DiNuzzo & Griffen 2020 – Technical discussion

Here we point out several issues regarding the implementation of the model of DiNuzzo and Griffen 2020, Proc. R. Soc. B 287: 20201095) in NetLogo. All issues cause the simulations to deviate from the model description in the paper and can influence the simulation outcomes in undesirable ways. Relevant lines of their code are highlighted in yellow; these lines are commented below the simulation code.

Netlogo code of simulation programme

(DiNuzzo & Griffen 2020, Section 1.3 of Supplementary Information)

```
1   patches-own [ quality
2               possible-consumption
3               expected-consumption
4               ]
5   breed [ active-animals active-animal ]
6   breed [ sedentary-animals sedentary-animal ]
7   turtles-own [ consumption-rate
8               avg-consumption-rate
9               time-since-moved ]
10  globals [ ;marginal-value
11          max-consumption
12          countdown
13          x
14          stop-countdown]
15
16  to setup
17    clear-all
18    setup-patches
19    setup-turtles
20    set countdown number-of-active + number-of-inactive
21    set stop-countdown 0
22    reset-ticks
23  end
```

```

to setup-patches
  resize-world (number-of-patches * -1) number-of-patches (number-of-patches * -1) number-of-patches  (1)
  ask patches
  [set quality (2 + random 8) (2)
  set pcolor scale-color green quality 1 10 ]
end

to setup-turtles
  create-active-animals number-of-active
  [ set color white
    set size .4
    setxy random-xcor random-ycor ]
  create-sedentary-animals number-of-inactive
  [ set color blue
    set size .4
    setxy random-xcor random-ycor ]

  ask turtles
  [ set time-since-moved number-of-active + number-of-inactive ]
end

to Go
  if stop-countdown > 50 or number-of-active + number-of-inactive = 0 (3)
  [ ask patches [ calculate-expected-consumption ]
    stop ]
  ask turtles
  [ set time-since-moved time-since-moved - 1 ]
  ask turtles
  [ calculate-consumption ]
  ask turtles
  [ calculate-avg-consumption-rate ]
  ask patches
    [ calculate-max-consumption ]
  ask patches
   [ calculate-expected-consumption ]
  move-turtles
  ifelse countdown > 1
  [ set countdown countdown - 1 ]
  [ set countdown number-of-active + number-of-inactive ]
  set stop-countdown stop-countdown + 1
  tick
end

to calculate-consumption
  set consumption-rate ( [ quality ] of patch-here ) / ( count turtles-here ) (4)
end

to calculate-avg-consumption-rate
  set avg-consumption-rate mean [ consumption-rate ] of turtles
end

to calculate-max-consumption
  ifelse TypeII-functional-response?
  [ifelse
    count turtles-here > 0
    [ let food-available (quality - ((count turtles-here + 1) * ( quality  / ( quality + count turtles-here + 1 )))) (5)
      ifelse food-available > max-feeding-rate (6)
      [set possible-consumption max-feeding-rate]
      [set possible-consumption food-available]]
    [ ifelse quality > max-feeding-rate (6)
      [set possible-consumption max-feeding-rate]
      [set possible-consumption quality]]]
  [ifelse
    count turtles-here > 0
    [ set possible-consumption ( quality ) / ( count turtles-here + 1 ) ]
    [ set possible-consumption quality ]]
  set max-consumption max [ possible-consumption ]  of patches
end

to calculate-expected-consumption
  ifelse TypeII-functional-response?
  [ifelse
    count turtles-here > 0
    [ let food-available (quality - ((count turtles-here + 1) * ( quality  / ( quality + count turtles-here + 1 )))) (5)
      ifelse food-available > max-feeding-rate (6)
      [set possible-consumption max-feeding-rate]
      [set possible-consumption food-available]]
    [ ifelse quality > max-feeding-rate (6)
      [set possible-consumption max-feeding-rate]
      [set possible-consumption quality]]]
  [ifelse
    count turtles-here > 0
    [ set possible-consumption ( quality ) / ( count turtles-here + 1 ) ]
    [ set possible-consumption quality ]]
end

to move-turtles
  set x random-float 1
  ask one-of turtles with-min [ time-since-moved ] (7)
  [ ifelse breed = active-animals
   [ ifelse consumption-rate > max-consumption (8)
        [ fd 0 ] (9)
```

```
117          [ if x < .8
118            [ move-to one-of patches with-max [ possible-consumption ]
119            set time-since-moved number-of-active + number-of-inactive
120            set stop-countdown 0 ]]]
121
122        [ ifelse consumption-rate > max-consumption (8)
123          [ fd 0 ] (9)
124          [ if x < .2
125             [ move-to one-of patches with-max [ possible-consumption ]
126              set time-since-moved number-of-active + number-of-inactive
127               set stop-countdown 0 ]]]
128     ]
129 end
```

