Atomic and Molecular Cluster Energy Surface Sampler (AMCESS)

Edison Florez,[*,†,‡,#] Andy D. Zapata–Escobar,[*,¶,§,#] Daniel Bajac,[¶,§] Alejandra Mendez,[‖] Juan Jose Aucar,[¶,§] and César Ibargüen Becerra[⊥]

† *School of Natural and Computational Sciences (SNCS), Massey University Auckland, Private Bag 102904, 0632 Auckland, New Zealand*

‡ *Centre for Theoretical Chemistry and Physics (CTCP), The New Zealand Institute for Advanced Study, Massey University Auckland, Private Bag 102904, 0632 Auckland, New Zealand*

¶*Physics Department, Natural and Exact Science Faculty, Northeastern University of Argentina, Corrientes, Argentina*

§*Institute of Modelling and Innovative Technology (IMIT), Corrientes, Argentina*

‖*Instituto de Astronomía y Física del Espacio, UBA-CONICET, Buenos Aires, Argentina*

⊥*Department of Chemistry, University of Antioquia, Medellín Colombia*

#*Contributed equally to this work*

E-mail: edisonffh@gmail.com; danianescobarv@gmail.com

Abstract

Exploration of the Potential Energy Surface (PES) of molecules or atoms clusters is a crucial step to analyze physical–chemistry properties and processes. The Atomic and Molecular Energy Surface Sampler (AMCESS) is an end-to-end package implemented in Python 3.9 to generate candidate structures for the critical points sampling of the PES. The AMCESS package uses simple input files and automates common procedures to explore the PES using the Simulated Annealing, Simplicial Homology Global Optimization (SHGO), and Bayesian Optimization to generate candidate structures for any kind of critical point, such as local minima or transition states. The package also allows the user to perform local searches around defined regions. The PES is generated computing the electronic energy using standard and powerful quantum chemistry packages such as PySCF and Psi4, also implemented in Python. The AMCESS main purpose is to be a user friendly package, easy to install, import, and run, available in most platforms and open-source. As a Python module, AMCESS can be integrated into any workflow. This package has code reviews with unit testing and continuous integration, code coverage tools, and automatically keeps documentation up–to–date. The source code is available https://gitlab.com/ADanianZE/amcess

1 Introduction

In a technical sense, optimization means to find the best element from a given set with regard to some criteria. Optimization is the key step in almost any field. For instance, optimization is crucial in economics and finance to predict the stocks price, maximize the profit and minimize risk. In civil engineering and operation research, optimization is involved in problems such as managing the traffic, allocating resources, planning routes, etc. In molecular modelling, for example, finding the native structure (a minimum) will ensure tracing the most stable conformation of a protein. Furthermore, in any chemical reaction, locating its extreme values is crucial since reagents and products (minima) are linked through a transition state (maximum).[1]

The Potential Energy Surface (PES) is a pivotal concept in applications of electronic structure methods and in several other important fields of study, such as reactivity, molecular structure, chemical bond, physical interaction, protein folding, transition states, etc.[2–8] Exploring the PES to obtain critical points, maximum or minimum, representing atomic or molecular structures for a given cluster is very often a problematic subject. Finding critical points becomes even more problematic when dealing with systems with a large number of atoms or molecules. In a few cases, with cluster of reduce size, it is possible to build geometries based on the conjunction of prior knowledge of leading interaction, the PES, and informed intuition about the most feasible structures. This problem becomes impractical as the size of the system get larger, Figure 1.[9] Thus, having alternative methodologies, approaches and tools to explore systematically and efficiently a PES is always appreciated.

