Arabidopsis Thaliana Network

Example for GeneNet 1.2.13 (August 2015) or later

This note reproduces the "Arabidopsis thaliana" network example from R. Opgen-Rhein and K. Strimmer. 2007. *From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data.* BMC Syst. Biol. **1**: 37. (http://dx.doi.org/10.1186/1752-0509-1-37)

The original source of the data is Smith et al. 2004. *Diurnal changes in the transcriptom encoding enzymes of starch metabolism provide evidence for both transcriptional and posttranscriptional regulation of starch metabolism in Arabidopsis leaves.* Plant Physiol. **136**: 2687-2699.

This example was suggested by Papapit Ingkasuwan, Division of Biotechnology, School of Bioresources and Technology, King Mongkut's University of Technology Thonburi, Bangkok, Thailand.

Inspect Data

```
library("GeneNet")
```

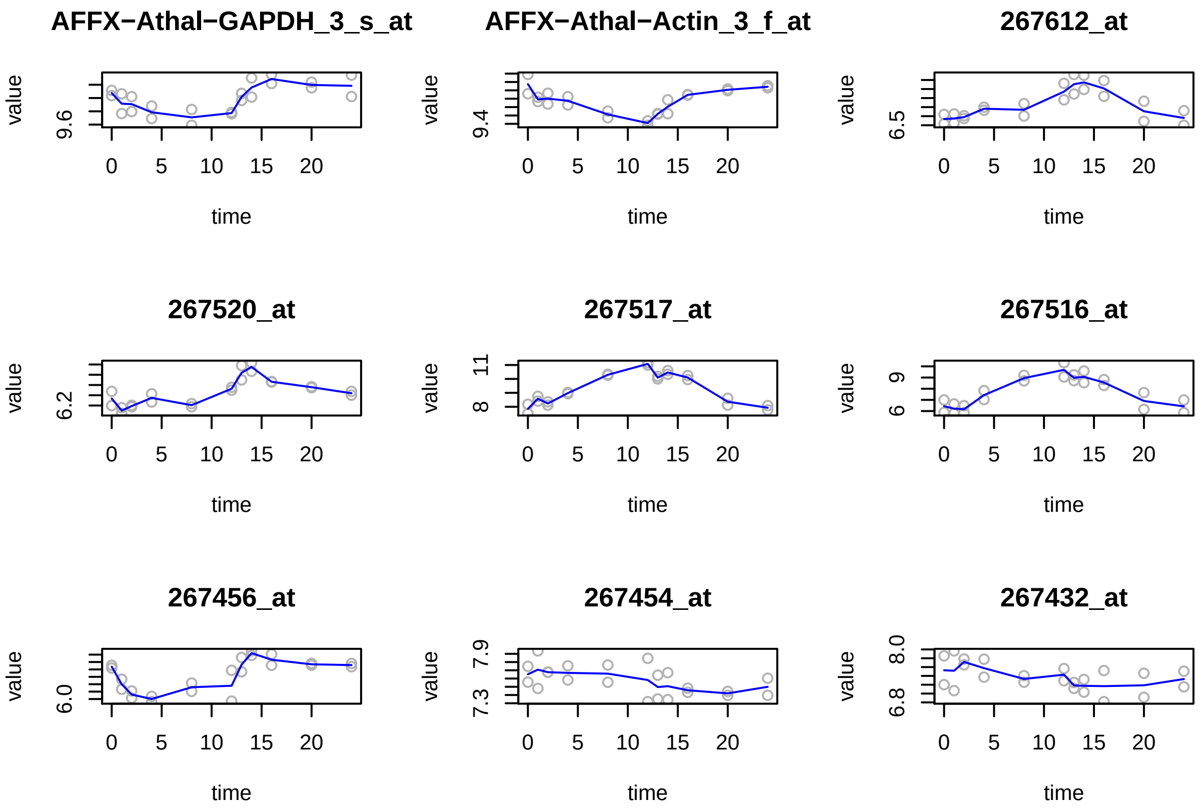
```
## Loading required package: corpcor
## Loading required package: longitudinal
## Loading required package: fdrtool
```

```
data("arth800")
summary(arth800.expr)
```

```
## Longitudinal data:
##  800 variables measured at 11 different time points
##  Total number of measurements per variable: 22
##  Repeated measurements: yes
##
##  To obtain the measurement design call 'get.time.repeats()'.
```

