

Dietary Parameter Search FORTRAN Code for estimating fish diets based on stable nitrogen & carbon isotope values.

Robert Thacker
Department of Astronomy, Queen's University

Linda Campbell
School of Environmental Studies & Department of Biology, Queen's University

The FORTRAN program used to conduct a parameter search for *Brycinus nurse* diet (Env. Biol. Fish. 74:247-260) based on mean $\delta^{15}\text{N}$ and $\delta^{13}\text{C}$ values of diet and predator. Explanatory notes are listed by the "c". The other programs have the same basic construction although more dietary items necessitate a larger number of loops.

If you use or adapt this program, please cite the following article: Campbell L.M., Wandera S.B., Thacker R.J., Dixon D.G. & Hecky R.E. 2005. Trophic niche segregation in the Nilotic ichthyofauna of Lake Albert (Uganda, Africa). Environmental Biology of Fishes. 74:247-260

```
"program brycinus"
c
      integer idiet_rc(4,10000000)
      real a(4),b(4),abm(4),stddev(4)
c
c dN15 values for dietary items
c
      a(1)=5.01
      a(2)=2.8
      a(3)=3.8
      a(4)=8.6
c
c dC13 values for dietary items
c
      b(1)=-20.3
      b(2)=-22.5
      b(3)=-13.1
      b(4)=-21.7
c
c set number of diets found and searched to zero
c
      i_found=0
      i_search=0
c
c loop over all possible values of the 1st dietary item
c
      do i1=0,100,1
         idiet1=i1
c
c loop over all possible values of the 2nd dietary item
c
```

Dietary FORTRAN Parameter Search Code, Thacker & Campbell 2005

```

do i2=0,100-idiet1,1
  idiet2=idiet1+i2
c
c loop over all possible values of the 3rd dietary item
c
      do i3=0,100-idiet2,1
        idiet3=idiet2+i3
c
c 4th value is calculated from 100 - sum of three previous
c
        i4=100-idiet3
        idiet=idiet3+i4
c
c error checking that should never be tripped
c
        if (idiet.ne.100) then
          write(*,*) idiet,i4,i3,i2,i1
          stop
        end if
c
c increment number of diets searched
c
        i_search=i_search+1
c
c calculate dN15 value for this particular diet
c
        rn15_1=float(i1)*a(1)+float(i2)*a(2)+float(i3)*a(3)
        rn15_2=float(i4)*a(4)
c
c values are multiples of 100, divide this out
c trophic fractionation 3.6
c
        rn15=(rn15_1+rn15_2)*0.01+3.6
c
c If the dN15 value matches the desired value (within the error
c bounds) then check the dC13 value of the diet, else skip to
c the next diet
c
        if (abs(rn15-10.1).lt.0.2) then
          rd13_1=float(i1)*b(1)+float(i2)*b(2)+float(i3)*b(3)
          rd13_2=float(i4)*b(4)
c
c values are multiples of 100, divide this out
c trophic fractionation 1.0
c
          rd13=(rd13_1+rd13_2)*0.01+1.0
c
c If the dC13 values matches the desired value (within the error
c bounds) then store the diet found. Increment the total number of
c diets found that match the required dN15 and dC13 values.
c
          if (abs(rd13-(-19.0)).lt.0.2) then
            i_found=i_found+1
            idiet_rc(1,i_found)=i1
            idiet_rc(2,i_found)=i2

```

Dietary FORTRAN Parameter Search Code, Thacker & Campbell 2005

```

        idiet_rc(3,i_found)=i3
        idiet_rc(4,i_found)=i4
    end if
end if
enddo
enddo
enddo
c
c Calculate average values for the diet items from all of the
c diets that match the requested dN15 and dC13 values
c
    do i=1,4
        sum=0.
        do j=1,i_found
            sum=sum+float(idiet_rc(i,j))
        end do
        rf1=sum/float(i_found)*0.01*a(i)
        rf2=sum/float(i_found)*0.01*b(i)
        abm(i)=sum/float(i_found)
    end do
c
c Print out the average dN15 and dC13 to check
c
    rn15_1=abm(1)*a(1)+abm(2)*a(2)+abm(3)*a(3)
    rn15_2=abm(4)*a(4)
    rn15=(rn15_1+rn15_2)*0.01+3.6
    rd13_1=abm(1)*b(1)+abm(2)*b(2)+abm(3)*b(3)
    rd13_2=abm(4)*b(4)
    rd13=(rd13_1+rd13_2)*0.01+1.0
    write(*,*) "average dN15, dC13",rn15,rd13
c
c Calculate standard deviation of all the dietary items
c
    do i=1,4
        sq=0.
        do j=1,i_found
            sq=sq+(idiet_rc(i,j)-abm(i))**2
        end do
        stddev(i)=sqrt(sq/float(i_found))
        write(*,*) 'Diet item',i,' val=',abm(i),stddev(i)
    end do
c
c Output some useful diagnostics
c
    write(*,*) 'Searched, found, ratio',i_search,i_found,
x            float(i_found)/float(i_search)

    end

```