

Análise Bayesiana para Inferência Causal na COVID-19

José Elvano Moraes

4/1/2021

Análise Bayesiana para Inferência Causal na COVID-19

Dataset = <https://dados.gov.br/dataset/bd-srag-2019>

SRAG 2019 - Banco de Dados de Síndrome Respiratória Aguda Grave Vigilância de Síndrome Respiratória Aguda Grave (SRAG) O Ministério da Saúde, por meio da Secretaria de Vigilância em Saúde (SVS), desenvolve a vigilância da Síndrome Respiratória Aguda Grave (SRAG) no Brasil, desde 2009, devido a pandemia de Influenza A(H1N1)pdm09. A partir disso, a vigilância de SRAG foi implantada na rede de vigilância de Influenza e outros vírus respiratórios, que anteriormente atuava apenas com a vigilância sentinela de Síndrome Gripal (SG).

Recentemente (2020), a vigilância da COVID-19, a infecção humana causada pelo novo Coronavírus, que vem causando uma pandemia, foi incorporada na rede de vigilância da Influenza e outros vírus respiratórios.

Esta página tem como finalidade disponibilizar o legado dos banco de dados (BD) epidemiológicos de SRAG, da rede de vigilância da Influenza e outros vírus respiratórios, desde o início da sua implantação (2009) até os dias atuais (2020), com a incorporação da vigilância da COVID-19.

Para mais informações, acessar:

Gripe/Influenza - <https://saude.gov.br/saude-de-a-z/gripe>

COVID-19 - <https://coronavirus.saude.gov.br/>

Nota

Este documento é ainda tão somente experimentação e análise exploratória em fase absolutamente incipiente. Por isso a falta de estrutura, gráficos não explicativos, *typos*, etc

```
dados <- read_csv("x_to_be_factors.csv",
  col_types = cols(
    FEBRE = col_factor(levels = c("1", "2", "9")),
    TOSSE = col_factor(levels = c("1", "2", "9")),
    GARGANTA = col_factor(levels = c("1", "2", "9")),
    DISPNEIA = col_factor(levels = c("1", "2", "9")),
    DESC_RESP = col_factor(levels = c("1", "2", "9")),
    SATURACAO = col_factor(levels = c("1", "2", "9")),
    DIARREIA = col_factor(levels = c("1", "2", "9")),
    VOMITO = col_factor(levels = c("1", "2", "9")),
```

```

OUTRO_SIN = col_factor(levels = c("1", "2", "9")),
HOSPITAL = col_factor(levels = c("1", "2", "9")),
EVOLUCAO = col_factor(levels = c("1", "2", "3", "9")),
RENAL = col_factor(levels = c("1", "2", "9")),
DIABETES = col_factor(levels = c("1", "2", "9")),
OBESIDADE = col_factor(levels = c("1", "2", "9")),
CLASSI_OUT = col_character(),
PERD_OLFT = col_factor(levels = c("1", "2", "9")),
PERD_PALA = col_factor(levels = c("1", "2", "9")),
VACINA = col_factor(levels = c("1", "2", "9")),
CLASSI_FIN = col_factor(levels = c("1", "2", "3", "4", "5"))))

## Warning: Missing column names filled in: 'X1' [1]

#View(dados)

```

Inferência da estrutura da Rede Causal usando o Algoritmo mmpc

```

#reselecionar retira a variavel automatica X1
xx <- select(dados,
  FEBRE,
  TOSSE,
  GARGANTA,
  DISPNEIA,
  DESC_RESP,
  SATURACAO,
  DIARREIA,
  VOMITO,
  OUTRO_SIN,
  HOSPITAL,
  EVOLUCAO,
  RENAL,
  DIABETES,
  OBESIDADE,
  #CLASSI_OUT, algoritmo nao suporta datatype = character
  PERD_OLFT,
  PERD_PALA,
  VACINA,
  CLASSI_FIN)

# dado não pode ser `tibble` nas funções de *bnlearn*
x = as.data.frame(xx)

# mmpc e mmhc não admitem NA
x_no_na <- na.omit(x)

# 1 influenza
# 5 COVID
srag_covid <- x_no_na %>% filter(CLASSI_FIN == 5)
srag_outras <- x_no_na %>% filter(CLASSI_FIN != 5)

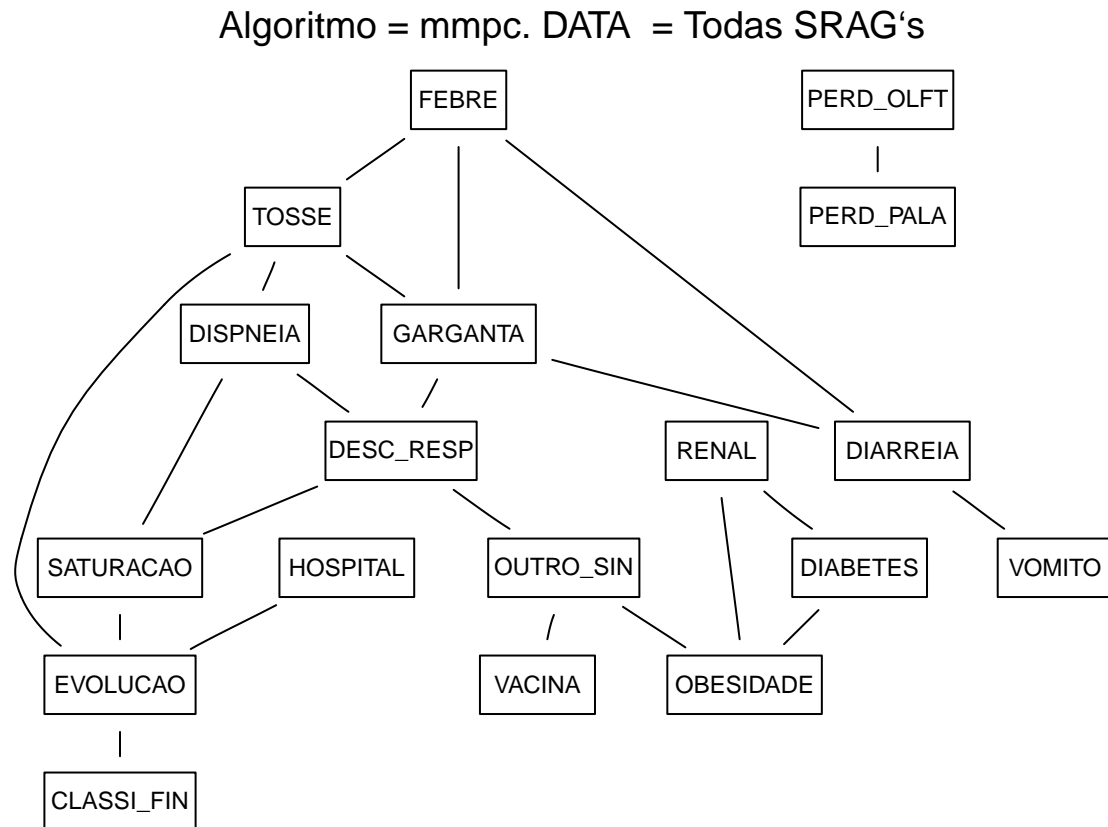
suppressWarnings(bn_mmpc <- mmpc(x_no_na))
suppressWarnings(bn_mmpc_outras <- mmpc(srag_outras))

```

```
suppressWarnings(bn_mmpc_covid <- mmpc(srag_covid))
```

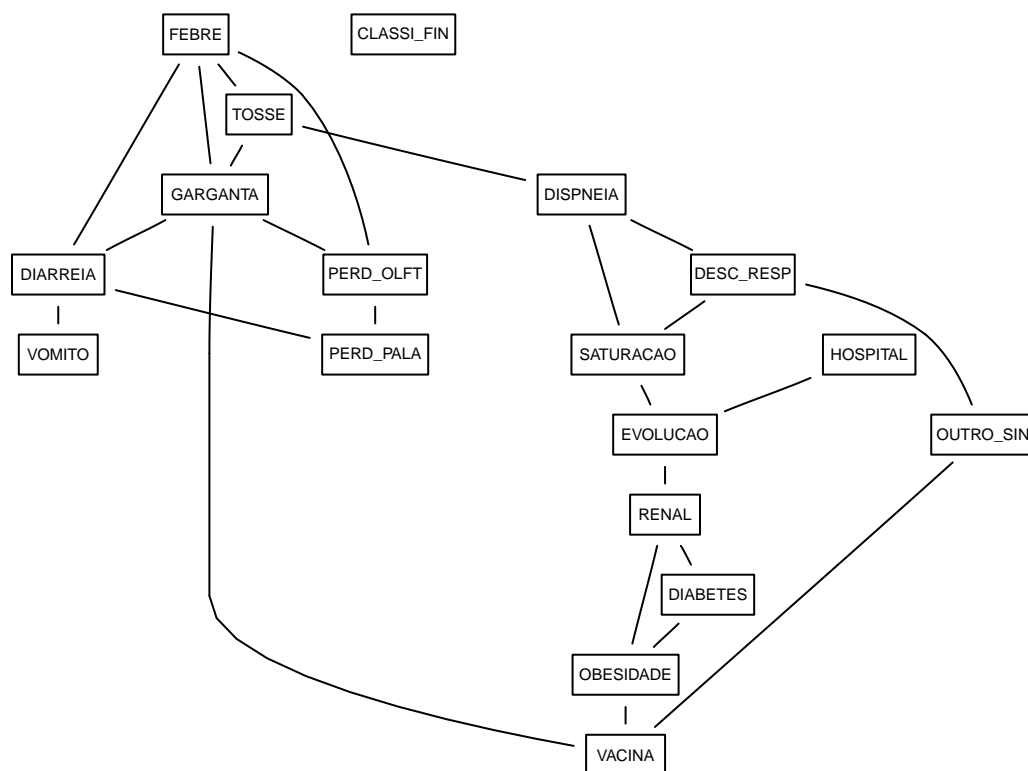
Max-Min Parents and Children (**mmpc**): a forward selection technique for neighbourhood detection based on the maximization of the minimum association measure observed with any subset of the nodes selected in the previous iterations (Tsamardinos et al. 2006). It learns the underlying structure of the Bayesian network (all the arcs are undirected, no attempt is made to detect their orientation).

```
graphviz.plot(bn_mmpc, shape = 'rectangle', main = "Algoritmo = mmpc. DATA = Todas SRAG`s")
```



