# Supplementary Information: Graph algorithms for predicting subcellular localization at the pathway level

Chris S Magnano and Anthony Gitter

## S1. Supplementary Methods

### S1.1. *Spatial Proteomics Case Study*

**Mass spectromety datasets:** To demonstrate a use case of pathway-based localization prediction, and validate its performance, we performed localization prediction on data from a study that measured proteomic quantification of primary fibroblasts during human cytomegalovirus (HCMV) infection. Two mass spectromety quantification methods were used at five timepoints: 24, 48, 72, 96, and 120 hours post infection (hpi). The first of these datasets provides label-free protein quantification at each timepoint. The other was quantified using isobaric labeling via tandem mass tags (TMTs). These two datasets will be referred to as the label-free and the TMT datasets, respectively.

Multi-organelle profiling was performed on the TMT dataset via gradient centrifugation to fractionate organelles. This process partially separates organelles into a set of subcellular fractions. While each of these fractions is not purely a single organelle, each organelle contains a unique signature in its quantification across the subcellular fractions. In the original analysis, clustering analysis was then used to group proteins with similar fraction profiles. Finally, each cluster was labeled as a certain organelle via a set of marker proteins, proteins known with high-confidence to localize to a particular organelle.

In the HCMV protein quantification, a set of marker proteins was curated from UniProt subcellular location annotations with experimental evidence; proteins that were annotated with multiple localizations were excluded from the marker set. Proteins that were not confidently assigned to a particular organelle were left as unlabeled in the original study's localization labeling. From this, there were 2,730 proteins in total, of which 1,229 had localization labels at 24hpi and 1,348 had labels at 120hpi. 574 of these proteins were marker proteins.

**Experimental scenarios:** We explored three different data availability scenarios for using the best performing model from the pathway database prediction task, GAT, to predict localizations in an experimental setting. For all three scenarios, the best performing hyperparameter combination from the pathway database prediction task was used with no further hyperparameter tuning. The three scenarios are as follows:

(1) First, we trained a model using data from the PathBank database as described in Section 2.1. Here we took the best performing pre-trained model with no further modifications and used it to predict localizations at the 120hpi timepoint.
(2) Second, we trained a model using a separate dataset that measured protein localization using a similar method on a different cell type and under a different biological condition, HeLa cells undergoing EGF stimulation.
(3) Third, we trained a model on the same HCMV infection experiment at the 24hpi timepoint.

This third scenario is impractical, as it would require a dataset to already exist for an identical cell type and condition. However, it gives a useful benchmark for predictive performance versus the practical second scenario.

**Pathway reconstruction:** Pathway reconstruction was performed using Omics Integrator 2[d], which is similar to Omics Integrator but uses a different optimization algorithm[e]. Omics Integrator 2 performs pathway reconstruction via the prize-collecting Steiner forest problem. Omics Integrator 2 was run using the Signaling Pathway Reconstruction Analysis Streamliner (SPRAS)[f]. Protein abundance fold-change was used as prizes for both fibroblast HCMV infection data and HeLa EGF stimulation data. The background protein interaction network for pathway reconstruction was a previously published network with 161,901 weighted edges based on merged interactions from the iRefIndex database v13[g] and kinase-substrate interactions from PhosphoSitePlus[h]. The Omics Integrator 2 hyperparameter $\omega$ was tested between 1 and 10, $\beta$ was tested between 1 and 5, and $\mu$ was tested between 0.1 and 1. Each hyperparameter was evaluated at 10 increments across its range. For the 120hpi timepoint, pathway parameter advising was used to select the top 50 pathways from 1,000 candidate hyperparameter combinations. For the 24hpi timepoint and EGF stimulation datasets, all nonempty pathways were used so that these pathway datasets were comparable in size to the pathway database datasets, resulting in training sets of 542 and 503 pathways, respectively. Reconstructed pathways from HCMV infection were large, with an average of 1,279 interactions per pathway at 120hpi. Other pathway sizes were more similar, with EGF stimulation pathways, Reactome pathways, and PathBank pathways having an average of 202, 91, and 229 average interactions, respectively. The top ranked pathway at 120hpi and for EGF stimulation can be viewed in Figures S8 and S9, respectively. All GAT models' hyperparameters were the same as those used in the highest performing GAT model when using Compartments features to predict PathBank labels, as recorded in Table S2.

## S1.2. *Pathway Localization Prediction Models*

### S1.2.1. *Neural Networks*

The maximum value in the output layer was used as the final class label prediction. All neural network models were trained using cross-entropy loss. All neural networks were implemented using PyTorch and the PyTorch Geometric package.

**Graph convolutional network:** The graph convolutional network incorporated a set of message-passing convolutional layers before the final set of fully connected layers. These convolutional layers allow for information to be shared across the topology of the input network. The $l^{th}$ convolutional layer $H^{(l)}$ is updated via the following rule:

[d] `https://github.com/fraenkel-lab/OmicsIntegrator2`
[e] `https://github.com/fraenkel-lab/pcst_fast`
[f] `https://github.com/Reed-CompBio/spras`
[g] `https://irefindex.vib.be/wiki/index.php/iRefIndex`
[h] `https://www.phosphosite.org`

$$H^{(l)} = ReLU(D^{-\frac{1}{2}}\widetilde{A}D^{\frac{1}{2}}H^{(l-1)}W^{(l-1)})$$

Where $\widetilde{A}$ is the adjacency matrix of the input pathway with added self-edges for all nodes, $D$ is a degree matrix normalization factor where $D_{ii} = \sum_j \widetilde{A}_{ij}$, and $W^{(l)}$ is a set of weights for the $l^{th}$ layer. This update rule provides a first-order approximation of spectral graph convolutions and is implemented in the *GCNConv* class in PyTorch Geometric.

**Graph attention network:** Graph attention networks extend graph convolutional networks by allowing each node to choose which of its neighbors to pay attention to. As opposed to taking the average of its neighbors, each node computes a weighted average of its neighbors in graph convolutional layers. Furthermore, many attention networks are multi-headed, where multiple attention weights are computed in parallel for each node. The number of heads to include is an input hyperparameter and often increases accuracy at the cost of increased computational complexity. We used the PyTorch Geometric *GATV2Conv* class for graph layers, which is a more expressive implementation of graph attention that allows for more diversity in attention between nodes.