Plot time series:

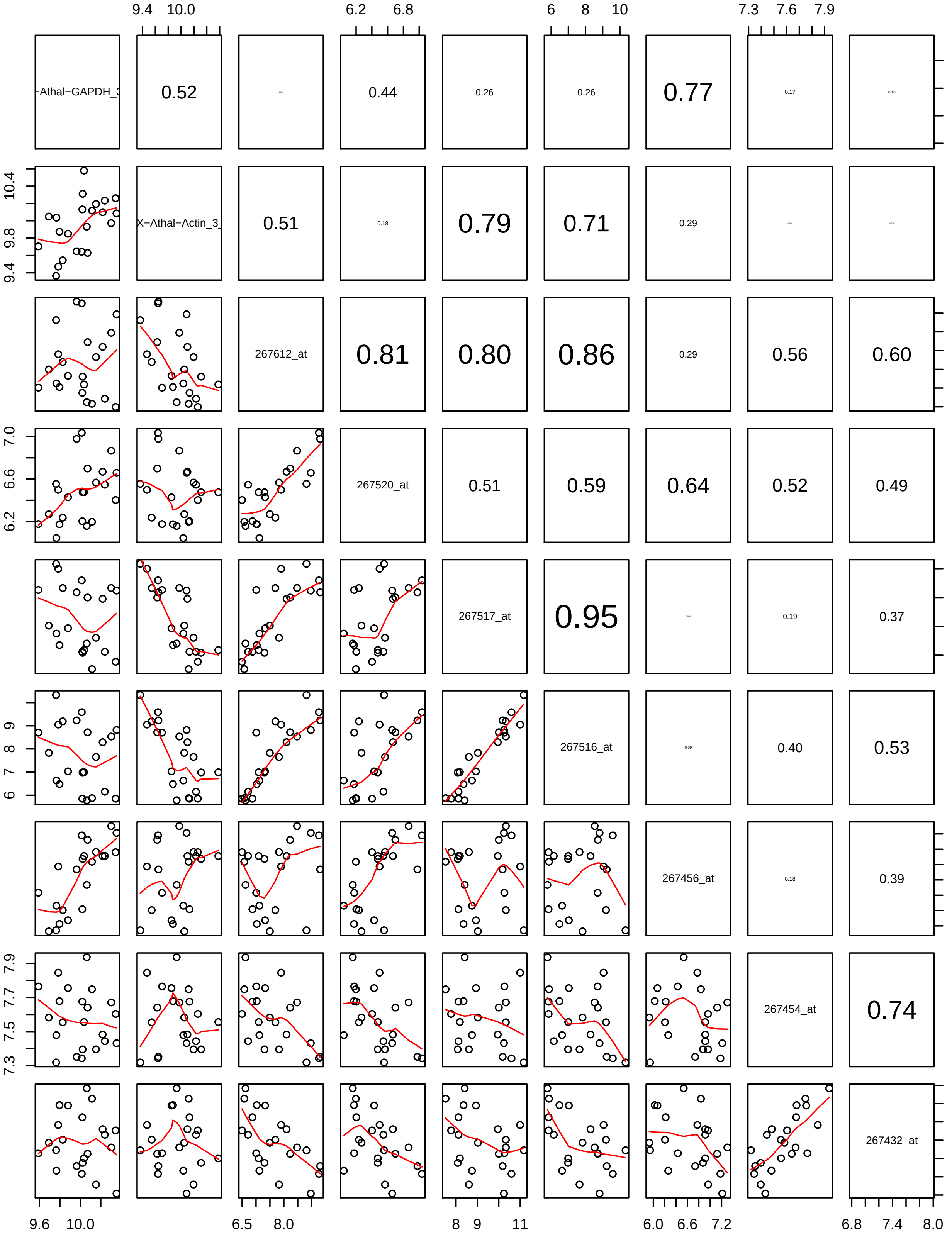
```
plot(arth800.expr, 1:9)
```



Inspect pairwise scatter plots:

```
panel.cor = function(x, y, digits=2, prefix="", cex.cor)
{
    usr = par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r = abs(cor(x, y))
    txt = format(c(r, 0.123456789), digits=digits)[1]
    txt = paste(prefix, txt, sep="")
    if(missing(cex.cor)) cex = 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex * r)
}
```

```
pairs(arth800.expr[,1:9], lower.panel=panel.smooth, upper.panel=panel.cor)
```


