LSCPM: finding communities in Link Streams by Clique Percolation Method

Alexis Baudin*, Lionel Tabourier and Clémence Magnien

September 25th, 2023

TIME 2023
30th International Symposium on Temporal Representation and Reasoning

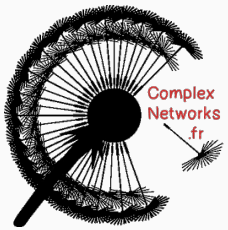
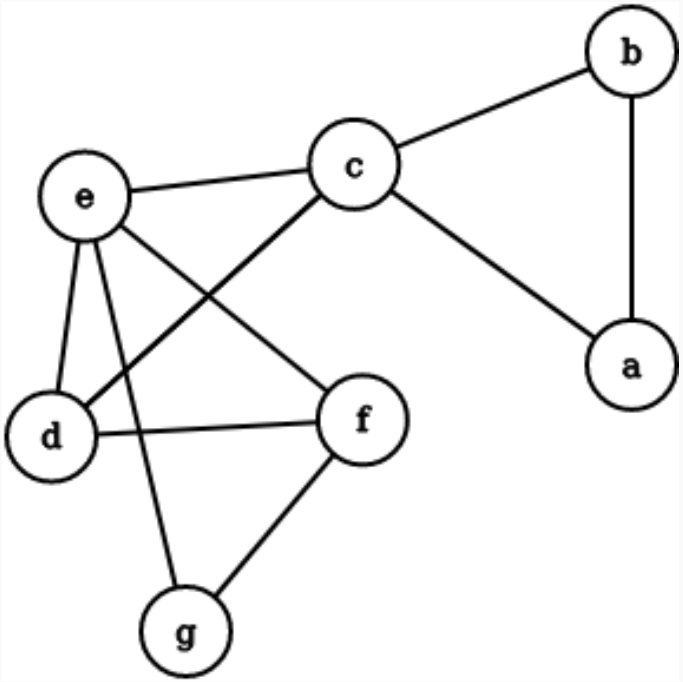


Table of contents

1 - Communities in graphs by Clique Percolation Method (CPM)

> Definition – Graph



Graph formalism

- Vertices: $a, b, ..., g$
- Interactions: edges $\{a, b\}$, $\{a, c\}$, ...

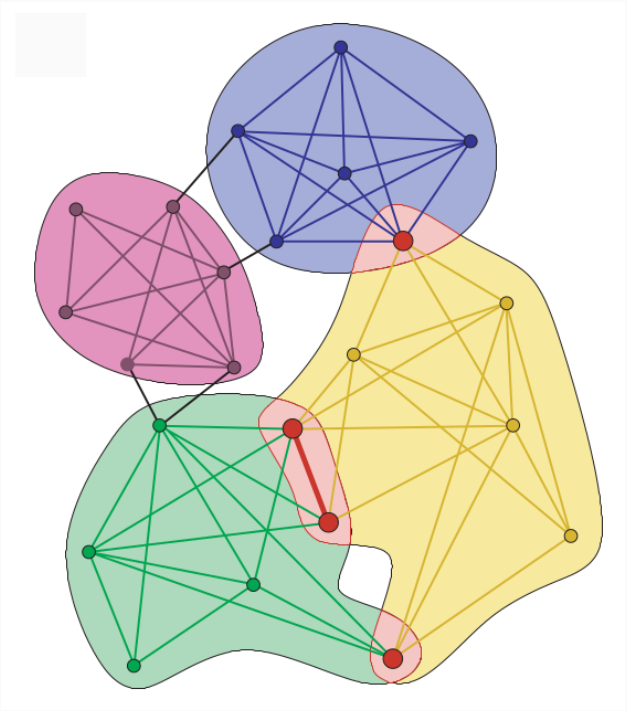
Examples

Contacts between people, link between web pages, neural activities, ...

> Communities in graphs

Communities: sets of vertices

- Densely connected inside
- Sparsly connected outside



Palla et al. 2005

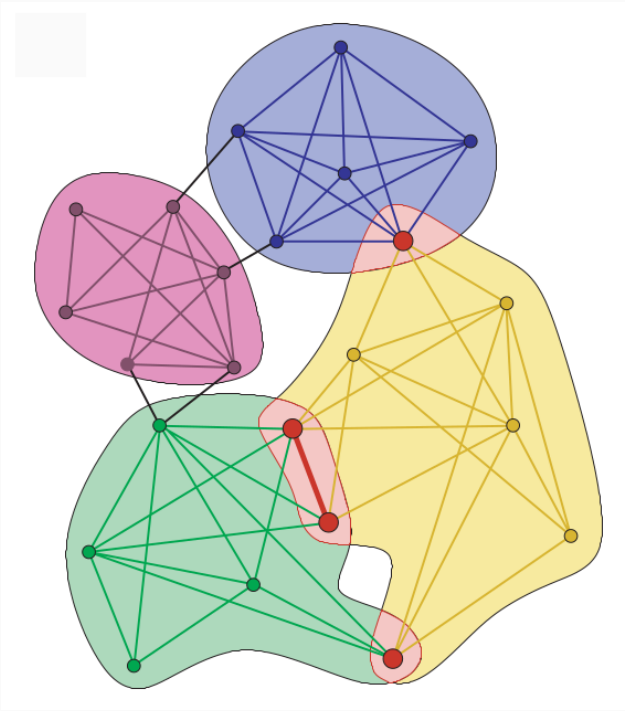
> Communities in graphs

Communities: sets of vertices

- Densely connected inside
- Sparsly connected outside

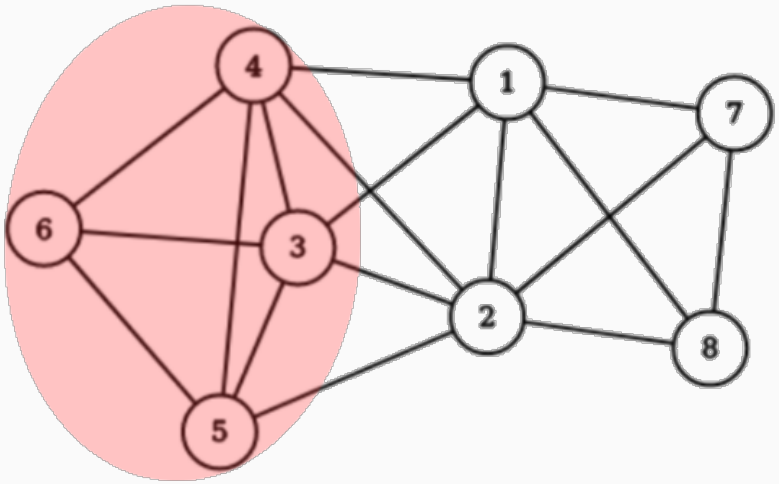
Interest:

- Locate areas of high interaction density
- Understanding the organizational structure of interactions
- Zoom in / out



Palla et al. 2005

> Clique Percolation Method in graphs (CPM)

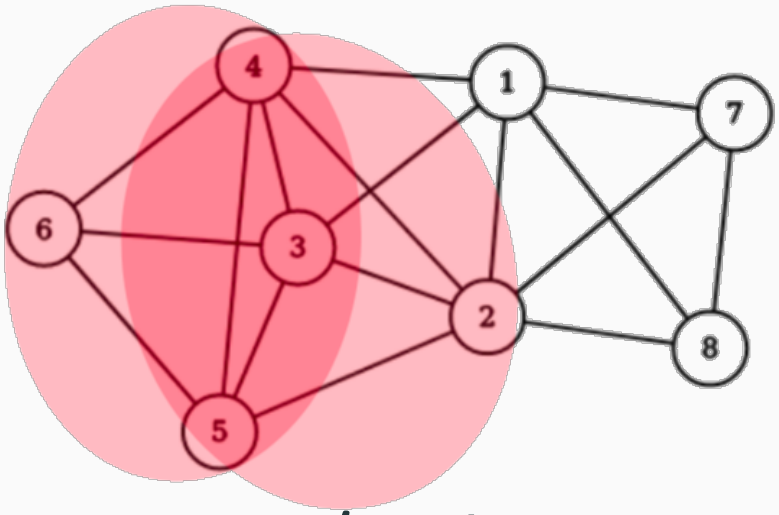


$k = 4$

k-clique

Set of k nodes all connected to each other.

> Clique Percolation Method in graphs (CPM)



$k = 4$

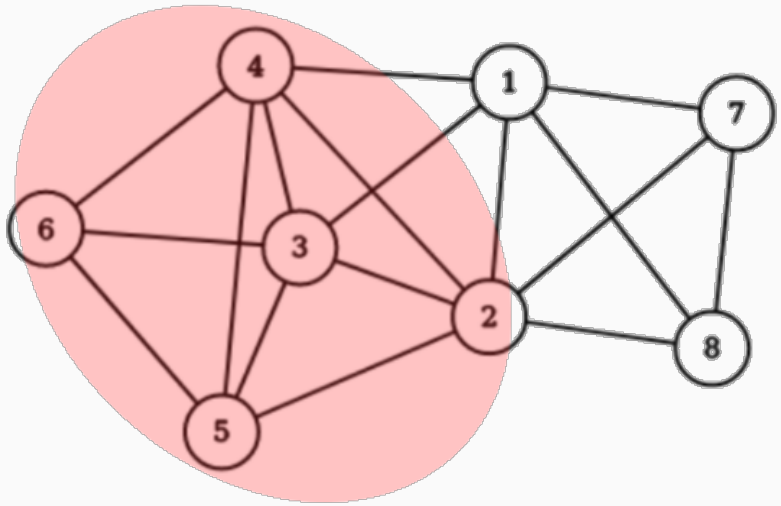
k-clique

Set of k nodes all connected to each other.

Grouping rule

Two k-cliques are adjacent if they share $k - 1$ nodes.

> Clique Percolation Method in graphs (CPM)



$k = 4$

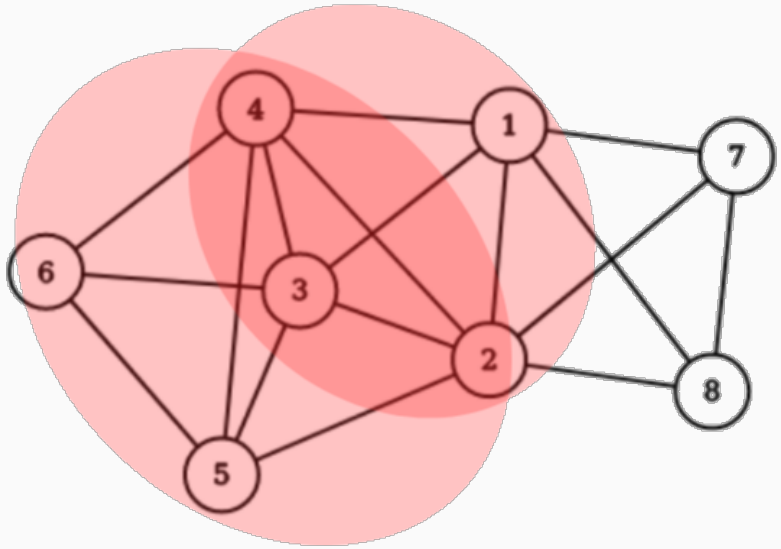
k-clique

Set of k nodes all connected to each other.

