Tutorial-3-Dealing_with_complicated_implication_networks

February 1, 2019

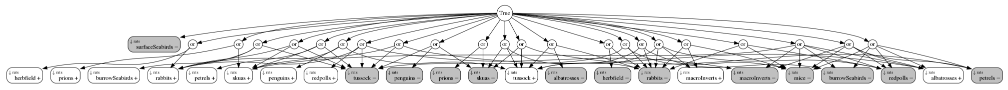
In this example, we'll use the list of PCU's below (see Tutorial 2 for how to obtain PCUs) that were obtained from simulating rat control on Macquarie Island.

```
In [1]: PCUList = [
          ['posrats_surfaceSeabirds'],
          ['negrats_rabbits', 'negrats_herbfield'],
          ['negrats_prions', 'negrats_skuas'],
          ['posrats_herbfield', 'posrats_rabbits'],
          ['posrats_prions', 'posrats_skuas'],
          ['posrats_tussock', 'posrats_mice',
           'posrats_rabbits'],
          ['negrats_rabbits', 'negrats_burrowSeabirds',
           'posrats_tussock'],
          ['negrats_penguins', 'negrats_skuas',
           'negrats_tussock'],
          ['negrats_rabbits', 'posrats_tussock',
           'negrats_skuas'],
          ['posrats_skuas', 'posrats_petrels',
           'negrats_tussock'],
          ['posrats_redpolls', 'posrats_mice',
           'posrats_rabbits'],
          ['posrats_tussock', 'negrats_petrels',
           'negrats_skuas'],
          ['negrats_tussock', 'posrats_burrowSeabirds',
           'posrats_rabbits'],
          ['negrats_macroInverts', 'posrats_mice',
           'posrats_rabbits'],
          ['posrats_albatrosses', 'negrats_tussock',
           'posrats_rabbits'],
          ['posrats_skuas', 'negrats_tussock',
           'posrats_rabbits'],
          ['posrats_tussock', 'posrats_skuas',
           'posrats_penguins'],
          ['negrats_rabbits', 'posrats_tussock',
           'negrats_albatrosses'],
          ['negrats_rabbits', 'posrats_tussock',
           'negrats_redpolls', 'posrats_macroInverts'],
 ['negrats_macroInverts', 'posrats_redpolls',
  'negrats_tussock', 'posrats_rabbits'],
 ['posrats_petrels', 'posrats_mice', 'posrats_macroInverts',
  'posrats_redpolls', 'negrats_albatrosses'],
 ['posrats_petrels', 'posrats_mice',
  'posrats_burrowSeabirds', 'posrats_redpolls',
  'negrats_albatrosses'],
 ['posrats_mice', 'posrats_burrowSeabirds',
  'posrats_macroInverts', 'posrats_redpolls',
  'negrats_albatrosses', 'negrats_skuas']
]
```

Our objective is to draw an implication network that can be interpreted for decision-makers planning a rat control programme.

In the first attempt, the implication network that results is very complicated and difficult to interpret.

```
In [3]: from findpcu import draw_implication_network, draw_implication_network2
        import os

        draw_implication_network2(PCUList,
                                  [],
                                  'attempt_1',
                                  niceNames = None,
                                  controlSymbol = 'downarrow')

        # call graphviz to create a png, display in markdown cell
        os.system("dot -Tpng attempt_1.dot > attempt_1.png")

attempt_1.pdf has been created


Out[3]: 0
```

PCUList 

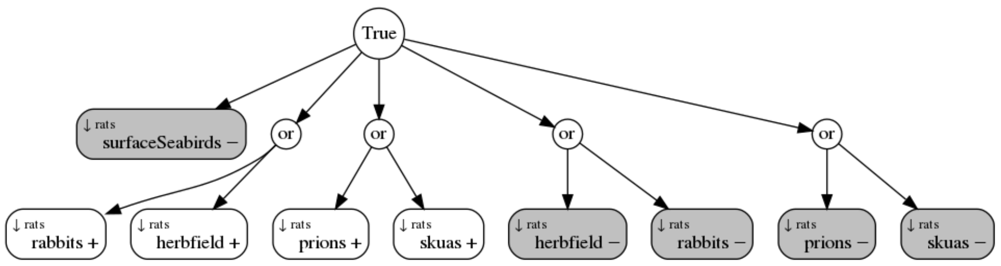
To simplify a network, a good first step is to split the PCUs up by length and present them in separate networks. Provided that every PCU is represented in the networks, then no information will be lost.

