ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

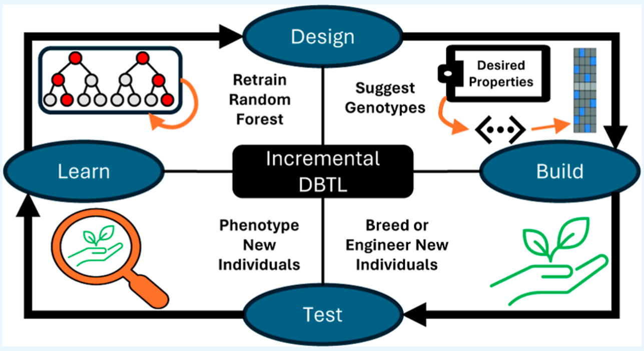
Incremental Inverse Design of Desired Soybean Phenotypes

Joseph Zavorskas, Harley Edwards, Mark R. Marten, Steven Harris, and Ranjan Srivastava*

Cite This: *ACS Omega* 2024, 9, 41208–41216

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: We present an application of computational inverse design, which reverses the conventional trial-and-error forward design paradigm, optimizes biological phenotype by directly modifying genotype. The limitations of inverse design in genotype-to-bulk phenotype (G-BP) mapping can be addressed via an established design paradigm: "design, build, test, learn" (DBTL), where computational inverse design automates both the design and learn phases. In any context, inverse design is limited by the fundamental "one-to-many" nature of the inverse function. G-BP inverse design is further limited by the number of single nucleotide polymorphisms that can be made to a member of the population while maintaining feasibility of genotype creation and biological viability. Considering these limitations, we propose a design paradigm based on incremental optimization of phenotype through a combined computational and experimental approach. We intend this work to be a foundational synthesis of well-known techniques applied to the context of genotype-to-bulk phenotype inverse design, which has not yet been performed in the literature. The design pipeline can optimize phenotype by either directly proposing genotypic changes, or simply by suggesting parents to be used for selective breeding. The soybean nested association matrix data set is used to present an in silico case study of the design pipeline by performing optimization that maximizes protein content while constraining other phenotypes. A random forest (RF) is used to model the genotype-to-phenotype relationship, and a genetic algorithm is used to query the RF until a feasible genotype with desired phenotype is discovered. After 20 in silico DBTL cycles, a final population of individuals with a mean protein content of 36.13%, an increase of three standard deviations above the original mean is suggested.

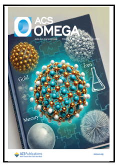
1. INTRODUCTION

Soybean production and processing is a globalized industry that supports a variety of other industries, including animal feed,[1] bioenergy,[2] plastics[3] and human foods.[4] This diversity of applications is possible because the soybean plant has many useful products including protein-rich soymeal[5] and soy oil, which contains the common food ingredient soy lecithin.[6] Due to the poultry industry's reliance on soybean meal,[7] our focus is on optimizing soybeans for protein production.

The concepts of forward and inverse design will frequently be referenced. Forward design refers to an iterative "trial-and-error" design paradigm in which design parameters are selected and then the resulting properties are experimentally determined.[8] In contrast, inverse design uses computational modeling or simulation alongside optimization to suggest design parameters based on certain desired properties.[9] Forward and inverse design terminology are most frequently used in metamaterials design, one of the first fields to establish an inverse design paradigm. Metamaterials, and more specifically plasmonic materials, are currently the most prolific field for inverse design. A comprehensive review of plasmonic inverse design by Ren S. et al.[10] reveals that there are two steps required for inverse design: (1) model the forward function, (2) optimize the inverse function using the forward model as a simulator. Plasmonic inverse design is so prolific because step 1 is already solved by deterministic forward function simulation using finite difference time-domain.[11] We have designed a metamaterials-inspired inverse design pipeline for optimizing soybean phenotype by synthesizing techniques from metamaterials and biology. Linking soybean genotype to phenotype is nontrivial, so mathematical modeling and/or machine learning (ML) are necessary for forward functional simulation in this context.

In biology, a true computational inverse design paradigm has been established in de novo protein design. De novo protein design benefits heavily from structural and functional redundancy across kingdoms of life. For instance, Madani et al. train their large language model for protein sequence design on over 280 million natural protein sequences.[12] Numerous

Received: February 21, 2024
Revised: September 4, 2024
Accepted: September 12, 2024
Published: September 30, 2024



https://doi.org/10.1021/acsomega.4c01704

publications by the Baker Lab[13−17] use molecular dynamics simulations or the networks RoseTTAFold[18] and AlphaFold2,[19] developed from similarly large data sets. Working at the genotype-to-bulk phenotype scale, only phenotypes as genotypes from the target organism are used, and readily available simulators that can predict phenotype from genotype are not available. Also, the intensive design of a particular protein cannot necessarily guarantee a desired change in genotype. Thus, we will explore how genotype-to-bulk phenotype (G-BP) design is currently performed in plant breeding as a primer.

While ML is currently in use within plant breeding and agronomy, it has not yet been extended into a computational inverse design paradigm. There are two general areas within agronomy that ML is commonly applied. These fields are genomic selection/genomic prediction (GS/GP) and phenotypic prediction. GS is an improvement upon its predecessor, marker assisted selection,[20] in that it does not require an explicit association between a biomarker and a trait.[21] Rather, GS uses genome-wide biomarker data, typically single nucleotide polymorphisms (SNPs),[22] which are low-cost to acquire and high density within the genome.[23] GS also does not establish an explicit connection between any biomarker and a given trait, but rather uses computational methods to capture several minor genotypic effects on a single trait.[24] Typically, a GS model is trained on data that has both phenotype and genotype information and can then be used to predict data with only genotype information. Because gathering phenotype information is not required after the model is constructed, time is saved in selecting parents with highly desirable traits for breeding.[25]

GP techniques used for GS span linear, Bayesian, and ML techniques. A common linear technique used in GP is best linear unbiased prediction, available with its spinoff techniques in the package rrBLUP.[26] Bayesian techniques, collected in the package BGLR[27] are better at capturing nonlinear and multivariate relationships than rrBLUP. ML techniques have been extensively benchmarked on known genotype/phenotype data sets such as wheat, soybean, and sorghum.[28,29] The most common ML methods used in GP are decision-tree based methods such as random forests[30] and gradient boosted trees (i.e., XGBoost),[31] and support vector machine.[32] Deep artificial neural networks[33] and convolutional neural networks[34] have also been applied to GP. Generally, it appears that there is a trade-off between model accuracy and model complexity (and therefore training time).[35] Though less relevant to G-BP design, ML has also been used to perform phenotype prediction from a plant's visual characteristics[36,37]

SNP arrays are extremely high-dimensional data sets, so there is also a third requirement for inverse design in this case: (3) dimensionality reduction. This case study uses the soybean nested association matrix (SoyNAM) data set, which has already been processed extensively by a genome-wide association study[38] and SNP saliency metrics.[39] Feature selection by these techniques or by computational techniques such as RF[40] are acceptable as dimensionality reduction. Much of the plant breeding literature is focused on the established technique of GS/GP, which aims to select parents which are most likely to lead to desirable phenotype change. Therefore, there has not been a focus on de novo computational inverse design through a G-BP lens.

