ETH zürich

Chair of Systems Design
www.sg.ethz.ch

Modeling time in interactions

Temporal paths in a 16thc. letter network

Ramona Roller
April 15, 2021

Question of today

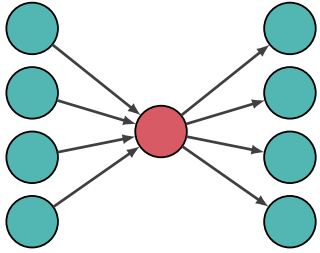
Who drives the spread of ideas?

My first approach

RQ: Who drove the spread of ideas during the Reformation?

- ☑ **Domain knowledge**
  - Reformation: division of Catholic Church
- ☑ **Data**
  - 26,663 letters; 3,348 reformers
- ☑ **Network**
  - nodes: reformers; edges: letters
- ☑ **Operationalisation**
  - 'driver for spread of ideas': betweenness centrality
  - letters reformer A passes on from B to C

- ➔ **Melanchthon: highest betweenness centrality**
- ➔ **Melanchthon: most important driver for idea spread**

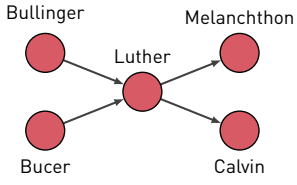


High betweenness
Low betweenness

Melanchthon

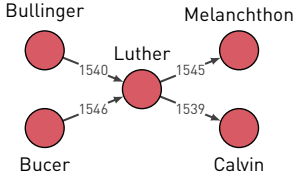
The problem

Time is ignored



betweenness$_{\text{Luther}}$ = 4

Time is considered



betweenness$_{\text{Luther}}$ = 1

- **Transitivity assumption:** A influences B independent of A's prior influences

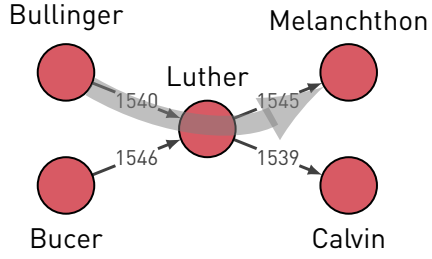
➔ **Information can spread backwards in time**

➔ **Network statistics in aggregated networks are flawed**

Time matters

- **Interactions and relations are time-stamped**
  - Communication: Sending date of letter
  - Co-location: Meeting date
  - Trade: Transaction date
  - ...
  - ➔ **Temporal ordering of edges influences causal relationships**

- **What we usually do**
  - Ignore time
  - Analyse aggregated networks
- **What we could do instead**
  - Analyse paths
  - Use `pathpy` (Scholtes 2017)



- ➔ **Today: How to use `pathpy` for historical networks**
- ➔ **Example: letter correspondence network of the Reformation**

The Reformation and the letter correspondence network

- **Reformation:** transformative movement of society in early modern Europe
- **Letters:** Main means of communication, insights into social processes
- **Data:** 9 letter editions
  Luther, Melanchthon, Zwingli, Bullinger, Karlstadt, Bucer, Myconius, Oekolampad, Vadian
- **Network:** nodes=reformers, edges=letters
- **Assumption:** Letters convey ideas


Luther


Melanchthon


Karlstadt


Bucer


Zwingli


Bullinger


Vadian


Myconius


Oekolampad

Setup and data preparation

- **Install and import `pathpy`**

```
pip install pathpy
import pathpy as pp
```

- **Prepare time stamps of edges**
  - Unix time stamps do not cover dates before January, 1st 1970
  - Compute time differences for each edge i: earliest sending date - sending date$_i$

- **Prepare data input table**

source	target	time
879	3970	0
81	3010	615
81	3010	1151
81	2197	2656
81	3010	2663

  - **row:** edge in network
  - **source:** sender of letter
  - **target:** receiver of letter
  - **time**: number of days from earliest sending date to sending date of this row