It is also a good idea to put highly-connected responses, and other pests, at the top as antecedents. We can spot highly-connected nodes by the number of arrows that are feeding into them, and placing other pest species at the top may be useful to decision-makers who are considering either flow-on effects on other pests, or a multi-species control programme.

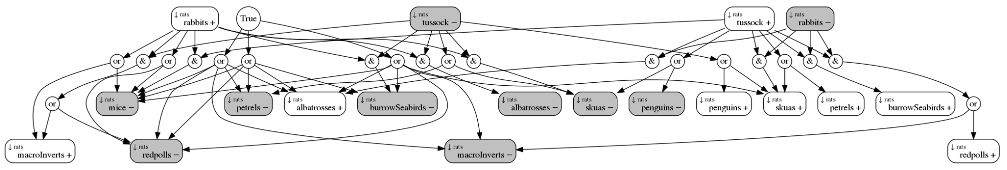
```
In [11]: PCUList1 = [ PCU for PCU in PCUList if len(PCU) <= 2 ]
         PCUList2 = [ PCU for PCU in PCUList if len(PCU) > 2 ]

         draw_implication_network2(PCUList1,
            [],
            'PCUList1_1', niceNames = None, controlSymbol = 'downarrow')
        draw_implication_network2(PCUList2,
            ['negrats_rabbits', 'posrats_rabbits', 'posrats_tussock', 'negrats_tussock'],
            'PCUList2_1', niceNames = None, controlSymbol = 'downarrow')

        os.system("dot -Tpng PCUList1_1.dot > PCUList1_1.png")
        os.system("dot -Tpng PCUList2_1.dot > PCUList2_1.png")
```

```
PCUList1_1.pdf has been created
PCUList2_1.pdf has been created
```

```
Out[11]: 0
```



PCUList1_1



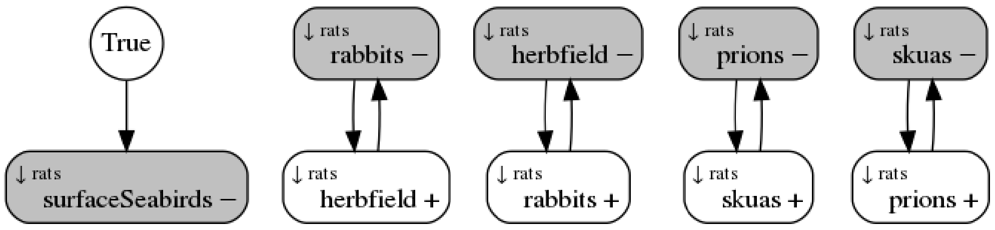
PCUList2_1

In the top network (PCUList1), we recognise the same bidirectional implication relationship that appeared previously, in the network representing rabbit control. Rabbits and herbfield are also in a bidirectional relationship. For short implications, the function draw_implication_network can be more useful. It will repeat the information contained in some of the PCUs (e.g. showing both bidirectional outcomes), but when the implications are short, this doesn't increase the complexity of the network much compared to the increase in clarity of the implications' meaning.

```
In [5]: draw_implication_network(PCUList1,
                                 'PCUList1_2',
                                 niceNames = None,
                                 controlSymbol = 'downarrow')
        os.system("dot -Tpng PCUList1_2.dot > PCUList1_2.png")
```

```
PCUList1_2.pdf has been created
```

```
Out[5]: 0
```



