

# Tarea 05

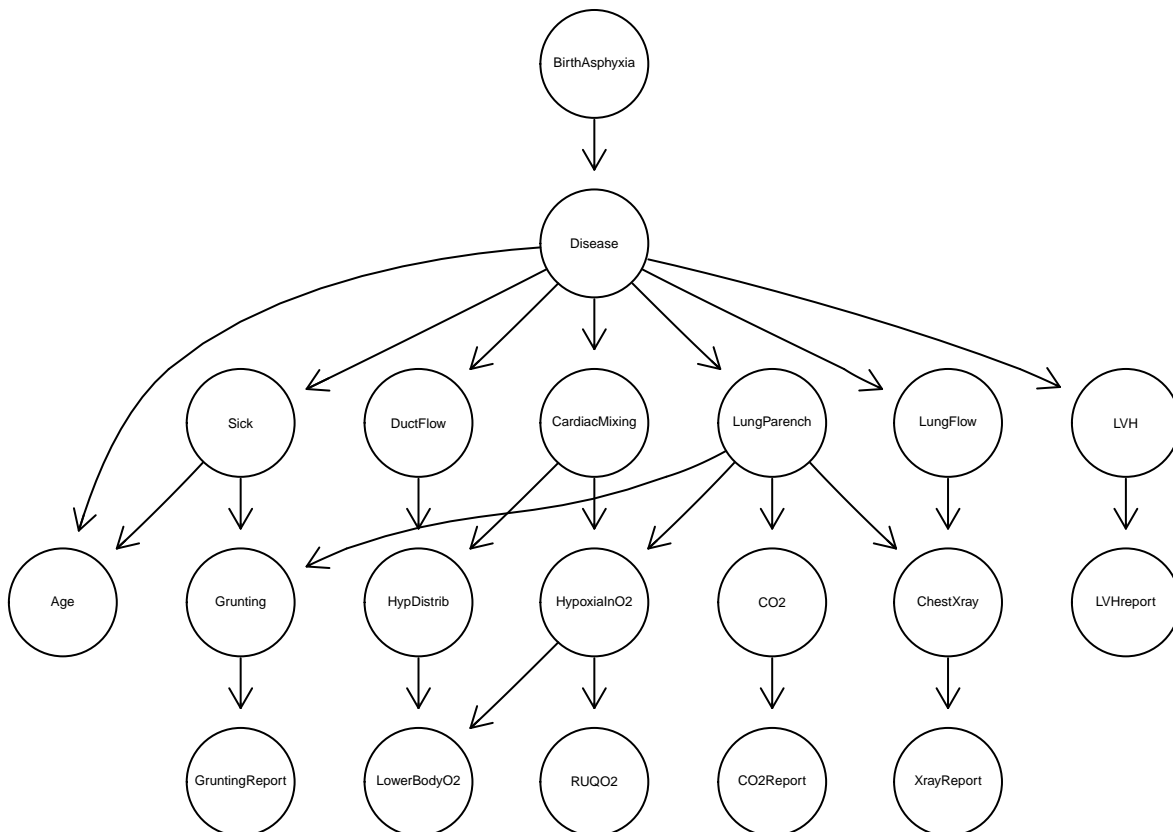
Mario Becerra 124362

17/02/2015

En este ejercicio utilizaremos la red Child (D. J. Spiegelhalter, R. G. Coewll). El objetivo es utilizar el algoritmo de hill climbing para recuperar la estructura de la red a partir de una muestra.

```
library(bnlearn)
child_net <- read.net("child.net")
graphviz.plot(child_net)
```

```
## Loading required namespace: Rgraphviz
```



Los datos muestra\_ch.dat fueron generados simulando de la red Child y los utilizaremos para recuperar la estructura utilizando como score el AIC y el BIC. Repetiremos la estimación tomando muestras de tamaño n=500,1000.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
```