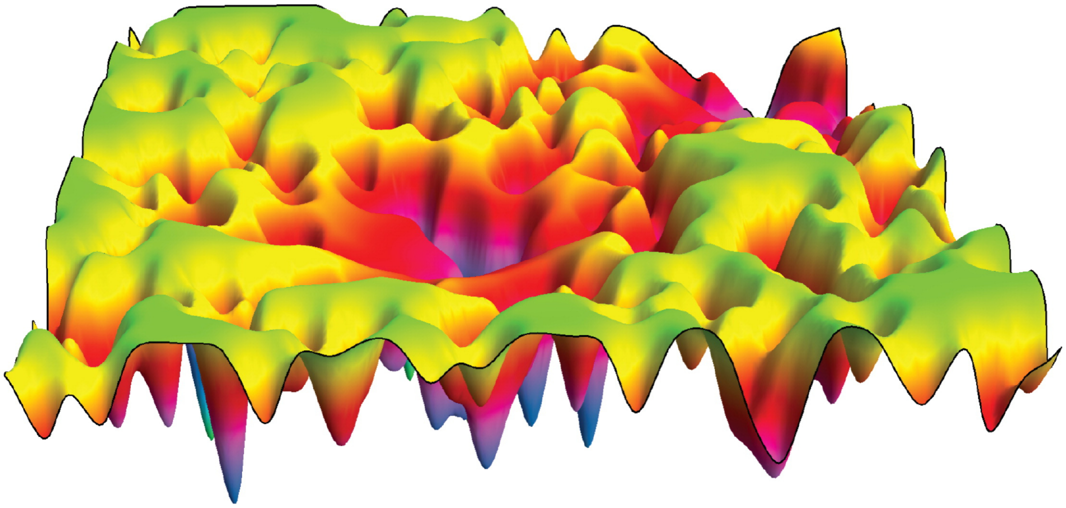


Figure 1: A schematic representation of a three–dimensional Potential Energy Surface (PES). Taken from reference.[9]

In this work, we present the Python package Atomic and Molecular Cluster Energy Surface Sampler (AMCESS). This is an object–oriented Python package to sample the PES for atomic and molecular clusters. AMCESS arise from the original idea of the ASCEC[10] (Spanish acronym for *Annealing Simulado Con Energía Cuántica*), which was written in FORTRAN77 and released about fifteen years ago. Two important remarks about ASCEC: *i)* It has been quite successfully used in a wide variety atomic and molecular systems with complex interaction: some examples include clusters linked via hydrogen bonding networks, microsolvation of cations and anions,[4,5,11] carbenes[12]i, metallic clusters[6,13,14] and clusters joined by means of van der Waals interactions;[15] *ii*) the ASCEC code is not easy to use for a beginner or non–experienced user since its learning curve is quite steep. The main purpose of the AMCESS project is to provide a simple, lightweight, and efficient platform to explore the PES maintaining ASCEC's reliability while offering a very user–friendly Python package for everyone who wants to produce high–quality PES sampling. The AMCESS package is currently under development and its first version is available in a public repository.[16]

AMCESS calculates the energy using PySCF,[17] the Python–based Simulations of Chemistry Framework. PySCF is an efficient platform for quantum chemistry calculations that can be used to simulate the properties of molecules and crystals using mean–field and post–mean–field methods, designed from the ground up pointing out to code simplicity. AMCESS have been written with a similar mindset, so that the energy is computed using any input function that takes in a array of coordinates (atom type and coordinates). The package only requires a minimal input information, such as the number of atoms and their coordinates.

Despite its click–and–go appeal, the package is designed to be used and customized for any specific case. As aforementioned, AMCESS is designed to be used in a Python environment. The package is capable to explore the potential energy surface using various techniques: Stochastic methods with Metropolis or Delta Energy acceptance criteria,[10,18] Inference methods (Bayesian Optimization)[19] and Combinatorial topology (Simplicial Homology Global Optimization - SHGO).[20]

2 The Sampling Techniques

Deterministic procedures based on derivatives (gradient and laplacian) are the most traditional methods used to find the maximum and minimum of a given function. These kind of methods, also called local–search methods, are initial conditions dependent and suitable for analytical known functions. However, the PES for atomic and molecular cluster is a function in N–dimension space, where N is proportional to the number of electrons and atoms. For such multidimensional problems, different approaches should be considered. For example, Stochastic methods are suitable to systematically sample the energy landscape. Within this group, the Monte Carlo technique[21] is a broad class of computational algorithms; it is a stochastic strategy that relies on repeated random sampling to obtain numerical results (probabilities). Moreover, in the last decade or so, machine–learning driven techniques have started to be implemented in atomic and molecular modelling with great success. A wide variety of approaches have been used in quantum–mechanical electronic structure computations, including Artificial Neutral Networks and Regression approaches.[22–24] In the following, we briefly describe some of these techniques, which have included in the AMCESS package.

