WHEN SHOULD WE *NOT* TRANSFER FUNCTIONAL ANNOTATION BETWEEN SEQUENCE PARALOGS?

MENGFEI CAO and LENORE J. COWEN
Department of Computer Science, Tufts University,
Medford, MA 02155, USA
Email: mengfei.cao@tufts.edu and lenore.cowen@tufts.edu

Current automated computational methods to assign functional labels to unstudied genes often involve transferring annotation from orthologous or paralogous genes, however such genes can evolve divergent functions, making such transfer inappropriate. We consider the problem of determining when it is correct to make such an assignment between paralogs. We construct a benchmark dataset of two types of similar paralogous pairs of genes in the well-studied model organism *S. cerevisiae:* one set of pairs where single deletion mutants have very similar phenotypes (implying similar functions), and another set of pairs where single deletion mutants have very divergent phenotypes (implying different functions). State of the art methods for this problem will determine the evolutionary history of the paralogs with references to multiple related species. Here, we ask a first and simpler question: we explore to what extent any computational method with access only to data from a single species can solve this problem.

We consider divergence data (at both the amino acid and nucleotide levels), and network data (based on the yeast protein-protein interaction network, as captured in BioGRID), and ask if we can extract features from these data that can distinguish between these sets of paralogous gene pairs. We find that the best features come from measures of sequence divergence, however, simple network measures based on degree or centrality or shortest path or diffusion state distance (DSD), or shared neighborhood in the yeast protein-protein interaction (PPI) network also contain some signal. One should, in general, not transfer function if sequence divergence is too high. Further improvements in classification will need to come from more computationally expensive but much more powerful evolutionary methods that incorporate ancestral states and measure evolutionary divergence over multiple species based on evolutionary trees.

Keywords: protein function prediction, paralogs

1. Introduction

When new genes are sequenced and deposited into databases, a variety of manual and automated curation is involved in associating functional annotation to these genes. One of the most common practices is to transfer functions based on some threshold of sequence similarity.[1] However, when this sequence similarity threshold results in automatically transferring functional annotation between all pairs of orthologous and paralogous genes, this is deeply problematic because there are cases when the functions of the genes have diverged.[2]

In this paper, we consider the question of transfer of functional annotation *solely for paralogous genes.* It was widely believed that paralogs were more likely to acquire divergent functions than orthologs (the so-called *ortholog conjecture*),[3,4] but in recent years, this assumption has been the subject of spirited debate.[3–6] The present study requires neither a positive nor negative resolution of the ortholog conjecture, nor does it directly shed light on the conjecture itself, since it focuses only on a practical problem in the field of automatic function prediction artificially restricted to a single species: we ask whether any computational method *with access*

to information based only on the single species in which the paralogs reside, can distinguish the pairs whose functional roles are similar from those where functional roles are diverged.

We construct a benchmark dataset of two types of paralogous pairs of genes in the well-studied model organism *S. cerevisiae:* one set of pairs where single deletion mutants have very similar phenotypes (implying similar function), and another set of pairs where single deletion mutants have very divergent phenotypes (implying different function). We are fortunate in that there exist data in *S. cerevisiae* where the similarity of phenotypes of deletion mutants has been categorized: in particular, the extensive phenotype data from Hillenmeyer et al.[7] who look at the phenotypes of homozygous single gene deletion knockouts under 418 different conditions such as depletion of certain amino acids or nutrients.

The Hillenmeyer et al data[7] allows us to construct a gold-standard benchmark dataset of paralogous yeast gene pairs, some with highly similar and some with highly dissimilar functions, as follows. We consider two different datasets of paralogous gene pairs in *S. cerevisiae*. The first dataset we construct from scratch by taking pairs of yeast genes with high sequence similarity. The second dataset is derived from the study of the putative whole genome duplication event for *S. cerevisiae* by Kellis et al.[8] who identify 450 paralogous gene pairs. For each of the paralog pairs in the two datasets we compute a co-fitness score[7] to represent to what extent the two gene deletion knockouts have similar phenotypes. We choose a subset of these paralogs with very high co-fitness score and a subset of these paralogs with very low co-fitness score. The subset with the high co-fitness score are our *same* or *conserved function* paralogs, and the subset with the low co-fitness score are our *divergent function* paralogs. Note that Hillenmeyer et al.[7] has already shown that when genes are clustered using such a co-fitness score, they find clusters that are consistent with shared Gene Ontology annotations for biological process and molecular functions. Gu et al.[9] also uses fitness effect data to study functional compensation among gene duplicates.

