Week 4: Working with network data (and an intro to Gephi)

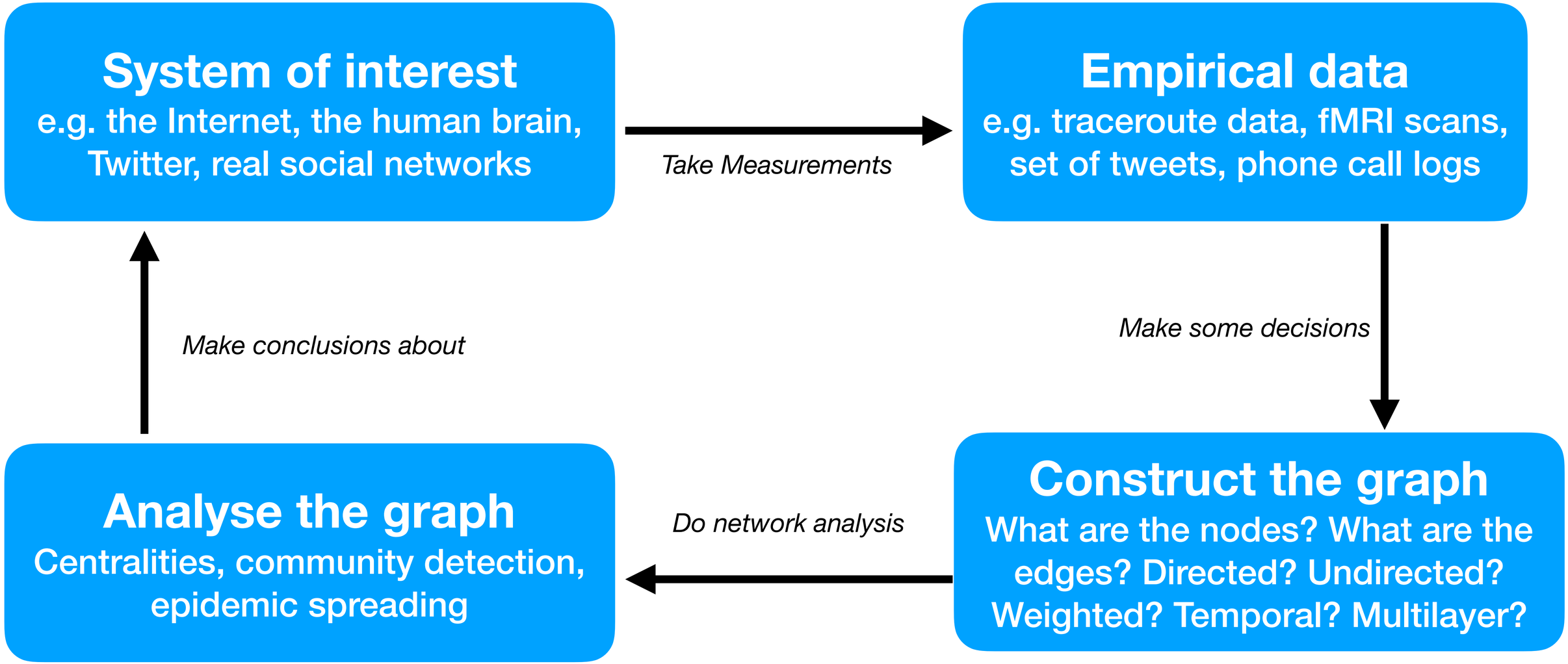
Naomi Arnold
https://narnolddd.github.io/

Morning all! We will start 9:05 as usual :-)