1.1. Limitations/Necessary Constraints.

First, while genetic engineering in soybeans is common and impactful,[41] European markets and others are still wary of genetically modified organisms (GMOs) in foods.[42] Any attempts to design or optimize soybean phenotype will therefore be limited by consumer perception. Selective breeding is not commonly considered GMO and can be a good workaround for this issue. This pipeline both keeps track of suggested changes on a SNP-by-SNP basis, and tracks the parents each suggested change comes from, making selective breeding easier. Agronomists trying to quickly improve their crop while avoiding GMO concerns can benefit from this technique, which circumvents typical "trial-and-error" forward design.

Next, while SNPs are the most descriptive way to encode genotype, working at the base pair level introduces a few limitations. The number of genotype changes that can be made in one iteration is greatly limited due to concerns with both viability and feasibility. Regarding viability, the more computationally suggested SNPs, the greater the probability that lethal or counterproductive mutations will be introduced. Feasibility is a more complicated consideration: using available technology, is it actually possible to create the suggested genotype? Two possible options for creating a suggested genotype are as follows: (1) perform selective breeding with parents likely to produce desired genotype; (2) use a CRISPR-based technique.[43] The former will allow genetic changes outside of the suggested SNPs but is simpler to execute. The latter can make a small number of targeted genetic changes but introduces GMO concerns and requires knowledge of extraneous biological techniques. Considering these limitations, we selected a constraint for the maximum number of SNPs which is large enough to allow for reasonable phenotype change but small enough to use either technique effectively.

All inverse design techniques are limited by the "one-to-many" nature of inverse function mapping.[44] In a biological context, "one-to-many" refers to the fact that a single output phenotype can be created by countless input genotypes. As a result, an optimization algorithm is much more likely to find local optima during optimization, and certain modeling techniques will not be able to complete training.[45] The one-to-many problem is further exacerbated by the high-dimensional nature of SNP data; inclusion of more variables introduces more nonunique solutions. As a result, an unconstrained genetic algorithm (GA) attempting to optimize a phenotype via genotype can suggest genotypes that require hundreds or thousands of genetic changes.

These limitations are critically detrimental to the rational use of inverse design in a direct genotype-to-bulk phenotype context. The proposed design pipeline will address the aforementioned limitations in multiple ways. Our principal contribution is a synergistic combination of computational techniques and experimentation inspired by the "design, build, test, learn" (DBTL) paradigm.[46] DBTL is already commonly used for intelligent, recursive forward design in metabolic engineering and systems biology.[47,48] We believe that this work goes beyond current DBTL uses in biology due to its integration of computational inverse design, which bolsters and streamlines both the design and learning aspects. This design paradigm can also be generalized to other fields or other biological structures to perform inverse design. In addition, to our knowledge, a paradigm to perform direct G-BP inverse design has not yet been established within the literature. The work described here is a novel synthesis of many established techniques (RF, GA, DBTL, and inverse design) to address the challenges that G-BP inverse design poses.

2. METHODS

2.1. Data Set. For this work, we used the SoyNAM data set.[49] Using this data set is advantageous because it is already well-curated, and feature selection has already been performed. In addition, it has the largest population size available for a well-curated, SNP-based data set. The data set contains 5487 individuals, each with the same 4401 SNP loci describing their genotype. Many phenotypes are available, but we avoid integer and Boolean phenotypes. For the case study, height (cm), protein content (% by mass), and seed size (weight in g of 100 seeds) were selected as outputs. Genotypes are pseudo-one-hot encoded with a 0, 1, or 2 for reference, heterozygous mutation, and homozygous mutation, respectively.

2.2. Compute. All code was run on an i7-8550U CPU with 12 GB of DDR4-2400 Hz RAM. The problem statement and proof of concept each take about 5 h, while the full case study requires 48 h of processor time.

2.3. Packages and Hyperparameters. The inverse design framework consists of a RF in Scikit-learn v1.2.2[50] as a forward simulator (genotype-to-phenotype), and a GA in pyGAD v3.0.0[51] to perform optimization on the inverse function (phenotype-to-genotype). Uniform Manifold Approximation and Projection (UMAP) v0.5[52] was used to generate 2-D renderings of genotype data.

For each algorithm, different hyperparameters were used to illustrate the problem statement, provide validation and proof of concept, and finally to carry out the case study. Table 1 displays GA hyperparameters used in each part of this project. For every RF, default sci-kit learn parameters for a RF regressor were used. Crossover and mutation probabilities were intentionally kept low to avoid excessive violation of the SNP constraint.

Table 1. GA Hyperparameters for Various Inverse Design Techniques[a]

parameter	PS GA	POP GA	CS GA
total iterations	10	10	100 (20 DBTL x 5)
repeats allowed	N/A	300	10
generations	50	20,000	50
population size	20	20	20
SNP tolerance	20	20	10
crossover probability	30%	30%	30%
mutation probability	0.2%	0.2%	0.2%
elites kept	5	5	5

[a]PS: problem statement, POP: proof of principle, CS: case study.

Default sci-kit learn parameters were used for the RF. 5-fold cross-validation was used to determine the accuracy of the RF on genotypes/phenotypes not part of the training set. Table 2 displays the results of this 5-fold cross-validation for predicting height and protein together, and height, protein, and seed size together. Including more phenotypes introduced more error but may be more relevant to an agronomical context.

Table 2. Five-Fold Cross-Validation of RF for Various Phenotype Combinations[a]

	protein only	height and protein	height, protein, and size
fold 1	1.44%	3.56%	4.31%
fold 2	2.27%	5.52%	5.53%
fold 3	1.46%	3.94%	4.48%
fold 4	1.56%	4.15%	4.86%
fold 5	1.80%	5.53%	6.06%
average MAPE	1.71%	4.54%	5.05%

[a]Mean average percentage error was used as an error metric.

