*Article*

# Structure comparison and evenness test of biological communities: Several platform-independent computational tools

**WenJun Zhang**
International Academy of Ecology and Environmental Sciences, Hong Kong, China; School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China
E-mail: wjzhang@iaees.org, zhwj@mail.sysu.edu.cn

*Received 16 November 2023*; *Accepted 22 December 2023*; *Published online 11 January 2024*; *Published 1 June 2024*


**Abstract**
The comparison of structural difference between biological communities and the evenness or diversity test of biological communities are always the focus in community studies. In present study, two computational tools were developed to conduct non-parametric statistical tests of between-community structural difference, among which a tool is for multi-samples comparison and the other is for single-sample comparison. Various distance measures are available for user's choice. Two computational tools were developed to test community evenness and one tool was based on the randomization test proposed in present study and the other was based on Ewens-Caswell neutrality model. The computational tools include both online and offline versions and can be used on various computing devices (PCs, iPads, smartphones, etc.), operating systems (Windows, Mac, Android, Harmony, etc.) and web browsers (Chrome, Firefox, etc). They can be used in the sciences as ecology and biology, etc.

**Keywords** community structure; species composition; community evenness; Ewens-Caswell neutrality model; non-parametric test.

**Computational Ecology and Software**
**ISSN 2220-721X**
**URL: http://www.iaees.org/publications/journals/ces/online-version.asp**
**RSS: http://www.iaees.org/publications/journals/ces/rss.xml**
**E-mail: ces@iaees.org**
**Editor-in-Chief: WenJun Zhang**
**Publisher: International Academy of Ecology and Environmental Sciences**

## 1 Introduction

The structure of a biological community refers to its species composition, relative abundance, etc. Community structure is determined by environment conditions, and interspecific and intraspecific interactions in the community (Damgaard, 2011; Zhang, 2011b, 2016, 2018). The comparison of structural difference between biological communities is a focus in community studies (Clarke, 1993; Schoenly and Zhang, 1999). Community evenness refers to the relative abundance of every species in a biological community. A community with equal individuals' number for each species is the most even and thus biodiversity of this community is maximal (Krebs, 1999; Zhang, 2007a). In present study, four platform-dependent computational tools were developed to conduct community comparisons and evenness tests.

## 2 Methodology

### 2.1 Between-community structure comparison

Randomization techniques have been widely used in non-parametric statistic tests (Solow, 1993; Manly, 1997; Zhang, 2007a-b, 2010, 2021a-c). The comparison of statistic difference in community structure and species composition in present computational tool is conducted using a randomization test for re-sampling (Clark, 1993; Krebs, 1999; Schoenly and Zhang, 1999; Zhang, 2011b). In addition to Euclidean distance and Manhattan distance, Pearson correlation, point correlation, quadratic correlation, and Jaccard coefficient based distance measures (Zhang, 2007a, 2015, 2016, 2018) are used in present study. Correlation based distance measures (including Jaccard coefficient) represent the trend similarity of abundance across species, while distance measures (including Euclidean distance and Manhattan distance) mainly represent the total difference in total species. See Zhang (2010, 2011b, 2018) for details of the randomization test. As an improvement, in the calculation of $w$, $r \geq r_0$ is revised as $|r| \geq |r_0|$ in present study.

Suppose $m$ samples were taken in the community A and $n$ samples were taken in the community B, and there are in total of $s$ species in both communities. The same species in two communities share the same species IDs. In the data area of the computational tool, the first column is species IDs, the followed $m$ columns are number of individuals of species for the community A, and the last $n$ columns are number of individuals of species for the community B.

The Javascript codes in this computational tool, CommStructComp, were based on my earlier Java software (Zhang, 2011b).

### 2.2 Between-community comparison of species composition

The comparison of statistic difference in species composition in present computational tool is conducted using a randomization test for re-sampling (Solow, 1993; Zhang, 2011a). See Zhang (2010, 2011a, 2018) for details of the randomization test. In addition to Euclidean distance, Manhattan distance, Chebyshov distance and Pearson correlation based distance (the four measures are for interval values, among which correlation based distance measure represents the trend similarity of abundance across species, while distance measures mainly represent the total difference in total species), more distance measures including those for nominal values (Zhang, 2007a, 2015, 2016, 2018, 2021c) are also used in present study.

In the data area of the computational tool, there are three columns, in which the first column is species IDs (the same species in two communities A and B share the same species IDs), the second column is species abundance (number of individuals, plant coverage, etc.) of the community A and the third column is species abundance (number of individuals, plant coverage, etc.) of the community B.

The Javascript codes in this computational tool, CommSpComp, were based on my earlier Java software (Zhang, 2011a).

Traditional parametric statistical methods such as pairwise data comparison and multivariate sample difference testing require data to conform to a normal distribution. However, species - abundance relationships in nature are often severely skewed and often log-normal, or even a negative exponential distribution. Therefore, traditional parametric statistical methods cannot be used in community comparisons. The randomization methods are a type of non-parametric statistical methods, which are especially suitable for community comparisons and difference testing. Thus the two tools above use the randomization test methods.

