinates.

Trajectory XYZ files repeat this block for each frame. CPMD trajectory analysis tools commonly expect a file named `TRAJEC.xyz`.

Atom Labels

Most analysis tools use 1-based atom labels. For example, `1 2` means atom 1 and atom 2 in the XYZ frame, not Python-style zero-based indices.

Common Units

- Distances: Angstrom.
- Angles: degrees.
- Temperature: kelvin.
- CPMD time step: atomic units unless a tool states otherwise.
- Some collision outputs distinguish between CPMD atomic-unit velocities and gqteaMD Angstrom/fs velocities.

Basic Workflows

Analyze a Bond Length

1. Start gqteaWinToga.
2. Open **Geometry**.
3. Click **Bond length analysis**.
4. Browse to `TRAJEC.xyz`.
5. Enter the maximum distance for the distribution, time step, sampling interval, temperature, two atom labels, and histogram bin width.
6. Keep **Show plots at the end** enabled if you want plot windows.
7. Keep **Save CSV outputs** enabled if you want CSV files in addition to `.dat` files.
8. Click **Analyze**.

Typical outputs include bond length time series, distribution data, free-energy data, summary text, and optional CSV files.

Build CPMD Input Files

1. Open **Inputs**.
2. Click **CPMD Inputs**.
3. Enter a prefix, total charge, periodic cell parameters, energy cutoff, and dual value.
4. Select a cell symmetry.
5. Browse to an XYZ starting geometry.
6. Click **Input Builder**.

The generated CPMD input files are based on the selected geometry and values entered in the form.

Select Frames and Generate Gaussian Inputs

1. Open **Tools**.
2. Click **Select Frames**.
3. Enter the starting frame, number of skipped frames, and stop frame.
4. Browse to `TRAJEC.xyz`.
5. Optionally enable Gaussian input generation and fill in checkpoint name, memory, processor count, route section, charge, and multiplicity.
6. Click **Select Frames**.

The tool writes `selected_frames.xyz`. If Gaussian generation is enabled, it also writes files named like `g16_input_1.gjf`, `g16_input_2.gjf`, and so on.

Plot CPMD Energy Data

1. Open **Tools**.
2. Click **Energy plots**.
3. Select the energy-file type.

4. Browse to the energy file.
5. Enter the simulation time step if needed.
6. Select the x-axis unit and one or more plot types.
7. Click **Plot**.

For CPMD ENERGY files, the program expects a valid numeric table with eight columns.

Features and Commands

The program is GUI-based. The "commands" below refer to the buttons and controls in each tool window.

Main Window

The main window organizes tools by category:

- **Geometry**: bond length, all bond distances, bond angle, all bond angles, dihedral angle, all dihedral angles, and hydrogen bond analysis.
- **Inputs**: CPMD, surface hopping, collision, Quantum ESPRESSO `cp.x`, ORCA, gqteaMD, and Vanderbilt `runatom.x` input builders.
- **Structural**: radial distribution function, mean residence time, autocorrelation function, and solvent box builders.
- **Thermo**: classical rate constant calculation.
- **Tools**: visualization, plotting, frame selection, converters, force processing, coordinate tools, and unit conversion.

Geometry Tools

Bond Length Analysis Use this tool to calculate bond lengths across a trajectory, average bond length, bond-length distribution, and Helmholtz free energy from the probability distribution.

Inputs:

- Maximum `r` for the distribution function.
- Simulation time step in atomic units.
- Sampling interval in frames.
- Simulation temperature in K.
- Two atom labels, for example `1 2`.
- Histogram bin width in Angstrom.
- `TRAJEC.xyz` trajectory file.

Options:

- **Show plots at the end**: opens plot windows after the run.
- **Save CSV outputs**: writes CSV versions of the main data tables.

Typical outputs:

- `bond_<atom1>_<atom2>.dat`
- `bond_distribution_<atom1>_<atom2>.dat`
- `free_energy_<atom1>_<atom2>.dat`
- `summary_<atom1>_<atom2>.txt`
- Optional CSV files with matching names.

All Bond Distance Analysis Use **Geometry > All bond distance analysis** to calculate distance statistics for every connected atom pair in an XYZ trajectory. This tool is intended for complete solute geometry monitoring in a molecular dynamics run. For a solute in a solvent box, provide the solute atom indices so solvent-solvent and solute-solvent distances are ignored.

How connectivity is detected:

- The program reads the first frame of the trajectory.
- Two atoms are considered connected if their first-frame distance is less than or equal to the **Maximum connection distance**.
- The default maximum connection distance is `1.5` Angstrom.
- If **Solute atom indices** is filled, only pairs where both atoms belong to the solute are considered.

Inputs:

- `trajectory.xyz` or `TRAJEC.xyz` trajectory file.
- **Maximum connection distance (A)**, for example `1.5`.
- **Solute atom indices**, for example `1 2 3 4 5`; leave blank to use all atoms.
- Output TXT filename, default `all_bond_analysis.txt`.