We note that a recent study of Plata and Vitkup[10] also considered the genetic robustness and functional evolution of gene duplicates in yeast, based on the same gene deletion knockout set of Hillenmeyer et al. However, they considered a measure that is different than our co-fitness score over the collection of gene deletion mutants. In particular, under the assumption that paralogs with similar function could mutually compensate for each other whereas paralogs with divergent function could not, they considered the average number of "sensitive" conditions (i.e. conditions where a growth defect was observed with a P value cutoff of 0.01) between paralog pairs. Paralogs with a small average number of conditions where there was a growth defect (also alternatively, with a small average fraction of conditions where there was data, to deal with missing data), they assumed meant that the paralogs were mutually able to compensate for one another in the deletion mutant. We discuss how well this measure correlates with our "similar function" co-fitness score below.

In addition to nucleotide and amino acid sequence similarity, we sought to investigate whether simple features of the PPI network would also help distinguish same function from divergent function paralogs. Mika and Rost[11] showed that PPI interactions were better conserved within species than across species: a sort of anti-ortholog conjecture for interlogs. Thus it is reasonable to think that the interaction partners of a gene will be more similar for genes

with similar functions; the problem is of course complicated by the fact that existing PPI data is both noisy and also extremely incomplete. We consider some simple well-studied parameters of this network, namely degree, shortest path distance, shared neighborhood, as well as our diffusion-based DSD measure,[12,13] which has been shown to be especially robust to noise and missing data,[14] to find out to what extent these are informative features for our problem.

2. Related work

The most related paper to the current one is the previously mentioned work of Plata and Vitkup.[10] In addition, there have been some previous studies that have tried to place paralogous gene pairs into different functional categories based on a variety of information sources, including the recent SIFTER[2] which performed quite well in the past two Critical Assessment of protein Functional Annotation algorithm (CAFA) experiments[15] for automated function prediction. Unlike the present study, SIFTER assumes access to information from ancestral states, not just the species in which the paralogous gene pairs themselves reside, so they are able to leverage the power of evolutionary information. Other work[16–19] has used gene expression levels, the number of shared interacting partners, and shared Gene Ontology annotations, in order to predict or assess which pairs are instances of conserved function, subfunctionalization, and neofunctionalization. In each of these papers, ground truth for the predictions are assessed in different ways. Zeng and Hannenhalli[16] compare tissue specific gene expression levels from an ancestral gene (a single-copy gene from a closely related species) and the duplicated genes, where in the neofunctionalization case, for example, they assume the ancestral gene's expression level should be lower than that of both duplicates. In addition to this being a somewhat controversial assumption, noise in measuring expression levels can impact their conclusions. Nakhleh's group[17] uses the yeast PPI network to study the problem of categorizing different evolutionary fates of duplicate genes, but in their case, instead of using the structure of the PPI network to assist in predicting the categories, they used the network to *define* their ground truth gold standard for the categories. In particular, they define gene pairs as similar and divergent in function based on comparing the number of known interacting partners of the ancestral gene and the duplicated genes, a measure that will be very sensitive to noise and incomplete data even in the relatively well-studied yeast interactome.[20]

Our method of determining ground truth for same and divergent functions is less noise sensitive than either of these two other methods, but it is much more restrictive than the methods of previous studies. First, it presents only two categories of functional similarity and divergence. More importantly, it makes use of extensive phenotype data from single deletion mutants: a dataset available for yeast but unavailable for most other species at this time. Thus these other measures may be the only ones available in other species; conversely, if one accepts that the single deletion phenotype data is the best measure of ground truth when available for this problem, then the subject of this paper, namely, determining which *other* more easily obtainable sequence and network measures best correlate with this standard, might be the most important application to studying computational transfer of functional annotation standards in other organisms of interest.

Finally, the most common way in the field to determine if paralogs share the same function

is simply to look up the curated functional annotations in a database based on a human-created ontology structure such as MIPS[21] and GO[22] to see if they are annotated with the same functional labels. However, we note that in many databases, paralogs with nearly identical or identical sequence are often annotated with the same functional labels, even if that annotation comes from experiments with only one of the paralogs.