```
##
##      filter
##
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
```

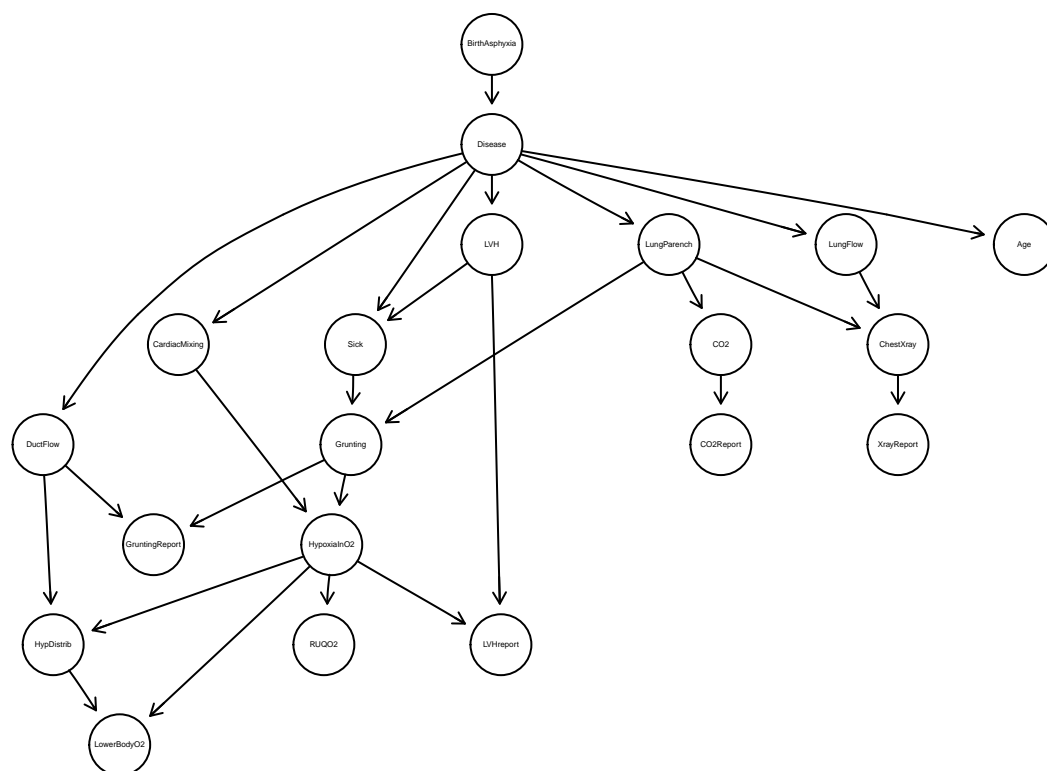
```
library(bnlearn)
set.seed(124362)
child <- read.csv('muestra_ch.dat')
child_500 <- sample_n(child, 500)
child_1000 <- sample_n(child, 1000)
```

En los próximos incisos utilizaremos la muestra de tamaño 500.

- Supón ahora que los expertos te informan que BirthAsphyxia debe ser nodo raíz (no puede haber arcos dirigidos hacia BirthAsphyxia) y GruntingReport, CO2Report, son nodos sin hijos (hojas). Incorpora esta información en tu red (utiliza el parámetro blacklist de la función hc) y aprende nuevamente las redes usando los criterios AIC y BIC