Grouping rule

Two k-cliques are adjacent if they share $k - 1$ nodes.

> Clique Percolation Method in graphs (CPM)



$k = 4$

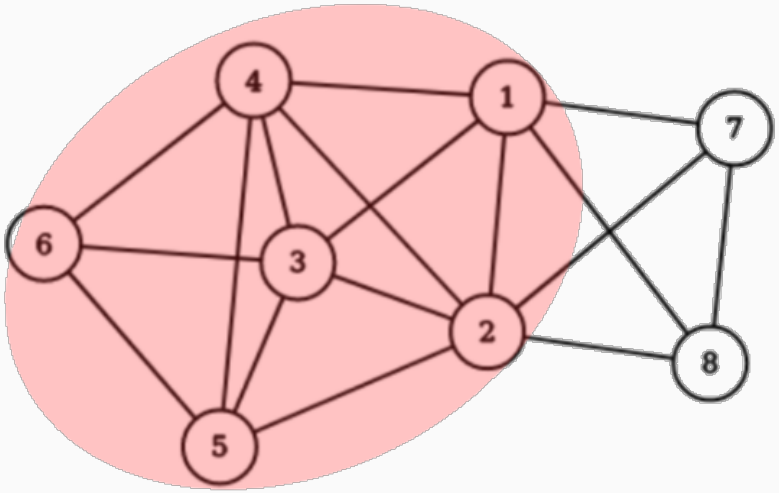
k-clique

Set of k nodes all connected to each other.

Grouping rule

Two k-cliques are adjacent if they share $k - 1$ nodes.

> Clique Percolation Method in graphs (CPM)



$k = 4$

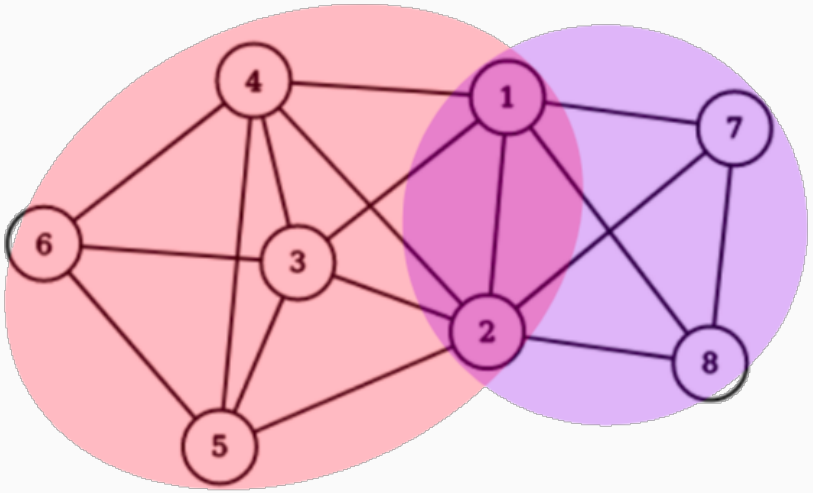
k-clique

Set of k nodes all connected to each other.

Grouping rule

Two k-cliques are adjacent if they share $k - 1$ nodes.

> Clique Percolation Method in graphs (CPM)



$k = 4$

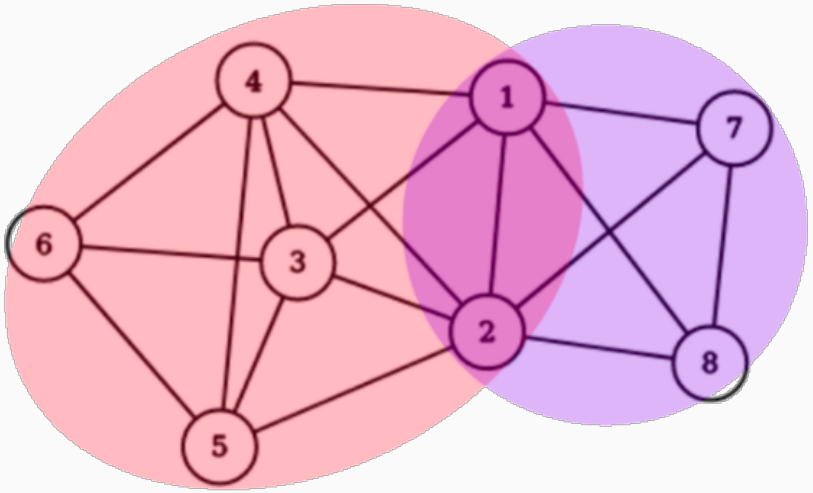
k-clique

Set of k nodes all connected to each other.

Grouping rule

Two k-cliques are adjacent if they share $k - 1$ nodes.

> Clique Percolation Method in graphs (CPM)



$k = 4$

k-clique

Set of k nodes all connected to each other.

Grouping rule

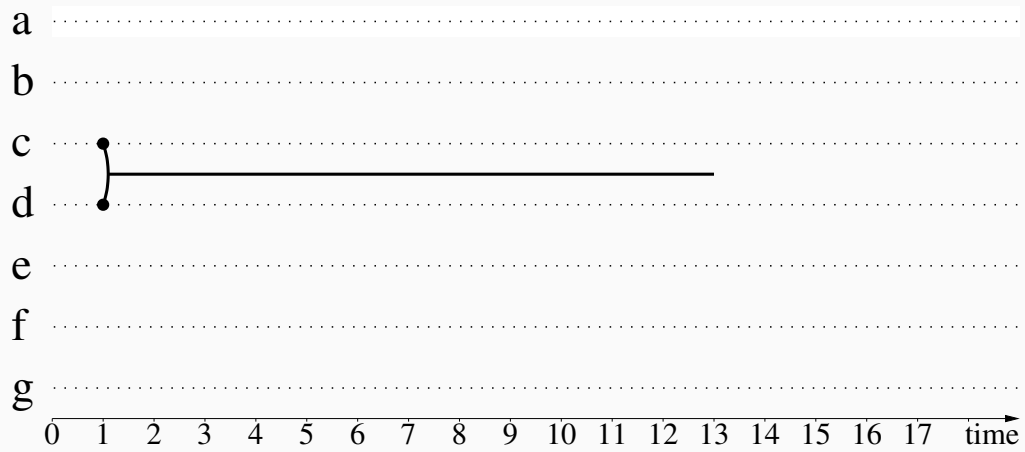
Two k-cliques are adjacent if they share $k - 1$ nodes.

Advantages of this definition of communities:

- Local definition
- Deterministic
- Communities can overlap

2 - Temporal communities in link streams (LSCPM)

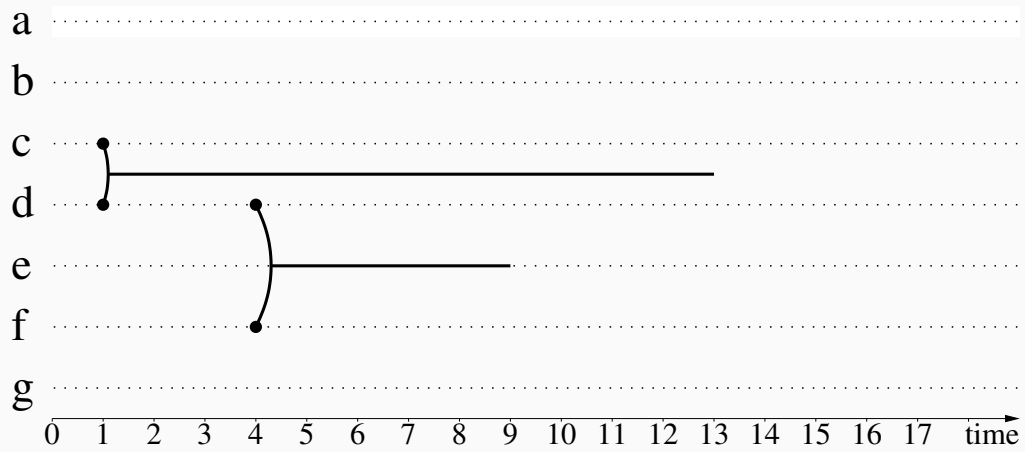
> Definition – Link stream



Link stream formalism

- Vertices: $a, b, ..., g$
- Time period: $[0, 18]$
- Interactions: temporal edges
  - c, d linked over $[1, 13]$

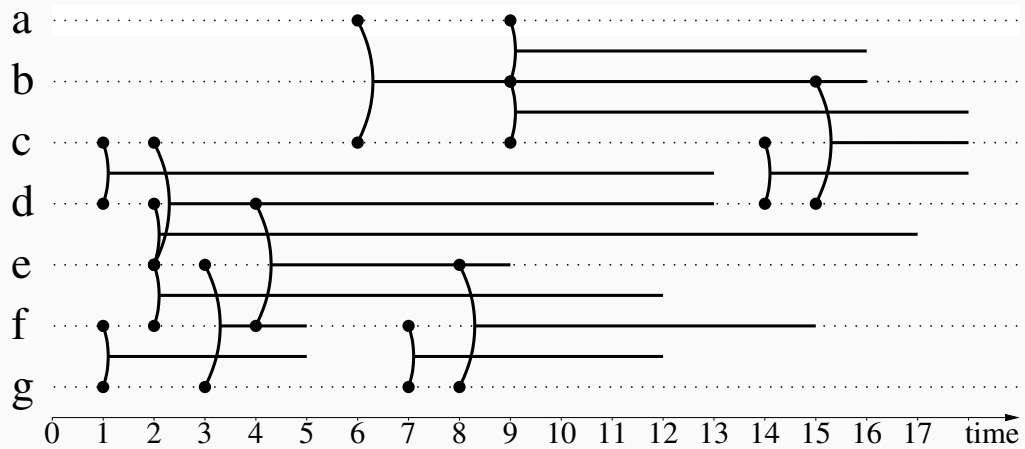
> Definition – Link stream



Link stream formalism

- Vertices: $a, b, ..., g$
- Time period: $[0, 18]$
- Interactions: temporal edges
  - c, d linked over $[1, 13]$
  - d, f linked over $[4, 9]$