3. Materials and Methods

3.1. *Physical interaction network*

We download all the 141,327 physical interactions compiled from 7601 publications by BioGRID, version 3.3.122 (date March 3rd, 2015), each interaction of which is experimentally verified and associated with one of the following experimental evidence codes: "Affinity Capture-Luminescence", "Affinity Capture-MS", "Affinity Capture-RNA", "Affinity Capture-Western", "Biochemical Activity", "Co-crystal Structure", "Co-fractionation", "Co-localization", "Co-purification", "Far Western", "FRET", "PCA", "Protein-peptide", "Protein-RNA", "Proximity Label-MS", "Reconstituted Complex", "Two-hybrid". While collecting these physical interactions, we adopt the scoring scheme from Cao et al.[13] and assign real value confidence scores in $(0, 1)$ as weights to interactions, where the scoring scheme weights interactions as higher confidence when they are verified by experiments from multiple publications, plus low-throughput experiments are deemed more reliable than high-throughput experiments. Since we only consider interactions associated for genes in the list of 5091 verified ORFs (open reading frames) from the *Saccharomyces* Genome Database (download date: April 11th, 2014), we exclude the interactions that are associated with non-verified ORFs. Using the data above, we build an undirected weighted graph where a node is a protein, a weighted edge between two nodes exists if and only if there is a physical interaction between the two nodes and the weight on each edge is calculated as the confidence score. As a result, we obtain a connected simple graph, involving 5043 nodes and 79594 edges, with diameter 5.

3.2. *Duplicated gene pairs*

We collect two sets of duplicated gene pairs in two ways. We construct the first set that we call the "SequenceCover" or "SC" set based on sequence similarity using the following process: We first collect the result from all against all BLAST searches over the 5043 proteins and then build a sequence similarity graph where a node is a protein and an edge between two proteins exists if and only if the sequence identity is at least 80% and the BLAST E-value is below 10^{-5}. We then find the maximal independent edge set using a naive heuristic algorithm from the graph which satisfies two conditions: (1) if an edge (a, b) is chosen, none of a or b's neighbors will be chosen; (2) no more edges can be added to the set without violating (1). Because these conditions together with the sequence similarity threshold settings are so strict, we will generate edges for protein pairs that have very high sequence similarity and any two of them in the set will not share a common node. Note that in order to analyze the gene pairs using their fitness data, we exclude from the edge set the edges that are incident to nodes that do not have fitness data available from Hillenmeyer et al.;[7] as a result, we are restricted to

the 3732 genes out of 5043 genes have fitness data available. However, we may find different maximal independent edge sets if we choose different random seeds.

We randomly pick one of these independent maximal edge sets, which then defines our first duplicated gene pair set. For the second set of duplicated gene pairs, we download all 450 WGD gene pairs from Kellis et al.[8] The Kellis et al[8]'s gene pair set consists of gene pairs that are believed to be paralogs derived from the whole genome wide duplication event, inferred by both sequence mapping and gene locus. This data set has also been widely used by many other groups studying function of paralogous genes.[16–19,23] Again we restrict ourselves to the subset of the WGD gene pair set where both nodes have fitness data from Hillenmeyer et al.[7]

Note that by restricting the SC set to gene pairs with a relatively high degree of sequence similarity, it will miss some pairs of distant paralogs. However, since our focus is not on the behavior of the landscape of all paralogs, but rather on the scenario where one might computationally decide to transfer functional annotation based on sequence similarity, this is a reasonable threshold.

3.3. *Fitness profile*

We download all the 1,982,156 fitness defect log ratio scores (where 188,642 scores are missing) derived from the homozygous gene deletion experiments from Hillenmeyer et al.[7] Each log ratio score indicates the fitness defect for one of the 4769 homozygous gene deletion strains involving 4742 genes under one of the 418 different testing environments such as depletion of amino acids or nutrients. With respect to a strain for one gene, Hillenmeyer et al.[7] defines a fitness profile as the 418-dimensional vector where each entry is the log ratio score corresponding to each testing environment. Using the fitness profile, we then follow Hillenmeyer et al's[7] analysis and calculate a co-fitness score for each pair of genes that captures the phenotype similarity based on the growth defect under different testing environments. As a preprocessing step accounting for the missing entries, we impute the missing value as follows: 1) when only one gene has fitness values missing for a given environment, we use the same fitness value for both. 2) when both genes have their fitness values missing for a given environment, we use the mean value over all strains under that environment for both. We calculate the co-fitness scores between any two genes as the cosine distance between the fitness profile vectors as defined in Hillenmeyer et al.[7] In total, we obtain co-fitness scores for $4769 \times (4769-1)/2 = 11,369,296$ unique pairs.