**Graph isomorphism network:** Graph isomorphism networks take advantage of the similarity between neighbor aggregation in graph neural networks and the Weisfeiler-Lehman (WL) graph isomorphism test. The WL graph isomorphism test is a heuristic algorithm for determining graph isomorphisms. For two graphs, in each iteration of the test every node aggregates its neighbors into a unique hash. These hashes are compared between the two graphs, and if they differ the graphs are known to be non-isomorphic. Iterations of the test are repeated until the user feels confident that the graphs are isomorphic; the algorithm cannot conclusively prove isomorphism.

The neighbor aggregation in each graph layer of a graph isomorphism network is formulated to be at least as powerful as the WL isomorphism test; the $l^{th}$ layer is guaranteed to generate different embeddings of two graphs if those graphs would be found to be non-isomorphic via the WL isomorphism test in $l$ iterations. The representation of each node $n$ in layer $l$ of a graph isomorphism network, $h_n^{(l)}$, is computed as:

$$h_n^{(l)} = MLP^{(l)}\left((1+\epsilon^{(l)})\cdot h_n^{(l-1)} + \sum_{u \in Adj(n)} h_u^{(l-1)}\right)$$

Where $MLP$ is a multi-layer perceptron, $\epsilon$ is a learned parameter, and $Adj(n)$ is the set of nodes adjacent to $n$ in the input pathway. We used the *GINConv* class in PyTorch Geometric for graph isomorphism layers.

### S1.2.2. *Probabilistic Graphical Models*

Given the nature of the label propagation inherent in the pathway level localization prediction task, and that many localization databases contain uncertain or even probabilistic data, probabilistic graphical models are a natural choice. As moving between subcellular locations

costs energy, it is unlikely to happen often within a single pathway. Therefore, we can make the assumption from a modeling perspective that the subcellular location of an interaction is dependent on the subcellular location of neighboring interactions within a pathway.

Probabilistic graphical models represent a set of $N$ random variables $\mathbf{y}$ as nodes and dependencies between them as a set of edges $E$. We created two types of pairwise undirected probabilistic graphical models, which we call NaivePGM and TrainedPGM. In these probabilistic graphical models, the random variables obey a local Markov property, such that each random variable is conditionally independent of all others given its neighbors in the graph.

The NaivePGM is a Markov random field, where the joint probability of all localizations can be factorized as

$$P(\mathbf{y}) = \frac{1}{Z} \prod_{i \in N} \phi_i(y_i) \prod_{i,j \in E} \phi_{ij}(y_i, y_j)$$

Where $Z$ is a normalizing function so that the combination of all possible configurations of $\mathbf{y}$ sum to 1 and $\phi_i(y_i)$ and $\phi_{ij}(y_i, y_j)$ are the unary and pairwise potential functions, respectively. The unary potential function defines the probabilities of a given node having each localization, while the pairwise potential functions define the joint probability of each pair of nodes that share an edge. For finding the task of finding the most likely configuration of $\mathbf{y}$, referred to as decoding, $Z$ can be ignored. In the NaivePGM, the input features are not used to parameterize the potential functions. Instead, the unary potential functions directly map the normalized features to class probabilities, and the joint probability tables directly map the normalized features to joint probability tables. This was chosen because both the ComPPI and Compartments scores represent confidences, with ComPPI scores directly representing probabilities for each localization. However, this use of scores means that the NaivePGM cannot use Uniprot keyword-derived features as they do not represent localization confidence.

The TrainedPGM is a conditional random field where the input features are treated as observations of additional variables. The probability of localization assignments $\mathbf{y}$ are then conditioned over the input features $\mathbf{x}$ as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i \in N} \phi(x_i, y_i) \prod_{i,j \in E} \phi(x_i, y_i, y_j)$$

Here, the unary potential functions are now conditioned on observations of features $x_i$ corresponding to each random variable $y_i$. Edge potentials represent the dependence between each node's state $y_i$ and its neighbor's state $y_j$ given its features $x_i$. Each random variable $y_i$ is conditionally independent of all other variables given its corresponding features $x_i$ and its neighbors' states $y_j$.

The potential functions in conditional random fields are typically log-linear functions of the form $e^{w_i^T \phi_f(x_i, y_i)}$, parameterized via a weight vector $\mathbf{w}$, and $\phi_f$ simply represents features for each node. Additionally, typically the feature weight vectors are shared between nodes or sets of nodes. Thus, the entire model can then be represented as:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \frac{1}{Z} exp\left( \sum_{i \in N} \mathbf{w}_f^T \phi_f(x_i, y_i) + \sum_{i,j \in E} \mathbf{w}_e^T \phi_e(x_i, y_i, y_j) \right)$$

Where $\phi_f(x_i, y_i)$ is the single unary potential function that represents features for each node, and $\phi_e(x_i, y_i, y_j)$ is the single pairwise potential function that represents combinations of states. The weight vector $\mathbf{w_f}$ is a set of weights for each feature to each possible configuration of $y_i$, while the weight vector $\mathbf{w_e}$ is a set of weights for each feature to each combination of configurations for $y_i$ and $y_j$.

When represented with these potentials, the log likelihood of the model parameters $\mathbf{w}$ can be easily represented, and is differentiable, allowing for parameters to be learned by maximum likelihood estimation via gradient-based optimization. Sets of nodes and edges can share the same set of model parameters, referred to as parameter tying. However, parameter learning for a conditional random field of this form did not converge when trained with stochastic gradient descent. This may be due to the underlying label distributions of different pathways being too different from each other.

Instead an alternative model formulation was chosen where potentials are represented by discriminative classifiers, specifically random forests. This type of model is referred to as a discriminative random field. This was chosen over a more traditional log linear parameterization due to better performance on tuning data during model selection.

A traditional construction of a probabilistic graphical models from the nodes and edges of a biological pathway would only provide predictions on the nodes of the graph, while we are interested in localization labels on the edges. To convert the input pathway into an appropriate graphical model, each pathway is converted into a bipartite graph, where a node is added that represents each pathway edge. First, all nodes from the original pathway are added to the graphical model. No edges from the original pathway are added. Instead, for each pathway edge $e_{ij}$ between pathway nodes $n_i$ and $n_j$, a graphical model node $n_{eij}$ is added representing the interaction. Then two edges are added to the graphical model going from each original node to the new interaction node, $n_{ie_{ij}}$ and $n_{je_{ij}}$. An overview of this process can be found in Figure S2. Each of the $K$ features $f_{ek}$ in $n_{eij}$ are computed as the normalized product of features from $n_i$ and $n_j$, here represented as $f_{ik}$ and $f_{jk}$:

$$f_{ek} = \frac{f_{ik} f_{jk}}{\sum_{l=1}^{K} f_{il} f_{jl}}$$

This is equivalent to how interaction localization probabilities are calculated in ComPPI. Parameters are tied such that all original nodes are represented by one set of model parameters, and all interaction nodes are represented by another. This can be seen in panel C of Figure S2, where $\phi_1$ is the set of model parameters that describes the relationship between input features for each protein and its localization, $\phi_2$ describes the relationship between each interaction's combined features and its localization, and $\phi_3$ describes the relationship between each protein and the interactions it participates in.

