TriGORank: A Gene Ontology Enriched Learning-to-Rank Framework for Trigenic Fitness Prediction

Sahiti Labhishetty[†‡], Ismini Lourentzou[†§], Michael Jeffrey Volk[*‡], Shekhar Mishra[*‡], Huimin Zhao[*‡], Chengxiang Zhai[†‡]
[*]Department of Chemical and Biomolecular Engineering, [†]Department of Computer Science
[‡]Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA.
{sahitil2, mjvolk3, smishr10, zhao5, czhai}@illinois.edu
[§]Virginia Tech, Blacksburg, VA, USA. ilourentzou@vt.edu

***Abstract*—Machine learning (ML) has been gaining interest in the metabolic engineering community as a means to automate prediction tasks. In this work, we introduce and study the task of using ML to recommend high-fitness triplet mutants as candidates for wet-lab experiments. We first utilize individual fitness and digenic fitness scores as features and train machine learning models that produce a ranked list, from high to low fitness scores, for triplet gene mutants of *S. cerevisiae*. Then, we incorporate prior metabolic knowledge from an existing gene ontology, by designing a novel graph representation and deducing features that can capture gene similarity and gene interactions. Experimental results show that our proposed gene ontology enriched model, termed *TriGORank*, improves both performance and explainability.**

I. Introduction

In experimental fields such as synthetic biology and metabolic engineering, most problems are combinatorial in nature and considering all possible variants can quickly become expensive. As an example, an average protein of 300 amino acids has 7.5×10^{30} potential variants. Engineering large biological systems suffer from this same intractability. In a relatively simple case of looking at all possible gene deletions for the well-studied yeast *S. cerevisiae*, for which there exist approximately $6,000$ genes with just two gene deletions (digenic mutants), there are 18 million possible mutants, while with three gene deletions (trigenic mutants), there are 36 billion possible mutants. Hence, there is an evident need for methods that can suggest promising design variants.

As an emerging solution, machine learning (ML) can be leveraged to recommend the most promising variants for experimentation, which can not only reduce the cost but also accelerate discovery. In this paper, we study how to use machine learning to predict the triple mutant fitness and rank high fitness candidates for experimental prioritization, a novel task that has not yet been studied in the existing work.

To make meaningful progress in engineering biological systems, we need to be able to learn from the experience of prior biological knowledge to find improved variants in high-order mutation space. ML has begun to prove its effectiveness at such protein engineering tasks. Several models have been developed to suggest promising high-order mutants that are unlikely to be engineered by typical wet-lab methods, *e.g.*, directed evolution [1]. However, recommending useful high-order mutations in systems biology and metabolic engineering is still in a nascent stage. In particular, a fundamental need in metabolic engineering is to predict the survival situation of a cell after a set of genes are simultaneously mutated so as to quickly identify the most promising sets with high fitness scores for increasing cell productivity.

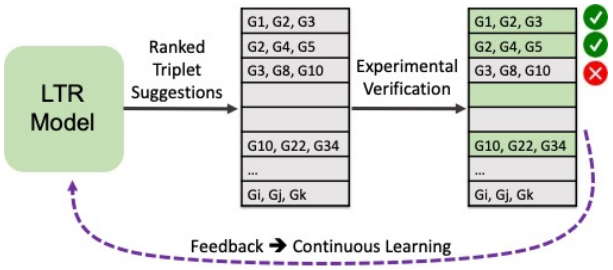

Fig. 1. Overview of the proposed trigenic fitness framework. A Learning-to-Rank (LTR) model provides a list of the top most promising gene triplet recommendations, and wet-lab experimental verification guides the model's for continuous updates through relevance feedback.

A comprehensive set of the digenic mutants and a small subset of trigenic mutants have been gathered in the existing research for *S. cerevisiae*, but it is highly unlikely that a full characterization of the trigenic mutant space will become available anytime in the near future, let alone the high-order mutant space.

We propose an ML-based triple mutant recommendation system (Figure 1) to accelerate the discovery of high fitness trigenic mutants for the organism *S. cerevisiae*. To construct high fitness triple gene knockout mutants (triplets[1]), an ML model can learn to recommend the triplets that are predicted to have high fitness, and the top candidate mutants can be constructed and tested in the wet-lab. Triplet fitness can be

[1]We interchangeably use the terms "triplet", "triplet mutant" and "trigenic mutants" throughout this paper to denote triplets of genes.

978-1-6654-0126-5/21/$31.00 ©2021 IEEE

predicted by utilizing the known digenic and individual fitness scores of the constituted genes, since digenic gene interactions have been shown to overlap with trigenic gene interactions [2]–[4]. For example, as shown in Figure 2, the three genes have three individual fitness scores. In addition, there exist three pairwise interactions, leading to three digenic fitness scores. Therefore, the triplet fitness score can be modelled as a function of the six digenic and individual fitness scores.

As an ML problem, our task has several special challenges. First, it is unclear how we should evaluate an ML algorithm in the application context of recommending promising candidates for experimentation. The regression error measures on fitness scores commonly used do not reflect the utility of results from the perspective of wet-lab experiments. We thus need to seek a new way to set up the ML problem and evaluate the results. Second, there are only six numerical features available for prediction, which is in contrast with many other ML problems that have a large features space. It is thus important to explore any additional features that can be leveraged.

In this work, we address these two challenges by proposing a learning-to-rank framework and by leveraging a knowledge resource (*i.e.*,, Gene Ontology) to construct additional interpretable features for the prediction of trigenic fitness scores. In contrast to prior work, our primary goal of setting up a predictive model is to identify the most promising candidates. To this end, we frame the task as learning-to-rank trigenic candidates in descending fitness order, and train supervised ML models that learn to predict trigenic mutant's fitness based on growth data from single and double mutants. Model performance is evaluated based on the ranking metric NDCG [5] at different cut-offs of the ranked list of trigenic mutants to simulate the perceived utility of a recommended list from the perspective of wet-lab experimentation. With this setup, we evaluated several representative state-of-the-art ML methods and established the first benchmark results of this new task.

