

## Species presence/abundance data: Symmetrical and asymmetrical coefficients

Does the value of the coefficient change in the same way when double-zeros and double-X (where X is any value  $> 0$ ) are added to the data table?

**Double-zero symmetrical dissimilarity** coefficients, which do not have upper limits, do not change when double-zeros or double-X (where  $X > 0$ ) are added to the two site vectors that are compared. Examples: Euclidean, Manhattan.

**Double-zero symmetrical binary similarity** coefficients increase in the same way when double-zeros or double-1 are added to the two site vectors that are compared. Example: Simple matching.

**Double-zero asymmetrical dissimilarity** coefficients do not change when double-zeros are added to the two site vectors that are compared, but they decrease when double-X (where X is any value  $> 0$ ) are added. Examples: Hellinger, chord, percentage difference.

**Double-zero asymmetrical binary similarity** coefficients do not change when double-zeros are added to the two site vectors that are compared, but they increase when double-1 are added. Examples: Jaccard, Sørensen, Ochiai.

# Script R – Objective: show, through an example, that

- **symmetrical binary *S* coefficients** increase in the same way when double-zeros or double-10 are added to the two site vectors (*D* coefficients do not change),
- whereas **asymmetrical binary *S* coefficients** do not change when double-zeros are added to the two site vectors that are compared, but they increase when double-10 are added (*D* coefficients do not change when double-zeros are added, but they decrease when double-10 are added).

# Data matrix in document “Some\_measures\_S\_and\_D.pdf”

```
M1 = matrix(c(1,0,0,1,1,0,1,0,0,1,1,1,0,0),2,7,byrow=TRUE)
```

```
M1
```

```
  [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]  1  0  0  1  1  0  1
[2,]  0  0  1  1  1  0  0
```

# **Add a pair of zeros**

```
M2 = matrix(c(1,0,0,1,1,0,1,0,0,0,1,1,1,0,0,0),2,8,byrow=TRUE)
```

```
M2
```

```
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]  1  0  0  1  1  0  1  0
[2,]  0  0  1  1  1  0  0  0
```

# **Add a pair of 10**

```
M3 = matrix(c(1,0,0,1,1,0,1,10,0,0,1,1,1,0,0,10),2,8,byrow=TRUE)
```

```
M3
```

```
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]  1  0  0  1  1  0  1 10
[2,]  0  0  1  1  1  0  0 10
```

```
#
```

```
# Simple matching similarity coefficient
```

```
(res = dist.binary(M1, method=2))
```

```
# 0.6546537
```

```
# S = 1 - 0.6546537^2 = 0.5714285
```

```
(res = dist.binary(M2, method=2))
```

```
# 0.6123724
```

```
# S = 1 - 0.6123724^2 = 0.625
```

```
(res = dist.binary(M3, method=2))
```

```
# 0.6123724
```

```
# S = 1 - 0.6123724^2 = 0.625
```

```
#
```

```
# Jaccard similarity coefficient
```

```
(res = dist.binary(M1, method=1))
```

```
# 0.7745967
```

```
# S = 1 - 0.7745967^2 = 0.4
```

```
(res = dist.binary(M2, method=1))
```

```
# 0.7745967
```

```
# S = 1 - 0.7745967^2 = 0.4
```

```
(res = dist.binary(M3, method=1))
```

```
# 0.7071068
```

```
# S = 1 - 0.7071068^2 = 0.5
```

```
#  
# Euclidean distance  
(res = dist(M1))  
# 1.732051  
(res = dist(M2))  
# 1.732051  
(res = dist(M3))  
# 1.732051  
  
#  
# Chord distance  
(res = dist(decostand(M1, "norm")))  
# 0.9194017  
(res = dist(decostand(M2, "norm")))  
# 0.9194017  
(res = dist(decostand(M3, "norm")))  
# 0.1701837  
  
#  
# Percentage difference  
(res = vegdist(M1, "bray"))  
# 0.4285714  
(res = vegdist(M2, "bray"))  
# 0.4285714  
(res = vegdist(M3, "bray"))  
# 0.1111111  
  
#  
# Species profile distance  
(res = dist(decostand(M1, "total")))  
# 0.5  
(res = dist(decostand(M2, "total")))  
# 0.5  
(res = dist(decostand(M3, "total")))  
# 0.1385662  
  
#  
# Modified mean character difference (load function beta.div.R)  
(res = beta.div(M1, method="modmeanchardiff")$D)  
# 0.6  
(res = beta.div(M2, method="modmeanchardiff")$D)  
# 0.6  
(res = beta.div(M3, method="modmeanchardiff")$D)  
# 0.5
```

---