```
BeginPackage["MultilayerNetworks`"];

networkdatasingle::usage = "description.";
networkdatabinned::usage = "description.";
networkgraph::usage = "description.";
networkdegrees::usage = "description.";
statisticaloverlaps::usage = "description.";
overlaps::usage = "description.";
randomness::usage = "description.";
correlationcoefficientintraintertwofeature::usage = "description.";
correlationcoefficientfivefeature::usage = "description.";
correlationcoefficienttwofeature::usage = "description.";
correlationcoefficientelevenfeature::usage = "description.";
```

```
Begin["`Private`"];
```

```
Clear[networkdatasingle]
networkdatasingle[feature_]:=Module[{rawaim,aim,campaign,seri},
rawaim=Symbol["data"][[All,feature]];
aim=DeleteCases[rawaim,"NA"];
campaign=Delete[Symbol["data"][[All,118]],Position[rawaim,"NA"]];
seri=Delete[Symbol["data"][[All,1]],Position[rawaim,"NA"]];
{aim,campaign,seri}]
```

```
Clear[networkdatabinned]
networkdatabinned[feature_,step_]:=Module[{rawaim,aim,campaign,seri,min,max,binningamoun
rawaim=Symbol["data"][[All,feature]];
aim=DeleteCases[rawaim,"NA"];
campaign=Delete[Symbol["data"][[All,118]],Position[rawaim,"NA"]];
seri=Delete[Symbol["data"][[All,1]],Position[rawaim,"NA"]];
min=Floor[Min[Sort[DeleteDuplicates[aim]]],0.1];
max=Ceiling[Max[Sort[DeleteDuplicates[aim]]]]+step;
binningamount=Length[DeleteCases[BinLists[aim,{min,max,step}],{}]];
binning=Table[Catch[Do[If[IntervalMemberQ[Interval[i],aim[[j]]]==True,Throw[i]],
{i,Partition[Range[min,max,step],2,1]}]],{j,Length[aim]}];
aim=DeleteCases[binning,Null];
{aim,campaign,seri}]
```

```
Clear[networkgraph]
networkgraph[aim_,campaign_,vertexsize_,vertexlabelsize_,imagesize_,vertexcolor_]:=Modul
singlesupportvalues,pairs,pairsupportvalues,liftvalues,allmatrixelements,likelypairs,
binarymatrix,graph},
binningmembers=Sort[DeleteDuplicates[aim]];
aimbaskets=Table[DeleteDuplicates[i],{i,Values@GroupBy[Thread[{aim,campaign}],Last->Firs
singlesupportvalues=Table[N[Count[Table[MemberQ[i,j],{i,aimbaskets}],
True]/Length[aimbaskets]],{j,binningmembers}];
pairs=Subsets[binningmembers,{2}];
pairsupportvalues=Table[N[Count[Table[SubsetQ[i,j],{i,aimbaskets}],True]/
Length[aimbaskets]],{j,pairs}];
liftvalues=pairsupportvalues/(DeleteCases[Flatten@UpperTriangularize[Table[singlesupport
[[j]]*singlesupportvalues[[k]],{j,Length[binningmembers]},{k,Length[binningmembers]}],1
allmatrixelements=Sort[Join[pairs,Reverse[pairs,2],Table[{i,i},{i,binningmembers}]]];
likelypairs=Extract[pairs,Position[liftvalues,x_/;x>1]];
binarymatrix=ArrayReshape[Table[If[j==True,1,0],{j,Table[MemberQ[likelypairs,i],
{i,allmatrixelements}]}],{Length@binningmembers,Length@binningmembers}];
graph=AdjacencyGraph[binarymatrix,{GraphLayout->Automatic,DirectedEdges->False,
EdgeShapeFunction->"Line",VertexSize->vertexsize,VertexStyle->vertexcolor,
VertexLabelStyle->Directive[Black,Italic,vertexlabelsize],VertexLabels->Flatten[MapThrea
{♯1->Placed[♯2,Center]}&,{Range[1,Dimensions[binarymatrix][[1]]],Table[StringRiffle[i,"
{i,binningmembers}]}]]},ImageSize->imagesize];{graph,Length@binningmembers}]
```

```
Clear[networkdegrees]
networkdegrees[aim_,campaign_]:=Module[{binningmembers,aimbaskets,
singlesupportvalues,pairs,pairsupportvalues,liftvalues,allmatrixelements,likelypairs,
binarymatrix,graph,degreesfornodes},
binningmembers=Sort[DeleteDuplicates[aim]];
aimbaskets=Table[DeleteDuplicates[i],{i,Values@GroupBy[Thread[{aim,campaign}],Last->Firs
singlesupportvalues=Table[N[Count[Table[MemberQ[i,j],{i,aimbaskets}],
True]/Length[aimbaskets]],{j,binningmembers}];
pairs=Subsets[binningmembers,{2}];
pairsupportvalues=Table[N[Count[Table[SubsetQ[i,j],{i,aimbaskets}],True]/
Length[aimbaskets]],{j,pairs}];
liftvalues=pairsupportvalues/(DeleteCases[Flatten@UpperTriangularize[Table[singlesupport
[[j]]*singlesupportvalues[[k]],{j,Length[binningmembers]},{k,Length[binningmembers]}],1