Tutorial aims

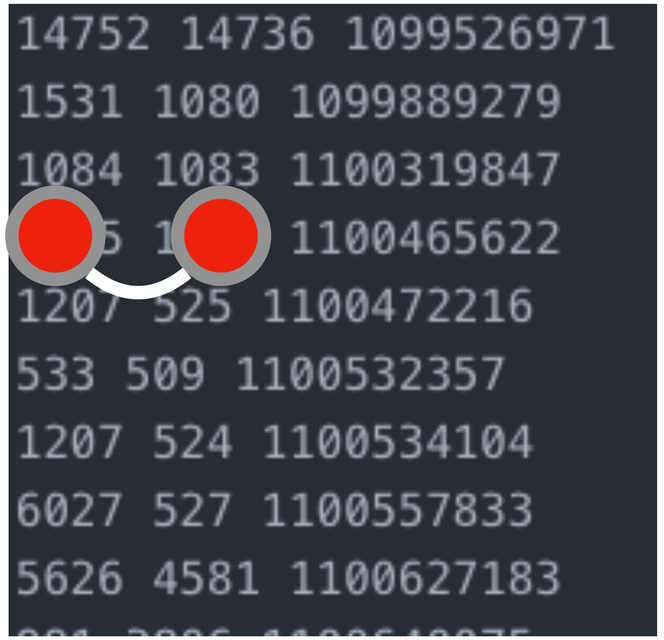
- Understand some of the **research design decisions** in constructing a network from data
- Understand the components of a meaningful **network visualisation**
- Lord of the Rings **Gephi demo**
- Coursework topic **Q&A session**

Studying systems as networks



Data Types: Edge List

- Usually a list with each row having **2 or more** comma/tab separated values
- First two values are **source** and **destination** nodes of edge
- Any extra values are **metadata** e.g. timestamp, edge weight

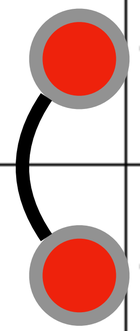


e.g. Facebook wall post dataset has user ids of wall poster and postee resp., and UNIX timestamp of when post was created

[Bimal Viswanath et al On the evolution of user interaction in Facebook. In *Proc. Workshop on Online Social Networks*, pages 37--42, 2009.]

Data types: Databases

eventName	eventSec	id	matchId	matchPeriod	playerId	positions	subEventId	subEventName	tags
Pass	1.656214	258612104	2057954	1H	122671	[{'y': 50, 'x': 50}, {'y': 53, 'x': 35}]	85	Simple pass	[{'id': 1801}]
Pass	4.487814	258612106	2057954	1H	139393	[{'y': 53, 'x': 35}, {'y': 19, 'x': 75}]	83	High pass	[{'id': 1801}]
Duel	5.937411	258612077	2057954	1H	103668	[{'y': 81, 'x': 25}, {'y': 83, 'x': 37}]	10	Air duel	[{'id': 703}, {'id': 1801}]



More complicated example: FIFA dataset where edges could be passes, specific types of passes, tackles etc.

[Luca Pappalardo et al, A public data set of spatio-temporal match events in soccer competitions, 2018, Nature]