Step-by-step:

1. Open **Geometry > All bond distance analysis**.
2. Click **Browse** and select the trajectory XYZ file.
3. Enter the maximum connection distance. Use `1.5` for the default behavior.
4. If the system contains a solute in solvent, enter the solute atom labels in **Solute atom indices**. Use 1-based labels, separated by spaces.
5. Enter the output filename or keep `all_bond_analysis.txt`.
6. Click **Analyze**.
7. Wait until the final summary appears in the text area.
8. Open the statistics file and pair mapping file in the trajectory directory.

Output files:

- `all_bond_analysis.txt`: three whitespace-separated columns.
- `all_bond_analysis_pairs.txt`: row-to-atom-pair mapping.

The statistics file contains:

```
# average variance standard_deviation
```

Each following row contains one connected atom pair:

```
average_distance  population_variance  standard_deviation
```

The mapping file contains the number of frames, atom scope, solute atom list when used, and the atom pair associated with each row.

Bond Angle Analysis Use this tool to calculate bond angles across a trajectory, angle distributions, and free-energy profiles.

Inputs:

- Maximum angle for the distribution function.
- Simulation time step in atomic units.
- Sampling interval in frames.
- Simulation temperature in K.
- Three atom labels, for example `1 2 3`.
- Histogram bin width in degrees.
- `TRAJEC.xyz` trajectory file.

Options:

- **Show plots at the end**.
- **Save CSV outputs**.
- **Use sin(theta) Jacobian**: applies the angular Jacobian correction to the distribution/free-energy calculation.

All Bond Angle Analysis Use **Geometry > All bond angle analysis** to calculate angle statistics for every connected `atom_i-atom_j-atom_k` angle in an XYZ trajectory. The central atom is `atom_j`. The tool first detects connected bonds from the first frame, then builds every angle where `atom_i` and `atom_k` are both connected to the central atom.

For a solute in a solvent box, provide the solute atom indices. The program then builds angles only from solute atoms, excluding any angle involving solvent atoms.

Inputs:

- `trajectory.xyz` or `TRAJEC.xyz` trajectory file.
- **Maximum connection distance (A)**, default `1.5`.
- **Solute atom indices**, for example `1 2 3 4 5`; leave blank to use all atoms.
- Output TXT filename, default `all_angles_analysis.txt`.

Step-by-step:

1. Open **Geometry > All bond angle analysis**.
2. Click **Browse** and select the trajectory XYZ file.
3. Enter the maximum connection distance used to detect first-frame bonds.
4. Enter solute atom labels if solvent atoms must be excluded.
5. Enter the output filename or keep `all_angles_analysis.txt`.

6. Click **Analyze**.
7. Wait until the text area reports that the analysis is completed.
8. Inspect the statistics file and triplet mapping file.

Output files:

- `all_angles_analysis.txt`: three whitespace-separated columns.
- `all_angles_analysis_triplets.txt`: row-to-atom-triplet mapping.

The statistics file contains:

```
# average_angle_degrees variance_degrees2 standard_deviation_degrees
```

Each following row contains one connected angle:

```
average_angle  population_variance  standard_deviation
```

The mapping file identifies each `atom_i atom_j atom_k` triplet and reports the first-frame angle in degrees.

Dihedral Angle Analysis Use this tool to calculate dihedral angles across a trajectory, dihedral distributions, and free-energy profiles.

Inputs:

- Maximum angle for the distribution.
- Simulation time step in atomic units.
- Sampling interval in frames.
- Simulation temperature in K.
- Four atom labels, for example `1 2 3 4`.
- Histogram bin width in degrees.
- `TRAJEC.xyz` trajectory file.

Options:

- **Show plots at the end**.
- **Save CSV outputs**.
- **Wrap dihedral to [-180, 180]**: reports wrapped dihedral angles instead of a 0 to 360 degree range.

All Dihedral Angle Analysis Use **Geometry > All dihedral angle analysis** to calculate statistics for every connected `atom_i-atom_j-atom_k-atom_l` dihedral path in an XYZ trajectory. The tool detects first-frame connectivity, then builds dihedrals around every connected central bond `atom_j-atom_k`.

For a solute in a solvent box, provide the solute atom indices. The program then builds only solute dihedrals and ignores all solvent atoms.

Dihedral convention:

- Dihedrals are signed angles in degrees.
- The range is `(-180, 180]`.

- This convention matches the single **Dihedral angle analysis** tool.

Inputs:

- `trajectory.xyz` or `TRAJEC.xyz` trajectory file.
- **Maximum connection distance (A)**, default `1.5`.
- **Solute atom indices**, for example `1 2 3 4 5`; leave blank to use all atoms.
- Output TXT filename, default `all_dihedral_analysis.txt`.

Step-by-step:

1. Open **Geometry > All dihedral angle analysis**.
2. Click **Browse** and select the trajectory XYZ file.
3. Enter the maximum connection distance used to detect first-frame bonds.
4. Enter solute atom labels if solvent atoms must be excluded.
5. Enter the output filename or keep `all_dihedral_analysis.txt`.
6. Click **Analyze**.
7. Wait until the text area reports that the analysis is completed.
8. Inspect the statistics file and quadruplet mapping file.