Notes:

(1) This line defines the size of the grid, which corresponds to the number of food patches. The term "number-of-patches" in line 26 is somewhat misleading, as it does not indicate the actual number of patches in the model, but rather the number of rows and columns in the quadratic grid of patches. Let us, for clarity, rename the parameter "number-of-patches" to *S*. One would expect that the size of the grid is given by S^2. According to line 26 of the code however the grid coordinates run from -*S* to +*S* in both the horizontal and the vertical direction, resulting in a grid of size $(2S+1)^2$. As a consequence, the setting *S*=7 results in 15^2=225 patches instead of the expected value of 49, and to obtain 49 patches, *S* would have to be set to 3. From the replication of DiNuzzo and Griffen's results, we believe that this is the setting they used, so there is no real error here, merely a potential for confusion.

(2) In line 28, a resource quality is randomly assigned to each patch. The article states that quality levels range from 1 to 9. In view of line 28, quality levels actually range from 2 to 9.

(3) Line 47 defines the stop condition for the simulation. Importantly, the simulation does not stop when the ideal free distribution (IFD) is reached. Even if the IFD has not yet been attained, the simulation stops when individuals cease to move for 50 time steps. This can readily occur when the population harbours a large proportion of very inactive individuals. In such a case, the stop criterion leads to a premature stop of the simulation and, hence, to an underestimation of the time it takes to reach the IFD.

(4) In line 69, the actual *per capita* intake rate of individuals is calculated. Let us, for clarity, write line 69 in a more succinct mathematical notation: $I(R,C) = R / C$, which states that the *per capita* intake rate (= "consumption-rate") $I(R,C)$ is given by the local resource quality *R* divided by the local number of consumers *C*. In other words, $I(R,C)$ is given by a linear ratio-dependent functional response (or a ratio-dependent type 1 functional response without limitation; see [Abrams & Ginzburg 2000, TREE 15(8): 337-341]. All this is fine as long as the standard version of the model of DiNuzzo and Griffen is concerned. The problem is that they use the expression in line 69 also in case of a type 2 (ratio-dependent) functional response, which, under their simplifying assumptions, is given by $I_2(R,C) = R/(C+R)$. In the simulations of Section 1.4 of their Supplementary Information, it would have been necessary to exchange $I(R,C)$ by $I_2(R,C)$ in line 69.

(5) In lines 80 and 98, the programme calculates the term "food-available" for the case of a type 2 functional response. What does this term indicate? Let us for clarity denote it by $L(R,C)$, for a patch with resource quality *R* and a number *C* of consumers. According to the programming code, $L(R,C)$ is given by $L(R,C) = R-(C+1)\cdot R/(R+C+1)$. The second term on the right-hand side corresponds to the total resource consumption $(C+1)\cdot I_2(R,C+1)$ under the assumption that the number of consumers on the patch under consideration is increased by one. Accordingly, $L(R,C)$ corresponds to the resources left *unconsumed* after the number of consumers is increased by one. The problem is that (in lines 83 and 101) $L(R,C)$ is equated with the expected intake rate ("possible-consumption") in the case that an individual would move to the patch and join the *C* consumers already present there. By definition, the expected intake is, in case of a type 2 functional response, given by $I_2(R,C+1)$. Accordingly, the term $L(R,C)$ needs to be exchanged by I(R,C+1).

