BENG 213 - Metabolic Promiscuity in *E.coli* - Checkpoint 2

Jason Hyun, Kevin Glass

Statistical Analysis

Domain-based characterization of enzymes extracted from iML1515 using CATH

We used the CATH database to characterize enzymes extracted from iML1515 in the previous checkpoint at a protein domain-level. CATH domains were used instead of the existing GEM-Pro domain annotations for two reasons, 1) original annotations were a mixture of SCOPe and PDP annotations without a unified hierarchy, and 2) CATH annotations were available for nearly all PDB structures. For each enzyme (defined as a set of genes), we took the PDB ID of each gene in the enzyme from GEM-Pro, then looked up all domains associated with each PDB using the CATH database to create domain annotations for each enzyme (**Figure 1**).

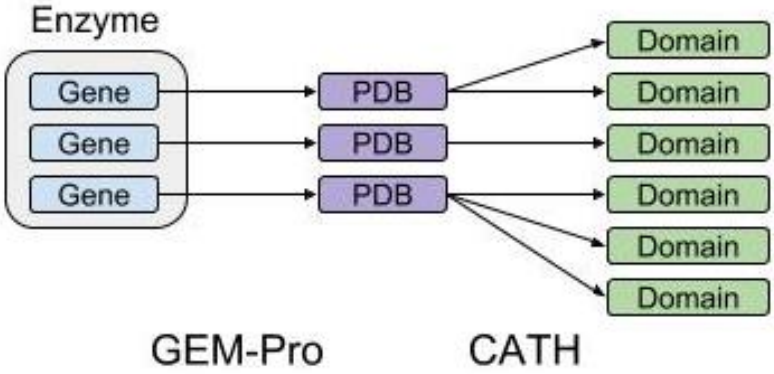

Figure 1: Pipeline for domain annotation of iML1515 enzymes

Of the 974 non-transport enzymes derived from iML1515, 523 enzymes had PDB annotations for some or all its genes. Of those, 504 enzymes had CATH annotations for some or all of their associated PDB structures, encompassing 2687 unique CATH domains. 447 enzymes had complete CATH annotations (i.e. all genes had PDBs, which all had CATH annotations).

Testing associations between CATH domain classifications and enzyme promiscuity

With these annotations, we aimed to test whether there are domains or domain classes in the CATH hierarchy that are associated with enzyme promiscuity. As a preliminary step, we visualized the domain annotations in the hierarchy to a depth of 4 (CATH hierarchy has a depth of 8), at which most leaf nodes had less than 5 enzymes associated (**Figure 2**). On visual inspection we could not see any obvious domain classes of significant size that appeared to be strongly associated with either promiscuity or specificity.

To test this more rigorously, we applied a statistical test to all domain classes of size 10 or greater, of which there were 30. For each domain class, a Fisher's exact test was applied to a 2x2 contingency table constructed based on counts of how many enzymes were in/not in the domain class and how many were/were not promiscuous.