Additionally, we design and evaluate novel gene ontology-driven feature representations, including but not limited to gene inter-connectivity, path/edge-based gene similarity, *etc.* The proposed features integrate prior biological knowledge from a curated knowledge graph, providing rich information regarding gene properties and functions, as well as their relations and interactions with each other. Compared to models that rely on fitness features solely, our proposed ontology-driven model, termed ***TriGORank***, produces superior performance. In particular, due to their fairly generic nature, *i.e.*, independence w.r.t. small differences in experimental conditions such as media, temperature, *etc.*, the proposed features improve model robustness. Most importantly, ***TriGORank*** encompasses interpretable gene representations that can be used for explaining the model predictions.

The contributions of our work can be summarized as follows: 1) We introduce the task of trigenic fitness prediction aiming to recommend potential high fitness triplets for prioritizing wet-lab experiments. To this end, we propose a learning-to-rank framework with direct application in wet-lab experiments. To the best of our knowledge, prioritizing candidate triplets is not studied in prior works. 2) We provide a thorough qualitative and quantitative experimental analysis with a variety of ML models, alongside varying testing scenarios. 3) To leverage existing biological knowledge, we propose ***TriGORank***, an interpretable model that incorporates novel ontology-driven interaction and path/edge-based similarity features that capture biological knowledge for individual genes and their relationships to each other. Quantitative and qualitative analysis shows that ***TriGORank*** results in performance improvements over traditional models, with up to 29.1% relative NDCG gain.

II. Related Work

There has been a substantial amount of works applying ML methods to biosystems engineering. For a review, we refer the reader to a recent survey [6]. Traditionally, growth has been studied in the context of gene essentiality, focusing on individual genes, and in the context of epistasis, focusing on the interactions between multiple genes [7], [8]. Mechanistic models that represent the current understanding of the metabolic pathways within cells have been used to address the challenge of mapping from genotype to phenotype. Strain-specific genome-scale models and kinetic models have been used to predict growth, but their failure to make accurate predictions has led to new methods that pair biology-inspired mechanistic models with data-driven ML approaches [9]–[11]. As of now, the consensus around phenotype modeling is that a combination of ML and mechanistic models will likely provide the best phenotype predictions, which is especially pertinent to engineering, but they also have the potential to be the most useful models for generating hypotheses for targeted interrogation of cellular mechanisms [12]. Our work makes steps toward this goal by contributing knowledge about how to optimize an ML model for prioritizing wet-lab experiments.

Given that gene ontologies capture a historical and hierarchical representation of the cell, researchers have utilized them in many forms for a range of prediction tasks. The main novelty of our work in using gene ontologies is that we model and characterize the *interactions* among a set of genes and the emphasis of the interpretability of the derived gene ontology based features. A common way of using gene ontology data is to generate features based on the enrichment of a particular gene. For example, Li et al. [13] built a regression model to predict synthetic lethality between pairs of frequently mutated cancer genes. Along with other minor features, the model primarily relies on the combination of two gene ontology enrichment vectors. Another approach is to use a graph embedding methodology on gene ontology data to learn embedded gene ontology features that can be used for a variety of prediction tasks. This method has been used in predicting missing and spurious protein-protein interactions [14]. Such approaches often prove to have more predictive power compared to traditional ML approaches, but they often lack a meaningful interpretation that is needed by biologists. Others have used gene ontologies as inspiration for the development of their deep learning architecture. DeepGO from Kulmanov

et al. [15] predicts protein functions by using protein sequence features, protein-protein interaction embeddings and then by implementing a novel hierarchical classifier inspired by gene ontologies. Similarly, Rifaiogluj et al. [16] leverage different levels of the gene ontology hierarchy to construct a neural network for predicting protein functions. Different from the previous work, we construct an "interaction subgraph" to represent the interactions of a set of genes and systematically define multiple interpretable features.

Biological interpretability of gene ontology data has been seriously considered in the work Yu et al. [9], in which the authors developed a model that captures the hierarchical structure of the cell by combining deletion strain genotypes and their corresponding gene ontology information to propagate affected ontology terms up the hierarchical structure, thus changing the state of cellular complexes of different biological size. A regression model is then used to predict cell growth based on the disrupted phenotype. Following this work, Ma et al. [17] proposed Dcell, a further iteration of the functionalized ontology where the structure of a deep neural network is embedded into the structure of the ontology hierarchy, making genotype modifications readily explainable since the network has a visual biological interpretation. Both of these methods allow for biological interpretation since states of affected biological complexes can be reinspected. However, the price paid by these models in order to achieve complete interpretability is that they cannot easily leverage the fitness scores of single mutants and digenic mutants and are based on the assumption that the gene ontologies contain complete knowledge about the fitness prediction problem. Unfortunately, gene ontologies are constructed based on literature and represent a very small fraction of the knowledge needed for modeling the complex problem of cell growth. Our proposed methods are complementary with these models with the advantage of flexibly accommodating many more features, including both fitness score features and many heuristically designed gene ontology features.

In the general context of using ML for predictive modeling, our work is different from most existing ones in that, although we use training data with numerical target variables, we frame the problem as a learning-to-rank problem and evaluate the task as a recommendation task with a constrained budget for examination of top-k candidates, instead of using the conventional regression error-based measures. Learning-to-rank [18] has been widely used in information retrieval for combining multiple ranking features. We adapt such a framework to the novel application context of candidate prioritization for experimentation and propose a threshold strategy appropriate for discretizing fitness scores for ranking evaluation.