Output files:

- `all_dihedral_analysis.txt`: three whitespace-separated columns.
- `all_dihedral_analysis_quadruplets.txt`: row-to-atom-quadruplet mapping.

The statistics file contains:

```
# average_dihedral_degrees variance_degrees2 standard_deviation_degrees
```

Each following row contains one connected dihedral:

```
average_dihedral  population_variance  standard_deviation
```

The mapping file identifies each `atom_i atom_j atom_k atom_l` quadruplet, reports the signed dihedral convention, and reports the first-frame dihedral in degrees.

Hydrogen Bond Analysis Use this tool to analyze hydrogen bonds from a trajectory.

Inputs:

- Maximum hydrogen bond length.
- Simulation time step in atomic units.
- Sampling interval in frames.
- Simulation temperature in K.
- Histogram bin widths for hydrogen bond length and angle.
- Donor atom label, hydrogen atom label, and acceptor atom label.
- `TRAJEC.xyz` trajectory file.

Click **Exec** or the main action button in the tool window to run the calculation.

Input Builders

CPMD Inputs Use **Inputs > CPMD Inputs** to create CPMD input files from an XYZ geometry.

Fields:

- **Prefix file name**: prefix used for generated files, for example `VitC`.
- **Total charge on the system**.
- **Periodic box parameters**: `a b c cosA cosB cosC`, separated by spaces.
- **Energy Cutoff (Ry)**.
- **Dual for rho expansion**.
- **Cell symmetry**:
  - `1 - CUBIC a=b=c alpha=beta=gamma=90`
  - `6 - TETRAGONAL a=b!=c alpha=beta=gamma=90`
  - `8 - ORTHORHOMBIC a!=b!=c alpha=beta=gamma=90`
- XYZ input file.

Action:

- **Input Builder**: generates the CPMD input files.

Surface Hopping Input Builder Use **Inputs > SH Input Builder** to prepare CPMD surface hopping and TDDFT input files.

Fields:

- Starting frame.
- Number of skipped frames.
- Final frame.
- Number of electronic states.
- Initial state for SHTDDFT.
- Prefix file name.
- Molecular dynamics time step.
- Total system charge.
- Periodic box parameters.
- Energy cutoff.
- Dual for rho expansion.
- Cell symmetry.
- Maximum number of steps.
- `GEOMETRY.xyz` input file.
- `TRAJECTORY` input file.

Actions:

- **Read SH parms**: reads and displays the entered parameters.
- **SH Input Builder**: generates surface hopping input files.

- **Help**: opens the tool help text.

Collision Input Use **Inputs > Collision Input** to create new geometry/velocity files for collision molecular dynamics.

Fields:

- **MD engine**: select `cpmd` or `gqteaMD`.
- **Attacker atom indices**: one or more 1-based atom labels, for example `12 13 14`.
- **Initial velocity**: atomic units for CPMD; the label changes for gqteaMD.
- **Target xyz coordinates**: target point in Angstrom, for example `10.0 12.5 8.0`.
- `GEOMETRY` or `GEOMETRY.xyz` file.

Options:

- **Zero velocities of non-attacker atoms**.
- **Compute center-of-mass KE**.

Typical outputs:

- `summary.txt`
- `newGeometry.xyz`
- `with-vibration-GEOMETRY`
- `without-vibration-GEOMETRY`
- `gqteaMD-GEOMETRY`
- `gqteaMD-without-vibration-GEOMETRY`

For CPMD collision simulations, the help text recommends restarting CPMD with:

```
RESTART WAVEFUNCTION COORDINATES VELOCITIES GEOFILE LATEST
```

Quantum ESPRESSO cp.x Input Builder Use **Inputs > cp.x Input Builder** to build Quantum ESPRESSO `cp.x` input files.

Main actions:

- **Open XYZ**: loads atomic coordinates from an XYZ file.
- **Preview input**: opens a preview of the generated input.
- **Save cp.x input**: writes the input file.
- **Help**: opens the bundled `cpx_input_description.pdf`.

Important behavior:

- If the output name is blank, the default name is based on the XYZ file, such as `<xyzname>_cp.in`.
- `ATOMIC_SPECIES` is inferred from XYZ symbols.
- Default pseudopotential names use the pattern `<Element>.UPF`; edit them if your pseudopotential filenames differ.

- `ATOMIC_POSITIONS` can be written in `alat`, `bohr`, `crystal`, or `angstrom`.
- Optional sections include velocities, cell parameters, constraints, occupations, atomic forces, Wannier plotting, and autopilot text.

ORCA Input Builder Use **Inputs > ORCA Input Builder** to prepare ORCA `.inp` files from XYZ coordinates.

Main actions:

- **Open XYZ**.
- **Preview ORCA input**.
- **Save ORCA input**.
- **Clear form**.
- **Research notes**.

