

Advertising feature

Raphtory for your network analysis needs

Richard G. Clegg, Queen Mary University of London
(But most of the work by many collaborators, Naomi Arnold, Cheick Ba, Ben Steer, Peijie Zhong, Raul Mondragon, many others)



Aim of this talk

- Evangelise Raptory temporal network software.
- Convince you to use Raptory in your research.
- Convince you to **credit** Raptory in your research.
- Recent publications using it (just from my group):
 - Transactions on Web, ICWSM, Nature Scientific Reports (x3), Proc ACM HCI (plus the usual workshops etc)

Obscure reference

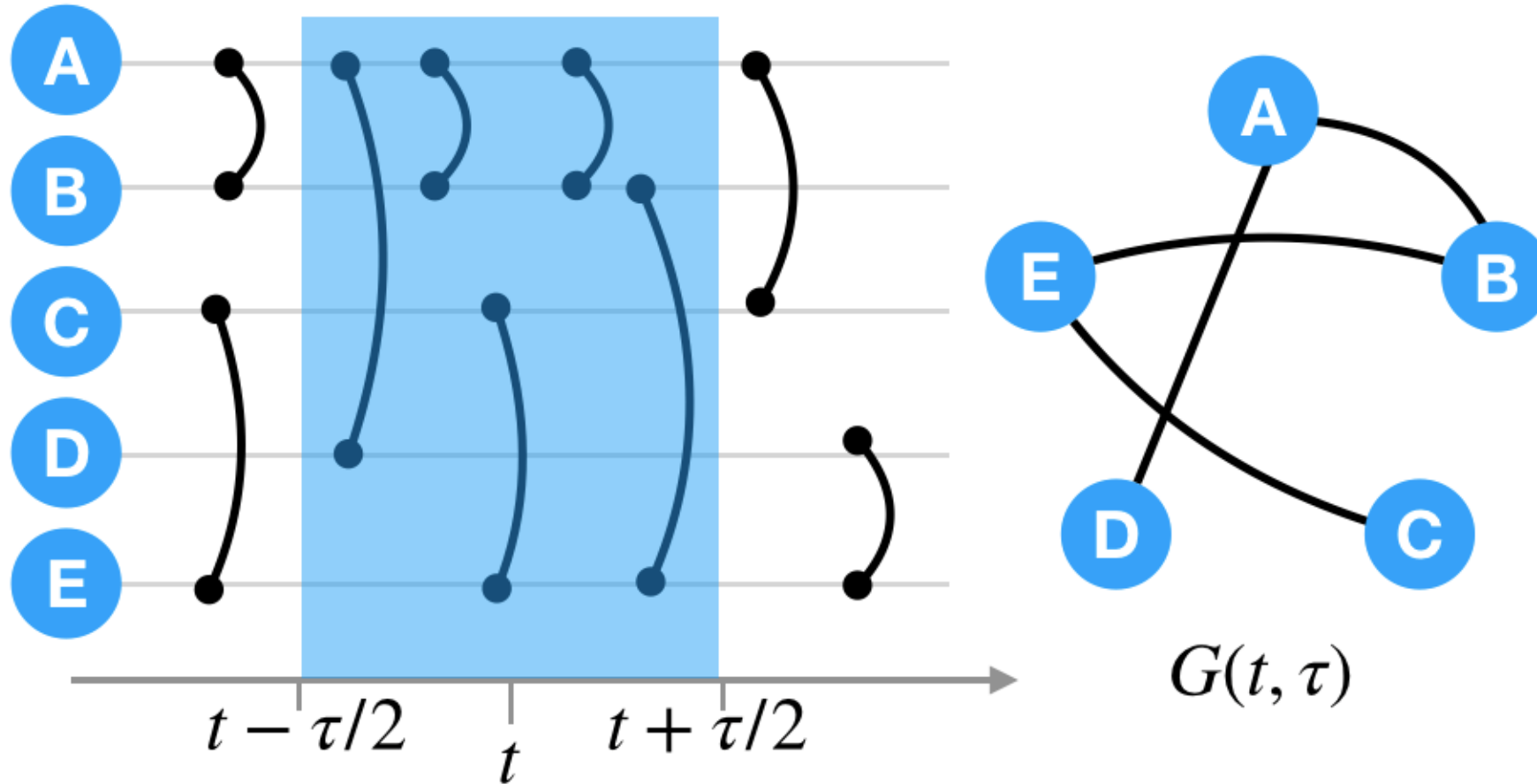


What is Raphtory anyway?

- Analyses network datasets (particularly temporal).
- Easy and quick: python API frontend, highly optimised Rust backend.
- Basic queries are easy:
 - Find all shortest paths from A to any destination node.
 - Run “page rank” to find the most important node.
- Time is included in a structured way from the start:
 - Find me a “time respecting” path leaving at 15:01.
 - Show me congestion levels on links averaged every minute.

Temporal networks