allmatrixelements=Sort[Join[pairs,Reverse[pairs,2],Table[{i,i},{i,binningmembers}]]];
likelypairs=Extract[pairs,Position[liftvalues,x_/;x>1]];
binarymatrix=ArrayReshape[Table[If[j==True,1,0],{j,Table[MemberQ[likelypairs,i],
{i,allmatrixelements}]}],{Length@binningmembers,Length@binningmembers}];
graph=AdjacencyGraph[binarymatrix];
degreesfornodes=MapThread[♯1->♯2&,{Range[1,Length[binningmembers]],
VertexDegree@graph}]](* gives node_number → degree *)
```

```
Clear[statisticaloverlaps]
(* input feature should be {aim,seri} *)
statisticaloverlaps[{feature__},threshold_]:=Module[{nodegroups,jaccard,overlapmatrix,in
distributefunction,expandedlinkslist},
nodegroups=Table[Normal@KeySort@KeyDrop[GroupBy[MapThread[#1->#2&,i],First->Last],"NA"]
{i,{feature}}];
jaccard[x_,y_]:=Module[{x1,y1,jaccardindex,result},
    x1=Values@x;y1=Values@y;
    jaccardindex[a_,b_]:=N@(Length@Intersection[a,b]/Length@Union[a,b]);
    result=Table[jaccardindex[x1[[i]],y1[[j]]],{i,Length@x},{j,Length@y}]];
overlapmatrix=Table[jaccard[k,l],{k,nodegroups},{l,nodegroups}];
interlayerlinks=MapThread[#1->#2&,{Subsets[Range@Length@nodegroups,{2}],Position[
overlapmatrix[[#1]][[#2]],x_/;x>=threshold]&@@@Subsets[Range@Length@nodegroups,{2}]}];
distributefunction[a_,b_]:=a->#&/@b;
expandedlinkslist=Flatten[KeyValueMap[distributefunction,Association@interlayerlinks],1]
```

```
Clear[overlaps]
(* input feature should be {aim,seri} *)
overlaps[{feature__}]:=Module[{nodegroups,jaccard,overlapmatrix},
nodegroups=Table[Normal@KeySort@KeyDrop[GroupBy[MapThread[#1->#2&,i],First->Last],"NA"]
{i,{feature}}];
jaccard[x_,y_]:=Module[{x1,y1,jaccardindex,result},
    x1=Values@x;y1=Values@y;
    jaccardindex[a_,b_]:=N@(Length@Intersection[a,b]/Length@Union[a,b]);
    result=Table[jaccardindex[x1[[i]],y1[[j]]],{i,Length@x},{j,Length@y}]];
overlapmatrix=Table[jaccard[k,l],{k,nodegroups},{l,nodegroups}]]
```

```
Clear[randomness]
(* input feature should be {aim} *)
randomness[{feature__},seed_]:=Module[{serirandom,randomnodegroups,jaccard,overlapmatrix
SeedRandom[seed];
serirandom=Table[RandomSample[Symbol["data"][[All,1]],Length@i],{i,{feature}}];
randomnodegroups=Table[Normal@KeySort@KeyDrop[GroupBy[MapThread[#1->#2&,i],First->Last]
{i,MapThread[{#1,#2}&,{{feature},serirandom}]}];
jaccard[x_,y_]:=Module[{x1,y1,jaccardindex,result},
    x1=Values@x;y1=Values@y;
    jaccardindex[a_,b_]:=N@(Length@Intersection[a,b]/Length@Union[a,b]);
    result=Table[jaccardindex[x1[[i]],y1[[j]]],{i,Length@x},{j,Length@y}]];
overlapmatrixrandom=Table[jaccard[k,l],{k,randomnodegroups},{l,randomnodegroups}]]
```

```
Clear[correlationcoefficientintraintertwofeature]
(* input should be expandedlinkslist, layer input should be 1 or 2 *)
correlationcoefficientintraintertwofeature[x_,layer_]:=Module[{intradegrees,interdegrees
threadedintrainter,spe2,pea2,correlation},
intradegrees=Which[layer==1,DeleteDuplicates@(Values@x)[[All,1]]/.Symbol["degreesthickn
layer==2,DeleteDuplicates@(Values@x)[[All,2]]/.Symbol["degreeswidth"]];
interdegrees=Which[layer==1,Values@Counts@(Values@x)[[All,1]],layer==2,
Values@Counts@(Values@x)[[All,2]]];
threadedintrainter=MapThread[{#1,#2}&,{intradegrees,interdegrees}];
spe2=N@SpearmanRho[threadedintrainter[[All,1]],threadedintrainter[[All,2]]];
pea2=N@Correlation[threadedintrainter[[All,1]],threadedintrainter[[All,2]]];
{spe2,pea2}]
```

```
Clear[correlationcoefficientfivefeature]
(* input should be expandedlinkslist *)
correlationcoefficient[x_,cor_]:=Module[{keys,values,threadedcorrelationlists,spe1,pea1,
keys:=Keys@x[[i]];
values:=Values@x[[i]];
threadedcorrelationlists=Table[{Which[keys[[1]]==1,values[[1]]/.Symbol["degreessteelgra
keys[[1]]==2,values[[1]]/.Symbol["degreesthickness"],keys[[1]]==3,values[[1]]/.Symbol[
keys[[1]]==4,values[[1]]/.Symbol["degreescoatwttop"],keys[[1]]==5,values[[1]]/.
