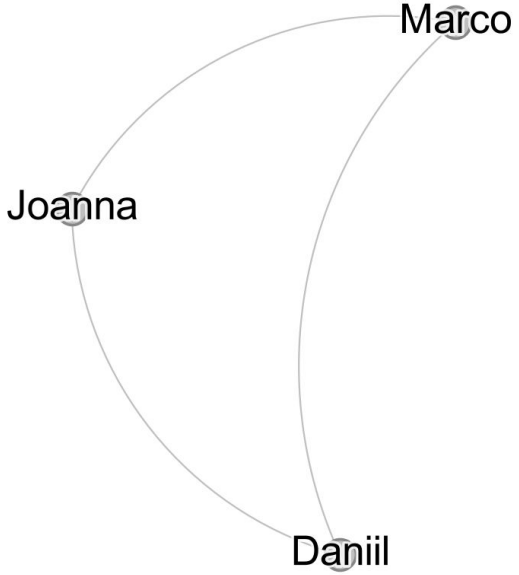
How do we store networks in machine-readable form?

a CSV table with network edges is a popular option:

Daniil,Joanna

Joanna,Marco

Daniil,Marco

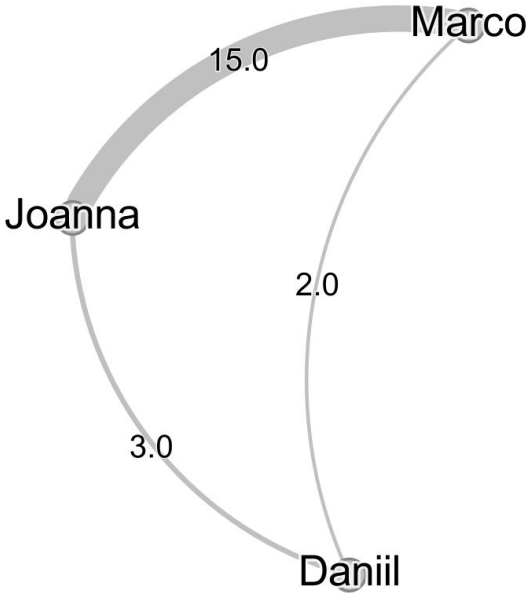


Adding weight information:

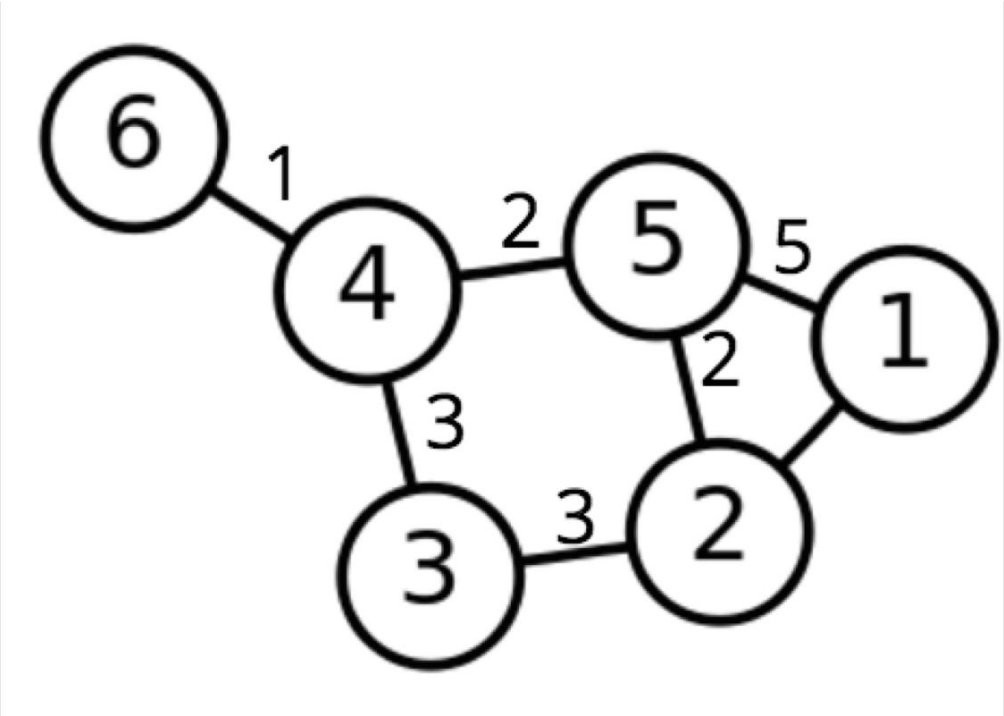
Daniil,Joanna,3

Joanna,Marco,15

Daniil,Marco,2



BTW we just introduced a weighted network (graph)