2.4. One-to-Many Problem Statement. For the problem statement and subsequent proof of concept, a simple fitness function was used: absolute error from desired phenotype as predicted by the RF. No phenotype constraints were applied; however, a constraint disallowing more than 20 SNP changes from an existing member of the population was enforced to address feasibility and viability concerns. The fitness function began by calculating the Hamming distance of each solution's genotype to each other member of the population. There are three options:

1. Solution is a member of the population: assign fitness = 0.
2. Solution has a neighbor within SNP constraint: calculate fitness normally.
3. Solution has no neighbors: replace with a known genotype, calculate fitness.

This genotype constraint guaranteed that all solutions will be anchored to an existing member of the population and will be within the desired number of SNPs.

First, all individuals with a genetic "neighbor" within 20 SNPs or less were found. These neighbors can be found in Supporting Information S1. The height and protein content phenotypes of an individual from the largest cluster of 19 individuals within 20 SNPs of each other were selected as a target, and this individual was deleted from the population. The goal was to validate the inverse design pipeline by regenerating the genotype of this removed individual by targeting its phenotype with inverse design. After training an RF on the remaining population, 10 GA runs were performed which used the new RF to predict phenotype within the fitness function. Each run, the individual closest to desired phenotype was saved and plotted against the original population in 2-D using UMAP.

2.5. Proof of Principle. To prove that the GA is capable of regenerating a certain genotype by setting its phenotype as the target, a smaller-scale experiment was performed using the same neighbor cluster described in Section 2.4. The RF was still trained on the full population minus the target. However, when a solution was above the maximum number of SNPs allowed, the pool of individuals it could be replaced with only contained the neighbors within its cluster. Also, only the 19 individuals in the cluster were checked by the fitness function, so this GA is able to operate much faster. Thus, more generations were allowed to find and settle in an optimum, as shown in Table 1. Ten GA runs were performed for each of the 19 individuals in the cluster.

2.6. Case Study. For the case study, phenotype constraints and a more stringent 10 SNP constraint were enforced. Ten SNPs were not necessarily the ideal number; one must strike a balance between taking small enough steps to retain biological viability and feasibility of creation and taking large enough steps to make significant phenotypic change. The SoyNAM data set is very well-curated, with a large population, a relatively low number of high-confidence SNP markers, and high minor allele frequencies (10−20%). Thus, we approached the problem with confidence that making multiple SNP changes at once would not result in an unviable organism. As a general rule of thumb, the less curated the data set, the fewer

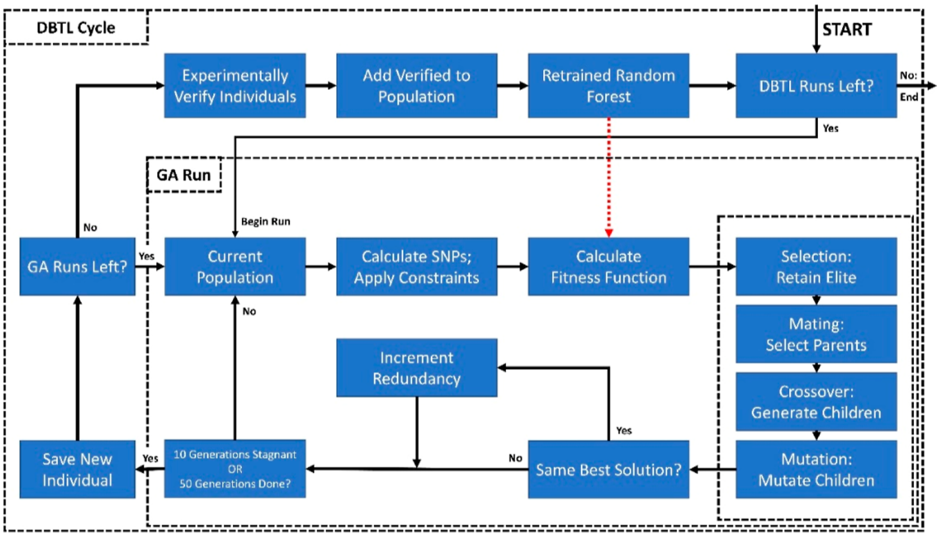


Figure 1. Full optimization pipeline. Inner loop: single GA run; keep individual closest to desired. Outer loop: after all GA runs finish, experiment and update new individuals. Red dotted line: RF is used to predict phenotype within the GA's fitness function.

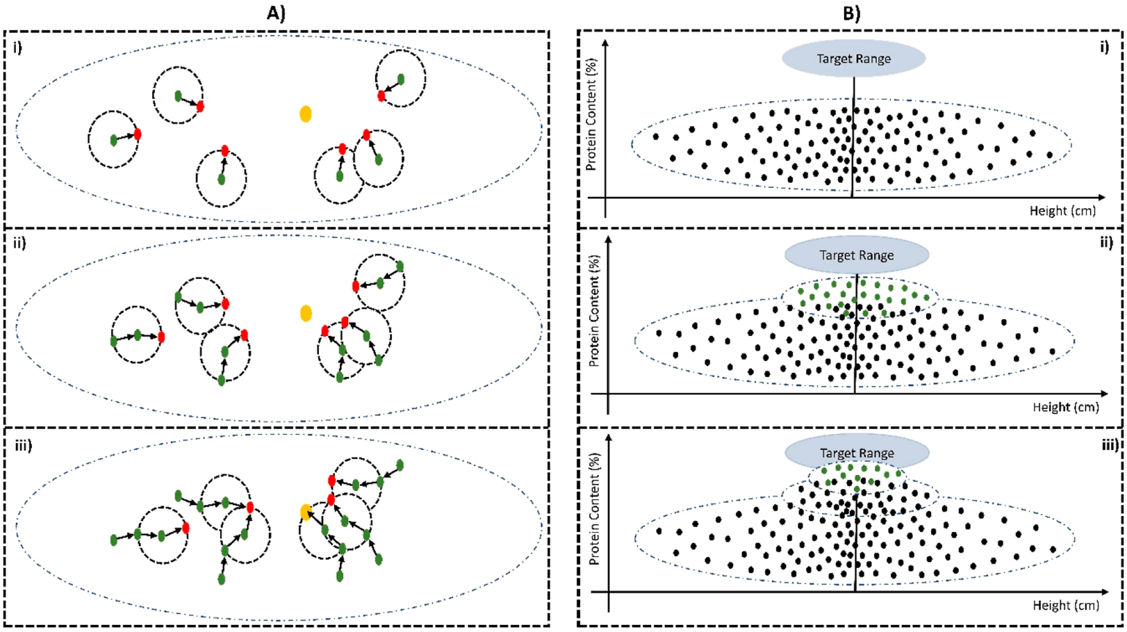


Figure 2. (A) Three consecutive DBTL cycles of abstractly represented genotype, depicting the SNP limitation as individuals moved toward the desired genotype (gold). (B) Three cycles of phenotype optimization by DBTL, illustrating the creation of new phenotypes and their new distributions until the target range was reached.