Common settings:

- Output filename. If blank, the default is `<xyzname>_orca.inp`.
- Title/comment line.
- Method, basis, task, charge, multiplicity, and coordinate mode.
- Extra simple keywords.
- Parallel settings such as `%pal nprocs`.
- Memory setting such as `%maxcore`.
- SCF, geometry, CPCM, and advanced ORCA blocks.

gqteaMD Input Builder Use **Inputs > gqteaMD Input Builder** to generate a gqteaMD TOML input file.

Main actions:

- **Open XYZ**.
- **Preview TOML**.
- **Save TOML**.
- **Clear form**.
- **Manual notes**.

Common settings:

- Output filename. If blank, the default is `<xyzname>_gqteaMD.toml`.
- Starting XYZ path.
- Force provider.
- Time step, number of steps, temperature, output intervals, restart behavior, and force-field sections.
- For xTB force provider, **OMP threads** writes `omp_num_threads` to the TOML so gqteaMD can set `OMP_NUM_THREADS` before xTB runs.
- Optional UFF or classical parameter blocks.

Vanderbilt runatom.x Input Builder Use **Inputs > Vanderbilt runatom.x Input Builder** to generate Vanderbilt `runatom.x` input files.

Controls:

- Mode selection for all-electron or generation input.
- **Also generate Makefile** option.
- **Generate** button.
- **Help** button.

Depending on the selected mode, the form shows the relevant atomic, pseudopotential, and state fields. The generated input is saved through a Save dialog, and a Makefile can be written alongside it.

Structural Tools

Radial Distribution Function Use **Structural > Radial Distribution Function** to calculate an RDF for selected atoms.

Typical inputs:

- Maximum radius.
- Bin width.
- Atom labels or atom types, depending on the selected workflow.
- Trajectory file, usually `TRAJEC.xyz`.

Action:

- **Read Params**: displays entered settings.
- **RDF calculation**: runs the calculation.

Mean Residence Time Use **Structural > Mean Residence Time** for MRT analysis of an XYZ trajectory.

The newer MRT tool supports:

- XYZ trajectory selection.
- Time step and time-step unit.
- Cutoff radius.
- Tolerance frames.
- Reference definition.
- Observed atom definition.
- Run, export, and clear actions.

Outputs commonly include survival/correlation data, event durations, and text summaries.

Legacy Mean Residence Time The legacy MRT workflow is retained for compatibility with older analyses.

Inputs include:

- Shell inner radius.
- Shell outer radius.

- Simulation time step.
- Sampling interval.
- Tolerance frames.
- Atom labels to exclude.
- Atom labels at the shell center.
- Element symbol to investigate.
- `TRAJEC.xyz` trajectory file.

Typical outputs:

- `mrt.dat`
- `mrt_total.dat`
- `mrt_summary.dat`

Autocorrelation Function Use **Structural > Autocorrelation function** to calculate velocity autocorrelation and related data.

Inputs:

- Start frame.
- Number of frames to use.
- `GEOMETRY.xyz` file.
- `TRAJECTORY` file.

Options:

- Save generated `newTRAJEC.xyz`.
- Compute PSD from the VAF.

Typical outputs:

- `PAF.dat`
- `VAF.dat`
- `PSD.dat` when PSD is enabled.
- `newTRAJEC.xyz` when the trajectory export option is enabled.

Single Solute Solvent Box Use **Structural > Single Solute solvent box** to create a box containing one solute and inserted solvent molecules.

Inputs and options:

- Box lattice vectors.
- Target density.
- Maximum insertion attempts.
- van der Waals scaling factor.
- Extra wall padding or minimum distance.
- Solvent XYZ file.
- Solute XYZ file.
- Randomly rotate solvent.
- Calculate density.

- Periodic minimum-image clash detection.
- Include centered solute.

Typical outputs:

- `single_solute_solvent.txt`
- `single_solute_box_cmass.xyz`
- A generated solvent-box XYZ file.

If the target number of solvent molecules cannot be inserted, the summary reports a warning and the number of successful insertions.

Mixture of Two Solvent Box Use **Structural > Mixture of two solvent box** to build a mixed-solvent box, optionally with a centered solute.

Inputs and options:

- Solvent A XYZ file.
- Solvent B XYZ file.
- Optional solute XYZ file.
- Mixture composition.
- Box/density and insertion settings.
- Random solvent rotation.
- Density calculation.
- Periodic minimum-image clash detection.
- Insert centered solute.

Outputs include a generated XYZ structure and a TXT summary with composition and insertion statistics.

Shared-Wall Double Solvent Box Use **Structural > Shared-wall double solvent box** to build two adjacent solvent regions that share one face. The module is intended for interface models, bilayer-like solvent arrangements, or any setup where two different solvent environments should occupy neighboring slabs in the same simulation box.

The generated structure is a single combined XYZ file. Box 1 occupies the lower z region, from `z = 0` to `z = c1`, and Box 2 occupies the upper z region, from `z = c1` to `z = c1 + c2`. The two boxes share the same `a` and `b` lattice dimensions, but each box has its own z height. No wall atoms or boundary markers are written at the shared interface.

