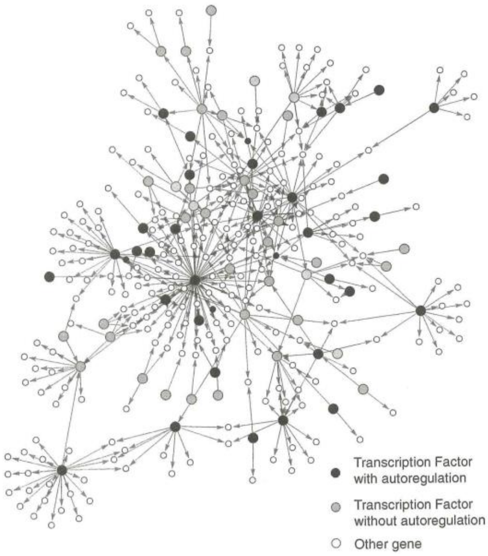
Matematik i systembiologi: netværksmotiver

Af Jan Brønnum Sørensen, Aalborg City Gymnasium

Bogen "An Introduction to Systems Biologi" af Uri Alon er anvendt som inspiration til denne opgavesamling.
Opgaverne er tænkt i samspil med et forløb i biologi eller bioteknologi, men kan også anvendes alene i matematik. Opgaverne forudsætter kendskab til kombinatorik, binomialfordeling og simulering, og kan derfor anvendes på b-niveau.
Der medfølger løsninger til alle opgaverne.

Opgave 1: Autoregulering i gennetværk

Figuren viser 424 gener (punkterne, uanset størrelse) og 519 interaktioner mellem generne (pilene) i et udsnit af e-coli bakterier. Pilens retning angiver, hvordan det ene gen påvirker det andet. Ud over de viste pile, er der desuden en interaktion fra hvert af de sorte gener til genet selv, en såkaldt autoregulering, som dog ikke er tegnet på figuren med en pil.
Spørgsmålet er nu, om autoregulering er mere forekommende i dette netværk, end hvis interaktionerne opstår helt tilfældet mellem generne.



1.1: Tæl antallet af gener med autoregulering på figuren.
1.2: Beregn sandsynligheden for, at en tilfældigt placeret interaktion i netværket med 424 gener er en autoregulering.
1.3: Beregn det forventelige antal gener med autoregulering i et netværk med 424 gener og 519 tilfældigt placerede interaktioner. Overvej, hvordan binomialfordelingen kan bringes i spil til dette.
1.4: Beregn spredningen i antallet af gener med autoregulering.
1.5: Overvej ud fra dine svar i opgave 1.3 og 1.4, om det reelle antal i opgave 1.1 er stort i forhold til en tilfældig placering, så man derfor kan sige, at autoregulering forekommer signifikant oftere i netværket end tilfældige placeringer ville give.
1.6: Lav en simulering af 519 tilfældige interaktioner i et netværk med 424 gener, og tæl antal selvreguleringer i simuleringen.
1.7: Gentag simuleringen et større antal gange og tegn et pindediagram (eller prikdiagram) over antal autoreguleringer i hver simulering. Dette kan enten gøres i samarbejde med resten af klassen, så hver gruppe laver et mindre antal simuleringer, som så samles, eller ved programmering af gentagelserne. Sammenlign resultatet med det virkelige antal autoreguleringer og med de beregnede værdier fra opgave 1.3 og 1.4.
1.8: Lav tilsvarende beregninger for en netværk med 100 gener og 500 interaktioner.

Opgave 2: Parvis regulering

Lad os nu se på interaktion mellem to gener med fokus på den situation, at der både er en interaktion fra A til B (A-B) og en interaktion fra B til A (B-A). Det vil vi kalde en parvis regulering.

Betragt først et netværk med 2 gener, A og B, og præcis 2 interaktioner.
2.1: Tegn de forskellige muligheder for, hvordan netværket kan se ud. Overvej grundigt, hvor mange forskellige muligheder der er.
2.2: Hvis interaktionerne placeres tilfældigt, hvad er sandsynligheden så for, at netværket bliver af en type, hvor A-B og B-A begge indgår?

