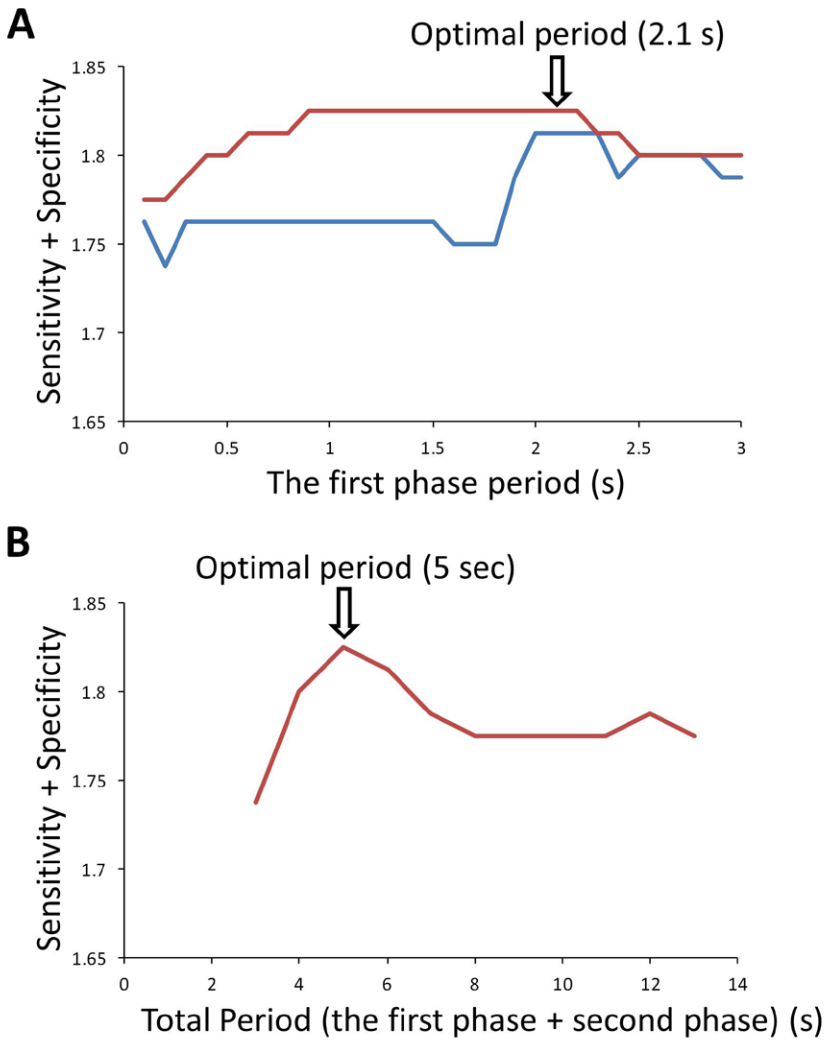

Fig. S1. Optimal periods for parameter calculations. (A) Different cut-off periods (0.1~3.0 seconds) and (B) total periods (3~13 seconds) were tested using the sum of sensitivity and specificity. Arrows indicate the optimal cut-off period (2.1 s) and total period (5 s). Red and blue lines represent the feeding behaviours on crab and fish, respectively. There was only the feeding behaviour on crab in the total period (B) because both of the two parameters used for identifying the feeding behaviour on fish were based on the first phase.

Movie 1. The body motion during the feeding behaviour on crab (crab-eating), reconstructed through the 3-axis accelerations and 3-axis angular velocities datasets. The data were taken at 200 Hz and the animation is played at 10 Hz. Note that the distance moved is not incorporated in the animation because estimation errors are significant when the acceleration values are relatively low during crab-eating.

Movie 2. The body motion during the feeding behaviour on fish (fish-eating), reconstructed through the 3-axis accelerations and 3-axis angular velocities datasets. The data were taken at 200 Hz and the animation is played at 10 Hz. Note that the cumulative distance is incorporated in the animation because the acceleration values are relatively high during fish-eating.

Movie 3. The body motion during the escape response (escape), reconstructed through the 3-axis accelerations and 3-axis angular velocities datasets. The data were taken at 200 Hz and the animation is played at 10 Hz. Note that the cumulative distance is incorporated in the animation because the acceleration values are relatively high during escape.

