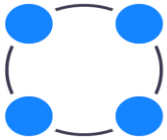


ИНСТИТУТ
БИОИНФОРМАТИКИ

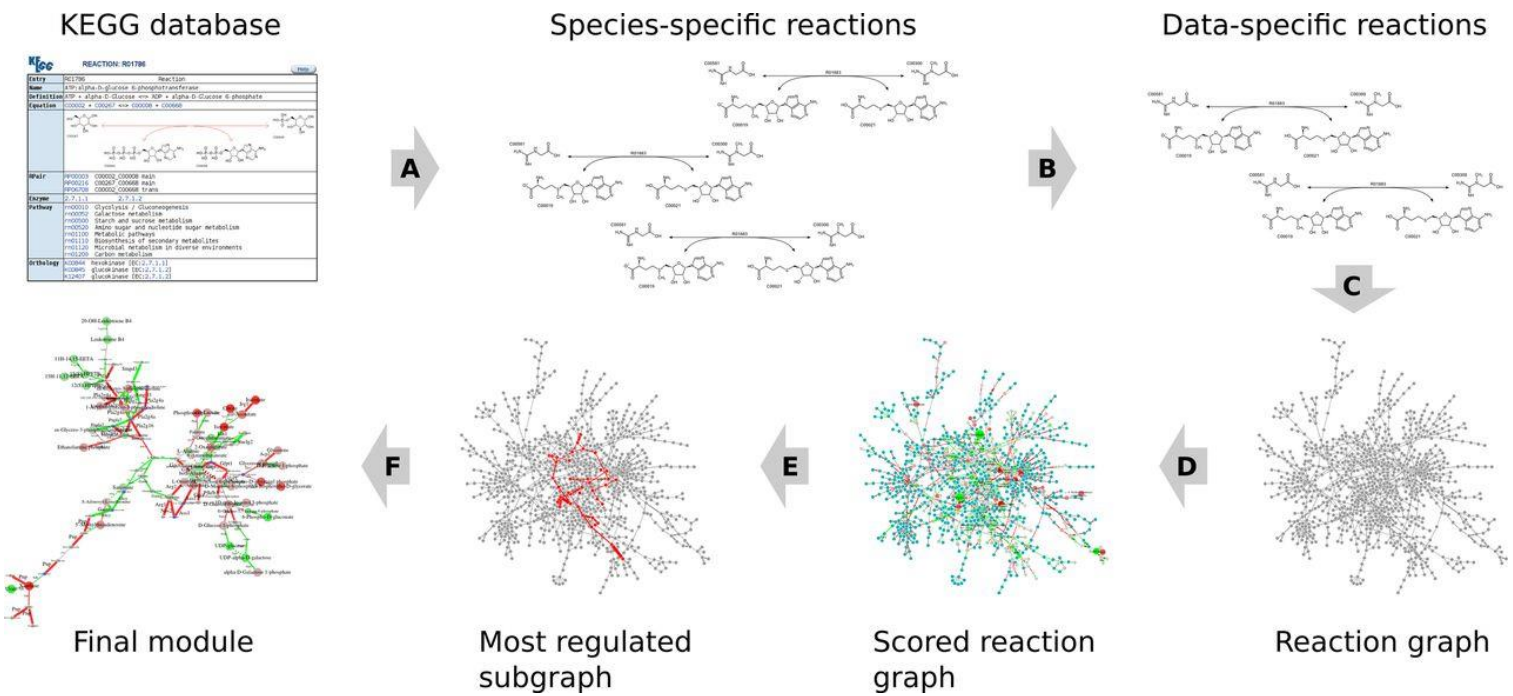
Automatic visualization of metabolic modules

(graphs from *gatom* package)