2.1 Simulated Annealing

The Simulated Annealing (SA) algorithm was inspired by the annealing process that takes place in metallurgy, whereby annealing a molten metal causes it to achieve its global minimum in terms of thermodynamic energy. The following scheme shows a general view of a single Monte Carlo cycle.

1. Select a molecule/atom at random (configuration a)
2. Calculate the total energy ($\mathcal{E}_a$)
3. Give the molecule/atom a random move (configuration b)
4. Calculate the new total energy ($\mathcal{E}_b$)
5. Accept the move from a to b with probability (Metropolis criterion)

$$\mathcal{P}(a \to b) = e^{-\beta \Delta \mathcal{E}} \tag{1}$$

where $\Delta\mathcal{E} = \mathcal{E}_b - \mathcal{E}_a$, $\beta = 1/k_B T$, and k_B is the Boltzmann constant. If the new configuration is rejected, the old configuration (a) is kept. The next temperature T is selected from a geometric grid only when the energy for the new configuration is lower than the old one ($\mathcal{E}_b < \mathcal{E}_a$).

Regarding to Metropolis criterion, the acceptance probability is compare with a uniform distribution of random numbers, the ASCEC criterion uses a relative change in energy ($(\mathcal{E}_b - \mathcal{E}_a)/\mathcal{E}_b$) to avoid rejecting "good" structures, randomly. Restrepo and coworkers[6,25] have