Temporal window centred on t , size τ



Some history



pommetry

- Designed at QMUL (PhD project Ben Steer 2017).
- Developed by spin out Pometry (2021-now).
- Seed funding and now 20 employees.
- I have no commercial interest (this is not a **sponsored** ad).
- Faster/more general than older networkx/graphtool/neo4j.
- Written from the ground up to understand time in networks.
- Current implementation by real software engineers (not one line written by me, guaranteed!)

Is it simple to use?

Install is “pip install raphtory”

```
import raphtory as rp
import pandas as pd
df = pd.read_csv("lotr.csv").reset_index()
lotr_graph = Graph()
lotr_graph.load_edges (df,
                       time="index", src="v", dst="w")
```

Supports common data ingestion formats.

Tell it which parts of the data are nodes/times.

This is “Lord of the Rings” character interactions.

Is it simple to use?

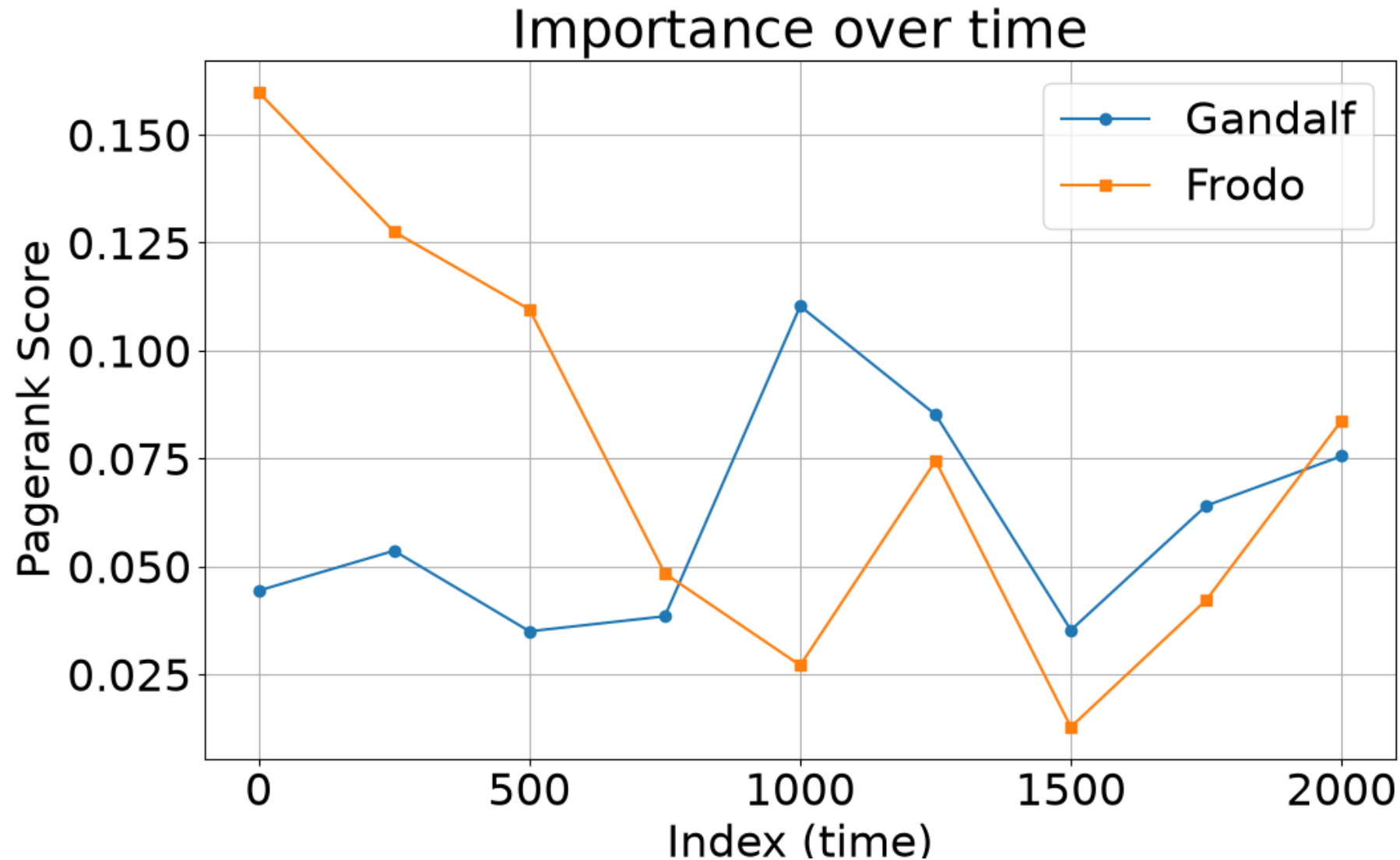
```
fimportance = []  
for win_graph in lotr_graph.rolling(window=250):  
    result=rp.algorithms.pagerank(win_graph)  
    fimportance.append(result.get("Frodo"))
```