Student: Anastasiia Gainullina
Supervisor: Alexey Sergushichev

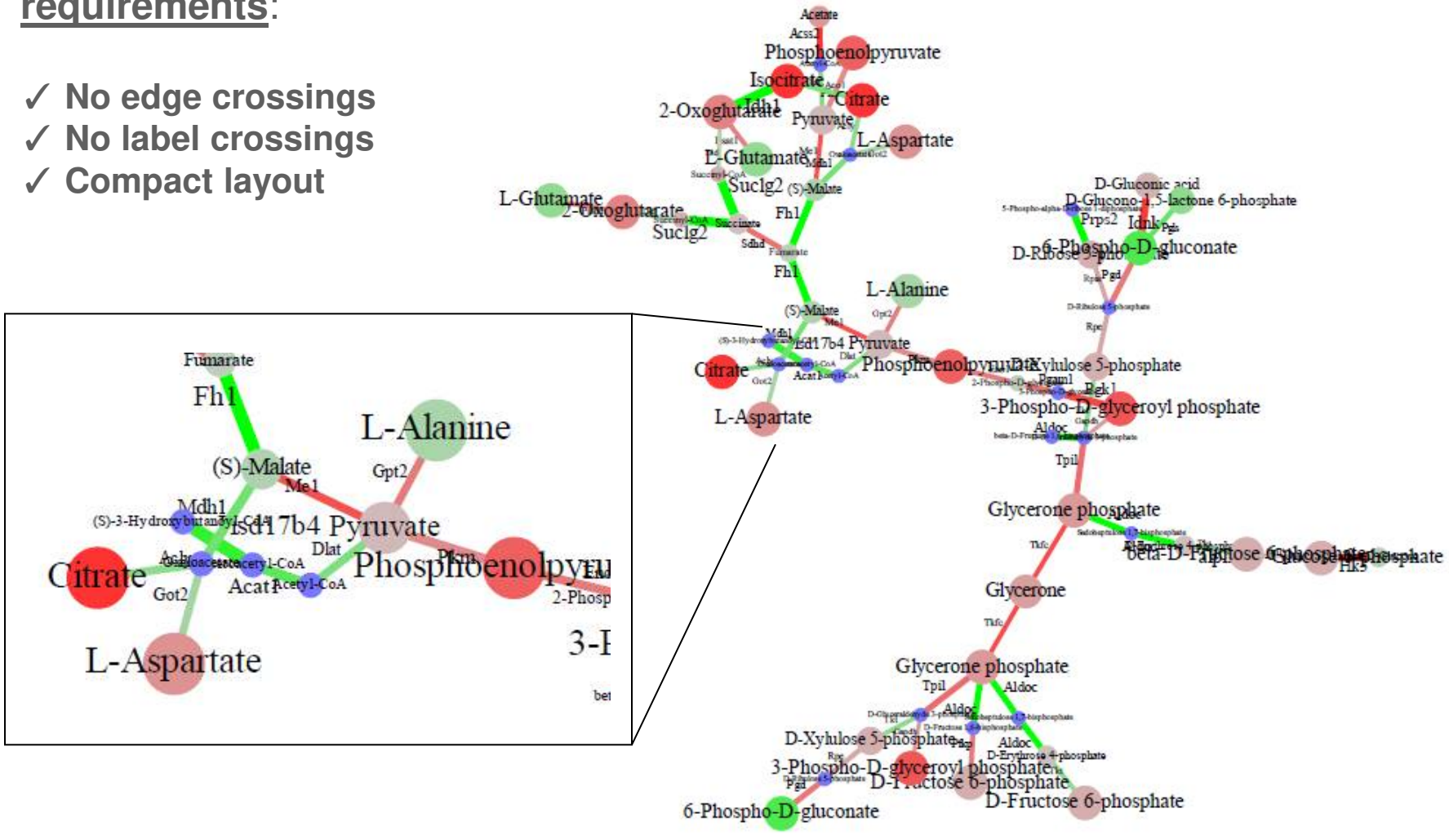
September
2017

Metabolic module (*gatom* graph) is marked out as a most **regulated subgraph** of a total **data-related reaction graph**:



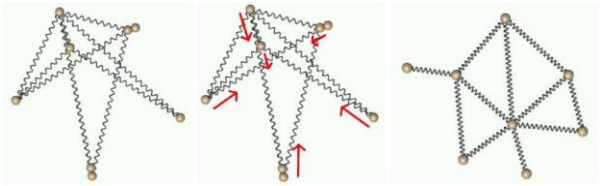
To build an algorithm that would draw *gatom* graphs satisfying the following **requirements**:

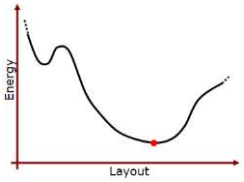
- ✓ **No edge crossings**
- ✓ **No label crossings**
- ✓ **Compact layout**