III. Problem Statement

Given a large collection of gene triplets with their associated trigenic fitness scores $\mathcal{C} = \{x_i, y_i\}_{i=1}^{|\mathcal{C}|}$, where $x_i = (G_i^{(1)}, G_i^{(2)}, G_i^{(3)})$ and $y_i \in \mathbb{R}$ is the trigenic fitness score for triplet i, we can potentially define the trigenic fitness prediction task in multiple ways. The most common problem

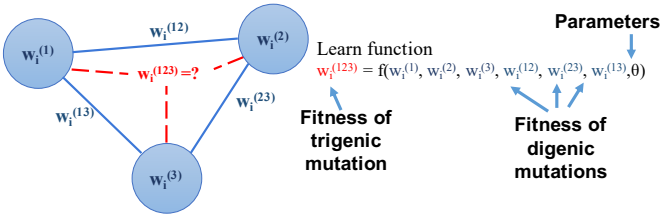

Fig. 2. Given single-mutant $\{w_i^{(1)}, w_i^{(2)}$ and $w_i^{(3)}\}$ and double-mutant $\{w_i^{(12)}, w_i^{(13)}, w_i^{(23)}\}$ fitness scores for triplet i, the goal is to learn a function f_θ that predicts the triplet interaction $w_i^{(123)}$.

formulation used in the existing literature is to treat the task as regression, *e.g.*, minimize the mean squared error between the ground-truth and the predicted fitness scores. However, we argue that such a way of framing the fitness prediction problem is inappropriate for our application context because we care more about the errors made on the high fitness triplets than those on low fitness triplets.

Indeed, our problem should be framed as to identify a subset of k most promising triplet candidates, *i.e.*, top-k mutants with the highest fitness scores. In other words, the goal is to create a scoring function $f_\theta(x_i)$ that ranks triplets and retrieves the top-k ones, *i.e.*,

$$\underset{\mathcal{X} \subset \mathcal{C}, |\mathcal{X}|=k}{\arg\max} \sum_{x_i \in R} f_\theta(\phi(x_i)), \tag{1}$$

where $\phi(x_i)$ is the feature representation for gene triplet i. The number k can be interpreted as the expected number of candidates that can be realistically tested using wet-lab experiments, which is generally determined by the available resources for performing such experiments.

Defined in this way, our problem is fundamentally a learning-to-rank problem, which has often been studied in the context of information retrieval and search engines [18]. However, this learning-to-rank framework cannot be directly applied to the triplet fitness prediction task since we do not have any notion of a "relevant document" that naturally occurs in the retrieval context. To address this problem, we sort all the candidates in descending order of their fitness scores and use the cutoff k to map them into two categories of candidates: target candidates (those above the cutoff k) and non-target candidates (those below the cutoff k), effectively mapping to a retrieval task with the goal of "retrieving" top-k target candidates in the dataset. With this mapping, we can also measure the performance of a predictive model based on the ranking accuracy of the model as reflected in the top-k results. An ideal result is one where all the top-k results are target candidates. Real results from a model would have a mixture of target candidates and non-target candidates in the top-k results. We can then measure the quality of a result list based on to what extent the target candidates have been ranked above the non-target ones by using a standard retrieval measure such as NDCG [5], which puts more weight on errors made in the top positions than those in the bottom positions of the top-k list. Such a measure intuitively reflects the utility of a top-k list from the perspective of avoiding experimenting with non-target candidates.

The formulated problem can be solved with supervised learning. However, the challenge lies in the limited amount of information available for each triplet. From available wet-lab experimental data, open-sourced from prior work [2], [3], there exist single-mutant and double-mutant (digenic) fitness scores for each triplet, as shown in Figure 2. Let $w_i^{(1)}, w_i^{(2)}$ and $w_i^{(3)} \in \mathbb{R}$ be the single-mutant fitness scores, and $w_i^{(12)}, w_i^{(13)}, w_i^{(23)} \in \mathbb{R}$ be the digenic fitness scores, respectively. Then, we wish to learn a function $f_\theta : \mathbb{R}^6 \rightarrow \mathbb{R}$ with input $\phi(x_i) = \{w_i^{(1)}, w_i^{(2)}, w_i^{(3)}, w_i^{(12)}, w_i^{(13)}, w_i^{(23)}\}$ such that $f_\theta(\phi(x_i)) = y_i$. Note that this feature space is extremely constrained. In the next section, we will discuss how to expand the feature set by proposing a knowledge-based triplet fitness prediction method ***TriGORank*** that leverages Gene Ontology to expand the feature space with additional interpretable features capturing gene properties and their interaction characteristics.

IV. *TriGORank*: Knowledge-driven triplet fitness prediction

As mentioned earlier, a major challenge in our problem setup is that there are only 6 fitness score features available, limiting the prediction accuracy. We address this challenge by proposing to expand the feature space with additional features constructed based on relevant knowledge resources, particularly the Gene Ontology (GO). We call our new method ***TriGORank***.

Gene Ontologies (GO) involve tree-structure knowledge sources manually curated by domain experts, and contain information about multiple biological terms and relations between them. Each biological term, referred to as *GO* term, is associated with a *name* and a *description*. Moreover, each GO term is related or connected to other GO terms through *relations* such as *is-a*, *part-of*, *regulates*. A single gene is associated with multiple GO terms (given by gene annotations), which are in turn related to other GO terms through different relations [19], [20]. In other words, the Gene Ontology resembles a hierarchical tree $\mathcal{T}(\mathcal{V}, \mathcal{E})$, where each GO term is a node $v_i \in \mathcal{V}$ and there exist an edge $e(v_i, v_j) \in \mathcal{E}$ connecting two nodes through a relation.

The general idea of ***TriGORank*** is motivated by the attempt to leverage more knowledge in predictive modeling, with two major benefits: 1) The knowledge about possible interactions between genes can directly contribute additional useful features for prediction, thus addressing the problem of limited features and enabling more discrimination. 2) The features constructed using knowledge resources are less prone to overfitting and more interpretable, which means that predictive modeling would require less training data and can also reveal correlated meaningful features with high or low fitness scores as interesting new hypotheses potentially useful for causal modeling of cell survival.

