VDE

James Engleback

June 27, 2022

Contents

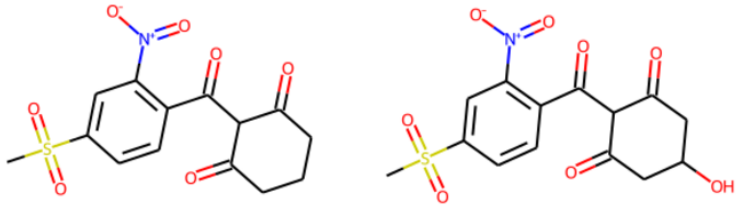

Figure 1: The HPPD inhibiting herbicide mesotrione and its primary metabolite 5-hydroxy-mesotrione in resistant strains of *A. fatua*.

1 Abstract

2 Introduction

2.1 Background

2.1.1 Herbicide Resistant Crops

Herbicide-resistant crops are important for global agriculture because they mitigate yield losses due to weeds and give farmers extra flexibility in their herbicide application programs, which is important to suppress emergence of herbicide-resistant weeds.

Herbicides kill plants by inhibiting key metabolic processes and their species-specificity is determined by susceptibility of herbicide target and their ability to metabolize the herbicide. HPPD inhibitors are a key herbicide class that cause leaf bleaching and death in susceptible plants. HPPD inhibition disrupts tyrosine catabolism which disrupts UV-protection via carotenoid production and photosynthetic electron shuttling via plastoquinone, leading to death by UV damage and radical toxicity.

Engineering HPPD-inhibitor resistance into plants have used the HPPD and metabolic enzymes from naturally resistant species like *Avena fatua*, which employs cytochrome P450 Cyp72A1 to initiate metabolism of mesotrione by ring hydroxylation at C_5. In this case, the C_5 hydroxylation acts as a target site for glutathione-S-transferases which conjugate glutathione to xenobiotics. The glutathione conjugate tags the xenobiotic for sequestration in the cell vacuole, which neutralises the threat.

Engineered Cyp72A1 has been explored as a means of HPPD herbicide in soybean, which is an important target recipient for HPPD resistance traits.

2.1.2 Cytochrome P450s

Cytochrome P450s are a ubiquitous class of heme-dependent oxido-reductases that are frequently involved in xenobiotic metabolism. Bacterial P450s have been engineered to catalyse a range of xenobiotic biotransformations. The bacterial P450 BM3 from *Bacillus megaterium* is one such bacterial P450 whose structure has been studied extensively. The A82F/F87V mutant has a broad substrate specificity, however it has no activity towards the HPPD herbicide mesotrione.

2.1.3 Virtual Directed Evolution

Enzymes can be designed computationally using a genetic algorithm that evaluates the fitness of mutants by simulating the interaction between a target substrate and the predicted structure of the mutant.

The structure of a mutant can be predicted based on a template using techniques such as side-chain repacking by stochastic sampling from rotamer libraries and loop remodelling by cyclic coordinate descent.

Binding site interaction can be predicted using molecular docking, which attempts to predict likely protein-ligand binding conformations. A combination of the energy score and the conformation of docked molecules can be used to estimate likelihood of desirable reaction and therefore the fitness of a mutant. In rounds of selection within a genetic algorithm, the fitness of a batch of mutants is evaluated by scoring desirability of protein-ligand binding, the fittest mutants are selected for breeding, in which mutants have elements of their genes recombined are further mutated, then the cycle repeats.

2.2 Technologies Used

2.2.1 Directed Evolution

2.2.2 Structure-Based Design

2.2.3 Protein Structure Prediction

2.2.4 Docking

2.2.5 Sequence Optimization Algorithms

2.2.6 Overview of this work

2.2.7 Engineering Problem

2.3 Overview of this Work

Here, in attempt to engineer a mutant of the Cytochrome P450 BM3 to hydroxylate mesotrione at the C_5 position is made by developing a *VDE* system, deploying it at scale on cloud infrastructure and identification on clusters of putatively active mutants.

