Node Duplication in Disease Maps using Graph Neural Networks

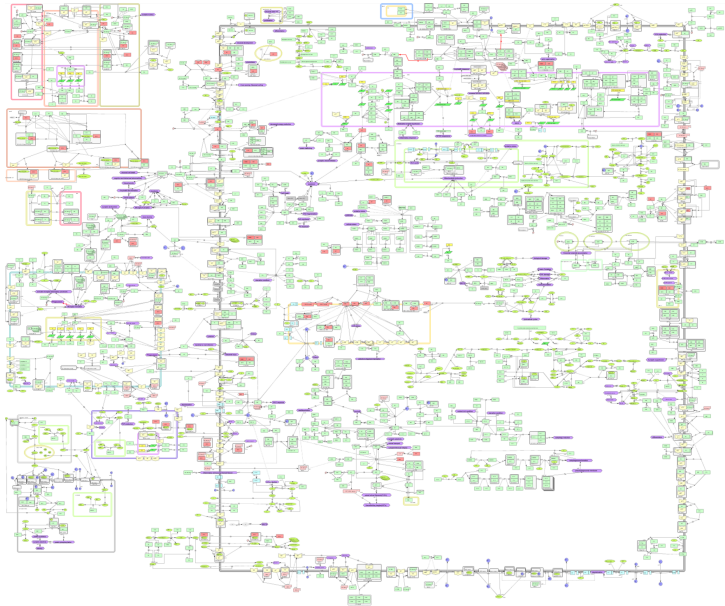
- In preliminary layouts of disease maps, a species alias may be connected to many different processes
- Unclear whether connections are meaningful or merely artifact of creation process
- Faithful network representation should not have such connections $\rightsquigarrow$ duplicate some nodes
- Decision which nodes may be duplicated is not trivial $\rightsquigarrow$ consider ML model trained on expert decisions
- ...

- Some diseases do not depend on a single biological pathway
  Alzheimer's Disease, Parkinson's Disease
- Due to complex interactions of biol. processes, or genetic or environmental factors
- Seek to facilitate a **systematic** understanding of involved biological entities and their relationships.

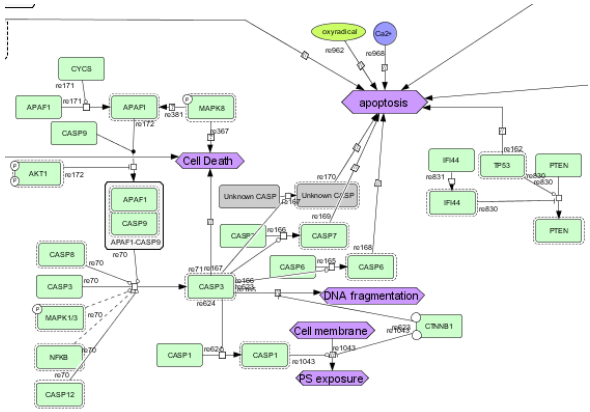
Disease Map

Comprehensive visual diagrams describing all known mechanisms relevant for a given disease

- particularly suited for visual, interactive exploration
- serve as comprehensive knowledge base

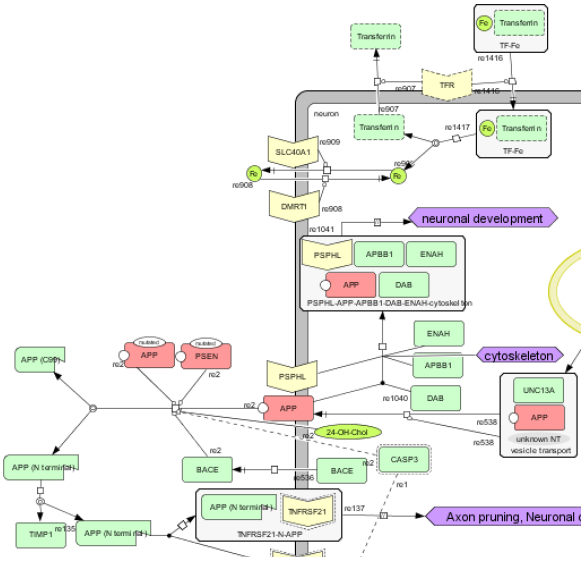


Zoomed-out view of the *AlzPathway* disease map, a diagram describing biological processes related to Alzheimer's Disease [1].



- **species alias**: visual representation of biological actor (**species**)
- **process**: interaction, relationship between species aliases
- complex species aliases: groups
- compartment boundaries

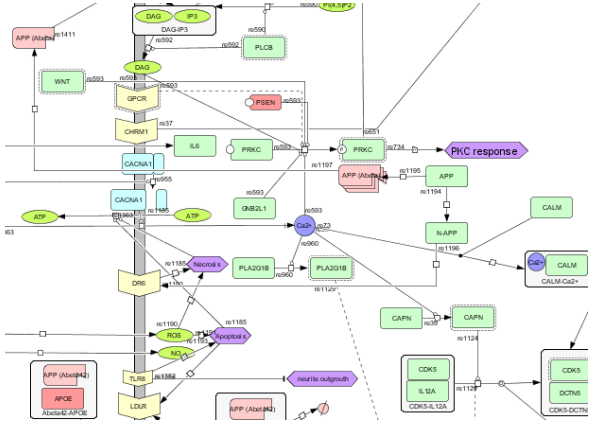
Can be interpreted as **attributed, bipartite graph** of species aliases and processes



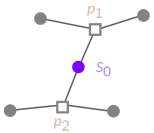
Network structure and layout need to convey complex information faithfully:

- Different kinds of connections
- complexes, compartments
- visual hierarchy
- layout constraints for specific subgraphs (line, circle)
- represent species by multiple aliases
- ...

Creating disease maps still involves considerable manual effort by domain expert



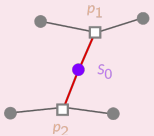
- A species alias may be connecting multiple processes
- This may be intentional
- ... or artifact of combining several subgraphs



To obtain an informative network structure, we need to distuingish...

Path (p_1, S_0, p_2) is semantically meaningful (**true connectivity**)

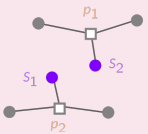
$\rightsquigarrow$ S_0 must not be duplicated



Path (p_1, S_0, p_2) is not meaningful (**false connectivity**)

There should be no paths implying false connectivity

$\rightsquigarrow$ S_0 should be **duplicated**



e.g. due to unrelated roles of S_0 in p_1, p_2, not stoichiometrically linked, unimportant byproduct

Objective 1

Assess whether a given species alias implies false connectivity (and should thus be duplicated)

Objective 2

Determine number of duplicates and attachment of edges

Objective 1

Assess whether a given species alias implies false connectivity (and should thus be duplicated)

Previous approaches would rely on **node centrality scores**

high centrality $\rightsquigarrow$ heterogeneous neighbourhood $\rightsquigarrow$ false connectivity

- node degree [2, 3]
- eigenvector centrality [4]
- communities (modularity)
  - contribution to modularity if node removed [5]
  - based on intra- & inter-community degrees [6]
- communities (semantic)
  - cellular compartment [4]
  - pathway annotation [7, 8, 9]

Objective 1

Assess whether a given species alias implies false connectivity (and should thus be duplicated)

- No clear requirements or guidelines (yet)
- Previous work relies on heuristic rules
- Decision potentially depends on biological domain knowledge.

