

Autobench: Automatização de benchmark para cálculos de estrutura eletrônica em química orgânica

Palavras-Chaves: Benchmark, Automatização, Química Computacional

Autores/as:

Gabriel Douglas da Silva IQ-DQO-Unicamp

Prof. Dr. Rodrigo Antonio Cormanich (orientador) IQ-DQO-Unicamp

INTRODUÇÃO:

O benchmark é uma etapa fundamental para realização de cálculos teóricos, é utilizado para comparar os resultados dos métodos *Ab initio* e/ou *Density Functional Theory* (DFT), além dos conjuntos de bases, com um valor de referência. Os *Ab initio* resolvem a equação de Schroedinger utilizando aproximações, como Born-Oppenheimer,^[1] combinação de orbitais atômicos (*Linear Combination of Atomic Orbitals* – LCAO) e eletrônicas, sendo o primeiro método desenvolvido o Hartree-Fock (HF). Contudo o HF apresentava muitos erros devido a parte eletrônica, logo foram desenvolvidos outros métodos, conhecidos como pós-HF, que vieram com o intuito de corrigir estes erros, alguns destes métodos são utilizados como referência devido a sua precisão, como é o caso do CCSD(T).^[2]

O DTF, diferente do *Ab initio*, utiliza a densidade eletrônica para obter informação da molécula, para isso é empregado um funcional como intermediário, sendo ele o responsável por adquirir a informação.^[3] A desvantagem do DFT é que embora a existência do funcional que relaciona a densidade de elétrons à energia de um sistema possa ser demonstrada matematicamente, a forma exata dele ainda é desconhecida, sendo necessário uma aproximação destes funcionais conhecidos como *exchange-correlation*, implicando em uma vasta quantidade de funcionais.^[4] Mas a principal vantagem do DTF é o tempo de cálculo, sendo mais rápidos que os métodos *Ab initio* e igualmente precisos, se utilizar o funcional correto para a molécula.

Os conjuntos de bases, conhecidos como *basis set*, são uma combinação linear de funções de base utilizadas para construir orbitais moleculares, que por sua vez representam aproximadamente a função de onda eletrônica. Assim, o uso de conjuntos de bases nos cálculos da estrutura eletrônica é obrigatório. A qualidade do cálculo e a precisão das informações derivadas dependem fortemente da combinação do método com o conjunto de bases escolhido, se não escolhida de forma correta, é a principal fonte de erro

do cálculo.^[4,5] A fim de obter menor erro no cálculo da referência, é realizado uma extrapolação do conjunto de bases, conhecido como *complete basis set* (CBS).

Devido a grande quantidade de funcionais presente no DFT e os métodos *Ab initio*, a realização de um benchmark torna-se necessário com o intuito de obter a melhor combinação método/funcional e conjunto de base, em relação a uma referência. Idealmente o valor de referência seria o experimental, mas devida a complexidade de obtenção e até impossibilidade de obter os valores, é utilizado um método e conjunto de base que não utilizam aproximações experimentais, neste caso o CCSD(T)/CBS. Ele é utilizado como padrão de ouro, pois se aproxima da resolução da equação de Schroedinger e o erro presente é conhecido. ^[5-7]

O tempo de cálculo é um fator importante no dia a dia, o benchmark tem como intuito fornece o método/funcional e conjunto de base com menor diferença de energia da referência e que obteve um menor tempo de cálculo. Como mencionado acima, os cálculos de DFT são mais rápidos que os de *Ab initio*, sendo que a velocidade varia de acordo com o tamanho do conjunto de base. Para o DFT, a escala de tempo é da terceira ordem em relação ao conjunto de base, enquanto para o CCSD(T) sua escala é de sétima ordem, portanto, torna-se inviável o uso do CCSD(T) no cotidiano. ^[8]

Ao realizar um benchmark, é necessário realizar a busca conformacional do sistema, seguida de uma otimização para minimização da energia e com as coordenadas encontradas na otimização, a preparação de diversos arquivos de cálculo de energia com os métodos/funcionais e conjuntos de bases, e por fim o tratamento estatístico dos dados.

Devido a grande quantidade de etapas e dados a serem tratados, a automatização das etapas por meio de scripts torna-se uma maneira amigável de se lidar com as informações. Alguns grupos de pesquisas atualmente já disponibilizam softwares e scripts a fim de agilizar o cotidiano de outros pesquisadores. Pode-se citar os softwares ENSO^[9] e CENSO^[10], recentemente desenvolvidos pelo grupo de pesquisa do professor Grimme, que exigem do usuário apenas que este saiba como desenhar a estrutura da molécula de interesse para que seja possível o usuário obter de maneira semiautomática propriedades estruturais e espectroscópicas.

Portanto, este projeto tem como finalidade automatizar benchmark, fazendo com que os grupos de pesquisas não necessitem desempenhar tempo organizando e tratando os cálculos teóricos.