Movie 4. The body motion during the intraspecific interaction (intra-attack), reconstructed through the 3-axis accelerations and 3-axis angular velocities datasets. The data were taken at 200 Hz and the animation is played at 10 Hz. Note that the distance moved is not incorporated in the animation because the acceleration values are relatively low during intra-attack.

Movie 5. The body motion during the intraspecific interaction (intra-escape), reconstructed through the 3-axis accelerations and 3-axis angular velocities datasets. The data were taken at 200 Hz and the animation is played at 10 Hz. Note that the distance moved is not incorporated in the animation because the acceleration values are relatively low during intra-escape.

Movie 6. The body motion during the routine movement (routine), reconstructed through the 3-axis accelerations and 3-axis angular velocities datasets. The data were taken at 200 Hz and the animation is played at 10 Hz. Note that the distance moved is not incorporated in the animation because the acceleration values are relatively low during routine.

Table S1. Summary of the behavioural data detected by the set threshold (2.0 *G*) for six white-streaked groupers, *Epinephelus ongus*

ID	TL (mm)	BW (g)	Crab-eating	Fish-eating	Escape	Intra-attack	Intra-escape	Routine
A	247	217	0/0	0/0	8/8	3/16	6/9	0
B	220	151	1/1	2/2	9/9	0/0	10/22	1
C	293	354	5/5	2/2	6/6	4/21	0/0	3
D	255	256	5/5	11/11	5/5	1/4	3/4	5
E	245	219	2/2	1/1	7/7	0/1	8/13	3
F	261	297	4/4	18/18	7/7	1/6	0/0	4
Total			**17/17**	**34/34**	**42/42**	**9/48**	**27/48**	**16**

The numbers indicate the number of the detected behaviour (before the slash) and the number of the observed behaviour (after the slash).

TL, total length; BW, body weight; Crab-eating, feeding behaviour on crab; Fish-eating, feeding behaviour on fish; Escape, escape response; Intra-attack, intraspecific interaction (attack); Intra-escape, intraspecific interaction (escape); Routine, routine movement.

Table S2. Summary of the means and standard errors of the parameters (maximum value, range, mean, standard deviation) derived from the 3-axis accelerations and 3-axis angular velocities in all behaviours

Phase	Parameter		Crab-eating (n=17)			Fish-eating (n=34)			Escape (n=42)			Intra-attack (n=8)			Intra-escape (n=27)			Routine (n=16)			ANOVA	
			mean	s.e.m	TK	mean	s.e.m	TK	mean	s.e.m	TK	mean	s.e.m	TK	mean	s.e.m	TK	mean	s.e.m	TK	*F*-value	*P*-value
Phase1	AX (*G*)	max	5.78	1.05	a	8.95	0.70	b	13.70	0.99	c	3.10	0.79	a	2.46	0.28	a	2.75	0.40	a	30.73	<0.01
		range	9.25	1.60	a	14.74	0.92	b	21.81	1.59	c	4.40	1.10	a	3.45	0.46	a	4.25	0.64	a	34.50	<0.01
		mean	0.03	0.06	a	0.09	0.04	a	0.30	0.06	b	0.13	0.08	a	0.48	0.05	b	0.17	0.07	a	8.09	<0.01
		s.d.	0.48	0.07	a	0.84	0.06	b	1.51	0.12	c	0.27	0.06	a	0.29	0.03	a	0.26	0.03	a	34.67	<0.01
	AY (*G*)	max	4.87	0.82	a	9.02	0.62	b	14.02	1.10	c	3.10	1.37	a	2.69	0.29	a	3.27	0.75	a	28.04	<0.01
		range	8.10	1.34	a	13.90	0.92	b	20.69	1.62	c	4.35	1.62	a	3.91	0.42	a	5.04	1.01	a	28.28	<0.01
		mean	-0.03	0.03	b	-0.15	0.03	a	0.00	0.02	ab	-0.01	0.04	bc	0.07	0.02	b	-0.04	0.05	bc	8.33	<0.01
		s.d.	0.45	0.06	a	0.82	0.06	b	1.43	0.13	c	0.26	0.07	a	0.29	0.03	a	0.26	0.05	a	26.49	<0.01
	AZ (*G*)	max	4.61	0.62	ab	6.98	0.63	b	9.72	0.83	c	2.13	0.25	a	2.33	0.23	a	3.72	0.64	ab	18.16	<0.01
		range	7.94	1.23	ab	11.98	1.08	b	16.20	1.41	c	2.54	0.49	a	3.02	0.43	a	5.83	1.16	ab	19.39	<0.01
		mean	0.87	0.03	-	0.90	0.01	-	0.84	0.02	-	0.94	0.01	-	0.88	0.02	-	0.71	0.17	-	1.70	0.14
		s.d.	0.45	0.06	ab	0.66	0.05	b	1.02	0.10	c	0.16	0.03	a	0.23	0.03	a	0.34	0.05	ab	18.30	<0.01
	MA (*G*)	max	7.29	1.24	a	12.01	0.82	b	18.41	1.32	c	5.23	1.23	a	3.87	0.34	a	5.30	0.93	a	28.73	<0.01
		range	6.88	1.24	a	11.46	0.83	b	17.91	1.33	c	4.70	1.26	a	3.21	0.35	a	4.71	0.96	a	28.79	<0.01
		mean	1.06	0.03	a	1.22	0.03	a	1.58	0.07	b	1.04	0.01	a	1.13	0.01	a	1.08	0.05	a	18.11	<0.01
		s.d.	0.61	0.10	a	1.13	0.08	b	1.99	0.18	c	0.28	0.07	a	0.26	0.03	a	0.37	0.06	a	29.00	<0.01
	GX (degree/s)	max	895	117	ab	1048	69	bc	1248	104	c	282	66	a	464	53	a	733	130	ab	12.12	<0.01
		range	1540	219	ab	1610	98	b	2119	202	c	443	111	a	729	81	a	1211	227	ab	10.91	<0.01
		mean	8	2	-	1	2	-	1	4	-	-4	3	-	5	4	-	5	3	-	0.78	0.57

