

# Comp790-166: Computational Biology

## Lecture 22

April 22, 2021

# Announcements

- Homework 2 is online. Due April 23. <https://github.com/stanleyn/Comp790-166-Comp-Bio/tree/main/Homework2>
- Project Signup Sheet <https://docs.google.com/spreadsheets/d/1bSZByipJJxlRGXNqECi02VHxbS39WV6-cXL0akbA2kI/edit?usp=sharing>
- Comments on projects? How are they going?

# Reminder : Project Writeup Template

The project writeup is structured just like a typical conference-style research paper.

- PDF [https://github.com/stanleyn/Comp790-166-Comp-Bio/blob/main/Projects/Project\\_writeup.pdf](https://github.com/stanleyn/Comp790-166-Comp-Bio/blob/main/Projects/Project_writeup.pdf)
- TeX [https://github.com/stanleyn/Comp790-166-Comp-Bio/blob/main/Projects/Project\\_writeup.tex](https://github.com/stanleyn/Comp790-166-Comp-Bio/blob/main/Projects/Project_writeup.tex)

# Today

- Technical Writing in Computational Biology
- Representation Learning on Graphs in Biology to wrap up what we have seen

# Technical Writing Motivation

- We are all busy. Make your paper clear and easy to understand.
- In Comp Bio we write for two different audiences.
- Notation, figure presentation, publicly available code goes a long way.
- **Communication is Your Job!**
  - Good writing through simple language and organization
  - Well-documented publicly available code

# Question

What part of technical writing do you find the most challenging?

# Abstract: A Self Contained Story

- An elevator pitch of the main points
- Someone should read this and know exactly what your paper is about.
- Sentence breakdown
  - 1 sentence background
  - 1 sentence about what is still missing
  - 1 sentence about what you did
  - 1 sentence about what results suggest
  - 1 inspirational sentence about how this advances the field.

# Introduction

General sections of an introduction.

- Problem motivation- what are we even talking about?
- Description of previous approaches to the problem.
  - Always highlight the work of others in a positive way
- A paragraph where you compare and contrast previous solutions. You can still discuss limitations by spinning them in relation to all of the positive things that the other authors have done.
- Paragraph giving an overview of your contributions. Someone might only read this section of your paper. You need to sell your contribution in a human-readable way.



# Methods: First Defining Your Notation

- Notation needs to be clearly defined. There should never be a symbol in an equation that has not been properly defined.
- Keep bolding, italics, upper-case and lower-case consistent
- Dimensions of matrices need to be consistent represented with the same letter (usually  $p$ ,  $d$ , or  $m$ )
- Indices should always map the same thing throughout the paper (for example  $i$  referring to cells and  $j$  referring to a feature of a cell)

# Example Defining Notation

We start with some notation. We assume that we have an undirected graph  $G = (V, E)$ , where there are  $n = |V|$  nodes with features on each node represented by a matrix  $X \in \mathbb{R}^{n \times p}$ . Let  $A$  be the adjacency matrix of the graph,  $D$  be the diagonal degree matrix, and  $S$  be the normalized adjacency matrix  $D^{-1/2}AD^{-1/2}$ . For the prediction problem, the node set  $V$  is split into a disjoint set of unlabeled nodes  $U$  and labeled nodes  $L$ , which are subsets of the indices  $\{1, \dots, n\}$ . We will further split the labeled nodes into a training set  $L_t$  and validation set  $L_v$ . We represent the labels by a one-hot-encoding matrix  $Y \in \mathbb{R}^{n \times c}$ , where  $c$  is the number of classes (i.e.,  $Y_{ij} = 1$  if  $i \in L$  is known to be in class  $j$ , and 0 otherwise, where the  $i$ th row of  $Y$  is all zero if  $i \in U$ ). Our problem is transductive node classification: assign each node  $j \in U$  a label in  $\{1, \dots, c\}$ , given  $G$ ,  $X$ , and  $Y$ .

Figure: from Huang *et al.* ICLR 2021.

# Methods : Problem Formulation

- A section where you mathematically define your problem with the notation you introduced.
- What are your inputs and outputs? What are the dimensions of the inputs and outputs and what do they represent?
- Even if you write out your problem in text format, reference the variables that you defined in the text.

For example: 'For each cell,  $\mathbf{x}_i \in \mathbb{R}^d$ , we wish to learn its label,  $y_i$  through the use of the graph,  $\mathcal{G}$ .

## Tip: Give Reminders

- It is good to keep reminding readers what notations and abstractions represent.
- For example, defining a graph? It doesn't hurt to remind them that nodes are cells and edges represent sufficient similarity between cells.
- Connect problem formulation to 'Figure 1'. In defining the overview of your problem, reference sub-panels of figure 1 of interest.