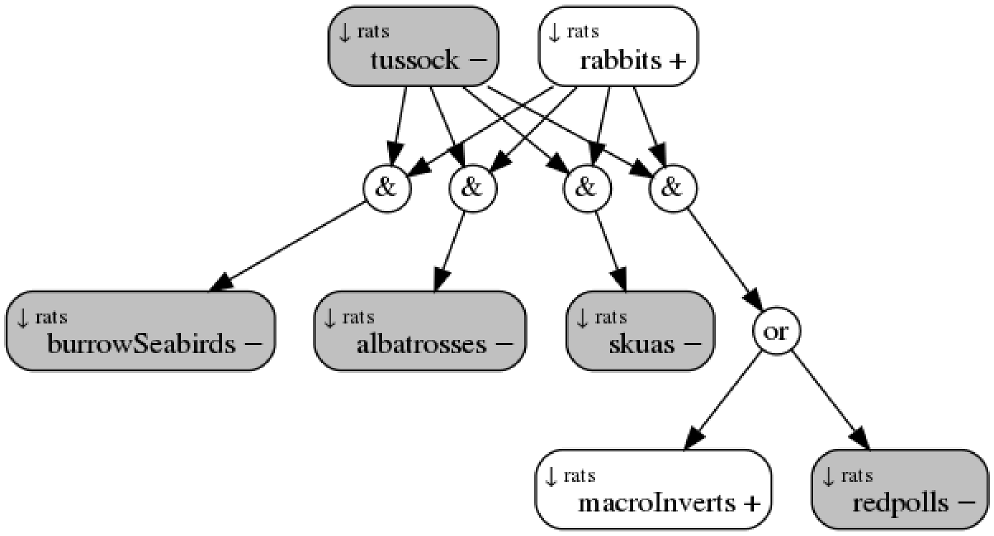
PCUList1_2

In the second network (PCUList2), we see that a symmetry in the effect of the combined response in rabbits and tussock. We can split those out into two separate networks and place the remainder of the PCUs in PCUList5 for further analysis.

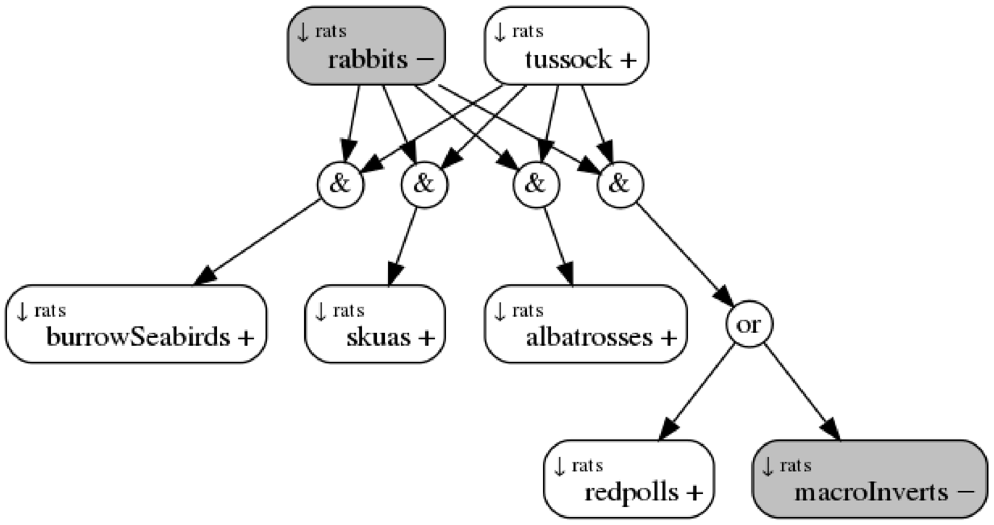
```
In [6]: PCUList3 = list() # rat and tussock relationships
        PCUList4 = list()
        PCUList5 = list() # other
        for PCU in PCUList2:

            if 'posrats_rabbits' in PCU and 'negrats_tussock' in PCU:
                PCUList3.append(PCU)
            elif 'negrats_rabbits' in PCU and 'posrats_tussock' in PCU:
                PCUList4.append(PCU)
            else:
                PCUList5.append(PCU)

        draw_implication_network2(PCUList3,
                    ['posrats_rabbits', 'negrats_tussock'],
                    'PCUList3_1', niceNames = None, controlSymbol = 'downarrow')
        draw_implication_network2(PCUList4,
                    ['negrats_rabbits', 'posrats_tussock'],
                    'PCUList4_1', niceNames = None, controlSymbol = 'downarrow')

        os.system("dot -Tpng PCUList3_1.dot > PCUList3_1.png")
        os.system("dot -Tpng PCUList4_1.dot > PCUList4_1.png")
```

```
PCUList3_1.pdf has been created
PCUList4_1.pdf has been created
```

```
Out[6]: 0
```



PCUList3_1



PCUList4_1

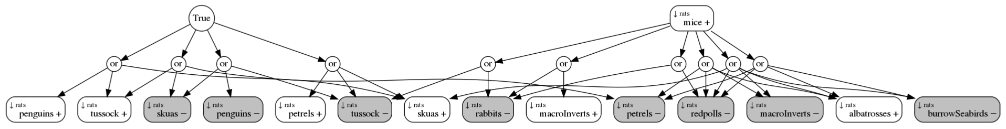
The symmetry between the two scenarios above may be useful for understanding the whole-system behaviour. For example, the response of burrow-nesting seabirds and albatrosses echoes what was found for rabbit control.