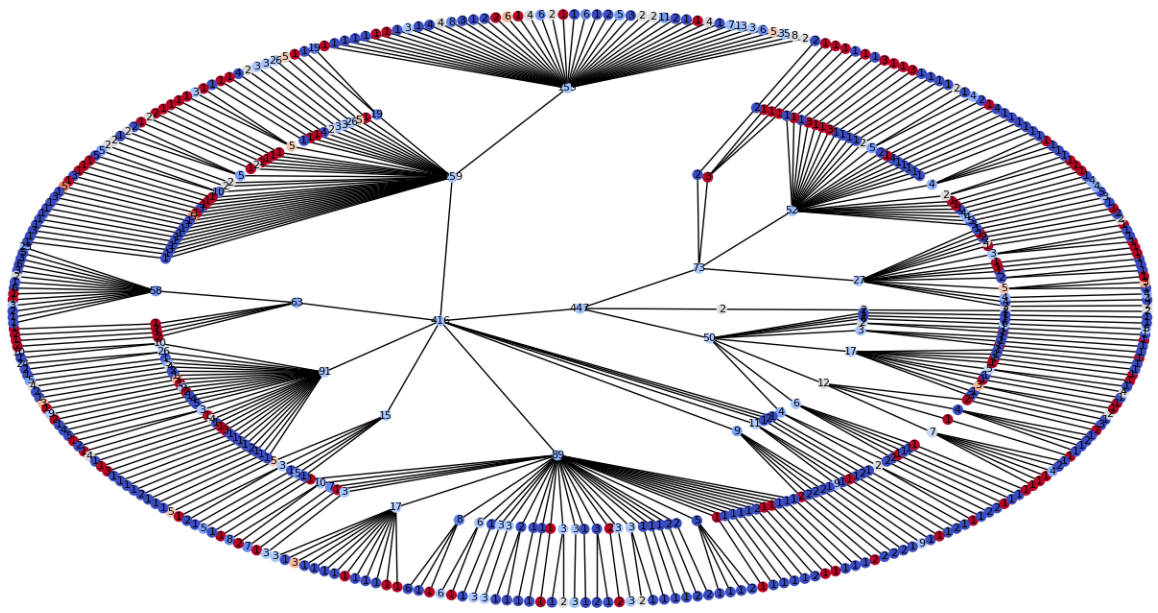

Figure 2: CATH domain hierarchy and enzyme promiscuity in iML1515, depth = 4. Each node corresponds to a CATH domain classification, labeled by the number of enzymes that contain a domain that falls into that classification. Nodes are colored based on the fraction of associated enzymes that are promiscuous (red = more promiscuous, blue = less promiscuous).

Without any correction, the Fisher's Exact tests yielded three CATH domain classes that were significant at a 5% threshold. However, two of those domain classes (3.40.30 and 3.40.30.10) were identical, as there were no enzymes with domains in the 3.40.30 class that were not also in the 3.40.30.10 subclass:

- 3.20.20: Size = 58, p-value = 0.01467
- 3.40.30: Size = 19, p-value = 0.04547
- 3.40.30.10: Size = 19, p-value = 0.04547

Since the tests are clearly not independent, we applied a Bonferroni correction. As there were 30 tests, the corrected significance threshold is 0.05/30 = 0.0017 to assure a FWER below 5%. At this threshold, no domain classes are significant. Alternatively, with the Benjamini-Hochberg correction to control FDR to be at most 5%, the corrected significant thresholds become (0.05/30, 0.05*2/30, 0.05*3/30) = (0.0017, 0.0033, 0.005) for the three domain classes when ranked by p-value, respectively. However, even with this less conservative correction, still no domain classes are significant. The controlled FWER or FDR would have to at least 44% (0.01467 * 30) for even a single domain class to be considered significant under correction.

This is consistent with our initial observation from the visualization that no large domain class appears to be significantly associated with enzyme promiscuity. This suggests that there is no protein domain from the CATH database that is individually informative with regards to enzyme promiscuity (though combinations of domains may be more informative). We examine using domain class combinations for predicting promiscuity in the supervised learning section.

Supervised Learning

Encoding CATH domain information as fixed-length vectors

Since each enzyme is associated with a variable number of CATH domains, we considered using one-hot encoding to convert this information into fixed-length vectors, where each domain corresponds to a position in a vector that is either 0 or 1 based on the presence or absence of that domain in a given enzyme. However, as there are thousands of unique CATH domains in our dataset, we leveraged the CATH domain hierarchy for dimension reduction. Similar to our statistical analysis, considered only CATH domain classes that were associated with some minimum number of enzymes. We then excluded CATH domain classes in which all children classes were also included, since a parent class is fully dependent on its child classes. The number of classes extracted at different class size thresholds is below (**Table 1**).

Table 1: Number of independent domain classes (equal to the dimension of the corresponding domain-based encoding) at varying minimum thresholds for number of associated enzymes.

Minimum Domain Class Size:	5	6	7	8	9
Number of Domain Classes:	67	51	40	37	33

The selected domain classes for a given threshold were used to encode each enzyme, in which each domain class corresponds to a position in a vector that is either 0 or 1 based on the presence or absence of a domain within that domain class in a given enzyme. We assessed our supervised learning methods for all five encoding lengths.