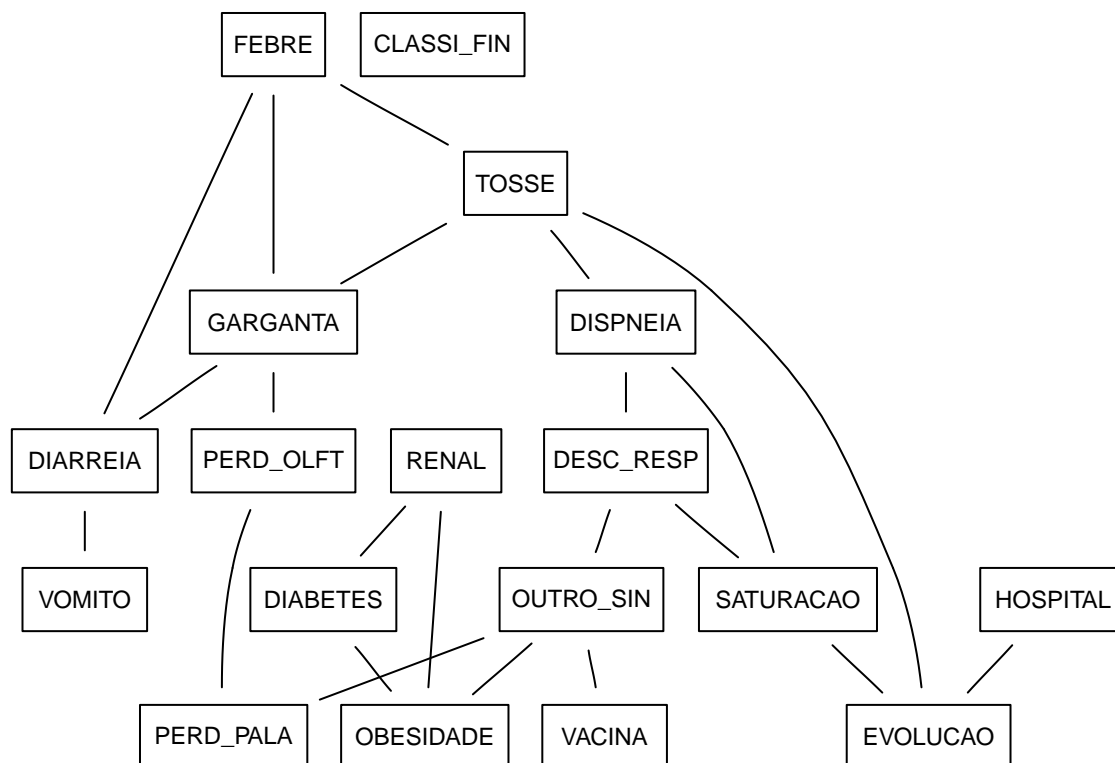
```
graphviz.plot(bn_mmpc_covid, shape = 'rectangle', main = "Algoritmo = mmpc. DATA = COVID")
```

Algoritmo = mmopc. DATA = COVID



```
graphviz.plot(bn_mmopc_outras, shape='rectangle', main = "Algoritmo = mmopc. DATA = Outras")
```

Algoritmo = mmopc. DATA = Outras

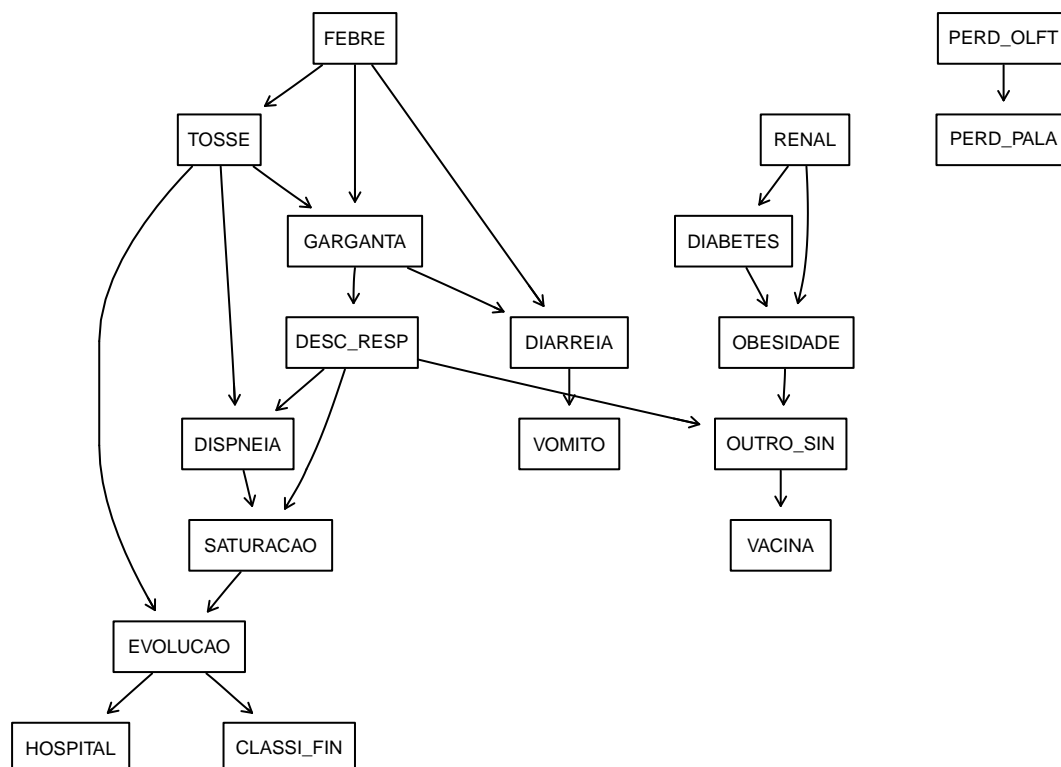


Inferência da estrutura da Rede Causal usando o Algoritmo mmhc

```
suppressWarnings(bn_mmhc <- mmhc(x_no_na))  
suppressWarnings(bn_mmhc_outras <- mmhc(srag_outras))  
suppressWarnings(bn_mmhc_covid <- mmhc(srag_covid))
```

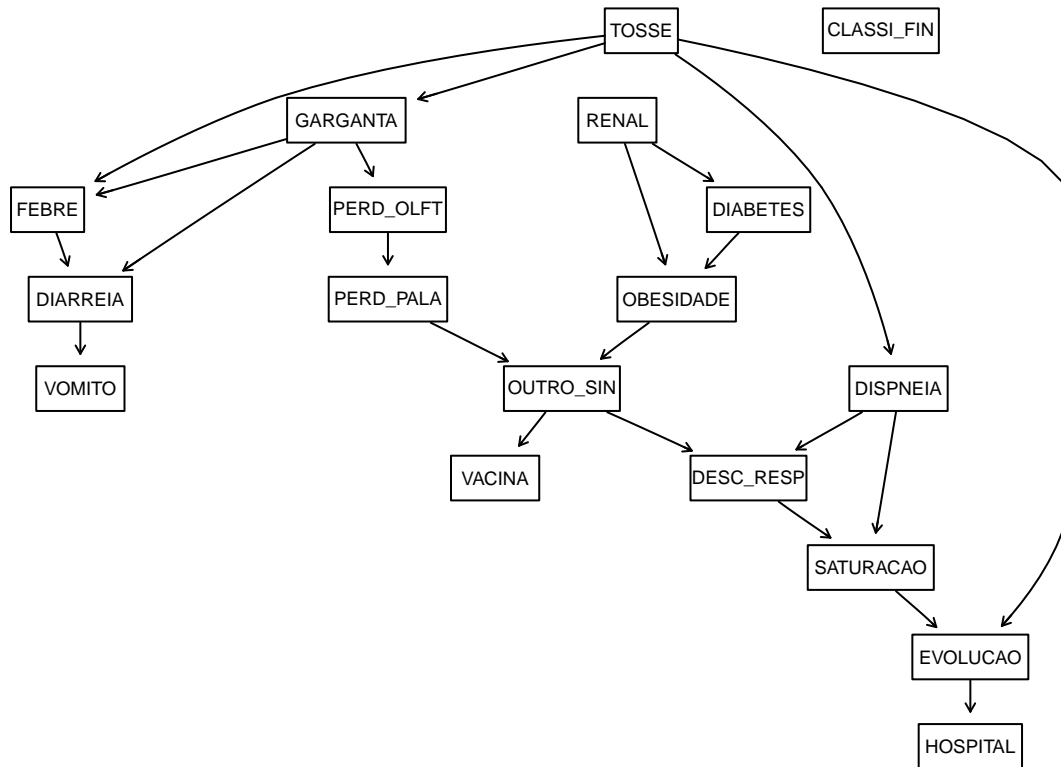
```
graphviz.plot(bn_mmhc, shape = 'rectangle', main = "Data = Todas SRAG's, Algoritmo = mmhc")
```