METODOLOGIA:

O programa Autobench é executado no sistema operacional Linux, sendo desenvolvido nas linguagens de programação Shell, nativa do Linux, e Python ^[11]. O Shell é utilizado como a principal ferramenta de manipulação, criação e gerenciamento dos arquivos gerados, sendo responsável pelo

funcionamento do programa. Já o Python tem como responsabilidade o tratamento dos resultados e a criação da interface de configuração. A principal vantagem em utilizar o Python para o tratamento de dados é a suas bibliotecas, neste projeto é utilizado as bibliotecas nativas: os, re, tkinter e sys e instaladas: NumPy ^[12], pandas ^[13] e Matplotlib ^[14].

Para a realização dos cálculos teóricos é utilizado os softwares: ORCA ^[15] e Gaussian ^[16], ficando à disposição do usuário da utilização de apenas um deles. Além destes, é utilizado o CREST ^[17] para realização da busca conformacional da molécula de interesse e xTB ^[18] para realização de uma pré-otimização.

RESULTADOS E DISCUSSÃO:

Estrutura do programa

O autobench é constituído de 4 partes, sendo elas: busca conformacional, pré-otimização, otimização e o benchmark (Figura 1). Para que inicie o cálculo o usuário deve entregar ao programa dois arquivos, sendo um xyz contendo a molécula ou os confôrmeros e outro chamado de settings, que contém as informações de como será realizado o cálculo.

O arquivo settings podem ser preparado chamando o script “autobench_prep” com o arquivo xyz, por exemplo: autobench_prep arquivo.xyz. Com isso será gerado o arquivo.settings, que contém as configurações de cálculo em cada etapa.

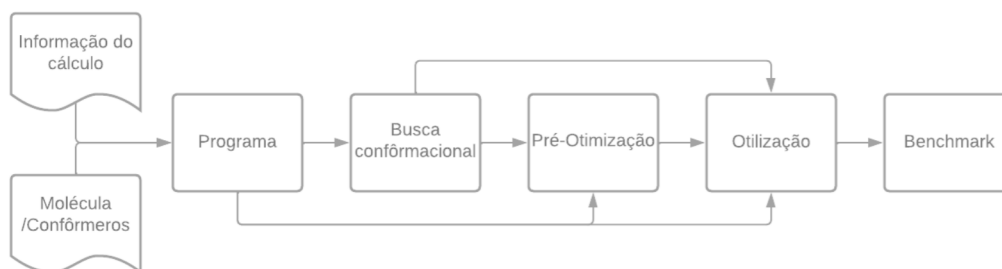


Figura 1. Fluxo grama de etapas para utilização do autobench

As partes 1 e 2, referente a busca conformacional e a pré-otimização, respectivamente, podem ser puladas. A parte 1 é realizada pelo programa CREST, sendo ele o responsável por realizar a busca conformacional da molécula do estudo. O usuário tem a liberdade de mudar as configurações da busca conformacional de acordo com as keywords do próprio CREST, desde o limite de energia entre os confôrmeros, até o número de processadores para o cálculo. Já parte 2, tem como objetivo minimizar o tempo de otimização realizando uma pré-otimização em xTB, que é um método rápido e preciso.

No caso das partes 3 e 4, a otimização e o benchmark, respectivamente, não podem ser pulados. A otimização e o benchmark podem ser realizados nos softwares ORCA e/ou Gaussian. A parte 3 tem como objetivo obter uma geometria de menor energia, além de que, nesta parte é realizado cálculo de frequência para adquirir as correções de energia para gibbs e entalpia, que serão utilizadas em seguida.

Na parte 4 é realizado os cálculos de energia dos métodos com os conjuntos de bases definidas pelo usuário, nesta etapa, o cálculo pode ocorrer de 3 formas diferentes: *func*, *base* e *both*. Se utilizar a opção *func*, em um conjunto de base fixo é calculado todos os métodos definidos pelo usuário, com isso é obtido uma energia diferente para cada *keyword* (método e base juntos), aquela *keyword* que tiver menor energia, será utilizado o método dela nos conjuntos de base informado pelo usuário, criando outras *keyword*, que resultará no melhor resultado (Figura 2-a). A opção *base* é similar a *func*, apenas alterando a ordem, sendo que inicialmente os métodos são variados (Figura 2-b). Já a *both*, é criado todas as combinações possíveis entre os conjuntos de bases e os métodos (Figura 2-c).

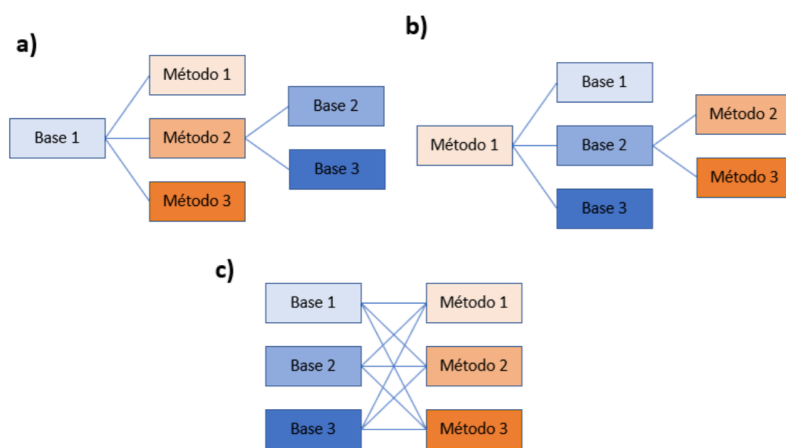


Figura 2. Fluxograma das maneiras de realizar o benchmark. a) *func*: começar com o conjunto de base fixo, variando os métodos; b) *base*: começar com um método fixo e variar os conjuntos de base; c) *both*: realizar todas as combinações possíveis entre os conjuntos de bases e os métodos.