Model selection for the prediction of enzyme promiscuity from CATH domains

We used the domain encodings the predict enzyme promiscuity (a true/false label, based on whether the enzyme can catalyze multiple reactions in iML1515). To select a model for this binary classification problem, we tested four different models: Support Vector Machine, (Multinomial) Naive Bayes, Decision Tree, and k-Nearest Neighbors. For SVM, linear and radial basis function kernels were tested. For k-Nearest Neighbors, k values from 5 to 20 were tested and results for the highest average accuracy for each encoding is reported. Unspecified parameters were set to defaults as implemented in sklearn. Average and standard deviation of prediction accuracy under 5-fold cross validation for these methods are in **Table 2**.

Across all encoding lengths, the order of performance between the models was the same: SVM (rbf) > SVM (linear) > kNN > Decision Tree > Naive Bayes. Accuracies ranged from 60.2% to 68.5% across all models and encodings; since the maximum average accuracy is low, this suggests that the CATH domain data on its own is not particularly predictive for enzyme promiscuity. Curiously, SVM with a radial basis function kernel performed the best in all cases with by far the least variance in prediction accuracy based on cross-validation, but also performed identically across different encoding lengths. At this time, we do not have an explanation for why SVM with rbf kernel performs identically for different encoding lengths.

Table 2: Performance of different supervised learning models at predicting enzyme promiscuity from CATH domains. Learning tested against multiple possible encodings, based on the number of CATH domain classes used for encoding. For k-NN, the optimal k between 5 and 20 is reported for each encoding length, which were 15, 6, 7, 7, and 9, respectively.

Model	Accuracy ± Standard Deviation (5-fold CV)				
SVM (linear)	0.682 ± 0.016	0.667 ± 0.009	0.675 ± 0.023	0.667 ± 0.038	0.673 ± 0.026
SVM (rbf)	0.685 ± 0.002	0.685 ± 0.002	0.685 ± 0.002	0.685 ± 0.002	0.685 ± 0.002
Naive Bayes	0.615 ± 0.018	0.602 ± 0.030	0.617 ± 0.043	0.622 ± 0.040	0.626 ± 0.034
Decision Tree	0.624 ± 0.029	0.637 ± 0.034	0.640 ± 0.042	0.637 ± 0.047	0.633 ± 0.038
k-NN*	0.653 ± 0.019	0.655 ± 0.032	0.644 ± 0.050	0.644 ± 0.057	0.633 ± 0.034
Min Class Size	5	6	7	8	9
Encoding Length	67	51	40	37	33

Performance of SVM at predicting enzyme promiscuity from CATH domains

We proceeded with the SVM model with radial basis function kernel, which consistently demonstrated the highest average accuracy compared to other tested models. To generate precision-recall and ROC curves, we swept through different thresholds for the signed distance from the separating hyperplane at which to label a point as promiscuous or not (in the standard implementation, the threshold is 0; positive signed distances would be assigned promiscuous and negative signed distances would be assigned specific). Positives are enzymes labeled as promiscuous, negatives are enzymes labeled as specific, and PR / ROC curves were generated for different encoding lengths based on the minimum domain class size accepted (**Figure 3**).