Final localization labels can be viewed as a maximum a posteriori (MAP) estimate of the configuration of all interaction node labels. Decoding was performed using loopy belief propa-

gation, which approximates the MAP estimate via a message passing algorithm. Loopy belief propagation was run for 10,000 iterations in all cases. Both the NaivePGM and TrainedPGM models were implemented in the Direct Graphical Models software library[i] v1.7.0.

#### S1.2.3. *Other Classification Models*

Two non-neural network classifiers were used to further examine the effect of incorporating topological information into localization prediction: logistic regression, referred to as Logit, and random forests, referred to as RF. Interactions were represented by concatenating the features of the two nodes that make up that interaction in alphanumeric order by protein identifier, as shown in Figure S1. Tested hyperparameter ranges for these models are listed in Table S1. Both models were implemented in Python 3.9 using the Scikit-Learn package v1.0.2.

### S1.3. *Data*

#### S1.3.1. *Pathway Databases*

Pathway datasets were constructed from the Reactome and PathBank databases. Pathways were downloaded from Pathway Commons, and localization information was retrieved from BioPax pathway representations using the PyBioPax package v0.1.0[j]. Reactome contains localization information for all edges. In PathBank, however, approximately 9% of edges have missing localization information. These edges with missing data were excluded from all analyses.

After the protein-complex collapsing step, all pathways with fewer than 4 nodes were excluded from the analysis. This resulted in 953 Reactome pathways and 467 PathBank pathways.

Both pathway databases contain a highly skewed distribution of localizations across all interactions. The rarest localization in both databases, secretory-pathway and nucleus in Reactome and PathBank, respectively, occurs in less than 0.5% of all edges. The most common localization, which is cytosol for both databases, consists of 38% of Reactome interactions and 52% of PathBank interactions.

#### S1.3.2. *Protein Localization Databases*

ComPPI is a meta-database for protein subcellular localizations. It combines data from 8 subcellular localization databases. It does not include data from Compartments. Proteins are assigned scores for each of 6 subcellular locations: cytosol, plasma membrane, mitochondrion, extracellular, nucleus, and secretory-pathway. These 6 locations were used for all predictions; localizations for all other data sources were mapped to these 6. ComPPI combines weights for different types of evidence across its data sources to give the probability of a protein to be found in a particular subcellular location. All human ComPPI data was retrieved on 2020-11-09.

[i] `https://research.project-10.de/dgmdoc/index.html`
[j] `https://github.com/indralab/pybiopax`

Compartments is a protein subcellular localization database that combines data from 4 different types of data: database annotations, experimental screens, automated text mining, and predictive sequence-based models. Each data source is given a confidence score between 1 and 5 based on the level of evidence. Compartments assigns proteins to 1 of 11 subcellular locations. These 11 locations were mapped to the 6 localizations in the ComPPI database. All Compartments data was retrieved on 2021-09-29.

#### S1.3.3. *UniProt Keyword Features*

To explore the value of additional protein data in predicting localization data, UniProt keywords were collected for all human proteins. UniProt keywords are a controlled hierarchical vocabulary that represent a variety of protein categories such as molecular function, disease participation, structural features, and biological processes. These keywords are manually assigned and include localization data. While UniProt keywords provide a range of protein-level data, they consist of hundreds of terms, many of which are only used by a handful of proteins. Thus, they are impractical to use directly as features.

Keywords were converted into features through dimensionality reduction. Principal component analysis was performed on all keywords present in at least 5% of human proteins. Technical keywords such as "3D-Structure" and "Reference proteome" were excluded as not pertaining directly to the protein itself. Each protein was then represented by the first 6 components, chosen by a dropoff in explained variance after the first 6 components. The most important keywords for these components represented a variety of biological concepts, from functional categories such as "Tumor suppressor" and "Lipid biosynthesis" to structural features such as "ANK repeat" and "Voltage-gated channel". These components only accounted for 42% of variance. However, given the diversity of keywords, it is unlikely a small number of features could fully represent them.

### S1.4. *Metrics*

We used a multiclass F1 score to evaluate predictive performance. The F1 score is the harmonic mean of the precision and recall. In order to extend the F1 score from binary classification to a multiclass setting, the multiclass F1 score is defined as the average F1 score over all classes, treating each class as a one-against-all binary classification. This aggressively penalizes predictions if either the precision or recall is low. It is calculated as:

$$F1 = \frac{1}{L}(\sum_{i=1}^{L} \frac{2P_i R_i}{P_i + R_i})$$

where $L$ is the number of class labels, $P_i$ is the precision of class $i$ treated as a one-against-all binary classification, calculated as:

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

and $R_i$ is the recall of class $i$ treated as a one-against-all binary classification, calculated as:

$$R_i = \frac{TP_i}{TP_i + FN_i}$$

where $FP_i$ is the number of instances incorrectly predicted to be class $i$, $TP_i$ is the number of instances correctly predicted to be class $i$, and $FN_i$ is the number of instances incorrectly predicted to not be class $i$. This is considered the "macro" F1 score.

## S2. Supplementary Results

### S2.1. *Database Localization Disagreements*

The Compartments database has more disagreement with pathway databases than ComPPI. Although the range of ComPPI scores for all interaction localizations is wide, including a significant number of proteins existing in localizations where they have a score of 0, in all subcellular locations except for secretory-pathways the median ComPPI score is highest for the corresponding Reactome localization (Figure S3). However, in Compartments the cytosol has the highest median score across the majority of all Reactome localizations (Figure S4).

Reactome and PathBank also have some disagreement in their localizations. On average, highly matching pathways Reactome and PathBank have 79% agreement in their assigned interaction localizations. Highly matching pathways were calculated as those with at least 90% of edges in one pathway appearing in the other. This shows a moderate amount of disagreement even between manually curated pathway databases.

