

```

1  # this file contains functions for dispersal, competition, niche evolution and drift
2
3
4  ##### rangeDispersal #####
5
6  # Function extends a species range based on a dispersal kernel and the inherent
  niche breadth of the species
7
8  ##### Arguments #####
9
10 # env = current environmental layer
11 # position = niche position of species
12 # breadth = niche breadth of species
13 # species.ras is the current species range in raster format (presence/absence)
14
15 disperse_ds <- function (demetable.species,
16                           env,
17                           env.table,
18                           dispersal.range,
19                           suitability.mode="block"){
20
21   # group demes by similarity of the niche - so for very similar niches the
  environment is calculated just once
22
23   # this is a clunky grouping method - there may be more efficient functions in
  data.table
24   demetable.species$niche1.breadth.group <- as.integer(round(demetable.species$
  niche1.breadth/niche.blocksize))
25   demetable.species$niche1.position.group <- as.integer(round(demetable.species$
  niche1.position/niche.blocksize))
26   demetable.species$niche2.breadth.group <- as.integer(round(demetable.species$
  niche2.breadth/niche.blocksize))
27   demetable.species$niche2.position.group <- as.integer(round(demetable.species$
  niche2.position/niche.blocksize))
28   if (demetable.species[,.N] < 35) {browser()}
29   # create groups of demes with similar environment to calculate dispersal
30   niche.groups <- demetable.species[, list(niche1.breadth.group,
  niche1.position.group, niche2.breadth.group, niche2.position.group, .N), by=list(
  niche1.breadth.group, niche1.position.group, niche2.breadth.group,
  niche2.position.group)]
31   niche.groups <- niche.groups[, list(niche1.breadth.group, niche1.position.group,
  niche2.breadth.group, niche2.position.group, N)]
32
33   # loop through the niche groups
34   for (k in 1:nrow(niche.groups)) {
35     niche <- niche.groups[k, ]
36     niche.values <- niche * niche.blocksize # turns the rounded niche group into
  niche values
37
38     demetable.nichegroup <- demetable.species[niche1.breadth.group == niche$
  niche1.breadth.group &
39
40                                     niche1.position.group== niche$
  niche1.position.group &
41                                     niche2.breadth.group == niche$
  niche2.breadth.group &
42                                     niche2.position.group== niche$
  niche2.position.group, 1:12]
43
44   # filter env cells to those within the spatial limits (for the niche group)
  bounds <- as.list(demetable.nichegroup[, list(xmin=(min(x)-dispersal.range
  ), xmax=(max(x)+dispersal.range), ymin=(min(y)-dispersal.range), ymax=(max(y)+
  dispersal.range))])
45   bounds[ bounds < 0 ] <- 0
46   #niche.extent <- extent(bounds[[1]], bounds[[2]], bounds[[3]], bounds[[4]])
47
48   env.table.dispersal <- env.table[col >= bounds$xmin &
49                                     col <= bounds$xmax &
50                                     row >= bounds$ymin &

