**Proposed feature - documentation**

TODO: Make a nice report with this information. Create the skeleton of the code for the feature.

**Motivation and description of the feature**

Our feature is a visualization module that allows the user to observe different elements of a certain reaction system in multiple ways. The main feature of the module is a graph visualization of the current reaction system, created with graphviz. We will also provide functions for reaction rate visualization over time and over a range of temperatures, evolution of arrhenius coefficients, as well as plots of concentrations as a function of time. For that last element, we will implement an ODE solver to compute the concentrations. The plots will have multiple parameters for user customization, such as log scale options, selection of one or multiple species, visualization of several curves on one graph, real time visualization, etc. As a bonus feature, we will also try to provide an easy to use GUI to visualize the different elements.

**How will the feature fit in our code**

The feature will integrate with the Reaction and ReactionSystem class seamlessly, and will allow the user to input and XML file, generate ReactionSystem objects, and compute all the necessary visualizations with those.

**Modules that will be created**

* *Module Plotting:* will contain the different methods allowing for clean, seaborn powered graphs to be displayed. The curve plots and graphviz graphs will be generated here.
* *Module Math:* Will contain the ODE solver and other math helper functions for the project.
* *Module GUI:* Will contain the functions for generating the GUI and connecting with the rest of the functions of the project.

**Map out the methods you plan on implementing**

* plot\_concentration(ReactionSystem, time\_range, temp\_range, logscale=False, species = None)
* plot\_arrhenius(T\_range, E\_range)
* plot\_reaction\_rate(ReactionSystem, time\_range, temp\_range)
* plot\_network(ReactionSystem)
* plot\_path(ReactionSystem)
* plot\_revolution(ReactionSystem, time\_range)
* ODE\_solver()
* boot\_GUI()
* plot\_in\_GUI(plot)

**How will the user use our feature?**

The user will most likely load an xml file with reactions, generate ReactionSystem objects with that information, and then call our plotting functions to visulaize the evolution of the system. If the user prefers a GUI, the GUI will handle the XML loading, and will have a very straightforward button mapping that will generate the graphs and plots with a couple of clicks.

**External dependencies**

The external dependencies will probably be limited, but there will be some critical inclusions, such as graphviz, the module for graph plotting and .DOT file creation; itertools, a module for improved iterators; matplotlib, for plotting certain graphs, and probably other elements that will be determined as the project is constructed.

*Sondak: The proceeding requirements can be in a section of your code documentation called `Future Features` or some analogous section title.*

plot\_network(species = “all”, ReactionSystem, type) : plot the network graph of given species, default to be all.

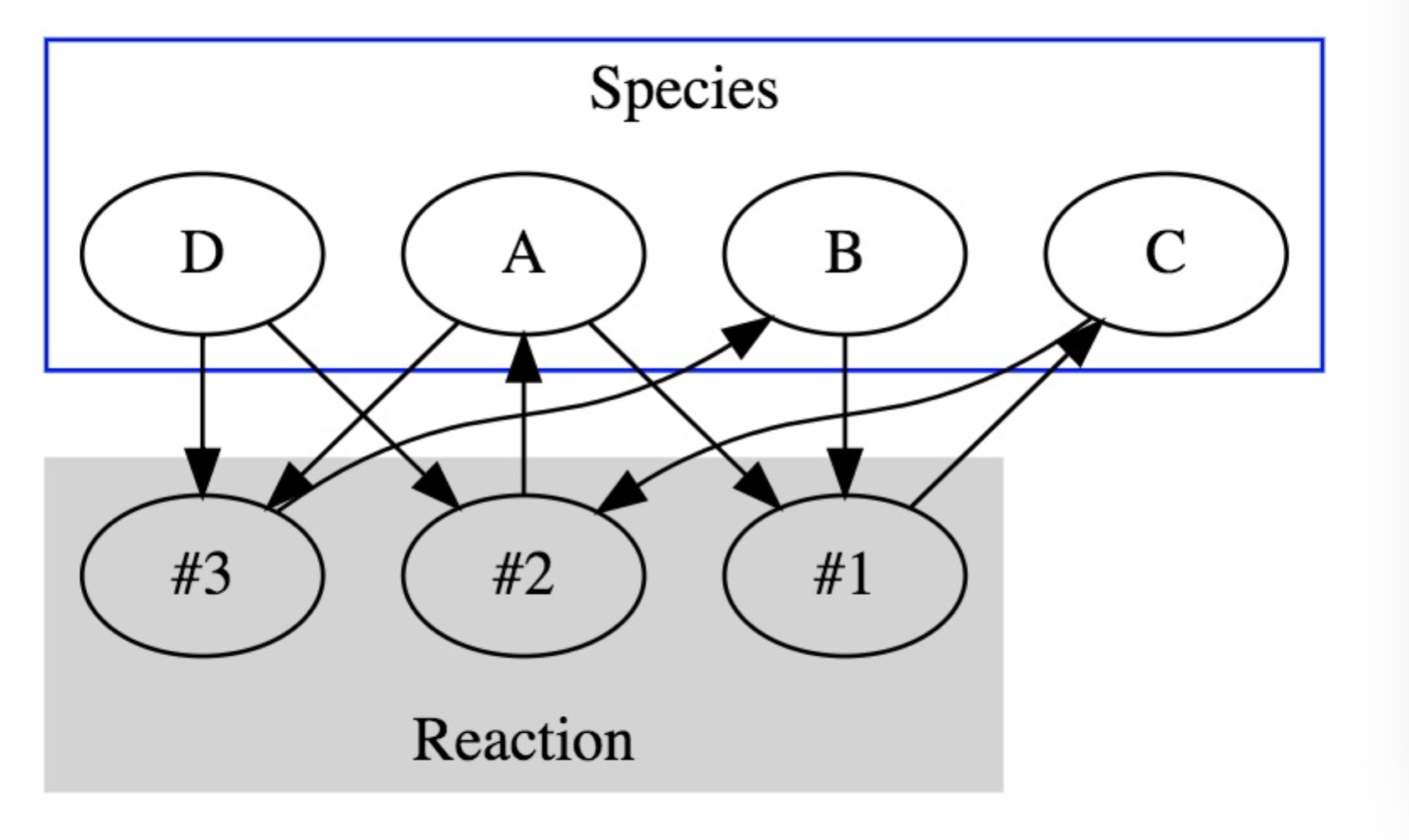
Type = “Bipartite”:

We construct a bipartite graph with species as nodes (u,1) in one side, and reactions as nodes (u,2) in another side.

Build edges (u,1) to (v,2) if edge u is the reactant of equation v.

Build edges (u,2) to (v,1) if edge v is the product of equation u.

From this graph, we could recompute the reaction equations.



#1: A + B = C

#2: C + D = A

#3: A + D = B

Type = “Hierarchical”:

We construct a network with species as nodes using graphviz.

Build undirected edges (u,v) if u and v are reactants in an equation.

Build directed edges (u,v) if v is u’s product in an equation.

Addtional option:

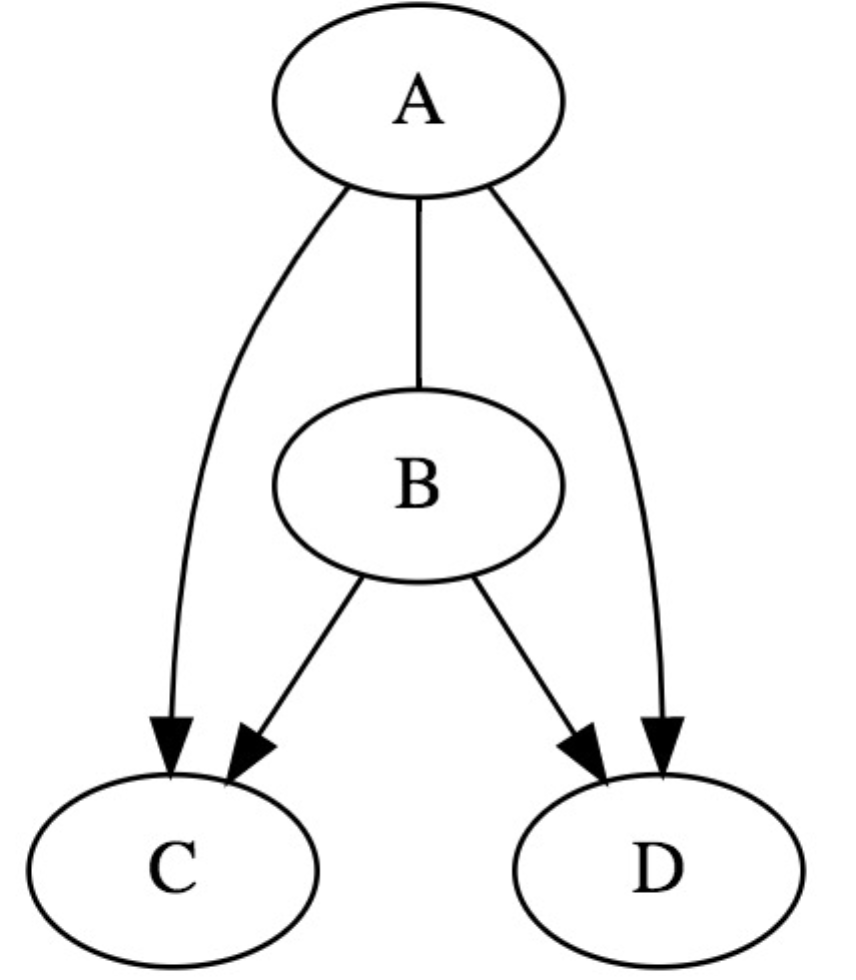
Set size (height/weight) of nodes as concentration rate of species.