How it works:

- Reads one or two solvent XYZ templates for each box.
- Optionally reads one solute XYZ template for each box.
- Recenters each molecule template, using center of mass when atomic masses are available.
- Estimates the target number of solvent molecules from the requested density, slab volume, solvent composition, and optional solute mass.

- Randomly inserts solvent molecules into Box 1 and then Box 2.
- Optionally randomizes solvent orientation before insertion.
- Keeps atoms inside the assigned z slab and applies the requested wall padding at the outside boundaries.
- Rejects placements that clash using van der Waals radii scaled by the selected vdW factor.
- Uses the full combined box for periodic minimum-image clash detection when that option is enabled.
- Reports insertion statistics and warns if the target number of molecules cannot be reached.

Shared geometry inputs:

- **Shared a b**: two lattice dimensions used by both boxes.
- **Box 1 c**: z height of the lower region.
- **Box 2 c**: z height of the upper region.
- **Periodic minimum-image clash detection**: checks overlaps across periodic boundaries using the full `a x b x (c1 + c2)` box.

Per-box inputs and options:

- Wall padding.
- Target density in g/cm^3.
- Solvent A and Solvent B composition percentages.
- Maximum insertion attempts.
- vdW scale.
- Optional random seed.
- Optional solute gap.
- Rotate solvent molecules during insertion.
- Include a centered solute.
- Solvent A, Solvent B, and optional solute XYZ files.

Typical outputs:

- `shared_wall_box.xyz`
- `shared_wall_box.txt`

Example workflow:

1. Open **Structural > Shared-wall double solvent box**.
2. Enter shared lattice values, for example `30.0 30.0`, in **Shared a b**.
3. Enter `25.0` for **Box 1 c** and `25.0` for **Box 2 c**. The combined box will have `c = 50.0`, with the shared interface at `z = 25.0`.
4. For Box 1, set **Density** to `0.95`, **Solvent A (%)** to `100`, **Solvent B (%)** to `0`, and browse for a Box 1 Solvent A XYZ file such as water.
5. For Box 2, set **Density** to `0.80`, **Solvent A (%)** to `100`, **Solvent B (%)** to `0`, and browse for a Box 2 Solvent A XYZ file such as an organic solvent.
6. Leave **Rotate Box 1** and **Rotate Box 2** enabled unless you need fixed solvent orientations.

7. Enable **Box 1 solute** or **Box 2 solute** only if that slab should contain a centered solute, then browse for the corresponding solute XYZ file and set a solute gap if needed.
8. Click **Build Shared-Wall Box**.
9. Inspect the progress and summary panel. If the summary warns that not all solvent molecules were inserted, try a larger slab, lower density, smaller vdW scale, or more insertion attempts.

Thermo Tools

Classical Rate Constant Use **Thermo > Classical rate constant** to calculate rate constants from classical transition-state-style inputs.

Inputs include numeric parameters such as temperature range and activation-energy-related fields. The tool validates each field and asks for an output directory.

Typical outputs:

- `rate_constant_vs_temperature.dat`
- `LnK_vs_invT.dat`

General Tools

3D Molecular Viewer Use **Tools > 3D Molecular Viewer** to inspect XYZ structures and trajectories.

Controls include:

- Browse for an XYZ file.
- Set upper bond-length limit.
- Visualization style.
- Atom display style.
- Atom scale.
- Bond thickness.
- Rotate the molecular system around the X, Y, and Z axes.
- Bond mode.
- Show atom numbers.
- Show atomic symbols.
- Measure distance, angle, or dihedral by entering atom labels.
- Set periodic box dimensions.
- Choose how the displayed periodic box is centered.
- Show or hide the box.
- Step through trajectory frames.
- Play/pause trajectory animation.
- Set frame step and playback speed.
- Save the current frame as XYZ.

The saved current frame includes the atoms from the displayed frame and a comment noting the frame number.

The rotation controls let you rotate the displayed molecular system around the X, Y, and Z axes without changing the coordinates stored in the loaded structure or trajectory. Enter rotation increments in degrees and click **Apply** to add those increments to the current molecular orientation. Click **Reset** to return the displayed molecular orientation to zero rotation. The `<` and `>` buttons beside each axis start continuous rotation in the negative or positive direction; click the same arrow again to stop it, or click another arrow to switch to that axis and direction.

When the periodic box is shown, it rotates with the molecular system. The **Box center** selector controls where the displayed orthorhombic box is placed:

- **Geometric center**: centers the box on the molecular system's geometric center.
- **Bottom at z=0**: centers the box in the a-b plane while placing the bottom face of the box at `z = 0`.

