Multiple Objective Library Design and Evaluation using an Evolutionary Algorithm constructed in an off-the-shelf Data Pipelining Toolkit

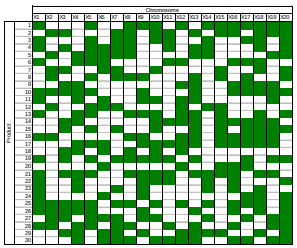
With the continued industrial trends in production of combinatorial and parallel libraries, coupled increasingly with out-sourced synthesis, the need for effective compound library design has never been greater. While there are many tried and tested techniques for library design that work exceptionally well, these usually rely on optimising a single method or are tied to a single descriptor. Where multiple metrics are needed to judge the selection from a library the number of available techniques declines sharply.
For the construction of in-house libraries, there is a need to be able to optimise for structural and pharmacophoric diversity, while simultaneously filtering for undesirable products with both physical properties and modelled ADME parameters. An additional constraint on the number of reagents used also often needs to be applied.
Genetic algorithms have a history of being applied to problems requiring optimisation of multiple simultaneous parameters and have previously been applied to library design. The piece of work described here is an effort to construct a working genetic algorithm selection process using our own in-house fingerprints, models, and descriptors utilising the commercial software Pipeline Pilot.

Genetic Algorithm Protocol

Initialisation:
- An SD file containing compounds and descriptors is read in
- Compound ID and descriptors are extracted to a text file
- An extra boolean field "Selected" is added to control the use of the compound by the GA
- The number of compounds, reagents, and clusters are then counted for use by the scoring system

Population
- Each model in the population is created from an empty data point
- It contains only the ID#, score, and the path to model file on disk/cache

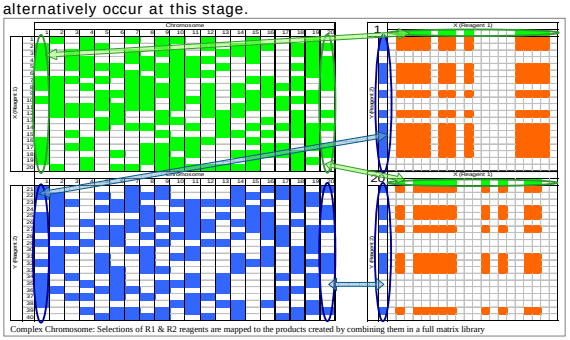
Constructing the Chromosome
- Inside each model, the text file for each model is read in
- Each compound gets assigned a random number (0-1).
  - Those scoring above 0.5 are selected in that model.
- The unselected compounds are filtered off and discarded
- The properties of the remaining compounds are used to score the starting models



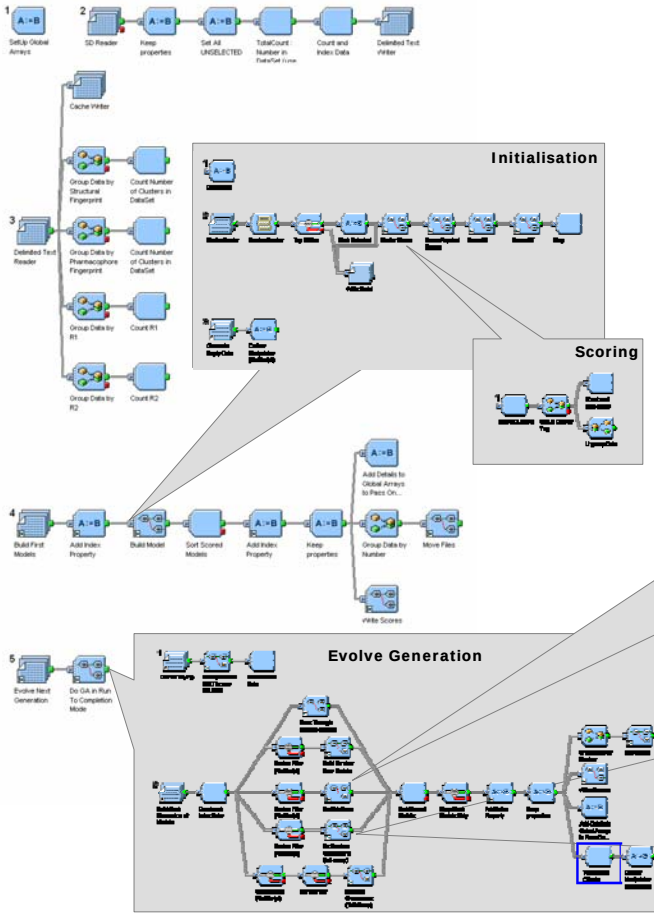
Sample chromosome. Compounds with the selected flag set "True" are shown in red. This is assigned at random in the first iteration and optimised there after.

