Progress week 1

Jan van Eck

In this progress report

- Check if APID and HuRi data are correct
- Merge APID and HuRi
- Train/Validate/Test split
- Questions
- Supplements

APID PPIs

- Only Binary and no inter-species interactions
- PPIs: 66206
- Unique proteins: 13317

Example:

	InteractionID	UniprotID_A	UniprotName_A	GeneName_A	UniprotID_B	UniprotName_B	GeneName_B
0	1818	P54727	RD23B_HUMAN	RAD23B	P55036	PSMD4_HUMAN	PSMD4
1	1819	P55036	PSMD4_HUMAN	PSMD4	Q9UMX0	UBQL1_HUMAN	UBQLN1
2	1826	Q9UMX0	UBQL1_HUMAN	UBQLN1	Q16186	ADRM1_HUMAN	ADRM1

HuRi PPI (PSI File)

- Test results are filtered out
- Most important columns are:

UniProt ID A	UniProt ID B	Ensemble G/T/P id A	Ensemble G/T/P id B

- 58 Ensembl protein ids were found in the Uniprot columns

2	uniprotkb:Q13515	uniprotkb:Q9UJW9
3	uniprotkb:P30049	uniprotkb:Q05519-2
4	ensembl:ENSP00000462298.1	uniprotkb:P43220

- UniProt mapping tool could not map the ENSP ids to UniProtKB
- BioMart could map 34 ENSP to UniProtKB (24 proteins missing)

Difference between HuRi UniProt and ENSP ids

- A uniprot ID can have multiple ENSP ids
- The paper counts all the possible ENSP combinations as PPIs
- Paper claims 52569 PPIs
- The dataset contains combinations: 52549 PPI
  - missing 20 compared to paper

Example of multiple ENSP ids for one UniProtKB id:

UniProtKB A	UniProtKB A	Ensembl G/T/P A	Ensembl G/T/P B
O75920-1	A1L3X0	ensembl:ENST00000380750.7|ensembl:ENSP00000370126.3|ensembl:ENSG00000205572.9|ensembl:ENST00000354833.7|ensembl:ENSP00000346892.3|ensembl:ENSG00000172058.15	ensembl:ENST00000508821.5|ensembl:ENSP00000424123.1|ensembl:ENSG00000164181.13

HuRi PSI-MI file unique Gene combinations

- Website claims: 52548 ENSG combinations
- Again, a UniProtID can have multiple ENSG ids
- After creating all possible ENSG id combinations we also get 52548 gene combinations from the PSI-MI file (so this matches)

Unique proteins HuRi

- Paper claims: 8275 PPIs
- Using all ENSP ids in dataset: 8274 PPI
  - missing one protein compared to paper
- Using only UniProtKB A/B column: 8215 proteins
  - When removing entries with unmappable ENSP ids: 8184 (31 missing)
    - 7 extra proteins missing on top of the original 24 unmappable ENSPs ids because they only occured in combination with unmappable ENSP id.

Merge APID and HuRi

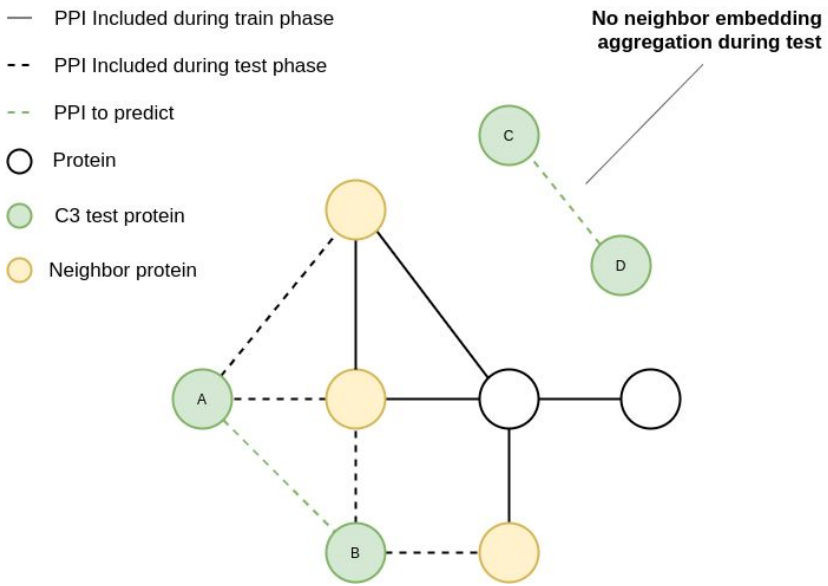
- ENSP to UniProtKB mappings gives worse result so we should work with UniProt ids
- After merging:
  - Unique proteins: 15364
  - PPIs: 111886
- What are we missing?
  - PPIs from HuRi: 553
  - Unique proteins from HuRi: 47
  - No missing data from APID
  - Main reason for missing data:
    - 24 ENSP ids unable to map to uniprot ids
    - 7 Uniprot ids which were only connected to one or more of the 24 unmappable ENSP ids
    - 16 Uniprot isoforms not found using the UniProtKB mapping tool