9.4
10.0
6.2
6.8
6
8
10
7.3
7.6
7.9
-Athal-GAPDH_3
0.52
0.063
0.44
0.26
0.26
0.77
0.17
0.10
10.0
9.6
10.4
9.8
9.4
X-Athal-Actin_3_
0.51
0.18
0.79
0.71
0.29
0.065
0.036
267612_at
0.81
0.80
0.86
0.29
0.56
0.60
8.5
7.5
6.5
7.0
6.6
6.2
267520_at
0.51
0.59
0.64
0.52
0.49
267517_at
0.95
0.058
0.19
0.37
11
10
9
8
9
8
7
6
267516_at
0.09
0.40
0.53
267456_at
0.18
0.39
7.2
6.6
6.0
7.9
7.7
7.5
7.3
267454_at
0.74
267432_at
8.0
7.6
7.2
6.8
9.6
10.0
6.5
8.0
8
9
11
6.0
6.6
7.2
6.8
7.4
8.0

Compute Partial Correlations and Select Relevant Edges

```
pcor.dyn = ggm.estimate.pcor(arth800.expr, method = "dynamic")
```

```
## Estimating optimal shrinkage intensity lambda (correlation matrix): 0.185
```

```
arth.edges = network.test.edges(pcor.dyn,direct=TRUE)
```

```
## Estimate (local) false discovery rates (partial correlations):
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting

##
## Estimate (local) false discovery rates (log ratio of spvars):
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting
```

```
dim(arth.edges)
```

```
## [1] 319600     10
```

We use the strongest 150 edges:

```
arth.net = extract.network(arth.edges, method.ggm="number", cutoff.ggm=150)
```

```
##
## Significant edges:  150
##     Corresponding to  0.05 %  of possible edges
##
## Significant directions:  10516
##     Corresponding to  3.29 %  of possible directions
## Significant directions in the network:  55
##     Corresponding to  36.67 %  of possible directions in the network
```

Construct Graph

```
library("graph")

node.labels = as.character(1:ncol(arth800.expr))
gr = network.make.graph( arth.net, node.labels, drop.singles=TRUE)
```

Some information about the graph Number of nodes:

```r
num.nodes(gr)
```

```
## [1] 107
```

Correlations:

```r
edge.info(gr)$weight
```

```
##    8~209   13~331   13~422   20~573   20~781   26~781   47~793   47~422 
## -0.03088  0.03289  0.03416 -0.03221  0.03300 -0.03751  0.03101  0.03302 
##   47~736   47~714   47~414   47~331   47~629   47~452   47~479   61~480 
##  0.03460  0.03553  0.03559  0.03897  0.04035  0.04036  0.04235  0.03112 
##   61~111   63~198   63~738   63~444   78~726   81~377   81~712   81~198 
##  0.03414 -0.03085 -0.03953 -0.04345 -0.03184  0.03139  0.03157  0.03184 
##   81~368   81~108   81~211    81~61   81~666   81~636   81~665   81~793 
##  0.03244  0.03311  0.03314  0.03334 -0.03342  0.03367 -0.03382  0.03402 
##   81~248   81~622   81~144   81~111   81~570   81~767   86~738   86~181 
## -0.03423 -0.03501 -0.03593  0.03912  0.04281 -0.05957  0.03068  0.03268 
##  100~296  100~245  100~412  101~443  108~603  108~272  111~537  111~496 
##  0.03181  0.03278  0.03401  0.03449  0.03091 -0.03117 -0.03102  0.03405 
##  126~783  155~679  181~783  198~779  198~686  209~738  226~573  269~783 
## -0.03384  0.03345  0.04190 -0.03128 -0.03631 -0.03711 -0.03118  0.03479 
##  272~289  272~603  272~414  272~560  272~452  272~726  289~414  289~452 
##  0.03364 -0.03782  0.03797  0.04137  0.04530 -0.05022  0.03132  0.03479 
##  289~786  289~726  299~444  328~519  331~714  331~479  331~414  331~452 
## -0.03598 -0.03933 -0.03135  0.03108  0.03194  0.03278  0.03404  0.03628 
##  331~422  414~786  414~452  414~726  422~479  422~736  422~714  422~629 
##  0.03881 -0.03464  0.03520 -0.03543  0.03104  0.03277  0.03573  0.03663 
##  422~627  443~565  443~573  443~600  444~738  452~714  452~726  452~603 
## -0.03972  0.03092 -0.03130 -0.03159  0.03340  0.03216 -0.03400 -0.03526 
##  479~677  479~714  479~793  479~629  480~738  539~360  539~758  539~778 
##  0.03261  0.03656  0.03837  0.04523 -0.04365  0.03080 -0.03104  0.03219 
##  539~596   539~93  539~585  539~197    539~4  540~738  558~783  558~342 
##  0.03269 -0.03381 -0.03483  0.03614  0.04977  0.04087  0.03174 -0.03211 
##   558~96  558~269  558~363  558~161  558~739  558~126  558~256  560~793 
##  0.03223  0.03271  0.03387 -0.03596 -0.03647 -0.03727 -0.04139 -0.03093 
##  560~627  560~726  570~623  570~767  570~256  570~234  570~460  570~219 
##  0.03807 -0.04152  0.03122 -0.03127 -0.03187  0.03225 -0.03322  0.03364 
##  570~378  570~135  570~699  570~585  570~576  570~422   570~38  570~554 
## -0.03373  0.03376  0.03565  0.03722  0.03794 -0.03815 -0.03951 -0.03989 
##  570~651   570~61  570~598  570~111  570~464  600~699  603~781  627~738 
##  0.04060  0.04262  0.04284  0.04411  0.04751  0.03330  0.03092 -0.03246 
##  627~661  627~281  629~714  629~793  636~738  677~714  679~783  726~786 
##  0.03495  0.03570  0.03806  0.04579  0.03234  0.03134  0.03705  0.03638 
##  779~798  781~783  783~640  783~187  783~547  783~454 
##  0.03063  0.03146 -0.03264 -0.03306  0.03761 -0.04094
```