- **Optional: Dictionary to map node index to node name for visualisation**

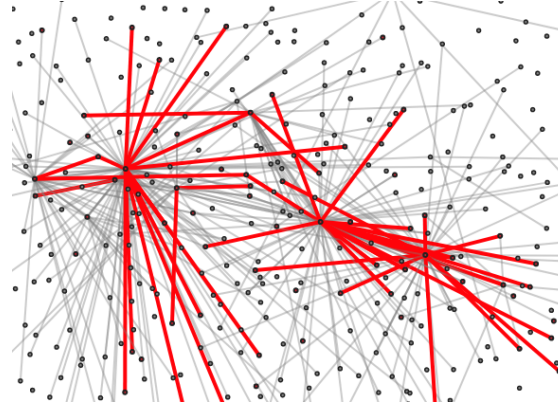
```
node_labels = {81:'luther;martin', 85:'melanchthon;philipp', 31:'bullinger;heinrich', ...}
```

Compute temporal network

Temporal network: Registers interactions per time point

```
ref_tempNet = pp.TemporalNetwork.read_file('input.csv', separator=',', directed=True)
```

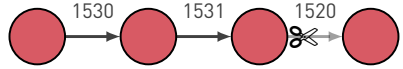
- Nodes: 3348
- Time-stamped links: 30043
- Links/Nodes: 8.97
  9 times more links than nodes
- Observation period: [0, 23550]
  1500-01-01 – 1564-06-24
- Observation length: 23550
  64.5 years
- Time stamps: 10960
  number of unique sending dates
- Avg. inter-event dt: 2.15
  days between sending of two letters
- Min/Max inter-event dt: 1/1505
  1 day/4.12 years



➔ **Dynamic visualisation of temporal network shows temporal ordering of edges**

Compute paths

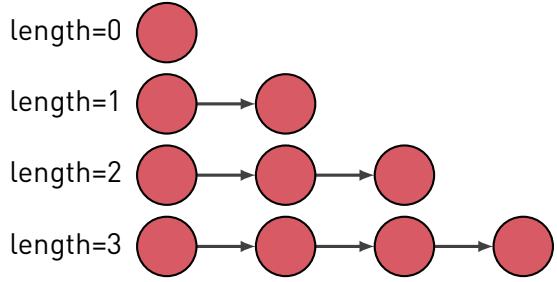
Paths are cut if temporal order is broken



Path

- Sequence of nodes that are connected by time-consecutive edges

Paths have different lengths




```
delta_t = 14 # in days
ref_paths = pp.path_extraction.paths_from_temporal_network(ref_tempNet, delta=delta_t)
```

Paths preserve temporal ordering of edges

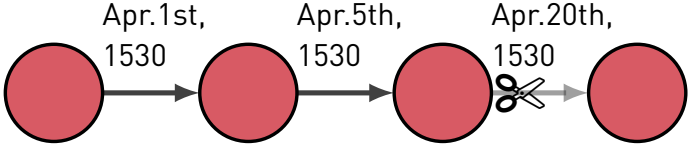
Define δt

δt

- Maximum allowed time difference of two consecutive edges in a path
- Accounts for forgetting ideas after a long time lag
- The larger δt the longer the obtained paths

Time difference $> \delta t \Rightarrow$ cut path

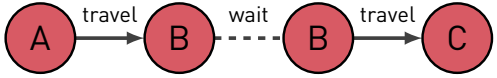
Assume $\delta t = 14$ days



δt cuts path if time lag between two sending dates is too long

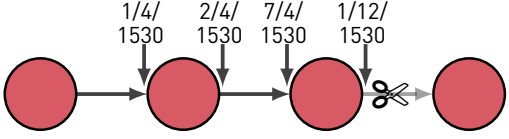
Select δt

- Depends on data
- δt = **travel + appropriate waiting time**
  - Travel time $\approx$ walking time
    mean: 35 h, SD: 29 h
  - Appropriate waiting time: 2 weeks
  - ➔ $\delta t = 14$ **days**



- ➔ **Caution: 1% reformers and 24% links used**
- ➔ `pathpy` **does not detect time scales**
  ⇒ use your brain/theory to select δt

Long waiting time ⇒ cut path



Betweenness centrality ⇒ drivers of spread of ideas

RQ: Who drove the spread of ideas during the Reformation?

```
# Aggregated network
ref_agg = pp.HigherOrderNetwork(ref_paths, k=1)
# Betweenness centralities for aggregated network and paths
btw_agg = pp.algorithms.centralities.betweenness(ref_agg)
btw_path = pp.algorithms.centralities.betweenness(ref_paths)
```