### 2.3 Ewens-Caswell Test of community evenness

2.3.1 Randomozation test

A randomization test is proposed here to test the evenness of a biological community. First, choose a diversity index (Simpson Index, Shaanon-Wiener Index, McIntosh Index, Berger-Parker Index, Hurlbert Index; Krebs, 1999; Zhang, 2007b) and calculate the observed evenness index. Second, randomly assign the total individuals into all species (in total of *m* species, the number of total individuals is *n*) and calculate the simulated evenness index. Repeat this procedure many times and find the proportion *p* that the observed evenness index is less than the simulated evenness index. If *p* is less than e.g., 0.01, then the community is even and there is not dominant species.

The computational tool of this method is named CommEvenTest-I in present study.

In the data area of the computational tool, there are two columns, in which the first column is species IDs and the second column is species abundance (number of individuals, plant coverage, etc.).

2.3.2 Ewens-Caswell test

The evenness of a biological community is tested by fitting Ewens-Caswell neutrality model, and the details of Ewens-Caswell test can be found in Zhang and Qi (2002) and Zhang and Zheng (2012). If a biological community is tested to be even using Ewens-Caswell test, then no dominant species in the community. The evenness is equivalent to the so-called diversity in many studies that use Shaanon-Wiener or other diversity indices. In a sense, the greater evenness means the more diversity.

The Javascript codes in this computational tool, CommEvenTest-II, were based on our earlier Java software (Zhang and Qi, 2002; Zhang and Zheng, 2012).

In the data area of the computational tool, there are two columns, in which the first column is species IDs and the second column is species abundance (number of individuals, plant coverage, etc.).

## 3 Computational Tools

In present study, I developed the computational tools, CommStructComp, CommSpComp, CommEvenTest-I, and CommEvenTest-II for structure comparison and evenness test of biological communities. They include both online (http://www.iaees.org/publications/journals/ces/articles/2024-14(2)/CommAnaly.htm) and offline versions, and can be used for various computing devices (PCs, iPads, smartphones, etc.), operating systems (Windows, Mac, Android, Harmony, etc.) and web browsers (Chrome, Firefox, Sougo, 360, etc)(Fig. 1). It can be used in the sciences as ecology and biology, etc.

Both user manual guide and offline tool can be found at:

http://www.iaees.org/publications/journals/ces/articles/2024-14(2)/e-suppl/CommAnaly.rar

Double-click the offline tool, it will be opened in the default web browser.