(6) Correcting the mistake in (5) would have prevented another problem that occurs in lines 81, 84, 99 and 102 of the code. The type 2 functional response $I_2(R,C)$ cannot exceed the "max-feeding-rate" 1, but the function $L(R,C)$, which is erroneously used instead of $I_2(R,C)$, is typically larger than 1 at low population densities. This has important implications for the simulation programme: for many patches (including patches with lowest *R*=2), the variable "food-available" (= $L(R,C)$) is larger than "max-feeding-rate" (=1). As a result, "possible-consumption" for all these patches is set to 1 (lines 82, 85, 100, 103), with as consequence that many patches (including patches with very low quality) are considered optimal (they all achieve the "max-consumption" value 1 in line 91). In a correctly operating simulation programme, this should have lead to a situation where *all* individuals are motivated to move (since their intake rate $I_2(R,C)$ is smaller than 1) and where the moving individuals would more or less move at random over the grid (to *any* cell with $L(R,C) > 1$). However, this is prevented by the bug mentioned in (4): even in the scenario with type 2 functional response the individual intake rate is determined by the type 1 functional response $I(R,C) = R/C$, which is – for the low population densities considered – typically larger than 1. As a consequence, very few individuals are actually motivated to move. All these problems would have been prevented if the "consumption-rate" $I(R,C)$ in line 69 and the "food-available" in lines 80 and 98 would have been replaced by the intake rate $I_2(R,C)$. With this change, the parameter "max-feeding-rate" is no longer necessary, and the code could have been streamlined considerably.

(7) In line 113, an individual is selected in order to be asked whether it is motivated and willing to move to a patch with a higher intake rate. According to DiNuzzo and Griffen's model description, individuals are selected at random from all individuals in the population. In contrast, the simulation code introduces a bias, by restricting the random selection to those individuals that have not moved for the longest time (with minimal "time-since-moved"). This deviation from the model description could accelerate the time until the IFD is reached. This feature also becomes important in (8).

(8) Lines 115 and 122 specify whether active or inactive individuals are motivated to move. As discussed above, the condition for *not* being motivated to move ("consumption-rate > max-consumption") is problematic in case of a type 2 functional response. But even in case of a type 1 functional response, it does not lead to the intended behaviour. In this case, "consumption-rate" corresponds to the intake rate $I(R,C) = R/C$ on the current patch, while "max-consumption" corresponds to the maximal value of $I(R,C+1) = R/(C+1)$, that is, to the maximal intake rate that can be achieved elsewhere, should the individual decide to join the C individuals already present on a different patch. Should $I(R,C)$ be strictly larger than this maximum, an individual is indeed not motivated to move. But it will move if its "consumption-rate" is equal to "max-consumption" – which can easily happen in view of the small number of possible resource levels. In such a case, individuals are *always* motivated to move, irrespective of the fact that they are already occupying a patch with maximal intake rate. This risks an infinite loop, when movements never cease.

(9) Lines 116 and 123 specify what individuals should do if they are on an optimal patch, where their "consumption-rate" is larger than the "max-consumption" that could be achieved elsewhere. In essence, the command "fd 0" tells them to do nothing. Importantly, this does not only mean that they do not move, but they also do not update their "time-since-moved". As a consequence, they eventually fill up the category "turtles with min[time-since-moved]" (turtles that have not moved for the longest time, see (7)) and it gets progressively more difficult to select non-optimal individuals from this category. It is therefore likely that there are still individuals with suboptimal intake rates after 50 time steps have passed and the simulation ends (see (3)). As a result the stopping rule in three would bring the simulation to a halt before the IFD is attained.