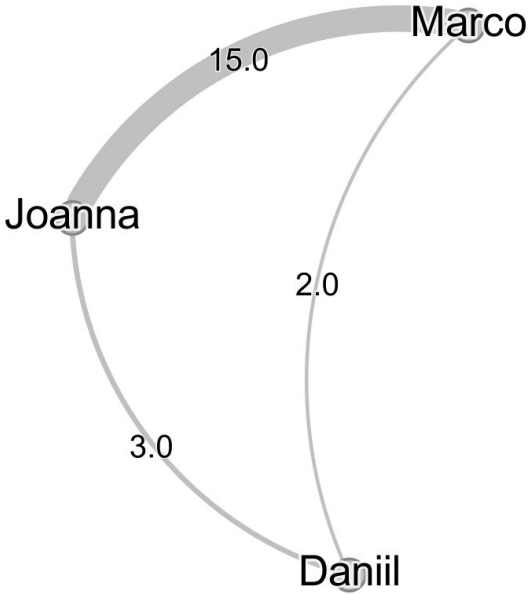


Adding weight information:

Daniil,Joanna,3

Joanna,Marco,15

Daniil,Marco,2



Another format: GraphML

XML-based format for network storage

```
<graphml>
  <graph id="G" edgedefault="undirected">
    <node id="n0"/>
    <node id="n1"/>
    <edge id="e1" source="n0" target="n1"/>
  </graph>
</graphml>
```

Another format: GEXF

Graph Exchange XML Format is another XML-based format, but this one is more complex and feature-rich. It allows for storing lots of metadata and visualisation parameters

```
<gexf>
    <meta lastmodifieddate="2018-03-20">
        <creator>Gexf.net</creator>
        <description>A hello world! file</description>
    </meta>
    <graph mode="static" defaultedgetype="directed">
        <nodes>
            <node id="0" label="Hello" />
            <node id="1" label="Word" />
        </nodes>
        <edges>
            <edge id="0" source="0" target="1" />
        </edges>
    </graph>
</gexf>
```

But a CSV table with network edges is the simplest

Daniil,Joanna,3

Joanna,Marco,15

Daniil,Marco,2

But a CSV table with network edges is the simplest. Here is what the CSV format accepted by Gephi looks like:

Source,Target,Weight

Daniil,Joanna,3

Joanna,Marco,15

Daniil,Marco,2

How do I obtain/produce such data?

There are graph datasets available in ready-to-use form:

- github.com/gephi/gephi/wiki/Datasets (datasets provided by the creators of Gephi)
- networks.skewed.de/ (a large repository of various networks — from dolphin friendships to Bitcoin transactions and scientific co-authorship)
- dracor.org (4330+ networks of theatre plays in 17 languages)

If you want to model your data as network on your own:

- First ask yourself, what is it you want to operationalize with a network
- Formalize the network for yourself:
  - What is a **node** (vertex) in our case?
  - What is an **edge** (connection)?
- Encode the edges in a machine-readable form — in other words, extract the network from a source (which can be text, historical objects, other types of sources, observations of life etc.)

Once you have a formalisation in your mind… there are several ways to actually extract the network

Many researchers build pipelines for automatically extracting networks from unstructured data