We now take a closer look at the PCUs that remain.

```
In [7]: draw_implication_network2(PCUList5,
                        ['posrats_mice'],
                        'PCUList5_1', niceNames = None, controlSymbol = 'downarrow')
        os.system("dot -Tpng PCUList5_1.dot > PCUList5_1.png")
```

```
PCUList5_1.pdf has been created
```

```
Out[7]: 0
```



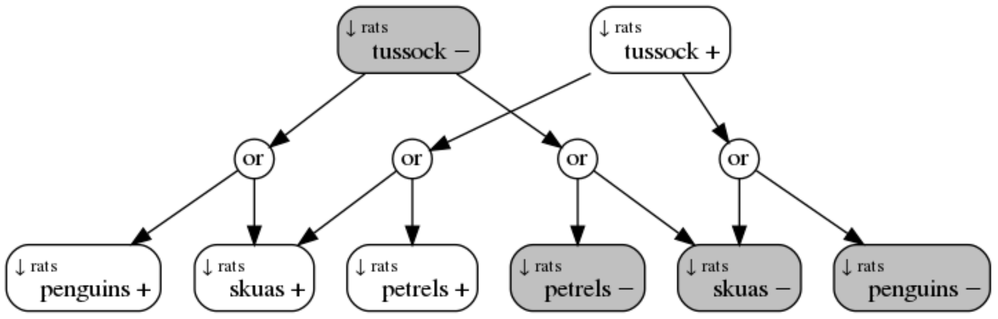
PCUList5_1

There is an obvious split between relationships involving mice and not. The relationships not involving mice have a symmetry, so we should choose an arrangement that emphasises that. Below, we choose to place the effect of rat control on tussock as the antecedent.

```
In [8]: PCUList6 = list() # for rules without mice
        PCUList7 = list() # for rules with mice

        for PCU in PCUList5:
            if 'posrats_mice' in PCU:
                PCUList7.append(PCU)
            else:
                PCUList6.append(PCU)

        draw_implication_network2(PCUList6,
                        ['posrats_tussock', 'negrats_tussock'],
                        'PCUList6_1', niceNames = None, controlSymbol = 'downarrow')
        os.system("dot -Tpng PCUList6_1.dot > PCUList6_1.png")
```

```
PCUList6_1.pdf has been created
```

```
Out[8]: 0
```



PCUList6_1

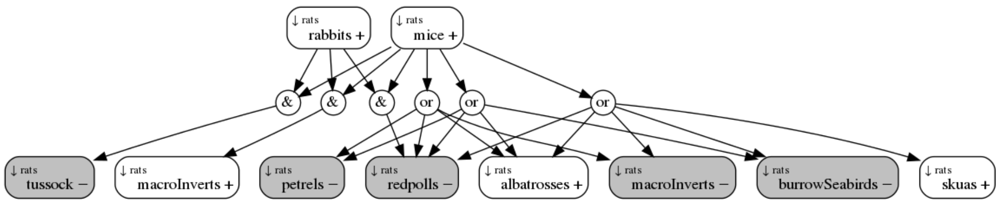
Now we consider the right-hand side, the relationships involving mice.

The importance of tussock suggests that the interaction between rabbits, mice, and tussock may be interesting in its own right. The simultaneous effect on rabbits is also involved in outcomes for macroinvertebrates and redpolls, and rabbits are another pest species, so we move the rabbit response to the antecedent.

```
In [9]: draw_implication_network2(PCUList7,
                        ['posrats_rabbits', 'posrats_mice'],
                        'PCUList7_1', niceNames = None, controlSymbol = 'downarrow')
        os.system("dot -Tpng PCUList7_1.dot > PCUList7_1.png")
```

```
PCUList7_1.pdf has been created
```

```
Out[9]: 0
```