More specifically, we can map all the three genes in a triplet to the nodes of GO and examine how they can be connected via the GO knowledge graph. For example, we can construct a Minimum Interaction Subgraph (MIS) that can link the three genes via relations in GO. Various features characterizing how those genes are connected in such MIS can then be constructed to represent their interactions in the knowledge space. Just as the fitness scores of pairwise combinations of the three genes reveal their actual interactions in a cell environment, we may also view the features constructed using GO as representing their potential "theoretical" interactions in the knowledge space, thus providing an interpretable complementary triplet representation. Note that the general idea of supplementing the empirical observations about gene interactions (*i.e.*, fitness scores in our case) with meaningful features characterizing their hypothesized ("theoretical") interactions in the knowledge space can be applied to any predictive modeling problem involving a set of genes. As will be shown later, the proposed ***TriGORank*** improves over baseline models using only empirical score features by up to 29.1%. Below we describe our proposed ontology-driven features.

Intersection features: The three genes $(G_i^{(1)}, G_i^{(2)}, G_i^{(3)})$ can be connected to same GO terms through the same relation. Such common connections are considered as intersection features. More formally, let $G^{(m)}$ gene be associated with a set of GO terms $GO^{(m)} = \left\{GO_1^{(m)}, GO_2^{(m)}, GO_3^{(m)}, \ldots, GO_{N_m}^{(m)}\right\}$, where N_m is the number of GO terms. The notation N_m indicates that the set cardinality of $GO^{(m)}$ can vary. Let $R\left(GO_j^{(m)}\right)$ be the set of all GO terms associated with $GO_j^{(m)}$ through relation R. Note that R can be anyone of the relations in the gene ontology for example *is-a*, *part-of* or *regulates*. All terms associated with gene $G^{(m)}$ through relation R can be obtained as

$$R\left(G^{(m)}\right) = \bigcup_{j=1}^{N_m} R\left(GO_j^{(m)}\right). \tag{2}$$

Genes connected to same GO terms in the ontology graph have a common connection which can be represented by the following similarity set among M genes via

$$S_R\left(G^{(1)}, \ldots, G^{(m)}, \ldots, G^{(M)}\right) = \bigcap_{m=1}^{M} R\left(G^{(m)}\right) = \bigcap_{m=1}^{M} \bigcup_{j=1}^{N_m} R\left(GO_j^{(m)}\right). \tag{3}$$

The cardinality of S_R, $|S_R|$, is a similarity measure among the considered M genes under relation R. For the three genes in the triplet i, we can compute a triplet similarity measure, and pairwise similarities between any two of the genes, as the cardinalities of $S_R\left(G_i^{(1)}, G_i^{(2)}, G_i^{(3)}\right)$, $S_R\left(G_i^{(1)}, G_i^{(2)}\right)$, $S_R\left(G_i^{(1)}, G_i^{(3)}\right)$, $S_R\left(G_i^{(2)}, G_i^{(3)}\right)$.

Path/Edge-based Similarity: Gene ontology relations are typically hierarchical, for example C is *part-of* B, B is *part-of* A. The paths connecting two genes indicate how similar the genes are, and the node depth influences the similarity between the genes. There are a few node-based, path-based semantic similarity measures [21], in this work we adopted an edge-based semantic similarity measure, Wu-Palmers similarity

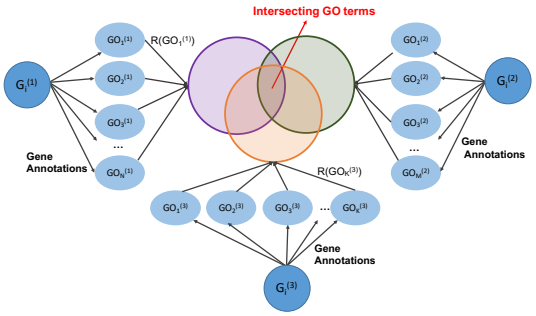

Fig. 3. Incorporating biological knowledge through ontology-driven interaction features.

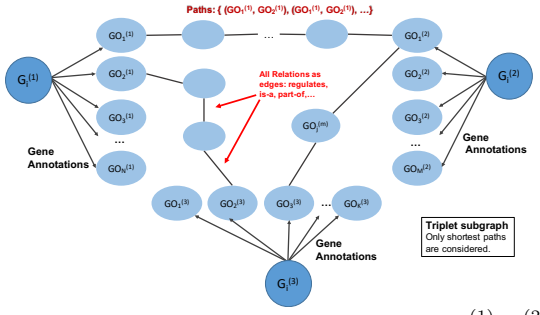

Fig. 4. Triplet subgraph $\mathcal{T}_s$ construction for gene triplet $(G_i^{(1)}, G_i^{(2)}, G_i^{(3)})$

[22]. More specifically, the similarity between two GO terms $GO_j^{(k)}$ and $GO_r^{(l)}$ is given by

$$s\left(GO_j^{(k)}, GO_r^{(l)}\right) = \frac{2 \times d\left(LCS\left(GO_j^{(k)}, GO_r^{(l)}\right)\right)}{d\left(GO_j^{(k)}\right) + d\left(GO_r^{(l)}\right)}, \quad (4)$$

where $LCS\left(GO_j^{(k)}, GO_r^{(l)}\right)$ refers to the closest common ancestor of the two genes, otherwise termed as Least Common Subsumer (LCS) [23], and $d\left(GO_j^{(k)}\right)$ is the node depth for GO term $GO_j^{(k)}$. Since each gene is associated with multiple GO terms, the final similarity between two genes is the average similarity between all associated GO terms, *i.e.*,

$$\tilde{s}\left(G^{(k)}, G^{(l)}\right) = \frac{\sum_{j=1}^{N_k} \sum_{r=1}^{N_l} s\left(GO_j^{(k)}, GO_r^{(l)}\right)}{N_k \times N_l} \quad (5)$$

