```
;;;;;;;;;;;;;;;;;;;;
;; Main Code
;;;;;;;;;;;;;;;;;;;;

extensions [cf csv ]
__includes ["collision.nls"
          "reaction.nls"
          "molecule-char.nls"
          "enzyme-kinetics.nls"
          "performance-rules.nls"
          "genome.nls"
          "data-collection.nls"
          "molecule-char-run.nls"
          ]

;;Global variables that affects all agents
globals [
  tick-delta                  ;; how much we advance the tick counter this time through
  max-tick-delta              ;; the largest tick-delta is allowed to be
  viscosity ;; thickness of cell
  performance ;; performance value, efficiency of the pathways
  mutation-prob ;; mutation probability
  performance-reproduce ;; RRT reproduction threshold
  generation-reproduce  ;; time necessary to replicate
  performance-reproduce-record ;; keeps track of current performance for counter
  generations-reproduce-record ;; keeps track of current time for counter
  generations ;; generation of proto-cell

;; probabilities that molecules will be removed from the simulation
;; due to implicit assumptions in the model of flux and side-reactions

  implicit-prob-gone
  implicit-prob-here
  implicit-prob-gone-a-1-n
  implicit-prob-here-a-1-n
  implicit-prob-gone-a-2-n
  implicit-prob-here-a-2-n
  implicit-prob-gone-a-3-n
  implicit-prob-here-a-3-n
  implicit-prob-gone-a-4-n
  implicit-prob-here-a-4-n
  implicit-prob-gone-a-5-n
  implicit-prob-here-a-5-n
  implicit-prob-gone-a-1-g
  implicit-prob-here-a-1-g
  implicit-prob-gone-a-2-g
```

```
  implicit-prob-here-a-2-g
  implicit-prob-gone-a-3-g
  implicit-prob-here-a-3-g
  implicit-prob-gone-a-4-g
  implicit-prob-here-a-4-g
  implicit-prob-gone-a-5-g
  implicit-prob-here-a-5-g
  implicit-prob-gone-a-1-r
  implicit-prob-here-a-1-r
  implicit-prob-gone-a-2-r
  implicit-prob-here-a-2-r
  implicit-prob-gone-a-3-r
  implicit-prob-here-a-3-r
  implicit-prob-gone-a-4-r
  implicit-prob-here-a-4-r
  implicit-prob-gone-a-5-r
  implicit-prob-here-a-5-r
  implicit-prob-gone-substrate-2
  implicit-prob-here-substrate-2
  implicit-prob-gone-substrate-3
  implicit-prob-here-substrate-3
  implicit-prob-gone-substrate-4
  implicit-prob-here-substrate-4
  implicit-prob-gone-substrate-5
  implicit-prob-here-substrate-5
  implicit-prob-gone-sub-s
  implicit-prob-here-sub-s

  birth-mut-counter ;; another artifact of the counter

;; probabilities for the normal enzyme to have a functional GoF mutation
  a-1-n-evo-prob
  a-2-n-evo-prob
  a-3-n-evo-prob
  a-4-n-evo-prob
  a-5-n-evo-prob
  a-1-g-evo-prob
  a-2-g-evo-prob
  a-3-g-evo-prob
  a-4-g-evo-prob
  a-5-g-evo-prob

;; GoF MPV
  a-1-g-performance-1
  a-2-g-performance-1
  a-3-g-performance-1
```

```
  a-4-g-performance-1
  a-5-g-performance-1

  a-1-g-performance-2
  a-2-g-performance-2
  a-3-g-performance-2
  a-4-g-performance-2
  a-5-g-performance-2


  a-1-n-performance-1
  a-2-n-performance-1
  a-3-n-performance-1
  a-4-n-performance-1
  a-5-n-performance-1

  a-1-r-performance-1
  a-2-r-performance-1
  a-3-r-performance-1
  a-4-r-performance-1
  a-5-r-performance-1
]

;; types of molecules in simulation
breed [substrate-1 substrate-1s]
breed [a-1-n]
breed [a-1-g]
breed [a-2-n]
breed [a-2-g]
breed [a-3-n]
breed [a-3-g]
breed [a-4-n]
breed [a-4-g]
breed [a-5-n]
breed [a-5-g]
breed [a-1-r]
breed [a-2-r]
breed [a-3-r]
breed [a-4-r]
breed [a-5-r]
breed [substrate-2]
breed [substrate-3]
breed [substrate-4]
breed [substrate-5]
breed [genome]
breed [modified-substrate-2]
```

```
breed [modified-substrate-3]
breed [modified-substrate-4]
breed [modified-substrate-5]
breed [modified-substrate-1-s]