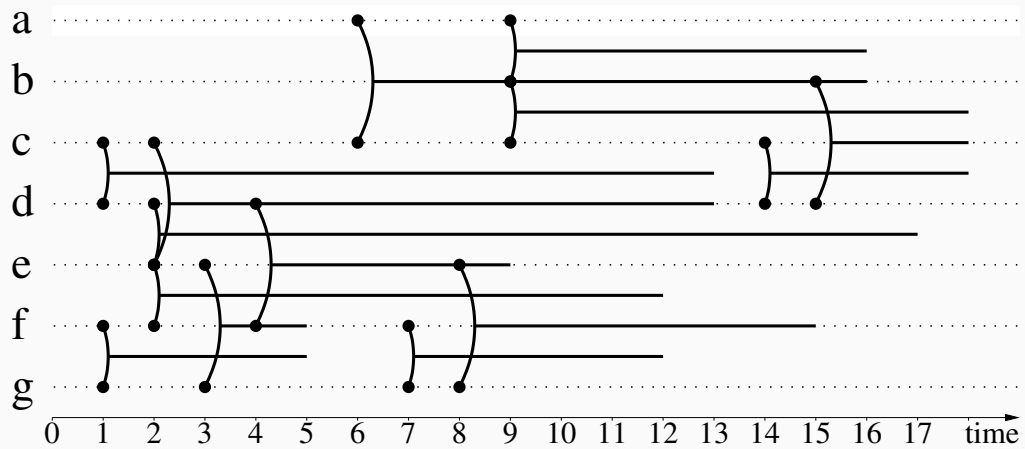
> Definition – Link stream



Link stream formalism

- Vertices: $a, b, ..., g$
- Time period: $[0, 18]$
- Interactions: temporal edges
  - c, d linked over $[1, 13]$
  - d, f linked over $[4, 9]$
  - ...

> Definition – Link stream



Link stream formalism

- Vertices: $a, b, ..., g$
- Time period: $[0, 18]$
- Interactions: temporal edges
  - c, d linked over $[1, 13]$
  - d, f linked over $[4, 9]$
  - ...

Advantages

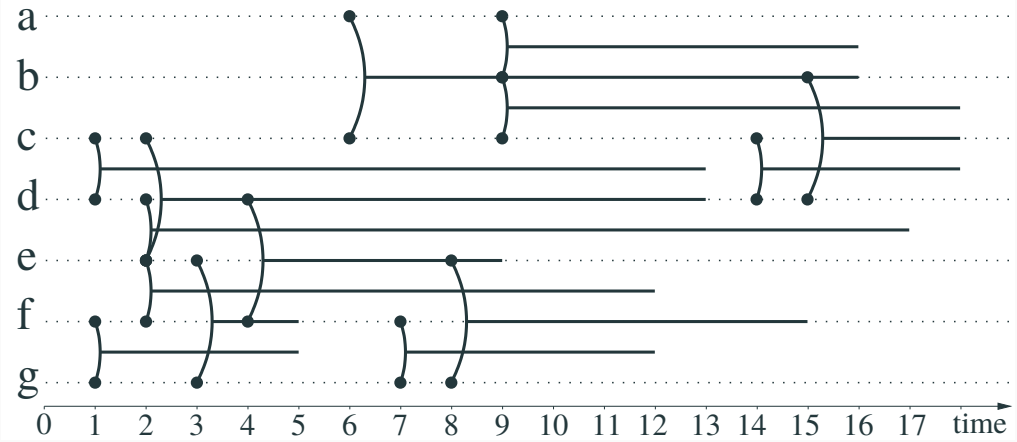
- deals directly with the stream of interactions
- no arbitrary choice of time scale
- time is continuous

> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*

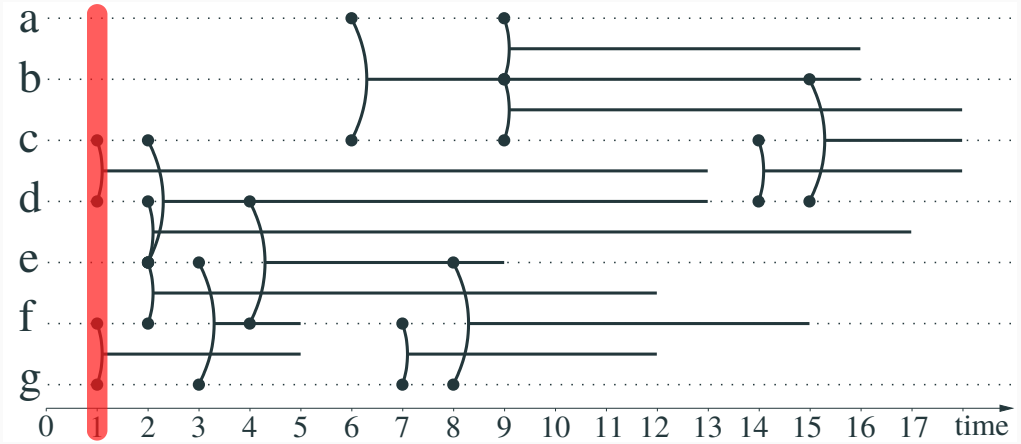


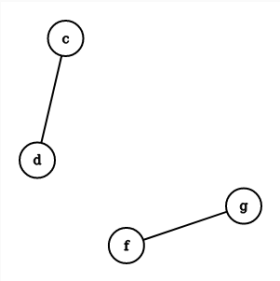
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





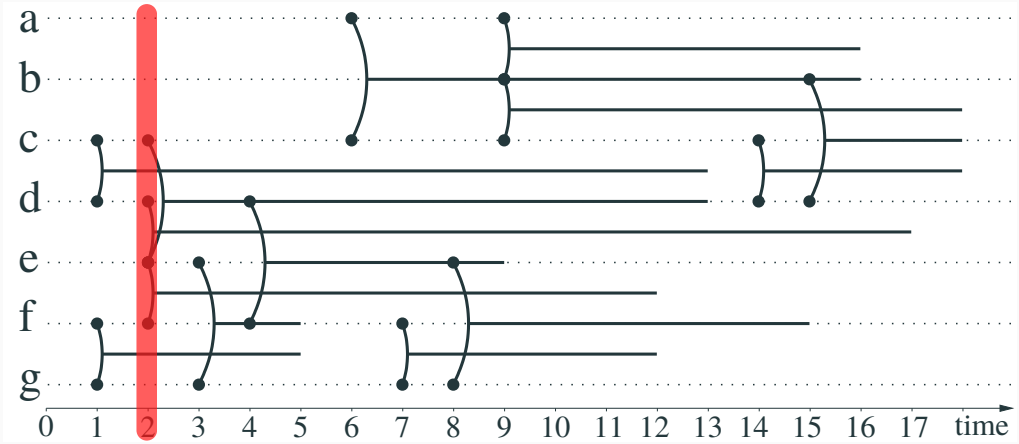
$t = 1$

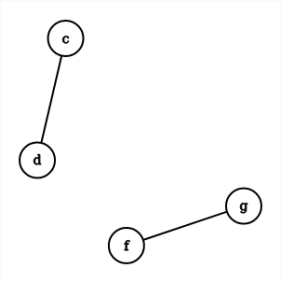
> First extension of CPM to Temporal Graphs (DCPM)

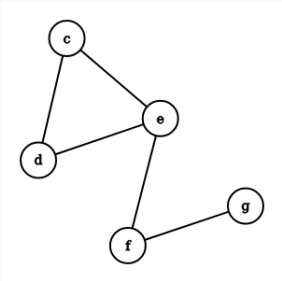
CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*







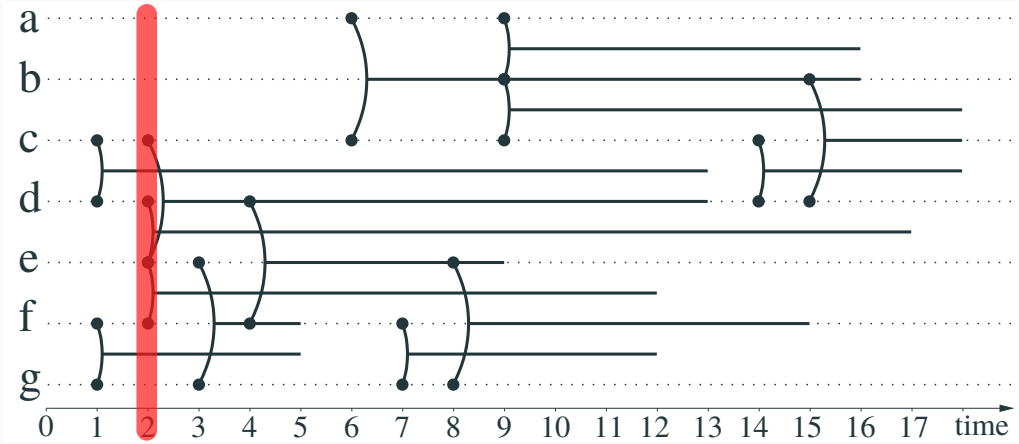
$t = 1 \longrightarrow t = 2$

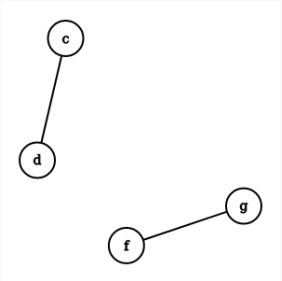
> First extension of CPM to Temporal Graphs (DCPM)

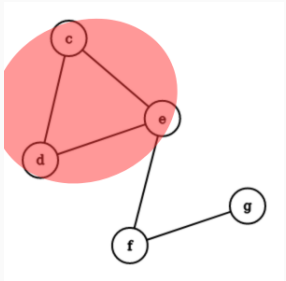
CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*