Eq.5 can be used to compute pairwise similarity between two genes. Thus, for the triplet $(G_i^{(1)}, G_i^{(2)}, G_i^{(3)})$, we compute three pairwise similarities $\tilde{s}\left(G_i^{(1)}, G_i^{(2)}\right)$, $\tilde{s}\left(G_i^{(1)}, G_i^{(3)}\right)$, $\tilde{s}\left(G_i^{(2)}, G_i^{(3)}\right)$. Finally, to compute similarity between all three genes, we compute the similarity between one gene, *e.g.*, G_3, and the least common subsumers for remaining two genes in the triplet, *e.g.*, $LCS\left(G_i^{(1)}, G_i^{(2)}\right)$. The process is repeated for all combinations of the three genes. The average similarity between all such combinations is treated as the triplet similarity, *i.e.*,

$$\bar{s}\left(G_i^{(1)}, G_i^{(2)}, G_i^{(3)}\right) = \frac{1}{3} \times \left[\tilde{s}\left(LCS\left(G_i^{(1)}, G_i^{(2)}\right), G_i^{(3)}\right) + \tilde{s}\left(LCS\left(G_i^{(1)}, G_i^{(3)}\right), G_i^{(2)}\right) + \tilde{s}\left(LCS\left(G_i^{(2)}, G_i^{(3)}\right), G_i^{(1)}\right)\right]. \quad (6)$$

The triplet similarity $\bar{s}\left(G_i^{(1)}, G_i^{(2)}, G_i^{(3)}\right)$ and the pairwise similarities $\tilde{s}\left(G_i^{(1)}, G_i^{(2)}\right)$, $\tilde{s}\left(G_i^{(1)}, G_i^{(3)}\right)$, $\tilde{s}\left(G_i^{(2)}, G_i^{(3)}\right)$ constitute the set of path/edge-based topological semantic similarity features integrated to ***TriGORank***.

Triplet subgraph: Finally, a gene ontology can be represented as an undirected multi-graph $\mathcal{T}(\mathcal{V}, \mathcal{E})$, with GO terms as nodes $v_i \in \mathcal{V}$ and all relations between nodes as edges $e(v_i, v_j) \in \mathcal{E}$. Each triplet can be represented as graph, termed triplet subgraph $\mathcal{T}_s$, by utilizing the connections between GO terms. We construct a triplet subgraph as follows. The node set includes the three genes and the GO terms associated with them, *i.e.*,

$$V_s = \left\{G_i^{(1)}, G_i^{(2)}, G_i^{(3)}, \bigcup_{j=1}^{N_1} GO_j^{(1)}, \bigcup_{k=1}^{N_2} GO_k^{(2)}, \bigcup_{l=1}^{N_3} GO_l^{(3)}\right\}.$$

The edge set $\mathcal{E}_s$ consists of edges of the shortest paths connecting from GO terms from one gene to another, *e.g.*, $e\left(GO_j^{(1)}, GO_k^{(2)}\right)$, $e\left(GO_k^{(2)}, GO_l^{(3)}\right)$, *etc.* Note that the edge set is undirected and can contain multiple parallel edges between two nodes, hence $\mathcal{T}_s$ is an undirected multi-graph. The triplet subgraph construction is illustrated in Figure 4. The shortest path connecting two genes indicates their "closeness" in the gene ontology. Multiple features can be constructed from $\mathcal{T}_s$ that can be used to predict triplet fitness scores. In this work, we construct features based on frequencies of GO terms in the shortest paths present in the triplet subgraph $\mathcal{T}_s$. We hypothesize that presence or absence of certain GO terms in $\mathcal{T}_s$ may correlate with high or low fitness scores. Given triplet i, the probability of a GO term appearing in a triplet path $p \in P$ is

$$P(GO_j^{(m)}|i) = \sum_{p \in P} \frac{\mathbb{1}\{GO_j^{(m)} \in p\}}{|P|} \quad (7)$$

where P is the set of all triplet paths, connecting GO terms of one gene to another. We perform feature selection to select top-K GO term features with the highest positive or negative correlation w.r.t. triplet fitness score values, *i.e.*, the high frequency of a GO term can indicate either high or low triplet fitness scores.

V. Experiments

Dataset: The triple interaction fitness data is obtained from prior work [4], which has released a dataset that contains the name of the three genes and their corresponding triplet fitness scores. To acquire individual and digenic fitness scores, we utilize an additional dataset [2], [3]. Note that both datasets are merged so that for each triplet, individual fitness scores and digenic fitness scores are retrieved. An example instance in this dataset is as follows, "*Triplet: (G_1:YAL054C, G_2:YAR002W, G_3:YML107C)*, with w_1:0.9875 w_2:0.9772, w_3:1.0228, $w_{1,2}$:0.9289 $w_{1,3}$:0.9401 $w_{2,3}$:0.6802, $w_{1,2,3}$:0.5284", where $\{w_1, w_2, w_3\}$, $\{w_{1,2}, w_{1,3}, w_{2,3}\}$ and $\{w_{1,2,3}\}$ are individual, digenic and triplet fitness score(s), respectively. Due to variations in experimental conditions in triplet construction, some instances have more than one triplet fitness score associated with them. As we are unable to capture experimental conditions from external laboratories, we ignore such conflicting triplets with inconsistent trigenic or digenic or individual fitness scores. The final dataset obtained contains

$8,114$ triplet instances. We describe data set splits for training, validation and testing in Subsection V-C.

A. Machine Learning Models

The set of machine learning methods evaluated, consists of a variety of supervised and learning-to-rank machine learning models, in particular:

LR: Linear regression models the relationship between features and outputs with a linear equation [24].

Ridge: Ridge regression is an extension of linear regression models, in which an additional L2 regularizer (sum of squared coefficients) is incorporated to reduce variance and decrease the size of the coefficients [25].

RF: Random Forests construct an ensemble of decision trees and output the final prediction based on majority voting [26]. We set the maximum number of leaf nodes to 350.

SGD: We also compare with a linear model that minimizes a regularized Huber loss with Stochastic Gradient Descent. This baseline down weights the importance of correctly predicting values for outliers [27].

SVR: We compare with Support Vector Regression, a max-margin regression method with a linear kernel function [28].

MLP: We also evaluate a two-layer dense neural network model with ReLU activation functions [29]. The number of neurons per layer is set to 64 and 32, for the first and second linear layers, respectively.

PolyBagging: We test how bagging and traditional feature expansion would affect the results. To this end, we evaluate an ensemble-based linear regression model with degree-3 polynomial features.

LambdaMART: Finally, we compare with a pairwise learning-to-rank model based on an ensemble of gradient boosted regression trees, optimized with gradient descent and an NDCG-weighted cost function [30].

