

The background of the slide features a complex network visualization. It consists of numerous nodes, represented by colored circles in shades of blue, orange, red, purple, and grey, connected by a dense web of thin, dark grey lines. The nodes are distributed across the frame, with a higher concentration on the right side. The overall aesthetic is technical and data-driven, typical of scientific software interfaces.

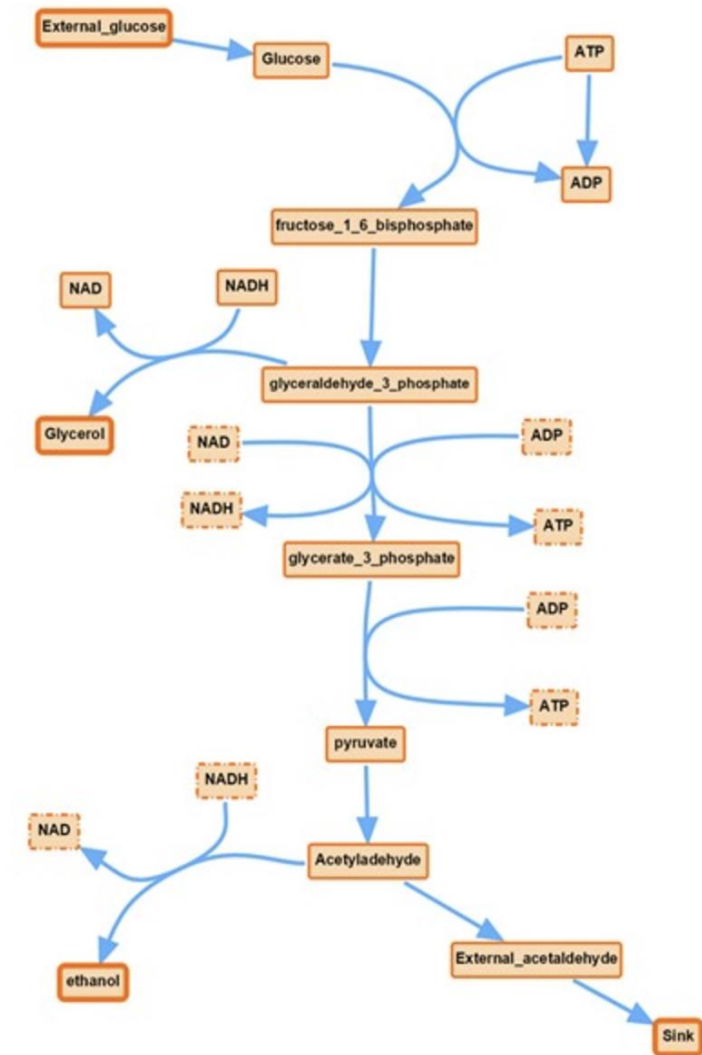
SBML VISUALIZER

Priyanka Talur

BACKGROUND

- Systems Biology Markup Language (SBML) [1]
- Used to visualize chemical reaction networks
- Works in conjugation with Tellurium (modeling software)

A



<https://doi.org/10.1093/bioinformatics/btac730>.

PROBLEM STATEMENT

- The model doesn't show the difference between fluxes
- My project aims to visualize more of that

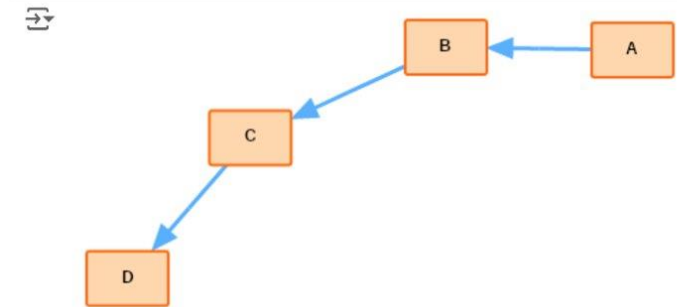
```
# Define and load a model
r = te.loada('''
A -> B; v
B -> C; v
C -> D; v
v = 0
''')

# Load the model into SBMLDiagrams
df = SBMLDiagrams.load(r.getSBML())

# Apply autolayout to arrange species and reactions
df.autolayout()

# Set line thickness for specific reactions

# Draw the model with specified line widths
df.draw()
```



<https://doi.org/10.1093/bioinformatics/btac730>.