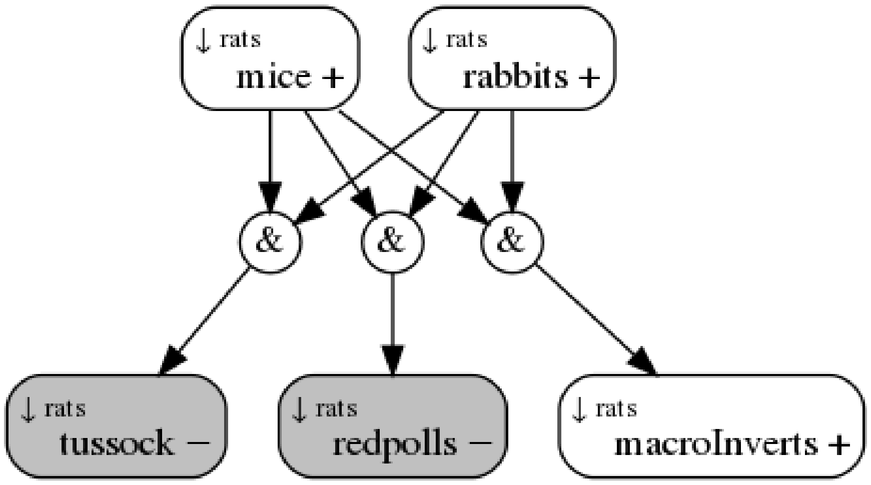
PCUList7_1

One could end the splitting of the remaining PCUs here. Alternatively, one could choose to emphasise the contingency upon the response of redpolls, by further splitting the network.

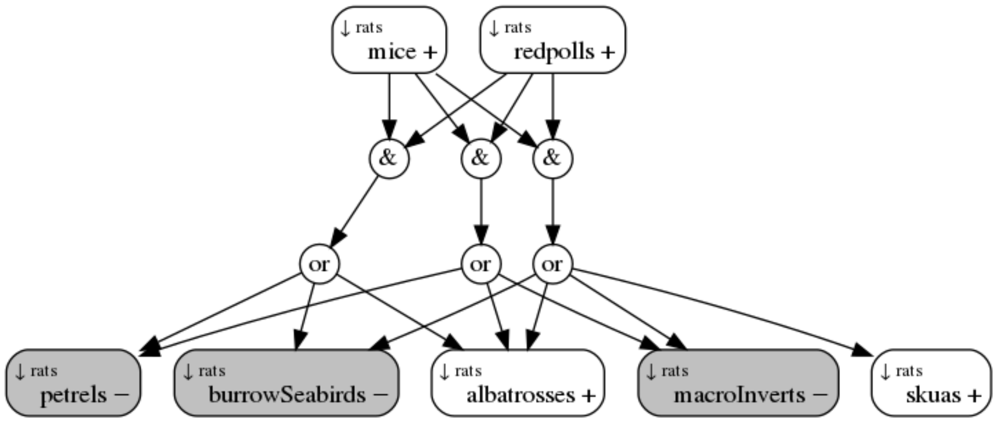
```
In [10]: PCUList8 = list() # for rabbit and mice relationships
         PCUList9 = list() # the remainder, which all involve redpolls

         for PCU in PCUList7:
             if 'posrats_rabbits' in PCU:
                 PCUList8.append(PCU)
             else:
                 PCUList9.append(PCU)

         draw_implication_network2(PCUList8,
                      ['posrats_rabbits', 'posrats_mice'],
                      'PCUList8_1', niceNames = None, controlSymbol = 'downarrow')
         os.system("dot -Tpng PCUList8_1.dot > PCUList8_1.png")

         draw_implication_network2(PCUList9,
                      ['posrats_redpolls', 'posrats_mice'],
                      'PCUList9_1', niceNames = None, controlSymbol = 'downarrow')
         os.system("dot -Tpng PCUList9_1.dot > PCUList9_1.png")
```

```
PCUList8_1.pdf has been created
PCUList9_1.pdf has been created
```

```
Out[10]: 0
```