turtles-own
[
 compound-name
 mass
 radius
 speed
 energy
 last-collision
 rxn-reactant ;; tag for complex formation
 rxn-type ;; tag for identifying which reaction a enzyme does
 rxn-time-dif ;; counter for binding
 rxn-prob ;; reaction probability
 performance-1 ;; performance for reaction type 1
 performance-2 ;; performance for reaction type 2, if the enzyme does more than
;; one reaction

 rxn-tot-1 ;; counts # of reactions in simulation so far of type 1
 rxn-tot-2 ;; same as above but for type 2
 mut-prob-run ;; probability of performance- mutaiton
 evo-tag ;; tag for recording GoF event
 r-tag ;; tag for recording paralogous gene event
 next-gen-tag ;; tag for resetting # of enzymes in model after replication\
]

to setup-globals
 ;; According to
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1225847/pdf/biophysj00084-
0290.pdf
 ;; viscosity of cytoplasm is roughly 0.0011 kPa
 set viscosity 0.0011
 ;; Arbitrary value to allow tick progression to occur in a reasonable manner
 set max-tick-delta 0.1073
 set init-substrate-1-conc 100
 ;; if changed, change in the go procedure as well
 set performance 500
 set performance-reproduce 0
 set generation-reproduce 40
 set performance-reproduce-record 0
 set generations-reproduce-record 0
 set birth-mut-counter 0
 implicit-probability
```

```
  genome-evo ;; holds tag for GoF history
  genome-r ;; holds tag for paralogous gene history
  resize-world -100 100 -100 100 ;; size of simulation in terms of patches
end


to diffusion
;; works according to Stokes theorem and mean-squared displacement
set init-speed sqrt (2 * (1.3806495279 * 10 ^ -19 * temperature * 10 ^ 20) /
    (6 * pi * viscosity * radius))
end

;; Means of having appropriate speeds
to kinetic
  set energy (0.5 * mass * (speed ^ 2))
end

to setup
  clear-all
  reset-ticks
  setup-globals
  number-of-agents
  ask turtles [
    diffusion
    set rxn-reactant nobody
    set evo-tag []
    set r-tag []
  ]
  calculate-tick-delta ;; allows collisions to occur between time-steps (continuous)
;; characterizes molecules at the beginning of the simulation
  substrate-1-char
  a-1-n-char
  a-2-n-char
  a-3-n-char
  a-4-n-char
  a-5-n-char
  substrate-2-char
  substrate-3-char
  substrate-4-char
  substrate-5-char
  modified-substrate-2-char-i
  modified-substrate-3-char-i
  modified-substrate-4-char-i
  modified-substrate-5-char-i
  modified-substrate-1-s-char-i
  a-1-r-char
```

```
  a-2-r-char
  a-3-r-char
  a-4-r-char
  a-5-r-char
  a-1-g-char
  a-2-g-char
  a-3-g-char
  a-4-g-char
  a-5-g-char
  genome-char
end

;;; Have to insert every rxn that is possible in the go function ;;;

to go
  ;; *Place end conditions here as a conditional*
  ask turtles
  [check-for-collision
  diffusion
  move
  if not (breed = genome) [set next-gen-tag 0]

  ;; Add rxn-functions here. For speed, write an if function
  ;; specifying the breed and rxn-reactant conditions, as shown below

  if breed = a-1-n and (rxn-reactant = "complex")

;; the three numbers next to dissociate are binding time, original mass, and
;; original radius of the enzyme

    [ask self [dissociate-a1-&-s 15 3 3]]
  if breed = a-1-n and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-1-n-&-sub-rxn]

  if breed = a-2-n and (rxn-reactant = "complex")
    [ask self [dissociate-a2-&-mod-a1 15 3 3]]
  if breed = a-2-n and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-2-n-&-substrate-2-rxn]

  if breed = a-3-n and (rxn-reactant = "complex")
    [ask self [dissociate-a3-&-mod-a2 15 3 3]]
  if breed = a-3-n and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-3-n-&-substrate-3-rxn]