Data = Todas SRAG's, Algoritmo = mmhc



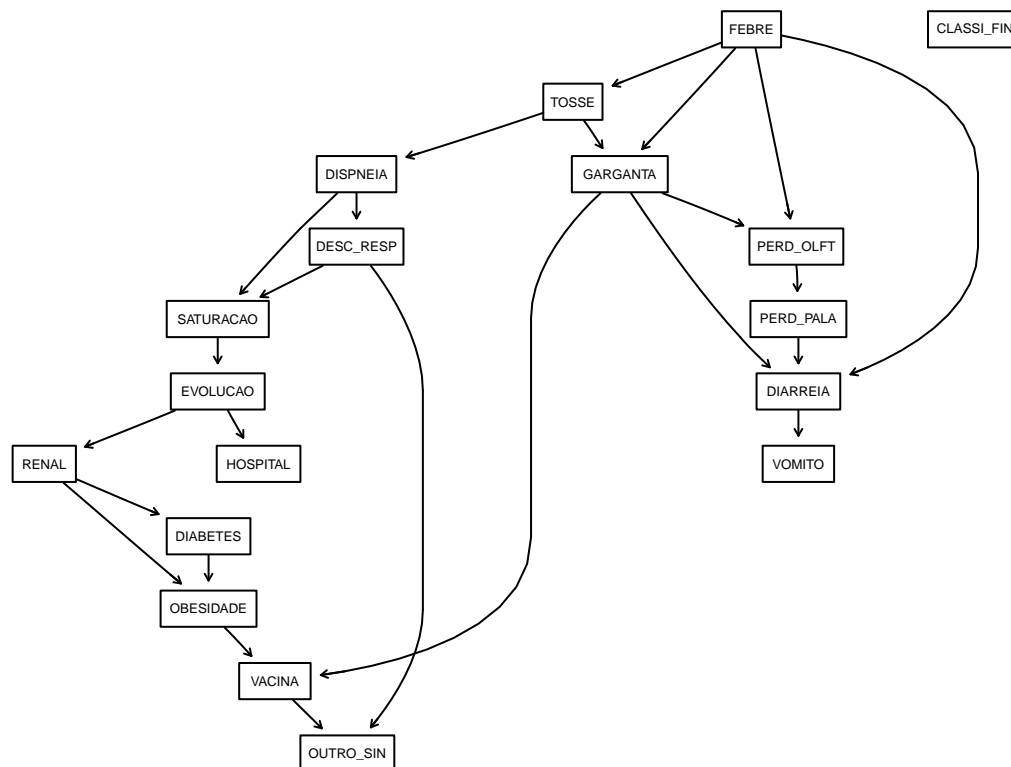
```
graphviz.plot(bn_mmhc_outras, shape = 'rectangle', main = "Data = outras SRAG, Algoritmo = mmhc")
```

Data = outras SRAG, Algoritmo = mmhc



```
graphviz.plot(bn_mmhc_covid, shape = 'rectangle', main = "Data = COVID, Algoritmo = mmhc")
```

Data = COVID, Algoritmo = mmhc

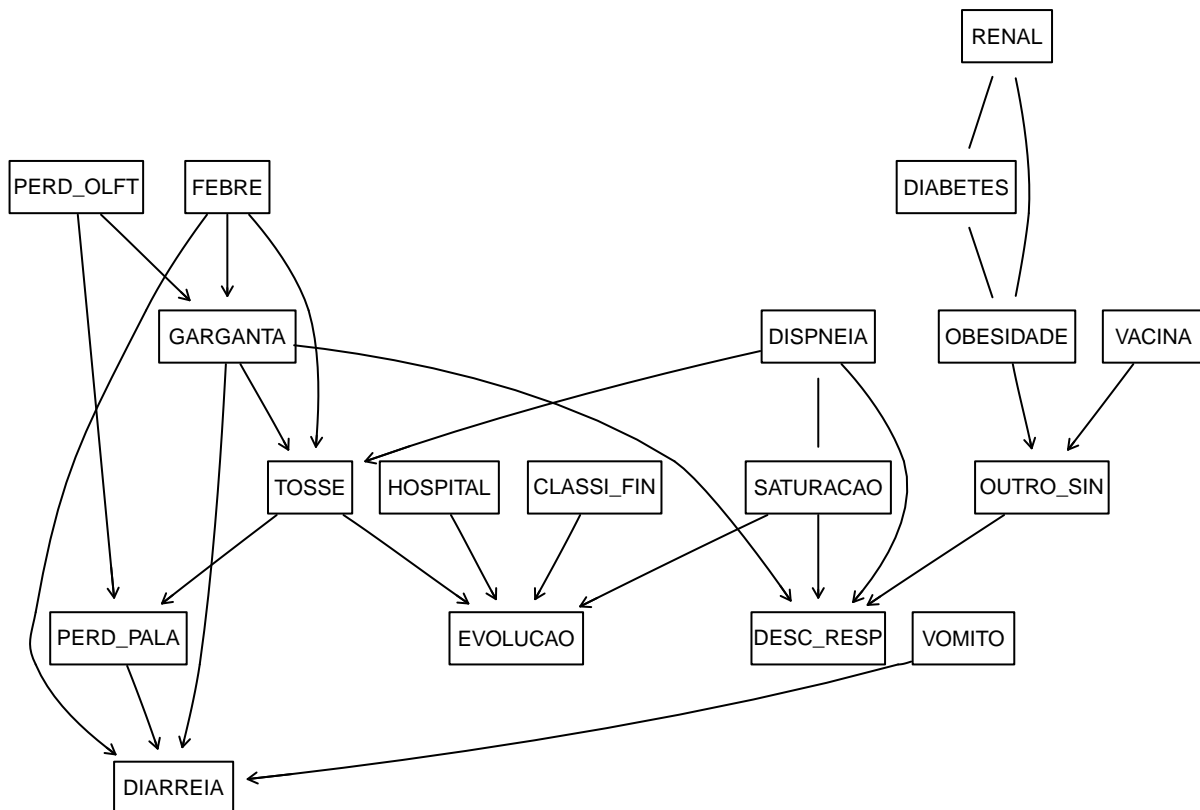


Incremental Association (iamb):

Based on the Incremental Association Markov blanket (IAMB) Algorithm (Tsamardinos et al. 2003), which is based on a two-phase selection scheme (a forward selection followed by an attempt to remove false positives).

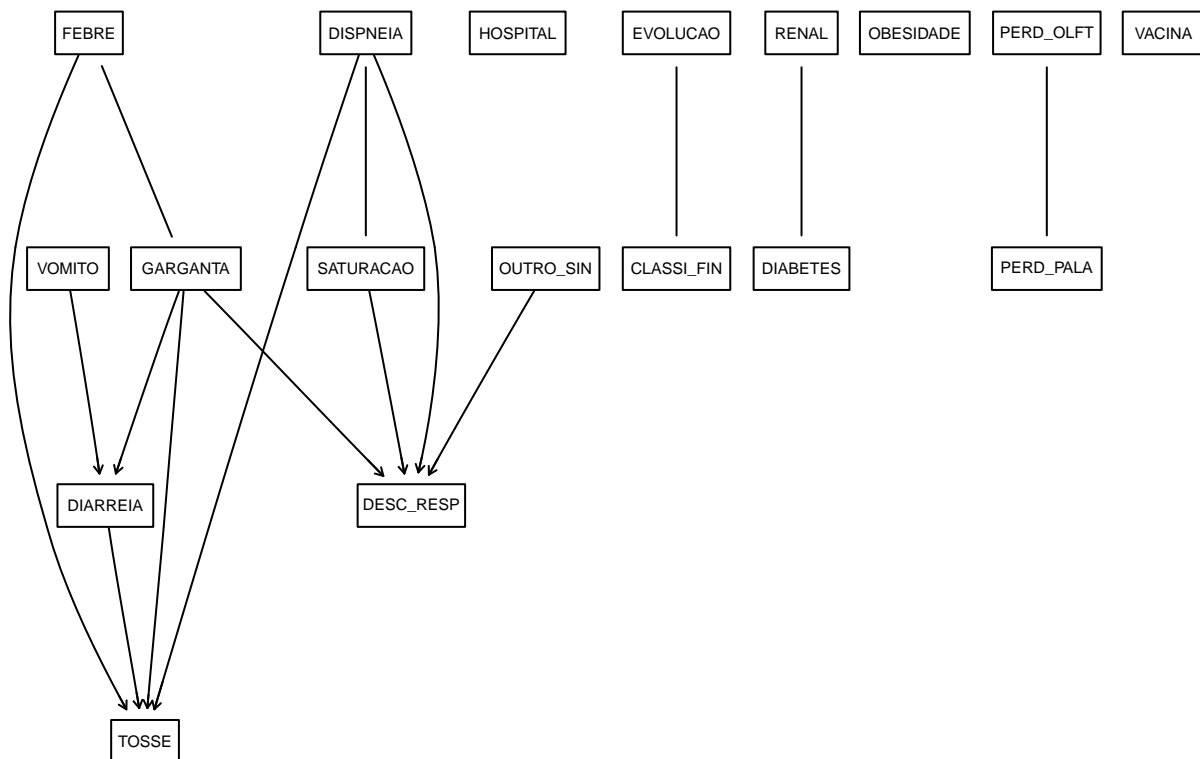
Inferência da estrutura da Rede Causal usando o Algoritmo iamb

```
suppressWarnings(bn_iamb <- iamb(x_no_na))  
graphviz.plot(bn_iamb, shape = 'rectangle')
```