O resultado do benchmark é apresentado em duas formas diferentes, um arquivo csv, contendo a variação da energia em relação a referência e uma tabela de erro absoluto médio.

CONCLUSÕES:

Devido a complexidade de se realizar um benchmark para iniciantes, além da vasta quantidade de dados a serem tratados e analisados pelo pesquisador, a automatização das etapas do processo de benchmark torna-se atrativa para o público leigo. Além de que, o Autobench é uma ferramenta que tem como objetivo principal facilitar o cotidiano de grupos de pesquisa que necessitam realizar este

procedimento inicialmente no projeto, retornando o melhor método e conjunto de base, dentre diversas, que melhor se enquadra para a molécula do estudo.

BIBLIOGRAFIA

- [1] - Agostini F, Curchod BFE.2022 Chemistry without the Born–Oppenheimer approximation. *Phil.Trans. R. Soc. A*380: 20200375.
- [2] – Řezáč, Jan; *et al.* Benchmark Calculations of Noncovalent Interactions of Halogenated Molecules. *J. Chem. Theory Comput.* 2013, 9, 5, 2151–2155
- [3] - LEWARS, Errol. Density Functional Calculations. *In: LEWARS, Errol. **Computational Chemistry**: Introduction to the Theory and Applications of Molecular and Quantum Mechanics.* Kluwer Academic Publishers, 2003. cap. 7, p. 385-436.
- [4] - Rimola, A.; Ugliengo, P. Computational Chemistry Methods. Encyclopedia. Available online: <https://encyclopedia.pub/entry/6769> (accessed on 27 July 2022).
- [5] - Řezáč, Jan; *et al.* Benchmark Calculations of Interaction Energies in Noncovalent Complexes and Their Applications. *Chem. Rev.* 2016, 116, 9, 5038–5071 Publication Date: March 4, 2016. <https://doi.org/10.1021/acs.chemrev.5b00526>
- [6] - Řezáč, Jan; *et al.* Describing Noncovalent Interactions beyond the Common Approximations: How Accurate Is the “Gold Standard,” CCSD(T) at the Complete Basis Set Limit?. *J. Chem. Theory Comput.* 2013, 9, 5, 2151–2155. Publication Date: April 4, 2013. <https://doi.org/10.1021/ct400057w>
- [7] - Sengupta, A., Ramabhadran, R. O., Raghavachari, K.. *J. Comput. Chem.* 2016, 37, 286–295. DOI: 10.1002/jcc.24050
- [8] – Sousa, Filipe S, *et al.* General Performance of Density Functionals. *J. Phys. Chem. A* 2007, 111, 42, 10439–10452. Publication Date: August 25, 2007. <https://doi.org/10.1021/jp0734474>
- [9] - No Title https://xtb-docs.readthedocs.io/en/latest/enso_doc/enso.html.
- [10] - Grimme, S.; Bohle, F.; Hansen, A.; Pracht, P.; Spicher, S.; Stahn, M. Efficient Quantum Chemical Calculation of Structure Ensembles and Free Energies for Nonrigid Molecules. *J. Phys. Chem. A* 2021, 125 (19), 4039–4054. <https://doi.org/10.1021/acs.jpca.1c00971>.
- [11] – Python Software Foundation. Disponível em: <https://www.python.org>. Acessado em: 02/08/2022
- [12] – NumPy. Disponível em: <https://numpy.org>. Acessado em: 02/08/2022
- [13] – The pandas development team. Disponível em: <https://pandas.pydata.org>. Acessado em: 02/08/2022
- [14] - MatHunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering.* 2007, 9 (3), 90-95. 10.1109/MCSE.2007.55
- [15] - Neese, F. (2012) The ORCA program system, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2, 73-78.
- [16] - Gaussian 16, Revision C.01, M. J. Frisch, G. W. Trucks, H. B. *et al.*, Gaussian, Inc., Wallingford CT, 2016.
- [17] - Pracht, Philipp and Bohle, Fabian and Grimme, Stefan. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Phys. Chem. Chem. Phys.*, 2020,22, 7169-7192
- [18] - C. Bannwarth, E. Caldeweyher, S. Ehlert, A. Hansen, P. Pracht, J. Seibert, S. Spicher, S. Grimme *WIREs Comput. Mol. Sci.*, 2020, 11, e01493. DOI: 10.1002/wcms.1493