```

```

51         row <- bounds$ymax, ]
52     #env.dispersal <- crop(env, niche.extent)
53
54     # filter env cells to those within the niche limits
55     niche1.min <- niche.values$niche1.position.group - niche.values$niche1.breadth /2
56     niche1.max <- niche.values$niche1.position.group + niche.values$niche1.breadth /2
57
58     #niche2.min <- niche.values$niche2.position.group - niche.values$niche2.breadth /2
59     #niche2.max <- niche.values$niche2.position.group + niche.values$niche2.breadth /2
60
61     env.table.dispersal <- env.table.dispersal[env1 >= niche1.min & env1 <=
62     niche1.max, ] # need to add 2nd niche dimension
63
64     #apply niche suitability function to env.dispersal to give suitability for 0 to 1
65     env.table.dispersal$suitability <- niche_suitability(env=env.table.dispersal$env1
66     , niche1.breadth = niche.values$niche1.breadth, niche1.position = niche.values$
67     niche1.position, suitability.mode = suitability.mode)
68
69     if (env.table.dispersal[, .N] > 0) {
70         #remove dispersal destinations which are within the bounding rectangle, but
71         #too far away
72         include.in.dispersal <- close.enough(demetable.nichegroup[, c("x", "y")],
73         env.table.dispersal[, c("col", "row")],
74         dispersal.range)
75         env.table.dispersal <- env.table.dispersal[include.in.dispersal]
76     }
77
78     # process this nichegroup only if there are cells to disperse to
79     if (nrow(env.table.dispersal) > 0) {
80
81         demetable.nichegroup.new <- demetable.nichegroup[0, ]
82         row.pointer <- 0
83
84         # loop through each deme for the niche group
85         for (d in 1:nrow(demetable.nichegroup)) {
86
87             deme <- demetable.nichegroup[d, ]
88
89             # remove once dispersal is working correctly
90             # if (do.display) {
91             #     display.update(list(one_deme=deme))
92             # }
93
94             # find the cells in dispersal distance
95             deme.dest <- env.table.dispersal[(col >= deme$x-dispersal.range
96             & col <= deme$x+dispersal.range
97             & row >= deme$y-dispersal.range
98             & row <= deme$y+dispersal.range), ]
99
100             #deme.dest <- deme.dest[row >= deme$y-dispersal.range & row <=
101             deme$y+dispersal.range, ]
102             new.count <- nrow(deme.dest)
103
104             new.amount <- deme$amount * deme.dest$suitability # should return a vector
105             new.rows <- (row.pointer+1):(row.pointer+new.count)
106
107             if (length(new.amount) > 0) {
108                 # create dispersed demes
109                 demetable.nichegroup.new <- rbind(demetable.nichegroup.new,
110                 list(cellID=deme.dest$cellID,
111                 x=deme.dest$col,
112                 y=deme.dest$row,
113                 amount=deme$amount * deme.dest$
114                 suitability), fill=T)
115
116                 demetable.nichegroup.new$speciesID[new.rows] <- deme$speciesID
117                 demetable.nichegroup.new$niche1.position[new.rows] <- deme$niche1.position
118                 demetable.nichegroup.new$niche1.breadth[new.rows] <- deme$niche1.breadth
119                 demetable.nichegroup.new$niche2.position[new.rows] <- deme$niche2.position

```

```

110 demetable.nichegroup.new$niche2.breadth[new.rows] <- deme$niche2.breadth
111 demetable.nichegroup.new$gene.pos1[new.rows] <- deme$gene.pos1
112 demetable.nichegroup.new$gene.pos2[new.rows] <- deme$gene.pos2
113 demetable.nichegroup.new$gene.pos3[new.rows] <- deme$gene.pos3
114
115 demetable.nichegroup.new$originCell[new.rows] <- deme$cellID
116
117 # apply distance function
118 weights <- distance.weights(source = deme[1, c("x", "y")],
119                             destinations = demetable.nichegroup.new[
120                                 new.rows, c("x", "y")],
121                             dispersal.range = dispersal.range,
122                             distance.type = "euclidean",
123                             distance.decay = "linear")
124
125 demetable.nichegroup.new$amount[new.rows] <- demetable.nichegroup.new$
126 amount[new.rows] * weights
127 new.count <- length(which(demetable.nichegroup.new$amount[new.rows] > 0))
128 demetable.nichegroup.new <- demetable.nichegroup.new[
129     demetable.nichegroup.new$amount > 0, ]
130
131 }
132
133 row.pointer <- row.pointer + new.count
134
135 demetable.nichegroup.new <- demetable.nichegroup.new[amount > 0, ]
136
137 # combine the niche group demetable rows
138 if (exists("demetable.species.new")) {
139     demetable.species.new <- rbind(demetable.species.new,
140     demetable.nichegroup.new)
141 } else {
142     demetable.species.new <- demetable.nichegroup.new
143 }
144 }
145
146 return(demetable.species.new)
147
148 }

```