In order to provide statistical analysis on the co-fitness scores for our targeted duplicated gene pairs, we define a z-score z_{cfs} as a normalized co-fitness score: $z_{cfs} = \frac{c_{ij}-\mu}{\sigma}$, where c_{ij} is the co-fitness score between gene i and gene j, μ is the mean co-fitness score over all pairs and σ is the standard deviation over all co-fitness scores. Therefore our empirical $p-value$ is computed as the probability that we will see a result at the normalized co-fitness score using the $t-distribution$ with $n-1$ degrees of freedom where n is the number of distinct pairs of strains, namely 11,369,296. We report for each of our targeted duplicated gene pairs the co-fitness score and their $p-values$, of which a higher value will indicate that the pair of genes is less likely to share phenotype similarity and thus less likely to carry out the same biological function. Since the problem we are trying to solve is to distinguish between paralogous gene pairs with divergent functions and that with shared functions, we need to have two separate

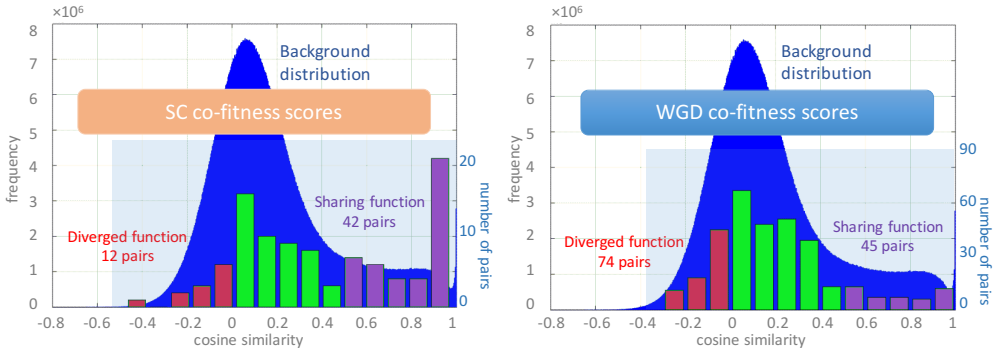

Fig. 1. Defining gene pairs with diverged functions and that with shared functions for the SC and WGD sets using the co-fitness scores.

sets of gene pairs, one set that with high confidence includes gene pairs with divergent functions and the other that with high confidence includes gene pairs with shared functions. We set the thresholds {0.00, 0.50} to define the gene pairs as with divergent functions if the co-fitness score is below 0.00 and the pairs to be with shared functions if the co-fitness score is above 0.50 as shown in Figure 1. Among the SC set where all genes in the 100 gene pairs have the fitness data available, we define 12 gene pairs with diverged functions and 42 gene pairs with shared functions; among the 450 WGD gene pairs where only 337 gene pairs have both genes' fitness data available, we define 74 gene pairs with diverged functions and 45 gene pairs with shared functions. The remaining unclassified gene pairs with co-fitness scores in the middle range, we decline to classify as "shared" or "divergent". Among the 54 classified gene pairs in the SC set, 78% pairs are considered as shared function pairs, while among the 119 classified gene pairs in the WGD set, only 38% pairs are considered as sharing function– this is not surprising as the WGD set includes gene pairs whose duplication event was in the very far past with a lot of evolutionary time to evolve mutations that could affect function.

Finally, we note that among the 119 pairs of paralogs that make up our WGD set, a total of 7 pairs lie on the same yeast chromosome. Among the 54 pairs of paralogs that make up our SC set, also a total of 7 pairs lie on the same yeast chromosome.

3.4. *Sequence similarity*

To measure amino acid similarity, for each of the classified gene pairs, we collect the BLAST bit-score, BLAST alignment length and the percentage identity as the protein sequence similarity measurements.