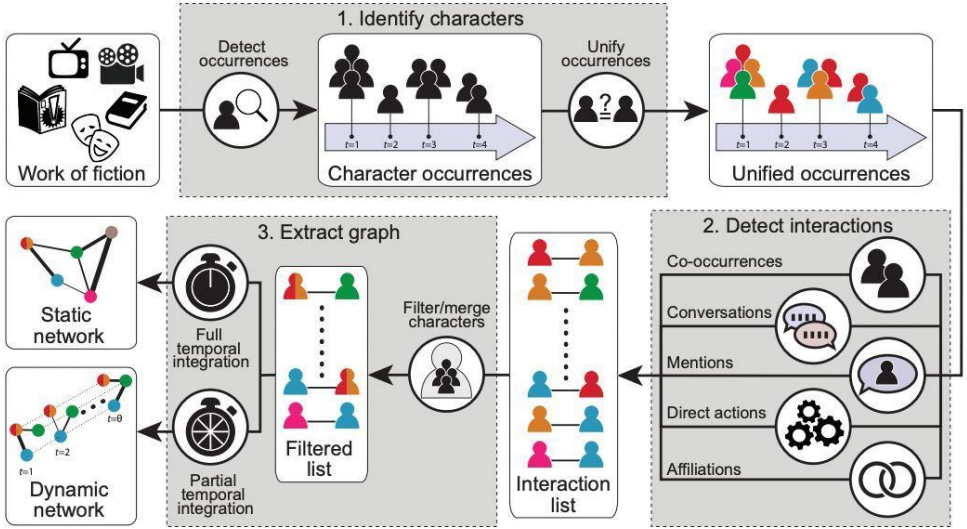
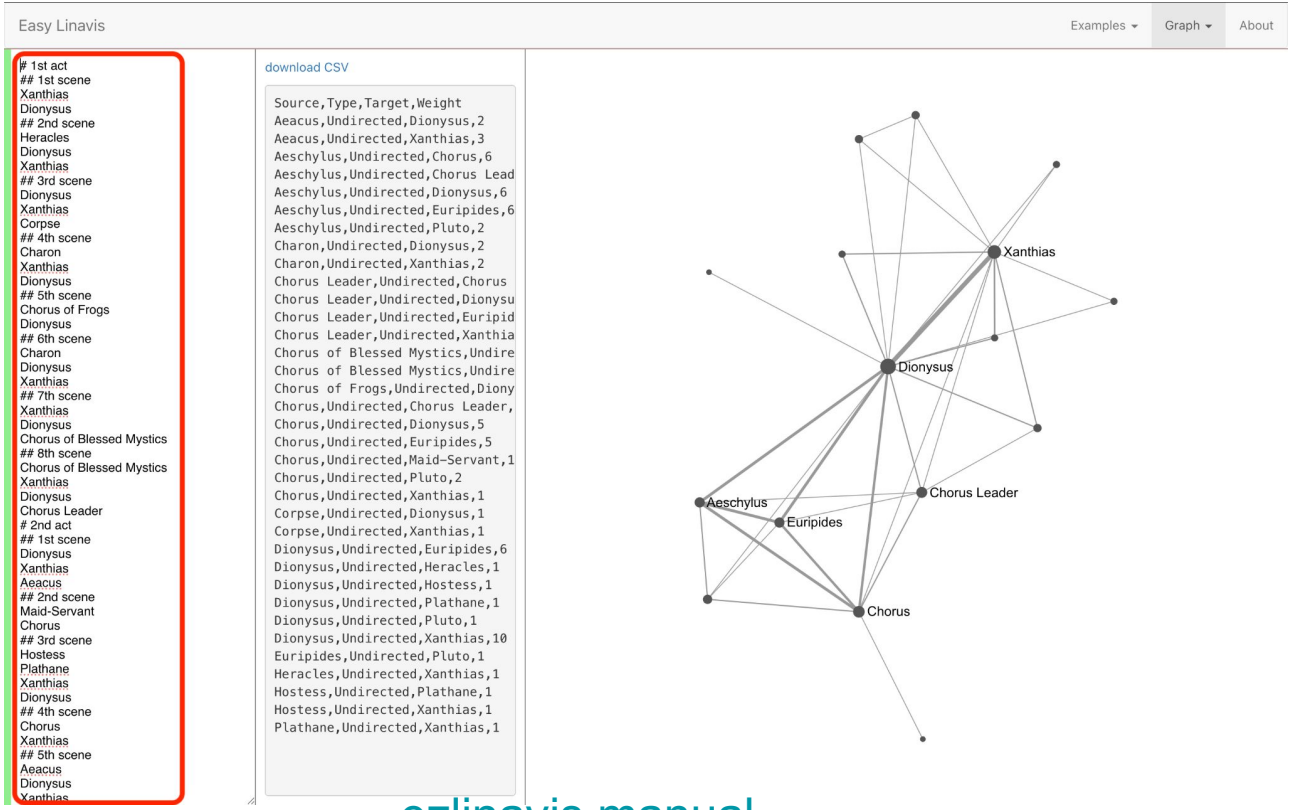


Fig. 1. Overview of the generic character network extraction process. Figure available at 10.6084/m9.figshare.7993040 under CC-BY license.

Labatut, Vincent, and Xavier Bost. 2019. 'Extraction and Analysis of Fictional Character Networks: A Survey'. ACM Computing Surveys 52 (5): 1–40. https://doi.org/10.1145/3344548.

Many encode by hand in a spreadsheet or in EzLinaVis:



ezlinavis manual

Let us play around: ezlinavis.dracor.org

(you can also find it on dracor.org under ‘Tools’ menu)

ezlinavis manual