Aggregated betweenness

1. Melanchthon: 392,567
2. Luther: 161,808
3. Bucer: 127,275
4. Bullinger: 85,941
5. Zwingli: 59,341

Path betweenness

1. Melanchthon: 17,639
2. Luther: 5,054
3. Bullinger: 4,696
4. Bucer: 2,622
5. Blarer: 1,926

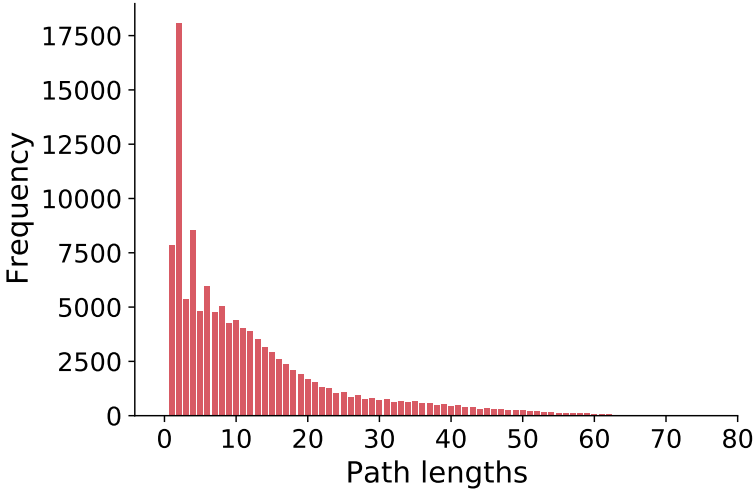
- **Aggregated networks overestimates betweenness centrality**
- **Melanchthon drives information spread**

New research question

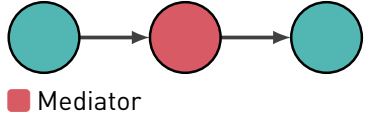
- [x] **Who drove the spread of ideas?**
- [ ] **Which communication patterns were used to spread ideas?**

Inspecting paths ⇒ communication patterns

RQ: Which communication patterns were used to spread ideas?



- Most frequent path length = 2
- Communication pattern ⇒ mediated communication
- **Why are mediators so common?**



Why is mediated communication so dominant?

Qualitative analysis

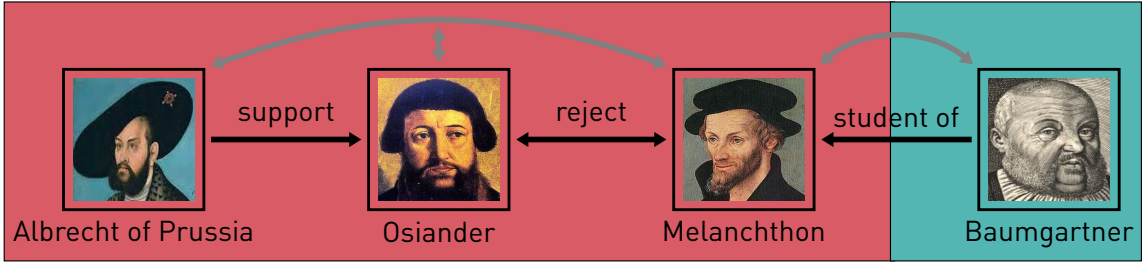
- Are paths a consequence of the quality of relationships?
- Inspect most frequent paths of length 2

Statistical analysis

- Is dominance of paths significant or random?
- Model selection with inferential statistics

Qualitative inspection of Melanchthon's path

Albrecht of Prussia → Melanchthon → Baumgartner (16)



⟷ letter correspondence

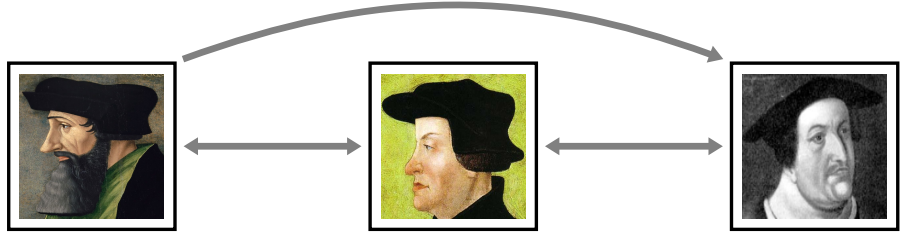
1st generation

2nd generation

- 2nd generation is more differentiated
- Similar patterns for other students of Melanchthon, e.g., Camerarius E., Veit, Huberius, Alber
- Melanchthon as diplomatic mediator

