```r
#### R program for the exact Bayes factors

exactPost = function(y,a=1)
{
  f = function(v) # Obtaining the coordinates of the subgraph A
  { f1 = function(x){ y = NULL; for(i in 1:dim(x)[2])
      { y = c(y,rep(0,x[1,i]),rep(1,x[2,i])) }
      y[-c(1,length(y))]+1 }
    f2 = function(i,z) min(cumsum(z==2)[which(cumsum(z==1)==i)])
    m <<- apply(v,1,sum)-1; y = cbind(0:m[1])
    for(i in 1:(dim(v)[1]-1))
    { z = c(0,f1(v[c(i,i+1),])); x = NULL
      for(ii in unique(y[,i]))
      { x = rbind(x,cbind(ii,f2(ii,z):m[i+1])) }
      z = NULL; for(ii in 1:dim(y)[1])
      { u = x[which(x[,1]==y[ii,i]),2]
        z = rbind(z,cbind(t(matrix(y[ii,],i,length(u))),u)) }
      y = z }
    return(split(as.data.frame(y+2,1:dim(y)[1]),
      apply(y,1,sum))) }
  g = function(y) # Counting the number of paths from neighbors
                  # This function runs slow for large values!
  { g1 = function(...) A[...]; g2 = function(...) B[...]
  k = length(y); B = array(1:prod(y[[k]]),y[[k]])
  z = diag(length(y[[k]])); A = array(0,y[[k]])
  A[do.call(g2,as.list(rep(2,length(y[[k]]))))] = 1
  for(i in 2:k){ x = y[[i]]; for(ii in 1:dim(x)[1])
  { j = 0; for(iii in 1:length(y[[k]]))
    { j = j+do.call(g1,x[ii,]-z[iii,]) }
  A[do.call(g2,x[ii,])] = j }}; N <<- do.call(g1,y[[k]]) }
  g(f(y+a))
  # Paths for the unconstrained graph
  M <<- exp(lgamma(sum(m)+1)-sum(lgamma(m+1)))
  return(N/M)
  }

ternaryBF = function(L)
{
  dev.new(width=28/5,height=4*sqrt(3)/2)
  par(mar=rep(0,4),family="serif"); l = L/sum(L)
  plot(c(-1/5,6/5),c(0,sqrt(3)/2),type="n",xlab="",
    ylab="",xaxt="n",yaxt="n",bty="n")
  lines(c(0,1),c(0,0),lwd=3)
  lines(c(0,1/2),c(0,sqrt(3)/2),lwd=3)
  lines(c(1/2,1),c(sqrt(3)/2,0),lwd=3)
  text(3/4,sqrt(3)/2,"independence",cex=3/2)
  text(-1/17,0,"(1)",cex=3/2)
  text(18/17,0,"(2)",cex=3/2)
  lines(c(1/2,1/2),c(0,tan(pi/6)/2),lty=2)
  lines(c(1/4,1/2),c(sin(pi/3)/2,tan(pi/6)/2),lty=2)
  lines(c(3/4,1/2),c(sin(pi/3)/2,tan(pi/6)/2),lty=2)
  x = l[3]+l[1]/2; y = sqrt(3)*l[1]/2
  points(x,y,pch=19,cex=2)
  text(x,y+1/17,expression(italic(L)),cex=3/2,font=8)
  arrows(x,y,l[3]/sum(l[2],l[3]),0,length=1/5,lwd=2)
  arrows(x,y,1/(1+y/sqrt(3)/x),sqrt(3)/(1+sqrt(3)*x/y),
    length=1/5,lwd=2)
  arrows(x,y,1/(1-sqrt(3)*(x-1)/y),sqrt(3)/(1-sqrt(3)*
  (x-1)/y),length=1/5,lwd=2)
  }

### Toy example
toy = matrix(c(1,2,0,2,0,0),2,3,byrow=T)

## Posterior H_s (stochastic dominance)
post1_H_s = exactPost(toy)
## Prior H_s
prior1_H_s = 1/dim(toy)[2]
# Bayes factor L_su
L1_su = post1_H_s/prior1_H_s

# Bayes factor L_0u (independence; fixed row totals)
library(BayesFactor)
re1 = contingencyTableBF(toy,sampleType="indepMulti",
      fixedMargin="rows")
L1_0u = exp(-re1@bayesFactor$bf)

## Posterior H_r (likelihood ratio ordering)
post1_H_r = exactPost(t(toy))
## Prior H_r
prior1_H_r = 1/factorial(dim(toy)[2])
# Bayes factor L_ru
L1_ru = post1_H_r/prior1_H_r
## Ternary plot
ternaryBF(c(L1_0u,L1_su,L1_ru))

### Nuns example
nun = matrix(c(2,1,4,1,2,1,4,2,0,6,0,8),2,6)

# Stochastic dominance
post2_H_s = exactPost(nun)
prior2_H_s = 1/dim(nun)[2]
```

```
L2_su = post2_H_s/prior2_H_s
# Independence
re2 = contingencyTableBF(nun,sampleType="indepMulti",
      fixedMargin="rows")
L2_0u = exp(-re2@bayesFactor$bf)
# Likelihood ratio ordering (takes a few minutes)
post2_H_r = exactPost(t(nun))
prior2_H_r = 1/factorial(dim(nun)[2])
L2_ru = post2_H_r/prior2_H_r

ternaryBF(c(L2_0u,L2_su,L2_ru))

### Tax example
tax = matrix(c(10,5,3,6,10,8,7,9,8,12,11,12,8,10,8,6,
      5,11),3,6)
# Stochastic dominance (takes a few minutes)
post3_H_s = exactPost(tax)
J = dim(tax)[2]
prior3_H_s = 2/(J^2)/(J+1)
L3_su = post3_H_s/prior3_H_s
# Independence
re3 = contingencyTableBF(tax,sampleType="indepMulti",
      fixedMargin="rows")
L3_0u = exp(-re3@bayesFactor$bf)

L3_s0 = L3_su/L3_0u; L3_s0
```