Symbol["degreestempendrtf"]],Which[keys[[2]]==1,values[[2]]/.Symbol["degreessteelgrade"
keys[[2]]==2,values[[2]]/.Symbol["degreesthickness"],keys[[2]]==3,values[[2]]/.Symbol[
keys[[2]]==4,values[[2]]/.Symbol["degreescoatwttop"],keys[[2]]==5,values[[2]]/.
Symbol["degreestempendrtf"]]},{i,Length@x}];
spe1=N@SpearmanRho[threadedcorrelationlists[[All,1]],threadedcorrelationlists[[All,2]]]
pea1=N@Correlation[threadedcorrelationlists[[All,1]],threadedcorrelationlists[[All,2]]]
correlation=Which[cor==1,spe1,cor==2,pea1]]
```

```
Clear[correlationcoefficienttwofeature]
(* input should be expandedlinkslist *)
correlationcoefficienttwofeature[x_,cor_]:=Module[{keys,values,threadedcorrelationlists,
keys:=Keys@x[[i]];
values:=Values@x[[i]];
threadedcorrelationlists=Table[{Which[keys[[1]]==1,
values[[1]]/.Symbol["degreesthickness"],keys[[1]]==2,values[[1]]/.Symbol["degreeswidth"
keys[[2]]==1,values[[2]]/.Symbol["degreesthickness"],keys[[2]]==2,values[[2]]
/.Symbol["degreeswidth"]]},{i,Length@x}];
spe2=N@SpearmanRho[threadedcorrelationlists[[All,1]],threadedcorrelationlists[[All,2]]]
pea2=N@Correlation[threadedcorrelationlists[[All,1]],threadedcorrelationlists[[All,2]]]
correlation=Which[cor==1,spe2,cor==2,pea2]]
```

```
Clear[correlationcoefficientelevenfeature]
(* input should be expandedlinkslist *)
correlationcoefficientelevenfeature[x_]:=Module[{keys,values,threadedcorrelationlists,co
keys:=Keys@x[[i]];
values:=Values@x[[i]];
threadedcorrelationlists=Table[{Which[keys[[1]]==1,values[[1]]/.Symbol["degreessteelgra
values[[1]]/.Symbol["degreesthickness"],keys[[1]]==3,values[[1]]/.Symbol["degreeswidth"
values[[1]]/.Symbol["degreescoatwttop"],keys[[1]]==5,values[[1]]/.Symbol["degreescoatwt
keys[[1]]==6,values[[1]]/.Symbol["degreesgalvtop"],keys[[1]]==7,values[[1]]/.Symbol["d
keys[[1]]==8,values[[1]]/.Symbol["degreestempendrtf"],keys[[1]]==9,values[[1]]/.Symbol
keys[[1]]==10,values[[1]]/.Symbol["degreestempfinalcool"],keys[[1]]==11,values[[1]]/.
Symbol["degreesoilingweightbottom"]],Which[keys[[2]]==1,values[[2]]/.Symbol["degreesste
keys[[2]]==2,values[[2]]/.Symbol["degreesthickness"],keys[[2]]==3,values[[2]]/.Symbol[
keys[[2]]==4,values[[2]]/.Symbol["degreescoatwttop"],keys[[2]]==5,values[[2]]/.
Symbol["degreescoatwtbottom"],
keys[[2]]==6,values[[2]]/.Symbol["degreesgalvtop"],keys[[2]]==7,values[[2]]/.Symbol["d
keys[[2]]==8,values[[2]]/.Symbol["degreestempendrtf"],keys[[2]]==9,values[[2]]/.Symbol
keys[[2]]==10,values[[2]]/.Symbol["degreestempfinalcool"],keys[[2]]==11,values[[2]]/.
Symbol["degreesoilingweightbottom"]]},{i,Length@x}];
correlation=N@SpearmanRho[threadedcorrelationlists[[All,1]],threadedcorrelationlists[[Al
```

```
End[];
EndPackage[];
```