B. Evaluation Metrics

We use NDCG (Normalized Discounted Cumulative Gain) to compute the ranking quality of the result. NDCG is often used in information retrieval to measure the quality of a ranked list of items, where each item is associated with a relevance value. A higher relevance value indicates that the item should be ranked high. The NDCG metric penalizes a ranked list if it contains high relevance items that are ranked lower, thus incorporating both precision and rank in the scoring.

Given a set of triplets, and based on triplet fitness scores, each triplet t_i is assigned a relevance value $rel_i \in (0, l)$. For example, 0 indicates low fitness and 1 indicates high fitness scores. In our setting, the top-k highest fitness triplets are assigned a relevance value of 1, and the rest are subsequently assigned a relevance value of 0. In a practical scenario, the top-k candidates are recommended for wet-lab experiments and a subject matter expert (SME) will have to adjudicate the results. For this reason, k should be rather small, *e.g.*, top-10 or top-30 triplets, as larger values would require substantial manual effort, *i.e.*, it would not be possible to manually evaluate or verify experimentally larger batches.

An ML model produces a ranking for set of triplets $\{t_1, t_2, ...t_n\}$, ordering them based on predicted relevance. Then, NDCG@k can be used to evaluate the top-k triplets in the produced ranked list as follows

$$NDCG@k(\{t_1, t_2, ...t_n\}) = \frac{\sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}}{IDCG} \quad (8)$$

where rel_i is the relevance value of triplet t_i, $IDCG$ is the ideal discounted cummulative gain of the ground-truth ranked list, *i.e.*, the ranked list produced by ground-truth fitness score data. Note that NDCG is a normalized measure with range $NDCG@k \in (0, 1)$.

C. Experiment setup

In our experiments, we set k = $10, 30, 100, 200$ as suggested by our SME. For example if $k = 200$, we consider top 200 as "relevant" (relevance value 1), which are our target candidates, and remaining as irrelevant (relevance value 0), which are non-target candidates. To maintain an equal distribution of target and non-target triplets in the training and testing data, we perform a stratified split $80/20$ resulting in $6,491$ and $1,620$ training and testing instances respectively. For $k = 200$, the train data contain 160 instances marked as target candidates (top-200) and test data contains 40 target candidates. In the following we present results with $k = 200$ only to save space, we have observed similar pattern with k = $10, 30, 100$ as well. For evaluating, $NDCG@\{10, 20, 30, 40\}$ is computed where the top-40 are the target triplets. For ***TriGORank***, we choose the top-10 GO terms from the Triplet Subgraph $\mathcal{T}_s$.

We perform random hyper-parameter tuning and cross-validation to search for the best hyper-parameter combination for each model. To make conclusions regarding gene ontology features in ***TriGORank***, results are averaged over a 5-fold cross-validation. To compare the variance and robustness of the results, we report $NDCG@k$ mean and standard deviation. Finally, except otherwise noted in Section V-A, repression models are optimized on with a mean squared error loss, and LambdaMART is based on NDCG@k loss

D. Experimental Results

We aim to address the following research questions: **RQ1**) What is the best performing ML model for identifying top-k high fitness triplet candidates, based on NDCG? **RQ2**) Does our proposed ***TriGORank***, that leverages knowledge from gene ontology data, improve the performance of an ML model? **RQ3**) How important are the newly incorporated gene ontology features of ***TriGORank***? **RQ4**) Does adding more top-scored GO term based features from the triplet subgraph $\mathcal{T}_s$ improve model performance?

Our first set of experiments target *RQ1*. Table I shows the NDCG performance for all models. From Table I, we infer that Random Forest (RF) performs the best overall. LambdaMART shows better performance at $NDCG@10, 20$, but this is a slight variation compared to the performance differences between RF and LambdaMART for $NDCG@30, 40$. We continue the next round of experiments using RF as the underlying model.

In the following results, we show that using ***TriGORank*** further increases model performance (RQ2). We refer to the model trained on individual and digenic fitness scores solely

TABLE I
PERFORMANCE OF TRADITIONAL ML MODELS

Models	NDCG@10	NDCG@20	NDCG@30	NDCG@40
Linear	0.0663	0.0791	0.0608	0.0502
Ridge	0.0663	0.0783	0.06014	0.0497
SVR	0.0636	0.0794	0.1073	0.0887
SGD	0.1389	0.1269	0.1220	0.1358
PolyBagging	0.0948	0.0967	0.0743	0.0614
MLP	0.0	0.0396	0.0534	0.06094
LambdaMART	**0.9337**	**0.8196**	0.6298	0.5553
RF	0.9266	0.8166	**0.7415**	**0.6822**

as **Baseline**. Further, for GO-term triplet subgraph features, we select Top-K GO term features using chi-squared test .We experiment with three variants of ***TriGORank***-RT, ***TriGORank***-T, and ***TriGORank***-R. ***TriGORank***-RT refers to utilizing both intersection and path/edge-based similarity features as well as the Top-K GO term features. ***TriGORank***-T and ***TriGORank***-R denote variations utilizing either solely Top-K GO term features (***TriGORank***-T) or only intersection and path/edge-based similarity (***TriGORank***-R). In Table II, we report average NDCGs over five-folds, along with standard deviation and average relative improvement percentages (performance gain percentages) w.r.t. the **Baseline**. All ***TriGORank*** variants perform better than the **Baseline**, with the exception of ***TriGORank***-T for Precision@40 and NDCG@40. The best variant overall can be considered ***TriGORank***-R, followed by ***TriGORank***-RT. The NDCG@10 is improved by up to 29% w.r.t. the **Baseline** model. The ontology-based features have a larger impact in ranking top triplets correctly, thus a larger increase in performance at NDCG@10,20 and gradually show lesser improvement towards NDCG@40. Since our goal is to identify the highest fitness triplets, the improvement in top positions is practically more beneficial.