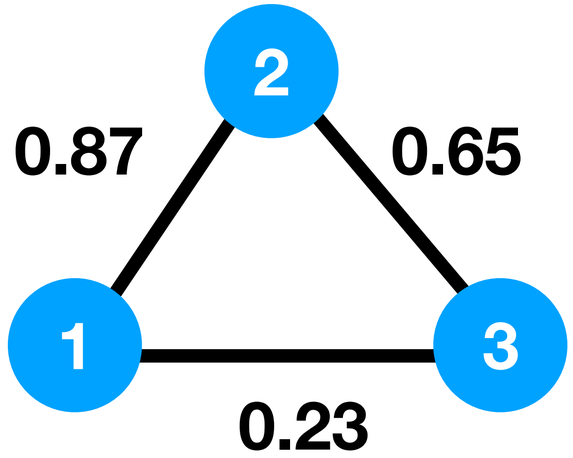
Data Types: Adjacency/Weight Matrix

Dest Node

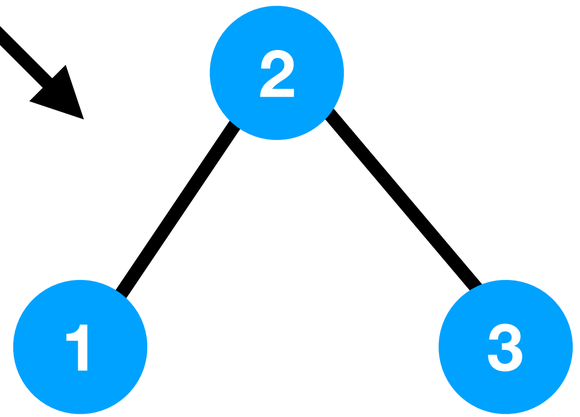
Source Node

	1	2	3
1	0	0.87	0.23
2	0.87	0	0.65
3	0.23	0.65	0

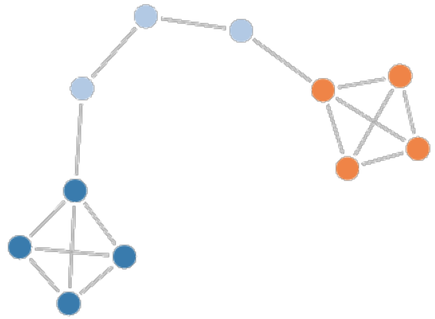
Value at **row i** and **column j** is the weight between node i and j



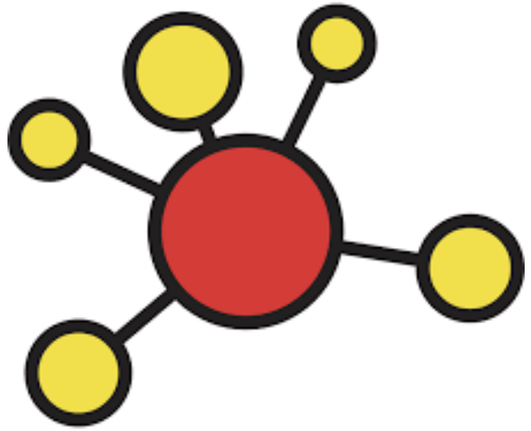
Threshold 0.6



Software/libraries for network visualisation



NetworkX



(iGraph)

Gephi

www.menti.com

40 86 42 7

Aaagh, my network is too big — thinning

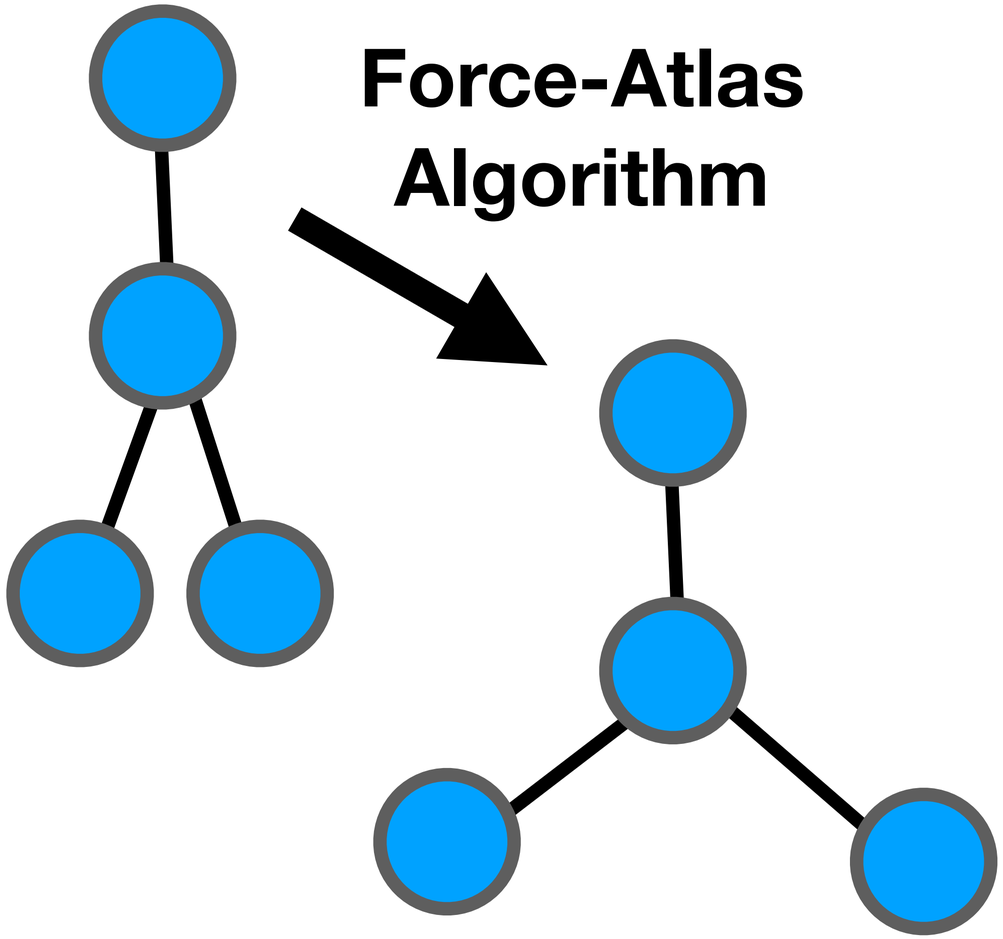
- **Windowing** — if the edges in your dataset have timestamps, look only at edges within a certain time window.
- **Random edge sample** (requires preprocessing in e.g. Python) — take a random sample of the edges in the dataset.
- **Random node sample** — take the network you get from a random sample of the nodes
- **Degree filter** — only include nodes of above a certain degree (e.g. get rid of nodes of degree 0,1)