Expansion to Complete Matrices

- Each gene corresponds to the selection of a reagent (R1 or R2)
- An extra component is inserted to select each compound based on the selection state of it's constituent reagents.
- For a massive virtual library, an enumeration of these could alternatively occur at this stage.



Complex Chromosome: Selections of R1 & R2 reagents are mapped to the products created by combining them in a full matrix library

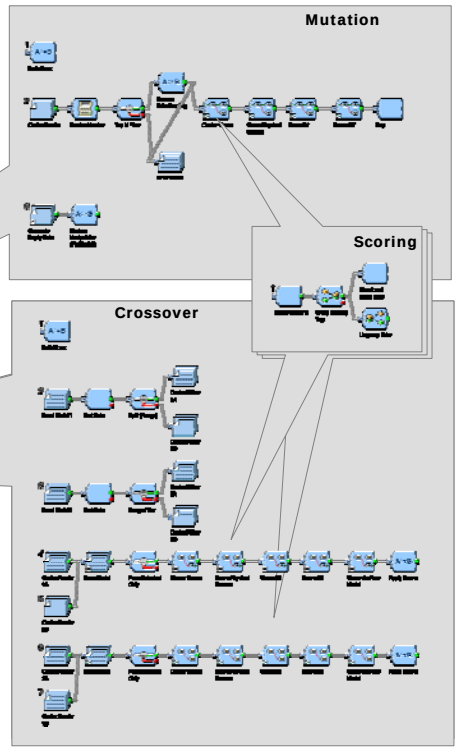


Genetic Operators

Mutation
- Each model is read in sequentially
- Each compound is assigned a random number
- The top n% are chosen and their selected flag is toggled
- The selected compounds are then scored

Crossover
- A pair of models is read in
- The compounds inside each are sorted by ID
- A random point at which to split the library is selected
- The head of model A is combined with the tail of model B, and vice versa
- This works as both libraries are the same length
- Crossover can be done between random models or against the best performing model of the previous generation

Other Operators
- If desired, other operators can be used at this stage
  - For example, increase (or decrease) the number of selections
- Random new models can also be added.



Scoring functions

- Cluster coverage in fingerprint space (structural)
- Cluster coverage in fingerprint space (pharmacophoric)
- The number of reagents (R1) selected
- The number of reagents (R2) selected
- Deviation from the ideal desired library size
- The overall property "score" of the library (where each Lipinski-type violation adds a negative score)

- Each score is normalised and summed to give a score for the library

- Additional scoring functions can be added as needed

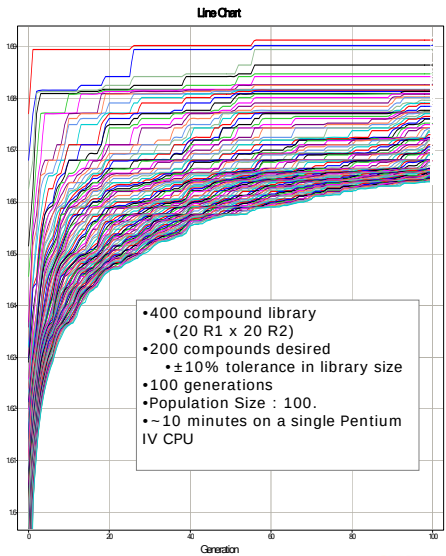
- Use of a pareto function in evaluating these scores is being investigated.

Convergence Criteria

The algorithm can be set to stop when:
- An average score is reached
- *n* models reach a certain score threshold
- The scores of the top *n* models are stable for *i* generations.

- Additionally, as the models are saved after every step and the procedure can be restarted from these to continue optimisation as needed

Optimisation



Conclusions & Future Directions

This system, while in it's infancy, has already been demonstrated to work well and allows a very flexible system of picking and analysis of compound libraries. The future possibilities of such a system are limitless, as the method of constructing the chromosome and scoring it can be configured to multiple different tasks.

The next planned development of this system is in the area of selection of compounds for screening, particularly where a subset of the compound bank is to be screened as a prequel to a HTS. The system will be configured to select from compounds using a variety of metrics and models to ensure that the ideal compound selection is made.

Inflammation Science

Gareth Wilden, Dept. of Physical and Metabolic Science,
AstraZeneca R&D Charnwood, Loughborough, UK. E-mail: gareth.wilden@astrazeneca.com

AstraZeneca