Number of directed (“forward”) and undirected (“none”) edges:

```r
table(  edge.info(gr)$dir )
```

```
## 
## forward    none 
##      55      95
```

Well-Connected Nodes

Nodes connected with many edges:

```r
sort(node.degree(gr), decreasing=TRUE)[1:10]
```

```
## 570  81 783  47 422 558 452 539 738 272
##  20  17  10   9   9   9   8   8   8   7
```

```r
arth800.descr[570]
```

```
## [1] "AP2 transcription factor - like protein"
```

```r
arth800.descr[81]
```

```
## [1] "ATRPAC43; DNA binding / DNA-directed RNA polymerase; DNA-directed RNA polymerase, putative, iden
```

```r
arth800.descr[558]
```

```
## [1] "structural constituent of ribosome; 60S ribosomal protein L35 (RPL35C), various ribosomal L35 pr
```

```r
arth800.descr[539]
```

```
## [1] "DNA binding / transcription factor; basic helix-loop-helix (bHLH) family protein, contains Pfam
```

```r
arth800.descr[783]
```

```
## [1] "RNA binding / RNA methyltransferase; tRNA/rRNA methyltransferase (SpoU) family protein, similar
```

Plot Network

```r
library("Rgraphviz")
```

```
## Loading required package: grid
```

For a more beautiful plot of the network set node and edge parameters: Set global node and edge attributes:

```r
globalAttrs = list()
globalAttrs$edge = list(color = "black", lty = "solid", lwd = 1, arrowsize=1)
globalAttrs$node = list(fillcolor = gray(.95), shape = "ellipse", fixedsize = FALSE)
```