Betragt nu i stedet et netværk med 100 gener.
2.3: Hvor mange forskellige mulige interaktioner er der i dette netværk?
Lad os nu placere 500 interaktioner tilfældigt i netværket. Bemærk, at dette tal er lille sammenlignet med det totale antal mulige interaktioner.
2.4: Hvad er sandsynligheden for, at en bestemt interaktion A-B er med i netværket?
2.5: Hvad er sandsynligheden (approksimativt) for, at både interaktionen A-B og interaktionen B-A er med i netværket?
2.6: Hvor mange forskellige kombinationer af 2 gener kan der udvælges i netværket?
2.7: Hvad er det gennemsnitlige antal parvise reguleringer i netværket (approksimativt)?

2.8: Lav en computersimulering af et netværk med 100 gener og 500 interaktioner, hvor interaktionerne placeres tilfældigt og tæl antal parvise reguleringer i det simulerede netværk.
2.9: Gentag simuleringen i opgave 2.8 et antal gange og tegn et pindediagram (eller prikdiagram) over antal parvise reguleringer. Sammenlign resultatet med dit svar i opgave 2.6.

Opgave 3: Interaktioner mellem 3 gener

Vi vil nu betragte mønstre af interaktioner mellem 3 gener.
3.1: Hvor mange forskellige interaktioner er der i et system med 3 gener?
3.2 Hvor mange måder kan man udvælge præcis 3 interaktioner blandt de mulige interaktioner mellem 3 gener?
3.3 Hvor mange måder kan man udvælge 3 interaktioner blandt 3 gener, så der dannes en ”ring” på formen A-B, B-C og C-A? Dette kaldes et ”feed backward loop”. Beregn og tegn.
3.4 Hvor mange måder kan man udvælge 3 interaktioner blandt 3 gener, så der er en interaktion fra det ene gen til begge de to øvrige, og yderligere en interaktion fra det ene af de øvrige gener til det andet? Altså A-B, A-C og B-C. Beregn og tegn.

Opgave 4: Feed backward loop

Vi vil nu se på nærmere på feed backward loop. Vi vil fortsat se på et system med 100 gener og 500 tilfældigt placerede interaktioner.
4.1: Hvad er sandsynligheden for, at en helt bestemt interaktion A-B er med i netværket?
4.2: Hvad er sandsynligheden (approksimativt) for, at A-B og B-C er med i netværket?
4.3: Hvad er sandsynligheden (approksimativt) for, at A-B, B-C og C-A alle er med i netværket?
4.4: På hvor mange forskellige måder kan der dannes en sådan ”ring” blandt de 3 gener A, B og C?
4.5: Hvor mange forskellige kombinationer af 3 gener kan der udvælges i netværket?
4.6: Hvad er det gennemsnitlige antal feed backward loop i netværket (approksimativt)?

4.7: Lav en passende computer simulering og optæl antal feed backward loop. Optællingen er dog ganske besværlig manuelt, så det bør nok kun gøres, hvis du har mulighed for at automatisere / programmere dette
4.8: Lav flere computersimuleringer og sammenlign resultaterne med det teoretiske resultat fra 3.6.

Opgave 5: Feed forward loop

Som i opgave 4, blot skal du nu se på feed forward loop. Lav tilsvarende beregninger og simuleringer.

Opgave 6: Sammenfatning

Opstil de forventede antal autoreguleringer, parvise regulerer og hver af de to typer loops med 3 gener for en netværk med 100 gener og 500 interaktioner.

Løsning til opgave 1.

1.1 Der er 33 autoreguleringer, hvis jeg ellers har talt rigtigt.

1.2 Da en interaktion kan gå til 424 forskellige gener, og netop én af disse muligheder giver en autoregulering, er sandsynligheden 1/424.

1.3 Når der placeres 519 interaktioner, kan det ses som en stikprøve på 519 ud af en population på 424^2=179776 mulige interaktioner. Da samme interaktion ikke kan forekomme flere gange, er der tale om en stikprøve *uden* tilbagelægning, så binomialfordelingen passer ikke perfekt, da det jo skulle være *med* tilbagelægning. Men da stikprøven er meget lille i forhold til det samlede antal, vil binomialfordelingen være en nogenlunde approksimation. Dermed bliver det forventede antal på $n \cdot p = 519 \cdot 1/424 \approx 1{,}2$, så ca. 1 autoregulering vil være forventet.