SNPs should be allowed per iteration. Unfortunately, there is no good way to guarantee viability computationally, but tuning the SNP constraint appropriately can help mitigate the chance to generate unviable or infeasible solutions.

When fitness was calculated normally, multiple constraints and rewards were applied to attempt to produce a compact soybean plant with small, protein-dense seeds. More specifically, the fitness functions (in eqs 1−3) for height (H) and seed size (S) penalize the solution heavily for values larger than each phenotype's mean value. These constraints are imposed to avoid unrealistic solutions (i.e., growing excessively tall plants with large seeds to achieve higher protein contents). The fitness function for protein (P) heavily penalized solutions below the mean value and afforded progressively larger rewards for phenotypes significantly above the mean. "Sol" is the phenotype of the current solution. μ and σ refer to the mean and standard deviation of the phenotype in the original population, respectively. Seed size rewards and penalties are larger because seed size has a smaller range.

2.7. DBTL Pipeline.

The overarching design pipeline we are suggesting is inspired by DBTL, iteratively suggesting new genotypes which are expected to provide improvements on phenotype in an incremental manner. A conceptual diagram of a full DBTL cycle is included in Figure 1. This pipeline is unique, because the suggestion of these new individuals (design) and their subsequent addition to and analysis with the original data (learn) is entirely automated via an inverse design framework. In practice, creation of suggested genotypes (build) and measurement of their phenotypes (test) must be done experimentally and must be done frequently to balance out the computational component.

In this case study, a DBTL-like pipeline was used to suggest small populations of soybean genotypes, which successively produced increased protein content subject to constraints. A conceptual illustration of both genotype and phenotype approaching their target values through DBTL is included in Figure 2. To be more specific, during the case study a GA runs five times to suggest five genotypes per DBTL cycle. At this point, these genotypes would need to be experimentally created, grown, and tested for phenotype. However, in this case study we took the RF's phenotype prediction as fact, added the GA-generated individuals to the population, and began the GA runs again.

3. RESULTS AND DISCUSSION

3.1. Problem Statement. Before presenting the case study, we will demonstrate the one-to-many problem. Ten GA runs were performed using a target phenotype whose genotype is known to have a cluster of neighbors within the 20 SNP limit. The results are displayed via UMAP representation, as shown in Figure 3. Hypothetically, validation of the pipeline by reconstructing this genotype should be feasible within the constraints of the pipeline. However, not only can the genotype not be reconstructed, but the GA also produced a different suggested genotype for each run despite having the same target phenotype. If many GA runs were performed, the original genotype would likely be found eventually; however, using the same GA hyperparameters as in the case study, reconstructing a specific genotype appeared prohibitively difficult. Even after drastically reducing the search space by imposing an SNP constraint, many local optima clouded the search space.

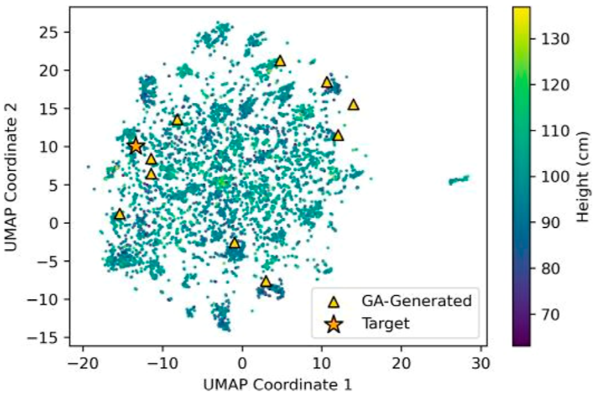


Figure 3. Dimensionally reduced demonstration of the one-to-many problem of inverse design. Genotype (in 2-D) of the selected target phenotype is displayed as an orange star. All GA-generated yellow triangles and the orange star shared the same predicted phenotype. This plot demonstrates the difficulty of exactly regenerating a genotype by targeting its phenotype through inverse design.

The purpose of this exercise was to demonstrate the drastic effects of the "one-to-many" nature of inverse function mapping on G-BP inverse design. As shown by UMAP in Figure 3, the same target phenotype could be represented by many, vastly different genotypes. The large changes in design parameters that generating these individuals would require are not feasible in G-BP design. It is clear that another approach is necessary, so we embraced "one-to-many" by taking an incremental approach.

3.2. Proof of Principle. To address the stated problem, the pipeline must be validated on a smaller scale. Table 3 displays the results of each individual within the largest cluster of neighbors after only allowing that cluster to act as parents. Many members could now be found using inverse design, but this small-scale technique still fell short on a few cluster members. The one-to-many problem still introduced difficulty in recreating an exact genotype, even at the scale of a few SNPs. Only 10 of the 19 members of the neighbor cluster can be found from the others after 10 GA runs. For the others, there were phenotype optima along the way that the GA finds and becomes stuck in. It appeared that if the target genotype was able to be reconstructed, it would be done relatively quickly. Many of the successful solutions were performed in one or two iterations.

This smaller scale validation showed that it was in fact possible to reconstruct a genotype via GA/RF inverse design by targeting its phenotype. However, the goal of inverse design was not to regenerate a specific genotype but rather to create a phenotype of interest by manipulating genotype. Most importantly, we can learn from the results of this proof of principle: due to the infinite number of solutions to the inverse genotype-phenotype function, DBTL should be introduced to allow for a more incremental and scaffolded approach to optimization. When an acceptable local optimum is found, it is best to avoid excess computational work and instead perform experimental verification and learn from the suggested individual.