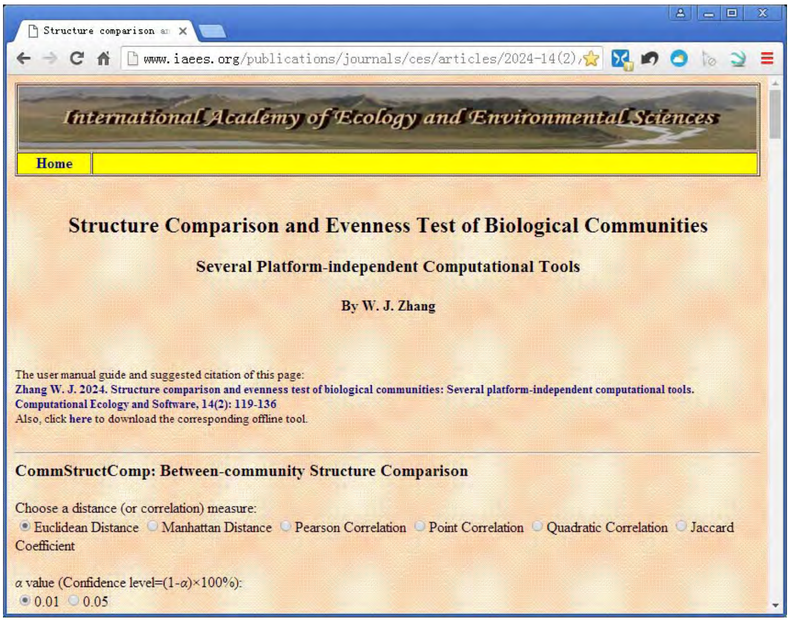


**Fig. 1** The web page of the computational tools.

Some codes for CommStructComp are

```
digits=10000;
var strs="";
var i,j,k,ts=m+n;
var a=new Array(s+1);
for(i=0;i<=s;i++) a[i]=new Array(ts+1+1);
for(i=1;i<=s;i++)
for(j=1;j<=ts+1;j++) {
if (!((i==s) & (j==ts+1))) a[i][j-1]=parseFloat(str.substring(0,str.indexOf(' ')));
else {a[i][j-1]=parseFloat(str);break;}
str=(str.substring(str.indexOf(' '))).trim(); }
ds=new Array(ts+1);
for(i=1;i<=ts;i++) ds[i]=new Array(ts+1);
b=new Array((ts*ts-ts)/2+ts+1);
d=new Array(sim+1);
g=new Array((ts*ts-ts)/2+ts+1);
cols=new Array(ts+1);
w=new Array(ts+1);
v=new Array(ts+1);
h=new Array(ts+1);
for(i=1;i<=ts-1;i++)
for(j=i;j<=ts;j++) {
ds[i][j]=0;
for(k=1;k<=s;k++) {
if (laa[0].checked)
ds[i][j]+=Math.pow(a[k][i]-a[k][j],2);
if (laa[1].checked)
ds[i][j]+=Math.abs(a[k][i]-a[k][j]); }
if (laa[0].checked)
ds[i][j]=Math.sqrt(ds[i][j]/s);
if (laa[1].checked)
ds[i][j]/=s;
ds[j][i]=ds[i][j]; }
var aa,bb,cc,dd,nn1,rr1,xbar,ybar;
for(i=1;i<=ts-1;i++)
for(j=i;j<=ts;j++) {
ds[i][j]=0;
if (laa[2].checked) {
xbar=0;ybar=0;
for(k=1;k<=s;k++) {
xbar+=a[k][i];
ybar+=a[k][j]; }
xbar/=s;
ybar/=s;
```

```
bb=0;cc=0;dd=0;
for(k=1;k<=s;k++) {
dd+=(a[k][i]-xbar)*(a[k][j]-ybar);
bb+=Math.pow(a[k][i]-xbar,2);
cc+=Math.pow(a[k][j]-ybar,2); }
ds[i][j]=1-dd/Math.sqrt(bb*cc); }
if ((laa[3].checked) | (laa[4].checked) | (laa[5].checked)) {
aa=0;bb=0;cc=0;dd=0;nn1=0;rr1=0;
for(k=1;k<=s;k++) {
if (Math.abs(a[k][i])>1e-08) nn1++;
if (Math.abs(a[k][j])>1e-08) rr1++;
if ((Math.abs(a[k][i])<=1e-08) && (Math.abs(a[k][j])<=1e-08)) aa++;
if ((Math.abs(a[k][i])<=1e-08) && (Math.abs(a[k][j])>1e-08)) bb++;
if ((Math.abs(a[k][i])>1e-08) && (Math.abs(a[k][j])<=1e-08)) cc++;
if ((Math.abs(a[k][i])>1e-08) && (Math.abs(a[k][j])>1e-08)) dd++; }
if (laa[3].checked)
ds[i][j]=1-(aa*dd-bb*cc)/Math.sqrt((aa+bb)*(cc+dd)*(aa+cc)*(bb+dd));
if (laa[4].checked)
ds[i][j]=1-Math.sin((aa+dd-(bb+cc))/(aa+bb+cc+dd)*3.1415926/2);
if (laa[5].checked)
ds[i][j]=(cc+bb)/(nn1+rr1-dd); }
ds[j][i]=ds[i][j]; }
var tp,r,r0,hs,ha,zm,om,dev,p;
var en,l,rwn,rbn,rw,rb,em;
var vv,rw1,rb1,sm;
for(j=1;j<=ts;j++) h[j]=j;
tp=m;
en=0;
for(i=1;i<=ts-1;i++)
for(j=i+1;j<=ts;j++) {
en++;
b[en]=ds[h[i]][h[j]]; }
for(j=1;j<=en;j++) g[j]=j;
for(i=1;i<=en-1;i++) {
k=i;
for(j=i;j<=en-1;j++)
if (b[j+1]<=b[k]) k=j+1;
l=g[i];
g[i]=g[k];
g[k]=l;
vv=b[i];
b[i]=b[k];
b[k]=vv; }
rw=0;
rb=0;
```

```
rwn=0;
rbn=0;
for(k=1;k<=en;k++) {
em=0;
for(i=1;i<=ts-1;i++)
for(j=i+1;j<=ts;j++) {
em++;
if (em!=g[k]) continue;
if (((i<=tp) & (j<=tp)) | ((i>tp) & (j>tp))) {
rw+=k;
rwn++; }
if ((i<=tp) & (j>tp)) {
rb+=k;
rbn++; }
} }
if (rwn==0) rw1=0;
if (rwn!=0) rw1=rw/rwn;
if (rbn==0) rb1=0;
if (rbn!=0) rb1=rb/rbn;
r=4*(rb1-rw1)/(ts*(ts-1));
r0=r;
strs+="Observed r="+String(Math.round(r0*digits)/digits)+"\n";
hs=0;
ha=0;
for(sm=1;sm<=sim;sm++) {
zm=0;
om=0;
for(j=1;j<=ts;j++) {
cols[j]=Math.floor(Math.random()+0.5);
if (cols[j]==0) {
zm++;
w[zm]=j; }
if (cols[j]==1) {
om++;
v[om]=j; } }
for(j=1;j<=ts;j++) {
if (j<=zm) h[j]=w[j];
if (j>zm) h[j]=v[j-zm]; }
tp=zm;
en=0;
for(i=1;i<=ts-1;i++)
for(j=i+1;j<=ts;j++) {
en++;
b[en]=ds[h[i]][h[j]]; }
for(j=1;j<=en;j++) g[j]=j;
```

```
for(i=1;i<=en-1;i++) {
k=i;
for(j=i;j<=en-1;j++)
if (b[j+1]<=b[k]) k=j+1;
l=g[i];
g[i]=g[k];
g[k]=l;
vv=b[i];
b[i]=b[k];
b[k]=vv; }
rw=0;
rb=0;
rwn=0;
rbn=0;
for(k=1;k<=en;k++) {
em=0;
for(i=1;i<=ts-1;i++)
for(j=i+1;j<=ts;j++) {
em++;
if (em!=g[k]) continue;
if (((i<=tp) & (j<=tp)) | ((i>tp) & (j>tp))) {
rw+=k;
rwn++; }
if ((i<=tp) & (j>tp)) {
rb+=k;
rbn++; }
} }
if (rwn==0) rw1=0;
if (rwn!=0) rw1=rw/rwn;
if (rbn==0) rb1=0;
if (rbn!=0) rb1=rb/rbn;
r=4*(rb1-rw1)/(ts*(ts-1));
if (Math.abs(r)>=Math.abs(r0)) hs++;
ha+=r;
d[sm]=r; }
ha/=sim;
dev=0;
for(i=1;i<=sim;i++)
dev+=Math.pow(d[i]-ha,2)/(sim-1);
dev=Math.sqrt(dev);
strs+="Expected r="+String(Math.round(ha*digits)/digits)+"\n";
strs+="Standard deviation of expected r="+String(Math.round(dev*digits)/digits)+"\n";
p=(hs+1)/(sim+1);
```