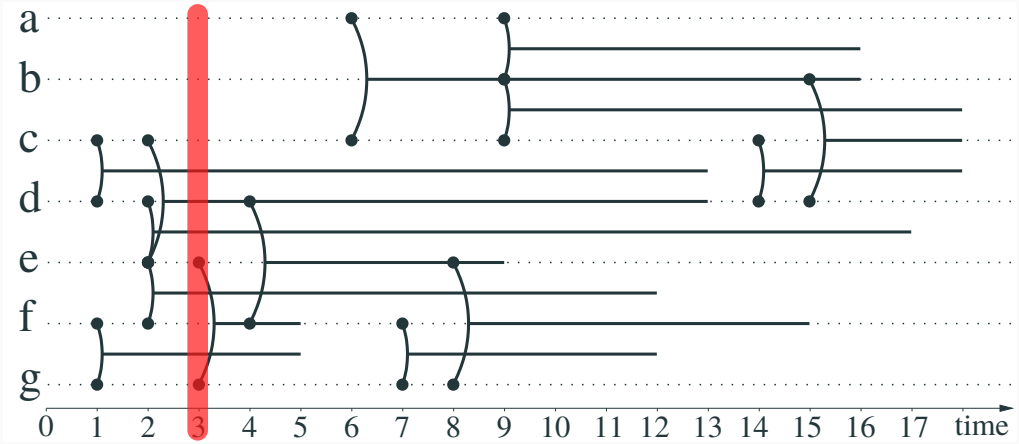
$t = 1 \longrightarrow t = 2$

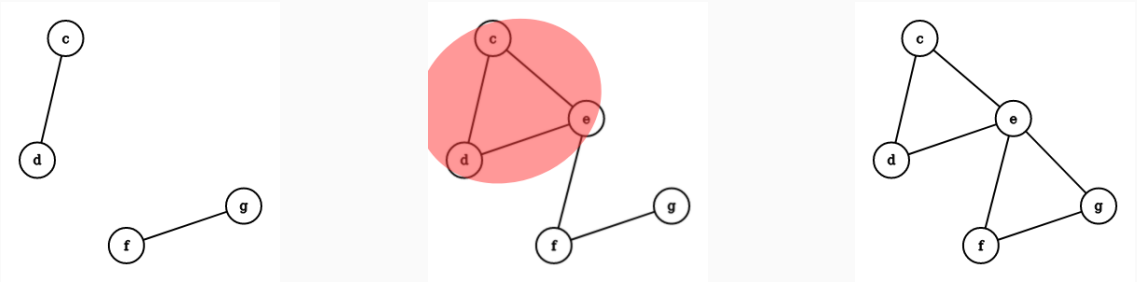
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





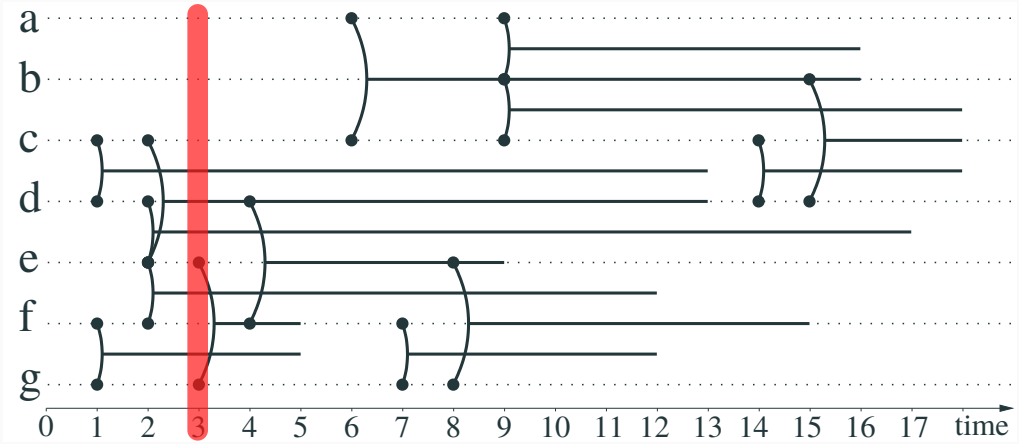
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3$

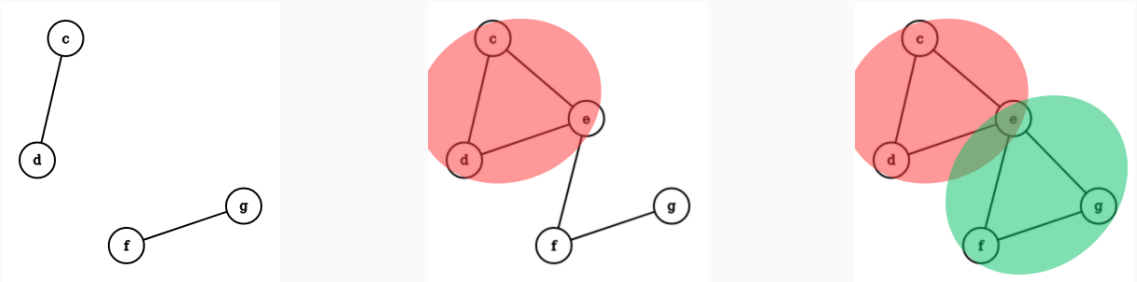
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





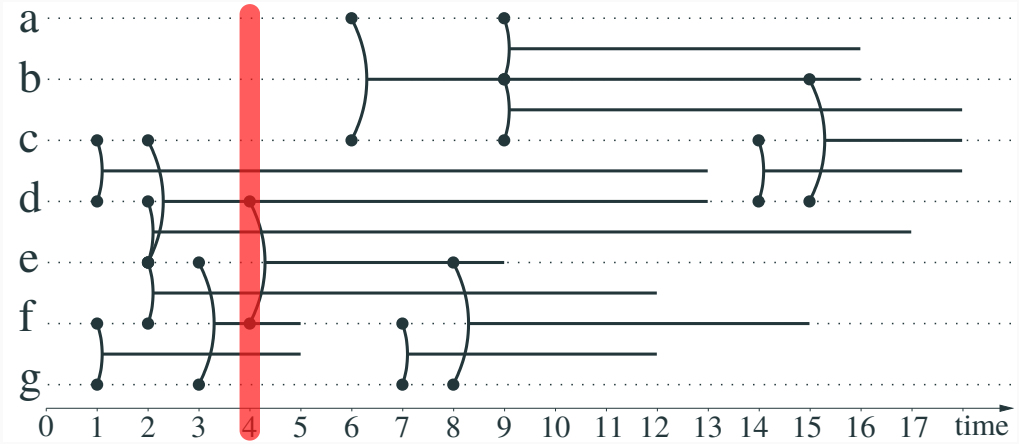
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3$

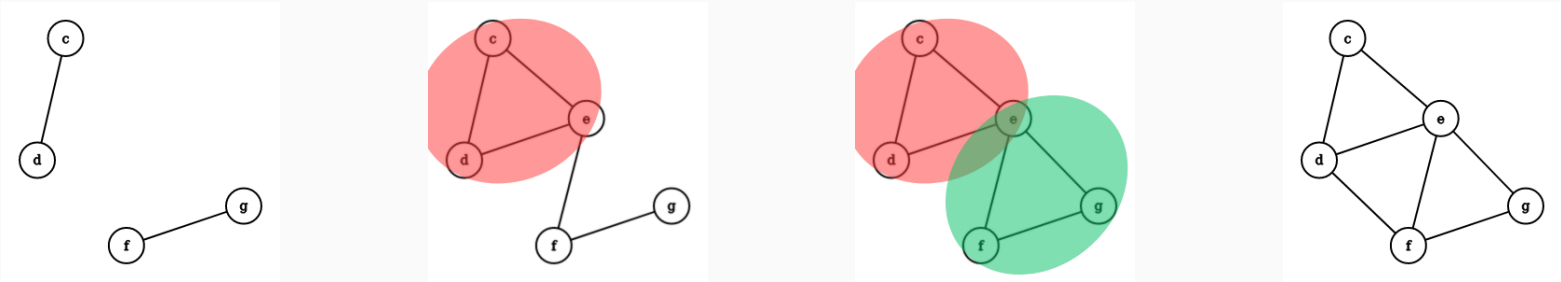
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





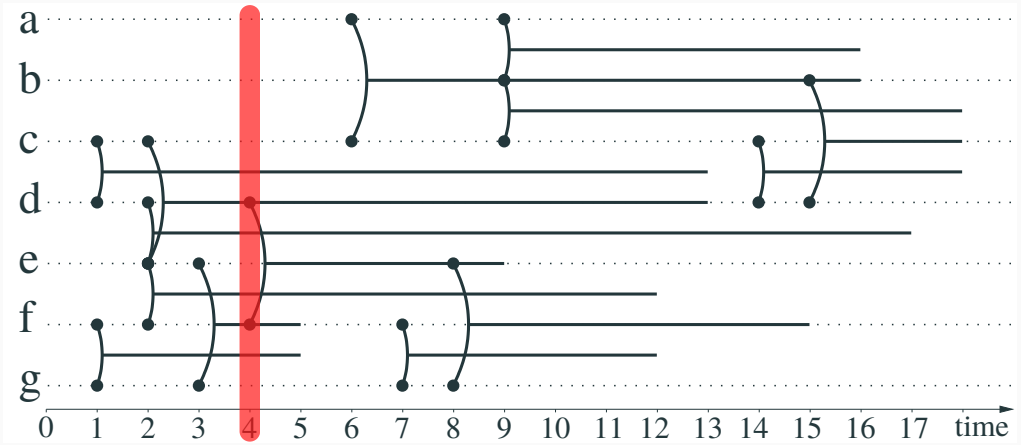
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4$

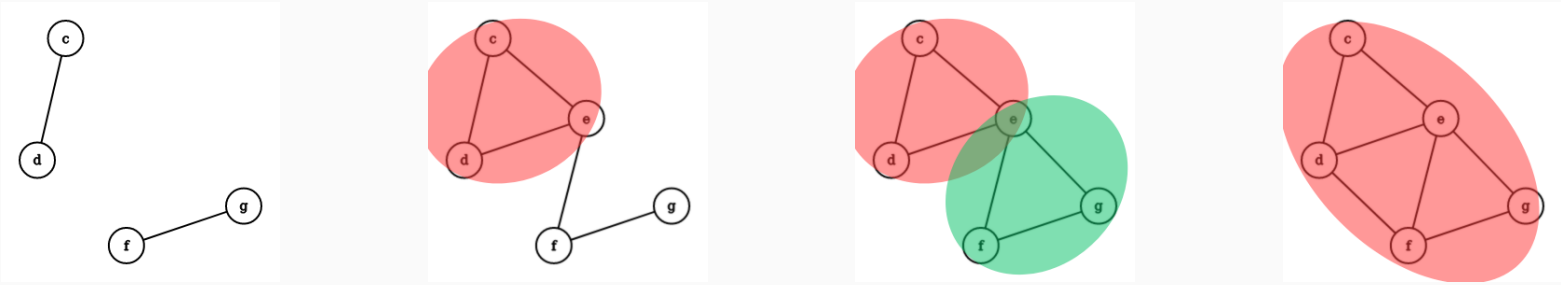
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