For nucleotide sequence similarity, for each of the classified gene pairs, we estimate the Ka score, the non-synonymous substitution rate, and the Ks score, synonymous substitution rate, as the nucleotide sequence divergence measurements.[24] More specifically, we compute the pairwise alignment for each gene pair using clustalw2,[25] then we translate the protein

alignment to a codon alignment and estimate Ka and Ks scores using the KaKs_Calculator of Zhang et al.[26] with default parameters. In addition, we also compute the Ka/Ks ratio, which is commonly considered as an indicator of selective pressure acting on a protein-coding gene. We note that for the more distant pairs, these statistics do not give reliable indications of expected evolutionary divergence, however, we can still calculate the values: we just need to assume their correlation with true evolutionary divergence is weaker.

3.5. *PPI network based measures*

For each of the classified gene pairs, we compute a set of network based similarities (or distances): the number of shared interacting neighbors, the normalized shared neighborhood size, the normalized degree difference, the normalized betweenness-centrality score difference, the shortest-path distance and the diffusion state distance (as defined by Cao et al.[13]). In the case of the normalized shared neighborhood size, degree difference and betweenness-centrality score difference, we simply divide by the maximum of the quantities for each paralog to normalize: i.e. to compute the normalized degree difference of paralogs A and B, we simply take $|deg(A) - deg(B)|/\max(deg(A), deg(B))$.

3.6. *Problem formulation*

For each of the measures defined above, we can rank the paralogous gene pairs according to each measure. However, in order to appropriately set a cutoff for each measure, beyond which we predict "conserved function" or "divergent function," we need a training set of labeled examples. First we report the predictive power of each measure described above using a leave-one-out cross validation paradigm. Namely, we learn the optimal cutoffs for classifying pairs as conserved or divergent based on all the data except the held out pair, and then classify the held-out pair according to those thresholds, and report percentage accuracy.

Then we look at the power of some standard machine learning methods when given access to all the features. In particular, we consider: decision trees, naive Bayes, support vector machines (with linear kernel), K-nearest-neighbor (with K=1), logistic regression, random forest, multilayer perceptron, one rule method, and AdaBoost with decision tree, all implemented in WEKA,[27] and see their power on the task of distinguishing the same-function from the divergent-function pairs also in leave-one-out cross validation.

4. Results

4.1. *Classification using each individual similarity measurement*

We assess the predictive power of each individual similarity/divergence measurement using the leave-one-out cross validation paradigm. Specifically, per each measurement, for each paralogous gene pair, we learn a classification threshold based on all the other gene pairs where we will classify pairs above (or below, as appropriate) the threshold as "similar function" and below the threshold (or above) as "diverged function". We then count the percent of pairs that we classify correctly. This list is somewhat deceptive in measuring true performance because of the unbalanced class sizes: but we find the nucleotide sequence-based scores uniformly more

informative than the protein sequence-based scores that we measure. Moreover the Ka and Ks scores remain good classifiers even if the thresholds are trained across the two datasets (see Table 2): for example when the Ks threshold for best classification is trained on WGD and tested on SC, and when the Ks threshold for best classification is trained on SC and tested on WGD, the percent accuracies become 88.89% and 80.67%, respectively. Figure 2 presents the scatter plot of the Ks score versus our co-fitness score.

None of the network measures perform as well as the sequence similarity measures, but the best performing network measures were related to shared neighborhood size.

Table 1. Accuracies for individual measurements			
Performance for leave-one-out-cross-validation (% Accuracy)		SC	WGD
Protein sequence measurements	AA percent identity	74.07%	78.99%
	AA BLAST alignment length	87.04%	69.75%
	AA BLAST bit-score	81.48%	63.03%
	AA ClustalW length	83.33%	76.47%
Sequence measurements	Ka	90.74%	76.47%
	Ks	88.89%	79.83%
	Ka/Ks	79.63%	71.43%
Network measurements	degree-difference (normalized)	70.37%	61.34%
	bc-difference (normalized)	72.22%	63.03%
	shared neighborhood size (SNH)	75.93%	73.11%
	normalized SNH	87.04%	72.27%
	shortest path distance	81.48%	62.18%
	DSD	74.07%	69.75%

Table 2. Cross-training performance accuracy		
	TrainOnWGDTestOnSC	trainOnSCTestOnWGD
Ka/Ks	64.81%	57.98%
Ka	87.04%	79.83%
Ks	88.89%	80.67%

4.2. *Common supervised learning methods for using all measurements*

Motivated by the observation above, we place all the 13 measurements into one feature vector for each gene pair and then try several common supervised learning methods. However, as shown in Table 3, none of the learning algorithms obtain better performance than the best performing individual measurement. Thus it remains an open question how to develop better algorithms that can separate the same-function from divergent function yeast paralogs in our set.