```

```
  if breed = a-4-n and (rxn-reactant = "complex")
    [ask self [dissociate-a4-&-mod-a3 15 3 3]]
  if breed = a-4-n and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-4-n-&-substrate-4-rxn]

  if breed = a-5-n and (rxn-reactant = "complex")
    [ask self [dissociate-a5-&-mod-a4 15 3 3]]
  if breed = a-5-n and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-5-n-&-substrate-5-rxn]

;;;;;;;;;;;;;;
;; a-#-s
;;;;;;;;;;;;;;

  if breed = a-1-g and (rxn-reactant ="complex") and (rxn-type = 1)
  [ask self [dissociate-a1s-&-mod-a2 15 3 3]]
  if breed = a-1-g and (rxn-reactant ="complex") and (rxn-type = 2)
  [ask self [dissociate-a1s-&-substrate-1 15 3 3]]
  if breed = a-1-g and (rxn-reactant = nobody)
  [a-1-g-&-substrate-3-rxn
    a-1-g-&-sub-rxn]

  if breed = a-2-g and (rxn-reactant ="complex") and (rxn-type = 1)
  [ask self [dissociate-a2s-&-mod-a3 15 3 3]]
  if breed = a-2-g and (rxn-reactant ="complex") and (rxn-type = 2)
  [ask self [dissociate-a2s-&-mod-a1 15 3 3]]
  if breed = a-2-g and (rxn-reactant = nobody)
  [a-2-g-&-substrate-4-rxn
    a-2-g-&-mod-a1-rxn]

  if breed = a-3-g and (rxn-reactant ="complex") and (rxn-type = 1)
  [ask self [dissociate-a3s-&-mod-a4 15 3 3]]
  if breed = a-2-g and (rxn-reactant ="complex") and (rxn-type = 2)
  [ask self [dissociate-a3s-&-mod-a2 15 3 3]]
  if breed = a-2-g and (rxn-reactant = nobody)
  [a-3-g-&-substrate-5-rxn
    a-3-g-&-mod-a2-rxn]

  if breed = a-4-g and (rxn-reactant ="complex") and (rxn-type = 1)
  [ask self [dissociate-a4s-&-sub 15 3 3]]
  if breed = a-4-g and (rxn-reactant ="complex") and (rxn-type = 2)
  [ask self [dissociate-a4s-&-mod-a-3 15 3 3]]
  if breed = a-2-g and (rxn-reactant = nobody)
```

```
  [a-4-g-&-sub-rxn
    a-4-g-&-mod-a3-rxn]

  if breed = a-5-g and (rxn-reactant = "complex") and (rxn-type = 1)
  [ask self [dissociate-a5s-&-mod-a-4 15 3 3]]
  if breed = a-5-g and (rxn-reactant = "complex") and (rxn-type = 2)
  [ask self [dissociate-a5s-&-mod-a-1 15 3 3]]
  if breed = a-5-g and (rxn-reactant = nobody)
  [a-5-g-&-mod-a4-rxn
    a-5-g-&-mod-a1-rxn]

;;;;;;;;;;
;; a-#-r
;;;;;;;;;;

;; Add rxn-functions here. For speed, write an if function
  ;; specifying the breed and rxn-reactant conditions, as shown below

  if breed = a-1-r and (rxn-reactant = "complex")
    [ask self [dissociate-a1r-&-s-s 15 3 3]]
  if breed = a-1-r and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-1-r-&-sub-s-rxn]

  if breed = a-2-r and (rxn-reactant = "complex")
    [ask self [dissociate-a2r-&-mod-a1s 15 3 3]]
  if breed = a-2-r and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-2-r-&-modified-substrate-2-rxn]

  if breed = a-3-r and (rxn-reactant = "complex")
    [ask self [dissociate-a3r-&-mod-a2s 15 3 3]]
  if breed = a-3-r and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-3-r-&-modified-substrate-3-rxn]

  if breed = a-4-r and (rxn-reactant = "complex")
    [ask self [dissociate-a4r-&-mod-a3s 15 3 3]]
  if breed = a-4-r and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
  [a-4-r-&-modified-substrate-4-rxn]

  if breed = a-5-r and (rxn-reactant = "complex")
    [ask self [dissociate-a5r-&-mod-a4s 15 3 3]]
  if breed = a-5-r and (rxn-reactant = nobody)
  ;; If an agent has more than 1 possible reaction, add it inside the below bracket
```

```
  [a-5-r-&-modified-substrate-5-rxn]

  if breed = genome
  ;; Every 3 ticks, genome produce is called
  [if ticks mod 3 = 0 [genome-produce]]

  ;; Mutation probability before hatching in genome. This command overrides the
command
  ;; from the mutation command prior to a hatched molecule's data.