Split graph into time windows

Loop over the graph doing “page rank”

Collect the page rank of the node called “Frodo”

Gandalf vs Frodo importance throwdown



Is it fast? (Let's play with netflow data?)

- 2.8TB of raw netflow CSV files totalling 20 billion rows.
- **Machine spec (1 x c5id.24xlarge):** 128GB of RAM, 24 cores
- Ingests in 26 minutes. PageRank 74ms. Connected components in 546ms.

Complex Query

- Spark 30x(4core 30GB) machines:
5% of data in 3.5 hrs
- RocketGraph (3.5 TB RAM):
5% of data in 27 seconds
100% of data timed out after 1.5 hrs.
- Raphtory (128GB RAM)
100% of data in 4.2 seconds

```
MATCH
  (E)←[nf1:Netflow]-(B)←[login1:Events2v]-(A), (B)←[prog1:Events1v]-(B)
WHERE A <> B AND B <> E AND A <> E
AND login1.eventID = 4624
AND prog1.eventID = 4688
AND nf1.dstBytes > 100000000
AND login1.epochtime < prog1.epochtime
AND prog1.epochtime < nf1.epochtime
AND nf1.epochtime - login1.epochtime ≤ 30
RETURN count(*) here
```

My favourite Raphtory discoveries

- Network: Luna/Terra crypto system. Discovery: Wallets responsible for the collapse of the \$3.5bn ecosystem (coordinated short selling).
- Network: alt-right social network gab. Discovery: the largest connected component shatters at quiet hours and reforms at busy hours as the English-speaking world wakes/sleeps.
- Network: words co-located in sentences. Discovery: a community of words that all describe rare cheeses.
- Network: Companies House directors. Discovery: A disused basement in Harley Street home to several hundred “shelf companies” analysing

Find out more

<https://raphtory.com>



Youtube: computerphile
Me rambling about it at length