⇝ Try to learn rules from examples provided by domain expert

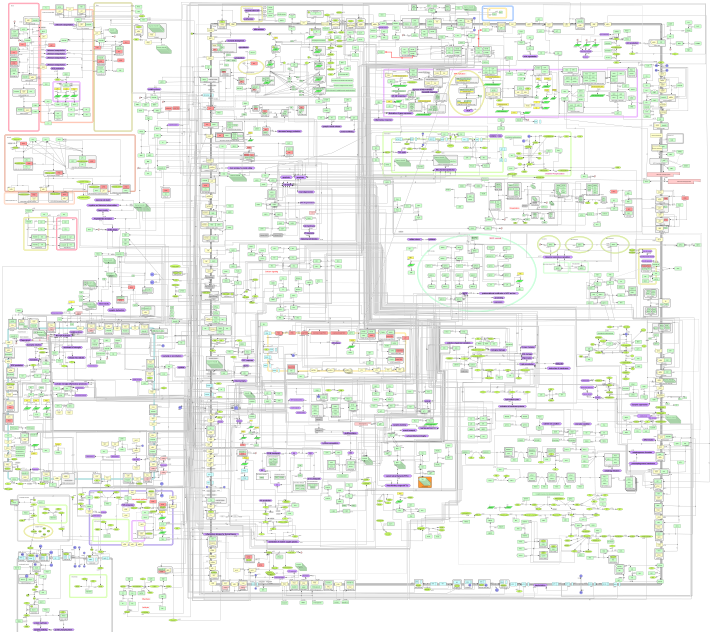
- Decision depends on *context* (neighbourhood) of given species alias

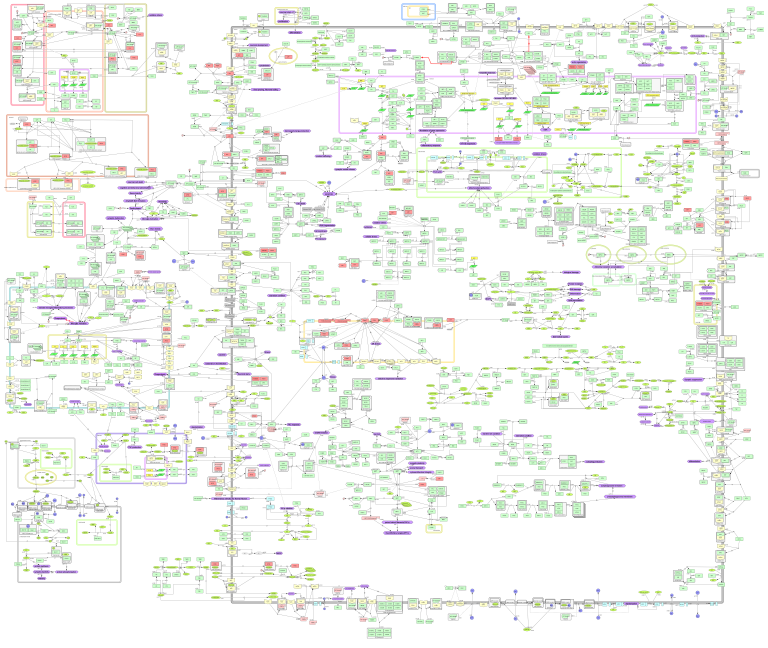
⇝ Exploit information on graph structure

Given **expert decisions**, train a ML model for **supervised node classification** to predict node duplication.

- **AlzPathway** is a disease map that describes signalling pathways related to Alzheimer's Disease
- Recently received additional curation of layout, including duplication of nodes
- Snapshots of intermediate progress were saved (**reorganisation steps**)
  Total for 18 reorganisation steps, 6 of them involving node duplications