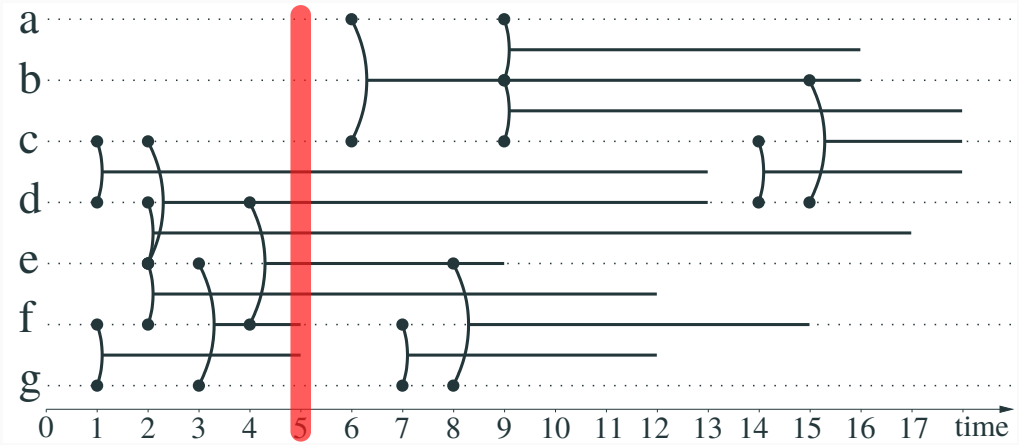
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4$

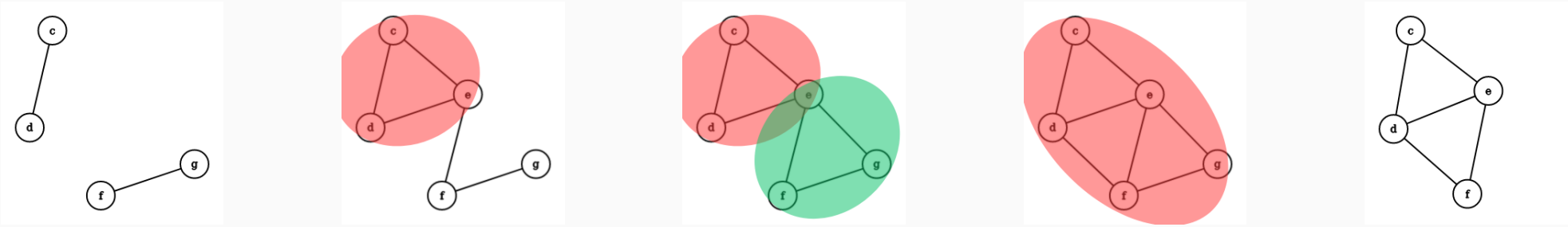
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





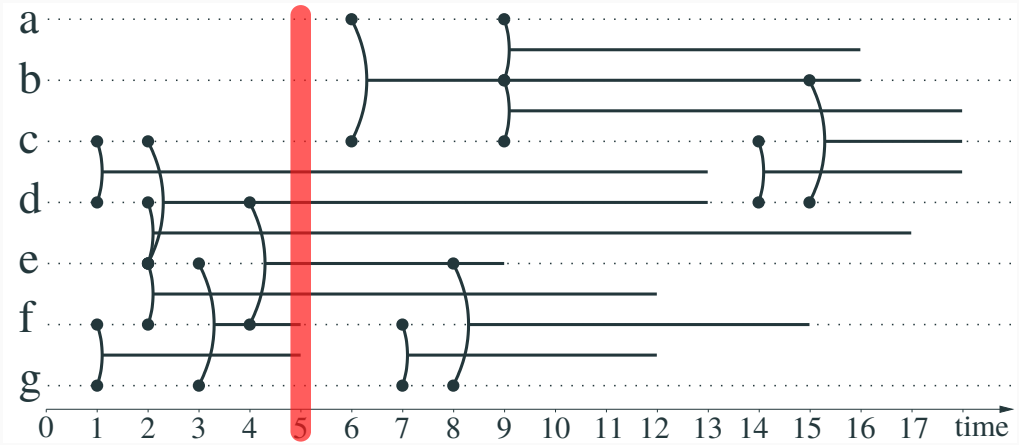
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4 \longrightarrow t = 5$

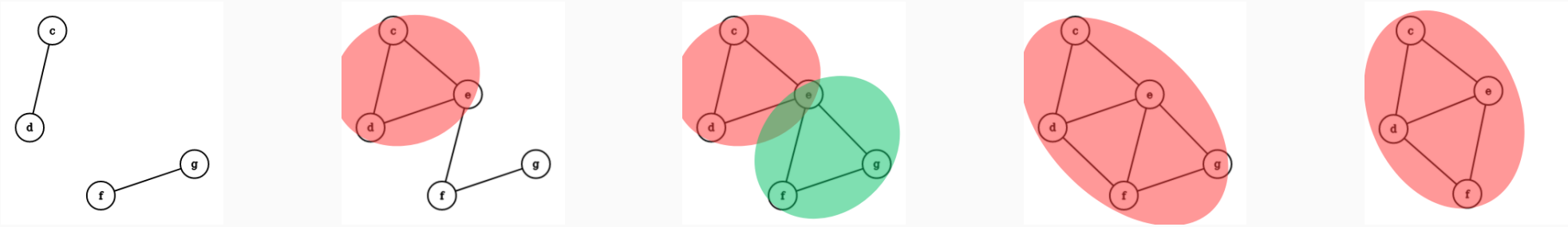
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





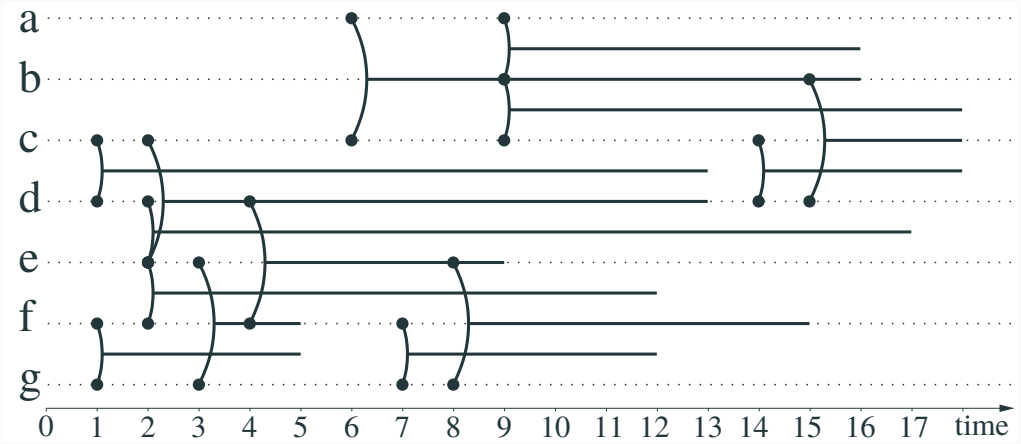
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4 \longrightarrow t = 5$

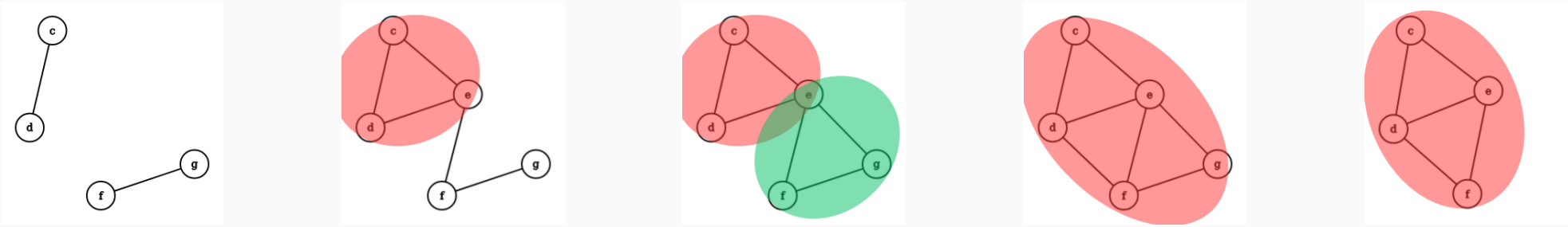
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





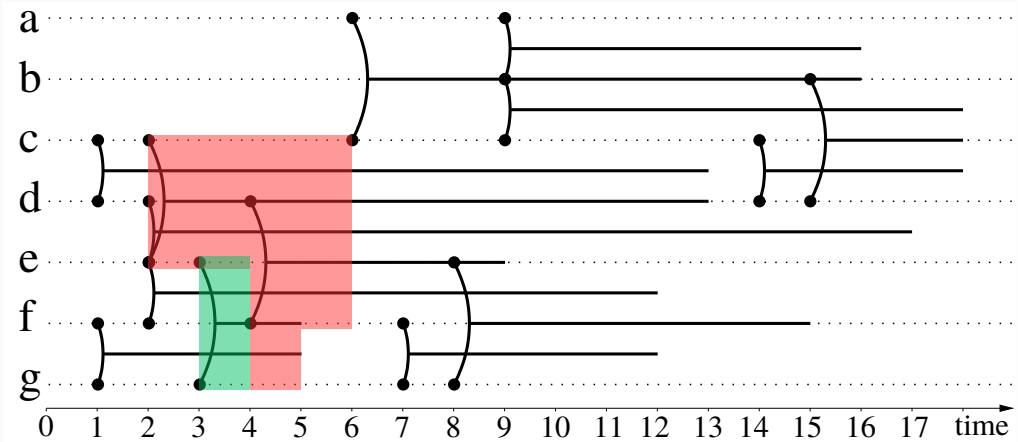
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4 \longrightarrow t = 5$

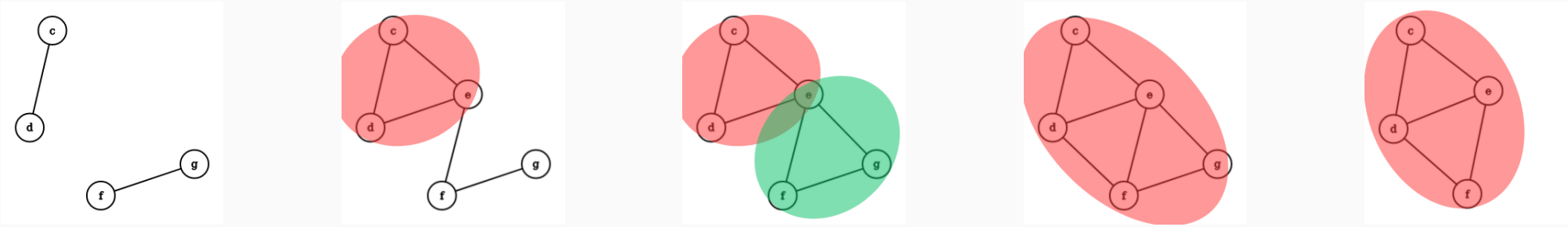
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