Moreover, we test whether adding more top-K GO terms would help improve the performance (RQ4). We experiment with $K = \{10, 30, 50\}$. From Table III, we infer that while the performance largely remains the same, with slight changes (increases or decreases in performances) up until Top-30 GO features, later the performance deteriorates. For Top-50 GO, the additional features could be redundant and cause overfitting, thus harming the NDCG performance. We also analyze Importance value for all ***TriGORank*** features (RQ3). In RF model, Feature importance value is computed based on total impurity reduction for each feature node in the decision tree as per [31]. The higher the importance score is, the more valuable and accurate this feature is for model predictions. Note that absolute values may appear low due to normalization (feature importance values are normalized to sum to 1). Table IV shows important features sorted according to their importance scores. We group gene-conditioned features together and sum their feature importance values. For example, "regulates_12+13+23+123 intersection" denotes the sum of importance values for $\{S_R(G_i^{(1)}, G_i^{(2)}), S_R(G_i^{(1)}, G_i^{(3)}), S_R(G_i^{(2)}, G_i^{(3)}), S_R(G_i^{(1)}, G_i^{(2)}, G_i^{(3)})\}$, where relation R is "regulates". Similarly, "S_12+13+23+123" represents the path/edge-based similarity features. Note that GO terms are not grouped. From Table IV, the fitness scores naturally dominate the importance values, as the component fitness score have direct influence on the triplet fitness score. Among the ***TriGORank*** features, the path/edge-based similarity features have the highest importance, followed by intersection features for relations "is-a", "regulates" and "part-of". The ungrouped GO terms lower importance scores, but they can potentially enable further analysis by taking into account their frequencies in the interaction paths in the triplet subgraph.

TABLE II
COMPARISON OF ***TriGORank*** VARIATIONS. RELATIVE PERFORMANCE GAINS SHOWN IN PARENTHESIS.

Models	NDCG@10	NDCG@20	NDCG@30	NDCG@40	Prec@40
Baseline	0.538±0.21	0.489±0.18	0.4595±0.13	0.457±0.099	0.43±0.1
TriGORank-T	0.581±0.19 (↑13.8)	0.510±0.15 (↑10.3)	0.451±0.13 (↑0.3)	0.457±0.12 (↓-0.4)	0.42±0.1 (↓-2.9)
TriGORank-R	**0.643± 0.11 (↑29.1)**	0.577±0.11 (↑26.0)	**0.536± 0.11 (↑20.2)**	**0.496±0.09 (↑9.6)**	0.44±0.09 (↑0.6)
TriGORank-RT	0.611±0.12 (↑21.6)	**0.587±0.15 (↑24.9)**	0.529±0.11 (↑17.7)	0.486± 0.09 (↑7.5)	**0.44±0.08 (↑2.2)**

TABLE III
PERFORMANCE OF THE TOP-K GO TERM FEATURES, $K = \{10, 30, 50\}$

TriGORank (Top-K GO)	NDCG@10	NDCG@20	NDCG@30	NDCG@40	Prec@40
Top-10 GO	0.7917	**0.8332**	**0.7327**	0.6409	0.575
Top-30 GO	**0.8116**	0.8119	0.7183	**0.6460**	0.575
Top-50 GO	0.7136	0.7788	0.6905	0.6220	0.600

An advantage of the triplet subgraph GO-term features is that each feature directly associates with a GO-term and a GO-term is a meaningful biological concept. For example, GO terms can signal higher or lower survival rates. Therefore, we further analyze and interpret the top GO term features. We count the frequency of a GO term in interaction paths of $\mathcal{T}_s$ of the top 100 high fitness triplets (FQ Top100) and in the bottom 100 triplets with the lowest fitness scores (FQ Bot100). In Table V, we present the top-12 GO terms sorted according to feature importance value.We observed that particular terms are overrepresented in high and low fitness triplet mutants. Many GO terms predominantly occur only in the top 100 or bottom 100 triplets, showing a strong positive or negative correlation with fitness scores. Some of the GO terms are meaningful. For example, from related work [2], [3], it is known that mutants with genes associated with DNA damage stimulus (related to *cellular response to DNA damage*) have reduced fitness (similar trends are observed in Table V, *i.e.*, 0 occurrences in top triplets versus 1317 occurrences in bottom triplets), whereas mutants with loss of protein biogenesis related terms can help improve fitness due to possible genetic suppression. The additional GO terms with high importance might shed light on interesting new hypotheses about the cell growth mechanism that can be further studied in system biology.

VI. CONCLUSION

In this paper, we apply machine learning to prioritize candidates in the downstream wet-lab experiments for trigenic mutants of organism *S. cerevisiae*. We frame the task as a ranking problem of identifying the top-k most promising triplet candidates with the highest fitness score. We experiment with several machine learning models, including regression and ranking, to create initial baseline results for this new task. Our results conclude that Random Forests perform the best, followed by LambdaMART; both outperform other methods by a substantial margin. As the set of features is very limited (individual and pairwise fitness scores), we propose ***TriGORank*** that leverages gene ontology knowledge to construct additional

TABLE IV
FEATURE IMPORTANCE VALUES

Features	Importance
Individual(G1+G2+G3) fitness scores	0.51554
Digenic (G1G2+G1G3+G2G3) fitness scores	0.35535
S_12+13+23+123 features	0.05746
isa_12+13+23+123 intersection	0.01642
regulates_12+13+23+123 intersection	0.01302
partof_12+13+23+123 intersection	0.01252
vacuole	0.01052
cellular response to DNA damage stimulus	0.00504
mitochondrion organization	0.00436
protein folding	0.00383
sporulation	0.00092
invasive growth in response to glucose limitation	0.00036
extracellular region	0.00023
structural molecule activity	0.00014