Inferência da estrutura da Rede Causal usando o Algoritmo gs

```
suppressWarnings(bn_gs <- gs(x_no_na))  
graphviz.plot(bn_gs, shape = 'rectangle')
```



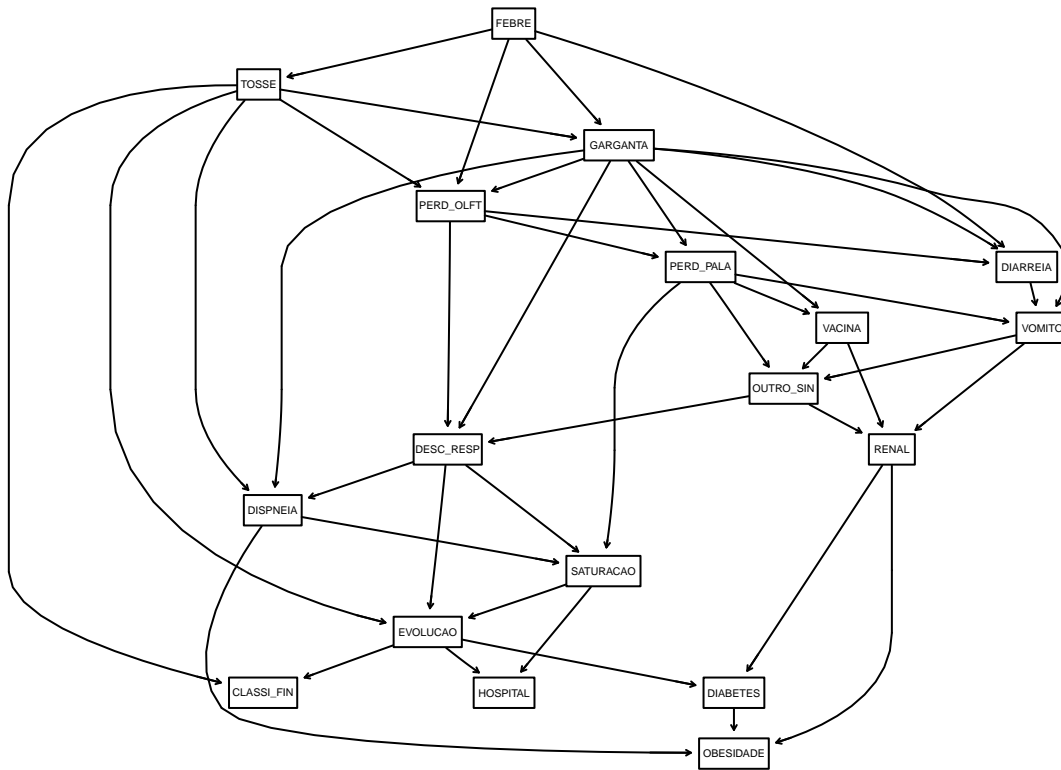
Inferência da estrutura da Rede Causal usando o Algoritmo hc

```

suppressWarnings(bn_hc <- hc(x_no_na))
suppressWarnings(bn_hc_outras <- hc(srag_outras))
suppressWarnings(bn_hc_covid <- hc(srag_covid))
graphviz.plot(bn_hc, shape = 'rectangle', main = "Algoritmo = hc DATA = Todas SRAG`s")

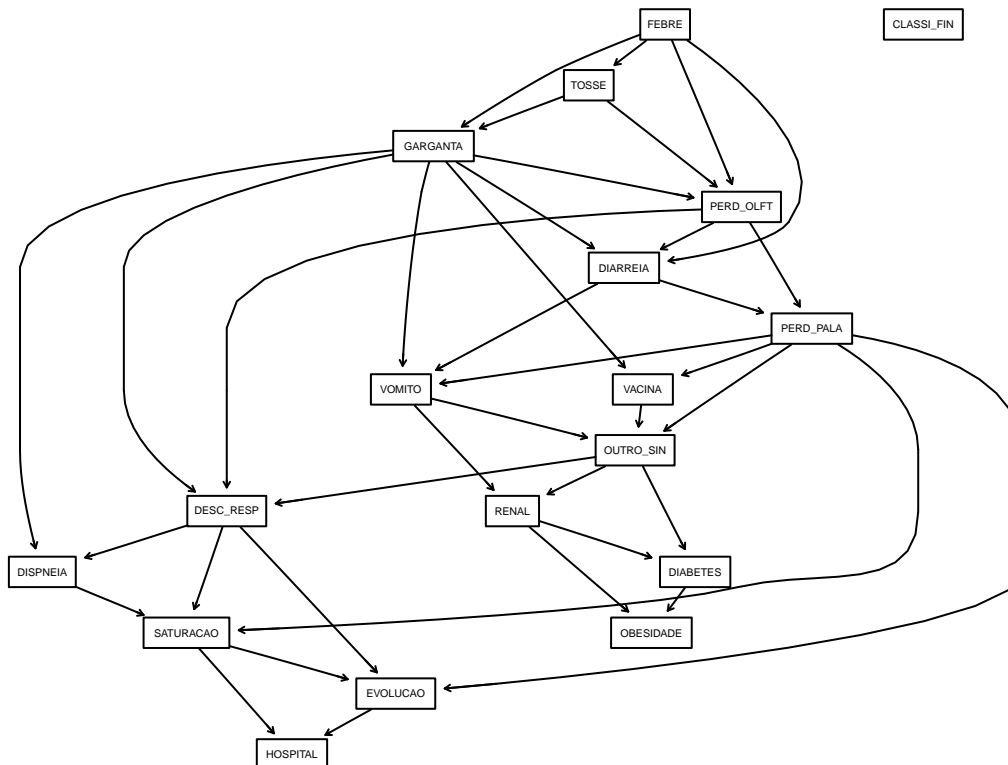
```


Algoritmo = hc DATA = Todas SRAG's



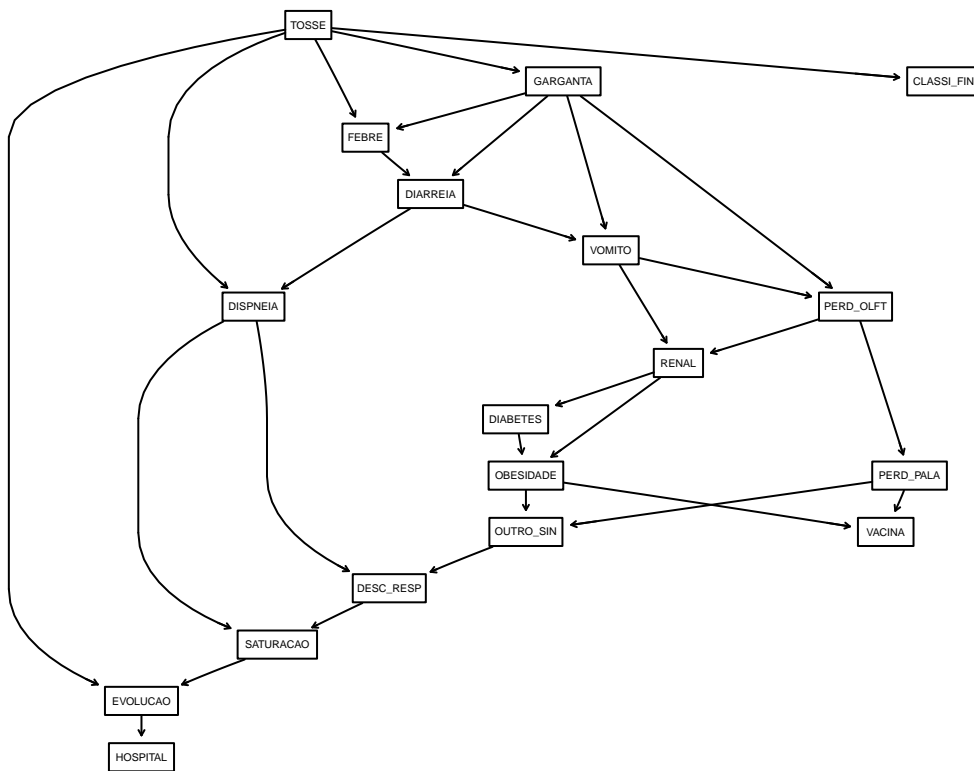
```
graphviz.plot(bn_hc_covid, shape = 'rectangle', main = "Algoritmo = hc DATA = COVID")
```

Algoritmo = hc DATA = COVID



```
graphviz.plot(bn_hc_outras, shape = 'rectangle', main = "Algoritmo = hc DATA = OUTRAS")
```

Algoritmo = hc DATA = OUTRAS



Ajustamento da estrutura do Algoritmo *HC* com os dados

```
fitted_hc = bn.fit(bn_hc, x_no_na)
fitted_hc_outras = bn.fit(bn_hc_outras, srag_outras)
```

```
## Warning in check.data(data, allow.missing = TRUE): variable CLASSI_FIN has
## levels that are not observed in the data.
```

```
fitted_hc_covid = bn.fit(bn_hc_covid, srag_covid)
```

```
## Warning in check.data(data, allow.missing = TRUE): variable CLASSI_FIN has
## levels that are not observed in the data.
```