Data / Reorganisation Steps

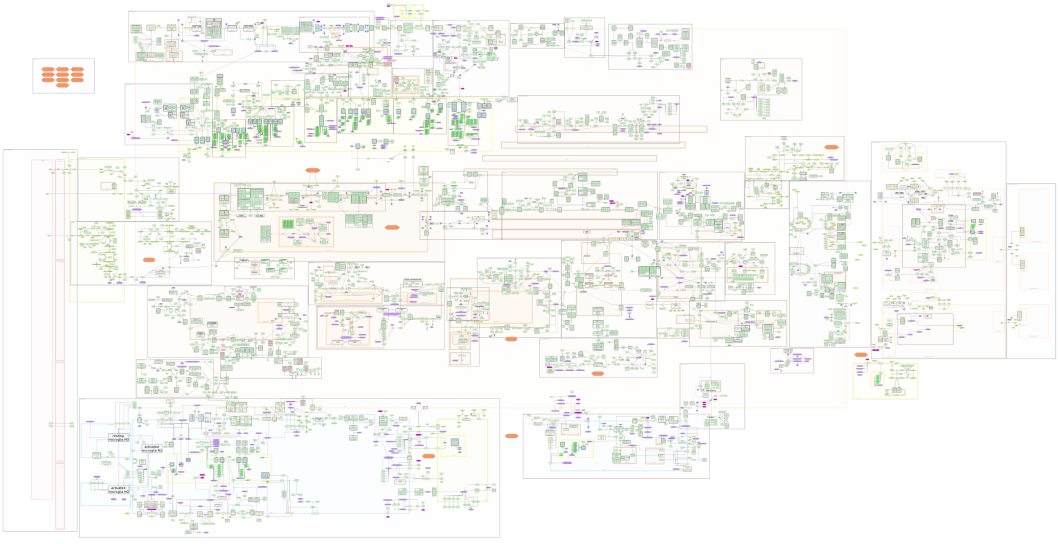




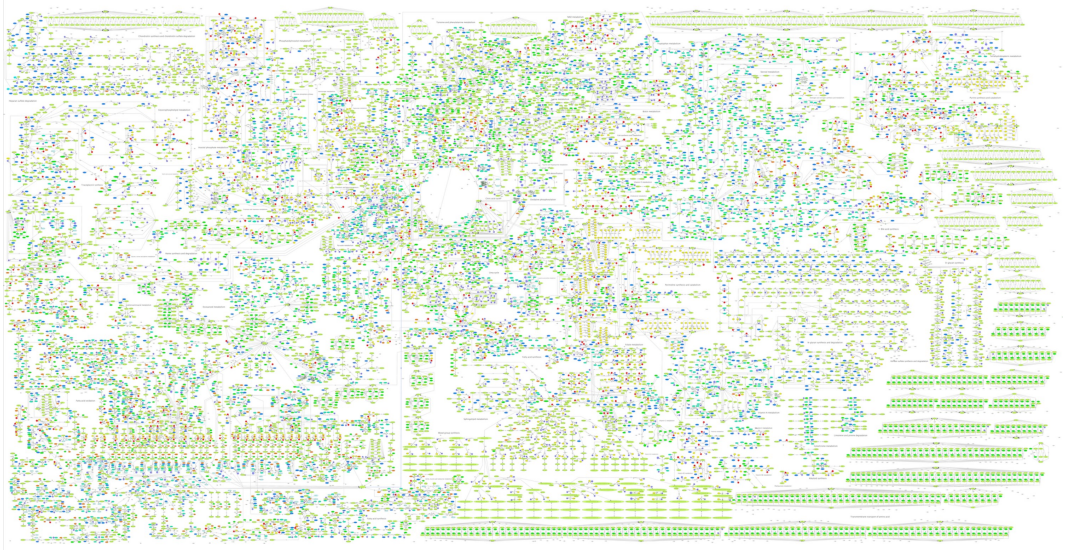
- Such reorganisation steps are hard to obtain in practice
  Only given for *AlzPathway*

- For *any* disease map graph, can create a single "step" by comparing it to its **collapsed** version
  - Replace all nodes (species aliases) corresponding to the same species with a single representative
  - Attach all edges of aliases to representative
- Consider two additional maps

TODO



Parkinson's Disease Map (PDMap) describes major pathways involved in pathogenesis of Parkinson's Disease

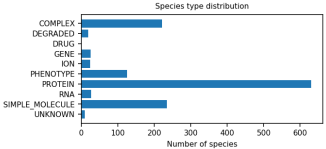


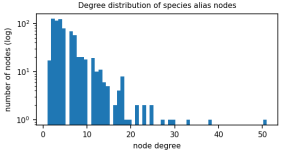
ReconMap [10], a visual representation of the *Recon 2* [11] GSMM

TODO datasets used

AlzPathwayReorgLast (collapsed)

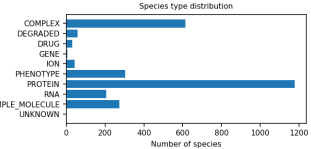
1315 species aliases, 982 reactions
719 (54.68%) for prediction
277 (38.53%) to-be-duplicated

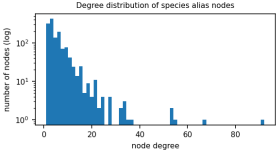




PDMap19 (collapsed)

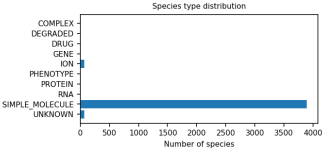
2706 species aliases, 2661 reactions
1389 (51.33%) for prediction
393 (28.29%) to-be-duplicated

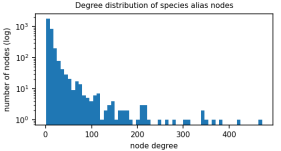




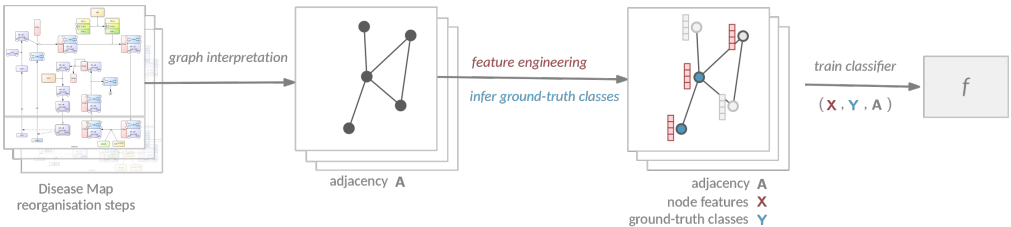
ReconMapOlder (collapsed)

4030 species aliases, 5535 reactions
3089 (76.65%) for prediction
741 (23.99%) to-be-duplicated

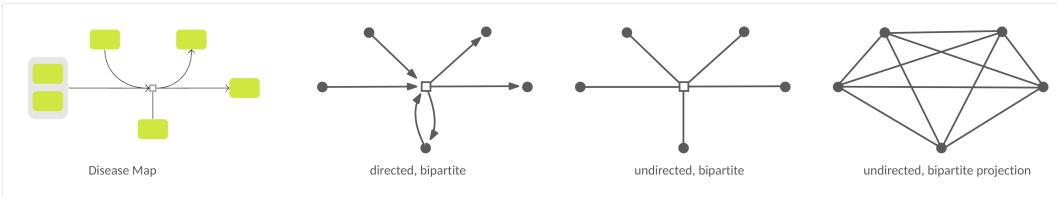




Given **expert decisions**, train a ML model for **supervised node classification** to predict node duplication.



1. **Graph Interpretation**



- Consider complex species aliases as single node
- No distinction between main substrates/products and side compounds
- Different interpretations for different situations

2. **Infer ground-truth labels**
   - Compare to next step in reorganisation sequence to identify *duplication parents*

3. & 4. **Feature Engineering & Classification**: in the following

Given **expert decisions**, train a ML model for **supervised node classification** to predict node duplication.

Recent work by Nielsen et al. [12]:

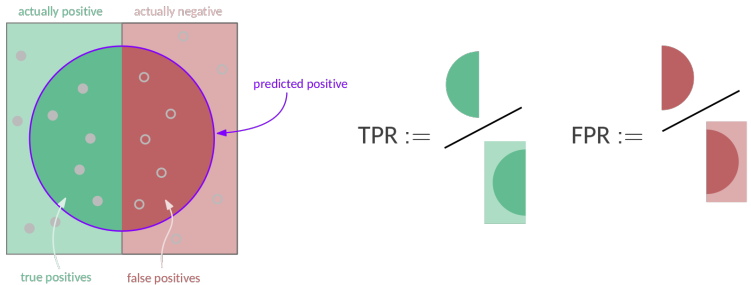
- Node features based on graph centralities
- Consider collapsed map plus reorganisation steps
- Supplied to Support Vector Machine classifier

We extend this in several directions:

- Explore different classifier (Graph Neural Networks)
- Explore importance of reorganisation steps
- Explore choice of features
- Heuristic for determining number of duplicates and edge attachment

- To compare classifiers, we need an **unbiased performance measure**
- Classifiers used herein yield a **confidence score** in $[0, 1]$ for a given example
- Obtain concrete classification by setting a **decision threshold**, yields...