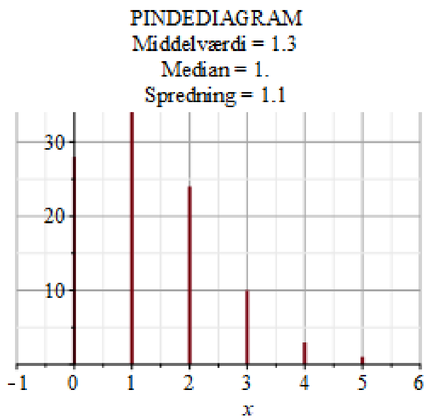
1.4 Spredningen kan tilsvarende udregnes vha. formlen $\sigma = \sqrt{n \cdot p \cdot (1-p)} \approx 1{,}1$

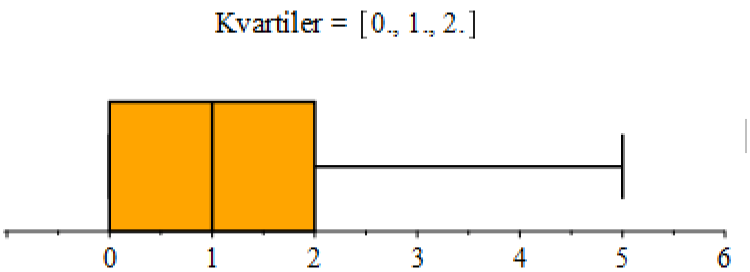
1.5 De 33 autoreguleringer på billedet er således langt ud over, hvad man ville forvente, hvis interaktionerne forekommer tilfældigt.

1.6 Simuleringen kan f.eks. laves ved at lave to lister med 519 tilfældige hele tal fra 1 til 424 i hver. Så skal det kontrolleres, at samme interaktion ikke er med flere gange, og derefter kan antallet af autoreguleringer tælles op.
I Excel kan kommandoen ”slumpmellem(1;424)” benyttes til at lave sådanne tilfældige tal, men det samme kan naturligvis også gøres i ethvert CAS program. Hvis man yderligere er skarp på sit program, kan man få det til selv at tælle op, men dette er ikke nødvendigt for at løse opgaven.
Her ses en sådan løsning i Maple, dog uden kontrollen af, om samme interaktion er med flere gange.

```
restart
with(Gym) :
with(Statistics) :
randomize( ) :
n := 519 :
m := 424 :
fra := [seq(rand(1 ..m)( ), i = 1 ..n)] :
til := [seq(rand(1 ..m)( ), i = 1 ..n)] :
antal := 0 :
for i from 1 to n do
   if fra[i] = til[i] then antal := antal + 1 : end if;
end do;
antal = 1
```

1.7 Resultaterne af 100 simuleringer kan f.eks. se således ud. Det stemmer fint med de beregnede værdier.





1.8 Der forventes 500/100 = 5 autoreguleringer.

Løsning til opgave 2.

2.1 Der er 4 forskellige interaktioner, hvoraf præcis 2 skal vælges, så antallet af kombinationer er K(4,2)=6. De ser således ud: A-A og A-B, A-A og B-A, A-A og B-B, A-B og B-A, A-B og B-B, B-A og B-B.

2.2 Da netop 1 af de 6 muligheder opfylder, at både A-B og B-A indgår, er sandsynligheden 1/6. Det kan også udregnes ved at indse, at sandsynligheden for, at den første interaktionen er A-B eller B-A er 1/2. Derefter er sandsynligheden 1/3 for, at den anden interaktion også bliver A-B eller B-A. Vha. multiplikationsprincippet givet det ligeledes 1/6.

2.3 Det er 100^2=10000 mulige interaktioner, da ethvert gen kan have en ”pil” til ethvert andet gen eller til sig selv.

2.4 Med 500 interaktioner ud af 10000 mulige, er sandsynligheden for en helt bestemt interaktion 500/10000 = 1/20.

2.5 Der er ikke helt uafhængighed, men med så få interaktioner (500) og af så mange mulige (10000), kan vi godt alligevel lade som om, det er tilfældet. Så bliver sandsynligheden for både at få A-B og B-A blot en anvendelse af multiplikationsprincippet, så det givet $(1/20)^2$=1/400.