# Example of a Comprehensive Figure 1

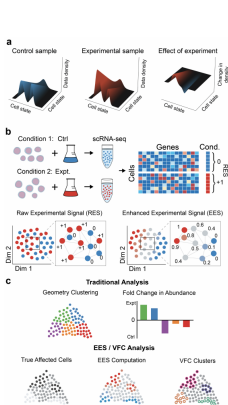


Figure: from Burkhardt *et al.* Nature Biotech. 2021.

# Schematic Illustrations

If you draw cells, or patients, make sure these are carried through the entire figure.

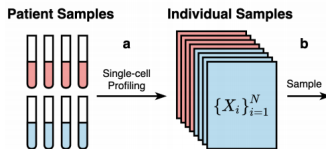


Figure: from Haidong Yi. <https://www.biorxiv.org/content/biorxiv/early/2021/04/14/2021.04.13.439702.full.pdf>

# Pseudo-Code

Writing good pseudo code is extremely helpful. It can often be more helpful than the entire methods section.

---

**Algorithm 2** xNetMF ( $G_1, G_2, p, K, \gamma_s, \gamma_a$ )

---

```
1: ===== STEP 1. Node Identity Extraction =====
2: for node  $u$  in  $\mathcal{V}_1 \cup \mathcal{V}_2$  do
3:   for hop  $k$  up to  $K$  do       $\triangleright$  counts of node degrees of  $k$ -hop neighbors of  $u$ 
4:      $d_u^k = \text{CountDegreeDistributions}(\mathcal{R}_u^k)$        $\triangleright 1 \leq K \leq \text{graph diameter}$ 
5:   end for
6:    $d_u = \sum_{k=1}^K \delta^{k-1} d_u^k$        $\triangleright$  discount factor  $\delta \in (0, 1]$ 
7: end for

8: ===== STEP 2. Efficient Similarity-based Representation =====
9: ===== STEP 2a. Reduced  $n \times p$  Similarity Computation =====
10:  $\mathcal{L} = \text{ChooseLandmarks}(G_1, G_2, p)$        $\triangleright$  choose  $p$  nodes from  $G_1, G_2$ 
11: for node  $u$  in  $\mathcal{V}$  do
12:   for node  $v$  in  $\mathcal{L}$  do
13:      $c_{uv} = e^{-\gamma_s \cdot \|d_u - d_v\|_2^2 - \gamma_a \cdot \text{dist}(f_u, f_v)}$ 
14:   end for
15: end for       $\triangleright$  Used in low-rank approx. of similarity graph (not constructed)

16: ===== STEP 2b. From Similarity to Representation =====
17:  $W = C[\mathcal{L}, \mathcal{L}]$        $\triangleright$  Rows of  $C$  corresponding to landmark nodes
18:  $[U, \Sigma, V] = \text{SVD}(W^\dagger)$ 
19:  $\tilde{Y} = C U \Sigma^{-\frac{1}{2}}$        $\triangleright$  Embedding: implicit factorization of similarity graph
20:  $\tilde{Y} = \text{Normalize}(\tilde{Y})$        $\triangleright$  Postprocessing: make embeddings have magnitude 1
21:  $\tilde{Y}_1, \tilde{Y}_2 = \text{Split}(\tilde{Y})$        $\triangleright$  Separate representations for nodes in  $G_1, G_2$ 
22: return  $\tilde{Y}_1, \tilde{Y}_2$ 
```

---

Figure: from <https://arxiv.org/pdf/1802.06257.pdf>

# Results

- Figure/table legends should be self-contained. For example, if there is some kind of confidence interval around your curve, tell us what it represents
- Plotting: try to choose appropriate axis to capture all of the datapoints. Don't just plot for example between 0 and 1 on the y-axis by default.
- Make sure that each panel of your results figures are clearly referenced in the text.
- Avoid sloppiness. Don't let a table flow over the margin. Try to avoid different fonts and font sizes between figures.
- Colors: choose them well. Try changing default colors and removing grids from plots, etc.



# Information to Include in Results

- **Baselines:** How were the baseline methods used? Did you use default parameters?
- (In real life...) you should be testing your method on several datasets (3 in biology is good).
- **Dataset description:** Describe these datasets, any pre-processing you did, and where the information can be accessed.
- **Description of Experiments:** Experiments need to be clearly described, including small details like the number of times you repeated such experiment. Always reference the figure or table where the results appear wrt a given experiment.

# Discussion

- Recap what you have done with an overall summary
- Explain how your work complements or addresses some unmet need in the field
- Summarize your results again
- Discuss limitations and future work
- **Inspirational Parting Thought:** What is the main reason people should care and why does your work advance the field?