### S2.2. *Pathway Localization Prediction*

Overall, models without any method to transfer information across a pathway, the RF, Logit, and fully connected neural network models, tended to undersmooth within each pathway (Figure S7). The distributions for the RF, Logit, and fully connected neural network models are right-skewed as compared to the true distribution, with a sizable proportion of pathways being predicted to have 5 or 6 different localizations. This is unsurprising, as these models contained no topological information with which to encourage proteins belonging to the same pathway to have the same localization. These models greatly underestimated the proportion of pathways with a single localization.

The TrainedPGM in particular tended to oversmooth (Figure S7). It predicted almost all pathways as having a single localization across both datasets. It also performs particularly poorly for pathways with 2 and 3 unique localizations. This is likely due to these pathways being more evenly split between multiple localizations than those with 4 or 5 localizations, resulting in a single localization prediction to perform poorly on these pathways.

## S3. Supplementary Figures

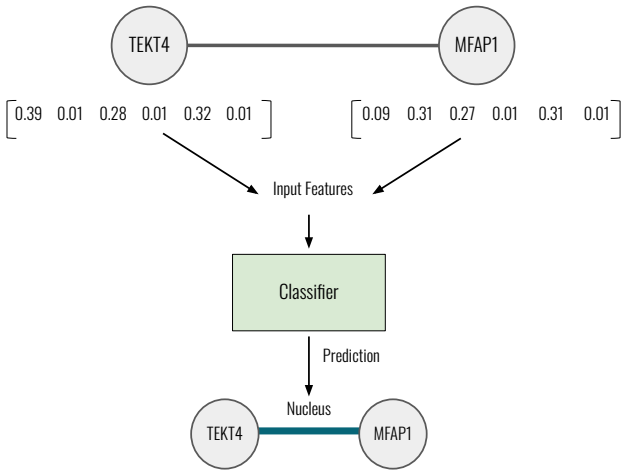


Fig. S1. Overview of how topology-free classifiers are used for the edge labeling task of localization prediction.

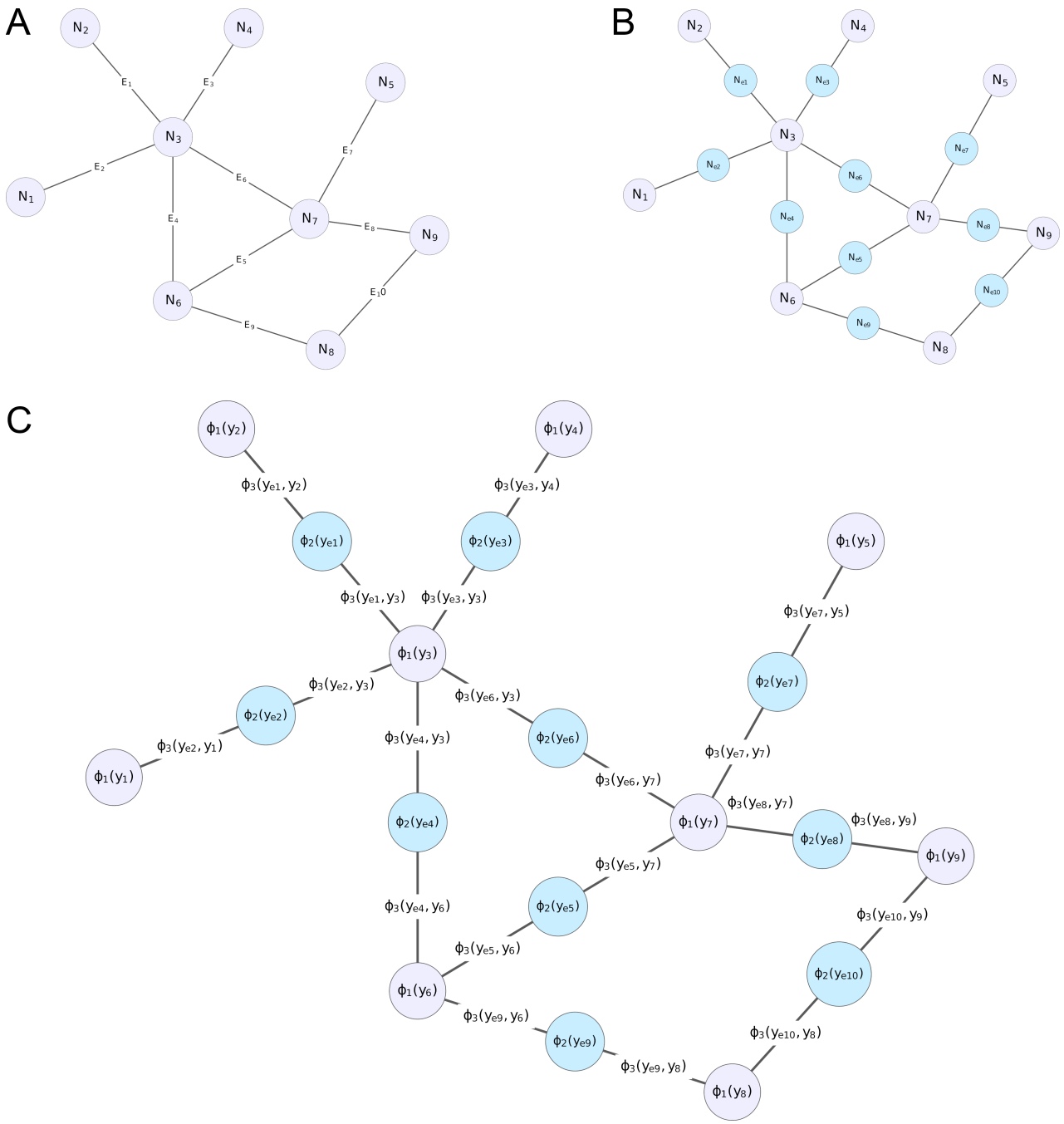


Fig. S2. Overview of how pathways were represented as probabilistic graphical models for interaction classification. Panel A shows the original pathway structure. Panel B shows the variables that are added to the graphical model to represent interactions in the pathway. Finally, Panel C shows how potential functions are used and tied. There are 2 sets of unary potentials, $\phi_1()$ and $\phi_2()$, which model the original nodes and the interaction nodes, respectively. $\phi_3()$ models how each interaction relates to its adjacent nodes.

