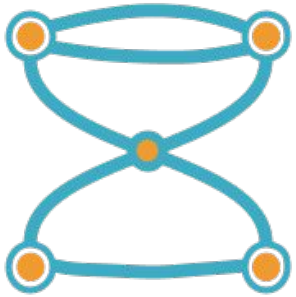


NetworkX Temporal

- **Tutorial: https://github.com/nelsonaloysio/c72h**
- **PyPI:** https://pypi.org/p/networkx-temporal
- **Documentation:** https://networkx-temporal.readthedocs.io
- **Repository:** https://github.com/nelsonaloysio/networkx-temporal
- **Preprint:** https://nelsonaloysio.github.io

NetworkX-Temporal

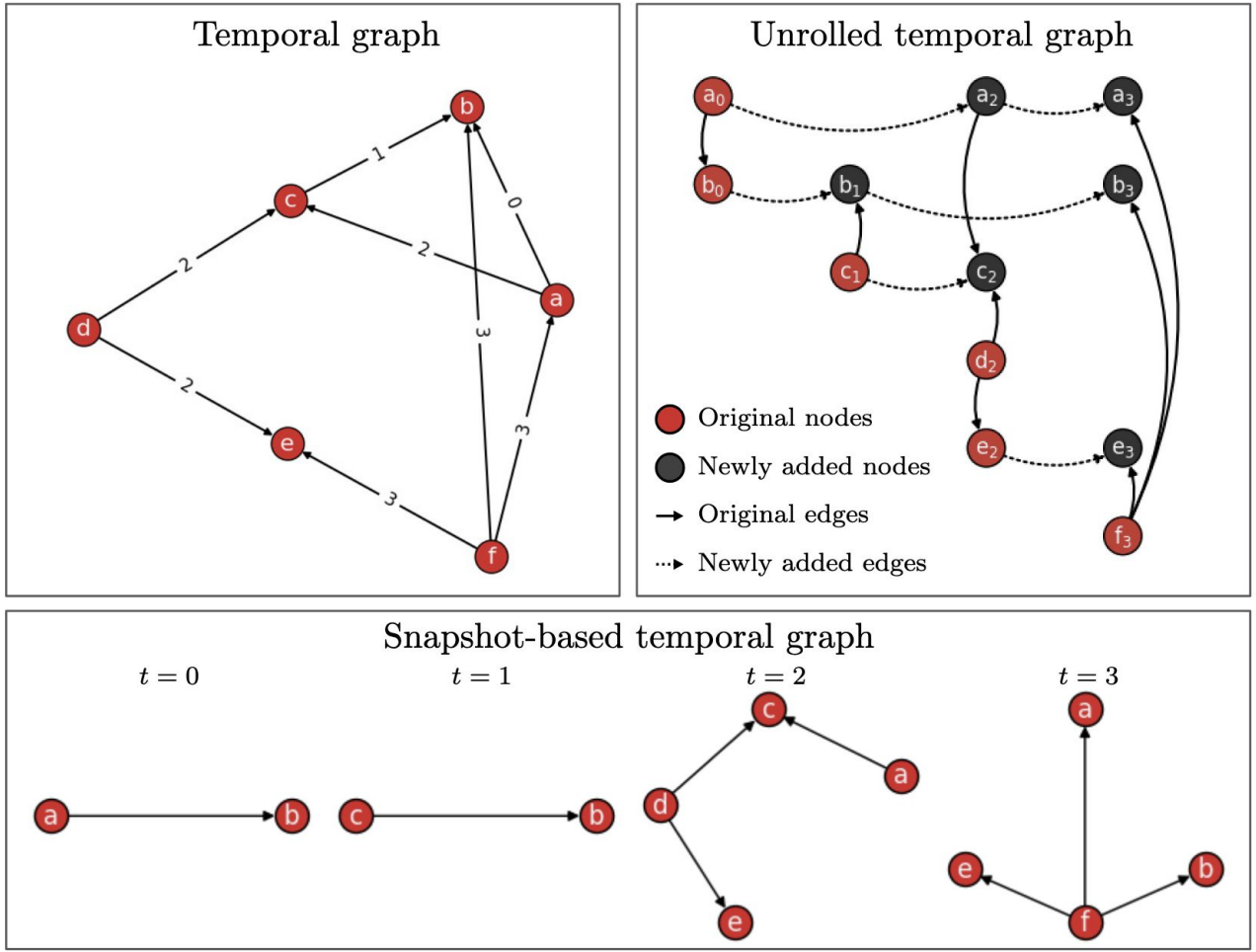
NetworkX-Temporal extends the NetworkX library to dynamic graphs, i.e., temporal network data.