We also wondered whether our method of filling in missing data from the Hillenmeyer et al.[7] experiments contributed to the misclassification rates we saw. Recall that when data

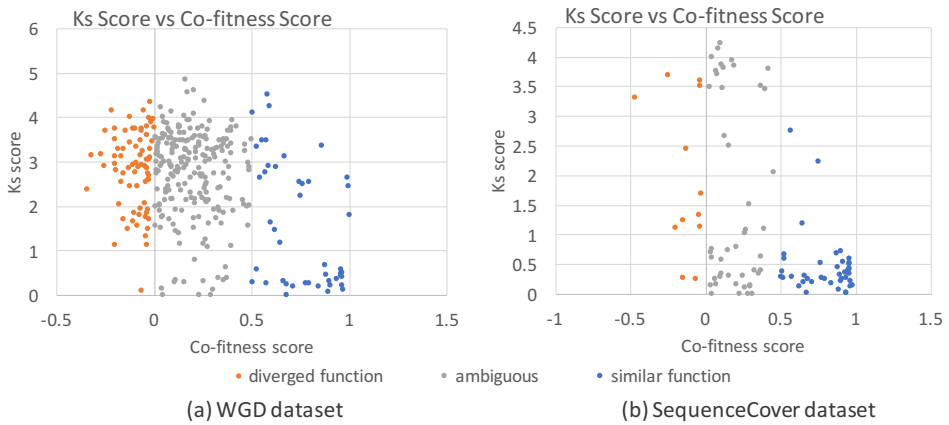

Fig. 2. Scatter plot of Ks score v.s. co-fitness scores for paralog gene pairs

Table 3. Accuracies for different learning algorithms

Performance for leave-one-out-cross-validation (% Accuracy)	SC	WGD
Decision Tree	79.63%	78.15%
Naïve Bayes	90.74%	73.95%
Support Vector Machine (linear kernel)	83.33%	78.15%
k-nearest-neighbor	90.74%	68.07%
Logistic regression	85.19%	75.63%
Random forest	87.04%	78.15%
Multilayer perceptron	87.04%	68.91%
One-Rule	83.33%	75.63%
AdaBoost + Decision Tree	85.19%	70.59%

was missing from a phenotype experiment, we filled in artificial fitness values: if the value was missing in only one of the paralogs, we matched the other paralog; if it was missing for both paralogs, we utilized the mean fitness value over all the deletion experiments for that phenotype for both paralogs. This would make yeast paralogs that are in fact divergent be more likely to have co-fitness scores that would result in our classification as "same function", if at least one had many missing values.

This did, in fact, seem to underlie some of the bad classification results for the WGD dataset in particular. For example, for the SC dataset, among the 48/54 pairs for which the Ks feature results in the correct classification, the average missing ratio is .41, whereas among the 6/54 pairs where the Ks feature results in incorrect classification, the average missing ratio is .37, whereas, for the WGD dataset, among the 95/119 examples where the Ks feature is correct, the average missing ratio is .12, whereas for the 24/119 examples where the Ks feature is wrong, the average missing ratio is .45. Removing all examples with missing data will result in too small a benchmark set; it thus remains an open question to find better ways to deal with missing values in construction of the benchmark datasets.

5. Some example paralog pairs

We looked in more detail at some of the pairs we classified as paralogs with divergent function. Because *S. cerevisiae* is so well-studied, we thought that some of the paralog pairs that we classified as divergent function, might have support from functional annotations in the SGD database, or in the literature. We found the situation quite heterogeneous– for some of the pairs we found support for functional divergence in the literature, for others, there seems to be no annotation indicating that anyone has noted any functional divergence in the two paralogs.

