**Large‐scale identification of genetic design strategies using local search**

**Desmond S Lun, Graham Rockwell *et al.***

The article describes an algorithm called Genetic Design through Local Search (GDLS) for finding effective genetic manipulations. The algorithm is heuristic-based, scalable and is based on local search with multiple search paths. This results in a quick and effective search of the space of genetic manipulations. The algorithm can thus find genetic design with greater production of desired metabolites way faster and more effortlessly than previously used algorithms.

**Introduction**  
The article does a very good job of mentioning previously developed algorithms that seek to solve the same problems, describing the state of the art well. However, none seem as comprehensive as GDLS. Thus, GDLS is innovative and an improvement over the others.

**Methods**  
The algorithm is demonstrated on gene knockouts, but the article states it can work for any bi-level FPA framework transformable to a mixed-integer linear program. It would be helpful to have more than one example of how the algorithm is applied.

The algorithm’s performance is measured in comparison to a global MILP search algorithm called OptKnock, which usually runs on a full model without FBA model reductions. OptKnock uses a different but more numerically stable MILP solver. In this case, a seemingly fair comparison was achieved by running both algorithms on the same reduced model along with GDLS’s slower but freely available solver. The algorithms were tested by maximizing acetate and succinate production in *Escherichia coli* using the latest available metabolic model at the time, iAF1260.

The code is supplemented along with the article making it easily possible to test whether the results described in the article are true to life.

**Results**  
The results are very convincing. The GDLS algorithm’s purpose is to check for likely effective simultaneous gene manipulations which conventional global search algorithms miss. GDLS succeeds in its mission, giving better results in shorter time than its global counterpart. Instead of CPU time growing exponentially with each gene manipulation, it grows linearly. However, for the first *k* genes, the algorithms are the same and take the same amount of time.