		s.d.	102	14	a	104	7	a	167	18	b	36	7	a	78	8	a	69	11	a	9.30	<0.01
	GY (degree/s)	max	1380	139	ab	2089	132	cd	2078	157	d	572	97	ab	529	60	a	1382	245	bc	19.33	<0.01
		range	2411	250	ab	3678	256	d	3636	279	d	896	189	ab	830	97	a	2289	409	bc	20.01	<0.01
		mean	3	4	-	2	2	-	6	3	-	-4	5	-	0	2	-	1	2	-	1.21	0.31
		s.d.	149	13	a	226	14	b	264	18	b	69	10	a	85	8	a	130	19	a	22.45	<0.01
	GZ (degree/s)	max	544	59	a	1306	105	b	2383	182	c	463	95	a	627	62	a	465	80	a	32.60	<0.01
		range	902	94	a	1686	113	a	3634	304	b	600	121	a	872	81	a	691	115	a	31.74	<0.01
		mean	-3	7	-	-6	8	-	-23	11	-	-4	11	-	-9	7	-	-1	6	-	0.85	0.51
		s.d.	86	10	a	154	12	b	322	22	c	74	17	ab	112	11	ab	55	9	a	34.93	<0.01
	MG (degree/s)	max	1569	166	a	2559	133	b	2997	204	b	746	109	a	885	75	a	1583	279	a	23.99	<0.01
		range	1563	166	a	2553	133	b	2988	204	b	739	109	a	876	75	a	1580	279	a	24.01	<0.01
		mean	99	13	a	112	8	a	219	16	b	69	13	a	113	11	a	57	7	a	21.12	<0.01
		s.d.	178	16	a	281	16	b	406	27	b	94	14	a	126	10	a	155	20	a	29.47	<0.01
Phase2	AX (*G*)	max	2.66	1.09	-	0.98	0.29	-	1.96	0.45	-	0.90	0.46	-	1.28	0.51	-	0.35	0.06	-	1.90	0.10
		range	3.99	1.65	b	1.40	0.51	ab	2.48	0.64	ab	1.33	0.84	ab	1.26	0.78	ab	0.18	0.03	a	1.98	0.08
		mean	0.01	0.07	ab	0.02	0.04	a	0.24	0.06	bc	0.12	0.07	abc	0.40	0.08	c	0.14	0.09	abc	5.14	<0.01
		s.d.	0.21	0.05	b	0.13	0.02	ab	0.24	0.04	b	0.16	0.05	ab	0.11	0.03	ab	0.04	0.01	a	3.68	<0.01
	AY (*G*)	max	1.56	0.63	-	0.97	0.39	-	1.56	0.49	-	0.83	0.66	-	0.52	0.25	-	0.23	0.04	-	1.26	0.29
		range	2.46	0.98	-	1.49	0.65	-	2.19	0.71	-	1.04	0.79	-	0.69	0.32	-	0.13	0.03	-	1.38	0.23
		mean	-0.04	0.02	a	-0.08	0.02	a	0.11	0.01	b	0.00	0.02	ab	0.05	0.01	b	-0.10	0.05	a	15.39	<0.01
		s.d.	0.13	0.04	-	0.10	0.03	-	0.16	0.03	-	0.09	0.04	-	0.09	0.02	-	0.03	0.01	-	1.66	0.15
	AZ (*G*)	max	3.66	1.22	-	1.36	0.15	-	1.59	0.21	-	1.36	0.24	-	1.86	0.76	-	0.94	0.04	-	2.46	<0.05
		range	4.60	1.87	b	0.93	0.35	ab	1.49	0.37	ab	0.81	0.46	ab	1.74	1.31	ab	0.11	0.03	a	2.29	<0.05