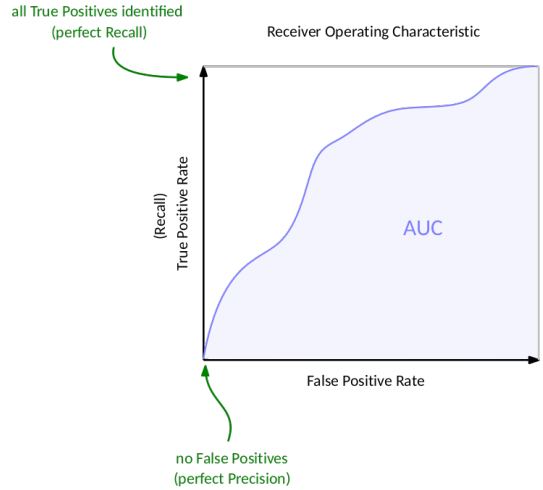
- **True Positive Rate** (TPR): $\#\,\text{true positives}/\#\,\text{actually positive}$
- **False Positive Rate** (FPR): $\#\,\text{false positives}/\#\,\text{actually negative}$



- Focus on positive class, Insensitive to class imbalance

ROC Curve: Plot TPR, FPR as function of decision threshold

- Useful properties:
  - Show overall behaviour with respect to variable threshold
  - Insensitive to class distribution
  - Insensitive to error costs
- Usually a tradeoff, choice depends on use-case
  - Accept only few high-confidence predictions → low FPR, but also low TPR (Recall)
  - Lower decision threshold → increase TPR at cost of increased FPR



Plot TPR and FPR as function of decision threshold

maybe recap slide here (then: ok, so now we have all the background we need to actually do stuff)

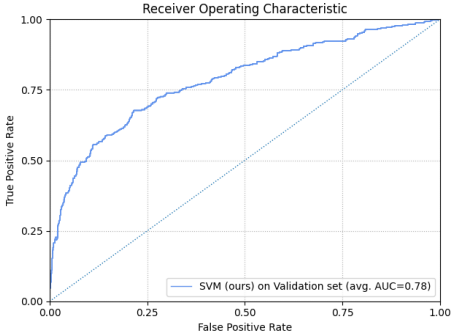
Reproducing work of Nielsen et al.:

Features for a node v

- Centrality scores (degree, betweenness, closeness and eigenvector centrality)
- Statistics of centrality scores of neighbours (*mean, min, max, stddev*)
- Clustering coefficient
- One-hot encoding of species type
- Number of nodes k hops from v for $k \in \{1, ..., 5\}$, normalised by resp. count in grid graph.

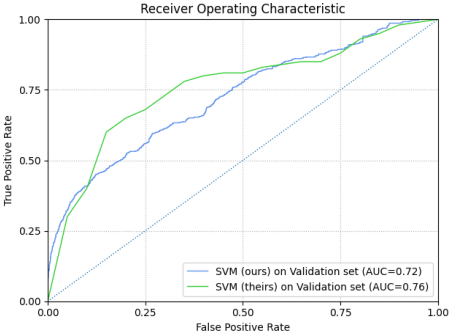
- Computed both on simple and bipartite graph interpretation where applicable
- Min-max normalised to $[0, 1]$

- **Train** on *AlzPathway* reorganisation steps plus collapsed version of first step
- **Evaluate** on *PDMap* and *ReconMap* (separately)



	AUC
SVM (theirs)	0.69
SVM (ours)	0.78

(a) (AlzPathwayReorg → PDMap)



	AUC
SVM (theirs)	0.76
SVM (ours)	0.72

(b) (AlzPathwayReorg → ReconMap)

- Our SVM implementation performs worse than that of Nielsen et al.
- Challenge when working with reorganisation steps: **contradictory examples**

- Duplicated node has positive label in G_k but negative label in G_j for $j < k$
- Reasonable only if we can assume that in reorg. step, all critical nodes are in fact being duplicated
- Likely not the case in *AlzPathway*, rather: simply not considered yet
- If reorganisations have little impact on features of other nodes:

$\rightsquigarrow$ Two training examples with potentially similar features but different label

- Nielsen et al.: exclude negative examples that are within 1% of feature space extent to ex. of positive class
- Here: Exclude node corresponding to positive example from previous reorganisation steps.
- More on this later

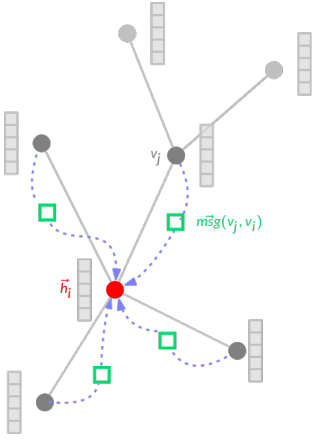
- Basic idea: duplication depends not only on characteristics of single node but also on its *context*
- GNN learns hidden representation of node based on its context.

GNN blueprint

New hidden representation of v is result of convolution localised around v:

$$\mathbf{h}_i' \leftarrow \textsc{Update}(\textsc{Agg}(\{\textsc{Msg}(\mathbf{h}_j, \mathbf{h}_i) \mid j \in \mathcal{N}_i\}))$$

1. $\textsc{Msg}(\mathbf{h}_j, \mathbf{h}_i)$: produce message from v_j to v_i
2. $\textsc{Agg}(\ldots)$: combine messages received by neighbours
3. $\textsc{Update}(\ldots)$: combine aggregated messages with own state



Simple GNN architectures:

- $\mathcal{N}_i$: 1-hop neighbourhood
- $\textsc{Msg}(\mathbf{h}_j, \mathbf{h}_i) = \mathbf{W}\mathbf{h}_j$
- $\textsc{Agg}(...) = \bigoplus_{j \in \mathcal{N}_i} \alpha_{ij} \textsc{Msg}(\mathbf{h}_j, \mathbf{h}_i)$

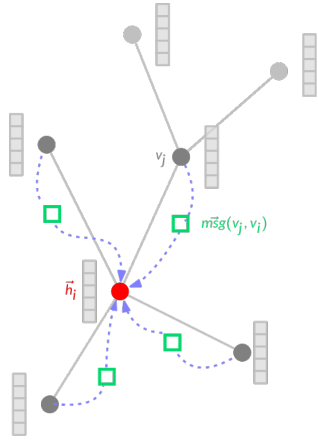
  $\bigoplus$: permutation-invariant agg., e.g. *sum*, *mean*, *max*
- $\textsc{Update}(...)$: activation function σ

GNN models

- **GCN** [13]: $\alpha_{ij} = 1/\sqrt{d_i d_j}$
- **GAT** [14]: α_{ij} determined by single-layer NN

$$e_{ij} := \sigma_{\text{att}}(\mathbf{a}^T(\mathbf{msg}_i \| \mathbf{msg}_j))$$

$$\alpha_{ij} := \text{softmax}_{j \in \mathcal{N}_i}(e_{ij}) := \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$$