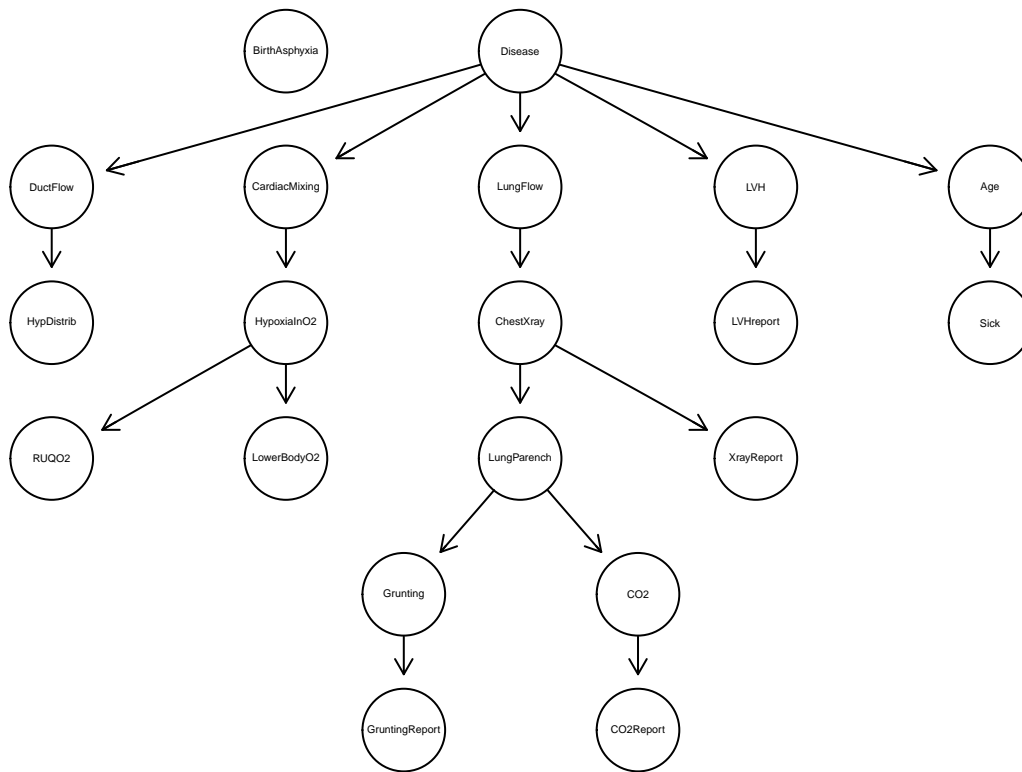
```
vars<-names(child)
black_1<-data.frame(from=vars[-1], to=vars[1])
black_2<-data.frame(from=vars[18], to=vars[-18])
black_3<-data.frame(from=vars[19], to=vars[-19])
blacklist<-rbind(black_1, black_2, black_3)
net_1_aic <- hc(child_500, score='aic', blacklist=blacklist)
net_1_bic <- hc(child_500, score='bic', blacklist=blacklist)
graphviz.plot(net_1_aic, main='AIC, n=500')
```

AIC, n=500



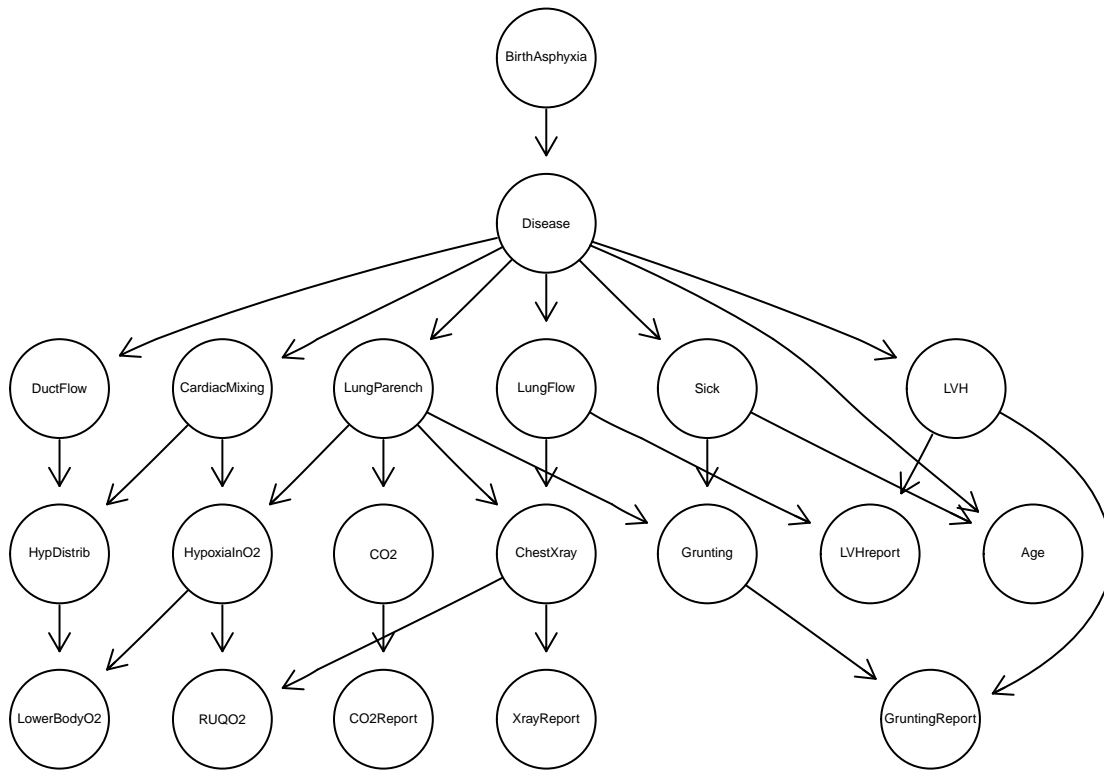
```
graphviz.plot(net_1_bic, main='BIC, n=500')
```

BIC, n=500



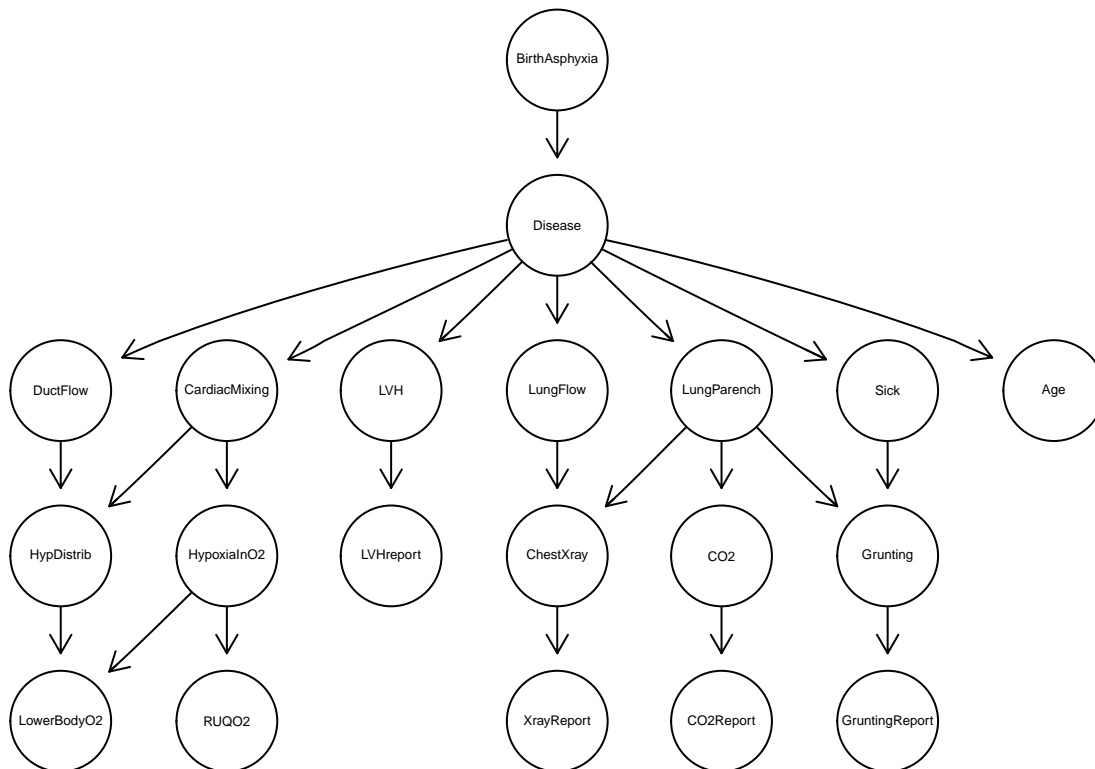