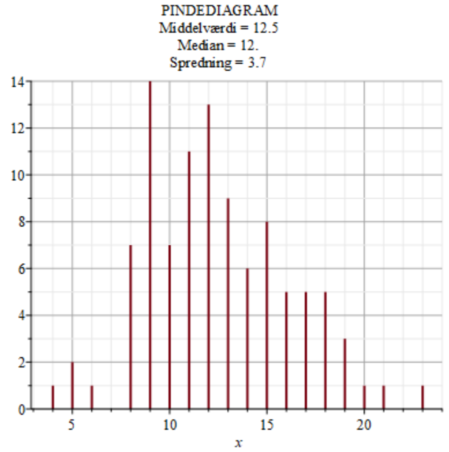
2.6 Der skal vælges 2 gener ud af 100, så det er K(100,2)=4950 forskellige muligheder.

2.7 Hvis vi igen benytter antagelsen om uafhængighed, fordi der kun er få interaktioner i netværket, bliver det forventede antal 1/400 af de 4950, altså ca. 12,4.

2.8 Simuleringen foregår på samme måde som i opgave 1, blot er optælling lidt mere besværlig, da man nu skal listen igennem for at finde par, hvor både A-B og B-A er med.
Her ses en sådan løsning i Maple, dog uden kontrollen af, om samme interaktion er med flere gange.

```
restart
randomize( ) :
n := 500 :
m := 100 :
tæl := 0 :
fra := [seq(rand(1 ..m)( ), i = 1 ..n)] :
til := [seq(rand(1 ..m)( ), i = 1 ..n)] :
for i from 1 to n − 1 do
 for j from i + 1 to n do
   if fra[i] = til[j] and til[i] = fra[j] then tæl := tæl + 1 : end if;
  end do;
end do:
tæl = 14
```

2.9 Resultaterne af 100 simuleringer kan f.eks. se således ud. Det stemmer fint med de beregnede approksimative værdier.



Løsning til opgave 3

3.1 Fra hvert gen er der 3 mulige interaktioner, én selvregulering og én til hver af de to øvrige gener. Der er derfor samlet 9 mulige interaktioner.

3.2 K(9,3) = 84

3.3 Der er blot to muligheder. Enten A-B, B-C, C-A eller A-C, C-B, B-A. Med andre ord kan ringen vende den ene vej rundt eller den anden vej rundt.

3.4 Der er 3 muligheder for at vælge det gener, som har interaktioner til begge de øvrige, og derefter to mulighed for at vælge, hvilken retning interaktionen mellem de øvrige skal vende. Så samlet er der 6 muligheder. De ser således ud:
A-B, A-C, B-C --- A-B, B-C, C-B --- B-A, B-C, A-C --- B-A, B-C, C-A ---, C-A, C-B, A-B --- C-A, C-B, B-A