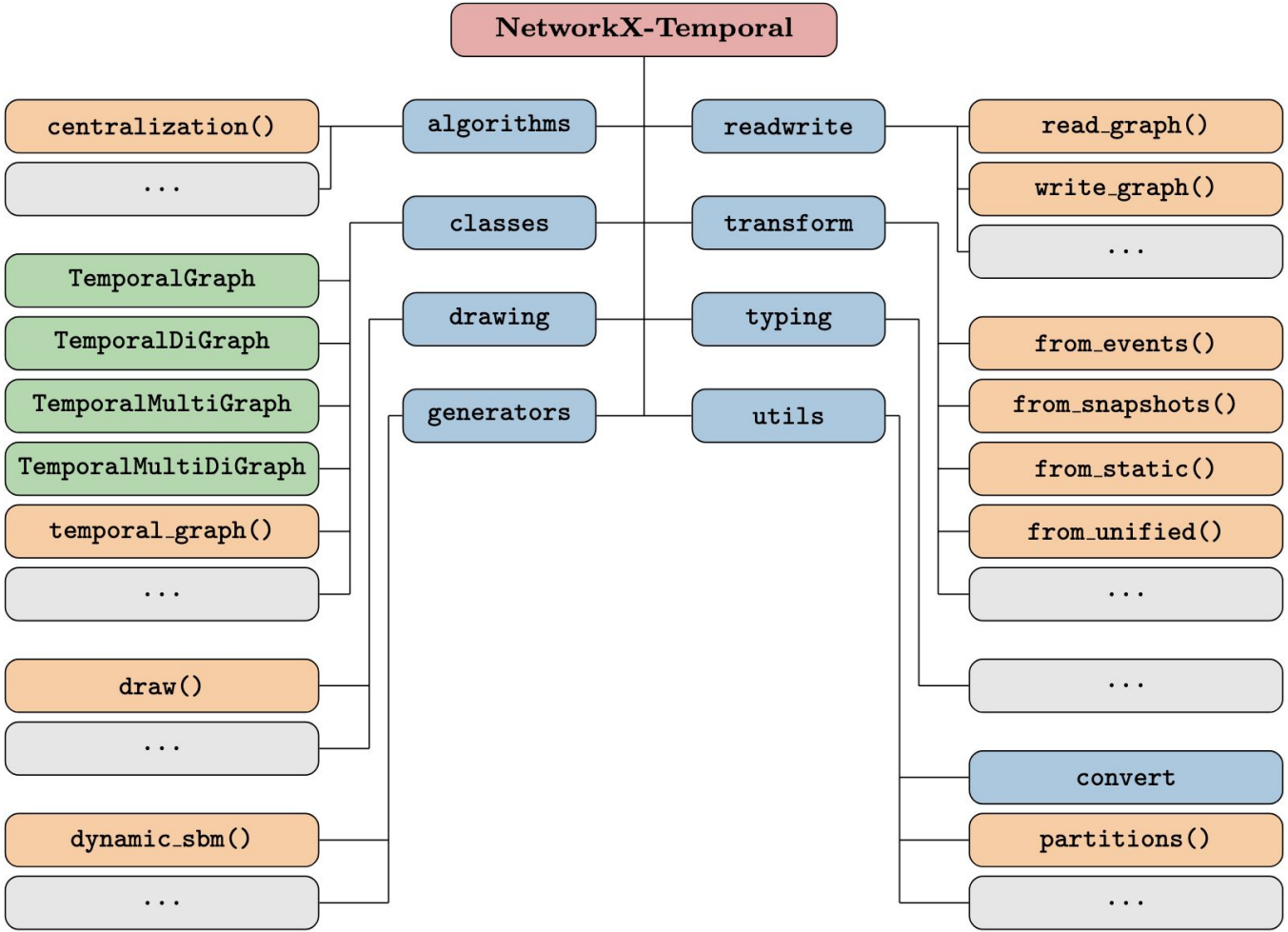
This package provides new `TemporalGraph` classes, which inherit NetworkX's graph classes and implement additional functions to manipulate temporal data within. Among others, it provides ways to `slice()` a graph into snapshots and `transform` or `convert` it to other libraries and formats.

