# Three State Formalism

Vaibhav

June 24, 2023

## 1 Boolean Formalism

We have the following rules in boolean formalism for updating a node during simulations

- $S_j = 1$ if $\sum_i adj[i][j] * S_i > 0$
- $S_j = -1$ if $\sum_i adj[i][j] * S_i < 0$
- In the case where the above sum is 0 we let the node have whatever state it had before and be somewhat 'unregulated'

```
for i in range(0,n):
    adj_sum+= (nodes_state[i]*adj[i][pos])
if adj_sum>0:
    buffer[pos]=1


elif adj_sum<0:
    buffer[pos]=-1

elif adj_sum==0:
    buffer[pos]=buffer[pos]
return(buffer)
```

Figure 1: Boolean Formalism

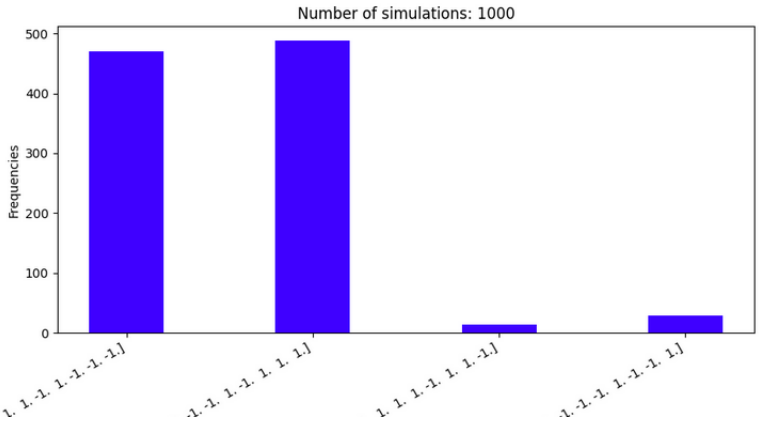


Figure 2: EMT Racipe Boolean SSF

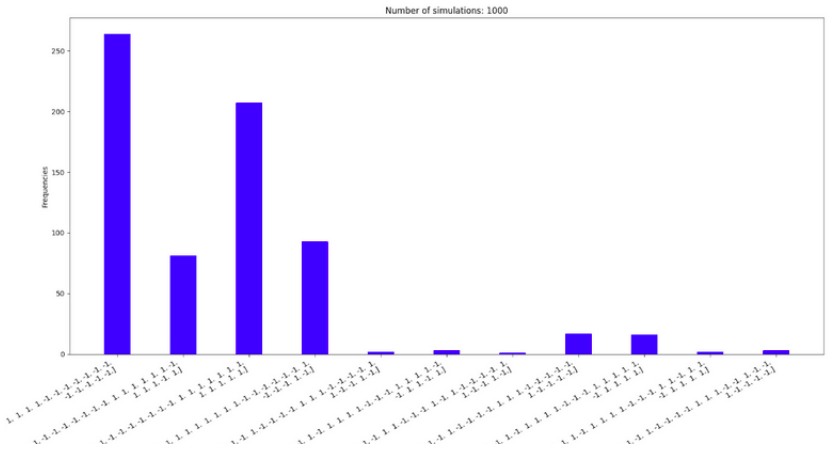


Figure 3: EMT Racipe 2 Boolean SSF

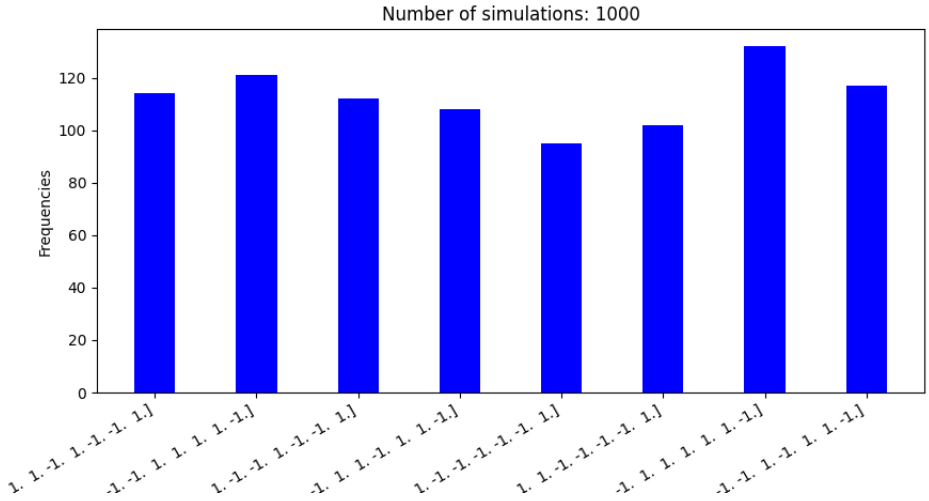


Figure 4: Melanoma Network Boolean SSF

## 2 Three State Formalism

Here we do not allow the unregulated nodes to affect other nodes. Therefore,

- $S_j = 1$ if $\sum_i adj[i][j] * S_i > 0$
- $S_j = -1$ if $\sum_i adj[i][j] * S_i < 0$
- In the case where the above sum is 0 we introduce a third inactive state 0, incapable of any forward regulation.