Example with $k = 3$*:*





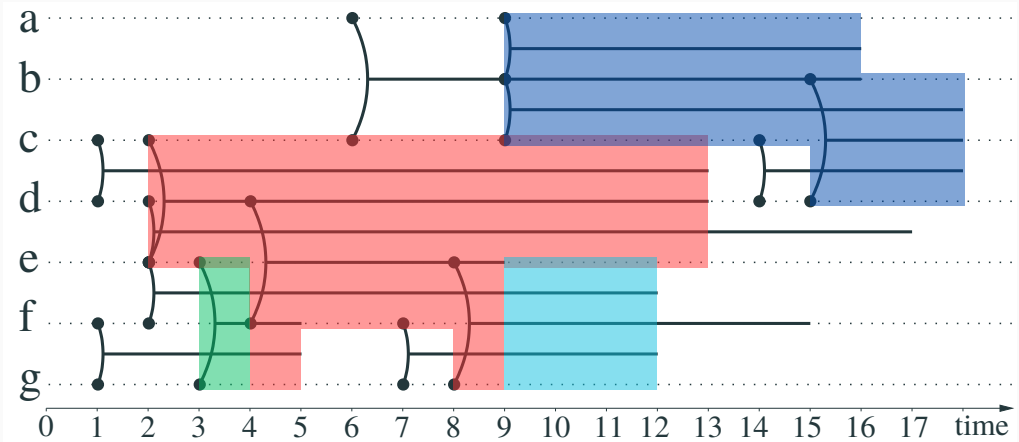
$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4 \longrightarrow t = 5$

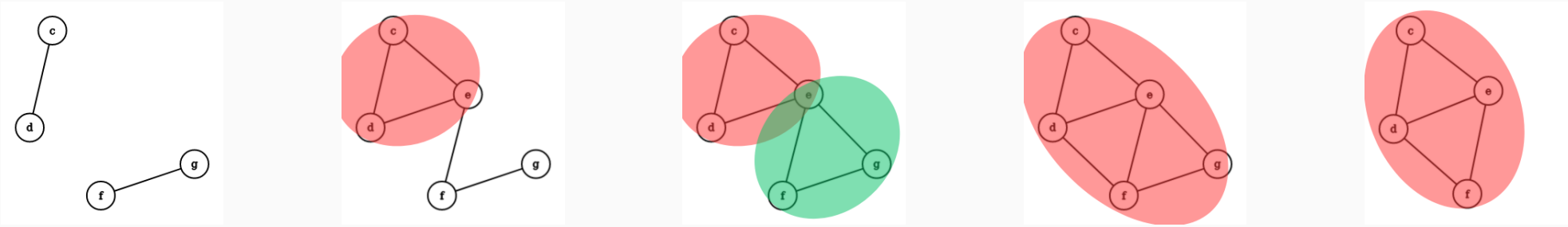
> First extension of CPM to Temporal Graphs (DCPM)

CPM first extended to temporal graphs by *Palla et al. (2007)*

$\Rightarrow$ CPM communities that evolve from one time step to the next

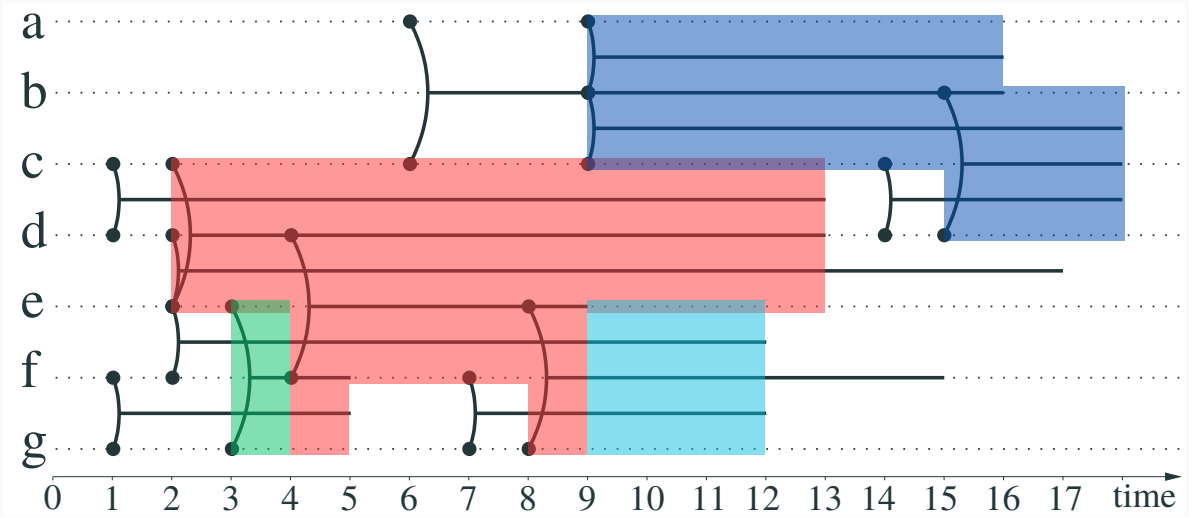
Example with $k = 3$*:*





$t = 1 \longrightarrow t = 2 \longrightarrow t = 3 \longrightarrow t = 4 \longrightarrow t = 5$

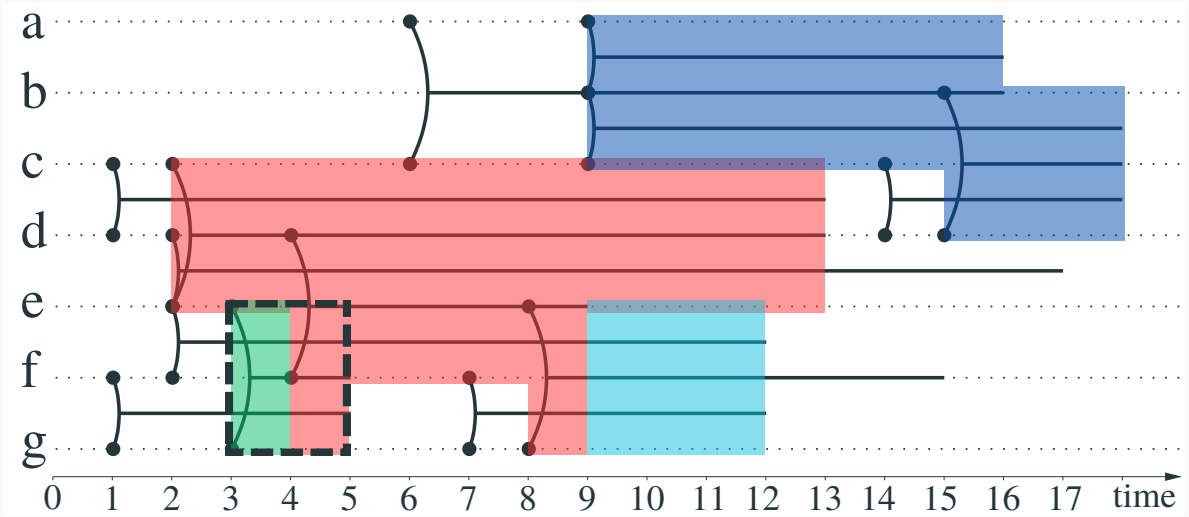
Dynamic CPM communities (DCPM):



Limits of DCPM algorithm:

- Computing communities at each time step: time consuming;
- Some temporal data expected to be grouped are not.

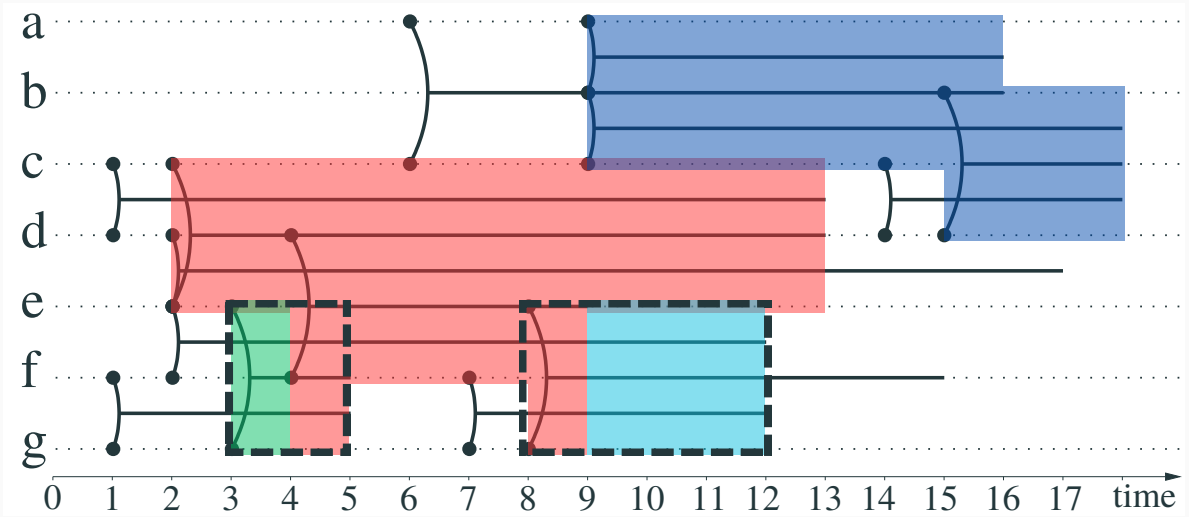
Dynamic CPM communities (DCPM):



Limits of DCPM algorithm:

- Computing communities at each time step: time consuming;
- Some temporal data expected to be grouped are not.