```
net_2_aic <- hc(child_1000, score='aic', blacklist=blacklist)
net_2_bic <- hc(child_1000, score='bic', blacklist=blacklist)
graphviz.plot(net_2_aic, main='AIC, n=1000')
```

AIC, n=1000



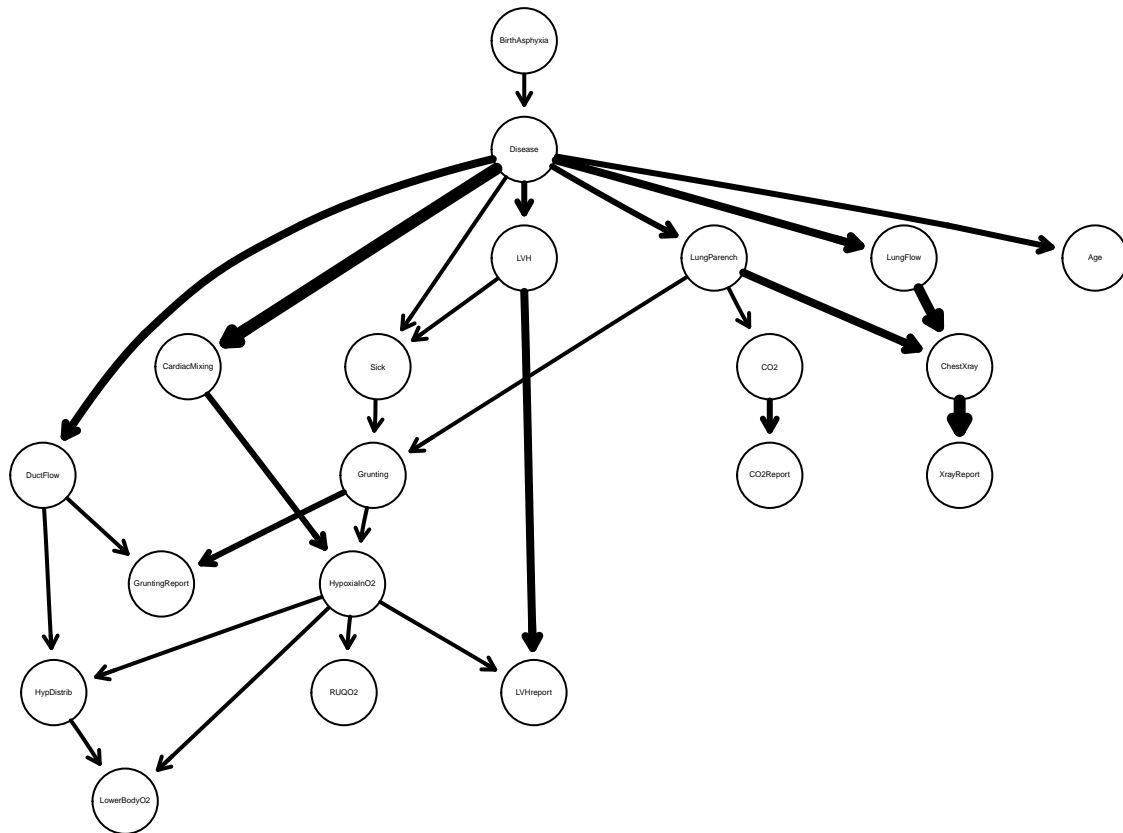
```
graphviz.plot(net_2_bic, main='BIC, n=1000')
```

BIC, n=1000



- Utiliza la función `arc.strength` con criterio AIC para calcular la fuerza de los arcos en la gráfica correspondiente al score AIC en el inciso anterior. La fuerza indica en cuanto reduce el score cada uno de los arcos. Incluye esta información en la gráfica de la red utilizando la función `strength.plot`.