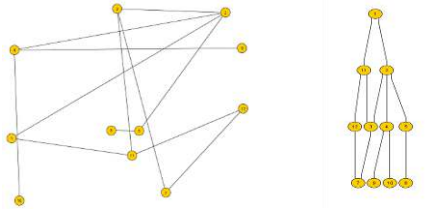
Some **families of algorithms** producing aesthetically pleasant **graph layouts:**

1. Force-directed (e.g., Kamada and Kawai)

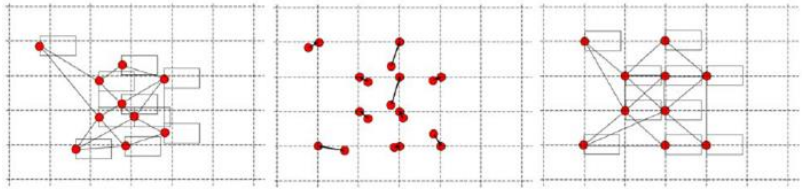




1. Hierarchical (Sugiyama's method)



1. Genetic

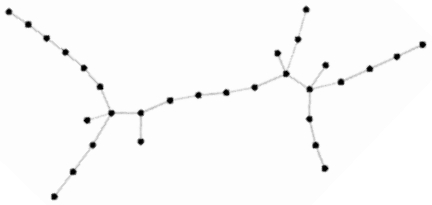


1. …

No edge crossings
No label crossings
Compact layout

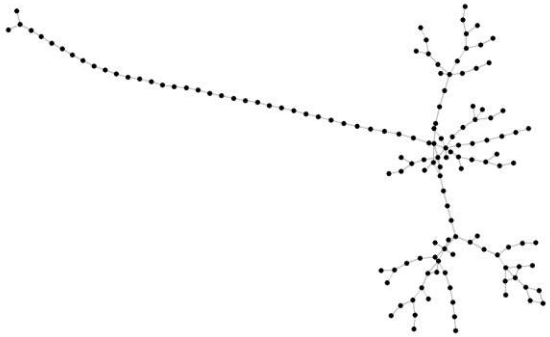
In general, diversity of *gatom* graphs can be divided into **three large groups**:

- "Simple"
  shabaana_tb765, graph3



- "Complex"
  shabaana_tb765, graph1

- "Long and complex"
  shabaana_tb765, graph4



I

- No edge crossings
- Compact layout

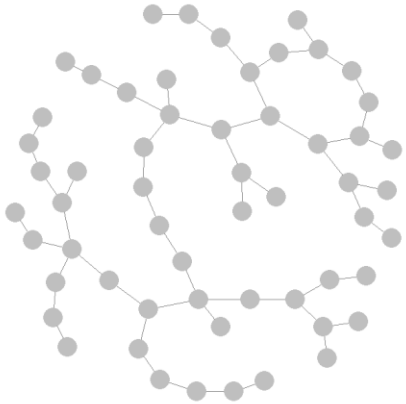
➡ Optimal choice of existing implementations

II

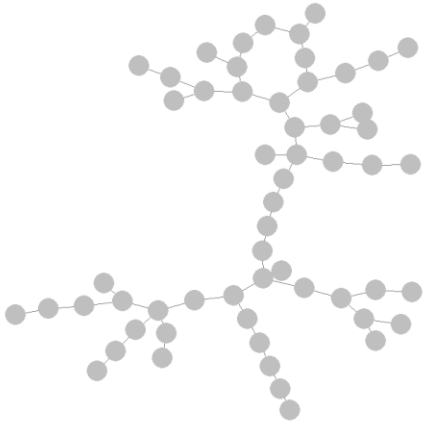
- No node and edge label crossings

➡ Realize special implementation of force-directed algorithm considering edge labels

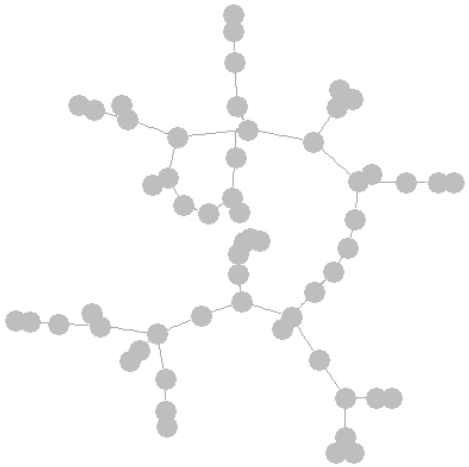
Force-directed algorithms (*igraph*, *sna* and *ForceAtlas2* R packages) produce **compact** and **typical** layouts with **no edge crossings**:



Fruchterman-Reingold =>
for **“Simple”** graphs



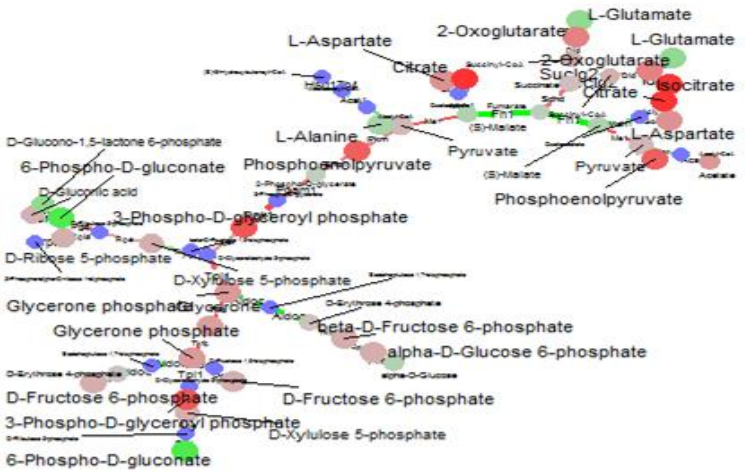
Kamada and Kawai =>
for **“Complex”** graphs



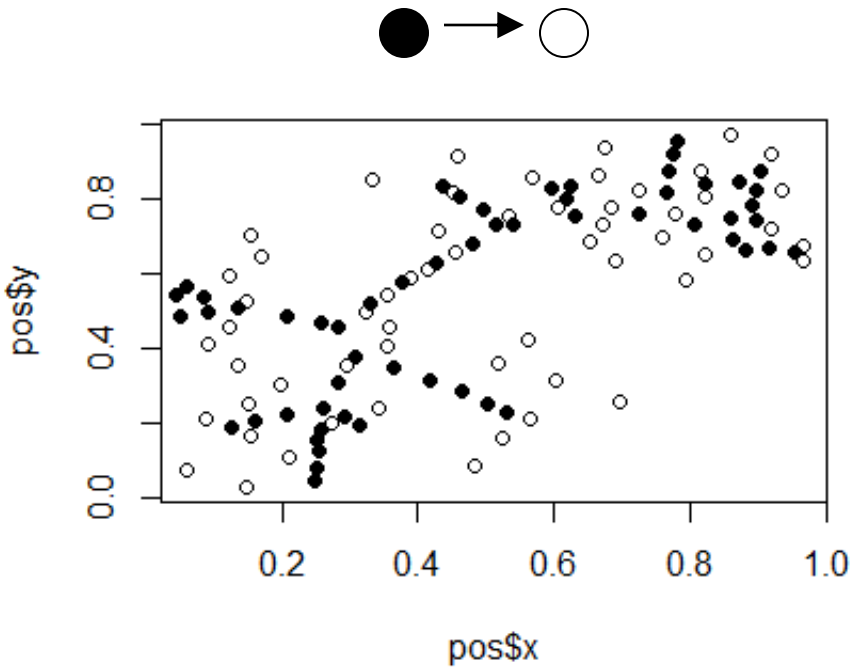
ForceAtlas2 (Gephi)
@ param gravity =>
for **“Long and complex”** graphs