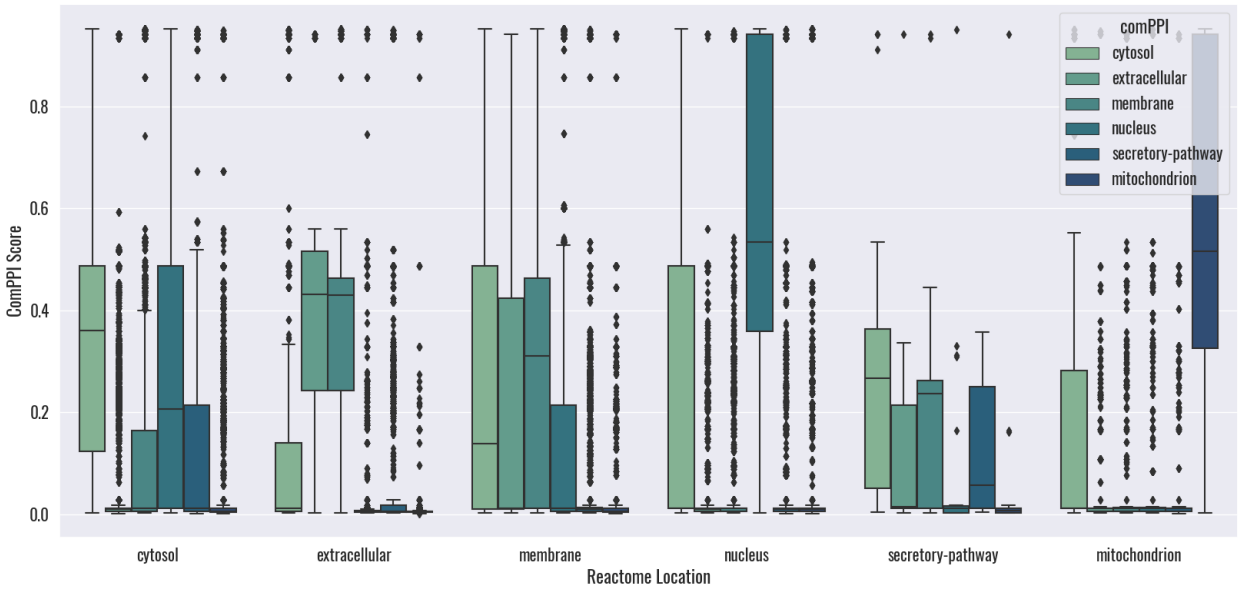


Fig. S3. Distribution of ComPPI protein scores by the localization of Reactome edges they belong to. Scores are the probability of a protein being in a given subcellular location.

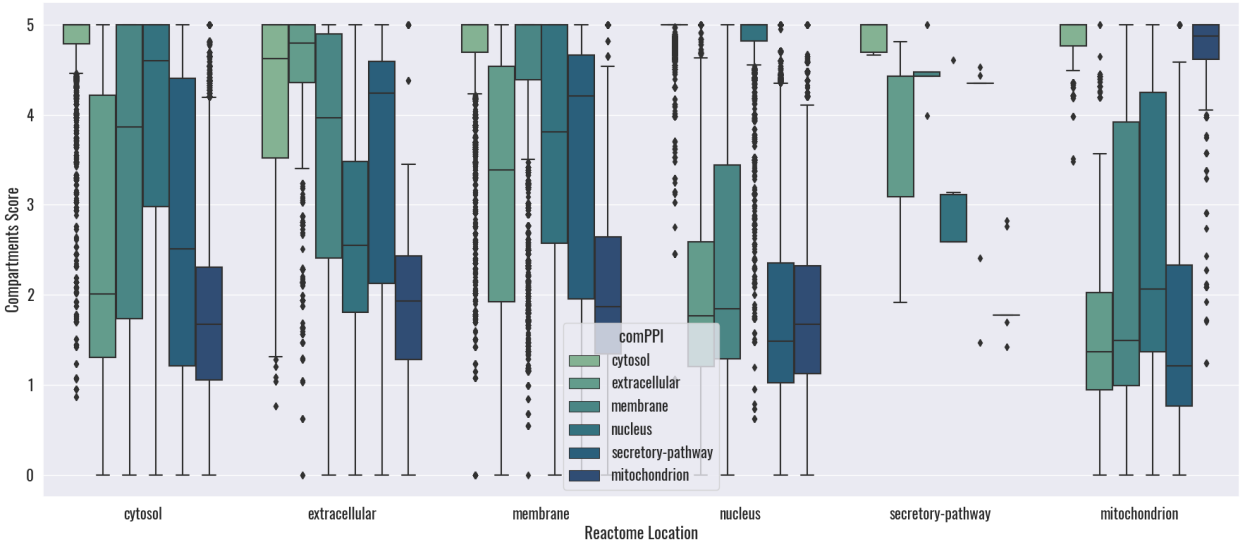


Fig. S4. Distribution of Compartments protein scores by the localization of Reactome edges they belong to. Scores are confidence scores of a protein being in a given subcellular location, weighted by the type and amount of evidence available.

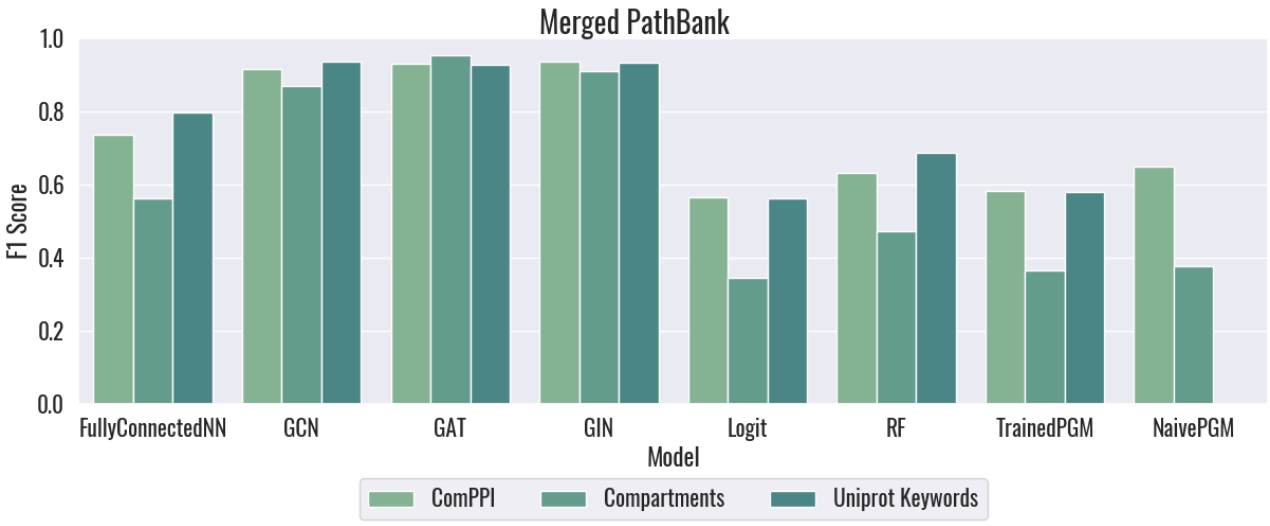


Fig. S5. F1 score of predictive performance on PathBank localizations. All pathway edges are merged and measured together, resulting in 97,792 edges total.