# Publishing in Comp Bio

- Conferences
  - ISMB
  - RECOMB
  - ACM BCB
- Journals
  - Bioinformatics
  - Cell Systems
  - Nature Journals (Nature Methods, Nature Biotech, Nature Communications)

# Writing a Conference Paper

- Self-contained, well-structured, making it easy to read and write
- Much faster in terms of review, revision
- Appealing to CS audience.

# Writing a Journal Article

- The main text is selling an algorithm to a broad audience.
- Heavily relies on supplemental text to get all of the relevant details.
- Very slow process. From initial submission to publication can take 1 year.
- Not as appealing to a CS audience.
- More appealing to biology audience.
- Very expensive to publish

# Providing Code

- It is good to provide code with your paper starting at the time of submission
- Repository should contain a pre-processed version of the data and instructions about how to run code on these data.

# From the Point of View of a Paper Consumer

- It is great to publish in fancy interdisciplinary journals
- It becomes less valuable to us on the CS side if the method is scattered over 100 pages of supplement
- Writing a version of your paper with all of the technical details for ArXiv is very good practice.

# A Word of Advice for Being a PhD Student in Comp Bio

Protect your expertise and your time. You are not a core facility.

- Prioritize collaborations that are mutually beneficial
- Make sure you publish your own papers without too many distractions of analyzing random datasets.
- Check where your potential collaborators put their comp bio people in the author list.



# Communicating Between Fields

- People will care about different things, between biology and computer science- tailor your details accordingly.
- You need to translate your complex model to a series of steps that don't involve mathematical phrases that we all take for granted. For example, don't say phrases like 'L1 penalty'

# Choosing What to Work On

Inspired by the talk of Quaid Morris

<https://www.youtube.com/watch?v=xueh6WnpRDQ>

- Don't be the state-of-the-art, be the benchmark (aka ask a new question)
- Choose hard problems rooted in biology that other people wouldn't have thought to ask because they don't read the biological literature.
- Watch the superstars who speak both languages. Watch how they publish and what they choose to work on.

# Transitioning and Summarizing What we Have Covered

We have focused on representing data as graphs and using the graphs to help us to answer questions.

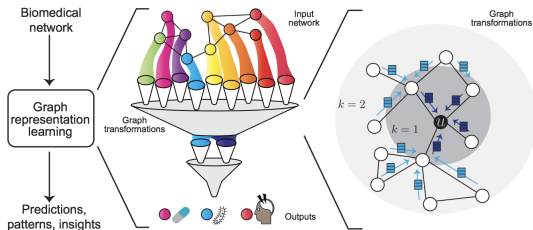
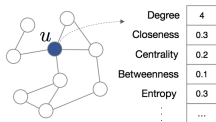


Figure: From <https://arxiv.org/abs/2104.04883>. For example. Assigning proteins to groups or people to outcomes.

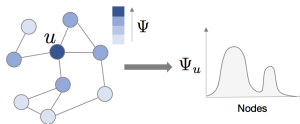
# Class 1: Graph Summary Statistics and Diffusion

Summary statistics and diffusion can describe patterns in the graph, importance of nodes,

**a** Graph theoretic techniques



**b** Random walks and diffusion

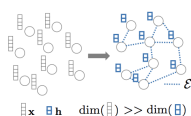


# Graph Structure and Diffusion and Papers

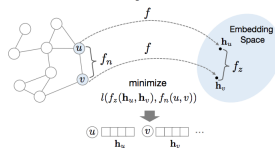
- **PhenoGraph:** Partition cells to cell clusters
- **BigClam:** For overlapping clustering
- **MAGIC:** for imputation in single cell data.
- **MELD:** for predicting the specificity of each cell to each condition.
- **Conos:** Combining multiple single cell datasets
- **REGAL:** graph alignment based on structural properties
- **Edge-Based Clustering:** Studying connectivity in the brain

# Class 2: Manifold Learning and Node Embedding

e Manifold learning



f Shallow network embeddings

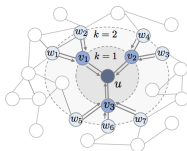


# Manifold Learning and Node Embedding Papers

- Node2Vec for node embedding (embedding)
- SUGAR for data augmentation in single-cell analysis (manifold)
- SLICER for trajectory inference (manifold)
- Grassmann Embedding for combining multiple datasets (manifold)
- Mashup for embedding nodes according to multiple relational definitions (embedding)

# Class 3: Machine Learning on Graphs

g Graph neural networks



h Graph generative models

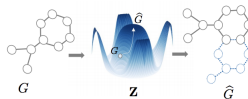


Figure: We haven't seen so much here.....



# Seen in ML on Graphs

- Correct and Smooth for predicting labels of nodes.
- A quarter lecture on another GNN topic next week?