3 Methods

The project was operated as a `git` repository which can be found here: https://github.com/jamesengleback/vde. The structure of the directory is:

```
docs/         # write up for this document and markdown docs
nb/           # jupyter notebooks used for data analysis
scripts/      # scripts to create and configure cloud machines to run algorithm on
vde/          # the vde algorithm configured to optimize BM3 for desirable mesotrione binding
```

This section details the implementation of this project:

- The project is dependent on a `python` packagge `enz`, developed here for protein structure prediction and molecular docking to predict the behaviour of mutants; described in 3.1.
- A score function that attempts to predict the likelihood of a C_5 hydroxylation of mesotrione is described in section 3.2.
- A genetic algorithm to optimize the sequence of BM3 mutants is discussed in section 3.3
- Section 3.5 describes execution of the algorithm at scale on cloud infrastructure.

3.1 `enz`

Abstraction and modularization of protein structure prediction and molecular docking was important for reducing complexity of experiments and developability of the algorithm. To this end the `python` package `enz` was created, an *Application Program Interface (API)* wrapper around the *PyRosetta* [1] protein structure prediction software and the *Autodock VINA* [2] binary, as well as utilities to handle file format conversion using *OpenBabel* [3]. The package is modular enough to allow replacement of its functionality-providing back-ends according to a users requirements and is hosted at https://github.com/jamesengleback/enz.

`enz` performs the following functions:

- **Protein Structure Prediction:** `enz` uses side chain repacking [4] functionality from *PyRosetta* for template-based structure prediction. This functionality is provided by *Pyrosetta*.
- **Docking:**
- **Return new atomic coordinates:** via `pandas` DataFrames, which can be used to score pose configurations.

The user-exposed command set is minimal so programs written using `enz` can be short.

3.2 Score function

Mutants were to be scored on the basis of presumed likelihood of a mutant being able to 5-hydroxylate mesotrione based on the docked mesotrione poses and their predicted binding

