

**Área: Química Teórica -
TEO**

Nº de Inscrição: _____
(Inserir o número de inscrição do autor
que fez a submissão)

QSARModelingPy: a Python package to build and validate QSAR models.

Héilton M. Reis Filho (IC),^{1*} João Paulo A. Martins (PQ).¹

joaopauloam@ufmg.br; helitonmrf@ufmg.br

¹Departamento de Química, UFMG

Palavras Chave: QSAR, variable selection, genetic algorithm, OPS, model validation.

Highlights

A complete Python package was developed to build and validate QSAR models.

Jupyter notebook, command line and graphical user interfaces allow its use in different levels of expertise.

Resumo/Abstract

QSARModelingPy evolves from QSARModeling (Martins and Ferreira, 2013), a software focused on the build and validation of QSAR and QSPR models. Developed in Python 3, this new version comes to solve some problems present in the previous version, written in Java 6. For example, the old version was black box, what indicates a lower suitability and scalability of the software, as well limits its execution to a local machine. To solve that, the Python version has multiple interfaces and is open-source. Because it has a Graphical User Interface, a Jupyter Notebook implementation, a headless command line version and a library hosted on the Python Package Index (PyPI), its use is facilitated from the end user with no programming knowledge to a fully programmatic or remote environment. In addition, some features have also been included such as the external validation and variable selection using Genetic Algorithm, which generally shows better results than the selection done with the Ordered Predictors Selection (OPS) algorithm.

2D and 4D-QSAR models were built with this program using the Genetic Algorithm for two data sets of synthetic tetronamides displaying inhibitory activity against the photosynthetic electron transport chain in isolated spinach chloroplasts and the photoautotrophic growth of the model cyanobacterium *S. elongatus* (Acosta *et al*, 2019). The obtained Q^2 values for the first assay were 0.79 (2D) and 0.75 (4D). For the latter, these values were 0.72 (2D) and 0.89 (4D).

Agradecimentos/Acknowledgments

The authors are thankful to UFMG for the structure.