Løsning til opgave 4

4.1 Helt som i opgave 2.4 er dette p=1/20.

4.2 Som i 2.5 regnes som om der er uafhængighed. Så det giver $p^2=1/400$.

4.3 På samme vis giver det så $p^3=1/8000$.

4.4 Dette blev udregnet i 3.3, så der er 2 muligheder.

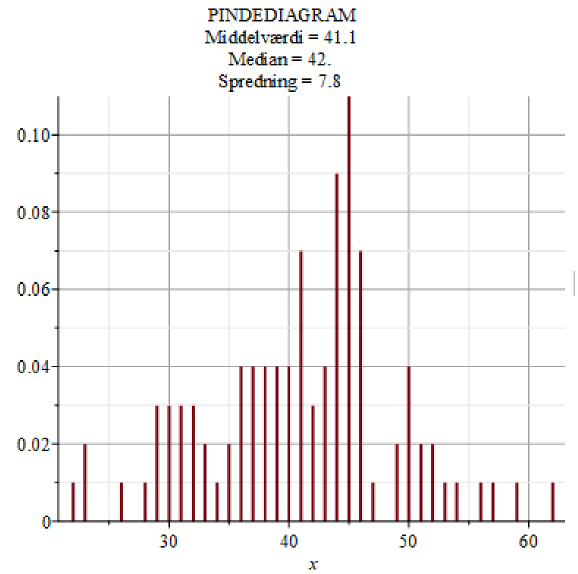
4.5 K(100,3) = 161700

4.6 Samlet giver det så et gennemsnitligt antal på $2 \cdot p^3 \cdot K(100,3) \approx 40$

4.7 Simuleringen laves på helt samme måde som i opgave 1 og 2, men optælling er langt mere besværlig, da man nu skal se på interaktioner mellem 3 gener. Her ses Maple kode til en sådan simulering:

```
Feedbackward Loop (A-B, B-C og C-A)
restart
randomize( ) :
n := 500 :
m := 100 :
p := n/m^2 = 1/20
antal := 0 :
fra := [seq(rand(1 ..m)( ), i = 1 ..n)] :
til := [seq(rand(1 ..m)( ), i = 1 ..n)] :
for i from 1 to n -2 do
 for j from i + 1 to n -1 do
  for k from j + 1 to n do
   if til[i] = fra[j] and til[j] = fra[k] and til[k] = fra[i] then antal := antal + 1 : end if:
   if til[i] = fra[k] and til[k] = fra[j] and til[j] = fra[i] then antal := antal + 1 : end if:
  end do:
 end do:
end do:
antal = 37
```

4.8 Resultaterne af 100 simuleringer kan f.eks. se således ud.



Løsning til opgave 5

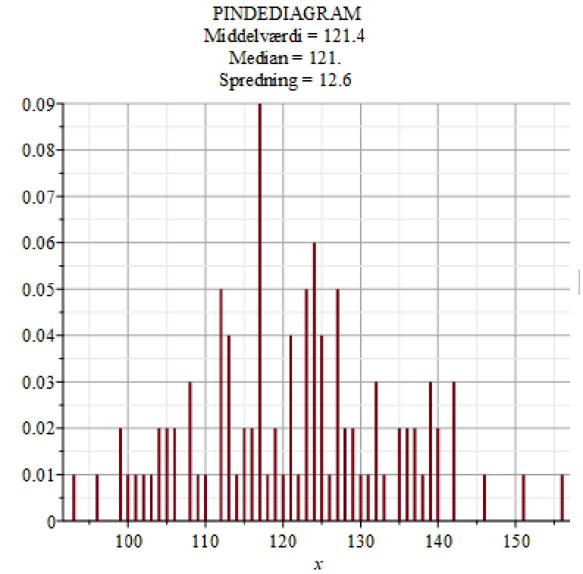
Det gennemsnitlige antal feed forward loops bliver 3 gange så stort som den gennemsnitlige antal feed backward loops, da der er 6 mulige konfigurationer for feed forward og 2 for feed backward.

Maple kode:

Feedforward Loop (A-B, A-C og B-C)

```
restart
randomize( ) :
with(Gym) :
n := 500 :
m := 100 :
p := n/m^2 = 1/20
antal := 0 :
fra := [seq(rand(1 ..m)( ), i = 1 ..n)] :
til := [seq(rand(1 ..m)( ), i = 1 ..n)] :
 for i from 1 to n − 2 do
 for j from i + 1 to n − 1 do
   for k from j + 1 to n do
     if fra[i] ≠ til[i] and fra[j] ≠ til[j] and fra[k] ≠ til[k] then
      if fra[i] = fra[k] and til[i] = fra[j] and til[j] = til[k] then antal := antal + 1 :
      elif fra[i] = fra[j] and til[i] = fra[k] and til[j] = til[k] then antal := antal + 1 :
      elif fra[j] = fra[i] and til[j] = fra[k] and til[i] = til[k] then antal := antal + 1 :
      elif fra[j] = fra[k] and til[j] = fra[i] and til[i] = til[k] then antal := antal + 1 :
      elif fra[k] = fra[i] and til[k] = fra[j] and til[i] = til[j] then antal := antal + 1 :
      elif fra[k] = fra[j] and til[k] = fra[i] and til[i] = til[j] then antal := antal + 1 :
      end if;
     end if;
   end do;
  end do;
end do:
antal = 138
```

Resultaterne af 100 simulering kan se således ud:



Løsning til opgave 6.

Der forventes ca. 5 autoreguleringer, ca. 12 parvise reguleringer, ca. 40 feed backward loops og ca. 120 feed forward loops i netværket.