PROJECT: SB VISUALIZER

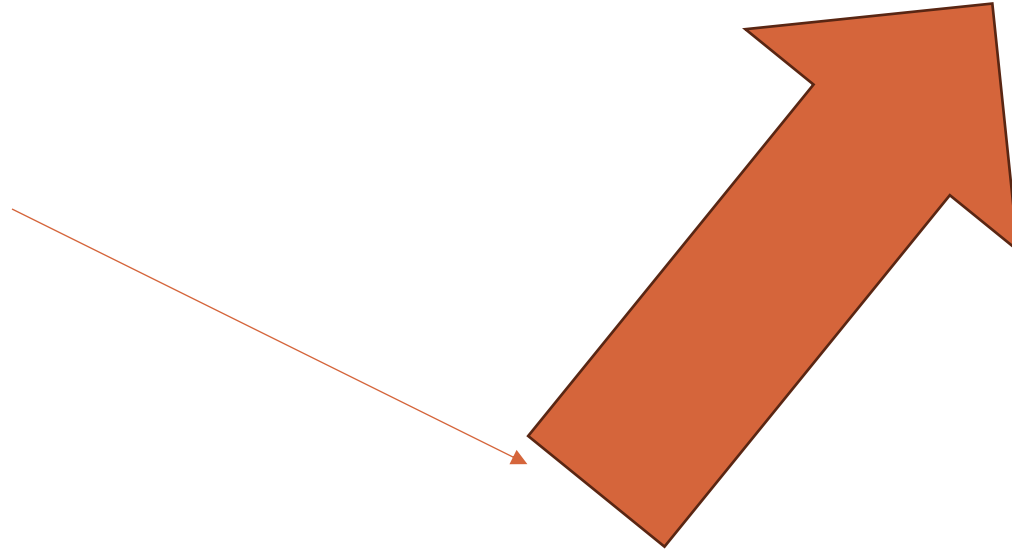
- My project aims to illustrate the flux of a species
- This is something that is not a feature of SBML
- SBML shows arrows from one species to another
- My project aims to change the thickness of the arrow to show flux
- And a color gradient representing fluxes

USERS AND COMPONENTS

- The main users of my project will be existing Python programmers
- Tellurium users
- SBML users (can use features of SBML)

THICKNESS

- After flux calculated
- Calculate the max flux
- Scale all the fluxes



COLOR GRADIENT

- Color scheme to represent positive and negative fluxes
- Based on RGB (255, 255, 255)



CALCULATE GRADIENT

- Obtain the maximum flux out of all the reactions
- Divide the rest of the fluxes by that maximum flux
- Multiplied this by 255 for red/blue and 1-this by 255 for green (depending on positive/negative)

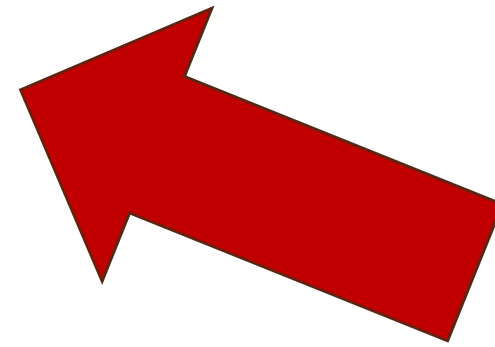
DEMO AND GITHUB

- The colors still don't show up in main
- Need to figure out how to make the colors show up
- <https://github.com/ptalur559/SBVisualizer>

FINISHED PRODUCT

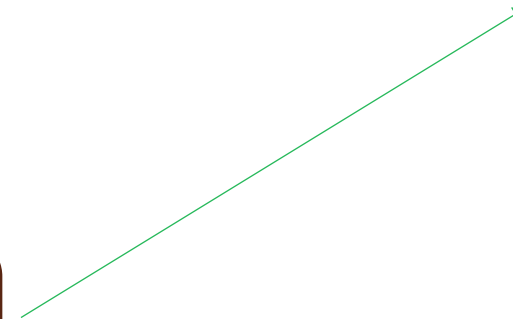
```
r = te.loada("""  
  J1: S1 -> S2; k1*S1;  
  J2: S2 -> S3; k2*S2;  
  S1 = 10; S2 = 0; S3 = 0; k1 = 0.1; k2 = 0.05;  
""")
```

S2



S3

S1



LESSONS LEARNED

- I learned a lot more about how to use GitHub and VSCode
- I learned more about SBML and what it's used for
- I need to figure out how to see the color changes in VSCode if I try it out in main

CITATIONS

[1] J. Xu, J. Jiang, and H. M. Sauro, "SBMLDiagrams: a python package to process and visualize SBML layout and render," *Bioinformatics*, vol. 39, no. 1, Nov. 2022, doi: <https://doi.org/10.1093/bioinformatics/btac730>.