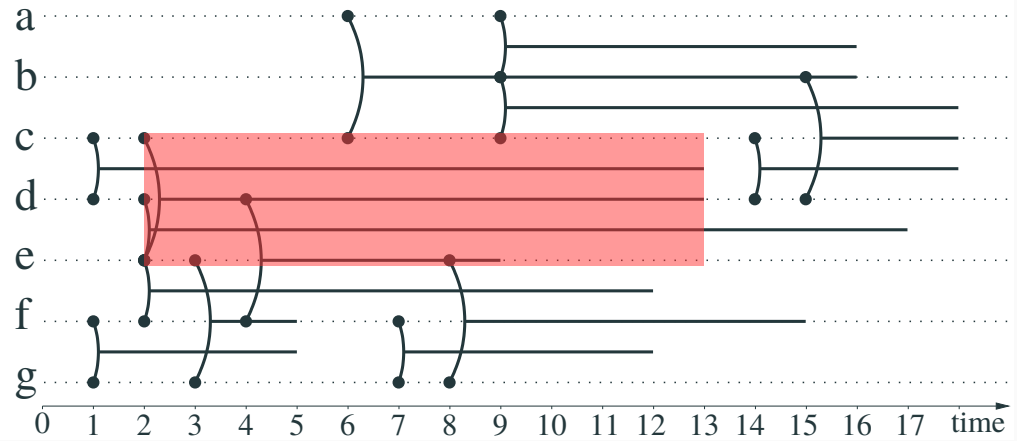
Dynamic CPM communities (DCPM):



Limits of DCPM algorithm:

- Computing communities at each time step: time consuming;
- Some temporal data expected to be grouped are not.

> Our algorithm: CPM in link streams (LSCPM)

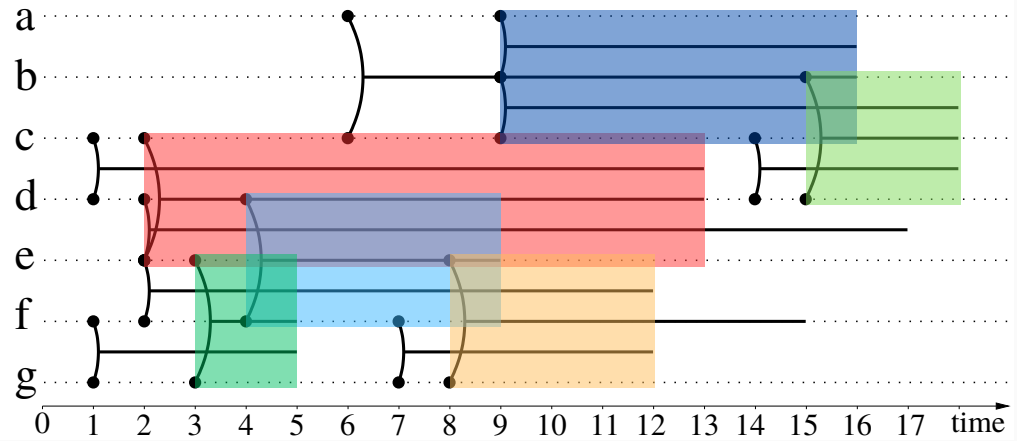


Example with $k = 3$.

k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

> Our algorithm: CPM in link streams (LSCPM)

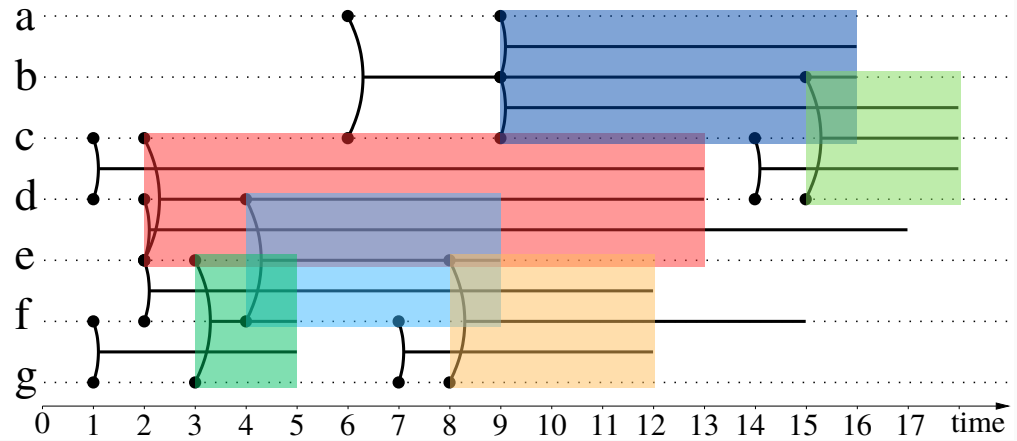


Example with $k = 3$.

k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

> Our algorithm: CPM in link streams (LSCPM)



Example with $k = 3$.

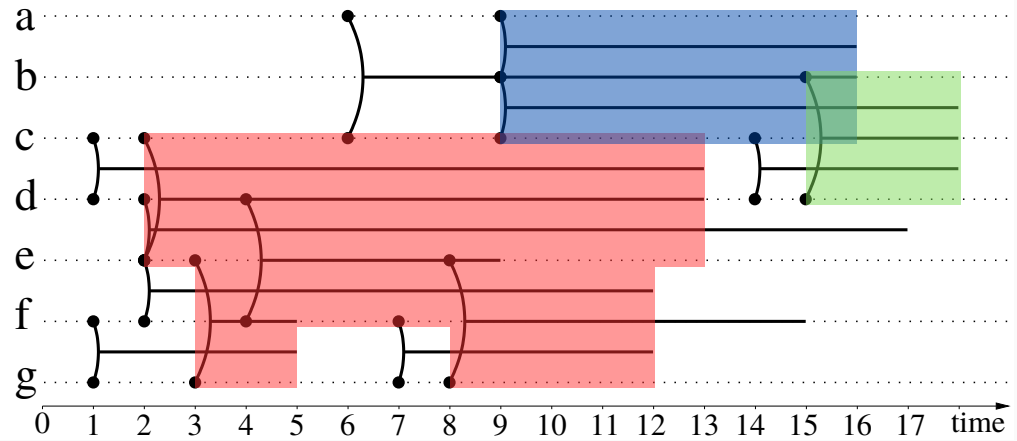
k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

Grouping rule

k-cliques adjacent: they share $k - 1$ nodes over a time interval.

> Our algorithm: CPM in link streams (LSCPM)



Example with $k = 3$.

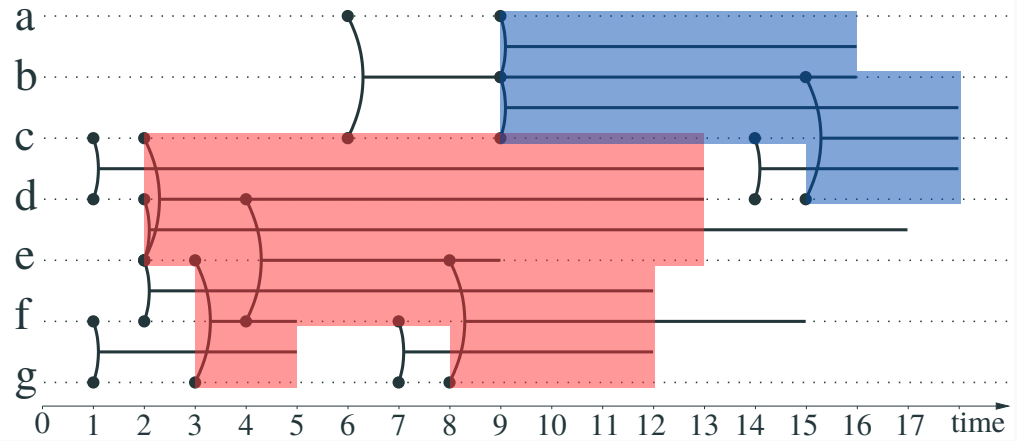
k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

Grouping rule

k-cliques adjacent: they share $k - 1$ nodes over a time interval.

> Our algorithm: CPM in link streams (LSCPM)



Example with $k = 3$.

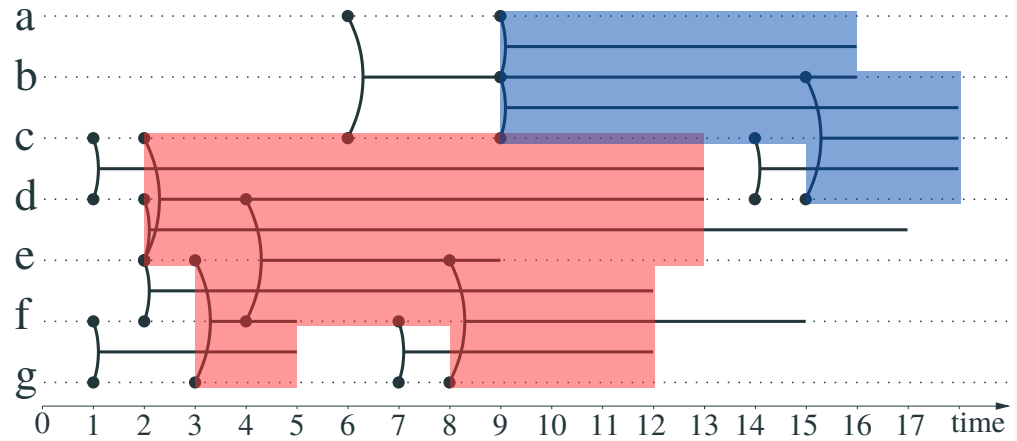
k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

Grouping rule

k-cliques adjacent: they share $k - 1$ nodes over a time interval.

> Our algorithm: CPM in link streams (LSCPM)



Example with $k = 3$.

k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

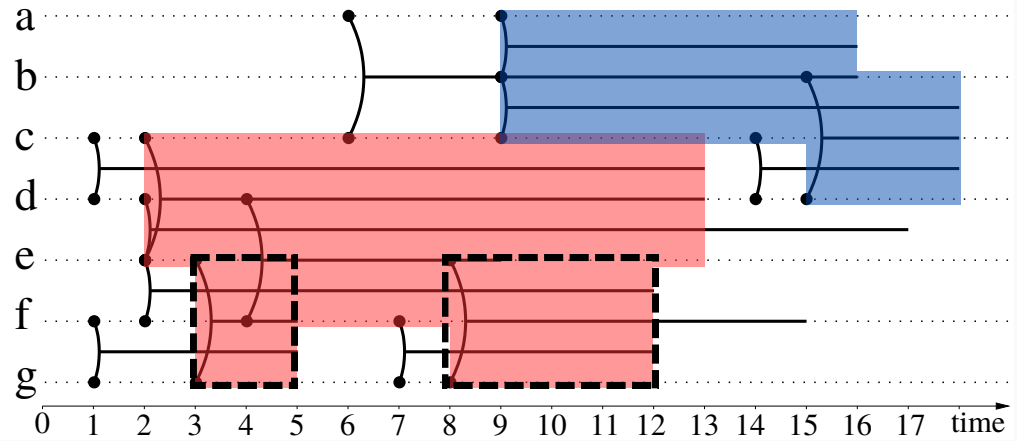
Grouping rule

k-cliques adjacent: they share $k - 1$ nodes over a time interval.