Bond mode controls how the viewer updates connectivity while stepping through or playing a trajectory:

- **Static first frame**: calculates bonds from frame 0 and reuses that same connectivity for all frames. This is useful when you want stable visual connectivity during normal vibrations or rotations.
- **Dynamic cached**: recalculates bonds for each frame and stores the result for faster revisiting of frames. This is useful for trajectories where bonds may form or break and you still want smooth playback.
- **Dynamic live**: recalculates bonds every time the current frame is rendered, without using cached bond lists. This is useful when you are actively changing bond-length settings or want the freshest possible connectivity during inspection.

To use this feature, open **Tools > 3D Molecular Viewer**, load an XYZ trajectory, then choose the desired option from the **Bond mode** selector in the playback/performance controls. The viewer updates the displayed bonds using the selected mode.

Example: if a trajectory shows two atoms separating during a dissociation event, choose **Dynamic cached** or **Dynamic live**. As you step through the frames, the bond disappears when the atom distance exceeds the current upper bond-length limit. If you choose **Static first frame**, that bond remains visible throughout playback because the viewer keeps the first-frame connectivity.

Energy Plots Use **Tools > Energy plots** to visualize CPMD or gqteaMD energy files.

For CPMD ENERGY files, plot options include:

- Fictitious and ionic kinetic energy.
- Temperature.
- Kohn-Sham potential energy.
- Kohn-Sham plus ionic kinetic energy.
- Total energy.
- CPU time by step.

X-axis units:

- Steps.
- Femtoseconds.
- Picoseconds.

For gqteaMD energy files, select the appropriate file type and x-axis option in the tool.

Molecular Axis Alignment Use **Tools > Molecular Axis Alignment** to align an XYZ molecule along a chosen axis and save a new XYZ file.

Actions:

- **Browse**: select input XYZ.
- **Save As**: choose output path.
- Run the alignment action from the tool window.

Select Frames Use **Tools > Select Frames** to extract a frame range from `TRAJEC.xyz`.

Inputs:

- Starting frame.
- Number of frames to skip between collected frames.
- Stop frame.
- Optional Gaussian settings.
- Charge and multiplicity when Gaussian input generation is enabled.

Outputs:

- `selected_frames.xyz`
- Optional Gaussian `.gjf` files.

Frame Selection by Interatomic Distance Range Use **Tools > Frame selection by interatomic distance range** to extract frames where the distance between two atoms falls in a selected interval.

Inputs:

- Minimum distance.
- Maximum distance.
- Two atom labels.
- `TRAJEC.xyz`.

Typical outputs:

- A selected-frames XYZ file in the input directory.
- A TXT file with selected-frame distances and statistics.
- `frame_closest_to_average.xyz`.

This tool is useful after bond-length free-energy analysis, for example to select frames near the minimum Helmholtz free energy.

CPMD Input to XYZ Converter Use **Tools > CPMD Input to XYZ Converter** to extract coordinates from a CPMD input file and write an XYZ file.

Actions:

- Browse for a CPMD input file.
- Enter an output filename.
- Click **Convert**.

The converter reads `CELL` values from the `&SYSTEM` block and atom coordinates from the `&ATOMS` block.

SH Geometry Analyzer Use **Tools > SH Geometry Analyzer** to group surface hopping trajectory frames by electronic state.

Inputs:

- Number of states.
- Root directory containing simulation subfolders.

Each simulation subfolder should contain:

- `SH_STATE.dat`
- `TRAJEC.xyz`

Outputs:

- `stateX.xyz` files inside each subfolder.
- Consolidated `stateX.xyz` files in the root directory.
- `sh_avg_perc.dat` with average state occupancy percentages.

If a subfolder is missing `SH_STATE.dat` or `TRAJEC.xyz`, the tool reports a warning and skips that subfolder.

Convert cp.x .pos File to trajec.xyz Use **Tools > Convert .pos file to trajec.xyz* to convert a Quantum ESPRESSO `cp.x` position trajectory to XYZ.

Inputs:

- `cp.x` input file, used to read atom labels and `nat`.
- `*.pos` trajectory file.

Output:

- XYZ trajectory file with coordinates converted from Bohr to Angstrom.

Compute Forces from cp.x .for File Use **Tools > Compute forces from cp.x .for file* to convert a Quantum ESPRESSO force trajectory into a trajectory force file.

Inputs:

- `cp.x` input file.
- `*.for` force trajectory file.

Click **Convert** to generate the converted force output.

Coordinate Converter Use **Tools > Coordinate converter** to convert between molecular coordinate file formats.

Controls:

- Input format selection.
- Output format selection.
- Browse for input file.
- Choose output file.
- **Convert**.

Supported formats are determined by the converter UI and parser implementation. The converter writes a new file rather than modifying the original.

Convert Coordinates from Angstrom to Bohr Use **Tools > Convert coordinates from A to Bohr** to convert XYZ coordinates from Angstrom to Bohr.

Inputs:

- Input XYZ file.
- Output filename.

Action:

- **Convert**.

Unit Converter Use **Tools > Unit Converter** for scalar unit conversion.

Controls:

- Category.
- Value.
- From unit.
- To unit.
- **Convert**.

If the value is blank or a unit is unsupported, the converter shows an error explaining what to fix.