Biologically it might mean that genes with basal production rates can't affect other genes
Upon simulating all the networks with three state formalism the following frequency distribution of states was obtained

```
    for i in range(0,n):
        adj_sum+= (nodes_state[i]*adj[i][pos])
    if adj_sum>0:
        buffer[pos]=1


    elif adj_sum<0:
        buffer[pos]=-1

    elif adj_sum==0:
        buffer[pos]=0
    return(buffer)
```

Figure 5: Terenary Formalism

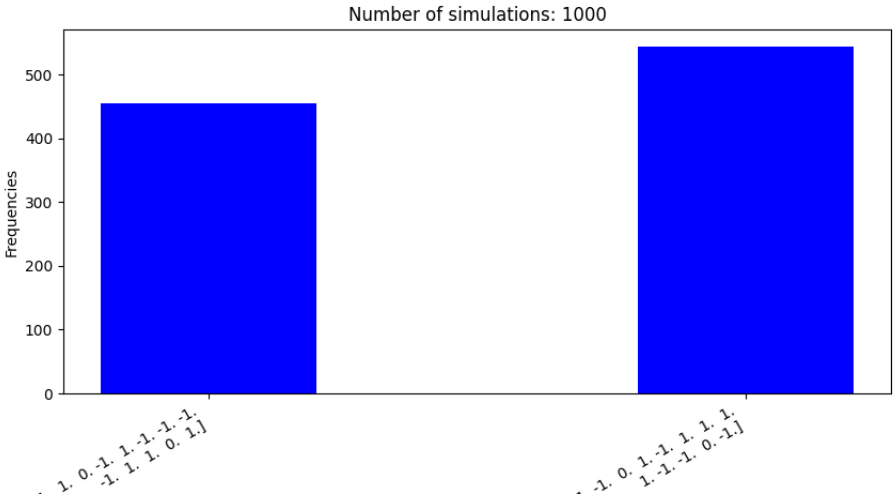


Figure 6: EMT Racipe 2 network simulated via terenary formalism

In the above figure, the two steady states have two nodes off which is different from the most frequent steady states obtained from boolean formalism

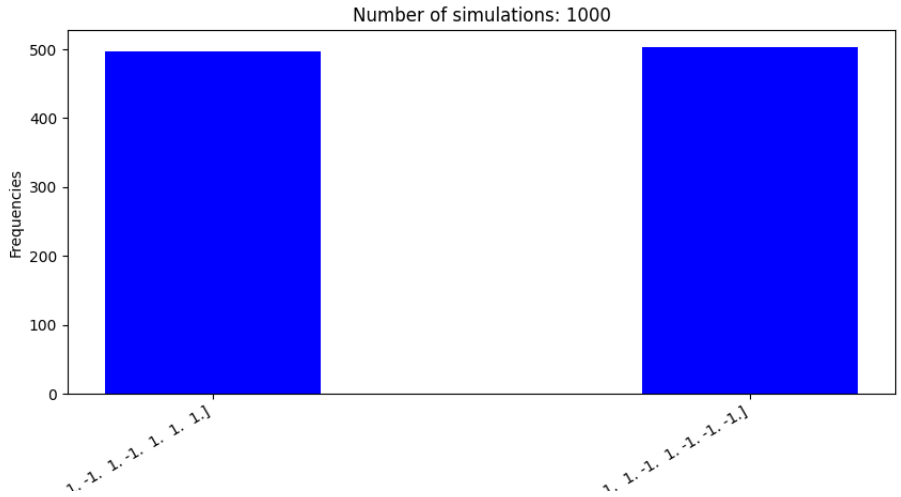


Figure 7: emt racipe terenary formalism

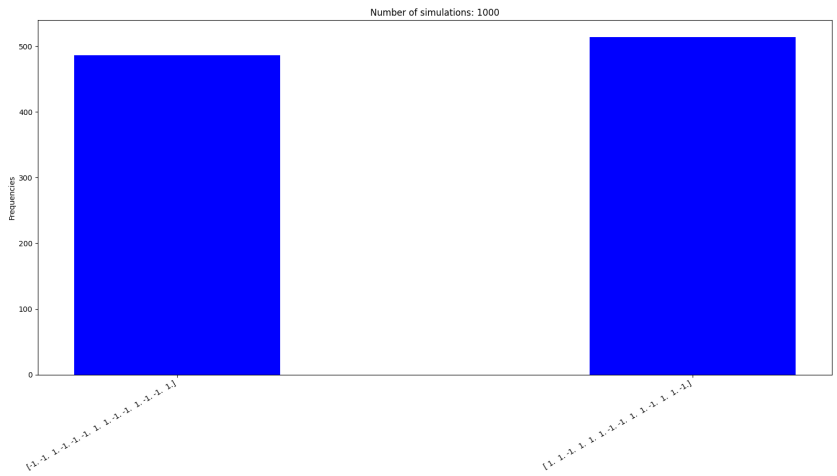


Figure 8: Melanoma Network terenary

Upon inspecting all the hybrid states of emt racipe and emt racipe 2 network, it was found out that they all have at least one 'loose/unregulated ' node

Terenary formalism takes out the effect of the loose node and it is seen that most hybrid states obtained via boolean formalism, when simulated via terenary formalism fall into either of the two states obtained hence indicating that hybrid states are there because of the 'loose' nodes.