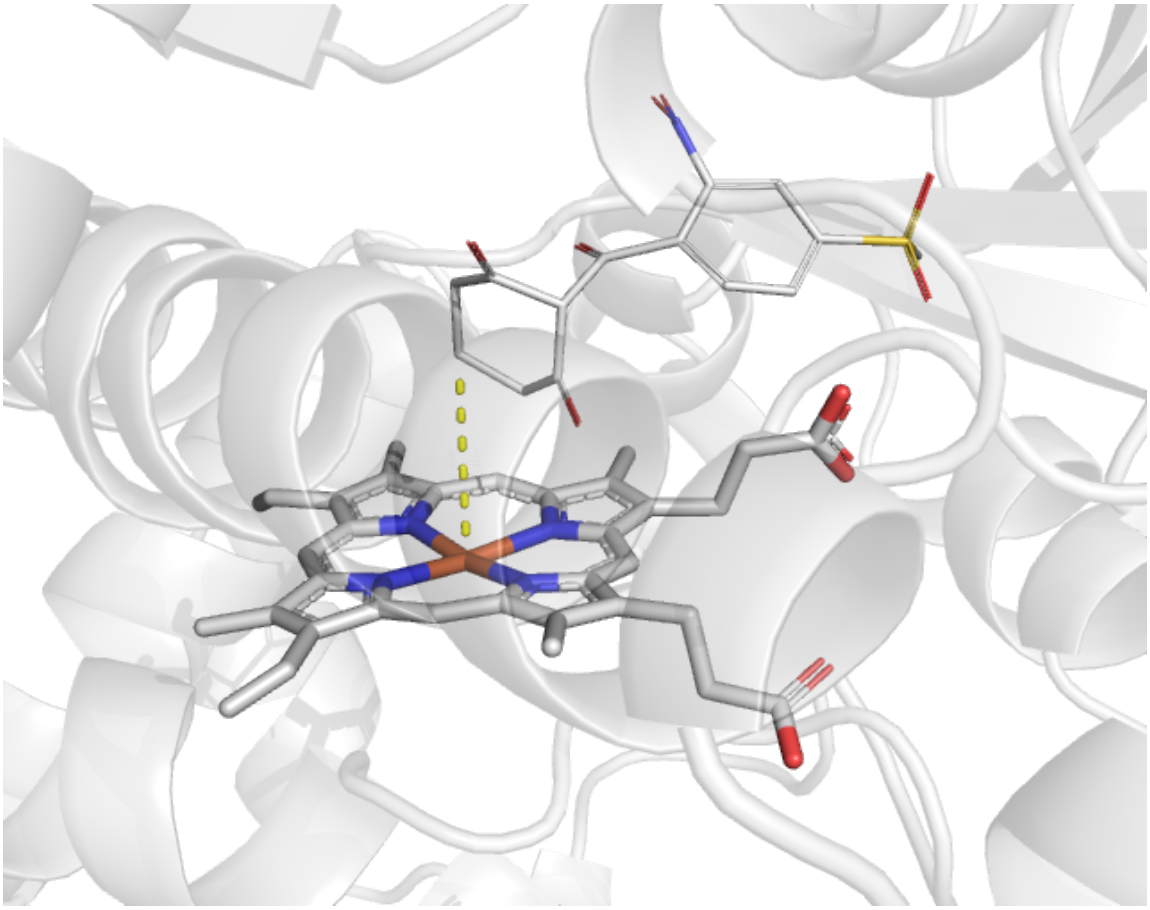

Figure 2: - Distance d between carbon C_5 of mesotrione and the heme iron of BM3, used in the fitness score (Å) marked by a yellow dashed line.

energies. Given that the likelihood of electron transfer from the heme iron to the recipient carbon atom decreases with distance d at a rate $\frac{k}{d^6}$ where d is distance between the target and the gheme iron [5].

The heuristic currently employed to estimate the desirability of each set of m docking results is described in equation 1:

$$score = \frac{1}{n} \sum_{i \in m}^{n} \Delta G_i \times d_i \tag{1}$$

where ΔG is a free energy estimation of the interaction calculated by *Autodock VINA* (given in *kcal/mol*) and d is the distance between the heme iron and the C_5 of mesotrione for each of m binding poses **(figure 2)**.

3.3 Genetic Algorithm

A simple genetic algorithm *(GA)* was used for sequence optimization during *VDE*. The *GA* was implemented in pure `python` and its built-in modules.

In this case, the *GA* repeated the following steps in each iteration:

1. **Initialize mutant population:** From the template sequence, generate p mutants each with one random point mutation.
2. **For n Iterations:**
   (a) **Evaluate *fitness* of each mutant:** Using multiprocessing, evaluate the score for each mutant in parallel, returning a mapping of sequences to respective scores.
   (b) **Select for best $\frac{1}{m}$ mutants:** where $\frac{1}{m}$ is the survival rate in each iteration.
   (c) **Repopulate gene pool by crossover and point mutation of selected mutants:** where two random members of the surviving mutants a and b are crossed by recombining sequences at a random cut point and introducing additional random point mutation. Repeat p times.

Algorithm 1 : A genetic algorithm

procedure GA($seq, popsize, n_{iter}$)
  for $p_i := 1, p_i \leq popsize$ **do**
    $pop_i := mutate(seq)$
  end for
  for $i := 1, i \leq n_{iter}$ **do**
    for all $mutant_j \in pop$ **do**
      $fitness_j := fn(mutant_j)$
    end for
  end for
end procedure

Algorithm 1 is implemented in `python` in the file `vde/ga.py` and makes use of multiprocessing to parallelise evaluations of a function.

3.4 Main Function

The program in `vde/main.py` executes the main functionality of *VDE*. It executes iterations of the genetic algorithm 3.3 where the evaluation function for a $sequence$ is:

Algorithm 2 : One fitness evaluation

procedure EVALUATE MUTANT($sequence$)
    structure = map_refold(sequence, pdb=`4KEY.pdb`) ▷ Predict mutant structure [4] [1].
    docking poses = dock(structure, mesotrione) ▷ [2]
    fitness = score(docking poses) ▷ Using score 3.2
    Return fitness
end procedure

3.5 Cloud Deployment

4 Results

5 Discussion and Future Work

References

[1] Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J Gray. "PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta". In: *Bioinformatics* 26.5 (2010), pp. 689–691.

[2] Oleg Trott and Arthur J Olson. "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading". In: *Journal of computational chemistry* 31.2 (2010), pp. 455–461.

[3] Noel M O'Boyle et al. "Open Babel: An open chemical toolbox". In: *Journal of cheminformatics* 3.1 (2011), pp. 1–14.

[4] Roland L Dunbrack Jr and Martin Karplus. "Backbone-dependent rotamer library for proteins application to side-chain prediction". In: *Journal of molecular biology* 230.2 (1993), pp. 543–574.

[5] Christopher C Moser et al. "Distance metrics for heme protein electron tunneling". In: *Biochimica et Biophysica Acta (BBA)-Bioenergetics* 1777.7-8 (2008), pp. 1032–1037.