		mean	0.89	0.02	ab	0.94	0.01	b	0.85	0.02	ab	0.95	0.01	ab	0.91	0.01	ab	0.75	0.12	a	2.76	<0.05
		s.d.	0.21	0.07	b	0.07	0.02	a	0.12	0.02	ab	0.06	0.02	ab	0.10	0.04	ab	0.03	0.01	a	2.42	<0.05
	MA (*G*)	max	4.08	1.39	-	1.92	0.43	-	2.82	0.57	-	1.74	0.58	-	2.07	0.79	-	10.99	0.03	-	1.57	0.17
		range	3.40	1.44	-	1.09	0.45	-	2.02	0.59	-	0.89	0.59	-	1.13	0.82	-	16.75	0.03	-	1.73	0.13
		mean	0.98	0.03	a	1.00	0.01	a	1.04	0.02	ab	1.01	0.01	ab	1.09	0.01	b	0.60	0.01	a	6.89	<0.01
		s.d.	0.23	0.09	-	0.09	0.03	-	0.16	0.04	-	0.07	0.05	-	0.08	0.05	-	2.93	0.01	-	1.66	0.15
	GX (degree/s)	max	429	157	b	142	51	ab	196	55	ab	74	32	ab	191	90	ab	1	2	a	2.26	0.05
		range	710	258	b	244	91	ab	325	83	ab	115	52	ab	251	106	a	0	3	a	2.72	0.02
		mean	2	1	-	2	1	-	-1	1	-	1	2	-	5	4	-	1	1	-	1.47	0.20
		s.d.	39	13	b	19	5	ab	32	6	b	10	3	ab	29	6	b	0	0	a	3.16	<0.01
	GY (degree/s)	max	593	203	b	207	60	ab	310	96	ab	208	137	ab	159	68	a	19	5	a	2.56	<0.05
		range	995	342	b	357	109	a	525	160	ab	378	259	ab	264	119	a	33	9	a	2.48	<0.05
		mean	-3	2	-	-3	1	-	3	1	-	-2	3	-	-1	2	-	-1	1	-	2.49	<0.05
		s.d.	61	18	b	31	7	a	51	9	b	29	13	ab	26	6	ab	6	1	a	3.49	<0.01
	GZ (degree/s)	max	228	75	ab	178	47	ab	320	73	b	222	113	ab	216	79	ab	17	2	a	1.72	0.13
		range	336	119	ab	235	62	ab	495	115	b	290	133	ab	298	103	ab	25	4	a	2.07	0.07
		mean	5	4	-	1	5	-	-1	4	-	2	6	-	3	5	-	-2	1	-	0.25	0.94
		s.d.	31	8	abc	31	5	ab	58	9	c	41	13	ab	44	9	b	5	1	a	3.96	<0.01
	MG (degree/s)	max	746	248	b	264	73	ab	472	127	ab	293	164	ab	317	136	ab	25	5	a	2.23	0.05
		range	742	248	b	260	73	ab	467	127	ab	290	164	ab	314	135	ab	24	5	a	2.23	0.06
		mean	39	7	bc	38	5	b	62	7	c	34	9	bc	50	8	bc	9	1	a	5.96	<0.01
		s.d.	75	22	b	39	9	ab	63	12	b	44	16	ab	46	10	ab	5	1	a	2.75	<0.05

Different lower case letters in TK represent significant differences detected by a Tukey-Kramer *post hoc* test (P<0·05).