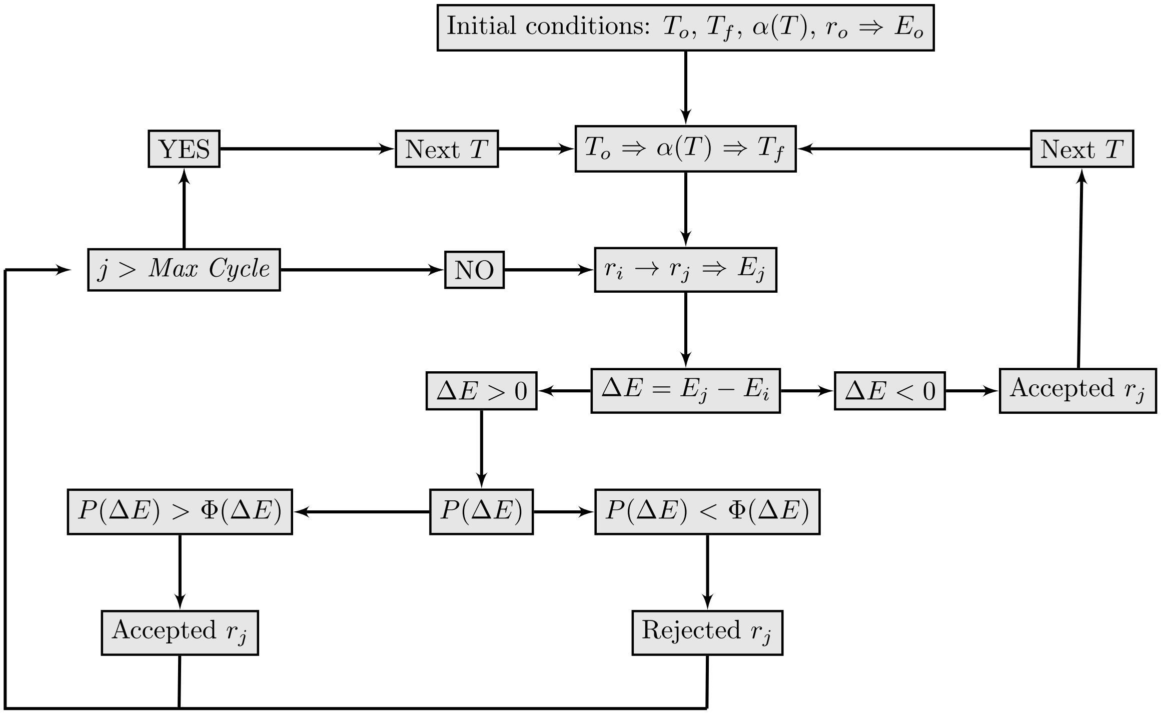

Initial conditions: T_o, T_f, $\alpha(T)$, $r_o \Rightarrow E_o$
YES
Next T
$T_o \Rightarrow \alpha(T) \Rightarrow T_f$
Next T
$j > Max\ Cycle$
NO
$r_i \rightarrow r_j \Rightarrow E_j$
$\Delta E > 0$
$\Delta E = E_j - E_i$
$\Delta E < 0$
Accepted r_j
$P(\Delta E) > \Phi(\Delta E)$
$P(\Delta E)$
$P(\Delta E) < \Phi(\Delta E)$
Accepted r_j
Rejected r_j

Figure 2: The ASCEC flow diagram. Where T_o is the initial temperature, T_f is the final temperature, $\alpha(T)$ is the quenching route and the function $\Phi(\Delta E) = |(\mathcal{E}_b - \mathcal{E}_a)/\mathcal{E}_b|$ is the new criterion for the acceptance of new configuration.[6,25]

found the latter acceptance criterion to be more adequate for PES in atomic and molecular cluster than the usual procedure of comparing again to a randomly generated number.

The Figure 2 shows a flow diagram for the Simulated Annealing procedure with a modified Metropolis acceptance criterion. On the Markov chain, new configurations with a lower energy are always accepted ($\Delta\mathcal{E} = \mathcal{E}_b - \mathcal{E}_a < 0$). However, when $\Delta\mathcal{E} > 0$ this new configuration is accepted when the probability of this change $P(\Delta\mathcal{E}) = exp(-\Delta\mathcal{E}/k_BT)$ is larger than the relative change in energy $\Phi(\Delta E) = |(\mathcal{E}_b - \mathcal{E}_a)/\mathcal{E}_b|$.

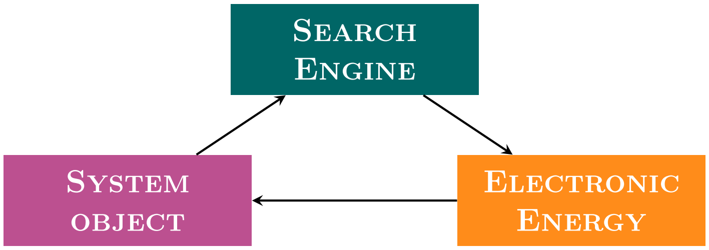
In addition, the Dual Annealing method, implemented in the Scipy Library,[26] has also been included into the AMCESS package. This method follows the Generalized Simulated Annealing, i.e., a generalization of the SA.[18] This technique uses the Tsallis–Stariolo form of the Cauchy–Lorentz visiting distribution, which avoids the possibility of getting trapped in local minima so the global minimum can be found. Moreover, a generalized Metropolis algorithm is used for the acceptance probability, which also provides the possibility of finding multiple local minima, if there are any.

2.2 Simplicial Homology Global Optimization (SHGO)

We have also included another extra sampling method, the Simplicial Homology Global Optimization (SHGO), which is a global optimization algorithm based on application of simplicial integral homology and combinatorial topology,[20] as implemented into the Scipy Library.[26] This approach is used to confine regions with local minima. Then, a local minimization routine is used to characterize the total energy of the cluster. Finally, the software orders the minima found in the local minimisation steps to find the approximate global minimum. The variables are the distance between the center of mass of the molecules that compound the cluster, and the Euler's angles between them. This characteristic coordinates also allowed us to introduce non–linear constraints in the problem statement, which can be use to avoid overlapping.

