Can build trees of with 3,4, or 5 clones. It takes in as the first argument the number of clones (3,4,5) and the second argument is a list containing the sizes of each of the respective shared clones (note that when I write c1c2c3 this means the number of mutations shared by c1c2c3 only – i.e. c4 doesn’t have them):

**3:** (c1,c2,c3, c1c2,c1c3,c2c3, c1c2c3)

**4:** (c1,c2,c3,c4 c1c2,c1c3,c1c4,c2c3,c2c4,c3c4, c1c2c3,c1c2c4,c1c3c4,c2c3c4, c1c2c3c4)

**5:** (c1,c2,c3,c4,c5 c1c2,c1c3,c1c4,c1c5,c2c3,c2c4,c2c5,c3c4,c3c5,c4c5,

c1c2c3,c1c2c4,c1c2c5,c1c3c4,c1c3c5,c1c4c5,c2c3c4,c2c3c5,c2c4c5,c3c4c5,

c1c2c3c4,c1c2c3c5,c1c2c4c5,c1c3c4c5,c2c3c4c5, c1c2c3c4c5)

And it makes a Z matrix with a column of 0s to represent the normal and then a 1 to represent if 2 clones share a mutation. For instance in the tree below we would represent this as:

[ 3 , [4,1,2, 0,0,3, 5] ]

5

C3

C2

C1

1

2

3

4

Running the following command:

dist = Z\_mtx\_creation(3,[4,1,2,0,0,3,5])

print(np.matrix(dist))

We obtain the following output:

[[0 1 0 0]

[0 1 0 0]

[0 1 0 0]

[0 1 0 0]

[0 0 1 0]

[0 0 0 1]

[0 0 0 1]

[0 0 1 1]

[0 0 1 1]

[0 0 1 1]

[0 1 1 1]

[0 1 1 1]

[0 1 1 1]

[0 1 1 1]

[0 1 1 1]]

We can see that, highlighting the above,

[[0 1 0 0]

[0 1 0 0]

[0 1 0 0]

[0 1 0 0]

[0 0 1 0]

[0 0 0 1]

[0 0 0 1]

[0 0 1 1]

[0 0 1 1]

[0 0 1 1]

[0 1 1 1]

[0 1 1 1]

[0 1 1 1]

[0 1 1 1]

[0 1 1 1]]

We see that there are 5 common mutations to clone 1, 2 and 3. We see clone 2 and 3 have 3 common mutations that clone 1 does not have. We see clone 1 has 4 mutations private to itself, clone 2 has 1 and clone 3 has 2. This describes the tree we see above accurately.