Network visualisation ingredients

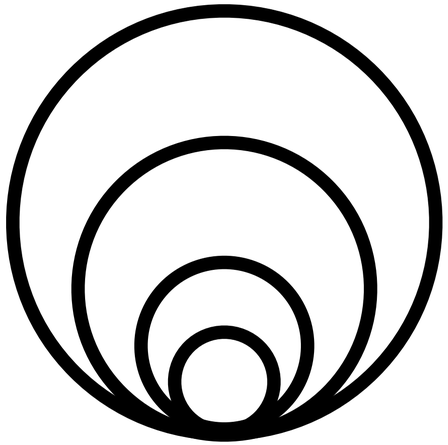
- **Node positions** — where the nodes are placed in the space?
- **Node features** — size/colour/shape?
- **Edge features** — thickness, colour, shape?

Node Positions

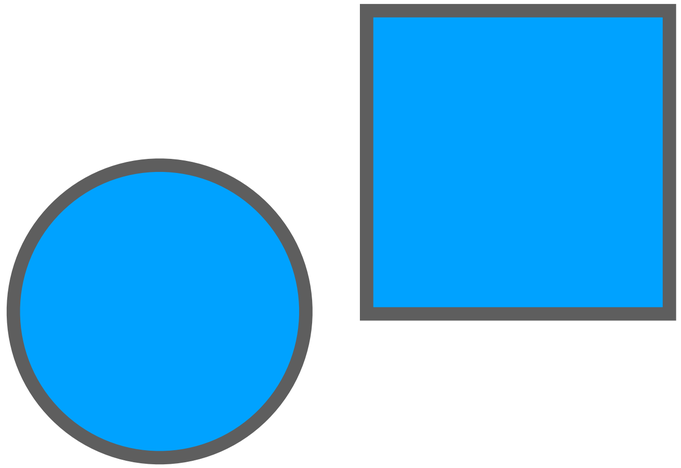
- Want to minimise **edge crossover**
- Put nodes that are **close to each other** by hops close to each other in the space
- **Layout algorithms** help with this (but are not perfect)



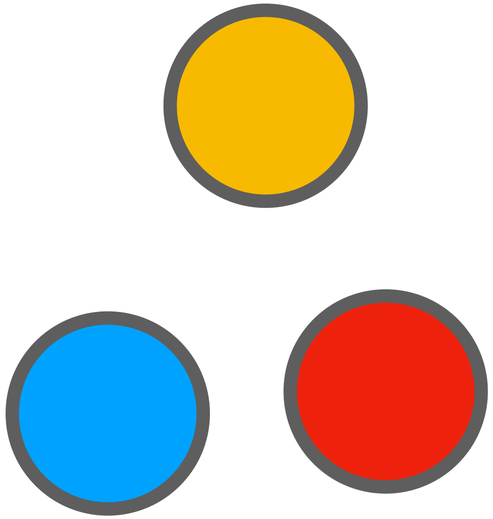
Node Properties



Size (usually some centrality measure)



Shape (different type of nodes in the graph?)