Comparing to DCPM (state of the art):

- No need to compute communities at each time step ✓
- All temporal cliques are grouped ✓

> Our algorithm: CPM in link streams (LSCPM)



Example with $k = 3$.

k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

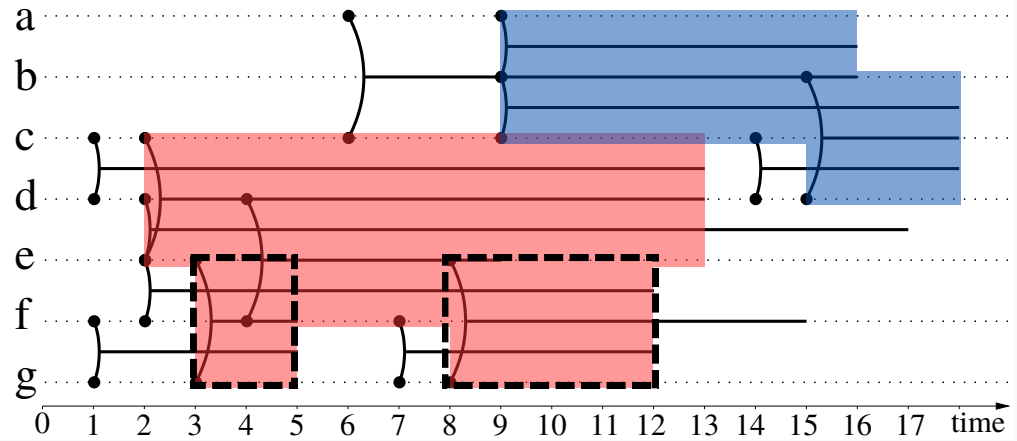
Grouping rule

k-cliques adjacent: they share $k - 1$ nodes over a time interval.

Comparing to DCPM (state of the art):

- No need to compute communities at each time step ✓
- All temporal cliques are grouped ✓

> Our algorithm: CPM in link streams (LSCPM)



Example with $k = 3$.

k-clique in link stream

k nodes all connected to each other during a time interval $[t_0, t_1]$.

Grouping rule

k-cliques adjacent: they share $k - 1$ nodes over a time interval.

Comparing to DCPM (state of the art):

- No need to compute communities at each time step ✓
- All temporal cliques are grouped ✓

→ LSCPM communities are **union** of DCPM communities

3 - Experiments on real datasets

Efficiency – computation times

Link stream	**# links**		
Households	2,136		
Highschool	5,528		
Infectious	44,658		
Foursquare	268,472		
Wikipedia	39,953,380		

Efficiency – computation times

		$k = 3$		$k = 4$	
Link stream	**# links**	**DCPM**	**LSCPM**	**DCPM**	**LSCPM**
Households	2,136	1.5s	0.1s	1.0s	0.1s
Highschool	5,528	3.6s	0.1s	1.9s	0.1s
Infectious	44,658	10min49s	1.4s	6min12s	3.3s
Foursquare	268,472	3h01min	9.2s	2h28min	43s
Wikipedia	39,953,380	-	13min44s	-	15min29s

Efficiency – computation times

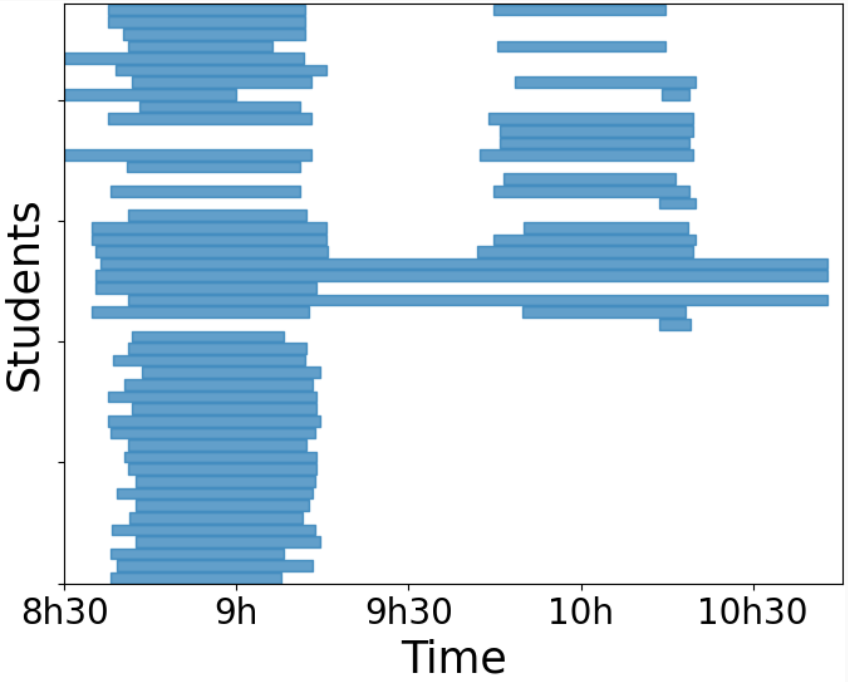
		$k = 3$		$k = 4$	
Link stream	**# links**	**DCPM**	**LSCPM**	**DCPM**	**LSCPM**
Households	2,136	1.5s	0.1s	1.0s	0.1s
Highschool	5,528	3.6s	0.1s	1.9s	0.1s
Infectious	44,658	10min49s	1.4s	6min12s	3.3s
Foursquare	268,472	3h01min	9.2s	2h28min	43s
Wikipedia	39,953,380	-	13min44s	-	15min29s

Consistency with metadata

Highschool: 70% of communities are within one class, 23% within two classes, 6% within three classes, 1% within four classes.

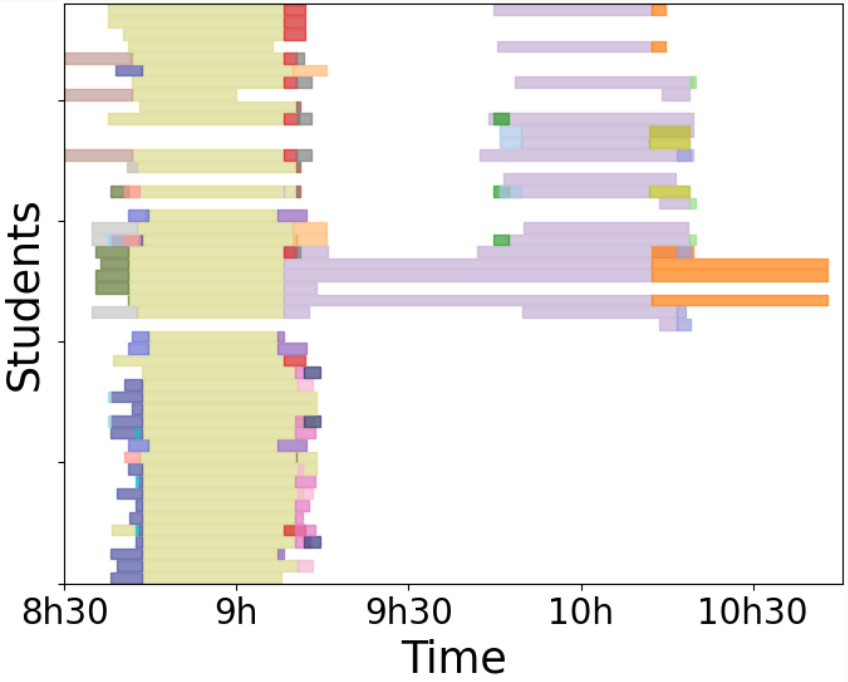
> Example of a LSCPM community

A highschool LSCPM community



> LSCPM communities are union of DCPM communities

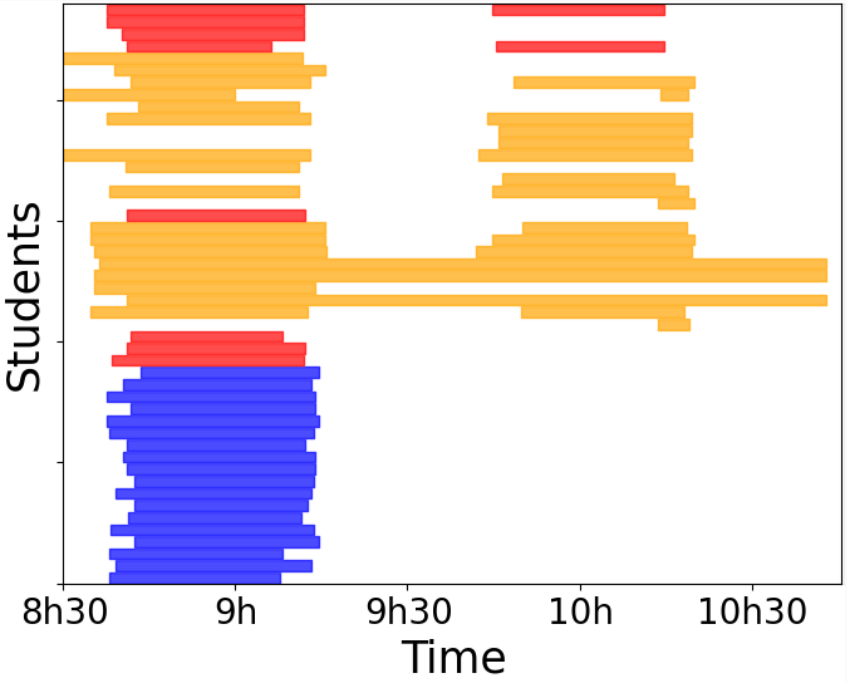
A highschool LSCPM community



Colors = DCPM communities.

→ Aggregates more information over time.

A highschool LSCPM community



Colors = classes.

→ Interpretation with metadata.

4 - Conclusion

4 - Conclusion

→ **Contributions**

- New definition of k-clique in link streams;
- New algorithm for k-clique enumeration;
- $\Rightarrow$ apply clique percolation to get temporal communities.

4 - Conclusion

→ **Contributions**

- New definition of k-clique in link streams;
- New algorithm for k-clique enumeration;
- $\Rightarrow$ apply clique percolation to get temporal communities.

→ **Communities obtained VS state of the art**

- Faster computed;
- Better aggregated in time.

LSCPM: finding communities in link streams

Thanks for your attention! Any questions?

Code available at:

`https://gitlab.lip6.fr/baudin/lscpm`

Alexis Baudin – `alexis.baudin@lip6.fr`