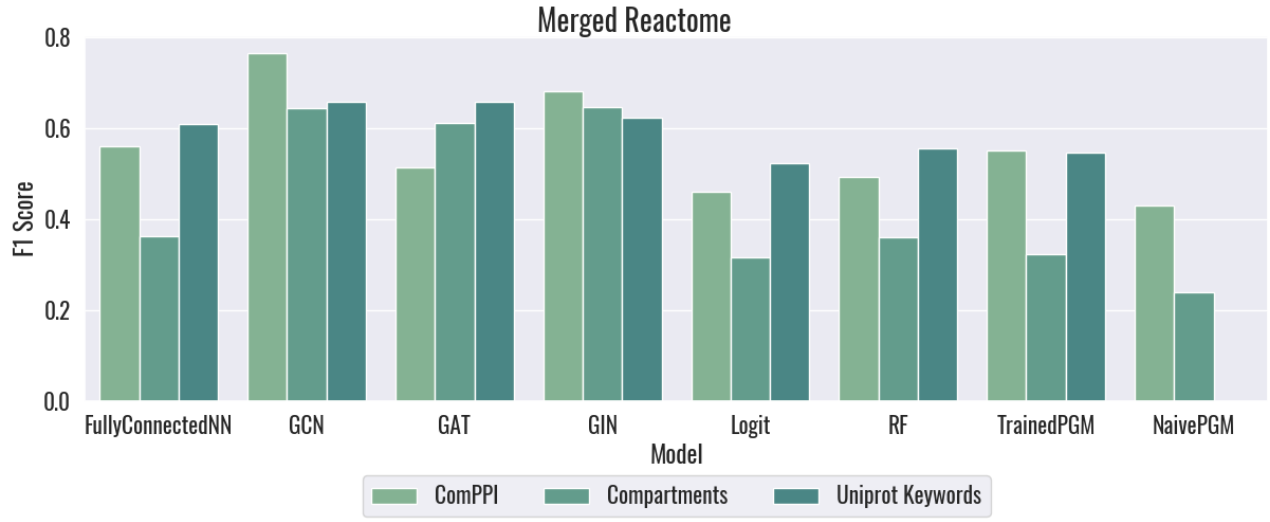


Fig. S6. F1 score of predictive performance on Reactome localizations. All pathway edges are merged and measured together, resulting in 83,855 edges total.

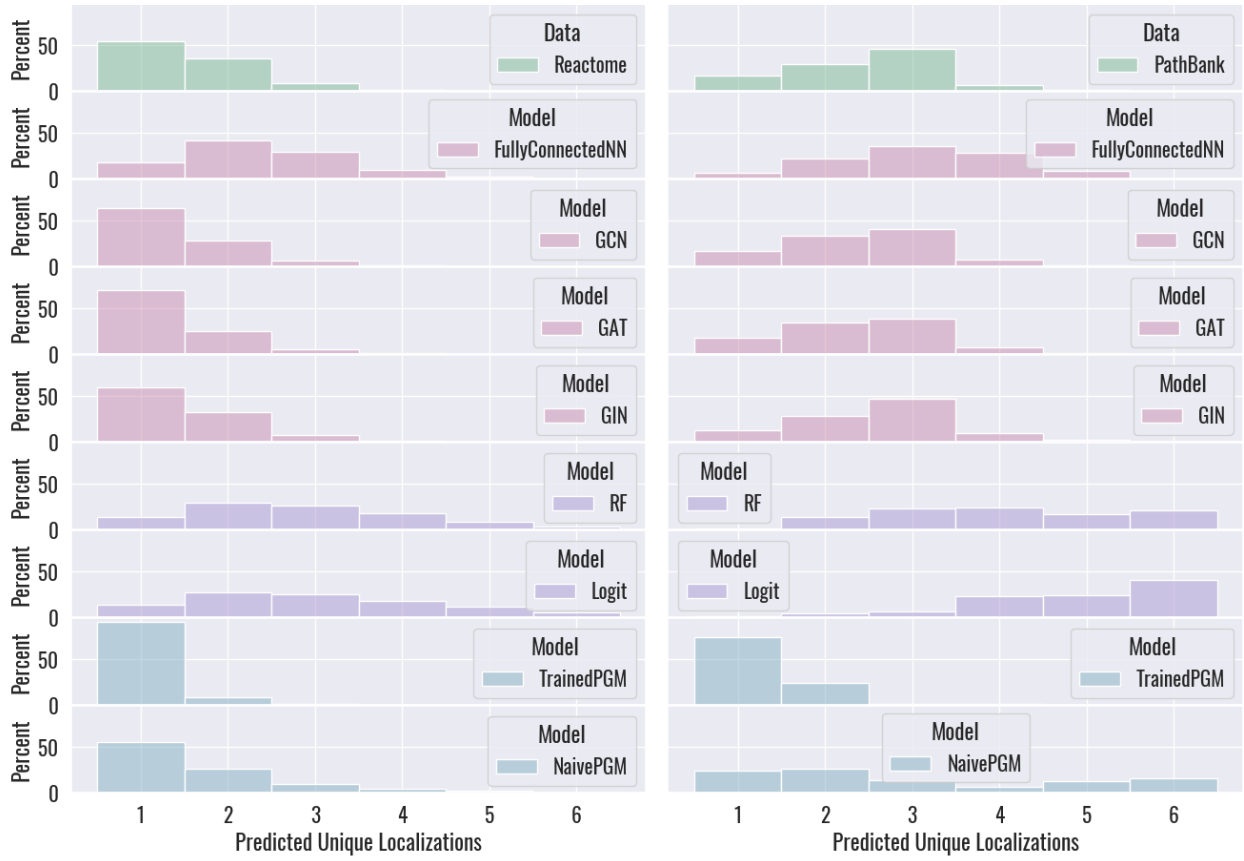


Fig. S7. Distributions of the number of unique localizations in each pathway database and predicted by each model on each pathway database. The left column shows distributions for predictions on the Reactome pathway database, and the right column shows distributions for predictions on the PathBank database.