Configuration

gqteaWinToga does not currently use a persistent user configuration file. Settings are entered directly in each tool window for each run.

Customization points:

- Choose output directories using **Browse**, **Save Dir**, **Save As**, or Save dialogs when available.
- Leave output-name fields blank in some builders to use automatic names such as `<xyzname>_cp.in`, `<xyzname>_orca.inp`, or `<xyzname>_gqteaMD.toml`.
- Toggle optional outputs such as CSV files, plot display, Gaussian input generation, random solvent rotation, density calculation, and minimum-image clash detection.
- Edit generated input files before running external simulation engines, especially pseudopotential names, basis settings, memory, processor counts, and advanced blocks.

Generated files are usually written in the selected output directory or beside the selected input file. Check the status text in each tool window after running a calculation.

Practical Examples

Example 1: Find a Bond-Length Free-Energy Minimum

1. Open **Geometry > Bond length analysis**.
2. Load `TRAJEC.xyz`.
3. Enter:

```
Maximum r: 3.0
Simulation time step: 5
Sampling interval: 1
Temperature: 300
Atom labels: 1 2
Histogram bin width: 0.02
```

4. Click **Analyze**.
5. Open the generated `free_energy_*.dat` or plot to locate the minimum.
6. Use **Tools > Frame selection by interatomic distance range** to extract frames near that distance.

Example 2: Prepare a CPMD Collision Restart

1. Run CPMD for one step and obtain `GEOMETRY.xyz`.
2. Open **Inputs > Collision Input**.
3. Select `cpmd` as the MD engine.
4. Enter attacker atom labels, initial velocity, and target coordinates.

5. Browse to `GEOMETRY.xyz`.
6. Click **Input Builder**.
7. Rename the generated collision geometry as needed for your CPMD restart.
8. Restart CPMD with the `GEOFILE` keyword so the new velocities are used.

Example 3: Group Surface Hopping Frames by State

1. Organize simulation folders under one root directory.
2. Ensure each subfolder contains `SH_STATE.dat` and `TRAJEC.xyz`.
3. Open **Tools > SH Geometry Analyzer**.
4. Enter the number of states.
5. Browse to the root directory.
6. Click **Extract Frames**.
7. Review `stateX.xyz` files and `sh_avg_perc.dat`.

Example 4: Convert a cp.x Trajectory

1. Open **Tools > Convert .pos file to trajec.xyz.*
2. Browse to the `cp.x` input file.
3. Browse to the `*.pos` file.
4. Click **Convert**.
5. Use the generated XYZ trajectory in the viewer or analysis tools.

Example 5: Analyze All Solute Bond Distances

Suppose atoms `1 2 3 4` are the solute and the remaining atoms are solvent. A small trajectory might begin like this:

```
5
frame 1
C  0.0  1.0  0.0
C  0.0  0.0  0.0
C  1.0  0.0  0.0
H  1.0  0.0  1.0
O  0.0  0.5  0.0
5
frame 2
C  0.0  1.1  0.0
C  0.0  0.0  0.0
C  1.1  0.0  0.0
H  1.0  0.0  1.1
O  0.0  0.5  0.0
```

Atom `5` is a solvent atom close to the solute, but it must not contribute to the solute bond analysis.

1. Open **Geometry > All bond distance analysis**.
2. Browse to the trajectory file.

3. Set **Maximum connection distance (A)** to `1.5`.
4. Set **Solute atom indices** to:

```
1 2 3 4
```

5. Keep the output filename as:

```
all_bond_analysis.txt
```

6. Click **Analyze**.

The statistics file has three columns:

```
# average variance standard_deviation
1.05000000       0.00250000       0.05000000
```

The pair mapping file tells which atom pair belongs to each row:

```
# frames_used 2
# atom_scope solute_atoms
# solute_atom_indices 1 2 3 4
# row atom_i atom_j element_i element_j first_frame_distance
1       1       2       C        C       1.00000000
```

Only solute-solute pairs are reported.

Example 6: Analyze All Solute Bond Angles

Use the same idea for all connected solute angles. For a water-like solute with atoms `1 2 3` and a nearby solvent atom `4`:

```
4
frame 1
H  0.0  0.0  0.0
O  1.0  0.0  0.0
H  1.0  1.0  0.0
H  1.0  0.5  0.0
4
frame 2
H  0.0  0.0  0.0
O  1.0  0.0  0.0
H  0.5  0.8660254  0.0
H  1.0  0.5  0.0
```

Atom `4` is solvent and should be excluded.

1. Open **Geometry > All bond angle analysis**.
2. Browse to the trajectory file.
3. Set **Maximum connection distance (A)** to `1.1`.
4. Set **Solute atom indices** to:

```
1 2 3
```

5. Keep the output filename as:

```
all_angles_analysis.txt
```

6. Click **Analyze**.

The statistics file has three columns:

```
# average_angle_degrees variance_degrees2 standard_deviation_degrees
74.99999995     225.00000163      15.00000005
```

The triplet mapping file identifies the angle:

```
# frames_used 2
# atom_scope solute_atoms
# solute_atom_indices 1 2 3
# row atom_i atom_j atom_k element_i element_j element_k first_frame_angle_degrees
1       1       2       3       H       O       H      90.00000000
```

The angle is interpreted as `atom_i-atom_j-atom_k`, so atom `2` is the central atom in this example.