Descrição estatística da rede para a variável EVOLUCAO DATA = Todas SRAG's

```
fitted_hc$EVOLUCAO
```

```
##
## Parameters of node EVOLUCAO (multinomial distribution)
##
## Conditional probability table:
##
## , , DESC_RESP = 1, SATURACAO = 1
```

```

##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.505486368 0.416623797 0.304909561
##      2 0.454291484 0.519658999 0.630490956
##      3 0.012251767 0.033291688 0.033591731
##      9 0.027970380 0.030425516 0.031007752
##
## , , DESC_RESP = 2, SATURACAO = 1
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.624580100 0.545839210 0.500000000
##      2 0.335804471 0.394720935 0.500000000
##      3 0.008803429 0.025992343 0.000000000
##      9 0.030812000 0.033447512 0.000000000
##
## , , DESC_RESP = 9, SATURACAO = 1
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.633333333 0.235294118 0.392857143
##      2 0.358333333 0.764705882 0.559523810
##      3 0.000000000 0.000000000 0.023809524
##      9 0.008333333 0.000000000 0.023809524
##
## , , DESC_RESP = 1, SATURACAO = 2
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.656088854 0.585708708 0.600000000
##      2 0.301864340 0.350253807 0.400000000
##      3 0.011701706 0.033580633 0.000000000
##      9 0.030345101 0.030456853 0.000000000
##
## , , DESC_RESP = 2, SATURACAO = 2
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.677426793 0.667549669 0.500000000
##      2 0.285335485 0.262086093 0.375000000
##      3 0.009684113 0.029801325 0.000000000
##      9 0.027553608 0.040562914 0.125000000
##
## , , DESC_RESP = 9, SATURACAO = 2
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.666666667 0.250000000 0.500000000
##      2 0.333333333 0.000000000 0.000000000
##      3 0.000000000 0.000000000 0.500000000
##      9 0.000000000 0.750000000 0.000000000
##
## , , DESC_RESP = 1, SATURACAO = 9

```

```

##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.529100529 0.395348837 0.310344828
##      2 0.428571429 0.465116279 0.637931034
##      3 0.021164021 0.116279070 0.000000000
##      9 0.021164021 0.023255814 0.051724138
##
## , , DESC_RESP = 2, SATURACAO = 9
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.428571429 0.384615385 0.333333333
##      2 0.457142857 0.500000000 0.666666667
##      3 0.000000000 0.038461538 0.000000000
##      9 0.114285714 0.076923077 0.000000000
##
## , , DESC_RESP = 9, SATURACAO = 9
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.590643275 0.400000000 0.471223022
##      2 0.374269006 0.600000000 0.500000000
##      3 0.005847953 0.000000000 0.007194245
##      9 0.029239766 0.000000000 0.021582734

```

Descrição estatística da rede para a variável EVOLUCAO DATA = COVID

```
fitted_hc_covid$EVOLUCAO
```

```

##
## Parameters of node EVOLUCAO (multinomial distribution)
##
## Conditional probability table:
##
## , , SATURACAO = 1, PERD_PALA = 1
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.5682656827 0.7150259067 0.7142857143
##      2 0.4095940959 0.2580310881 0.2857142857
##      3 0.0009225092 0.0010362694 0.0000000000
##      9 0.0212177122 0.0259067358 0.0000000000
##
## , , SATURACAO = 2, PERD_PALA = 1
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.7461645746 0.7585170341 0.5000000000
##      2 0.2301255230 0.2164328657 0.5000000000
##      3 0.0000000000 0.0000000000 0.0000000000
##      9 0.0237099024 0.0250501002 0.0000000000
##

```

```

## , , SATURACAO = 9, PERD_PALA = 1
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.4500000000 0.2857142857 0.6842105263
##      2 0.5500000000 0.5714285714 0.3157894737
##      3 0.0000000000 0.0000000000 0.0000000000
##      9 0.0000000000 0.1428571429 0.0000000000
##
## , , SATURACAO = 1, PERD_PALA = 2
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.4320983563 0.5560318432 0.3181818182
##      2 0.5364859491 0.4083486426 0.6818181818
##      3 0.0024854189 0.0030618494 0.0000000000
##      9 0.0289302757 0.0325576648 0.0000000000
##
## , , SATURACAO = 2, PERD_PALA = 2
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.5795431575 0.6256320297 0.3333333333
##      2 0.3895290075 0.3406253225 0.3333333333
##      3 0.0022235698 0.0030956558 0.0000000000
##      9 0.0287042652 0.0306469921 0.3333333333
##
## , , SATURACAO = 9, PERD_PALA = 2
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.4375000000 0.3142857143 0.4285714286
##      2 0.5468750000 0.6000000000 0.5714285714
##      3 0.0000000000 0.0000000000 0.0000000000
##      9 0.0156250000 0.0857142857 0.0000000000
##
## , , SATURACAO = 1, PERD_PALA = 9
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.3316582915 0.4077669903 0.5000000000
##      2 0.6281407035 0.5728155340 0.4788732394
##      3 0.0030150754 0.0000000000 0.0000000000
##      9 0.0371859296 0.0194174757 0.0211267606
##
## , , SATURACAO = 2, PERD_PALA = 9
##
##      DESC_RESP
## EVOLUCAO      1      2 9
##      1 0.6410256410 0.5833333333
##      2 0.3333333333 0.4166666667
##      3 0.0256410256 0.0000000000
##      9 0.0000000000 0.0000000000
##

```

```
## , , SATURACAO = 9, PERD_PALA = 9
##
##      DESC_RESP
## EVOLUCAO      1      2      9
##      1 0.3750000000 0.2857142857 0.4410876133
##      2 0.5859375000 0.4285714286 0.5407854985
##      3 0.0000000000 0.0000000000 0.0000000000
##      9 0.0390625000 0.2857142857 0.0181268882
```

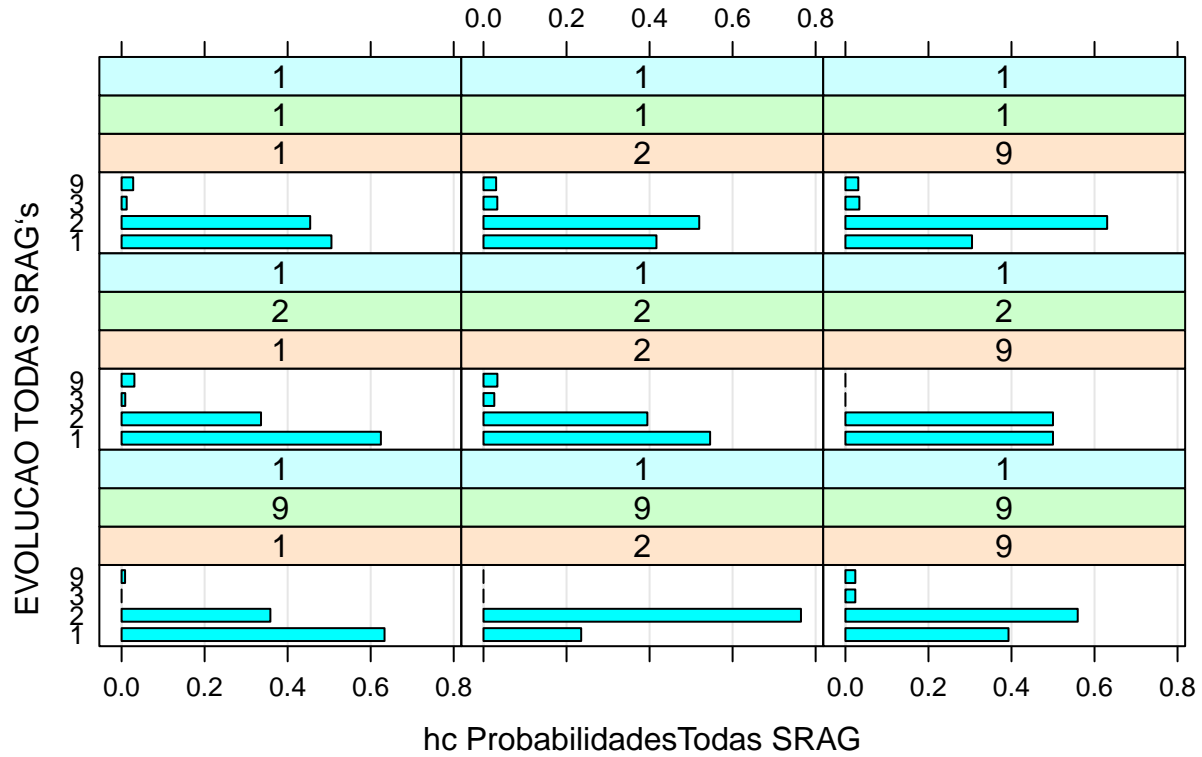
Descrição estatística da rede para a variável EVOLUCAO DATA = OUTRAS

```
fitted_hc_outras$EVOLUCAO
```