Fig. S8. Topology of the best ranked pathway reconstruction of HCMV infection at 120hpi, containing 1,226 interactions. Pathways were reconstructed using Omics Integrator 2.

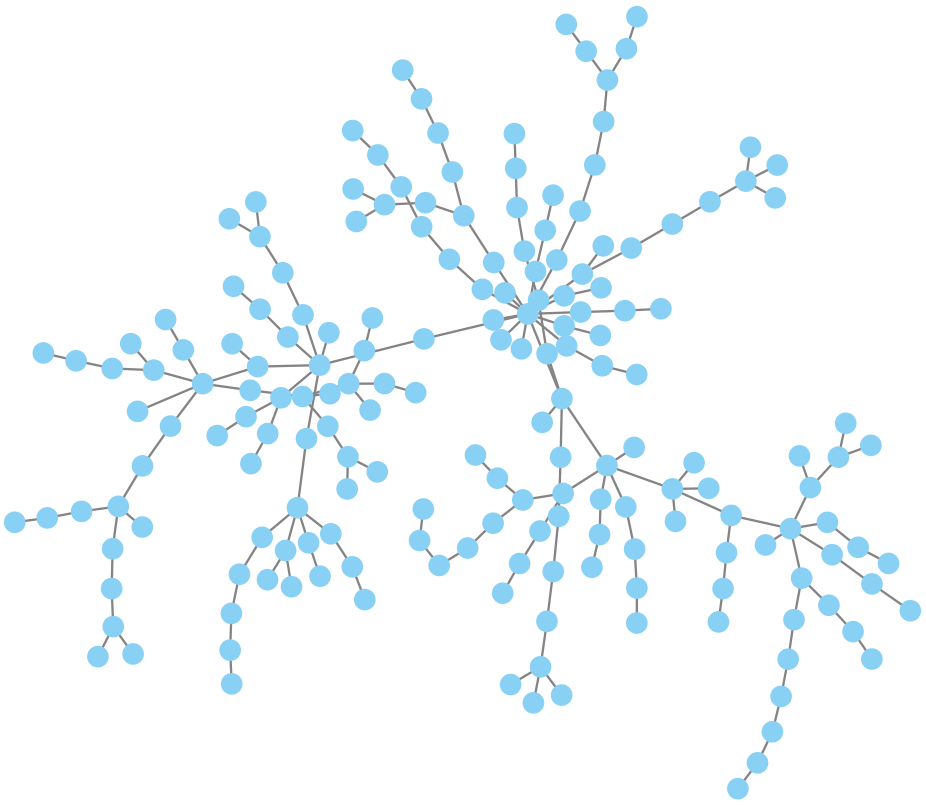

Fig. S9. Topology of the best ranked pathway reconstruction of EGF stimulation, containing 187 interactions. Pathways were reconstructed using Omics Integrator 2.

## S4. Supplementary Tables

Table S1: Hyperparameter ranges searched for each model.

| Model | Hyperparameter | Description | Range |
|---|---|---|---|
| All Neural Networks | Learning Rate | Learning rate for training. | $10^{-5} - 0.01$ |
| | Linear Depth | The number of linear layers. | $1 - 5$ |
| | Convolutional Depth | The number of convolutional layers. Not used by linear network. | $1 - 10$ |
| | Dim | The number of dimensions in hidden layers. | $24 - 128$ |
| | Dropout | Whether or not to add dropout while training. | True, False |
| Graph Convolutional Network | | No Unique Parameters. | |
| Graph Attention Network | Heads | The number of attention heads. | $1 - 5$ |
| Graph Isomorphism Network | | No Unique Parameters. | |
| Fully Connected Network | Activation | Activation function used. | Tanh, ReLU |
| Random Forest | max_depth | Maximum tree depth. | $1 - 10$ |
| | min_samples_split | Minimum samples to create branches. | $2 - 10$ |
| | n_estimators | Number of trees. | $1 - 100$ |
| | class_weight | Whether to balance class weights. | True, False |
| Logistic Regression | tol | Tolerance for training. | $10^{-6} - 0.1$ |
| | penalty | Regularization penalty to use. | L2, None |
| | C | Regularization strength (lower is stronger). | $0.01 - 100$ |
| | class_weight | Whether to balance class weights. | True, False |

Table S2: All hyperparameter values used.

