FCM Lib Import

Thucydides Trap Dev

Preamble

```
logisticActvn[t_] := Max[0, LogisticSigmoid[7.5 t - 3.5]];
SetAttributes[logisticActvn, {Listable, NumericFunction}]

linearActvn[t_] := Min[Max[0, t], 1];
SetAttributes[linearActvn, {Listable, NumericFunction}]


(* LU Activation: *)
{$activationFxn, $activationBias} = {linearActvn, 0};

trapvs = Import["./trap-verts.csv", "CSV"];
trapvs[[ ;; , 2 ;;]] = StringTrim /@ trapvs[[ ;; , 2 ;;]];
n = Length@trapvs
```

```
17
```

```
addjitter = 0.075;
strongOrLink = $activationThreshold / 2 ;
orLink = strongOrLink(*-0.05*);
weakOrLink = $activationThreshold / 4;
unsure = orLink;
andLink = $activationThreshold / 3;
subAndLink = $activationThreshold + 2 * addjitter;
```

Dynamic Specification

Static Specification

Illustrations

Continuous Activation Exploration

```
fcms = {dtrapFCM, strapFCM};
```

Engineered Masks + Vertex Leverage Analysis

```
evcnt = Round[EigenvectorCentrality@dtrapFCM, 0.001];
bwcnt = Round[(# / Total[#]) &@BetweennessCentrality@dtrapFCM, 0.001];
clcnt = Round[(# / Total[#]) &@ClosenessCentrality@dtrapFCM, 0.001];

Row@{
  TableForm[
   Transpose[Transpose[trapvs[[ ;; , {1, 2, 3}]]]~Join~{evcnt, bwcnt, clcnt}],
   TableHeadings → {None,
     Style[#, Bold] & /@{"", "Label", "Full Description", "Eig.", "Bwc.", "Close"}}
  ],
  Graph[dtrapFCM,
   GraphLayout → "SpringEmbedding",
   ImageSize → 72 × 6
  ]
 }

(* 4:geod, 8:dipl,
14:econdep - highest/equal closeness centrality,
15:allyTangle next up,
10:ShrdCult,
*)

evcnt = Round[EigenvectorCentrality@strapFCM, 0.001];
bwcnt = Round[(# / Total[#]) &@BetweennessCentrality@strapFCM, 0.001];
clcnt = Round[(# / Total[#]) &@ClosenessCentrality@strapFCM, 0.001];

Row@{
  TableForm[
   Transpose[Transpose[trapvs[[ ;; , {1, 2, 3}]]]~Join~{evcnt, bwcnt, clcnt}],
   TableHeadings → {None,
     Style[#, Bold] & /@{"", "Label", "Full Description", "Eig.", "Bwc.", "Close"}}
  ],
  Graph[strapFCM,
   GraphLayout → "RadialEmbedding",
   ImageSize → 72 × 6
  ]
 }
```