-the same graph-

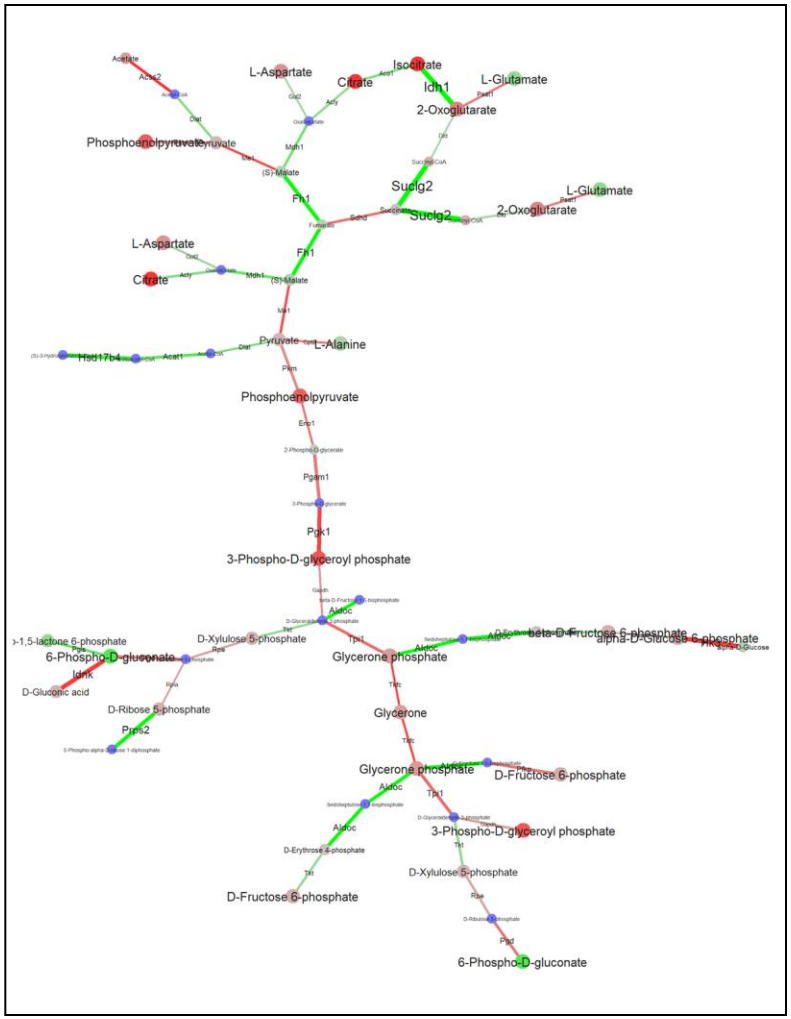
To get rid of **label crossings** let us find optimal positions of all **labels** and put **nodes** to their **centers**:

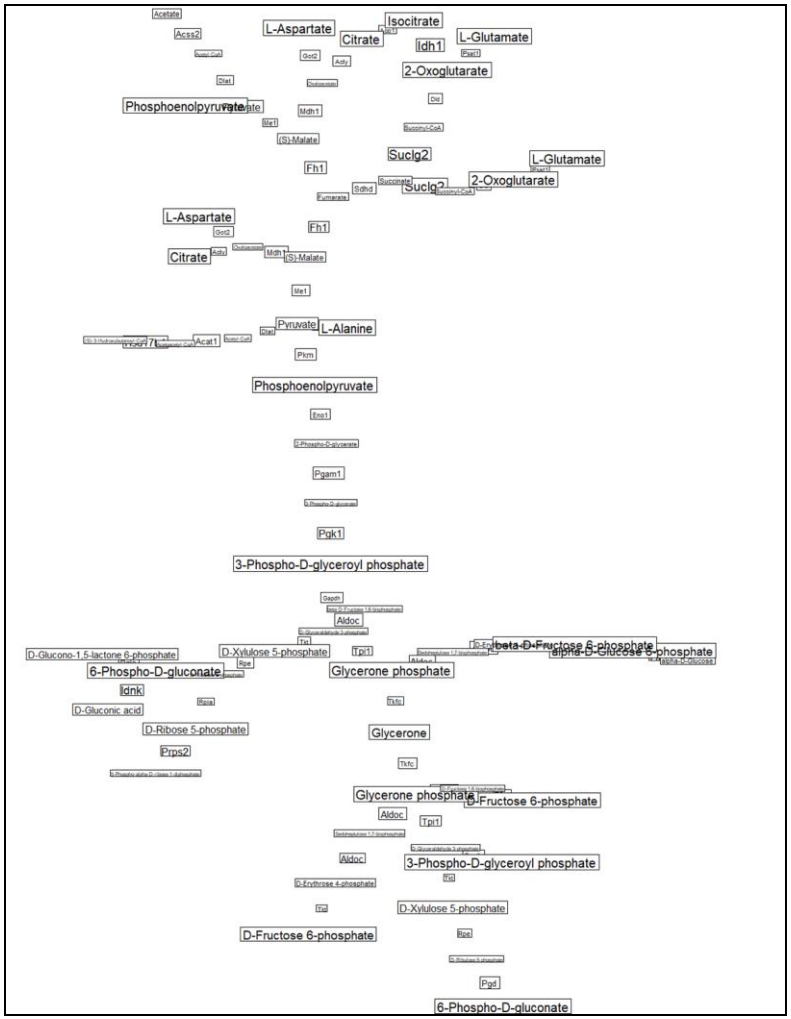


output of *ggrepel* algorithm



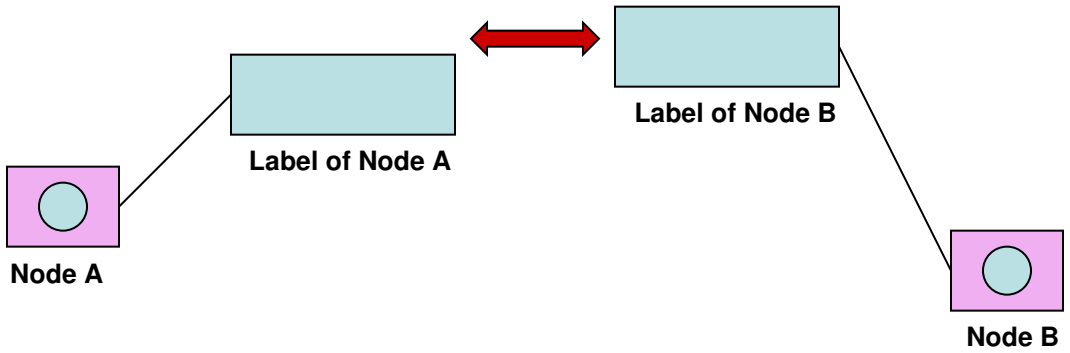
First of all, **labels** were represented as **boxes** with some **padding**:





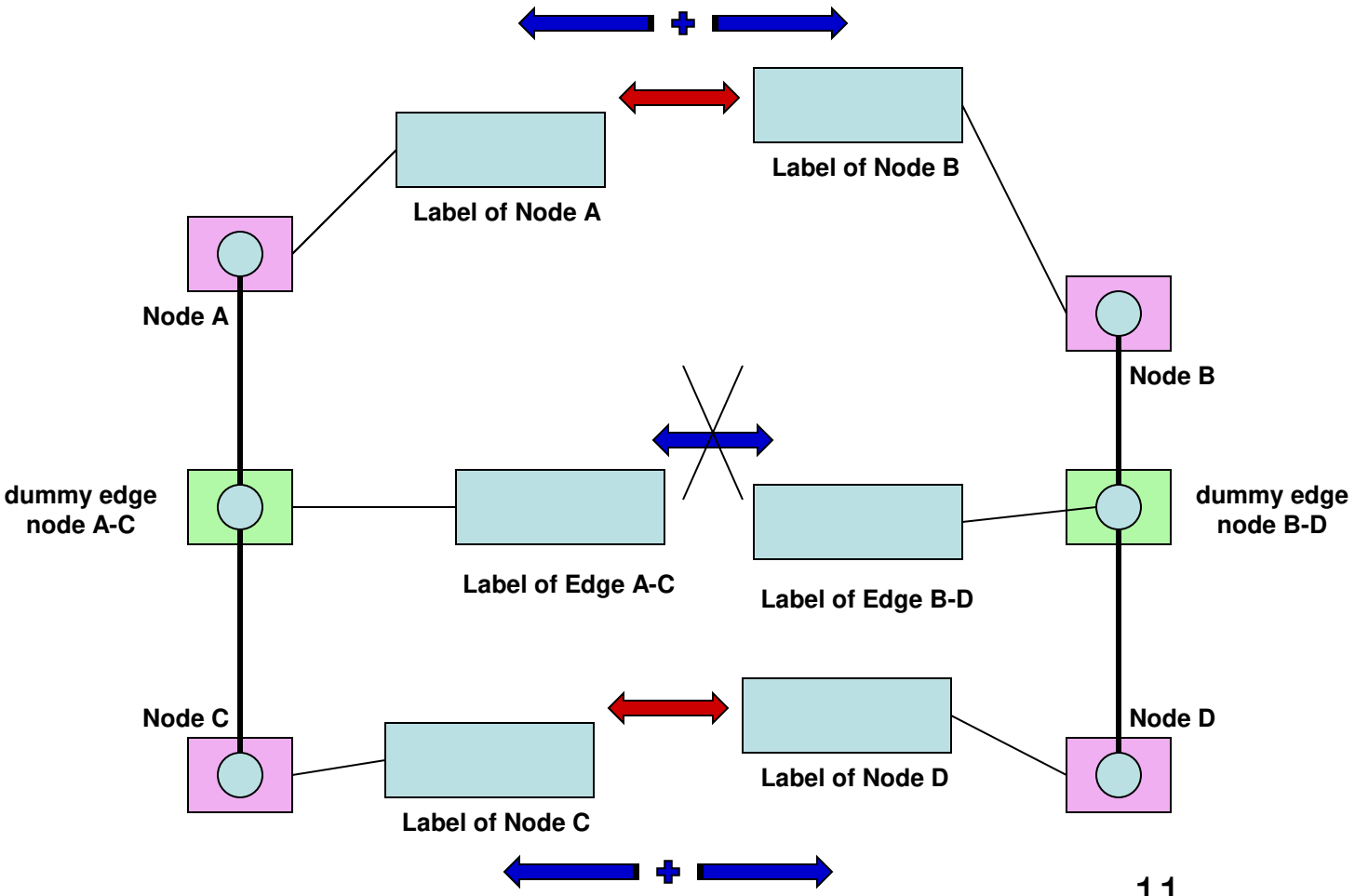
Realized **“label algorithm”** inherited principles of typical **physical algorithm**:

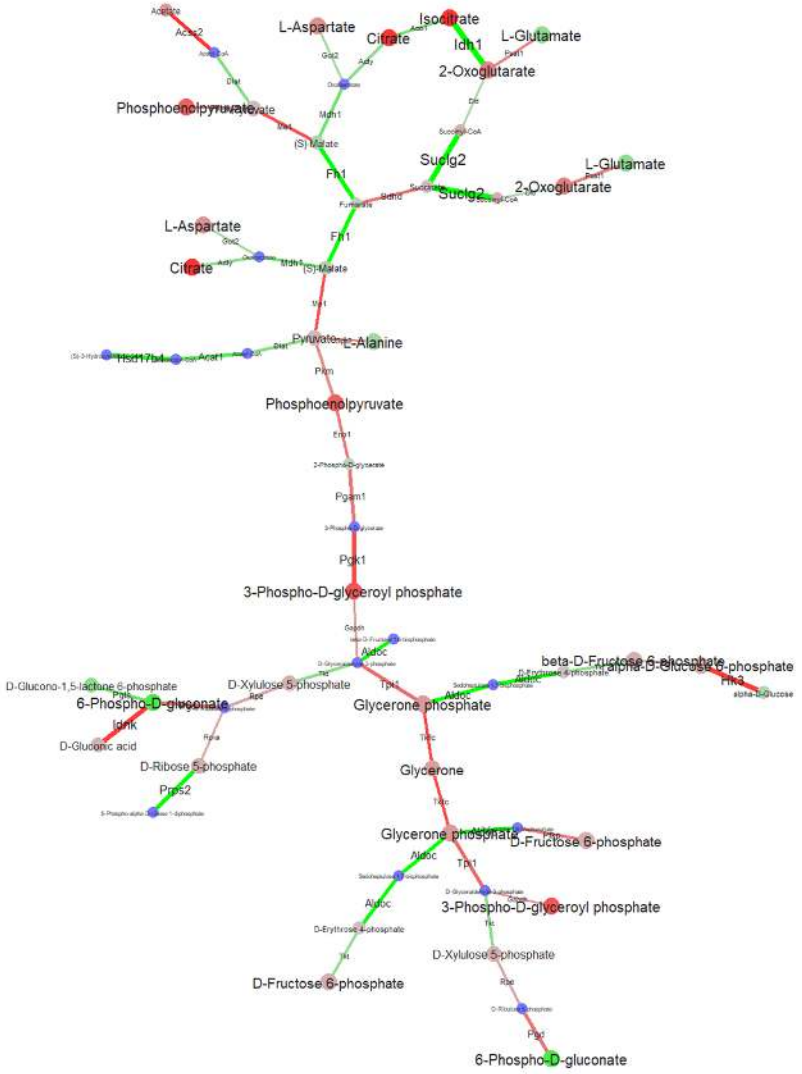
- **Force ~ label intersection**



- $$\vec{F} = k \cdot \frac{q_1 \cdot q_2}{r^2} \cdot \frac{\vec{r}}{|r|}$$

How about saving **linearity of edges**?





https://github.com/anastasiiaNG/graph_drawing

Thank you for attention!