3.3. Case Study. The goal of this in silico case study, to maximize protein content by inversely designing genotype in an incremental manner, was successful. Using the DBTL pipeline, 100 unique genotypes were created which are displayed on a histogram alongside the original distribution of the population in Figure 4A. Experimental verification was simulated by adding individuals to the population and retraining the RF between each DBTL iteration. Importantly, we recognized that by performing this case study fully in silico it was likely that some overfitting occurred. Overfitting will be addressed at length in the following sections; however, if this pipeline is applied to in vivo G-BP inverse design, overfitting is mitigated by intermittent experimental verification.

Table 3. Proof of Principle Results for Individuals within Cluster of Neighbors[a]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
iterations required	–	**1**	–	–	–	**1**	**6**	–	**1**	**1**	–	**1**	**1**	**1**	–	–	**2**	**2**	–
SNPs of best fitness	6	**0**	7	10	6	**0**	**0**	1	**0**	**0**	4	0	**0**	**0**	5	7	**0**	**0**	5
lowest SNPs required	2	**0**	5	10	5	**0**	**0**	1	**0**	**0**	2	0	**0**	**0**	2	6	**0**	**0**	2

[a] "SNPs of best fitness" indicates the genetic difference between the individual with closest predicted phenotype to target. "Lowest SNPs required" is closest genotype to the target achieved during any GA run. Individuals whose genotype could be reconstructed are **Bolded**, and those not found in 10 iterations are marked by "–".

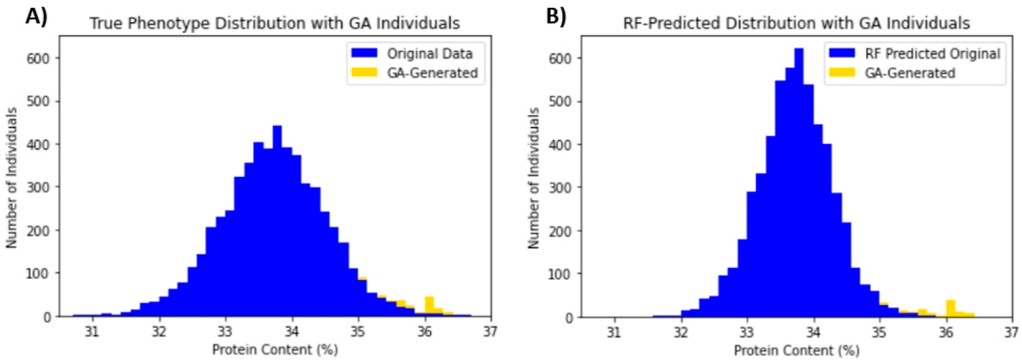


Figure 4. Both stacked histograms display the protein phenotype of the yellow GA-generated individuals compared to (A) the true protein phenotype distribution and (B) the protein phenotype distribution as predicted by RF. The RF tends to compress the phenotype distribution toward its mean. (A,B) Both use the same genotypes as input for the blue data, but (B) is the phenotype generated by the pipeline. GA-generated individuals begin to form a new distribution around 36% protein content, toward the tail.

Regardless, after 20 DBTL cycles, it appears that the GA-generated individuals formed a new population distribution. The final DBTL cycle generated five individuals whose average protein content was 36.13%, an increase of 3.04 standard deviations based on the original distribution. On the histogram, many individuals cluster within 36.0−36.2% protein content, showing that the pipeline is likely approaching the point of diminishing returns. In Supporting Information S1, genotype and phenotype are provided for all final GA-generated individuals.

As shown in Figure 5, the average phenotype difference between a new individual and its closest neighbor ("parent") in the original population was small to begin with but became larger as more DBTL cycles occured. At a late stage of inverse design, the pipeline appeared to latch onto strong individual and continue make small changes to them. This is likely an artifact of overfitting, because as new individuals were added to the population, they were weighted more heavily in RF training. Therefore, the same parent (e.g., parent #3890) can appear to produce larger and more desirable phenotype differences in future generations (shown in Table 4) due to the influence of the new individuals on the RF. This result further evidenced that this pipeline must be performed with frequent experimental verification. Note that Figure 5 shows phenotype differently between only the original population and new individuals, showing that phenotype changes move slowly at first, then very quickly, followed by reaching a point of diminishing returns. Table 4 includes new individuals in the analysis, and shows that in late stages, small changes dominate, indicating DBTL may be unable to further produce improvements without experimentation.

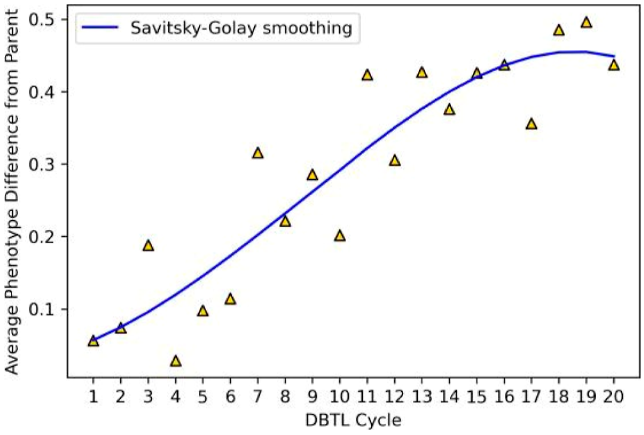


Figure 5. Average phenotype difference from parent (original population) genotype based on RF prediction. Savitsky−Golay smoothing was used to generate a smooth fit curve for the data.

Table 4. Results for the Five Individuals in the Final DBTL Cycle[a]

	1	2	3	4	5
parent index	3890	**5574**	**5495**	**5549**	**5539**
phenotype difference from parent (RF)	0.626	−0.01	−0.01	0.06	−0.01
SNPs required from parent	8	10	8	7	10

[a] At this stage, small changes are all that can be made. **Bold**: individuals previously generated by inverse design.

Selected metrics for the five individuals produced by the final DBTL cycle are included in Table 4. A similar analysis for all individuals generated by the design pipeline is included in Supporting Information S1. As shown in Table 3, across all solutions the pipeline tends to use close to or more than the maximum number of allowed SNPs for each solution. This driving force to use more SNPs is also justification for the application of the SNP restriction as a hard constraint, while the phenotype constraints are applied as softer penalty and reward functions. This SNP constraint is what prevents the GA from making infeasible suggestions to acquire phenotype improvements. As shown in Figures 4 and 6, this DBTL pipeline pushes the boundaries of existing phenotypes, but does not make outlandish or infeasible genotype suggestions.