Train/Validate/Test split

- This paper shows that using the same train/validation/test splits of the same datasets, as well as making significant changes to the training procedure (e.g. early stopping criteria) precludes a fair comparison of different architectures:
  - https://arxiv.org/pdf/1811.05868.pdf
- They also show that simpler GNN architectures (GraphSAGE) are able to outperform the more sophisticated ones when they use 100 randomized train/val/test splits:
- Maybe we should think about creating multiple train/val/test splits as well

Train/Test set?

- Graph conv networks are based on neighbor embedding aggregations
- Can we include the edges of C3 (A,B) when testing while excluding all the edges of (A,B) during training?
- We need to think about the best GNN algorithm for C3. If we can only use (C,D) like network components for C3, basic GCN will probably not work.



Questions

- Can you check the code for merging APID and HuRi in this order?
  a. https://github.com/janvaneck1994/stage2/blob/master/Research/APID%20analyses.ipynb
  b. https://github.com/janvaneck1994/stage2/blob/master/Research/HuRi%20analyses.ipynb
  c. https://github.com/janvaneck1994/stage2/blob/master/Research/Merge%20APID%20and%20HuRi.ipynb
- Should we do something with the missing 47 proteins?
- Is there any code example for C1, C2, C3 splitting?
- Should we create multiple train/val/test splits (slide 9)?
- Can we include the edges of C3 when testing while excluding all the edges of C3 during training? (slide 10)

Theory

Usefull GNN methods for edge prediction:

- Graph Convolutional Networks from Kipf and Welling:
  - Semi-Supervised Classification with Graph Convolutional Networks
- GraphSAGE from Hamilton et al.:
  - Inductive Representation Learning on Large Graphs
- Graph Isomorphism Networks (GIN) from Xu et al.:
  - How Powerful are Graph Neural Networks?

Video lectures:

Graph Convolutional networks (17:13):

https://www.youtube.com/watch?v=7JELX6DiUxQ

Graphsage explanation (51:09):

https://www.youtube.com/watch?v=7JELX6DiUxQ

GIN explanation:

https://www.youtube.com/watch?v=H6oOhEIB3yE

Code examples

GCN PPI prediction using TensorFlow

http://snap.stanford.edu/deepnetbio-ismb/ipynb/Graph+Convolutional+Prediction+of+Protein+Interactions+in+Yeast.html

Graphsage implementation using PyTorch:

https://github.com/bkj/pytorch-graphsage
https://towardsdatascience.com/hands-on-graph-neural-networks-with-pytorch-pytorch-geometric-359487e221a8

PyTorch Geometric for GNN implementations:

https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.conv.SAGEConv

What to do next

- Set up a script to generate SeqVec embeddings
- Graph analyses of the new interactome
- C1, C2, C3 splits
- Reading literature

The future looks bright

- When using this implementation of a basic GCN on the HuRi PPIs (HuRi+APID gave GPU memory issues) without SeqVec embeddings, I get an AUC of 88 with a 2% test split which translates to a C1 test:
  - http://snap.stanford.edu/deepnetbio-ismb/ipynb/Graph+Convolutional+Prediction+of+Protein+Interactions+in+Yeast.html

```
Epoch: 0001 train_loss= 0.70523 val_roc= 0.89578 val_ap= 0.89326 time= 0.66723
Epoch: 0002 train_loss= 0.70508 val_roc= 0.90812 val_ap= 0.91111 time= 0.59767
Epoch: 0003 train_loss= 0.70411 val_roc= 0.89751 val_ap= 0.90599 time= 0.58281
Epoch: 0004 train_loss= 0.70116 val_roc= 0.88927 val_ap= 0.90289 time= 0.61928
Epoch: 0005 train_loss= 0.69527 val_roc= 0.88699 val_ap= 0.90228 time= 0.62831
Epoch: 0006 train_loss= 0.68536 val_roc= 0.88618 val_ap= 0.90197 time= 0.59233
Epoch: 0007 train_loss= 0.67035 val_roc= 0.88590 val_ap= 0.90183 time= 0.59302
Epoch: 0008 train_loss= 0.64790 val_roc= 0.88539 val_ap= 0.90149 time= 0.58597
Epoch: 0009 train_loss= 0.62557 val_roc= 0.88489 val_ap= 0.90129 time= 0.58480
Epoch: 0010 train_loss= 0.60149 val_roc= 0.88421 val_ap= 0.90100 time= 0.59716
Optimization Finished!
Test ROC score: 0.88143
Test AP score: 0.89842
```