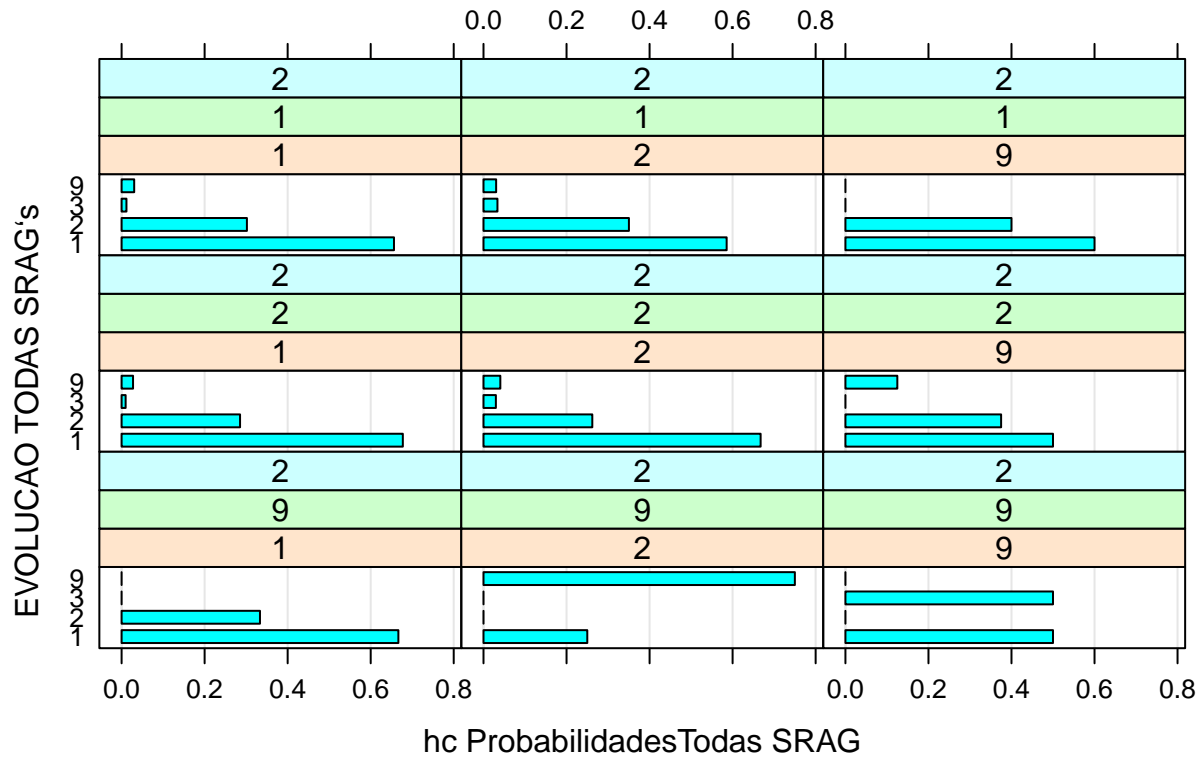
```
##
## Parameters of node EVOLUCAO (multinomial distribution)
##
## Conditional probability table:
##
## , , SATURACAO = 1
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.69641989 0.54903212 0.40566038
##      2 0.21075652 0.30780685 0.41509434
##      3 0.06236838 0.11167837 0.13207547
##      9 0.03045521 0.03148266 0.04716981
##
## , , SATURACAO = 2
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.78845512 0.72133978 0.25000000
##      2 0.12679346 0.15849448 0.25000000
##      3 0.04304304 0.08218232 0.25000000
##      9 0.04170838 0.03798343 0.25000000
##
## , , SATURACAO = 9
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.74725275 0.56000000 0.67088608
##      2 0.16483516 0.20000000 0.25316456
##      3 0.05494505 0.24000000 0.02531646
##      9 0.03296703 0.00000000 0.05063291
```

```
suppressWarnings(bn.fit.barchart(fitted_hc$EVOLUCAO, xlab = "hc ProbabilidadesTodas SRAG", ylab = "EVOLUCAO"))
```

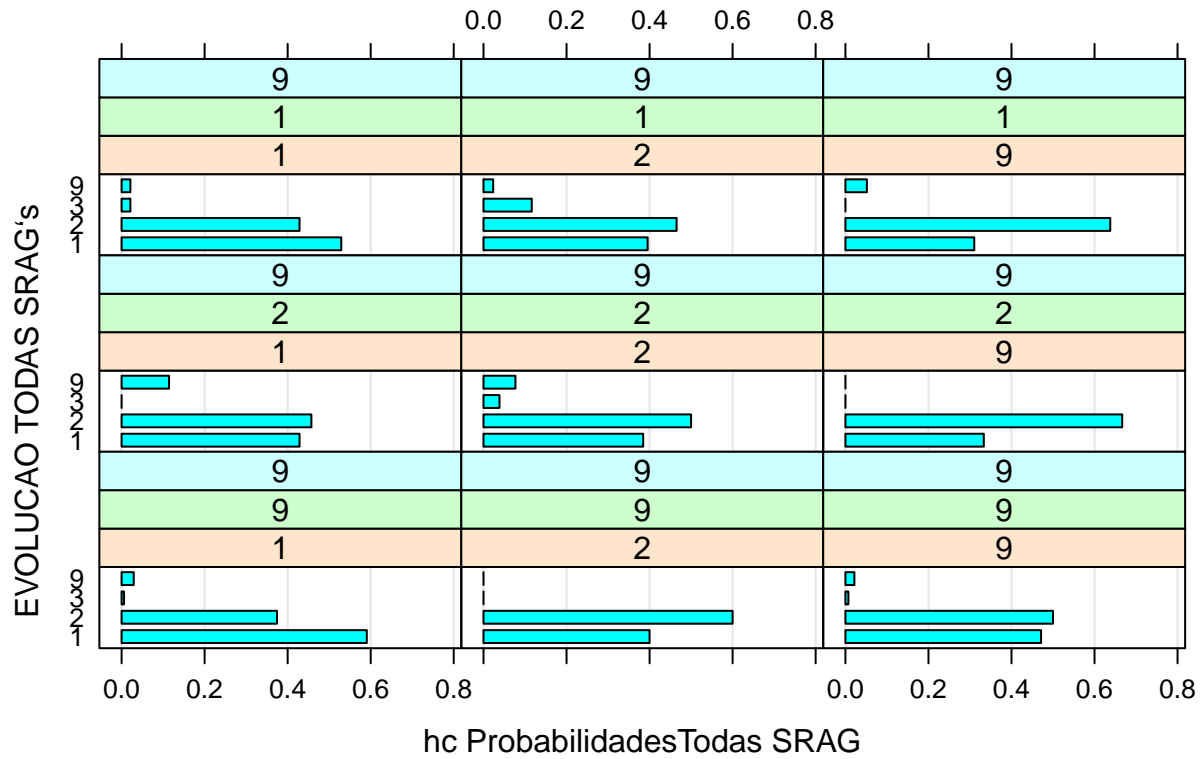
Conditional Probabilities for Node EVOLUCAO