For example, GPP1 and GPP2 are an example of a paralog pair where some functional divergence is known. In particular, GPP1 and GPP2 seem to behave very similarly under aerobic conditions but very differently under anerobic conditions.[28] Another paralog pair where functional divergence is documented is OAF1 and PIP2. Both genes are involved in fatty acid induction of the peroxisomal β-oxidation machinery involving regulation by the oleate response element, and form a heterodimer. But OAF1 binds fatty acids and PIP2 does not.[29] GDH1 and GDH3 are both involved in glutamate biosynthesis, but their regulation indicates that they are utilized under different growth conditions: expression of GDH3 is induced by ethanol and repressed by glucose, whereas GDH1 expression is high in either carbon source.[30] TPK1 and TPK3, together with a third gene, TPK2, are functionally redundant for cell viability, but they have differing protein targets, and also recognize and affect the transcription of different sets of gene targets.[31] In each of these cases, manual inspection implies that functional labels, at least at the top levels of MIPS or GO, would correctly transfer between paralogs, despite these pairs having documented different roles within these broad functional categories.

On the other hand, the majority of the paralog pairs which we mark as "functionally divergent" have no indication in the literature or SGD database that any functional divergence between the paralogs is yet documented. ALK1 and ALK2 is a typical case. Despite a low co-fitness score, this gene pair is currently annotated in a fashion very similar to "same function" pairs: the summary description of ALK2 reads: *Protein kinase; along with its paralog, ALK1, required for proper spindle positioning and nuclear segregation following mitotic arrest, proper organization of cell polarity factors in mitosis, proper localization of formins and polarity factors, and survival in cells that activate spindle assembly checkpoint; phosphorylated in response to DNA damage; ALK2 has a paralog, ALK1, that arose from the whole genome duplication; similar to mammalian haspins.* The description for ALK1 is identical except with the names interchanged.

6. Discussion

The problem of predicting when functional annotation terms should transfer between sequence homologs and paralogs is a difficult but urgent one in the field of automatic prediction of protein function. Here, we have done a very simple study in a single, well-studied species without leveraging the wealth of evolutionary information that is available in sequences. Clearly any reasonable solution will have to leverage this evolutionary information in order to make more accurate predictions.

One clue as to the difficulty of the task might come from the alternative definitions of Plata and Vitkup.[10] We sought to measure how our co-fitness score correlated with the genetic robustness measures of Plata and Vitkup:[10] for each duplicate pair, following their paper, the

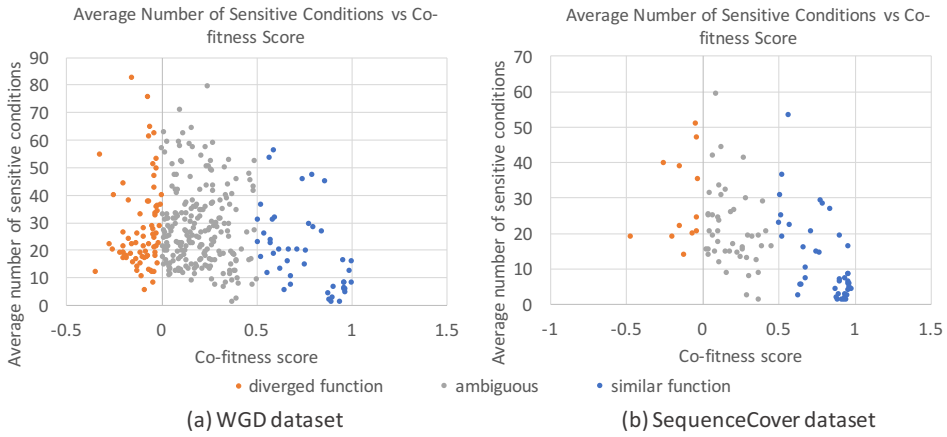

Fig. 3. Scatter plot of average number of sensitive conditions v.s. co-fitness scores for paralog gene pairs.

number of "sensitive" conditions is measured for each deletion mutant, where a "sensitive" condition is defined as a growth defect with $p < 0.01$. We report the number of sensitive conditions, averaged over the two paralogs in the pair, and plot its correlation with our co-fitness score. We find a negative correlation of -0.2046 ($P < 1.49E^{-04}$) (WGD pairs) and a negative correlation of -0.5484 ($P < 2.74E^{-09}$) (SC pairs). A scatter plot appears in Figure 3.

We notice that in both datasets, there are small but distinct set of pairs with both a very low number of average sensitive conditions, and a very high co-fitness score. For these pairs, it is hard to tell if the paralogs are similar function, or if no phenotype is frequently observed because there are third or fourth copy duplicate genes that can buffer both deletion mutants. Thus full understanding of both our co-fitness and their sensitive condition scores may require the consideration of higher order duplicates.