Some codes for CommSpComp are

```
var digits=10000;
var i,j,k,strs="";
var a=new Array(m+1);
for(i=0;i<=m;i++) a[i]=new Array(2+1);
for(i=1;i<=m;i++)
for(j=1;j<=3;j++) {
if (!((i==m) & (j==3))) a[i][j-1]=parseFloat(str.substring(0,str.indexOf(' ')));
else {a[i][j-1]=parseFloat(str);break;}
str=(str.substring(str.indexOf(' '))).trim(); }
var dum,inn;
x=new Array(m+1);
y=new Array(m+1);
dum=1e+10;
for(i=1;i<=m;i++)
for(j=1;j<=2;j++)
if (a[i][j]<=dum) dum=a[i][j];
if (dum<0)
for(i=1;i<=m;i++)
for(j=1;j<=2;j++)
a[i][j]-=dum;
k=-10000;
for(j=1;j<=m;j++)
for(i=1;i<=2;i++) {
inn=1;
dum=a[j][i];
while (m!=0) {
if ((Math.abs(dum-Math.floor(dum))<1) & (!(Math.abs(dum-Math.floor(dum))<=1e-08))) {
inn*=10;
dum*=10;
if (Math.floor(dum+1e-08)!=Math.floor(dum)) break; }
else break; }
if (inn>k) k=inn; }
for(i=1;i<=m;i++)
for(j=1;j<=2;j++)
a[i][j]*=k;
for(i=1;i<=m;i++) {
x[i]=a[i][1];
y[i]=a[i][2]; }
ar=new Array(m+1);
br=new Array(m+1);
ai=new Array(m+1);
aj=new Array(m+1);
p1=new Array(m+1);
```

```
p2=new Array(m+1);
dum=1e+10;
for(i=1;i<=m;i++) {
if (x[i]<=dum) dum=x[i];
if (y[i]<=dum) dum=y[i]; }
if (dum<0) {
for(i=1;i<=m;i++) x[i]-=dum;
for(i=1;i<=m;i++) y[i]-=dum; }
k=-1000000;
for(j=1;j<=2;j++) {
for(i=1;i<=m;i++) {
inn=1;
if (j==1) dum=x[i];
else if (j==2) dum=y[i];
while (m!=0) {
if ((Math.abs(dum-Math.floor(dum))<1) && (!(Math.abs(dum-Math.floor(dum))<=1e-08))) {
inn*=10;
dum*=10;
if (Math.floor(dum+1e-08)!=Math.floor(dum)) break; }
else break; }
if (inn>k) k=inn; } }
for(i=1;i<=m;i++) {
x[i]*=k;
y[i]*=k; }
for(i=1;i<=m;i++) {
ai[i]=x[i];
aj[i]=y[i]; }
var dis=0,aa,bb,cc;
//euclidean
if (sel==1) {
aa=0;
for(k=1;k<=m;k++) aa+=Math.pow(ai[k]-aj[k],2);
dis=Math.sqrt(aa/m); }
//manhattan
if (sel==2) {
aa=0;
for(k=1;k<=m;k++) aa+=Math.abs(ai[k]-aj[k]);
dis=aa/m; }
//chebyshov
if (sel==3) {
dis=-1e+10;
for(k=1;k<=m;k++)
if (Math.abs(ai[k]-aj[k])>=dis) dis=Math.abs(ai[k]-aj[k]); }
//pearsoncorre
if (sel==4) {
```

```
var xbar=0,ybar=0;
aa=0;bb=0;cc=0;
for(k=1;k<=m;k++) {
xbar+=ai[k];
ybar+=aj[k]; }
xbar/=m;
ybar/=m;
for(k=1;k<=m;k++) {
aa+=(ai[k]-xbar)*(aj[k]-ybar);
bb+=Math.pow(ai[k]-xbar,2);
cc+=Math.pow(aj[k]-ybar,2); }
dis=1-aa/Math.sqrt(bb*cc); }
//angularcos
if (sel==5) {
aa=0;bb=0;cc=0;
for(k=1;k<=m;k++) {
aa+=ai[k]*aj[k];
bb+=Math.pow(ai[k],2);
cc+=Math.pow(aj[k],2); }
dis=1-aa/Math.sqrt(bb*cc); }
//multivaluean
if ((sel>=6) & (sel<=9)) {
var pp=1,qq=1,ii,jj,xsquare,sum;
var tempi=new Array(m+1);
var tempj=new Array(m+1);
var temp=new Array(m+1);
for(i=1;i<=m;i++)
temp[i]=new Array(m+1);
tempi[1]=ai[1];
tempj[1]=aj[1];
for(k=1;k<=m;k++) {
jj=0;
for(ii=1;ii<=pp;ii++)
if (Math.abs(ai[k]-tempi[ii])>1e-08) jj++;
if (jj==pp) {
pp++;
tempi[pp]=ai[k]; }
jj=0;
for(ii=1;ii<=qq;ii++)
if (Math.abs(aj[k]-tempj[ii])>1e-08) jj++;
if (jj==qq) {
qq++;
tempj[qq]=aj[k]; } }
for(k=1;k<=pp;k++)
for(jj=1;jj<=qq;jj++) {
```

```
temp[k][jj]=0;
for(ii=1;ii<=m;ii++)
if ((Math.abs(ai[ii]-tempi[kk])<=1e-08) & (Math.abs(aj[ii]-tempj[jj])<=1e-08)) temp[k][jj]++; }
sum=0;
for(k=1;k<=pp;k++) {
tempi[k]=0;
for(jj=1;jj<=qq;jj++) tempi[k]+=temp[k][jj];
sum+=tempi[k]; }
for(k=1;k<=qq;k++) {
tempj[k]=0;
for(jj=1;jj<=pp;jj++) tempj[k]+=temp[jj][k]; }