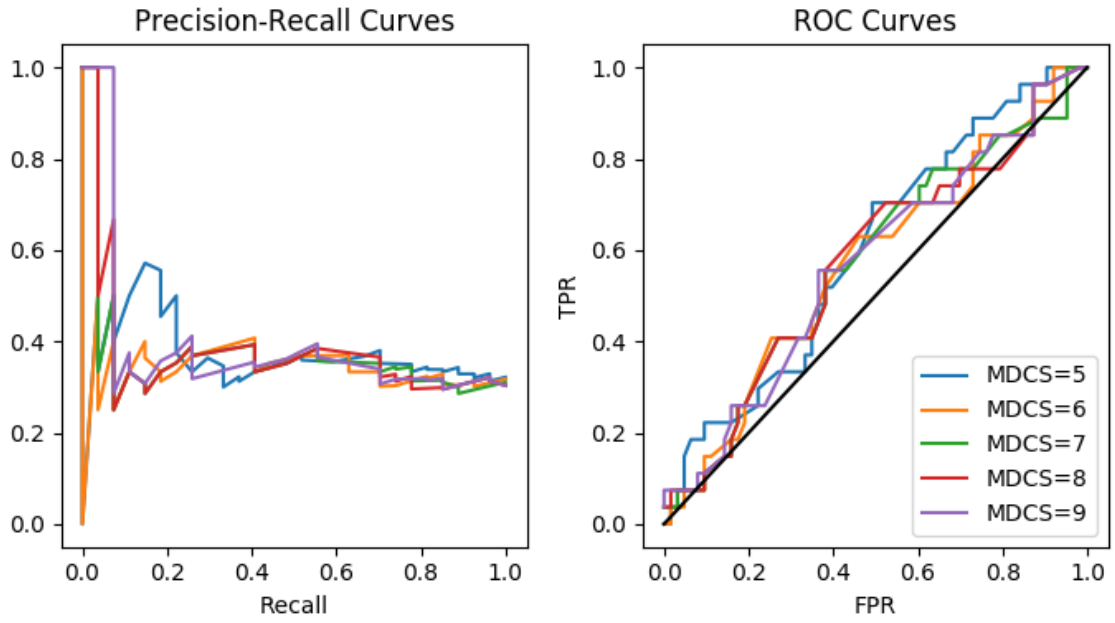

Figure 3: Precision-Recall and ROC curves for SVM model with rbf kernel for predicting enzyme promiscuity from CATH domains. Curves for different encoding lengths based on the minimum domain class size (MDCS) used for encoding, as described previously.

Unfortunately, the plots suggest that the SVM model is only marginally better than random classification. For all 5 domain encodings tested, we find then when recall is low (fraction of promiscuous enzymes detected) the precision can be very high (fraction of promiscuous enzyme calls that are correct), but precision rapidly plummets to baseline once recall reaches 25% (about a third of all enzymes are labeled promiscuous). Similarly, all ROC curves lie very close to the random prediction line. Overall, while there may be a better approach towards encoding domain information into vectors for supervised learning, results so far suggests that CATH domains are not very informative with respect to enzyme promiscuity.

So far, we have examined enzyme domain structure and exact matching of enzyme substrates (Jaccard index of shared substrates approach, from Checkpoint 1). While neither data type appears to be very predictive with respect to enzyme promiscuity on its own, it possible that combining these disparate datasets may be more informative. Finally, we hope to improve our initial enzyme substrate approach by integrating substrate structural information.

Incorporating substrate structures

Towards incorporating substrate structural information, we have obtained MOL files for metabolites from BIGG (about 830 of the original 1167) and processed them using RDKit to obtain molecular fingerprints. The fingerprints were then run through a similarity analysis to obtain a molecular similarity matrix. We are currently in the process of attaching the molecular similarity information to the existing structural information using an enzyme to enzyme similarity matrix. We plan on applying supervised learning methods to make predictions based off of patterns of overall reaction similarity.

To determine enzyme substrate similarity, we plan on using a similarity matrix calculated from the similarity of two metabolites to each other. For each reaction. From that, the maximum values of each similarity score from each column would be added together and averaged. This would produce our enzyme structural similarity score.

Other farther off implementations of structural information are possible incorporation of other graph based methods of identification (Balaban index), as well as trying a combination of different fingerprinting techniques (RDKit also provides other fingerprinting tools other than what we are working in progress to use). Other possible future methods include doing hierarchical clustering on the substrate similarity matrix and then training a machine learning algorithm on promiscuity based off of an enzyme belonging to certain clusters.

Other possible directions are also incorporating substrate structural information into the model itself and trying to also train based upon a combination of the reaction similarity compared to the aggregate chemical similarity.

All code and datasets used are available here: https://github.com/jhyun95/ecoli_promiscuity