Crab-eating, feeding behaviour on crab; Fish-eating, feeding behaviour on fish; Escape, escape response; Intra-attack, intraspecific interaction (attack); Intra-escape, intraspecific interaction (escape); Routine, routine movement; ANOVA, analysis of variance; s.e.m, standard error; TK, Tukey-Kramer test; s.d., standard deviation; AX, lateral acceleration; AY, forward acceleration; AZ, vertical acceleration; MA, vector sum of the accelerations; GX, pitch angular velocity; GY, roll angular velocity; GZ, yaw angular velocity; MG, vector sum of the angular velocities

Table S3. Summary of the stomach content analysis of 158 white-streaked groupers, *Epinephelus ongus* (252±26 mm total length)

Prey types		Family	N	W (g)	F
Crustaceans			**27**	**50.53**	**21**
	Crabs	Portunidae	12	39.005	10
		Xanthidae	4	2.82	4
		Majoidae	1	0.08	1
	Shrimps	Hippolytidae	8	6.555	7
		Alpheidae	1	1.02	1
		Unidentified	1	1.05	1
Fishes			**14**	**41.16**	**14**
		Pomacentridae	4	12.58	4
		Labridae	1	10.47	1
		Holocentridae	1	0.48	1
		Pempheridae	1	4.5	1
		Lutjanidae	1	7.16	1
		Unidentified	6	5.97	6
Others			**1**	**18.69**	**1**
	Octopus	Unidentified	1	18.69	1
Total			**42**	**110.38**	**33**

Thirty three individuals had some prey items in their stomachs.