xsquare=0;
for(k=1;k<=pp;k++)
for(jj=1;jj<=qq;jj++) xsquare+=temp[k][jj]*temp[k][jj]/(tempi[k]*tempj[jj]);
xsquare=sum*(xsquare-1);
if (sel==6) dis=1-Math.sqrt(xsquare/(xsquare+sum));
if (sel==7) dis=1-Math.sqrt(xsquare/(sum*Math.max(pp-1,qq-1)));
if (sel==8) dis=1-Math.sqrt(xsquare/(sum*Math.min(pp-1,qq-1)));
if (sel==9) dis=1-Math.sqrt(xsquare/(sum*Math.sqrt((pp-1)*(qq-1))));
}
var f,fr,sm,c,cs,temp,nx,nxy,dxy,pvalue;
dxy=dis;
nx=0;nxy=0;
for(i=1;i<=m;i++) {
ar[i]=Math.floor(x[i]+y[i]);
nx+=Math.floor(x[i]);
nxy+=ar[i]; }
br[1]=ar[1];
for(i=2;i<=m;i++) br[i]=br[i-1]+ar[i];
w=new Array(nxy+1);
cols=new Array(nxy+1);
fr=0;
for(sm=1;sm<=sim;sm++) {
c=0;
for(i=1;i<=nxy;i++) w[i]=i;
while (nxy!=0) {
cs=Math.floor((nxy-c)*Math.random()+1);
cols[c+1]=w[cs];
if (cs<nxy-c)
for(j=cs+1;j<=nxy-c;j++) w[j-1]=w[j];
c++;
if (c>=nxy) break; }
for(i=1;i<=m;i++)
p1[i]=0;
for(j=1;j<=m;j++) {
```

```
if (ar[j]==0) continue;
if (j==1) temp=0;
else temp=br[j-1];
for(i=1;i<=nx;i++)
if ((cols[i]>temp) & (cols[i]<=br[j])) p1[j]++; }
for(i=1;i<=m;i++) p2[i]=ar[i]-p1[i];
for(i=1;i<=m;i++) {
ai[i]=p1[i];
aj[i]=p2[i]; }
dis=0;
//euclidean
if (sel==1) {
aa=0;
for(k=1;k<=m;k++) aa+=Math.pow(ai[k]-aj[k],2);
dis=Math.sqrt(aa/m); }
//manhattan
if (sel==2) {
aa=0;
for(k=1;k<=m;k++) aa+=Math.abs(ai[k]-aj[k]);
dis=aa/m; }
//chebyshov
if (sel==3) {
dis=-1e+10;
for(k=1;k<=m;k++)
if (Math.abs(ai[k]-aj[k])>=dis) dis=Math.abs(ai[k]-aj[k]); }
//pearsoncorre
if (sel==4) {
var xbar=0,ybar=0;
aa=0;bb=0;cc=0;
for(k=1;k<=m;k++) {
xbar+=ai[k];
ybar+=aj[k]; }
xbar/=m;
ybar/=m;
for(k=1;k<=m;k++) {
aa+=(ai[k]-xbar)*(aj[k]-ybar);
bb+=Math.pow(ai[k]-xbar,2);
cc+=Math.pow(aj[k]-ybar,2); }
dis=1-aa/Math.sqrt(bb*cc); }
//angularcos
if (sel==5) {
aa=0;bb=0;cc=0;
for(k=1;k<=m;k++) {
aa+=ai[k]*aj[k];
bb+=Math.pow(ai[k],2);
```

```
cc+=Math.pow(aj[k],2); }
dis=1-aa/Math.sqrt(bb*cc); }
//multivaluean
if ((sel>=6) & (sel<=9)) {
var pp=1,qq=1,ii,jj,xsquare,sum;
var tempi=new Array(m+1);
var tempj=new Array(m+1);
var temp=new Array(m+1);
for(i=1;i<=m;i++)
temp[i]=new Array(m+1);
tempi[1]=ai[1];
tempj[1]=aj[1];
for(k=1;k<=m;k++) {
jj=0;
for(ii=1;ii<=pp;ii++)
if (Math.abs(ai[k]-tempi[ii])>1e-08) jj++;
if (jj==pp) {
pp++;
tempi[pp]=ai[k]; }
jj=0;
for(ii=1;ii<=qq;ii++)
if (Math.abs(aj[k]-tempj[ii])>1e-08) jj++;
if (jj==qq) {
qq++;
tempj[qq]=aj[k]; } }
for(k=1;k<=pp;k++)
for(jj=1;jj<=qq;jj++) {
temp[k][jj]=0;
for(ii=1;ii<=m;ii++)
if ((Math.abs(ai[ii]-tempi[kk])<=1e-08) & (Math.abs(aj[ii]-tempj[jj])<=1e-08)) temp[k][jj]++; }
sum=0;
for(k=1;k<=pp;k++) {
tempi[k]=0;
for(jj=1;jj<=qq;jj++) tempi[k]+=temp[k][jj];
sum+=tempi[k]; }
for(k=1;k<=qq;k++) {
tempj[k]=0;
for(jj=1;jj<=pp;jj++) tempj[k]+=temp[jj][k]; }
xsquare=0;
for(k=1;k<=pp;k++)
for(jj=1;jj<=qq;jj++) xsquare+=temp[k][jj]*temp[k][jj]/(tempi[k]*tempj[jj]);
xsquare=sum*(xsquare-1);
if (sel==6) dis=1-Math.sqrt(xsquare/(xsquare+sum));
if (sel==7) dis=1-Math.sqrt(xsquare/(sum*Math.max(pp-1,qq-1)));
if (sel==8) dis=1-Math.sqrt(xsquare/(sum*Math.min(pp-1,qq-1)));
```

```
if (sel==9) dis=1-Math.sqrt(xsquare/(sum*Math.sqrt((pp-1)*(qq-1))));
}
dum=dis;
if (Math.abs(dum)>=Math.abs(dxy)) fr++; }
pvalue=fr/sim;
```

