KinBot 2.0 Manual

Authors: Judit Zádor and Ruben Van de Vijver

December 2018

Contents

1 Introduction

KinBot is a code to automatically locate stationary points on PES's for H, C, O and S. More information can be found on the website https://kinbot.sandia.gov and on the GitHub repository https://github.com/zadorlab/KinBot.

2 Installation

2.1 Prequisites

To install KinBot, the following software is necessary

- Pybel (Python distribution of OpenBabel): https://openbabel.org/docs/dev/UseTheLibrary/PythonInstall.html
- RDKit including python bindings: https://www.rdkit.org/docs/Install.html
- A PBS or SLURM workload manager
- A quantum chemistry package, currently all testing has been done with Gaussian, which is the preferred software to use. The use of NWChem and QChem is being developed and tested.
- The fork of ASE available on http://github.com/zadorlab/ase

Optionally:

- MESS (http://tcg.cse.anl.gov/papr/codes/mess.html) or MESMER (https://sourceforge.net/projects/mesmer/) to calculate rate coefficients
- PESViewer (https://github.com/rubenvdvijver/PESViewer) to visualize the PES

IMPORTANT:

Pybel, RDKit and ASE need to be in the python path.

2.2 KinBot

Download the code from https://github.com/zadorlab/KinBot

Run the setup.py file:

```
python setup.py build
python setup.py install
```

3 Testing

There are several test classes in the tests/ directory.

4 Running KinBot

4.1 Input

Make a json file (e.g. called *input.json*) in the directory where the calculations will be run

The json dictionary lists the options and parameters of the KinBot run. The parameters are given below.

4.1.1 General parameters

PARAMETER	DEFAULT	USE	DESCRIPTION

title	N/A	Required	Title of the current KinBot run
verbose	0	Optional	Print verbose information to the log file

4.1.2 Reactant information

PARAMETER	DEFAULT	USE	DESCRIPTION
smiles	N/A	Required if no structure is given	SMILES of the starting well
structure	N/A	Required if no smiles is given	Structure of the starting well written in one vector in which each atom occupies four consecutive elements: the atom symbol followed by the x, the y and the z Cartesian coordinates of the atom.
charge	0	Optional	Charge of the PES
mult	0	Required	Multiplicity of the PES

4.1.3 Which steps to take

PARAMETER	DEFAULT	USE	DESCRIPTION
reaction_search	1	Optional	Boolean tells whether to search for reactions
families	['all']	Optional	List of the reaction families to use in the reaction search. If 'all' is put in the list, all the families are used.
homolytic_scissions	0	Optional	Break all single non-cyclic bonds and calculate the energies of the products which are showed on the final plots
barrier_threshold	0.	Required	Barrier threshold above which barriers are neglected
scan_step	30	Optional	Number of scan points for reaction families where no geometry template is given but rather the TS is searched for via scanning the energy as a function of the bond length of two atoms
pes	0	Optional	Boolean tells whether to search for the reactions of a specific well or for the full multiwell PES
simultaneous_kinbot	5	Optional	Maximum number of simultaneous kinbot runs during a full pes search

high_level	0	Optional	Define whether high-level optimizations and frequency calculations are necessary, if not keep low-level results are used
conformer_search	1	Optional	Boolean tells whether to perform conformational search
rotor_scan	0	Optional	Boolean wells whether to perform 1D hindered rotor scans
nrotation	12	Optional	Number of energy evaluations along a 1D hindered rotor scan
me	0	Optional	Automatically run the master equation calculations
rotation_restart	3	Optional	Number of HIR restarts in case a lower energy point gets found
max_dihed	5	Optional	Maximum number of diherals for which exhaustive conformation searches are done
random_conf	500	Optional	Number of random conformers in case no exhaustive search is done

4.1.4 Quantum chemistry information

PARAMETER	DEFAULT	USE	DESCRIPTION
qc	gauss	Optional	Which quantum chemistry package to use: available ones are gauss for Gaussian, nwchem and qchem
methodclass	dft	Optional	Class of methods used, other available values are mp2 or scf
gaussian_command	g09	Required if Gaussian is used as quantum chemistry package	Command to run Gaussian
method	b3lyp	Optional	Low-level method for reaction searches and conformational searches
basis	6-31G	Optional	Low-level basis set for reaction searches and conformational searches

high_level_method	M062X	Optional	High-level method for energy and frequency calculations
high_level_basis	6-311++G(d,p)	Optional	High-level basis set for energy and frequency calculations
integral	*No default*	Optional	Integral grid for Gaussian, which is only used for the high-level calculations