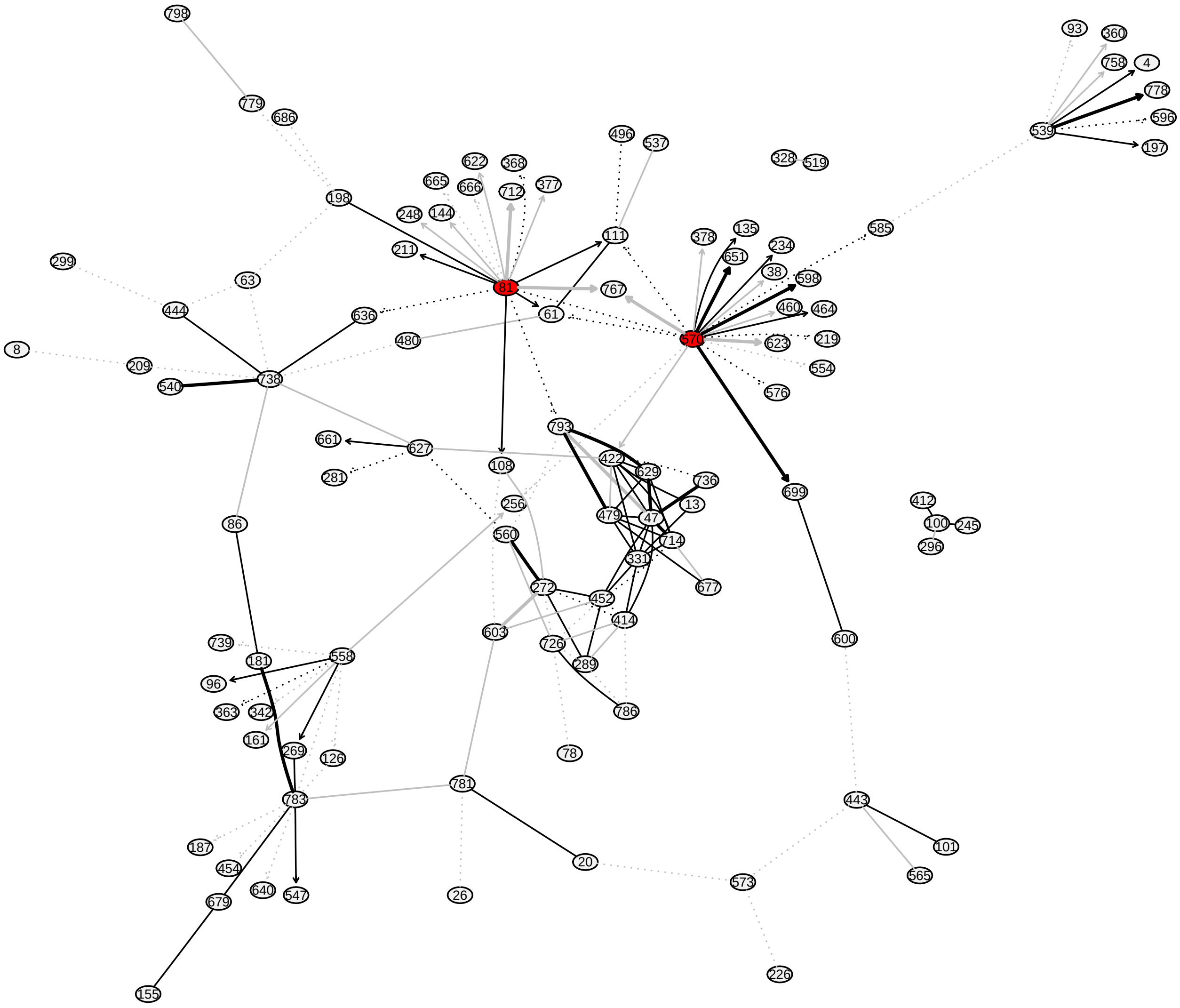
Set attributes of some particular nodes:

```
nodeAttrs = list()
nodeAttrs$fillcolor = c('570' = "red", "81" = "red") # highlight hub nodes
```

Set edge attributes:

```
edi = edge.info(gr) # edge directions and correlations
edgeAttrs = list()
edgeAttrs$dir =  edi$dir # set edge directions
cutoff = quantile(abs(edi$weight), c(0.2, 0.8)) # thresholds for line width / coloring
edgeAttrs$lty = ifelse(edi$weight < 0, "dotted", "solid") # negative correlation
edgeAttrs$color = ifelse( abs(edi$weight <= cutoff[1]), "grey", "black") # lower 20% quantile
edgeAttrs$lwd = ifelse(abs(edi$weight >= cutoff[2]), 2, 1) # upper 20% quantile
```

```
plot(gr, attrs = globalAttrs, nodeAttrs = nodeAttrs, edgeAttrs = edgeAttrs, "fdp")
```


798
779
686
93
360
758
4
778
596
539
197
496
537
328
519
622
368
665
666
712
377
198
248
144
111
378
135
585
651
234
211
38
598
299
63
81
767
460
464
444
636
61
480
570
623
219
8
209
554
540
738
576
793
661
627
422
108
629
736
281
699
412
13
256
479
47
100
245
86
560
714
296
331
272
677
452
414
603
600
739
726
558
181
289
96
786
363
342
161
269
126
78
781
783
443
101
187
20
454
565
640
547
573
679
26
226
155