N, total number; W, total weight; F, frequency of occurrence

Script 1. The custom R program to find the optimal thresholds of the parameters using the sum of sensitivity and specificity.

```
#########################################################################
## Note that rather than the conventional "<-", we use "=" as the
## assignment operator.
#########################################################################

### 1. Finding the start-point for data extraction ###

# Read the time-series data (3-axis accelerations and 3-axis
# angular velocities) from the data file. The data file is a comma
# separated file, has 7 columns of values, and each column is assigned
# to a variable. The variable names are "point" (sequential serial number),
# "ax" (lateral acceleration), "ay" (forward acceleration)
# "az" (vertical acceleration), "gx" (pitch angular velocity)
# "gy" (roll angular velocity), "gz" (yaw angular velocity)

original = read.csv("data1.csv")

# calculations of the vector sum of accelerations ("ma") and
# angular velocities ("mg")

original$ma = sqrt(original$ax^2 + original$ay^2 + original$az^2)
original$mg = sqrt(original$gx^2 + original$gy^2 + original$gz^2)

# find the maximum absolute value in the 3-axis accelerations

temp1 = data.frame(abs(original$ax),
                   abs(original$ay),
                   abs(original$az))
original$max3 = apply(temp1, 1, max)

## find the start-point and save to a csv file after looping over
## nrow(pointdata)-1.
filename1 = "startpoint.csv"
```

```
pointdata = original[original$max3 > 2,]
   # We set the max3 to > 2, because all the feeding behaviours
   # exceeded the absolute 2.0 G in at least one of the three axes.
out = paste("startpoint", sep = ",")
write(out, file = filename1, append = TRUE)
out = paste(pointdata[1, 1] + 1, sep = ",")
write(out, file = filename1, append = TRUE)

index = 0
for(i in 1:(nrow(pointdata) - 1)){
  if (index <= 1000){
    # We set the index to <= 1000, because we want to extract
    # 5 seconds of the data (200Hz * 5 sec = 1000 data points)
    index = index + pointdata[i + 1, 1] - pointdata[i, 1]
  }
  else{
    out = paste(pointdata[i + 1, 1] + 1, sep = ",")
    write(out, file = filename1, append = TRUE)
    index = 0
  }
}

### the end of 1 ###


### 2. Calculating the parameters ###
# The following packages are required.
require(e1071)
require(stringr)
require(plyr)
require(reshape2)

# Custom functions for parameter calculations
absmax = function(x){
  max(abs(x))
}
```

```
range = function(x){
  max(x) - min(x)
}

fn1 = function(x) c(
  absmax = absmax(x),
  range = range(x),
  avg = mean(x),
  sd = sd(x),
  a2 = 1
)

pointdata = read.csv(filename1)

for(i in 1:(nrow(pointdata))) {
  startpoint = pointdata[i, "startpoint"]
  if(startpoint + 999 < nrow(original)) {

    # Parameter calculations for the first phase
    timeseries = original[startpoint:(startpoint + 419),]
    # We set 419 here because we wanted to extract 0 ~ 2.1 seconds
    # of the data (200Hz * 2.1 sec = 420 data points)
    timeseries = timeseries[c("ax", "ay", "az", "gx",
                          "gy", "gz", "ma", "mg")]
    output = data.frame(aaply(t(timeseries), 1, fn1))
    output$names = row.names(output)
    output = reshape(output, idvar = "a2", timevar="names",
                  direction="wide")

    # Parameter calculations for the second phase
    timeseries2 = original[(startpoint + 420):(startpoint + 999),]
    # We set 420 and 999 here because we wanted to extract 2.1 ~ 5
    # seconds of the data
    timeseries2 = timeseries2[c("ax", "ay", "az","gx", "gy",
                            "gz", "ma", "mg")]
```

```
    output2 = data.frame(aaply(t(timeseries2), 1, fn1))
    output2$names = row.names(output2)
    output2 = reshape(output2, idvar = "a2", timevar="names",
                    direction="wide")

    out = data.frame(output[2:ncol(output)], output2[2:ncol(output2)])
    out$startpoint = startpoint
    write.table(out, file = "parameters.csv",
              row.names = FALSE, col.names = FALSE, append = TRUE,
              quote = FALSE, sep = ",")
  }
}

### the end of 2 ###


### 3. Calculating the sum of sensitivity and specificity to find
### the optimal thresholds

# Read the datasets (fish ID, behavioral data that were determined by
# the video, and calculated parameters are needed).

# Here, our datasets have "fish" ("a"~"f"), "behavior"
# (Fc, Feeding-crab; Ff, Feeding-fish; Es, Escape; Ia, Intra-attack;
# Ie, Intra-escape; Rm, Routine movements) , "gxgyrange" (the ratio of
# the range of pitch angular velocity to the range of yaw angular
# velocity in the first phase), "avgmg2" (the mean vector sum of the
# angular velocities in the second phase), "sdax" (the standard
# deviation of the lateral acceleration in the first phase), and
# "gzgyrange" (the ratio of the range of yaw angular velocity to the
# range of roll angular velocity in the first phase).

test = read.csv("data2.csv")