Phenotype can be a soft constraint because optimization of protein content does not necessarily drive the other phenotypes in a certain direction to maximize protein content. As shown in Figure 6A, the pipeline was capable of finding solutions slightly below, just at, or slightly above the population mean. The solutions slightly above were allowed because the reward for exceptional protein content outweighed the penalty for minor violations of height and seed size constraints. On the whole, however, the pipeline seemed to do better at following the seed size constraints than the height constraints.

A RF with default parameters was used as the forward function simulator. It appears that a more powerful technique, such as XGBoost[53] or an artificial neural network,[54] with tuned hyperparameters could be superior due to the high-

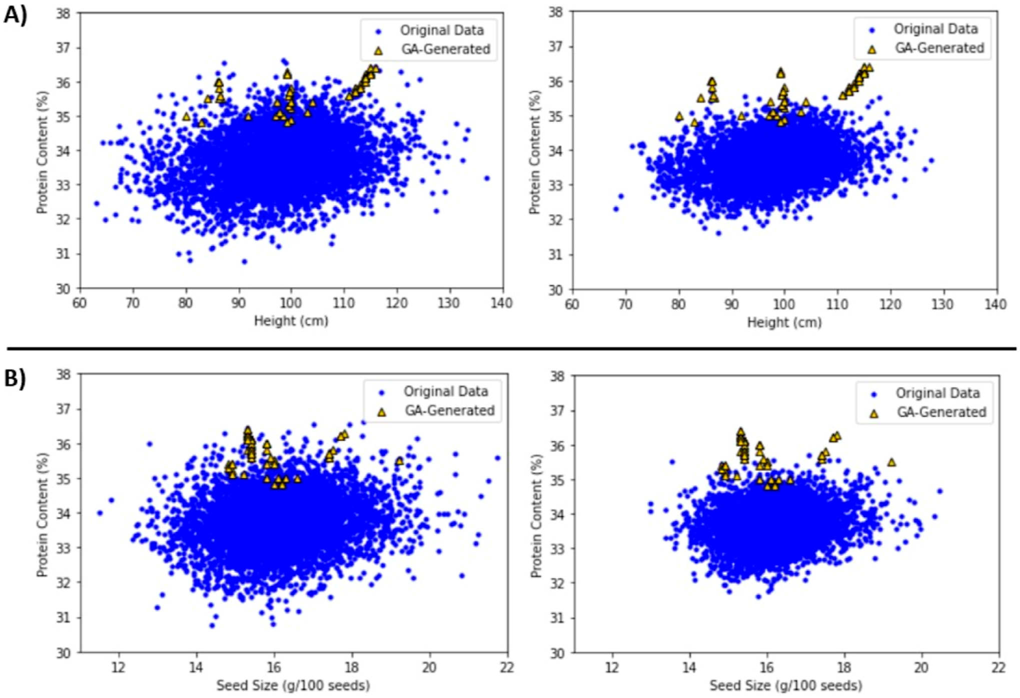


Figure 6. Scatterplots displaying phenotype distributions of (A) height and (B) seed size vs protein content. Left Figures: DBTL solutions compared to true phenotypes. Right Figures: DBTL solutions compared to RF-predicted phenotypes. GA-generated individuals are realistic solutions that push the boundary of protein content. This effect is more apparent when compared to the phenotypes the RF is able to predict.

dimensionality and interdependence of the data. The effects of using an untuned RF are evident in Figures 4B and 6B, which display DBTL solutions alongside the population distribution as predicted by the RF. The RF is poor at predicting phenotypes near the extremes of the population, resulting in a compression of the phenotype distribution. When compared to the distribution the RF is able to predict, the GA-generated individuals appear to have a much more significant increase in protein content than when compared to the true phenotypes. Therefore, a technique that can more faithfully predict the full range of true phenotypes would be able to further push the boundaries of protein content.

Azodi et al.[35] have benchmarked many regression techniques for SoyNAM and other biological data sets. They show that gradient boosted trees (e.g., XGBoost) and ANNs both have increases in accuracy over RF. Azodi et al. also benchmarked many linear regression techniques for phenotype prediction, which can match or exceed the accuracy of XGBoost and ANNs. In future work, switching to one of these algorithms is highly recommended; however, note that both XGBoost and ANNs are less accurate than RF or linear methods for biological data sets other than SoyNAM. There are two reasons for this. First, the SoyNAM data set is extremely well-curated and contains few features compared to other data sets: 4401 SNPs determined by multiple techniques to have significant effects on phenotype. More importantly, the SoyNAM data set is a collection of nearly 5500 individuals; therefore, it has the best feature-to-sample ratio of any publicly available SNP-based database. In conclusion, the choice of regression algorithm must be based on the data set at hand and time constraints.

4. CONCLUSION

Any design paradigm in G-BP inverse design must intelligently approach the inherent one-to-many nature of inverse function mapping. Due to this, we present a computational inverse design pipeline inspired by design, build, test, learn, in which the "design" and "learn" phases are semiautomated. This technique can save time and resources over trial-and-error forward design techniques by either directly suggesting a genotype to be created or by intelligently suggesting the parent organism(s) for use in selective breeding.

To our knowledge, this is a novel application of computational inverse design in a direct genotype-to-bulk phenotype context. In addition, this generative DBTL framework is generalizable to any field, especially those in which large changes in design parameters are not feasible. At a smaller scale, it is also generalizable to other genotyping data than SNPs.

Incremental inverse design as presented is an effective technique for performing genotype-to-bulk phenotype inverse design. There may be other techniques that are superior to this particular application for biological inverse design, but we simply intended to lay the groundwork for inverse design to occur within the field of agronomy/plant breeding or general genotype-to-phenotype mapping. Despite issues with overfitting stemming from introducing in silico individuals to the pipeline, we have presented a useful in silico demonstration of computational inverse design in biology that can be used to make phenotype improvements while maintaining viability and practicality. Phenotype and SNP constraints are enforced to make performing computational and experimental design in an alternating and incremental fashion feasible for agronomists and engineers alike.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.4c01704.

