

```

varimax4 <- function(T4,varA=TRUE,varB=TRUE,varC=TRUE,varD=TRUE){
  # INPUT
  # T4 : output of function tucker {multiway}
  # varA: TRUE if A is varimax rotated (default=TRUE)
  # varB: TRUE if B is varimax rotated (default=TRUE)
  # varC: TRUE if C is varimax rotated (default=TRUE)
  # varD: TRUE if D is varimax rotated (default=TRUE)
  # OUTPUT
  # A : matrix A after (possible) varimax rotation (compensated in GA)
  # B : matrix B after (possible) varimax rotation (compensated in GB)
  # C : matrix C after (possible) varimax rotation (compensated in GC)
  # D : matrix D after (possible) varimax rotation (compensated in GD)
  # G : core G after (possible) varimax rotation of A, B, C, D
  require(rTensor)
  A <- T4$A
  B <- T4$B
  C <- T4$C
  D <- T4$D
  G <- T4$G
  size <- dim(G)
  c1 <- size[1]
  c2 <- size[2]
  c3 <- size[3]
  c4 <- size[4]
  GRt <- as.tensor(G)
  if (varA == TRUE){
    GA <- unfold(GRt,row_idx=1,col_idx=c(2,3,4))@data
    AR <- varimax(A)
    GRt <- fold(solve(AR$rotmat)%*%GA,row_idx=1,col_idx=c(2,3,4),modes=c(c1,c2,c3,c4))
    AR <- AR$loadings[,]
  }
  else{
    AR <- A
  }
  if (varB == TRUE){
    GB <- unfold(GRt,row_idx=2,col_idx=c(3,4,1))@data
    BR <- varimax(B)
    GRt <- fold(solve(BR$rotmat)%*%GB,row_idx=2,col_idx=c(3,4,1),modes=c(c1,c2,c3,c4))
    BR <- BR$loadings[,]
  }
  else{
    BR <- B
  }
  if (varC == TRUE){
    GC <- unfold(GRt,row_idx=3,col_idx=c(4,1,2))@data
    CR <- varimax(C)
    GRt <- fold(solve(CR$rotmat)%*%GC,row_idx=3,col_idx=c(4,1,2),modes=c(c1,c2,c3,c4))
    CR <- CR$loadings[,]
  }
  else{
    CR <- C
  }
  if (varD == TRUE){
    GD <- unfold(GRt,row_idx=4,col_idx=c(1,2,3))@data
    DR <- varimax(D)
    GRt <- fold(solve(DR$rotmat)%*%GD,row_idx=4,col_idx=c(1,2,3),modes=c(c1,c2,c3,c4))
    DR <- DR$loadings[,]
  }
  else{
    DR <- D
  }
  out <- list()
  out$A <- AR
  out$B <- BR
  out$C <- CR
  out$D <- DR
  out$G <- GRt@data
  return(out)
}

```