2.3 Bayesian inference with Gaussian processes

Following the latest trends in optimization techniques, we included a widely implemented approach in machine–learing field: the Bayesian inference method with Gaussian processes (GP). This method is composed of two quite abstract concepts. On one hand, the Bayesian inference relies in the Bayes' theorem.[27] This theorem allows one to determine the occurrence probability of an event, based on prior knowledge of conditions that might be related to that event. On the other hand, a Gaussian process is a family of randomly distributed functions that feature a multivariate Gaussian distribution. A detailed and comprehensive overview of this approach can be found in Ref. 19. This approach allows one to find the global and local minima of high–dimensional surfaces with relatively low computational cost. Two important features must be define beforehand: the kernel and the acquisition function. In a nutshell, the kernel function characterizes the rugosity of the PES while the acquisition



Cluster	Sampler	Energy
Molecule/Atom	Simulated Annealing	PySCF
Coordinates	GP Bayesian Inference	Custome function
Rotation/Translation	Dual Annealing	
Sphere	SHGO	

Figure 3: The APIs composing the AMCESS code. Only some of theirs attributes and methods are show. See Figure 4 for more details.

function will define the algorithm used for exploring/exploiting the surface. In this work, we implemented the Gaussian processes Bayesian inference from the GPyOpt code, from the Sheffield machine learning group and collaborators.[28]

3 Technical details

The AMCESS package consists of three application programming interfaces (APIs): System Object, Search Engine and Electronic Energy, which are illustrated in Figure 3 and 4. A more detailed description of these APIs, their properties, methods and communication paths are given in the following sections.

3.1 System Objects

The System Object API features three classes: `Atom`, `Molecule` and `Cluster`. The `Atom` class represents individual atoms and incorporates atomic properties, such as atomic mass and symbol. The `Molecule` class represents a molecule composed of at least one atom. This