Qualitative inspection of Zwingli's path

Oekolampad ➔ Zwingli ➔ Vadian (15)



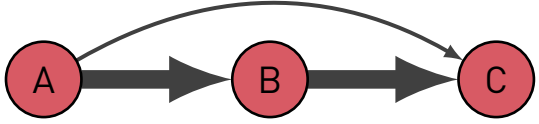
⟷ letter correspondence

- Busy Vadian prioritises his receivers
- Among all paths X ➔ Zwingli ➔ Vadian, Vadian does not reply to 66% of the original senders
- Zwingli as helpful mediator

Statistical inspection of paths: concepts

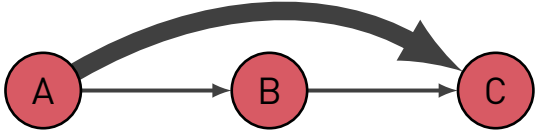
Are paths significant for information flow in network or do they occur at random?

Significant path



- A and C mainly communicate indirectly

Random path



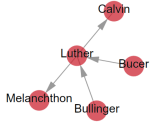
- A and C mainly communicate directly and sometimes indirectly

- **Perform statistical test to find out**

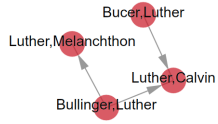
Statistical inspection of paths: models

Are paths significant for information flow in network or do they occur at random?

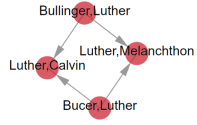
- **Higher-Order network model**
  - Nodes: paths of a specific length
- **Statistical test**
  - $\text{Paths}_{\text{length } n} \stackrel{?}{=}$ significant
  - Remove time stamps
  - Construct new paths
  - $\text{Paths}_{\text{observed}} \stackrel{?}{=} \text{paths}_{\text{randomised}}$
  - ➔ If different
    $\Rightarrow$ observed paths are significant
- **Multi-order network model**
  - Stacked HONs
  - Statistical tests for many path lengths



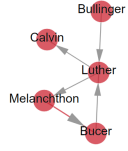
Aggregated



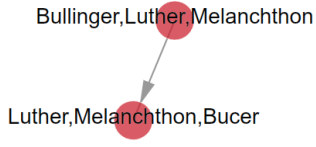
HON_{obs}, len $= 1$



HON_{rand}, len $= 1$



Aggregated



HON_{obs}, len $= 2$

Higher- and Multi-Order Models for statistical inspection of paths

- Build multi-order model: automatically generates higher-order models
- Build higher-order models to run statistical test manually

```
# Multi-order model (stack HON_k)
multi_order = pp.MultiOrderModel(ref_paths, max_order=10)
# Whihc HON is best? Model selection with maximum likelihood
best_order = multi_order.estimate_order(ref_paths)

# Build separate HON and randomise paths (null_model=True)
hon_2 = pp.HigherOrderNetwork(ref_paths, k=2)
hon_2_null = pp.HigherOrderNetwork(ref_paths, k=2, null_model=True)
# Compare random and observed HONs
A_2 = hon_2.adjacency_matrix()
idx_2 = hon_2.node_to_name_map()
A_2n = hon_2_null.adjacency_matrix()
idx_2n = hon_2_null.node_to_name_map()
for (sender,receiver) in hon_2_null.edges:
    # Path is X times more likely in observed model than in randomised one
    print((sender,receiver), A_2[idx_2[sender],idx_2[receiver]] - A_2n[idx_2n[sender],idx_2n[receiver
    ]])
```

Results of statistical inspection of paths

Ran multi-order model

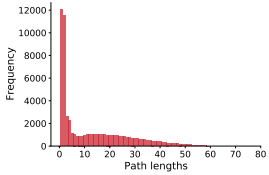
- Significant path length $= 2$
- Mediated communication is main communication pattern for spread of ideas during Reformation

Reason for caution

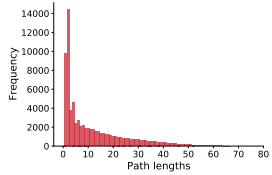
- Qualitative and statistical results do not always match
  - Statistical model considers subpaths
- δt robust?