4.1.5 Computational environment

PARAMETER	DEFAULT	USE	DESCRIPTION
queuing	pbs	Optional	pbs or slurm
scratch	/scratch/jzador	Optional	Scratch directory of the calculations
username	jzador	Required	User name of the current user
queue_name	medium	Required	Queue name of the queue the calculations will be submitted to
slurm_feature	knl	Optional	What comes with -C in SLURM
ppn	1	Optional	Number of cores per quantum chemistry calculation
zf	4	Optional	Number of characters in file numbers

4.1.6 Master equation parameters

PARAMETER	DEFAULT	USE	DESCRIPTION
me_code	mess	Optional	Which ME code to use, mess or mesmer
mess_command	mess	Optional	Command to run MESS
TemperatureList	1000	Required if MESS is used	See MESS manual
PressureList	760	Required if MESS is used	See MESS manual
EnergyStepOverTemperature	.2	Required if MESS is used	See MESS manual
ExcessEnergyOverTemperature	30	Required if MESS is used	See MESS manual
ModelEnergyLimit	400	Required if MESS is used	See MESS manual
CalculationMethod	direct	Required if MESS is used	See MESS manual
ChemicalEigenvalueMax	0.2	Required if MESS is used	See MESS manual
EnergyRelaxationFactor	200	Required if MESS is used	See MESS manual

EnergyRelaxationPower	0.85	Required if MESS is used	See MESS manual
EnergyRelaxationExponentCutoff	12	Required if MESS is used	See MESS manual
Epsilons	7.08,310.387	Required if MESS is used	See MESS manual
Sigmas	2.576,6.000	Required if MESS is used	See MESS manual
Masses	4.0,87.0	Required if MESS is used	See MESS manual
mesmer_command	mesmer	Optional	Command to run MESMER

4.2 Running KinBot or pes

To run KinBot, go to the directory with your input file (called for example *input.json*) and write:

```
kinbot input.json
```

Optionally, to explore a full PES instead of only one well, run

```
pes input.json
```

For the pes script, directories will be created for each well into which the KinBot script will be run. If you already have one or multiple wells explored, start pes from the parent directory and rename the directories to the chemid of the well that was explored.

4.3 Restarting KinBot or pes

Both with KinBot and though the pes script, it is possible to restart a calculation where it left off after a previous run. It is however important to keep certain parameters unchanged during restart, else there is no guarantee the code will run cleanly. A few things to keep in mind:

- The numbering of atoms is key when restarting kinbot or pes. To restart a pes from a different well, use the xyz structure instead of defining smiles in the input files.
- Parameters that can be changed:
    - homolytic_scissions
    - barrier_threshold
    - simultaneous_kinbot
    - high_level (when using pes, the barrier_threshold that is propagated thoughout the calculations will depend on whether high-level calculations are done or not)
    - conformer_search (only switch this on if no high-level calculations are done yet)
    - rotor_scan

Additionally, the pes script can be run without calling KinBot. This feature allows to quickly read everything that has been done so far and postprocess all the calculations. For this the no-kinbot option needs to be added to the command line:

```
pes input.json no-kinbot
```

In this mode, it is also possible to highlight all pathways between to species. The chemids are the ids of the molecules which can be found in the summary files and in the pesviewer.inp files. For bimolecular products, the chemids are sorted from lowest to highest and concatenated with underscores. The output of this run is written in the pesviewer.inp file, which can be visualized with the PESViewer code (see below)

```
pes input.json no-kinbot allpaths chemid1 chemid2
```

Similarly, it is possible to select only the lowest energy pathway:

```
pes input.json no-kinbot lowestpath chemid1 chemid2
```

And also, to visualize all reactions from one species:

```
pes input.json no-kinbot well chemid1
```

Finally, a functionality allows to only select pathways which have a branching fraction of 1% or more for a particular well at a given temperature. This is started from the initial well, and iteratively expanded by considering the products of the reactions with a sufficiently large branching fraction from the previous iteration. It is important to note that the results depend on the well this search is started from.

```
pes input.json no-kinbot temperature temperature
```

4.4 Output

4.4.1 Summary file

A file called summary_{chemid}.out is written which is a summary of the reaction search (chemid is the id of the well the search started from). This file contains all the successful and failed reaction searches, together with the barrier heights and the product identifiers.