Set weight of edges as progress rate in an equation.

Color edges if they are from the same equation.

We may encounter imbalanced parameter if we try to set size/weight to the graph. If so, we could provide an option for user to separate major/minor species.

From this graph, we could have a basic understanding of the relationships among reactions.



A + B = C + D

plot\_path(element\_filter, start = None, end = None, ReactionSystem): plot the path of species containig specific element

We construct a network with species under specific element filter using graphviz.

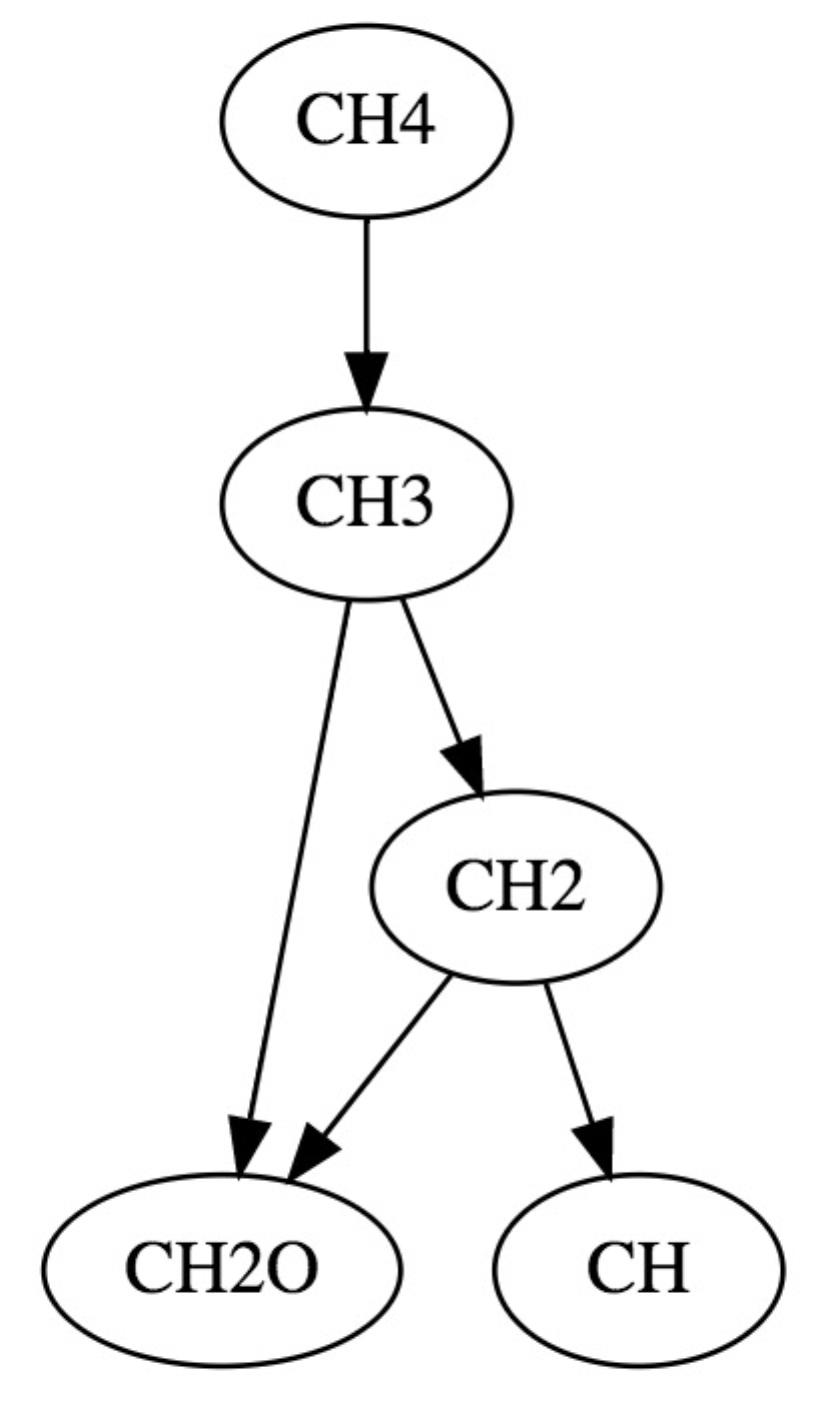
For elementary reaction:

Build directed edges (u,v) if v is u’s product in an equation.

Additional option:

Add attribute “+ related\_reactant” on edges.

Set start species / end species for the graph.



For reversible reaction:

Undirected edge / double arrows

plot\_evolution(graph, type, time\_range)

plot the initial, intermidiate and final state of a graph within time\_range to get the change of concentration.

require ODE solver to calculate concentration (and/or progress rate)

plot three different graph with same graph structure but different node size (and/or edge weight)