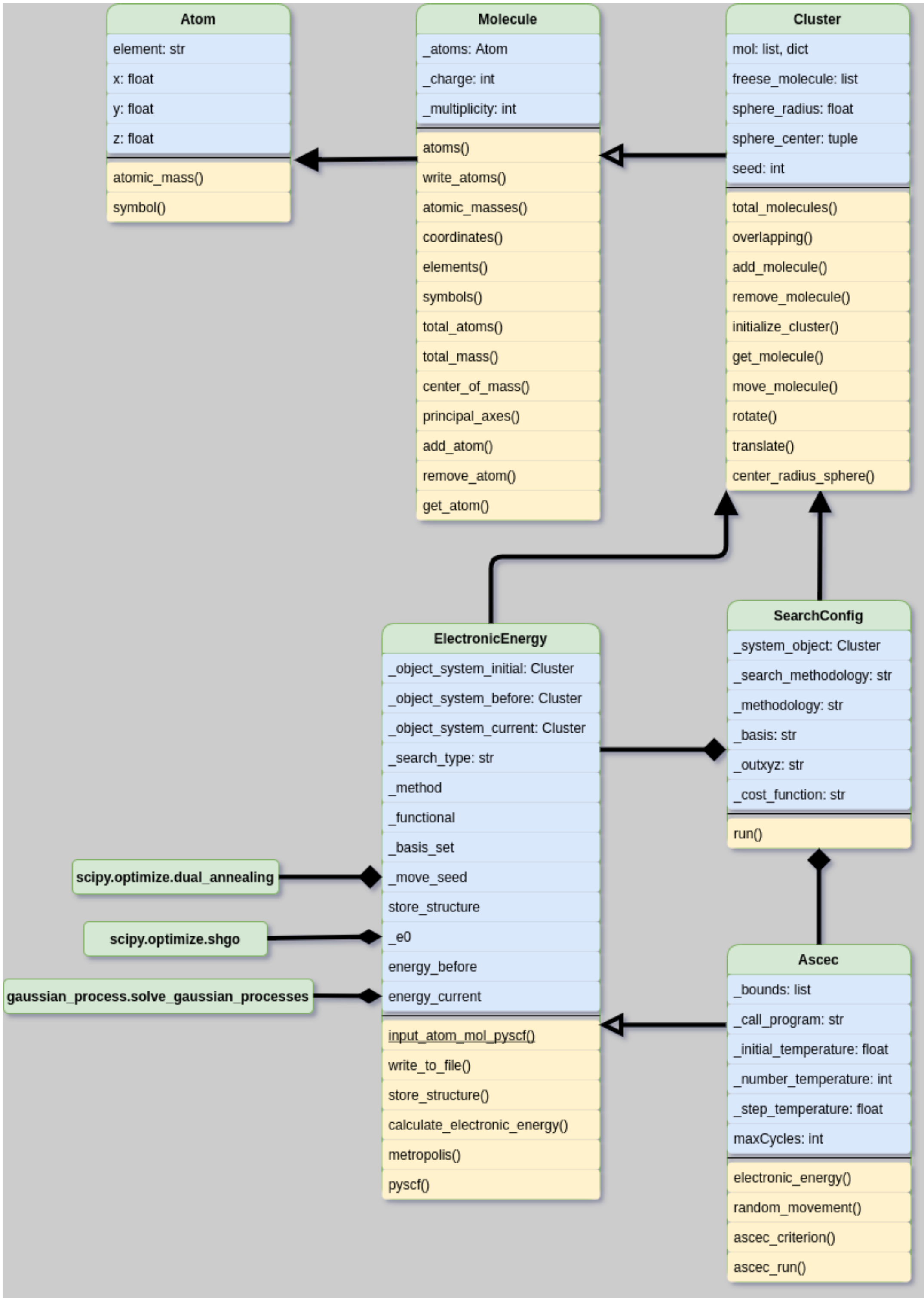


Figure 4: Unified Modeling Languages (UML) Diagram of AMCESS. Arrow with triangular head indicate association, arrow with unfilled triangular head indicate heredity, and the another type of arrow is to indicate composition.

object features a significant amount of properties (atomic charges, mass and coordinates, for example) and methods (include and remove atoms, for instance). For simplicity, we refer from now on to the `Atom` and `Molecule` objects as fragments. The `Cluster` object is defined as a collection of atoms and/or molecules, and it must be composed of at least two object. A particular feature of the clusters defined is that within each fragment the atomic cartesian coordinates are internally fixed. Hence, the molecules behave as rigid objects and they can be translated, rotated or randomly moved. An example of how these object are defined in AMCESS is shown in Code 1.

```
>>> from amcess.base_molecule import Atom, Molecule, Cluster
>>> water_dict = {
...     "atoms": [("O", 0, 0, 0), ("H", 0.587, 0.757, 0), ("H", -0.587, 0.757, 0)],
...     "charge": 0,
...     "multiplicity": 1,
... }
>>> water_mol = Molecule.from_dict(water_dict)
>>> water_pentamer = Cluster(5 * water_mol)
>>> water_pentamer.sphere_radius = 2.0
>>> water_pentamer.sphere_center = (3.0, 3.0, 3.0)
```

Code 1: Example of Atom, Molecule and Cluster class using the AMCESS code. Creation of a water pentamer cluster into a sphere of radius 2.0 Angstrom and centered at (3, 3, 3).

3.2 Search engine

The Search Engine API is the core of the AMCESS package. This application articulates the optimization algorithm used to sample the Potential Energy Surface of a `Cluster` object. Simultaneously, the Search application communicates with the Electronic Energy API, which determines the electronic energy of a given configuration (see details in Section 3.3).

The exploration of the PES is conducted by the `SearchConfig` class. This object requires a `Cluster` object as input and it features five customizable arguments, which includes the search algorithm (described in Section 2), the electronic structure method, the basis set used for the computation, the file where optimization calculation will be written and the cost function to be minimized. The main method of the search object is `run`, which allows one to perform the calculation. This method requires different minimal input, depending on the search algorithm defined as shown in Code 2.

```
>>> from amcess.search_engine import SearchConfig
    # starting from a water pentamer CLUSTER object
    # Simulated Annealing search
>>> obj_sa = SearchConfig(water_pentamer)
>>> obj_sa.run(initial_temperature=1000, step=0.2, number_temperatures=10, max_cycles=20)
    # dual annealing search
>>> obj_da = SearchConfig(water_pentamer, search_methodology="dual_annealing")
>>> obj_da.run(maxfun=10,maxiter=10)
    # shgo searching
>>> obj_shgo = SearchConfig(water_pentamer, search_methodology="SHGO")
>>> obj_shgo.run(sampling_method="sobol", n=2)
    # Bayesian optimization search
>>> obj_bayesian = SearchConfig(water_pentamer, search_methodology="Bayesian")
>>> obj_bayesian.run(initer=10, maxiter=10)
```