4.4.2 PESViewer input file

An input file is written for the PESViewer (https://github.com/rubenvdvijver/PESViewer). To visualize the reactions found, install the code and run:

```
pesviewr pesviewer.inp
```

This displays a graph showing the reactant and all products, including the barrier heights and well and product energies. The x-positions of all stationary points can be moved (drag and drop) to make the figure clearer. The figures of the molecules can also be moved (drag and drop).

5 KinBot file structure

KinBot contains 4 files in the root directory and 5 sub-directories

Root:

- kinbot/
  - __init__.py
    Empty file, necessary for python to recognize the directory as a part of the project
  - bfgs.py
    BFGS algorithm used to optimize a geometry with one or more updated coordinates
  - bond_combinations.py
    Files to enumerate all possible bond forming and bond breaking steps (not operational yet)
  - cheminfo.py
    Methods to link the KinBot species representation to the cheminformatics packages RDKit and OpenBabel.
  - conformers.py
    Methods to generate and run conformer searches for cyclic and open-chain molecules and transition states
  - constants.py
    List of constants necessary for KinBot
  - find_motif.py
    Algorithm to find a structural motif in a species
  - frequencies.py
    Algorithm to get the frequencies from a full hessian matrix by projecting out translation, external rotation and internal rotation.
  - geometry.py
    List of methods to calculate key values based on a geometry and to perform geometry transformations
  - hindered_rotors.py
    Methods to generate the geometries along a scan and run them.
  - homolytic_scissions.py
    Method to search for all homolytic bond scission products of a species
  - irc.py
    Methods to run the Intrinsic Reaction Coordinate calculations starting from a transition state and to identify the products and verify the connectivity to the original well
  - kb.py
    General script to run KinBot (only one well, see pes.py for a full PES search)
  - license_message.py
    License message that is printed when running KinBot and in the output and log files
  - mesmer.py
    Methods to build MESMER input files and automatically run them
  - mess.py
    Methods to build MESS input files and automatically run them
  - modify_geom.py
    Method to update a geometry based on a list of coordinates that need to change
  - molpro.py
    Method to build MolPro input files

    - optimize.py
      Collection of methods to, starting from a geometry (either a well or saddle point), goes through the list of conformer searches, high-level optimizations and hindered rotor calculations
    - parameters.py
      Parameters and options of the KinBot calculations
    - pes.py
      General script to run a full PES search
    - postprocess.py
      Scripts to postprocess KinBot and write the appropriate output files and figures
    - qc.py
      Methods to couple KinBot to quantum chemistry software packages through ASE
    - reac_family.py
      Generic method for a reaction family
    - reac_*.py
      There is a separate file per reaction family (replacing the asterisk with the family name) which contains the necessary data and recipe for the reaction family
    - reaction_finder.py
      Methods to find the candidate reactions of a species
    - reaction_generator.py
      Methods to go through each candidate reaction, generate a transition state structure, optimize that structure, do IRC calculations and optimize the products
    - stationary_pt.py
      Class for a stationary point on the PES, holding all the data of that stationary point, and providing algorithm to analyze the stationary point (find cycles, calculate aID's, etc.)
    - symmetry.py
      Algorithm to calculate the external and internal symmetry numbers and numbers of optical isomers
    - thread_kinbot.py
      Algorithm to thread many KinBot runs, used by the pes.py script.
    - zmatrix.py
      Algorithms to convert a geometry in cartesian coordinates to a zmatrix representation and vice versa
- tpl/
  This directory contains the templates for KinBot, these are simple text files that are read by KinBot and converted to useful files by formatting them appropriately.
- tests/
  This directory contains a collection of tests for KinBot
- examples/
    - gvl_reactions/
      Directory with input files for an example of the exploration of the reactions for gamma valerolactone

    - pentyl_pes/
      Directory with input files for an example of the full exploration of a PES for the n-pentyl radical
    - propanol_homolytic_scissions/
      Directory with input files for an example of the search for all homolytic scission reactions in propanol
    - propanol_radical_ME/
      Directory with input files for an example of the full exploration of a well, including conformation searches, high-level optimizations and hindered rotor calculations, of which the results are collected in an input file for MESS. The rate coefficients are subsequently calculated with MESS.
- docs/
  This directory contains the manual of KinBot
- __init__.py
  to Empty file, necessary for python to recognize the directory as a part of the project
- LICENSE
  License message
- README.md
  README file for KinBot
- setup.py
  Setup file used to build and install KinBot