TABLE V
FREQUENCIES (IN TOP-100 AND BOTTOM-100 TRIPLETS) AND IMPORTANCE VALUES OF THE GO TERM FEATURES

Features	FQ Top100	FQ Bot100	Importance
cytoskeletal protein binding	0	455	0.0128
cytoplasm	730	761	0.0099
vacuole	294	204	0.0079
cellular response to DNA damage	0	1317	0.0057
protein folding	160	295	0.0052
plasma membrane	168	101	0.0048
signaling	83	192	0.0044
mitochondrion organization	119	185	0.0038
sporulation	276	25	0.0036
cellular ion homeostasis	31	208	0.0032
structural molecule activity	16	1100	0.0028
nucleobase	159	44	0.0024

interpretable features, characterizing gene interactions and similarity. We evaluate several ***TriGORank*** variants, all of which outperform the baseline, suggesting the overall effectiveness of ***TriGORank***. Our study shows the great promise of using ML to identify high fitness trigenic mutants, thus potentially accelerating the discovery of promising mutants while also reducing experimental costs. We also show that ***TriGORank*** can reveal a few interpretable GO features that might shed light on possibly interesting new hypotheses about the underlying causal mechanisms of cell growth. The idea of ***TriGORank*** is general and can be applied to many other predictive modeling problems involving gene interactions to improve prediction accuracy and interpretability.

ACKNOWLEDGMENT

This work was funded by U.S. Department of Energy award DE-SC0018420. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Energy.

REFERENCES

[1] A. Madani, B. McCann, N. Naik, N. S. Keskar, N. Anand, R. R. Eguchi, P.-S. Huang, and R. Socher, "Progen: Language modeling for protein generation," *arXiv preprint arXiv:2004.03497*, 2020.

[2] M. Costanzo, A. Baryshnikova, J. Bellay, Y. Kim, E. D. Spear, C. S. Sevier, H. Ding, J. L. Koh, K. Toufighi, S. Mostafavi *et al.*, "The genetic landscape of a cell," *science*, vol. 327, no. 5964, pp. 425–431, 2010.

[3] M. Costanzo, B. VanderSluis, E. N. Koch, A. Baryshnikova, C. Pons, G. Tan, W. Wang, M. Usaj, J. Hanchard, S. D. Lee *et al.*, "A global genetic interaction network maps a wiring diagram of cellular function," *Science*, vol. 353, no. 6306, 2016.

[4] E. Kuzmin, B. VanderSluis, W. Wang, G. Tan, R. Deshpande, Y. Chen, M. Usaj, A. Balint, M. M. Usaj, J. Van Leeuwen *et al.*, "Systematic analysis of complex genetic interactions," *Science*, vol. 360, no. 6386, 2018.

[5] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM TOIS*, vol. 20, no. 4, pp. 422–446, 2002.

[6] M. J. Volk, I. Lourentzou, S. Mishra, L. T. Vo, C. Zhai, and H. Zhao, "Biosystems design by machine learning," *ACS synthetic biology*, vol. 9, no. 7, pp. 1514–1533, 2020.

[7] L. J. Lu, *Gene essentiality: methods and protocols*. Springer, 2015.

[8] D. Segre, A. DeLuna, G. M. Church, and R. Kishony, "Modular epistasis in yeast metabolism," *Nature genetics*, vol. 37, no. 1, pp. 77–83, 2005.

[9] M. K. Yu, M. Kramer, J. Dutkowski, R. Srivas, K. Licon, J. F. Kreisberg, C. T. Ng, N. Krogan, R. Sharan, and T. Ideker, "Translation of genotype to phenotype by a hierarchy of cell subsystems," *Cell systems*, vol. 2, no. 2, pp. 77–88, 2016.

[10] G. Stephanopoulos, A. A. Aristidou, and J. Nielsen, *Metabolic engineering: principles and methodologies*. Elsevier, 1998.

[11] B. Palsson, *Systems biology*. Cambridge university press, 2015.

[12] G. B. Kim, W. J. Kim, H. U. Kim, and S. Y. Lee, "Machine learning applications in systems metabolic engineering," *Current opinion in biotechnology*, vol. 64, pp. 1–9, 2020.

[13] J. Li, L. Lu, Y.-H. Zhang, M. Liu, L. Chen, T. Huang, and Y.-D. Cai, "Identification of synthetic lethality based on a functional network by using machine learning algorithms," *Journal of cellular biochemistry*, vol. 120, no. 1, pp. 405–416, 2019.

[14] X. Zhong and J. C. Rajapakse, "Predicting missing and spurious protein-protein interactions using graph embeddings on go annotation graph," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 1828–1835.

[15] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2018.

[16] A. S. Rifaioglu, T. Doğan, M. J. Martin, R. Cetin-Atalay, and V. Atalay, "DEEPred: automated protein function prediction with multi-task feed-forward deep neural networks," *Scientific reports*, vol. 9, no. 1, pp. 1–16, 2019.

[17] J. Ma, M. K. Yu, S. Fong, K. Ono, E. Sage, B. Demchak, R. Sharan, and T. Ideker, "Using deep learning to model the hierarchical structure and function of a cell," *Nature methods*, vol. 15, no. 4, pp. 290–298, 2018.

[18] T.-Y. Liu, "Learning to Rank for Information Retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, p. 225–331, mar 2009. [Online]. Available: https://doi.org/10.1561/1500000016

[19] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[20] The Gene Ontology Consortium, "The gene ontology resource: enriching a gold mine," *Nucleic Acids Research*, vol. 49, no. D1, pp. D325–D334, 2021.

[21] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang, "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products," *Bioinformatics*, vol. 26, no. 7, pp. 976–978, 2010.

[22] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," in *Proceedings of ACL'94*, 1994, pp. 133–138.

[23] W. W. Cohen, A. Borgida, H. Hirsh *et al.*, "Computing least common subsumers in description logics," in *AAAI*, vol. 1992, 1992, pp. 754–760.

[24] J. F. Kenney and E. Keeping, "Linear regression and correlation," *Mathematics of statistics*, vol. 1, pp. 252–285, 1962.

[25] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[26] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[27] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. of ICML'04*, 2004, p. 116.

[28] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *JMLR*, vol. 2, no. Dec, pp. 125–137, 2001.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[30] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," *Microsoft Research Tech. Report MSR-TR-2010-82*, 2010.

[31] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," *The Wadsworth statistics/probability series*, vol. 358, 1984.