Code 2: Minimal code example for the sampling of the PES for the water pentamer $(H_2O)_5$. See Code 1 for the making of the water pentamer CLUSTER object. Here is shown how to run all the four methodologies valailables, Simulated Annealing, Dual Annealing, SHGO, and Bayesian Optimization. If the level of theory is not defined, Hartree–Fock hamiltonian and STO basis set are used to compute the energy.

3.3 Electronic energy

The Electronic Energy API articules the electronic structure calculation according to the method defined in the `SearchConfig` class input. Here, we only show examples with the PySCF[17] package to compute the electronic energy, with various methods available: Hartree–Fock, Density Function Theory, Møller–Plesset perturbation theory and Coupled Cluster Single–Double method. The API is composed of a single class that automatically generates the corresponding intpus for the PySCF package and prints out to the corresponding file the calculations performed.

3.4 Software quality assurance

The AMCESS package has been written in Python3.9 following two of the most common procedures of software quality assurance: unit–testing and code–coverage. On one had, unit–testing allows each of the components of the code to be tested individually in order to verify their correct operation. On the other hand, code–coverage provides a measurement of the unit–testing performed over the source code. The AMCESS package testing is distributed over four suites with a 95% of coverage.

The first version of the AMCESS package is available in a public repository 16 with a GNU General Public License. The package was written following the PEP 8 - Style Guide for Python code, and this styling was tested automatically by using flake8 software. The source code is still under development, and all new versions are automatically tested with the tox package. Moreover, the package documentation[1] is public and generated automatically using the read–the–docs service.

4 Conclusions

In this manuscript, we present the Atomic and Molecular Energy Surface Sampler (AMCESS) package, a tool written in Python3.9 to explore the Potential Energy Surface (PES) for atomic and molecular systems and generate candidate structures for local and global energy minima. The exploration of the PES can be performed using a wide variety of algorithms included in the package: simulated annealing, dual annealing, a global optimization method (SHGO) and a Bayesian inference approach.

The AMCESS is a user friendly package, easy to install, import, and run, available in most platforms and open-source. As a Python module, AMCESS can be integrated into any workflow. This package has code reviews with unit testing and continuous integration, code coverage tools, and automatically keeps documentation up–to–date.

[1] https://ADanianZE.gitlab.io/amcess

References

(1) Truhlar, D. G.; Garrett, B. C.; Klippenstein, S. J. Current status of transition-state theory. *The Journal of physical chemistry* **1996**, *100*, 12771–12800.

(2) Vargas-Caamal, A.; Cabellos, J. L.; Ortiz-Chi, F.; Rzepa, H. S.; Restrepo, A.; Merino, G. How many water molecules does it take to dissociate HCl? *Chemistry–A European Journal* **2016**, *22*, 2812–2818.

(3) Flórez, E.; Maldonado, A. F.; Aucar, G. A.; David, J.; Restrepo, A. Microsolvation of methylmercury: structures, energies, bonding and NMR constants (199 Hg, 13 C and 17 O). *Physical Chemistry Chemical Physics* **2016**, *18*, 1537–1550.

(4) Zapata-Escobar, A.; Manrique-Moreno, M.; Guerra, D.; Hadad, C.; Restrepo, A. A combined experimental and computational study of the molecular interactions between anionic ibuprofen and water. *The Journal of chemical physics* **2014**, *140*, 05B611_1.

(5) Ibargüen, C.; Manrique-Moreno, M.; Hadad, C.; David, J.; Restrepo, A. Microsolvation of dimethylphosphate: a molecular model for the interaction of cell membranes with water. *Physical Chemistry Chemical Physics* **2013**, *15*, 3203–3211.