Some codes for CommEvenTest-I are

```
var w=new Array(m+1);
for(i=1;i<=m;i++)
w[i]=a[i][1];
var dum,inn;
dum=1e+10;
for(i=1;i<=m;i++)
if (w[i]<=dum) dum=w[i];
if (dum<0)
for(i=1;i<=m;i++)
w[i]-=dum;
k=-10000;
for(j=1;j<=m;j++) {
inn=1;
dum=w[j];
while (m!=0) {
if ((Math.abs(dum-Math.floor(dum))<1) & (!(Math.abs(dum-Math.floor(dum))<=1e-08))) {
inn*=10;
dum*=10;
if (Math.floor(dum+1e-08)!=Math.floor(dum)) break; }
else break; }
if (inn>k) k=inn; }
var sum=0;
for(i=1;i<=m;i++) {
w[i]*=k;
sum+=w[i]; }
var strs="";
digits=10000;
var h=0;
if (sel==1) {
for(i=1;i<=m;i++)
h+=w[i]*(w[i]-1)/(sum*(sum-1));
h=1-h;
h/=1-(sum-m)/(m*(sum-1)); }
if (sel==2) {
for(i=1;i<=m;i++)
h+=w[i]/sum*Math.log(w[i]/sum);
h=-h;
```

```
h/=Math.log(m); }
if (sel==3) {
for(i=1;i<=m;i++)
h+=Math.pow(w[i],2);
h=(sum-Math.sqrt(h))/(sum-Math.sqrt(sum));
h/=(sum-sum/Math.sqrt(m))/(sum-Math.sqrt(sum)); }
if (sel==4) {
var max=-1e+08;
for(i=1;i<=m;i++)
if (w[i]>max) max=w[i];
h=1-max/sum;
h/=1; }
if (sel==5) {
for(i=1;i<=m;i++)
h+=w[i]*(sum-w[i])/(sum*(sum-1));
h/=sum*(1-1/m)/(sum-1); }
var sm,h1,f=0,sum1,sum2=sum-m,p;
for(sm=1;sm<=sim;sm++) {
for(i=1;i<=m;i++)
w[i]=0;
sum1=0;
for(i=1;i<=m;i++) {
w[i]=Math.floor(sum2*Math.random()+1);
sum1+=w[i];
if (sum1>sum2) {
w[i]=sum2-(sum1-w[i]);
break; } }
for(i=1;i<=m;i++)
w[i]+=1;
var h1=0;
if (sel==1) {
for(i=1;i<=m;i++)
h1+=w[i]*(w[i]-1)/(sum*(sum-1));
h1=1-h1;
h1/=1-(sum-m)/(m*(sum-1)); }
if (sel==2) {
for(i=1;i<=m;i++)
h1+=w[i]/sum*Math.log(w[i]/sum);
h1=-h1;
h1/=Math.log(m); }
if (sel==3) {
for(i=1;i<=m;i++)
h1+=Math.pow(w[i],2);
h1=(sum-Math.sqrt(h1))/(sum-Math.sqrt(sum));
h1/=(sum-sum/Math.sqrt(m))/(sum-Math.sqrt(sum)); }
```

```
if (sel==4) {
var max=-1e+08;
for(i=1;i<=m;i++)
if (w[i]>max) max=w[i];
h1=1-max/sum;
h1/=1; }
if (sel==5) {
for(i=1;i<=m;i++)
h1+=w[i]*(sum-w[i])/(sum*(sum-1));
h1/=sum*(1-1/m)/(sum-1); }
if (h<h1) f++;
}
p=f/sim;
```