# Separating the modeling data and test data.
# If you want to conduct the cross validation test, then use different
```

```
# fish IDs for the "model" and "testafter"
model = test[test$fish == "a" | test$fish == "b" | test$fish == "c" |
             test$fish == "d" | test$fish == "e" | test$fish=="f", ]
testafter = test[test$fish == "a" | test$fish == "b" |
                 test$fish == "c" | test$fish == "d" |
                 test$fish == "e" | test$fish == "f", ]


## 3.1. Calculating the sum of sensitivity + specificity
## for Feeding-crab
fcname = "mapfc.csv"
out = paste("b1", "b2", "sens", "spec", "youden", sep = ",")
write(out, file = fcname, append = TRUE)


for(i in 1:100){
  b1 = 0.02 * i
  # change the "gxgyrange" cut-off value from 0.02 to 2.00 to find the
  # optimal threshold
  a = model[model$gxgyrange >= b1,]
  if(nrow(a) >= 1){
    a$node1 = 1
  }
  b = model[model$gxgyrange < b1,]
  if(nrow(b) >= 1){
    b$node1 = 0
  }
  model2 = rbind(a, b)

  for(j in 1:100){
    b2 = 0.5 * j
    # change the "avgmg2" cut-off value from 0.5 to 50 to find the
    # optimal value
    a = model2[model2$avgmg2 >= b2,]
    if (nrow(a) >= 1){
      a$node2 = 1
```

```
    }
    b = model2[model2$avgmg2 < b2,]
    if(nrow(b) >= 1){
      b$node2 = 0
    }
    model3 = rbind(a, b)

    a = model3[model3$node1 * model3$node2 == 1,]
    if(nrow(a) >= 1){
      a$estimateFc = "Fc"
    }
    b = model3[model3$node1 * model3$node2 == 0,]
    if(nrow(b) >= 1){
      b$estimateFc = "O"
    }
    model4 = rbind(a, b)
    # calculation of the sensitivity
    sens = nrow(model4[model4$behavior == "Fc" &
                       model4$estimateFc == "Fc",]) /
      nrow(model4[model4$behavior == "Fc",])
    # calculation of the specificity
     spec = nrow(model4[model4$behavior != "Fc" &
                       model4$estimateFc != "Fc",]) /
      nrow(model4[model4$behavior != "Fc",])
    youden = sens + spec
    out = paste(b1, b2, sens, spec, youden, sep = ",")
    write(out, file = fcname, append = TRUE)
  }
}

# extracting the optimal thresholds
mapfc = read.csv(fcname)
M = max(mapfc$youden)
a = which(mapfc$youden == M)
# the optimal threshold of "gxgyrange"
cutb1=mean(mapfc[a, "b1"])
```

```
# the optimal threshold of "avgmg2"
cutb2=mean(mapfc[a, "b2"])

# mapping the sum of sensitivity and specificity against the
# "gxgyrange" and "avgmg2"
mapfc = matrix(mapfc$youden, nrow = 100)
mapfc = t(mapfc)
x = 1:nrow(mapfc)
y = 1:ncol(mapfc)
filled.contour(x, y, mapfc)

## the end of 3.1 ##

## 3.2. Calculating the sum of sensitivity + specificity for
# Feeding-fish
ffname = "mapff.csv"
out = paste("b1", "b2", "sens", "spec", "youden", sep = ",")
write(out, file = ffname, append = TRUE)

for(i in 1:100){
  # change the "sdax" cut-off value from 0.02 to 2.00 to find the
  # optimal threshold
  b1 = 0.02 * i
  a = model[model$sdax >= b1,]
  if(nrow(a) >= 1){
    a$node1 = 1
  }

  b = model[model$sdax < b1,]
  if(nrow(b) >= 1){
    b$node1 = 0
  }
  model2 = rbind(a, b)

  for(j in 1:100){
    # change the "gzgyrange" cut-off value from 0.02 to 2.00 to find
```

```
    # the optimal threshold
    b2 = 0.02 * j
    a = model2[model2$gzgyrange <= b2,]
    if(nrow(a) >= 1){
      a$node2 = 1
    }
    b = model2[model2$gzgyrange > b2,]
    if(nrow(b) >= 1){
      b$node2 = 0
    }
    model3 = rbind(a, b)
    a = model3[model3$node1*model3$node2 == 1,]
    if(nrow(a) >= 1){
      a$estimateFf = "Ff"
    }
    b = model3[model3$node1*model3$node2 == 0,]
    if(nrow(b) >= 1){
      b$estimateFf = "O"
    }

    model4 = rbind(a, b)
    # calculation of the sensitivity
    sens = nrow(model4[model4$behavior == "Ff" &
                       model4$estimateFf == "Ff",]) /
      nrow(model4[model4$behavior == "Ff",])
    # calculation of the specificity
    spec = nrow(model4[model4$behavior != "Ff" &
                       model4$estimateFf != "Ff",]) /
      nrow(model4[model4$behavior != "Ff",])
    youden = sens + spec
    out = paste(b1, b2, sens, spec, youden, sep = ",")
    write(out, file = ffname, append = TRUE)
  }
}

# extracting the optimal thresholds
```

```
mapff = read.csv(ffname)
M = max(mapff$youden)
a = which(mapff$youden == M)
# the optimal threshold of "sdax"
cutb1 = mean(mapff[a, "b1"])
# the optimal threshold of "gzgyrange"
cutb2 = mean(mapff[a, "b2"])

# mapping the sum of sensitivity and specificity against the "sdax"
# and "gzgyrange"
mapff = matrix(mapff$youden, nrow = 100)
mapff = t(mapff)
x = 1:nrow(mapff)
y = 1:ncol(mapff)
filled.contour(x, y, mapff)

## the end of 3.2 ##
### the end of 3 ###
```