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| FullyConnectedNN | Reactome | ComPPI | | |
| | | | lRate | 0.002 |
| | | | l_depth | 2 |
| | | | dropout | 0.500 |
| | | | dim | 83 |
| | | | activation | tanh |
| FullyConnectedNN | Reactome | Compartments | | |
| | | | lRate | 0.008 |
| | | | l_depth | 1 |
| | | | dropout | 0.500 |
| | | | dim | 82 |
| | | | activation | tanh |
| FullyConnectedNN | Reactome | Uniprot KW | | |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| | | | lRate | $8.38e-04$ |
| | | | l_depth | 3 |
| | | | dropout | 0.500 |
| | | | dim | 98 |
| | | | activation | relu |
| FullyConnectedNN | PathBank | ComPPI | | |
| | | | lRate | 0.010 |
| | | | l_depth | 1 |
| | | | dropout | 0 |
| | | | dim | 41 |
| | | | activation | tanh |
| FullyConnectedNN | PathBank | Compartments | | |
| | | | lRate | 0.009 |
| | | | l_depth | 2 |
| | | | dropout | 0 |
| | | | dim | 103 |
| | | | activation | relu |
| FullyConnectedNN | PathBank | Uniprot KW | | |
| | | | lRate | 0.006 |
| | | | l_depth | 5 |
| | | | dropout | 0.500 |
| | | | dim | 81 |
| | | | activation | tanh |
| GCN | Reactome | ComPPI | | |
| | | | lRate | 0.004 |
| | | | l_depth | 1 |
| | | | dropout | 0 |
| | | | dim | 81 |
| | | | c_depth | 6 |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| GCN | Reactome | Compartments | | |
| | | | lRate | $1.48e-04$ |
| | | | l_depth | 1 |
| | | | dropout | 0.500 |
| | | | dim | 108 |
| | | | c_depth | 2 |
| GCN | Reactome | Uniprot KW | | |
| | | | lRate | 0.003 |
| | | | l_depth | 4 |
| | | | dropout | 0 |
| | | | dim | 87 |
| | | | c_depth | 3 |
| GCN | PathBank | ComPPI | | |
| | | | lRate | 0.002 |
| | | | l_depth | 1 |
| | | | dropout | 0 |
| | | | dim | 44 |
| | | | c_depth | 8 |
| GCN | PathBank | Compartments | | |
| | | | lRate | 0.006 |
| | | | l_depth | 2 |
| | | | dropout | 0 |
| | | | dim | 96 |
| | | | c_depth | 1 |
| GCN | PathBank | Uniprot KW | | |
| | | | lRate | 0.003 |
| | | | l_depth | 3 |
| | | | dropout | 0 |
| | | | dim | 108 |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| | | | c_depth | 5 |
| GAT | Reactome | ComPPI | | |
| | | | lRate | 0.003 |
| | | | l_depth | 2 |
| | | | dropout | 0 |
| | | | dim | 48 |
| | | | c_depth | 7 |
| | | | num_heads | 4 |
| GAT | Reactome | Compartments | | |
| | | | lRate | $2.78e-04$ |
| | | | l_depth | 4 |
| | | | dropout | 0.500 |
| | | | dim | 46 |
| | | | c_depth | 1 |
| | | | num_heads | 5 |
| GAT | Reactome | Uniprot KW | | |
| | | | lRate | 0.010 |
| | | | l_depth | 2 |
| | | | dropout | 0 |
| | | | dim | 31 |
| | | | c_depth | 4 |
| | | | num_heads | 1 |
| GAT | PathBank | ComPPI | | |
| | | | lRate | 0.003 |
| | | | l_depth | 4 |
| | | | dropout | 0.500 |
| | | | dim | 45 |
| | | | c_depth | 4 |
| | | | num_heads | 3 |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| GAT | PathBank | Compartments | | |
| | | | lRate | 0.002 |
| | | | l_depth | 3 |
| | | | dropout | 0.500 |
| | | | dim | 43 |
| | | | c_depth | 3 |
| | | | num_heads | 4 |
| GAT | PathBank | Uniprot KW | | |
| | | | lRate | 0.004 |
| | | | l_depth | 1 |
| | | | dropout | 0 |
| | | | dim | 39 |
| | | | c_depth | 2 |
| | | | num_heads | 4 |
| GIN | Reactome | ComPPI | | |
| | | | lRate | 0.005 |
| | | | l_depth | 2 |
| | | | dropout | 0 |
| | | | dim | 95 |
| | | | c_depth | 3 |
| GIN | Reactome | Compartments | | |
| | | | lRate | $9.77e-04$ |
| | | | l_depth | 4 |
| | | | dropout | 0.500 |
| | | | dim | 24 |
| | | | c_depth | 1 |
| GIN | Reactome | Uniprot KW | | |
| | | | lRate | 0.001 |
| | | | l_depth | 3 |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| | | | dropout | 0.500 |
| | | | dim | 60 |
| | | | c_depth | 2 |
| GIN | PathBank | ComPPI | | |
| | | | lRate | 0.002 |
| | | | l_depth | 4 |
| | | | dropout | 0 |
| | | | dim | 68 |
| | | | c_depth | 1 |
| GIN | PathBank | Compartments | | |
| | | | lRate | 0.001 |
| | | | l_depth | 3 |
| | | | dropout | 0.500 |
| | | | dim | 102 |
| | | | c_depth | 3 |
| GIN | PathBank | Uniprot KW | | |
| | | | lRate | $9.86e-04$ |
| | | | l_depth | 2 |
| | | | dropout | 0 |
| | | | dim | 80 |
| | | | c_depth | 1 |
| Logit | Reactome | ComPPI | | |
| | | | C | 0.620 |
| | | | class_weight | balanced |
| | | | penalty | l2 |
| | | | tol | $1.00e-06$ |
| Logit | Reactome | Compartments | | |
| | | | C | 91.893 |
| | | | class_weight | balanced |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| | | | penalty | l2 |
| | | | tol | 0.062 |
| Logit | Reactome | Uniprot KW | | |
| | | | C | 7.604 |
| | | | class_weight | balanced |
| | | | penalty | l2 |
| | | | tol | 0.078 |
| Logit | PathBank | ComPPI | | |
| | | | C | 0.044 |
| | | | class_weight | balanced |
| | | | penalty | l2 |
| | | | tol | $1.84e-05$ |
| Logit | PathBank | Compartments | | |
| | | | C | 0.451 |
| | | | class_weight | balanced |
| | | | penalty | l2 |
| | | | tol | $4.71e-06$ |
| Logit | PathBank | Uniprot KW | | |
| | | | C | 33.176 |
| | | | class_weight | balanced |
| | | | penalty | l2 |
| | | | tol | 0.033 |
| RF | Reactome | ComPPI | | |
| | | | class_weight | balanced |
| | | | max_depth | 3 |
| | | | min_samples_split | 2 |
| | | | n_estimators | 76 |
| RF | Reactome | Compartments | | |
| | | | class_weight | balanced |

Continued on next page

Table S2: All hyperparameter values used. (Continued)

| Model | Dataset | Features | Hyperparameter | Value |
|---|---|---|---|---|
| | | | max_depth | 9 |
| | | | min_samples_split | 3 |
| | | | n_estimators | 81 |
| RF | Reactome | Uniprot KW | | |
| | | | class_weight | balanced |
| | | | max_depth | 6 |
| | | | min_samples_split | 3 |
| | | | n_estimators | 100 |
| RF | PathBank | ComPPI | | |
| | | | class_weight | balanced |
| | | | max_depth | 10 |
| | | | min_samples_split | 10 |
| | | | n_estimators | 72 |
| RF | PathBank | Compartments | | |
| | | | class_weight | balanced |
| | | | max_depth | 10 |
| | | | min_samples_split | 10 |
| | | | n_estimators | 92 |
| RF | PathBank | Uniprot KW | | |
| | | | class_weight | balanced |
| | | | max_depth | 10 |
| | | | min_samples_split | 10 |
| | | | n_estimators | 58 |