Install

The package supports **Python 3.7+** and is readily available from PyPI:

```
$ pip install networkx-temporal
```


Temporal graph
b
c
a
d
e
f
1
0
2
2
2
3
3
Unrolled temporal graph
a_0
a_2
a_3
b_0
b_1
b_3
c_1
c_2
d_2
e_2
e_3
f_3
Original nodes
Newly added nodes
Original edges
Newly added edges
Snapshot-based temporal graph
$t = 0$
$t = 1$
$t = 2$
$t = 3$
a
b
c
b
c
a
d
e
a
e
f
b


NetworkX-Temporal
algorithms
classes
drawing
generators
readwrite
transform
typing
utils
centralization()
...
TemporalGraph
TemporalDiGraph
TemporalMultiGraph
TemporalMultiDiGraph
temporal_graph()
...
draw()
...
dynamic_sbm()
...
read_graph()
write_graph()
...
from_events()
from_snapshots()
from_static()
from_unified()
...
...
convert
partitions()
...

Convert and transform graphs

This package allows to transform a `TemporalGraph` between different graph representations:

- Static graphs: flattened or aggregated version of the temporal graph.
- Snapshot-based temporal graphs: a list of node- or edge-level snapshots.
- Event-based temporal graphs: a sequence of edge-level events (interactions).
- Unified temporal graphs: a single graph with time-stamped nodes and edges.

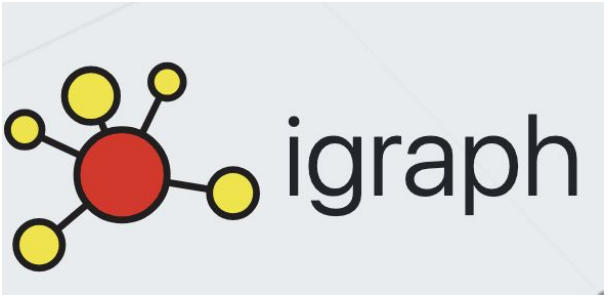
```
>>> G = TG.to_static()        # TG = tx.from_static(G)
>>> STG = TG.to_snapshots()   # TG = tx.from_snapshots(STG)
>>> ETG = TG.to_events()      # TG = tx.from_events(ETG)
>>> UTG = TG.to_unified()     # TG = tx.from_unified(UTG)
```

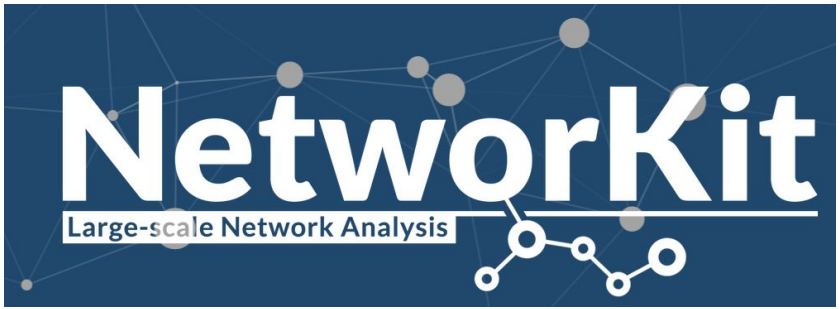
In addition, both static and temporal graphs may be converted to the following graph formats:


DGL
DEEP
GRAPH
LIBRARY


graph-tool

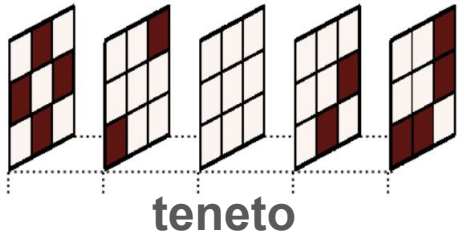

DyNetx


igraph


NetworKit
Large-scale Network Analysis


SNAP


STELLAR
GRAPH


teneto


72h

Data Science

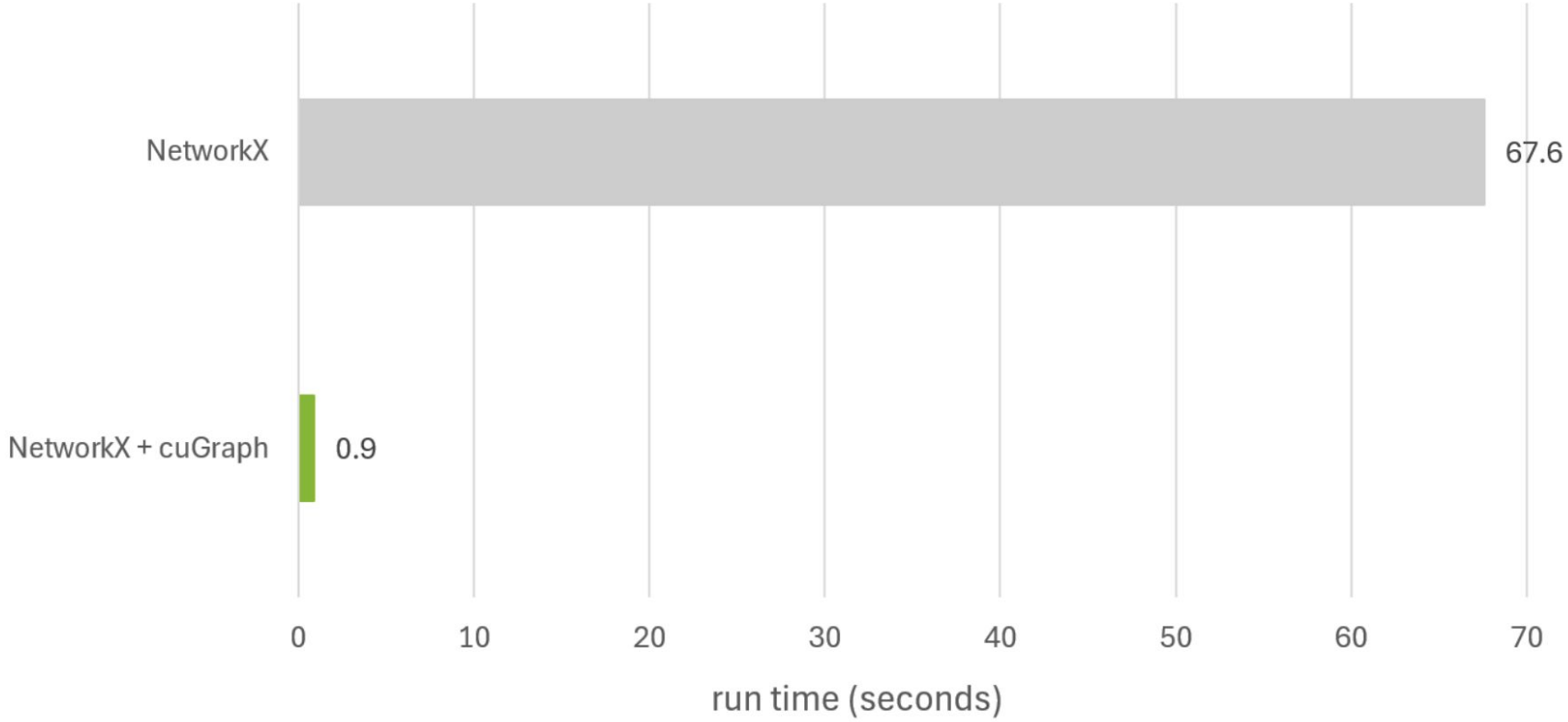
English

NetworkX Introduces Zero Code Change Acceleration Using NVIDIA cuGraph

Oct 22, 2024

By Rick Ratzel and Jenn Yonemitsu

PageRank algorithm used to compute values for a citation graph of U.S. patents (4M nodes, 16M edges) is 70x faster than NetworkX on CPU



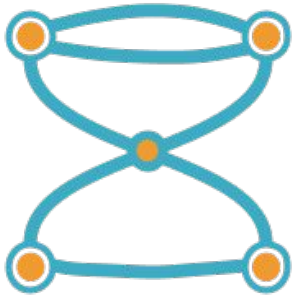
File: demo.ipy

```
import pandas as pd
import networkx as nx

url = "https://data.rapids.ai/cugraph/datasets/cit-Patents.csv"
df = pd.read_csv(url, sep=" ", names=["src", "dst"], dtype="int32")
G = nx.from_pandas_edgelist(df, source="src", target="dst")

%time result = nx.betweenness_centrality(G, k=10)
```

Output:

```
user@machine:/# ipython demo.ipy
CPU times: user 7min 36s, sys: 5.22 s, total: 7min 41s
Wall time: 7min 41s

user@machine:/# NX_CUGRAPH_AUTOCONFIG=True ipython demo.ipy
CPU times: user 4.14 s, sys: 1.13 s, total: 5.27 s
Wall time: 5.32 s
```

- **Software:** NetworkX 3.4.1, cuGraph/nx-cugraph 24.10
- **CPU:** Intel Xeon Gold 6128 CPU @ 3.40GHz 45GB RAM
- **GPU:** NVIDIA Quadro RTX 8000 50GB RAM



NetworkX Temporal

- **Tutorial: https://github.com/nelsonaloysio/c72h**
- **PyPI:** https://pypi.org/p/networkx-temporal
- **Documentation:** https://networkx-temporal.readthedocs.io
- **Repository:** https://github.com/nelsonaloysio/networkx-temporal
- **Preprint:** https://nelsonaloysio.github.io