Example 7: Analyze All Solute Dihedral Angles

For a solute chain `1-2-3-4` in a solvent environment, use **All dihedral angle analysis**. Example trajectory:

```
5
frame 1
C  0.0  1.0  0.0
C  0.0  0.0  0.0
C  1.0  0.0  0.0
H  1.0  0.0  1.0
O  0.0  0.5  0.0
5
frame 2
C  0.0  1.0  0.0
C  0.0  0.0  0.0
C  1.0  0.0  0.0
H  1.0  0.0 -1.0
O  0.0  0.5  0.0
```

Atom `5` is solvent and should not be used to build dihedral paths.

1. Open **Geometry > All dihedral angle analysis**.
2. Browse to the trajectory file.
3. Set **Maximum connection distance (A)** to `1.1`.
4. Set **Solute atom indices** to:

```
1 2 3 4
```

5. Keep the output filename as:

`all_dihedral_analysis.txt`

6. Click **Analyze**.

The statistics file has three columns:

```
# average_dihedral_degrees variance_degrees2 standard_deviation_degrees
0.00000000    8100.00000000      90.00000000
```

The quadruplet mapping file identifies the dihedral:

```
# frames_used 2
# dihedral_convention signed_degrees_minus180_to_180
# atom_scope solute_atoms
# solute_atom_indices 1 2 3 4
# row atom_i atom_j atom_k atom_l element_i element_j element_k
element_l first_frame_dihedral_degrees
1        1        2        3        4        C        C        C
H     -90.00000000
```

Dihedrals are reported as signed degrees using the range `(-180, 180]`.

Troubleshooting

"No file was selected!"

You closed a file dialog without choosing a file. Click **Browse** again and select the required input.

"Please input a valid value for ..."

One of the numeric fields is blank or contains text that cannot be converted to a number. Check the field named in the message and enter a valid value.

"Invalid format for ... Please input exactly two atom labels."

The atom-label field has the wrong number of labels. Bond analysis needs two labels, bond angle needs three, and dihedral angle needs four. Separate labels with spaces, not commas.

"Failed to open file" or "Failed to read file"

The selected file could not be opened or parsed. Confirm that:

- The file still exists.
- You have permission to read it.
- The file format matches the selected tool.
- XYZ files begin with the atom count and contain valid coordinate lines.

"The file is empty!"

The selected file has no readable content. Choose a valid geometry, trajectory, or input file.

"Invalid line format in TRAJECTORY file"

The trajectory file does not match the expected numeric layout. Check that the file came from the expected simulation program and has not been truncated or edited incorrectly.

"The stop frame must be less than the total number of frames"

The selected stop frame is outside the trajectory. Use the frame count displayed after browsing for the trajectory, then choose a smaller stop frame.

ENERGY file format errors

The CPMD energy plotter expects a valid ENERGY file with eight numeric columns. If plotting fails, verify that the file is complete and that you selected the correct plot type.

Solvent box insertion warnings

If the solvent box builder cannot insert all requested molecules, the target density or minimum-distance settings may be too restrictive. Try one or more of the following:

- Increase the box dimensions.
- Lower the target density.
- Reduce the minimum distance or van der Waals scaling factor.
- Increase the maximum number of insertion attempts.

Missing `SH_STATE.dat` or `TRAJEC.xyz`

The SH Geometry Analyzer skips subfolders that do not contain both files. Add the missing files or remove incomplete subfolders from the selected root directory.

Molecular viewer does not open or shows no 3D view

The viewer depends on OpenGL and GLFW. Confirm that `glfw`, `PyOpenGL`, and `PyOpenGL-accelerate` are installed and that your graphics driver supports OpenGL in the current desktop session.

Packaged Windows executable cannot find GLFW

When packaging with `auto-py-to-exe`, include the GLFW DLL manually if needed. It is usually located at a path similar to:

```
venv\Lib\site-packages\glfw\glfw3.dll
```

Version, Contributors, and License

Version shown by the application:

```
gQTEA-0.3.1 Molecular Analysis Toolkit
```

The launcher source notes that the program was revised and updated in March 2026.

Core development team:

- Ademir J. Camargo - ajc@ueg.br
- Valter H. C. Silva - fatioleg@ueg.br
- Solemar S. Oliveira - solemar@ueg.br
- Hamilton B. Napolitano - hamilton@ueg.br
- Luciano Ribeiro - lribeiro@ueg.br
- Flavio O. Sanches - flavio.neto@ifg.edu.br

No license file is currently included in the source tree. Before distributing the program publicly, add a `LICENSE` file so users know how they may use, modify, and redistribute the code.