Some codes for CommEvenTest-II are

```
digits=10000;
var w=new Array(p+2);
var sum=0;
for(i=1;i<=p;i++) {
w[i]=x[i][1];
sum+=w[i]; }
sum=Math.round(sum);
var strs="",k1,nk1,h,wn,eh,varinf,sh,nh,xx,zz,d1,d2,fr,xx,cd;
h=0;
for(i=1;i<=p;i++)
h+=w[i]/sum*Math.log(w[i]/sum);
h=-h;
var w1=new Array(sum+2);
var w2=new Array(sum+2);
k1=p-1;
nk1=sum-k1;
for(i=1;i<=p;i++)
w1[i]=w[i];
w1[1]=1;
w2[1]=1;
for(i=2;i<=nk1;i++)
w2[i]=1.0/i;
for(i=2;i<=k1;i++)
for(j=2;j<=nk1;j++) {
w1[j]=w2[j];
w2[j]=((j+i-2)*w2[j-1]+i*w1[j])/(j+i-1); }
wn=1;
for(i=2;i<=nk1;i++)
wn=((i+k1-1)*wn+p*w2[i])/(i+k1);
```

```
var siga=0,sigb=0,sigc=0;
for(j=1;j<=nk1;j++) {
siga+=Math.log(nk1-j+1)*w2[j];
sigb+=(nk1-j+1)*w2[j]*Math.pow(Math.log(nk1-j+1),2);
w2[j]=Math.log(j); }
siga*=p/wn;
sigb*=p/wn;
eh=Math.log(sum)-siga/sum;
for(i=1;i<=nk1;i++)
for(j=1;j<=nk1-i+1;j++)
sigc+=w2[i]*w2[j]*w1[nk1-i-j+2];
sigc*=p*k1/wn;
varinf=(sigb+sigc-siga*siga)/(sum*sum);
sh=Math.sqrt(varinf);
nh=(h-eh)/sh;
xx=Math.log(p);
zz=eh/xx;
d1=2*zz*(zz*(1-zz)*xx*xx/varinf-1);
d2=d1*(1-zz)/zz;
fr=(Math.log(p)-eh)*h/((Math.log(p)-h)*eh);
xx=ftestp(fr,d1,d2);
cd=(1-xx)*100;
```