(6) Romero, J.; Reyes, A.; David, J.; Restrepo, A. Understanding microsolvation of Li+: structural and energetical analyses. *Physical Chemistry Chemical Physics* **2011**, *13*, 15264–15271.

(7) Murillo, J.; David, J.; Restrepo, A. Insights into the structure and stability of the carbonic acid dimer. *Physical Chemistry Chemical Physics* **2010**, *12*, 10963–10970.

(8) Hincapie, G.; Acelas, N.; Castano, M.; David, J.; Restrepo, A. Structural studies of the water hexamer. *The Journal of Physical Chemistry A* **2010**, *114*, 7809–7814.

(9) Wales, D. J. Exploring energy landscapes. *Annual review of physical chemistry* **2018**, *69*, 401–425.

(10) Pérez, J.; Restrepo, A. ASCEC V–02: Annealing Simulado con Energía Cuántica. *Property, development and implementation: Grupo de Química–Física Teórica, Instituto de Química, Universidad de Antioquia: Medellín, Colombia* **2008**,

(11) Rojas-Valencia, N.; Ibargüen, C.; Restrepo, A. Molecular interactions in the microsolvation of dimethylphosphate. *Chemical Physics Letters* **2015**, *635*, 301–305.

(12) Gomez, S.; Guerra, D.; Lopez, J. G.; Toro-Labbe, A.; Restrepo, A. A detailed look at the reaction mechanisms of substituted carbenes with water. *The Journal of Physical Chemistry A* **2013**, *117*, 1991–1999.

(13) Pérez, J. F.; Florez, E.; Hadad, C. Z.; Fuentealba, P.; Restrepo, A. Stochastic search of the quantum conformational space of small lithium and bimetallic lithium- sodium clusters. *The Journal of Physical Chemistry A* **2008**, *112*, 5749–5755.

(14) Moreno, N.; Ferraro, F.; Flórez, E.; Hadad, C.; Restrepo, A. Spin–Orbit Coupling Effects in Au m Pt n Clusters (m+ n= 4). *The Journal of Physical Chemistry A* **2016**, *120*, 1698–1705.

(15) Ibargüen, C.; Guerra, D.; Hadad, C.; Restrepo, A. Very weak interactions: structures, energies and bonding in the tetramers and pentamers of hydrogen sulfide. *RSC Advances* **2014**, *4*, 58217–58225.

(16) The AMCESS package authors, Atomic and Molecular Cluster Energy Surface Sampler. https://gitlab.com/ADanianZE/ascec, 2021.

(17) Sun, Q.; Zhang, X.; Banerjee, S.; Bao, P.; Barbry, M.; Blunt, N. S.; Bogdanov, N. A.; Booth, G. H.; Chen, J.; Cui, Z.-H., et al. Recent developments in the PySCF program package. *The Journal of chemical physics* **2020**, *153*, 024109.

(18) Tsallis C, S. D. Generalized Simulated Annealing. *Physica A, , (1996).* **1996**, *233*, 395–406.

(19) Garnett, R. *Bayesian Optimization*; Cambridge University Press, 2022; in preparation.

(20) Endres, S. C.; Sandrock, C.; Focke, W. W. A simplicial homology algorithm for Lipschitz optimisation. *Journal of Global Optimization* **2018**, *72*, 181–217.

(21) Kroese, D. P.; Taimre, T.; Botev, Z. I. *Handbook of monte carlo methods*; John Wiley & Sons, 2013; Vol. 706.

(22) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. Gaussian Process Regression for Materials and Molecules. *Chemical Reviews* **2021**, *121*, 10073–10141.

(23) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *Journal of Chemical Theory and Computation* **2019**, *15*, 3678–3693.

(24) Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930–17955.

(25) Pérez, J. F.; Hadad, C.; Restrepo, A. Structural studies of the water tetramer. *International Journal of Quantum Chemistry* **2008**, *108*, 1653–1659.

(26) Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272.

(27) Wikipedia, Bayes' theorem. https://en.wikipedia.org/wiki/Bayes%27_theorem, 2021.

(28) The GPyOpt authors, GPyOpt: A Bayesian Optimization framework in python. http://github.com/SheffieldML/GPyOpt, 2016.