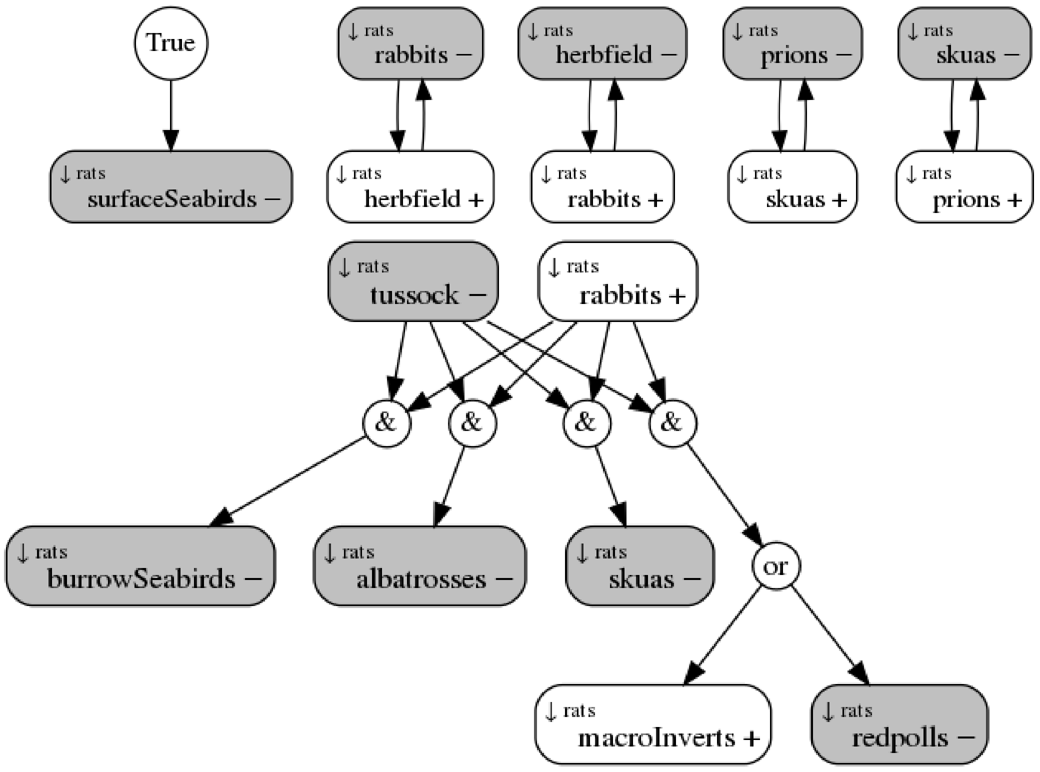
PCUList8_1

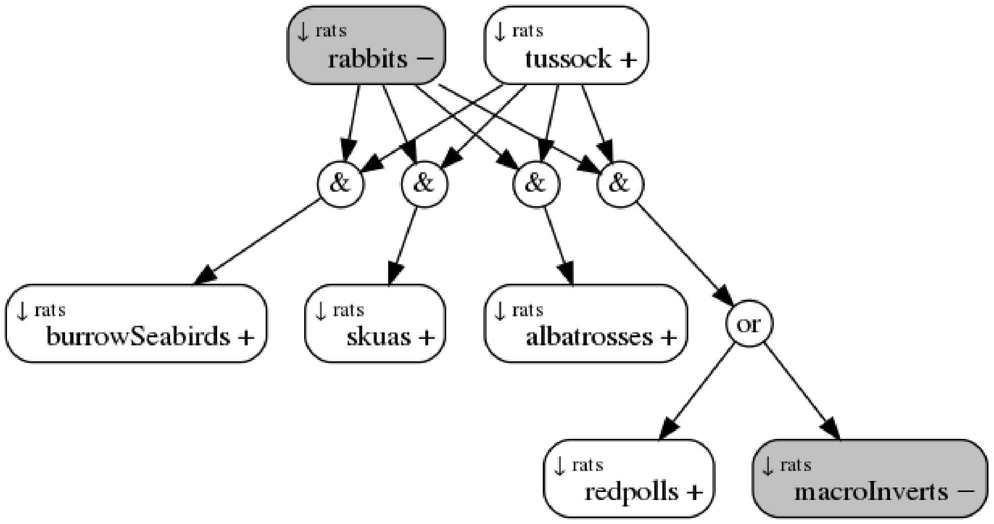


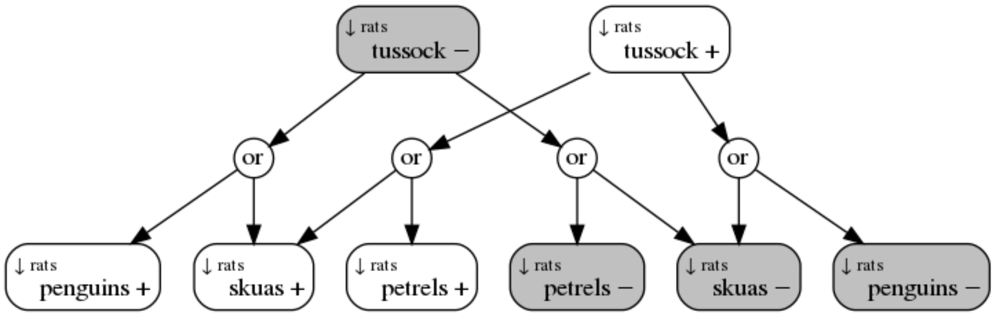
PCUList9_1

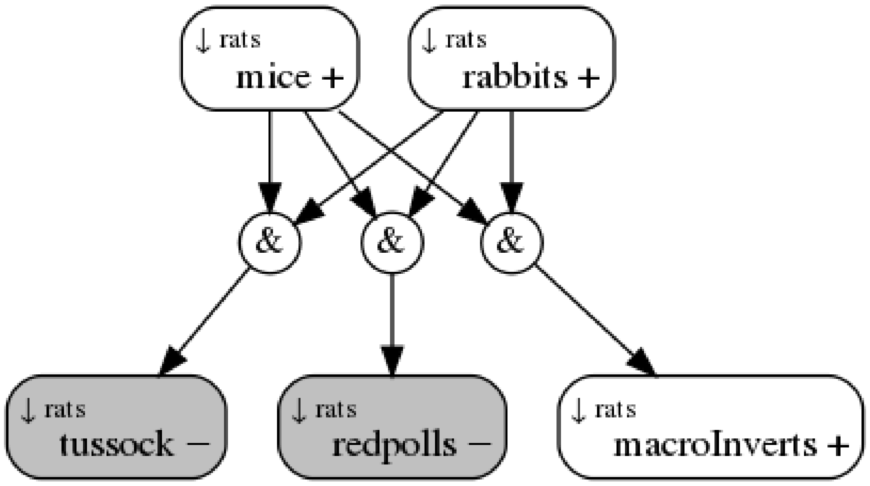
0.1 Potential final presentation

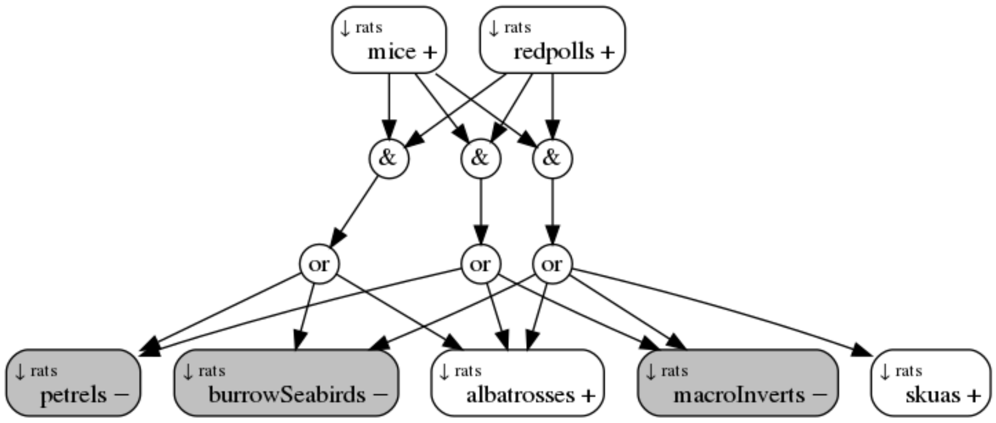
The presentation below combines all of the subnetworks that we split above. The decisions that were made to produce this network were somewhat arbitrary, and the best choice depends upon the needs of conservation decision-makers, and may also reflect our causal understanding of the relationships between species.




↓ rats rabbits −
↓ rats tussock +
&
&
&
&
↓ rats burrowSeabirds +
↓ rats skuas +
↓ rats albatrosses +
or
↓ rats redpolls +
↓ rats macroInverts −


↓ rats tussock −
↓ rats tussock +
or
or
or
or
↓ rats penguins +
↓ rats skuas +
↓ rats petrels +
↓ rats petrels −
↓ rats skuas −
↓ rats penguins −


↓ rats mice +
↓ rats rabbits +
&
&
&
↓ rats tussock −
↓ rats redpolls −
↓ rats macroInverts +


↓ rats mice +
↓ rats redpolls +
&
&
&
or
or
or
↓ rats petrels −
↓ rats burrowSeabirds −
↓ rats albatrosses +
↓ rats macroInverts −
↓ rats skuas +