```
strength_1 <- arc.strength(net_1_aic, child_500)
strength.plot(net_1_aic, strength_1)
```



En esta última gráfica se pueden ver las relaciones espurias, puesto que aparecen con arcos menos gruesos. En particular, se puede ver que los arcos más gruesos, es decir, los de la mayor fuerza, son los que eran verdaderos en la red original, como *Disease* a *CardiacMixing* o *Disease* a *DuctFlow*.

Ahora repetiremos el aprendizaje de estructura 500 veces con el objetivo de explorar un número mayor de estructuras y reducir el impacto de máximos locales en el proceso de aprender estructura. Para esto se generan muestras bootstrap de los datos (muestras con reemplazo de la misma dimensión que los datos originales), y para cada muestra se aprende una red. Las redes aprendidas se promedian con el fin de producir un modelo más robusto. Crearemos la red promedio utilizando los arcos presentes en al menos el 80% de las redes. Esta proporción mide la fuerza de cada arco.

- Utiliza la función `boot.strength` (algoritmo `hc`, argumentos `list(score = "aic", blacklist = blist)`) para generar 500 redes. La función te regresará un `data.frame` con la fuerza de cada arco, imprime los 20 arcos con mayor fuerza.

```
boot_strength <- boot.strength(child, R=500, algorithm='hc', algorithm.args=list(score='aic', blacklist=
arrange(boot_strength, desc(strength))[0:20,]
```

##	from	to	strength	direction
## 1	BirthAsphyxia	Disease	1	0.752
## 2	Disease	BirthAsphyxia	1	0.248
## 3	Disease	DuctFlow	1	0.855
## 4	Disease	CardiacMixing	1	0.819
## 5	Disease	LungFlow	1	0.875
## 6	Disease	LVH	1	0.848
## 7	Disease	Sick	1	0.633
## 8	Disease	LungParench	1	0.374

## 9	Disease	Age	1	0.867
## 10	DuctFlow	Disease	1	0.145
## 11	DuctFlow	HypDistrib	1	0.996
## 12	CardiacMixing	Disease	1	0.181
## 13	CardiacMixing	HypDistrib	1	0.989
## 14	CardiacMixing	HypoxiaInO2	1	0.986
## 15	LungFlow	Disease	1	0.125
## 16	LungFlow	ChestXray	1	0.993
## 17	LVH	Disease	1	0.152
## 18	LVH	LVHreport	1	0.899
## 19	Sick	Disease	1	0.367
## 20	Sick	Age	1	0.908

- Utiliza la función `averaged.network` con un parámetro `threshold` de 0.8 para obtener la red promedio.

```
avg_net<-averaged.network(boot_strength, vars,0.8)
graphviz.plot(avg_net)
```