Conditional Probabilities for Node EVOLUCAO

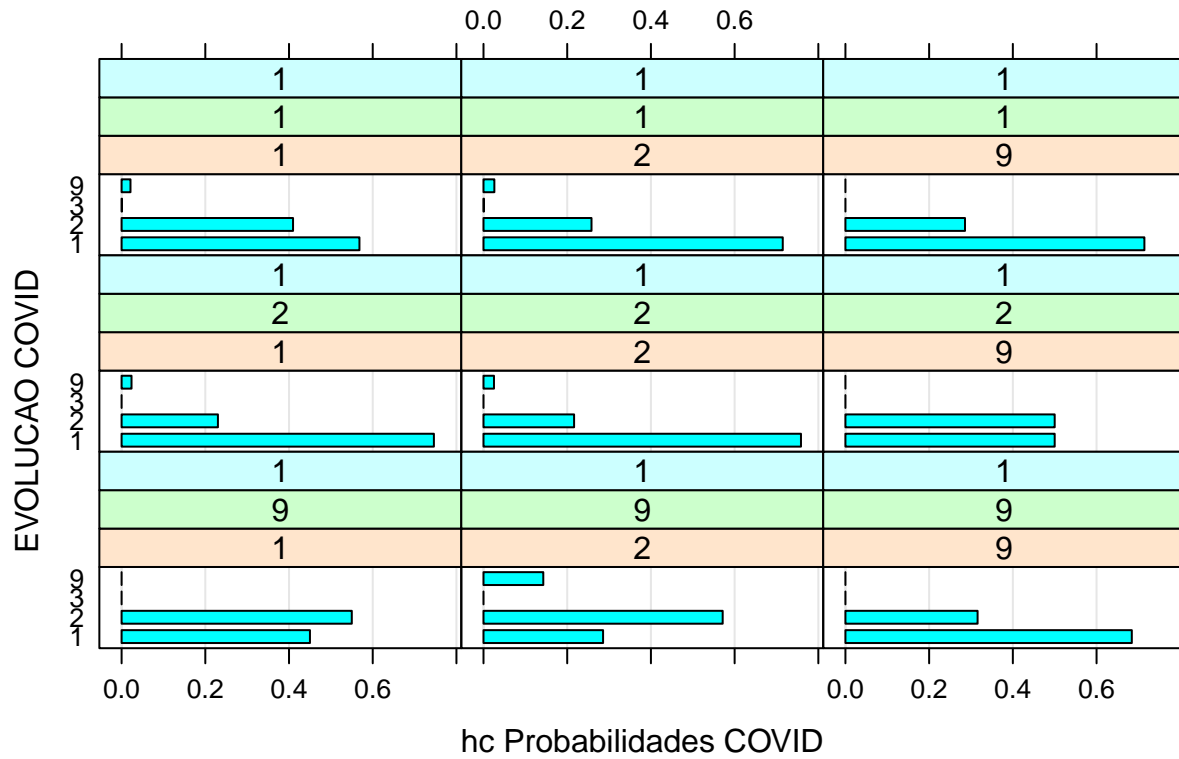


Conditional Probabilities for Node EVOLUCAO

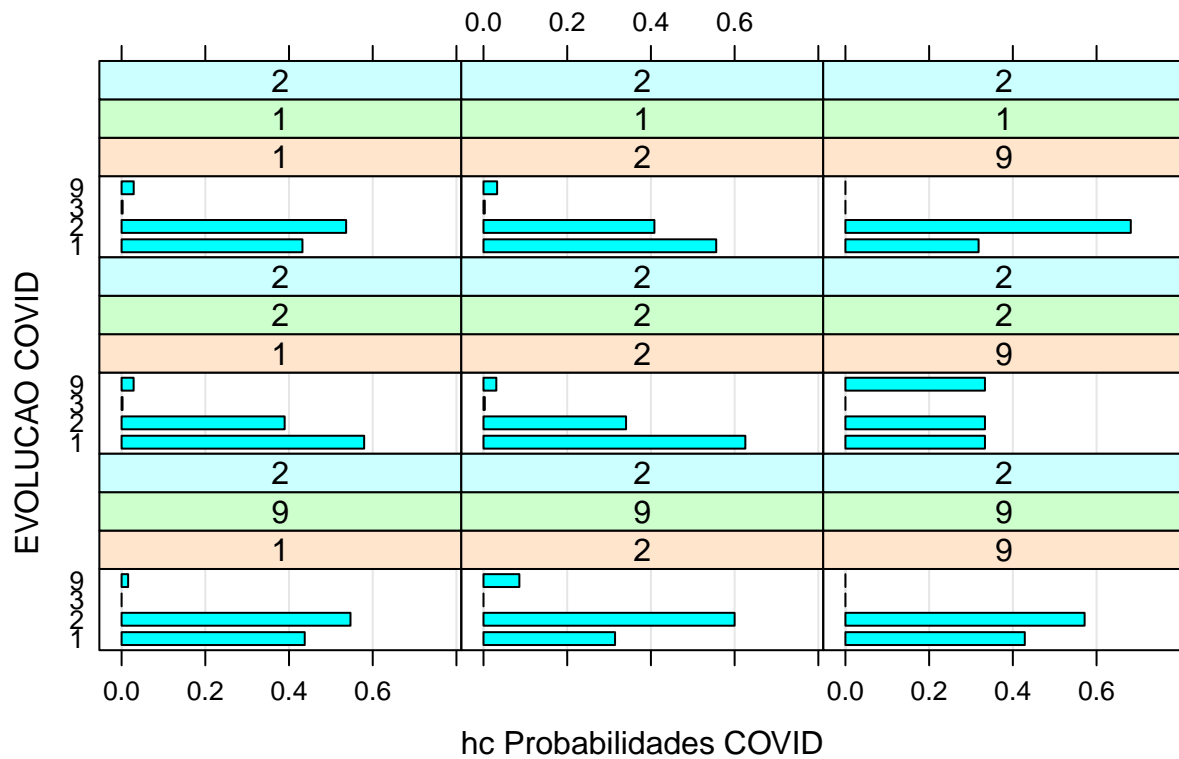


```
suppressWarnings(bn.fit.barchart(fitted_hc_covid$EVOLUCAO, xlab = "hc Probabilidades COVID", ylab = "EVOLUCAO TODAS SRAG's"))
```

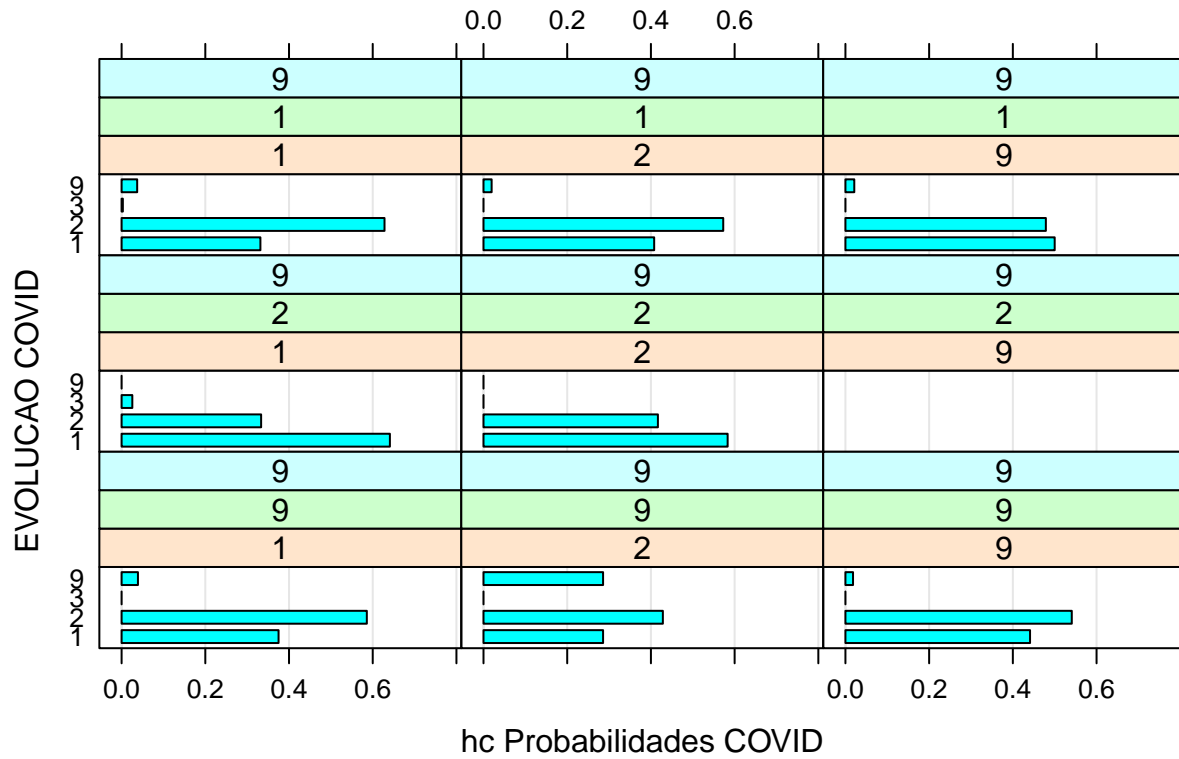

Conditional Probabilities for Node EVOLUCAO



Conditional Probabilities for Node EVOLUCAO

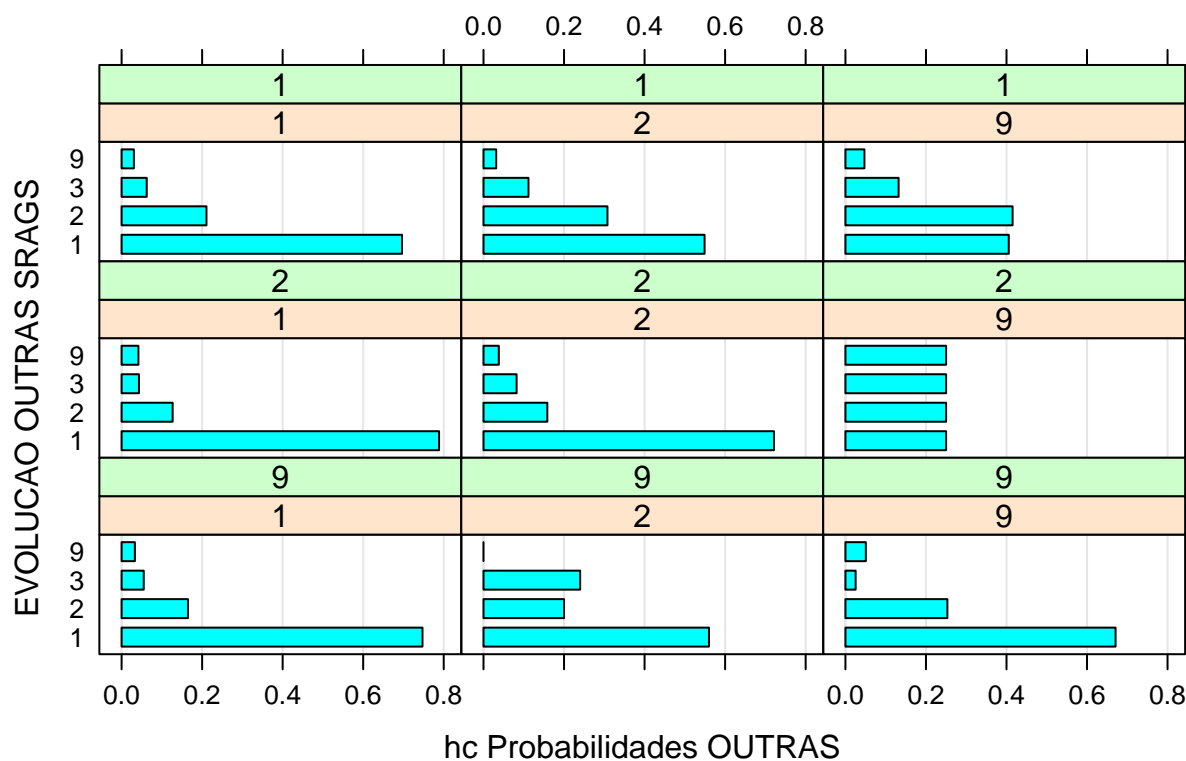


Conditional Probabilities for Node EVOLUCAO



```
suppressWarnings(bn.fit.barchart(fitted_hc_outras$EVOLUCAO, xlab = "hc Probabilidades OUTRAS", ylab = "EVOLUCAO COVID"))
```

Conditional Probabilities for Node EVOLUCAO



```
#graphviz.chart(fit_hc, type = "barchart", grid = TRUE, bar.col = "black",
#               strip.bg = "yellow")
```