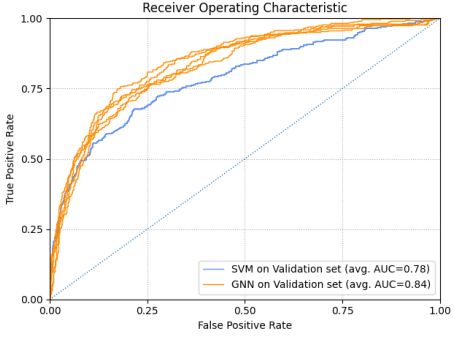
Apply GNN to previous task

- same input features
- message-passing on bipartite projection

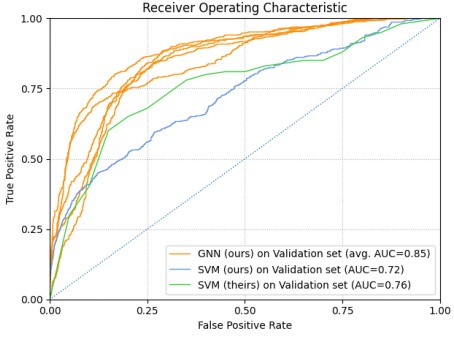
Architecture:

- 2 message-passing layers
- 2 fully-connected layers before and after message-passing

- *PDMap*: GNN model is slightly better
- *ReconMap*: outperforms our and their SVM
- Variablity w.r.t. random initialisation

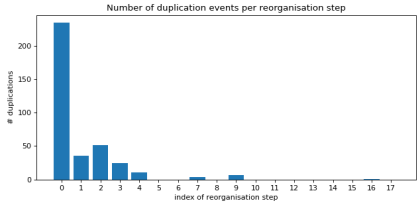


(a) (AlzPathwayReorg → PDMap)



(b) (AlzPathwayReorg → ReconMap)

- Note that we are evaluating on single collapsed disease maps
- ... but are also using reorganisation steps for training

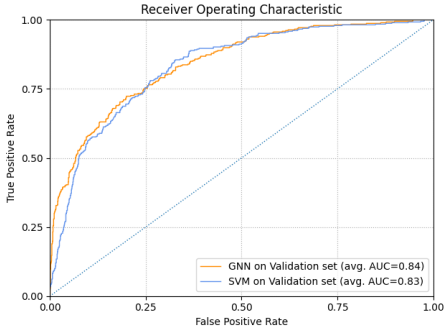


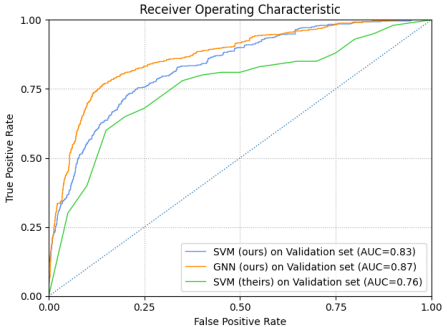
- most duplication events due to step from collapsed map
- node features in later reorganisation steps may be different from typical features of collapsed maps
- amplifies class imbalance
- need to consider evaluation methods and use-case

Are reorganisation steps actually important *for this task*?

$\rightsquigarrow$ **Train** on collapsed version of last reorg. step

- *PDMap*:
  - Gap between SVM, GNN disappears
  - GNN has advantage in high-confidence region
- *ReconMap*:
  - SVM now outperforms approach by Nielsen et al. (with reorg. steps)
  - No substantial impact on GNN performance





Summary

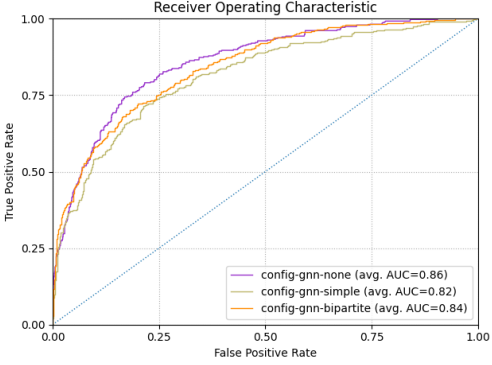
Similar, or better, performance while using only finished disease map

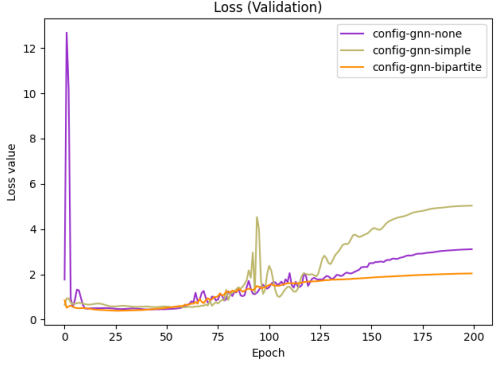
- Reorganisation steps potentially much harder to obtain in pratice
- Circumvent problem of contradictory examples

1. Should we perform message-passing on bipartite or simple graph structure?
2. How useful are message-passing layers at all? (GNN vs. MLP)

1. Compare both variants
   - For simple graph: compute feature vectors for all nodes, exclude reaction nodes from prediction
2. Replace message-passing layers by fully-connected layers

- Here, message-passing layers provide no substantial advantage
- Slight performance gain with simple MLP
- Different behaviour of loss





(AlzPathwayLast→ PDMap)

- Address class imbalance in GNN models
  - Undersample majority class
  - Weigh class in loss function

$$\mathcal{L}_{\text{BCE weighted}} = {}^{1}/_{n} \sum_{i=1}^{n} w_i y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i)$$

- Attention mechanism for GNN
  - coefficient α_{ij} determined by single-layer NN

$$e_{ij} := \sigma_{\text{att}}(\mathbf{a}^T(\mathbf{msg}_i \| \mathbf{msg}_j))$$

$$\alpha_{ij} := \text{softmax}_{j \in \mathcal{N}_i}(e_{ij}) := \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$$

  - ability to focus on "important" messages (*attention*)

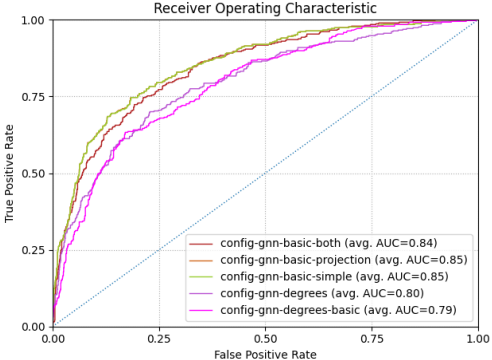
No substantial effect on model performance

$\rightsquigarrow$ consider quality of input data instead

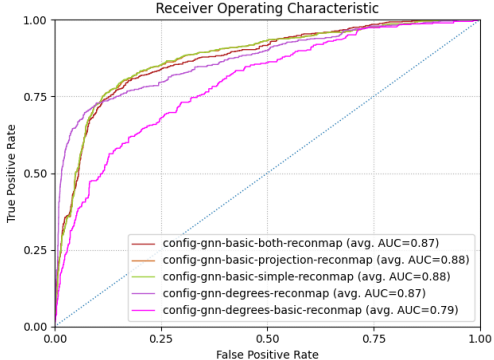
- Which of the used features are actually useful?