- **Justify assumptions and parameter choices**
- **Do robustness checks**

Significant path length $= 1$

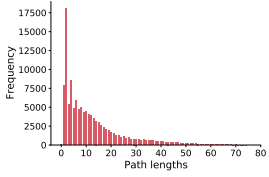


$\delta t = 7$

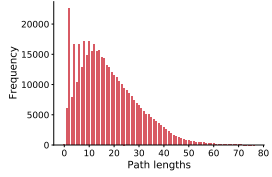


$\delta t = 10$

Significant path length $= 2$



$\delta t = 14$



$\delta t = 20$

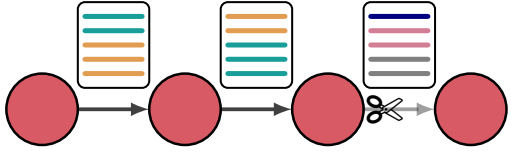
Scope and limitations of `pathpy` for letter network

Scope

- **Evolution of network**
  Visualise dynamic formation of edges
- **Accurate topological network measures**
  Paths account for temporal ordering of edges
- **Detect significant communication patterns**
  Inferential statistics with Higher- and Multi-Order Models
- **Dynamic and static visualisations**
  Alluvial diagrams, diffusion simulations, directed acyclic graphs, ...

Limitations

- **Preseved quantity assumption**
  Ideas conveyed in letters remain fixed



- **Constant time scale assumption**
  $\delta t = const$ for observation period
- **No time scale detection**
  choose δt by thinking
- **Randomisation restricted to network data**
  Theory and external data are excluded

➔ **Addresses problem of time ignorance in networks**

➔ **Not domain-specific: models are restricted to network data**

The Chair of Systems Design: Origin of `pathpy`

The Chair of Systems Design

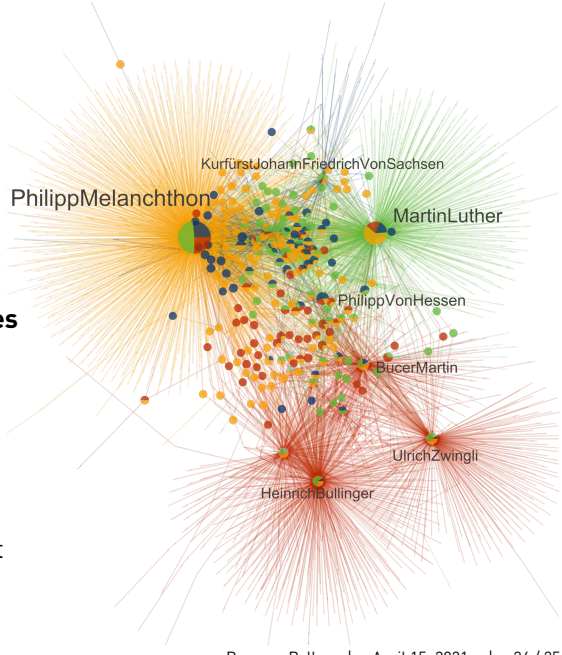
- **Professor**: Frank Schweitzer
- **Approach:** Data-driven modeling of complex systems
- **Research areas**
  - Structure and dynamics of systems
  - Social organisations
  - Economic networks
- **Team:** multidisciplinary
  physics, maths, computer- and social science, economics

`pathpy` resources

- **Main developer:** Prof. Ingo Scholtes (Wuppertal)
- `pathpy` **website:** `https://www.pathpy.net/index.html`
  - Videos
  - Tutorial
  - Papers

Summary

- **Time matters in network**
  - Interactions are time-stamped and temporally ordered
- **Aggregated networks ignore time**
  - Time can run backwards
  - Leads to wrong network measures
- **`pathpy` accounts for temporal ordering of edges**
  - Accurate topological network measures
  - Dynamic visualisations
  - Statistical inference
- **Insights for Reformation**
  - Melanchthon: main driver for idea spread
  - 1-hop mediated communication is dominant communication pattern



Thank you

Ramona Roller

PhD candidate for data-driven modeling of socio-historic systems

ETH Zurich

Chait of Systsem Design (Prof. Frank Schweitzer)

https://www.sg.ethz.ch/team/people/ramona_roller/

rroller@ethz.ch

Scholtes, Ingo (2017).
pathpy.
https://www.pathpy.net/index.html.