Ajustamento da estrutura do Algoritmo *MMHC* com os dados

Descrição estatística da rede para a variável EVOLUCAO Algoritmo = mmhc

```
fitted_mmhc = bn.fit(bn_mmhc, x_no_na)
fitted_mmhc_outras = bn.fit(bn_mmhc_outras, srag_outras)

## Warning in check.data(data, allow.missing = TRUE): variable CLASSI_FIN has
## levels that are not observed in the data.

fitted_mmhc_covid = bn.fit(bn_mmhc_covid, srag_covid)

## Warning in check.data(data, allow.missing = TRUE): variable CLASSI_FIN has
## levels that are not observed in the data.

fitted_mmhc$EVOLUCAO

##
## Parameters of node EVOLUCAO (multinomial distribution)
##
```

```

## Conditional probability table:
##
## , , SATURACAO = 1
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.532615761 0.450960349 0.324324324
##      2 0.427397759 0.486522839 0.615384615
##      3 0.011439565 0.031312207 0.031185031
##      9 0.028546915 0.031204605 0.029106029
##
## , , SATURACAO = 2
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.669581056 0.642998257 0.533333333
##      2 0.291438980 0.288204532 0.333333333
##      3 0.010418944 0.030912260 0.066666667
##      9 0.028561020 0.037884951 0.066666667
##
## , , SATURACAO = 9
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.546835443 0.392405063 0.442477876
##      2 0.407594937 0.493670886 0.525073746
##      3 0.012658228 0.075949367 0.005899705
##      9 0.032911392 0.037974684 0.026548673

```

```
fitted_mmhc_covid$EVOLUCAO
```

```

##
## Parameters of node EVOLUCAO (multinomial distribution)
##
## Conditional probability table:
##
##      SATURACAO
## EVOLUCAO      1      2      9
##      1 0.474769608 0.624878405 0.423948220
##      2 0.493781014 0.343202821 0.546925566
##      3 0.002427445 0.002553502 0.000000000
##      9 0.029021933 0.029365272 0.029126214

```

```
fitted_mmhc_outras$EVOLUCAO
```

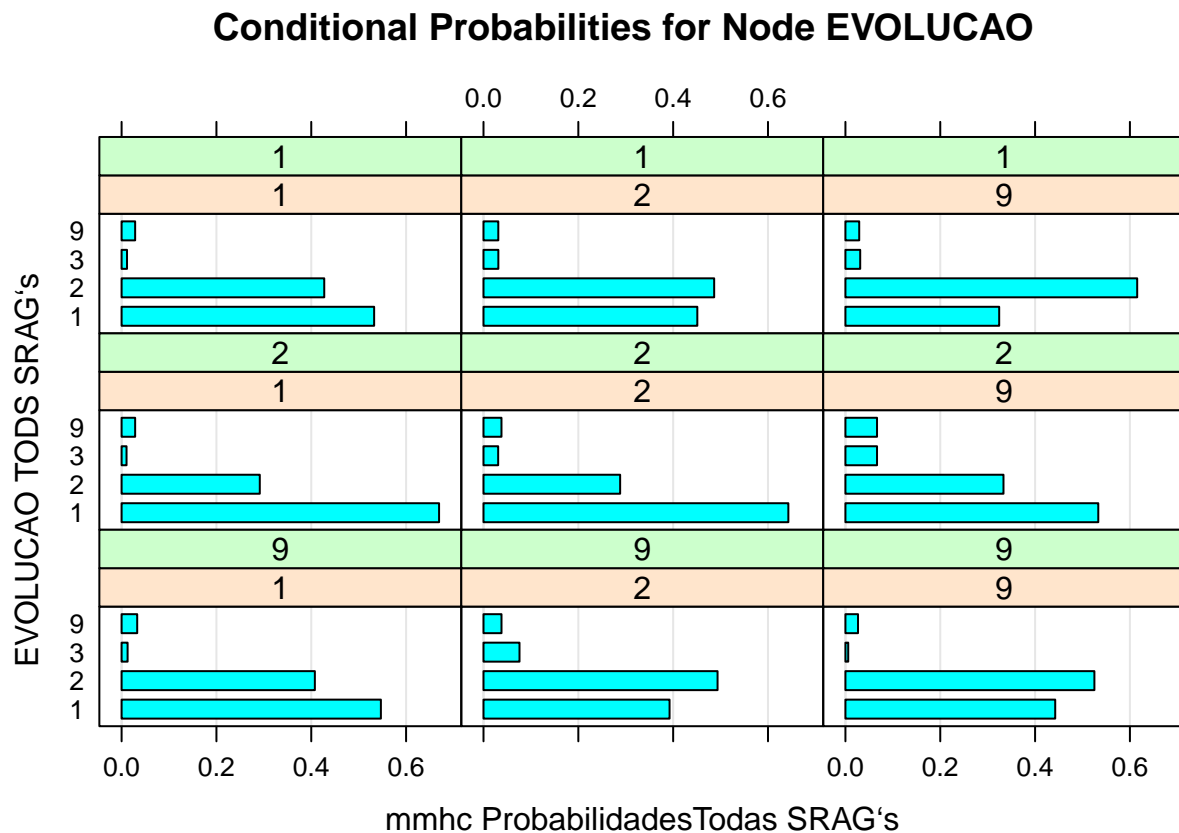
```

##
## Parameters of node EVOLUCAO (multinomial distribution)
##
## Conditional probability table:
##
## , , SATURACAO = 1
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.69641989 0.54903212 0.40566038

```

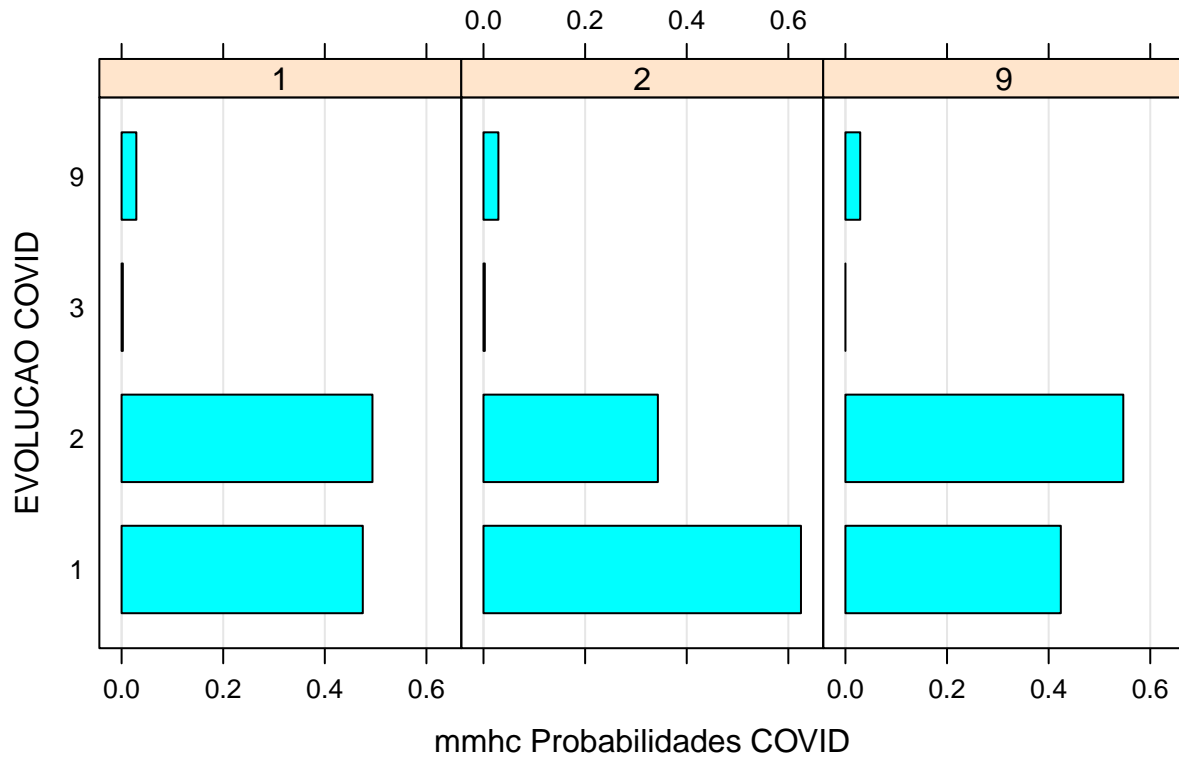
```
##      2 0.21075652 0.30780685 0.41509434
##      3 0.06236838 0.11167837 0.13207547
##      9 0.03045521 0.03148266 0.04716981
##
## , , SATURACAO = 2
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.78845512 0.72133978 0.25000000
##      2 0.12679346 0.15849448 0.25000000
##      3 0.04304304 0.08218232 0.25000000
##      9 0.04170838 0.03798343 0.25000000
##
## , , SATURACAO = 9
##
##      TOSSE
## EVOLUCAO      1      2      9
##      1 0.74725275 0.56000000 0.67088608
##      2 0.16483516 0.20000000 0.25316456
##      3 0.05494505 0.24000000 0.02531646
##      9 0.03296703 0.00000000 0.05063291
```

```
suppressWarnings(bn.fit.barchart(fitted_mmhc$EVOLUCAO, xlab = "mmhc ProbabilidadesTodas SRAG`s", ylab =
```



```
suppressWarnings(bn.fit.barchart(fitted_mmhc_covid$EVOLUCAO, xlab = "mmhc Probabilidades COVID", ylab =
```

Conditional Probabilities for Node EVOLUCAO



```
suppressWarnings(bn.fit.barchart(fitted_mmhc_outras$EVOLUCAO, xlab = "mmhc Probabilidades OUTRAS", ylab = "EVOLUCAO COVID"))
```

Conditional Probabilities for Node EVOLUCAO