Mask Engineering

Comparisons

```
Manipulate[
 fins = ((FCMEvolSeq[#, inp, mask] &) /@ fcms);
 res = Transpose@{fcms, fins};
 Panel[$activationFxn];
 Row[{
   TableForm[
    Transpose[{inp, mask}~Join~Chop[SetAccuracy[fins[[;; , -1, ;;]], 3], 10^-2]],
    TableHeadings → {
      MapThread[Style[(#1 <> #2 <> #3), 14] &,
       {trapvs[[;; , 3]], ConstantArray["|", n], trapvs[[;; , 2]]}],
      {"inp", "mask", "Dyn.", "Static"}
     },
    TableAlignments → Right, TableSpacing → {2, 1.5}
   ],

   TabView[
    Table[
     (FCMView[Graph[#1, GraphLayout -> "RadialEmbedding",
           ImageSize → 72 × 6], #2, trapvs] &) @@ res[[ev]],
     {ev, Range@Length@res}
    ], 1
   ],

   GraphicsRow[
    MatrixPlot[#,
       ColorRules → {x_ /; 0 ≤ x ≤ 0.45 → LightBlue,
         x_ /; 0.45 < x ≤ 0.65 → Orange, x_ /; x ≤ 0.65 → White},
       ImageSize → Large,
       ImageMargins → 0,
       FrameTicks → {None, Automatic} ,
       FrameTicksStyle → Directive[20, Bold]
      ] & /@ (Transpose /@ fins),
    ImageMargins → 0
   ]

  }, "|",
  ImageSize → 72 × 32
],
```

```
 {{inp, egmask0(*ConstantArray[0,n]*)(*RandomInteger[{0,1},n]*)}, ControlType → None},
 {{mask, (*ConstantArray[0,n]*)egmask0}, ControlType → None},


 Dynamic@Panel@Grid[{
     {SetterBar[Dynamic[{$activationFxn, $activationBias}], {{linearActvn, 0},
        {logisticActvn, 0}, {UnitStep, 0.5}}], Text[{$activationFxn, $activationBias}]},
     Outer[Text[Style[trapvs[[#, 2]], 14]] &, Range[n]],
     Outer[Checkbox[Dynamic[inp[[#]]], {0, 1}] &, Range[n]],
     Outer[Checkbox[Dynamic[mask[[#]]], {0, 1, -1}] &, Range[n]]
    }, Alignment → Right
   ]
]
(* culture node mediates change from war to no war *)
rule = x_ ?NumberQ /; 0.5 < Abs[x] < 1.2 :> {Style[x, Bold, Background → LightRed]};

TableForm[
 Transpose[Chop[SetAccuracy[fins[[1]], 2], 10^-1]] /. rule,
 TableHeadings → {
   MapThread[Style[(#1 <> #2 <> #3), 14] &,
    {trapvs[[ ;; , 3]], ConstantArray["|", n], trapvs[[ ;; , 2]]}],
   ("t=" <> ToString@#) & /@ Range[(Dimensions@fins)[[2]]]
  },
 TableAlignments → Right, TableSpacing → Automatic(*{2,2.5}*)
](*/.rule*)
```

Exhaustive Search

Linear Activation Fxn

```
{$activationFxn, $activationBias} = {linearActvn, 0}

searchResults = Table[
   {
    (*IntegerDigits[inp, 2,n],*)
    Last@
     Last@FCMEvolSeq[dtrapFCM, IntegerDigits[inp, 2, n], IntegerDigits[inp, 2, n]],
    Last@Last@FCMEvolSeq[strapFCM, IntegerDigits[inp, 2, n], IntegerDigits[inp, 2, n]]
   },
   {inp, 0, 2^n - 1 (*300*)}
  ];
Export["dyn-stat-cmp-linear.csv", searchResults];

dWarStates = (Last /@ searchResults);
sWarStates = (First /@ searchResults);
Histogram /@ {dWarStates, sWarStates}
Quartiles /@ {dWarStates, sWarStates}
```

Logistic Activation Fxn

Heaviside Activation Fxn

```
{$activationFxn, $activationBias} = {UnitStep, 0.5}
searchResults = Table[
   {
    (*IntegerDigits[inp, 2,n],*)
    Last@
     Last@FCMEvolSeq[dtrapFCM, IntegerDigits[inp, 2, n], IntegerDigits[inp, 2, n]],
    Last@Last@FCMEvolSeq[strapFCM, IntegerDigits[inp, 2, n], IntegerDigits[inp, 2, n]]
   },
   {inp, 0, 2^n - 1 (*300*)}
  ];
Export["dyn-stat-cmp-step.csv", searchResults];

dWarStates = (Last /@ searchResults);
sWarStates = (First /@ searchResults);
Histogram /@ {dWarStates, sWarStates}
Quartiles /@ {dWarStates, sWarStates}

labels = trapvs[[ ;; , 2]];
labels[[Select[Range@n, (Abs[divmeas[[#]] - Max[divmeas]] < 0.05) &]]]

2^(n-1)

65 536
```