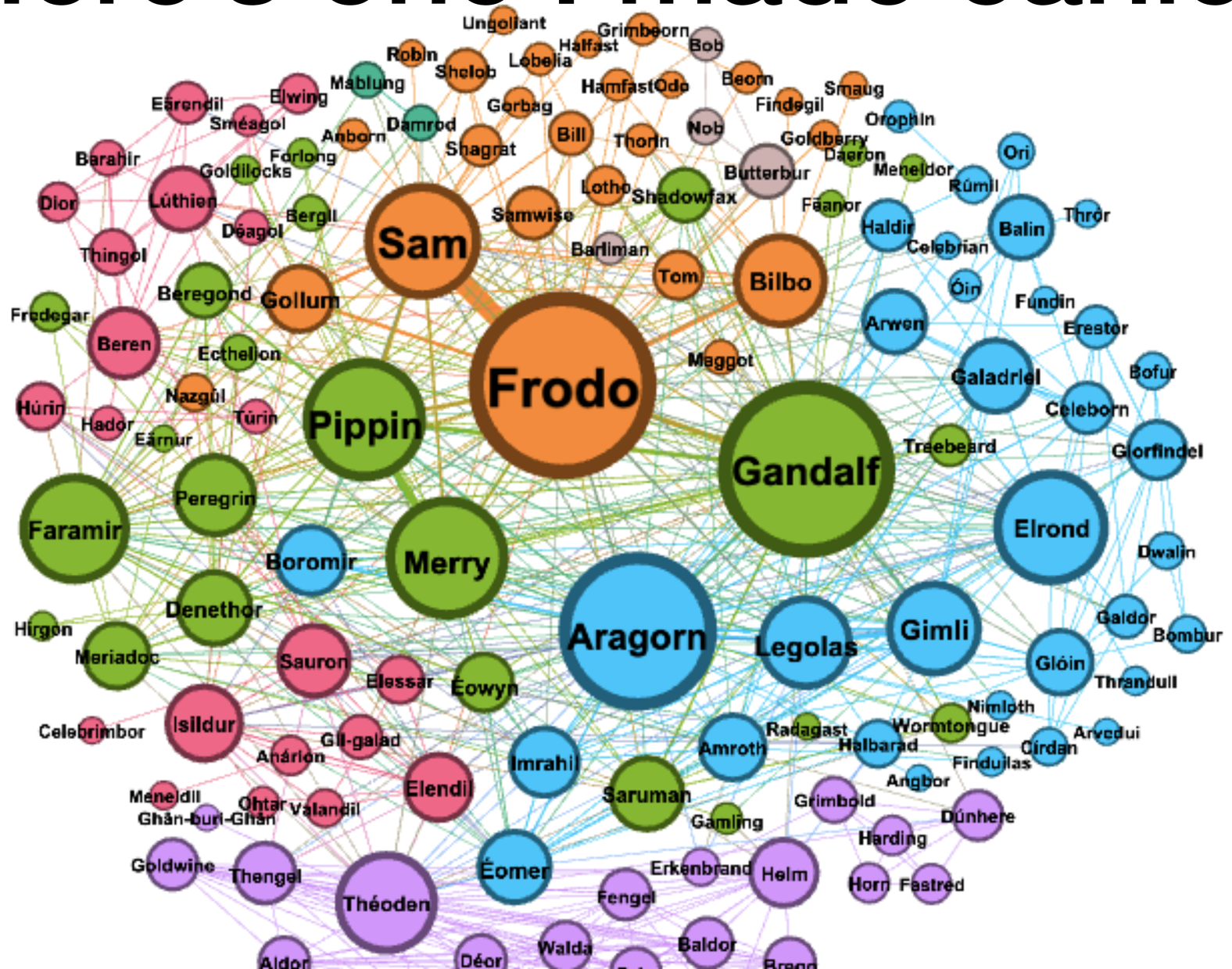


Colour (usually community-related)

Rest of tutorial: LOTR Gephi Demo

Dataset of LOTR character cooccurrences (occurrences within the same sentence)
https://github.com/Raphtory/Examples/blob/main/src/main/scala/examples/lotr/lotr.csv

Here’s one I made earlier



Takeaways

- **Lots of decisions** to make when constructing a network — think and brainstorm before importing to Gephi
- Some **preprocessing** may be needed for visualisations to be meaningful
- Visualisation is an **iterative process**, lots of fun to be had with it
- **Many tutorials available** for Gephi and its different features to get started

Thanks for listening!