The SC and WGD benchmark datasets are available at bcb.cs.tufts.edu/paralogs

7. Acknowledgements

We thank the entire Tufts BCB group for helpful discussions.

References

1. I. Friedberg, *Briefings in Bioinformatics* **7**, 225 (2006).
2. S. M. Sahraeian, K. R. Luo and S. E. Brenner, *Nucleic Acids Research* , p. gkv461 (2015).
3. N. L. Nehrt, W. T. Clark, P. Radivojac and M. W. Hahn, *PLOS Comput Biol* **7**, p. e1002073 (2011).
4. R. A. Studer and M. Robinson-Rechavi, *Trends in Genetics* **25**, 210 (2009).
5. A. M. Altenhoff, R. A. Studer, M. Robinson-Rechavi and C. Dessimoz, *PLOS Comput Biol* **8**, p. e1002514 (2012).
6. X. Chen and J. Zhang, *PLOS Comput Biol* **8**, p. e1002784 (2012).
7. M. E. Hillenmeyer, E. Fung *et al.*, *Science* **320**, 362 (2008).
8. M. Kellis, B. W. Birren and E. S. Lander, *Nature* **428**, 617 (2004).
9. Z. Gu, L. M. Steinmetz, X. Gu, C. Scharfe, R. W. Davis and W.-H. Li, *Nature* **421**, 63 (2003).
10. G. Plata and D. Vitkup, *Nucleic Acids Research* **42**, 2405 (2014).

11. S. Mika and B. Rost, *PLOS Comput Biol* **2**, p. e79 (2006).
12. M. Cao, H. Zhang, J. Park, N. M. Daniels, M. E. Crovella, L. J. Cowen and B. Hescott, *PLOS One* **8**, p. e76339 (2013).
13. M. Cao, C. M. Pietras, X. Feng, K. J. Doroschak, T. Schaffner, J. Park, H. Zhang, L. J. Cowen and B. Hescott, *Bioinformatics* **30**, i219 (2014).
14. I. Fried, A. Cannistra, C. Casey, A. Piel, M. Crovella and B. Hescott, *ISMB Late Breaking Research* (2015).
15. P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur *et al.*, *Nature Methods* **10**, 221 (2013).
16. J. Zeng and S. Hannenhalli, *BMC Genomics* **14**, p. 1 (2013).
17. Y. Zhu, Z. Lin and L. Nakhleh, *G3: Genes— Genomes— Genetics* **3**, 2049 (2013).
18. A. Baudot, B. Jacq and C. Brun, *Genome Biology* **5**, p. 1 (2004).
19. L. Hakes, J. Pinney, S. Lovell, S. Oliver and D. Robertson, *Genome Biology* **8**, p. 1 (2007).
20. J.-F. Rual, K. Venkatesan *et al.*, *Nature* **437**, 1173 (2005).
21. A. Ruepp, A. Zollner, D. Maier *et al.*, *Nucleic Acids Research* **32**, 5539 (2004).
22. M. Ashburner, C. A. Ball *et al.*, *Nature Genetics* **25**, 25 (2000).
23. C. Roth, S. Rastogi, L. Arvestad, K. Dittmar, S. Light, D. Ekman and D. A. Liberles, *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution* **308**, 58 (2007).
24. Z. Yang and R. Nielsen, *Molecular Biology and Evolution* **17**, 32 (2000).
25. M. A. Larkin, G. Blackshields *et al.*, *Bioinformatics* **23**, 2947 (2007).
26. Z. Zhang, J. Li, X.-Q. Zhao, J. Wang, G. K.-S. Wong and J. Yu, *Genomics, Proteomics & Bioinformatics* **4**, 259 (2006).
27. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, *ACM SIGKDD Explorations Newsletter* **11**, 10 (2009).
28. A.-K. Påhlman, K. Granath, R. Ansell, S. Hohmann and L. Adler, *Journal of Biological Chemistry* **276**, 3555 (2001).
29. A. Gurvitz and H. Rottensteiner, *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* **1763**, 1392 (2006).
30. A. DeLuna, A. Avendaño, L. Riego and A. González, *J. of Biol. Chem.* **276**, 43775 (2001).
31. L. S. Robertson and G. R. Fink, *PNAS* **95**, 13783 (1998).