S1 (xlsx)—DBTL inputs and results (XLSX)

■ AUTHOR INFORMATION

Corresponding Author

Ranjan Srivastava − *Department of Chemical and Biomolecular Engineering, University of Connecticut, Storrs, Connecticut 06269, United States*; orcid.org/0000-0003-4309-605X; Email: rs@uconn.edu

Authors

Joseph Zavorskas − *Department of Chemical and Biomolecular Engineering, University of Connecticut, Storrs, Connecticut 06269, United States*

Harley Edwards − *Department of Chemical, Biochemical, and Environmental Engineering, University of Maryland, Baltimore County, Baltimore, Maryland 21250, United States*

Mark R. Marten − *Department of Chemical, Biochemical, and Environmental Engineering, University of Maryland, Baltimore County, Baltimore, Maryland 21250, United States*

Steven Harris − *Department of Plant Pathology, Entomology, and Microbiology, Iowa State University, Ames, Iowa 50011, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.4c01704

Notes

Code Availability: Code is available on GitHub at https://github.com/SrivLab/SoybeanInverse.

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant no. 2006190 and no. 2006189.

■ REFERENCES

(1) Willis, S. The use of soybean meal and full fat soybean meal by the animal feed industry. In *12th Australian Soybean Conference*; Soy: Bundaberg, Australia, 2003.

(2) Pradhan, A.; Shrestha, D. S.; Van Gerpen, J.; Duffield, J. The energy balance of soybean oil biodiesel production: a review of past studies. *Trans. ASABE* **2008**, *51* (1), 185−194.

(3) Swain, S. N.; Biswal, S. M.; Nanda, P. K.; Nayak, P. L. Biodegradable soy-based plastics: opportunities and challenges. *J. Polym. Environ.* **2004**, *12*, 35−42.

(4) Deng, L. Current progress in the utilization of soy-based emulsifiers in food applications—A Review. *Foods* **2021**, *10* (6), 1354.

(5) Cromwell, G. L. *Soybean Meal—An Exceptional Protein Source*; Soybean Meal InfoCenter, 2012.

(6) List, G. R. Soybean lecithin: Food, industrial uses, and other applications. *Polar lipids* **2015**, 1−33.

(7) Wilkinson, J. M.; Young, R. H. Strategies to reduce reliance on soya bean meal and palm kernel meal in livestock nutrition. *J. Appl. Anim. Nutr.* **2020**, *8* (2), 75−82.

(8) Xu, S.; Wang, Y.; Zhang, B.; Chen, H. Invisibility cloaks from forward design to inverse design. *Sci. China: Inf. Sci.* **2013**, *56*, 1−11.

(9) Wang, J.; Wang, Y.; Chen, Y. Inverse design of materials by machine learning. *Materials* **2022**, *15* (5), 1811.

(10) Ren, S.; Mahendra, A.; Khatib, O.; Deng, Y.; Padilla, W. J.; Malof, J. M. Inverse deep learning methods and benchmarks for artificial electromagnetic material design. *Nanoscale* **2022**, *14* (10), 3958−3969.

(11) Hao, Y.; Mittra, R. *FDTD Modeling of Metamaterials: Theory and Applications*; Artech House, 2008.

(12) Madani, A.; Krause, B.; Greene, E. R.; Subramanian, S.; Mohr, B. P.; Holton, J. M.; Olmos, J. L., Jr.; Xiong, C.; Sun, Z. Z.; Socher, R.; et al. Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.* **2023**, *41* (8), 1099−1106.

(13) Watson, J. L.; Juergens, D.; Bennett, N. R.; Trippe, B. L.; Yim, J.; Eisenach, H. E.; Ahern, W.; Borst, A. J.; Ragotte, R. J.; Milles, L. F.; et al. De novo design of protein structure and function with RFdiffusion. *Nature* **2023**, *620* (7976), 1089−1100.

(14) Anishchenko, I.; Pellock, S. J.; Chidyausiku, T. M.; Ramelot, T. A.; Ovchinnikov, S.; Hao, J.; Bafna, K.; Norn, C.; Kang, A.; Bera, A. K.; et al. De novo protein design by deep network hallucination. *Nature* **2021**, *600* (7889), 547−552.

(15) Chevalier, A.; Silva, D. A.; Rocklin, G. J.; Hicks, D. R.; Vergara, R.; Murapa, P.; Bernard, S. M.; Zhang, L.; Lam, K. H.; Yao, G.; et al. Massively parallel de novo protein design for targeted therapeutics. *Nature* **2017**, *550* (7674), 74−79.

(16) Kim, D. E.; Jensen, D. R.; Feldman, D.; Tischer, D.; Saleem, A.; Chow, C. M.; Li, X.; Carter, L.; Milles, L.; Nguyen, H.; et al. De novo design of small beta barrel proteins. *Proc. Natl. Acad. Sci. U.S.A.* **2023**, *120* (11), No. e2207974120.

(17) Wu, K.; Bai, H.; Chang, Y. T.; Redler, R.; McNally, K. E.; Sheffler, W.; Brunette, T. J.; Hicks, D. R.; Morgan, T. E.; Stevens, T. J.; et al. De novo design of modular peptide-binding proteins by superhelical matching. *Nature* **2023**, *616* (7957), 581−589.

(18) Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G. R.; Wang, J.; Cong, Q.; Kinch, L. N.; Schaeffer, R. D.; et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, *373* (6557), 871−876.

(19) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583−589.

(20) Dekkers, J. C. Commercial application of marker-and gene-assisted selection in livestock: strategies and lessons. *J. Anim. Sci.* **2004**, *82* (suppl_13), E313−E328.

(21) Jannink, J. L.; Lorenz, A. J.; Iwata, H. Genomic selection in plant breeding: from theory to practice. *Briefings Funct. Genomics* **2010**, *9* (2), 166−177.

(22) Yang, L.; Zhao, D.; Meng, Z.; Xu, K.; Yan, J.; Xia, X.; Cao, S.; Tian, Y.; He, Z.; Zhang, Y. QTL mapping for grain yield-related traits in bread wheat via SNP-based selective genotyping. *Theor. Appl. Genet.* **2020**, *133*, 857−872.

(23) Vignal, A.; Milan, D.; SanCristobal, M.; Eggen, A. A review on SNP and other types of molecular markers and their use in animal genetics. *Genet., Sel., Evol.* **2002**, *34* (3), 275−305.

(24) Desta, Z. A.; Ortiz, R. Genomic selection: genome-wide prediction in plant improvement. *Trends Plant Sci.* **2014**, *19* (9), 592−601.