1. Structural features on simple or bipartite graph representation? Or both?
   - Processing and memory cost, implementation
2. How well does (directed) degree alone serve as predictor?
   - threshold used as heuristic in previous work
   - high in-degree, low out-degree (or vice versa) $\rightsquigarrow$ cofactor/byproduct $\rightsquigarrow$ duplicate?
   - balanced in-/out degrees $\rightsquigarrow$ main substrate/product

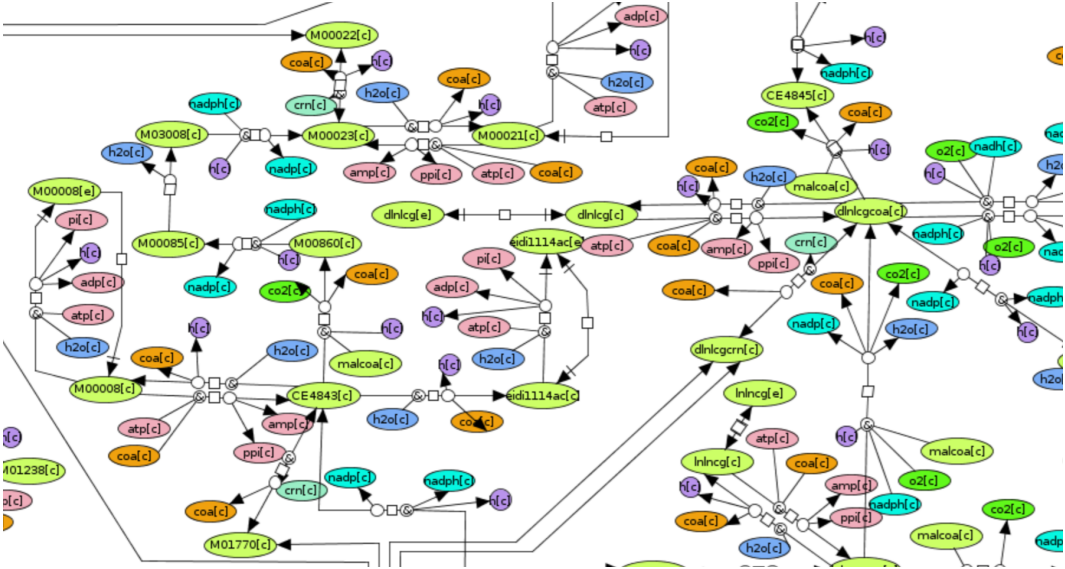
- No difference between using features based on simple or bipartite graph interpretation (using both hurts model performance)
- Degrees already strong indicator
- directed degrees particularly relevant for *ReconMap*



(a) (AlzPathwayLast→ PDMap)



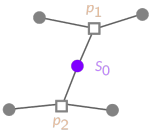
(b) (AlzPathwayLast→ ReconMap)



Detail view of *ReconMap*

- some species commonly represented with unit in/out-degree

- True/false connectivity may not depend on graph-structural characteristics alone
- **Basic idea**: Encode biological semantics of adjacent processes
- e.g. p_1 and p_2 are unrelated biological processes $\rightsquigarrow$ duplicate S_0?



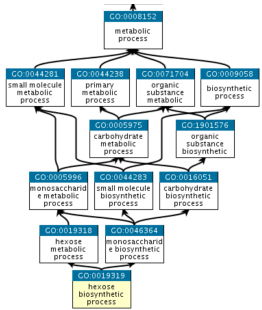
- Species in disease maps are associated with Gene Ontology terms

Gene Ontology

Directed, acyclic graph of biological **terms** and their **relationships**

- terms: molecular functions, cellular compartments, biological processes
- relationships: is−a, part−of, regulates, ...

- terms are of varying specifity, hierarchical structure
- here: focus on biological process subgraph



Parent terms for a given GO term [15]

Derive a fixed-length **embedding** vector for each GO term s.t.

$$\text{semantic similarity of terms} \approx \text{cosine similarity of embeddings}$$

Advantages of embedding-based approach:

- Can provide embeddings directly as input feature
- Can handle complex species aliases by aggregating embeddings of contained species aliases
- GNNs could potentially learn complex patterns

node2vec

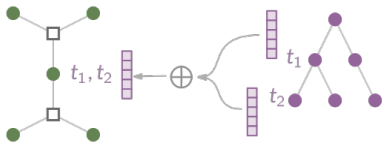
Given a simple non-attributed graph, find node embeddings s.t. for embeddings $\mathbf{z}_u, \mathbf{z}_v$ of nodes u, v

$$\mathbf{z}_u \cdot \mathbf{z}_v \approx P(\text{“}u, v \text{ co-occur on random walk”})$$

- $P(...)$ estimated by sampling random walks starting at each node
- Embeddings optimised via Gradient Descent

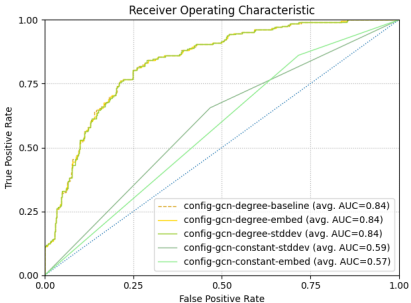
1. Extract identifiers from Disease Map
   UniProt for *AlzPathway*, Entrez Gene for *PDMap*
2. Obtain GO/BP graph
3. Map identifiers to GO terms
4. **Obtain embeddings for terms**
5. **Map embeddings to nodes**
   - Multiple terms for species: aggregate with *mean*
   - Complex species aliases: aggregate with *mean*
6. **Create node feature**

- use raw embeddings
- measure of spread of embeddings of neighbours

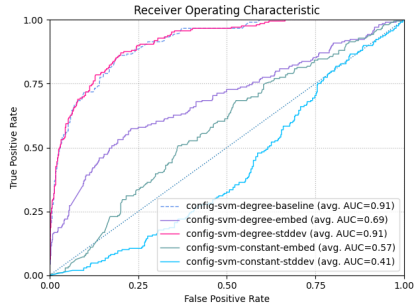


Use as single feature, or combined with degree.

- No performance gain; challenges
  - Quality of annotations
    large number of annotations for some species; specifity
  - Quality of embeddings
    node2vec hyperparameters



(a) Performance of the GNN classifier on different feature sets.



(b) Performance of the SVM classifier on different feature sets.

Objective 2

Determine number of duplicates and attachment of edges

- ⇝ *Partition* the neighbourhood of v.
  - Basic idea: In a good partition, each subset of neighbours should be homogeneous
- ⇝ Find a **clustering** of the neighbourhood

Requirements:

- Do not know number of clusters in advance
- Cannot assume convex clusters
- Assign all points to a cluster, no noise
- Instances can be of different scales
- Avoid parameters
- Never want single cluster or n clusters

Agglomerative Clustering

1. Initially, each point is assigned its own cluster
2. Iteratively, two clusters with smallest inter-cluster distance D are merged

$\rightsquigarrow$ clustering tree (**dendrogram**)

- For inter-cluster distance, use **single linkage**: $D(C_1, C_2) := \min_{p \in C_1, q \in C_2} d(p, q)$.
- Treshold on max. distance inside a cluster yields concrete clustering

How to find threshold?

