Subject: Re: drug score/evolutionary algorithm
Posted by thomas on Mon, 08 Apr 2019 14:13:04 GMT
View Forum Message <> Reply to Message

Hello Patrick,

DataWarrior does not use the CDK (chemistry development kit). It is based on our own cheminformatics functionality, which is part of the DataWarrior source code. Most of it is also available as open-source on GitHub as "OpenChemLib".

Otherwise your assumptions are mostly correct. I summarize here with my words:

The evolutionary algorithm does not calculate the 'Drug Score' of the 'Property Explorer' unless you define something similar as fitness criteria, as you already have concluded. It uses random mutations to generate structurally slightly changed molecules from a set of previously generated molecules (the previous generation). For applying a random change on a previous generation molecule, DataWarrior generates a list of possible mutations, e.g. adding a atom (C,N,O or other) to any existing atom, changing the bond order of an existing bond, doing a ring expansion, migrating a substituent, ...
For everyone of these mutations DataWarrior is assigning a likelihood by looking at the atom types that disappear or a created during the change. An atom type is basically the atomic number of an atom and its neighbours plus bond orders between them, plus whether atoms are in a ring, aromatic, allylic, stabilized by carbonyl-neighbours, etc. Now comes into the game, whether you have selected 'drug like' or natural product like', because depending on that DataWarrior uses a different atom type frequency table to calculate mutation likelihoods. The effect is finally, that DataWarrior when selecting a mutation will prefer those that produce drug-like (or NP-like) atoms rather than destroys them. These are atoms with a local environment that is typical for drug like molecules.

Your idea to create more molecules in the last generation than in all the others makes perfectly sense, provided that you are not looking for one, but for many molecules that fulfill your fitness criteria. Alternatively, you may leave the algorithm running after fitness maximization has been achieved. Then, the algorithm will meander around and produce more molecules with a high fitness, which are connected through a similarity chain. A more diverse set of fit molecules will be created, if you let the algorithm run many times with very small generation sizes. In fact, I am currently working on a new functionality to generate random drug-like or NP-like molecules using a new evolution path for every individual molecule starting from a very tiny seed-molecule. This work surprisingly well. For that I fixed issues with the atom typing and the Mutator class.

Please don't hesitate to als, if things are still unclear.

Thomas