(25) Matei, G.; Woyann, L. G.; Milioli, A. S.; de Bem Oliveira, I.; Zdziarski, A. D.; Zanella, R.; Coelho, A. S. G.; Finatto, T.; Benin, G. Genomic selection in soybean: accuracy and time gain in relation to phenotypic selection. *Mol. Breed.* **2018**, *38*, 117.

(26) Endelman, J. B. Ridge regression and other kernels for genomic selection with R package rrBLUP. *Plant Genome* **2011**, *4* (3), 250−255.

(27) Pérez, P.; de Los Campos, G. Genome-wide regression and prediction with the BGLR statistical package. *Genetics* **2014**, *198* (2), 483−495.

(28) Farooq, M.; van Dijk, A. D.; Nijveen, H.; Mansoor, S.; de Ridder, D. Genomic prediction in plants: opportunities for ensemble machine learning based approaches. *F1000Research* **2022**, *11*, 802.

(29) Tong, H.; Nikoloski, Z. Machine learning approaches for crop improvement: Leveraging phenotypic and genotypic big data. *J. Plant Physiol.* **2021**, *257*, 153354.

(30) Sarkar, R. K.; Rao, A. R.; Meher, P. K.; Nepolean, T.; Mohapatra, T. Evaluation of random forest regression for prediction of breeding value from genomewide SNPs. *J. Genet.* **2015**, *94*, 187−192.

(31) Li, B.; Zhang, N.; Wang, Y. G.; George, A. W.; Reverter, A.; Li, Y. Genomic prediction of breeding values using a subset of SNPs identified by three machine learning methods. *Front. Genet.* **2018**, *9*, 237.

(32) Zhao, W.; Lai, X.; Liu, D.; Zhang, Z.; Ma, P.; Wang, Q.; Zhang, Z.; Pan, Y. Applications of support vector machine in genomic prediction in pig and maize populations. *Front. Genet.* **2020**, *11*, 598318.

(33) Sandhu, K. S.; Lozada, D. N.; Zhang, Z.; Pumphrey, M. O.; Carter, A. H. Deep learning for predicting complex traits in spring wheat breeding program. *Front. Plant Sci.* **2021**, *11*, 613325.

(34) Ma, W.; Qiu, Z.; Song, J.; Li, J.; Cheng, Q.; Zhai, J.; Ma, C. A deep convolutional neural network approach for predicting phenotypes from genotypes. *Planta* **2018**, *248*, 1307−1318.

(35) Azodi, C. B.; Bolger, E.; McCarren, A.; Roantree, M.; de Los Campos, G.; Shiu, S. H. Benchmarking parametric and machine learning models for genomic prediction of complex traits. *G3: Genes, Genomes, Genet.* **2019**, *9* (11), 3691−3702.

(36) Pound, M. P.; Atkinson, J. A.; Townsend, A. J.; Wilson, M. H.; Griffiths, M.; Jackson, A. S.; Bulat, A.; Tzimiropoulos, G.; Wells, D. M.; Murchie, E. H.; et al. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience* **2017**, *6* (10), gix083.

(37) Jiang, Y.; Li, C. Convolutional neural networks for image-based high-throughput plant phenotyping: a review. *Plant Phenomics* **2020**, *2020*, 4152816.

(38) Tadist, K.; Najah, S.; Nikolov, N. S.; Mrabti, F.; Zahi, A. Feature selection methods and genomic big data: a systematic review. *J. Big Data* **2019**, *6* (1), 79.

(39) Liu, Y.; Wang, D.; He, F.; Wang, J.; Joshi, T.; Xu, D. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. *Front. Genet.* **2019**, *10*, 1091.

(40) Rogers, J.; Gunn, S. Identifying feature relevance using a random forest. In *Subspace, Latent Structure and Feature Selection: Statistical and Optimization Perspectives Workshop, SLSFS 2005, Bohinj, Slovenia, February 23−25, 2005*, Revised Selected Papers; Springer: Berlin, 2006; pp 173−184.

(41) Lee, H.; Park, S. Y.; Zhang, Z. J. *An Overview of Genetic Transformation of Soybean*; IntechOpen, 2013.

(42) Berschneider, J. Chances and Limitations of European Soybean Production: Market Potential Analysis. Master's Thesis, Universität Hohenheim, Stuttgart, Germany, 2016.

(43) Yuan, Q.; Gao, X. Multiplex base-and prime-editing with drive-and-process CRISPR arrays. *Nat. Commun.* **2022**, *13* (1), 2771.

(44) Dai, P.; Sun, K.; Yan, X.; Muskens, O. L.; de Groot, C. H.; Zhu, X.; Hu, Y.; Duan, H.; Huang, R. Inverse design of structural color: finding multiple solutions via conditional generative adversarial networks. *Nanophotonics* **2022**, *11* (13), 3057−3069.

(45) Kabir, H.; Wang, Y.; Yu, M.; Zhang, Q. J. Neural network inverse modeling and applications to microwave filter design. In *IEEE Trans. Microwave Theory Tech.*, 2008, pp 867−879..

(46) Whitford, C. M.; Cruz-Morales, P.; Keasling, J. D.; Weber, T. The design-build-test-learn cycle for metabolic engineering of streptomycetes. *Essays Biochem.* **2021**, *65* (2), 261−275.

(47) Gurdo, N.; Volke, D. C.; McCloskey, D.; Nikel, P. I. Automating the design-build-test-learn cycle towards next-generation bacterial cell factories. *New Biotechnol.* **2023**, *74*, 1−15.

(48) Campbell, K.; Xia, J.; Nielsen, J. The impact of systems biology on bioprocessing. *Trends Biotechnol.* **2017**, *35* (12), 1156−1168.

(49) Xavier, A., Beavis, W. D., Specht, J. E., Diers, B., Muir, W. M., Rainey, K. M. SoyNAM: Soybean nested association mapping dataset. *CRAN: Contributed Packages*, 2015

(50) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Duchesnay, E. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(51) Gad, A. F. PyGAD: An intuitive genetic algorithm python library. *arXiv (Computer Science.Neural and Evolutionary Computing)*, June 11, 2021, 2106.06158, ver. 1. https://arxiv.org/abs/2106.06158.

(52) McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv (Statistics.Machine Learning)*, September 18, 2020, 1802.03426, ver. 3. https://arxiv.org/abs/1802.03426.

(53) Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM, 2016; pp 785−794.

(54) Jain, A. K.; Mao, J.; Mohiuddin, K. M. Artificial neural networks: A tutorial. *Computer* **1996**, *29* (3), 31−44.