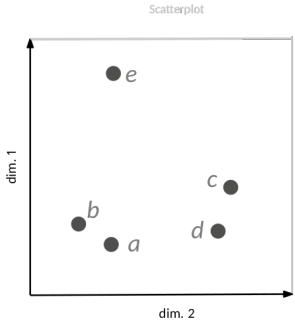
- Assume clusters are density-separated

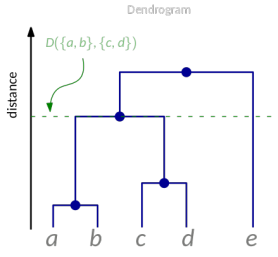
⇝ large "step" in distance between clusters

- Look for strongest increase in step size

What to use as point distance measure d?

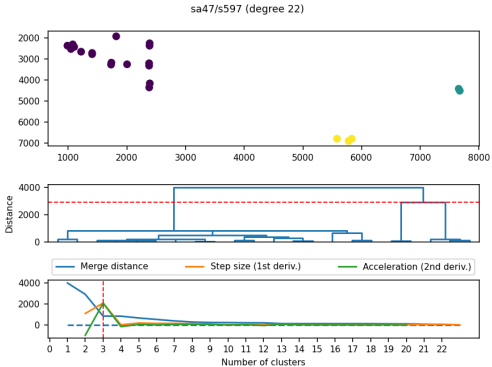
- Here: position in layout
  limited to *AlzPathway reorg. steps*

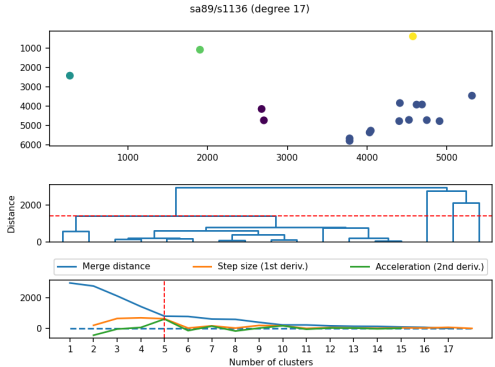




- Systematic evaluation of attachment is not trivial because edges may have been removed in reorg. step
- Look at examples instead

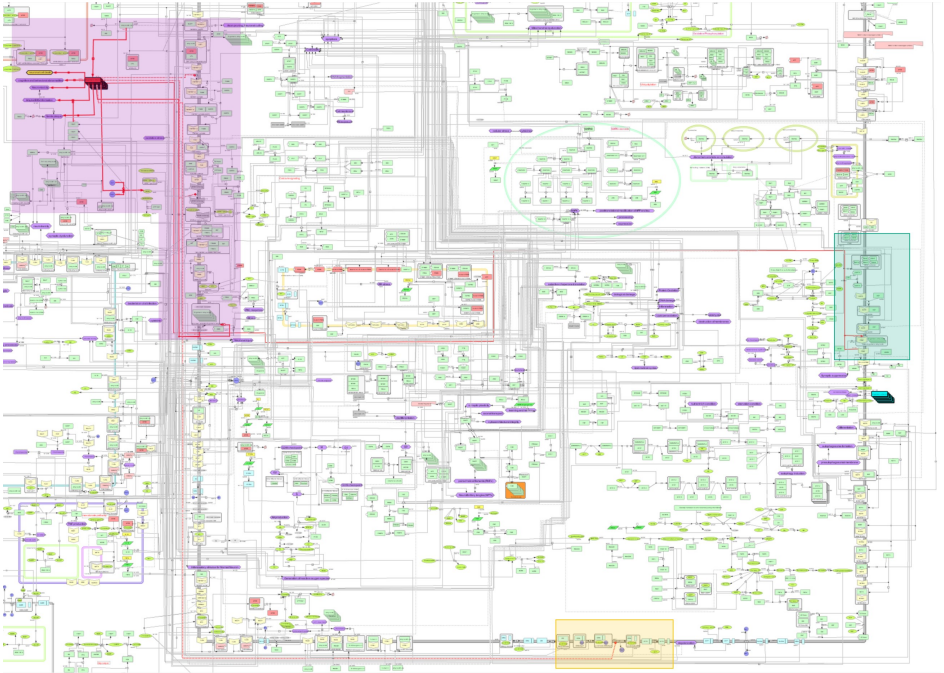
- Intuitive clusterings in almost all cases
- May provide useful suggestions

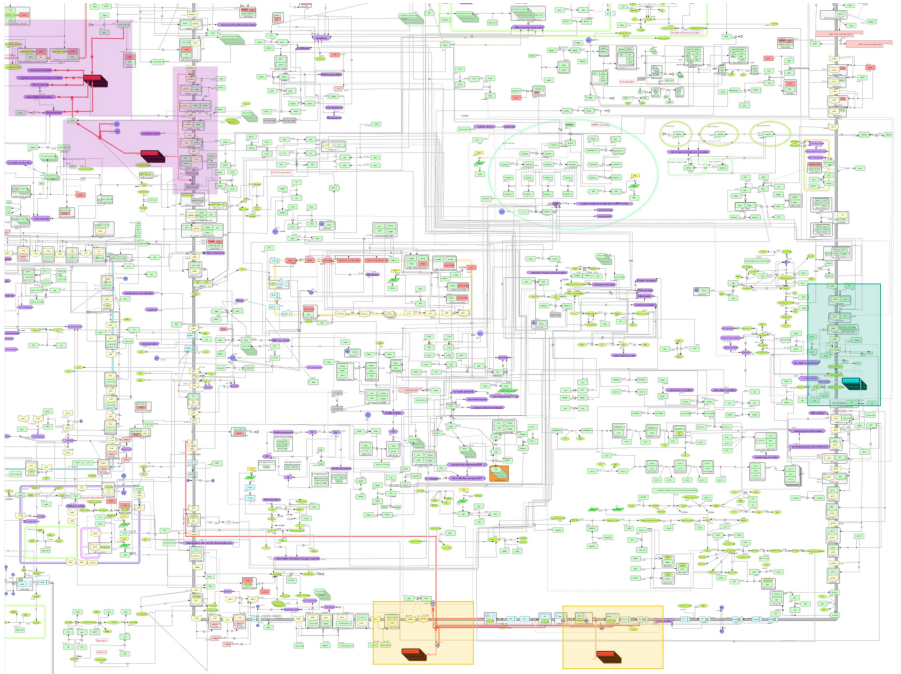




Compare suggestion to real reorganisation step

- Many other reorganisations in neighbourhood (nodes, edges added and removed)
- Neighbours reattached to other already existing alias of same species
- Here: more duplicates than indicated by clustering were introduced