**References**


Clarke KR. 1993. Non-parametric multivariate analyses of changes in community structure. Australia Journal of Ecology, 18:117-143

Damgaard C. 2011. Measuring competition in plant communities where it is difficult to distinguish individual plants. Computational Ecology and Software, 1(3): 125-137 http://www.iaees.org/publications/journals/ces/articles/2011-1(3)/1-Damgaard-Abstract.asp

Krebs CJ. 1999. Ecological Methodology (2nd edition). Addison-Wesley Educational Publishers, Inc., USA

Manly BFJ. 1997. Randomization, Bootstrap and Monte Carlo Methods in Biology (2nd Edition). Chapman & Hall, London, UK

Schoenly KG, Zhang WJ. 1999. IRRI Biodiversity Software Series. V. RARE, SPPDISS, and SPPANK: programs for detecting between-sample difference in community structure. IRRI Technical Bulletin No. 5. International Rice Research Institute, Manila, Philippines

Solow AR. 1993. A simple test for change in community structure. Journal of Animal Ecology, 62: 191-193

Zhang WJ. 2007a. Computer inference of network of ecological interactions from sampling data. Environmental Monitoring and Assessment, 124: 253-261. https://link.springer.com/article/10.1007/s10661-006-9223-8

Zhang WJ. 2007b. Methodology on Ecology Research. Sun Yat-sen University, Guangzhou, China

Zhang WJ. 2010. Computational Ecology: Artificial Neural Networks and Their Applications. World Scientific, Singapore. http://www.worldscientific.com/worldscibooks/10.1142/7436

Zhang WJ. 2011a. A Java algorithm for non-parametric statistic comparison of network structure. Network

Biology, 1(2): 130-133.
http://www.iaees.org/publications/journals/nb/articles/2011-1(2)/Java-algorithm-non-parametric-statistic-comparison-network-structure.pdf

Zhang WJ. 2011b. A Java program for non-parametric statistic comparison of community structure. Computational Ecology and Software, 1(3): 183-185.
http://www.iaees.org/publications/journals/ces/articles/2011-1(3)/Java-program-non-parametric-statistic-comparison-community-structure.pdf

Zhang WJ. 2015. Calculation and statistic test of partial correlation of general correlation measures. Selforganizology, 2(4): 65-77.
http://www.iaees.org/publications/journals/selforganizology/articles/2015-2(4)/statistic-test-of-partial-correlation-of-general-correlation-measures.pdf

Zhang WJ. 2016. Selforganizology: The Science of Self-Organization. World Scientific, Singapore.
https://www.worldscientific.com/worldscibooks/10.1142/9685

Zhang WJ. 2018. Fundamentals of Network Biology. World Scientific Europe, London, UK.
https://www.worldscientific.com/worldscibooks/10.1142/q0149

Zhang WJ. 2021a. A statistical simulation method for causality inference of Boolean variables. Network Biology, 11(4): 263-273.
http://www.iaees.org/publications/journals/nb/articles/2021-11(4)/a-method-for-causality-inference-of-Boolean-variables.pdf

Zhang WJ. 2021b. Causality inference of linearly correlated variables: The statistical simulation and regression method. Computational Ecology and Software, 11(4): 154-161.
http://www.iaees.org/publications/journals/ces/articles/2021-11(4)/causality-inference-of-linearly-correlated-variables.pdf

Zhang WJ. 2021c. Causality inference of nominal variables: A statistical simulation method. Computational Ecology and Software, 11(4): 142-153.
http://www.iaees.org/publications/journals/ces/articles/2021-11(4)/causality-inference-of-nominal-variables-with-statistical-simulation-method.pdf

Zhang WJ. 2024a. MetaAnaly: The platform-independent computational tool for meta-analysis in the paradigm of new statistics. Network Biology, 14(2): 187-214.
http://www.iaees.org/publications/journals/nb/articles/2024-14(2)/8-Zhang-Abstract.asp

Zhang WJ. 2024b. SampSizeCal: The platform-independent computational tool for sample sizes in the paradigm of new statistics. Network Biology, 14(2): 100-155.
http://www.iaees.org/publications/journals/nb/articles/2024-14(2)/5-Zhang-Abstract.asp

Zhang WJ, Qi YH. 2002. Statistic test and network software on statistic significance of biological diversity and evenness. Modern Computer, 145: 6-7.

Zhang WJ, Qi YH. 2024. ANOVA-nSTAT: ANOVA methodology and computational tools in the paradigm of new statistics. Computational Ecology and Software, 14(1): 48-67.
http://www.iaees.org/publications/journals/ces/articles/2024-14(1)/4-Zhang-Abstract.asp

Zhang WJ, Zheng H. 2012. A program for statistic test of community evenness. Computational Ecology and Software, 2(1): 80-82.
http://www.iaees.org/publications/journals/ces/articles/2012-2(1)/program-for-statistic-test-of-communityevenness.pdf