Simplify machine learning task while maintaining same performance

- **reorganisation steps not essential**
  hard to obtain in practice
- need not compute structural features on both simple and bipartite graph
  substantial computational cost for large networks

Using directed degree as feature yields good performance

- ... but other structural features yet improve classifier performance slightly
- Supports previous work from a different angle: Degrees also useful to model given decisions of expert

NNs slightly outperform SVM classifiers

- Open how useful this advantage is in practice
- No gain through message-passing (in this implementation)

Embedding-based approach to encode **Gene Ontology term annotations as node features**

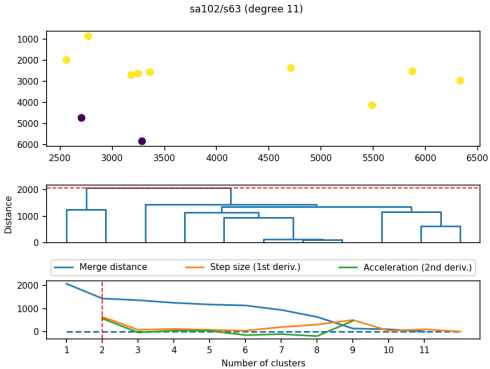
- Not useful in this implementation
- ...

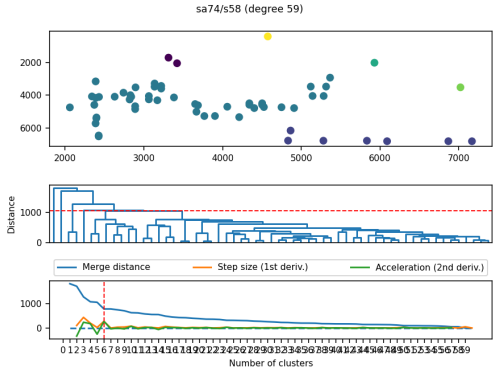
Suggestions on number of duplicates and edge attachment: **agglomerative clustering approach provides reasonable results**

Challenges

Thanks for listening

Additional examples for Clustering





[1] Marek Ostaszewski. *AlzPathway Regorganisation Steps for Increased Modularity*. Zenodo, June 25, 2021. DOI: 10.5281/zenodo.5032278. URL: https://zenodo.org/record/5032278 (visited on 06/28/2021).

[2] H. Ma and A. Zeng. "Reconstruction of Metabolic Networks from Genome Data and Analysis of Their Global Structure for Various Organisms". In: *Bioinformatics* (2003). URL: /paper/Reconstruction-of-metabolic-networks-from-genome-of-Ma-Zeng/2e645deb09836d6a1276959a7f8abd70820e63d9 (visited on 05/28/2021).

[3] S. Schuster et al. "Exploring the Pathway Structure of Metabolism: Decomposition into Subnetworks and Application to Mycoplasma Pneumoniae". In: *Bioinformatics* 18.2 (Feb. 1, 2002), pp. 351–361. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/18.2.351. URL: https://academic.oup.com/bioinformatics/article/18/2/351/225832 (visited on 11/20/2020).

[4] Ichcha Manipur et al. "Clustering Analysis of Tumor Metabolic Networks". In: *BMC Bioinformatics* 21.10 (Aug. 25, 2020), p. 349. ISSN: 1471-2105. DOI: 10.1186/s12859-020-03564-9. URL: https://doi.org/10.1186/s12859-020-03564-9 (visited on 01/06/2021).

[5] Mikael Huss and Petter Holme. "Currency and Commodity Metabolites: Their Identification and Relation to the Modularity of Metabolic Networks". In: *IET Systems Biology* 1.5 (Sept. 1, 2007), pp. 280–285. ISSN: 1751-8849, 1751-8857. DOI: 10.1049/iet-syb:20060077. arXiv: q-bio/0603038. URL: http://arxiv.org/abs/q-bio/0603038 (visited on 06/09/2021).

[6] Roger Guimerà and Luís A. Nunes Amaral. "Functional Cartography of Complex Metabolic Networks". In: *Nature* 433.7028 (7028 Feb. 2005), pp. 895–900. ISSN: 1476-4687. DOI: 10.1038/nature03288. URL: https://www.nature.com/articles/nature03288 (visited on 06/09/2021).

[7] Markus Rohrschneider et al. "A Novel Grid-Based Visualization Approach for Metabolic Networks with Advanced Focus&Context View". In: *Graph Drawing*. Ed. by David Eppstein and Emden R. Gansner. Red. by David Hutchison et al. Vol. 5849. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 268–279. ISBN: 978-3-642-11804-3. DOI: 10.1007/978-3-642-11805-0_26. URL: http://link.springer.com/10.1007/978-3-642-11805-0_26 (visited on 06/11/2021).

[8] G. Joshi-Tope et al. "Reactome: A Knowledgebase of Biological Pathways". In: *Nucleic Acids Research* 33 (suppl_1 Jan. 1, 2005), pp. D428–D432. ISSN: 0305-1048. DOI: 10.1093/nar/gki072. URL: https://doi.org/10.1093/nar/gki072 (visited on 10/13/2021).

[9] A. Lambert, J. Dubois, and R. Bourqui. "Pathway Preserving Representation of Metabolic Networks". In: *Computer Graphics Forum* 30.3 (June 2011), pp. 1021–1030. ISSN: 01677055. DOI: 10.1111/j.1467-8659.2011.01951.x. URL: http://doi.wiley.com/10.1111/j.1467-8659.2011.01951.x (visited on 06/09/2021).

[10] Alberto Noronha et al. "ReconMap: An Interactive Visualization of Human Metabolism". In: *Bioinformatics* 33.4 (Feb. 15, 2017), pp. 605–607. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw667. URL: https://doi.org/10.1093/bioinformatics/btw667 (visited on 07/09/2021).

[11] Ines Thiele et al. "A Community-Driven Global Reconstruction of Human Metabolism". In: *Nature Biotechnology* 31.5 (5 May 2013), pp. 419–425. ISSN: 1546-1696. DOI: 10.1038/nbt.2488. URL: https://www.nature.com/articles/nbt.2488 (visited on 10/18/2021).

[12] Sune S. Nielsen et al. "Machine Learning to Support the Presentation of Complex Pathway Graphs". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2019), pp. 1–1. ISSN: 1557-9964. DOI: 10.1109/TCBB.2019.2938501.

[13] Thomas N. Kipf. *Graph Convolutional Networks*. URL: http://tkipf.github.io/graph-convolutional-networks/ (visited on 12/07/2020).

[14] Petar Veličković et al. *Graph Attention Networks*. Feb. 4, 2018. arXiv: 1710.10903 [cs, stat]. URL: http://arxiv.org/abs/1710.10903 (visited on 01/05/2021).

[15] *Gene Ontology Overview*. Gene Ontology Resource. URL: http://geneontology.org/docs/ontology-documentation/ (visited on 11/22/2021).