  ]


  ;calculate-tick-delta

  ;; Survival conditions
  if performance < -10000 or ticks > 2000 [
    stop
    ]
  ;; For the reproduction of the cell
  ;; induce mutation prob change after reproduction...?
  if (performance-reproduce - performance + performance-reproduce-record < 0)
  and
  (generation-reproduce - ticks + generations-reproduce-record < 0)
  [set generations (1 + generations)
    set performance-reproduce-record performance
    set generations-reproduce-record ticks
    set birth-mut-counter 1
    ;; if changed here, change above in setup as well
    ;; This performance change represents the new generation's ability
    ;; to handle the environment under specified RTPV
    ;set performance 500

;; adds a tag that allows 30 random enzymes to survive
    ifelse count a-1-n > 30[
    ask n-of 30 a-1-n
    [set next-gen-tag 1]]
    [ask a-1-n
      [set next-gen-tag 1]]

    ifelse count a-1-g > 30[
    ask n-of 30 a-2-n
    [set next-gen-tag 1]]

    [ask a-2-n
      [set next-gen-tag 1]]
```

```
ifelse count a-3-n > 30[
ask n-of 30 a-3-n
[set next-gen-tag 1]]

[ask a-3-n
  [set next-gen-tag 1]]
ifelse count a-4-n > 30
[
ask n-of 30 a-4-n
[set next-gen-tag 1]]

[ask a-4-n
  [set next-gen-tag 1]]
;; a-5-n
ifelse count a-5-n > 30[
ask n-of 30 a-5-n
[set next-gen-tag 1]]

[ask a-5-n
  [set next-gen-tag 1]]
;; a-1-g
ifelse count a-1-g > 30
[ask n-of 30 a-1-g
  [set next-gen-tag 1]]

[ask a-1-g
  [set next-gen-tag 1]]
ifelse count a-2-g > 30
[ask n-of 30 a-2-g
  [set next-gen-tag 1]]
[ask a-2-g
  [set next-gen-tag 1]]
;; a-3-g
ifelse count a-3-g > 30
[ask n-of 30 a-3-g
  [set next-gen-tag 1]]
[ask a-3-g
  [set next-gen-tag 1]]
ifelse count a-4-g > 30
[ask n-of 30 a-4-g
  [set next-gen-tag 1]]
[ask a-4-g
  [set next-gen-tag 1]]
;; a-5-g
 ifelse count a-5-g > 30
[ask n-of 30 a-5-g
```

```
  [set next-gen-tag 1]]
[ask a-4-g
  [set next-gen-tag 1]]
;; a-1-r
ifelse count a-1-r > 30
[ask n-of 30 a-1-r
  [set next-gen-tag 1]]
[ask a-1-r
  [set next-gen-tag 1]]
ifelse count a-2-r > 30
[ask n-of 30 a-2-r
  [set next-gen-tag 1]]
[ask a-2-r
  [set next-gen-tag 1]]
ifelse count a-3-r > 30
[ask n-of 30 a-3-r
  [set next-gen-tag 1]]
[ask a-3-g
  [set next-gen-tag 1]]
ifelse count a-4-r > 30
[ask n-of 30 a-4-r
  [set next-gen-tag 1]]
[ask a-4-r
  [set next-gen-tag 1]]
 ifelse count a-5-r > 30
[ask n-of 30 a-5-r
  [set next-gen-tag 1]]
[ask a-5-r
  [set next-gen-tag 1]]

;;;;;;;;;;;;;;;
;; substrate-1s
;;;;;;;;;;;;;;;
ifelse count substrate-1 > 300
[ask n-of 30 substrate-1
  [set next-gen-tag 1]]
[ask substrate-1
  [set next-gen-tag 1]]

;; substrate-2
ifelse count substrate-2 > 300
[ask n-of 30 substrate-2
  [set next-gen-tag 1]]
[ask substrate-2
  [set next-gen-tag 1]]
```

```
ifelse count substrate-3 > 300
[ask n-of 30 substrate-3
  [set next-gen-tag 1]]
[ask substrate-3
  [set next-gen-tag 1]]


ifelse count substrate-4 > 300
[ask n-of 30 substrate-4
  [set next-gen-tag 1]]
[ask substrate-4
  [set next-gen-tag 1]]


ifelse count substrate-5 > 300
[ask n-of 30 substrate-5
  [set next-gen-tag 1]]
[ask substrate-5
  [set next-gen-tag 1]]

ifelse count modified-substrate-1-s > 300
[ask n-of 30 modified-substrate-1-s
  [set next-gen-tag 1]]
[ask modified-substrate-1-s
  [set next-gen-tag 1]]

ifelse count modified-substrate-2 > 300
[ask n-of 30 modified-substrate-2
  [set next-gen-tag 1]]
[ask modified-substrate-2
  [set next-gen-tag 1]]


ifelse count modified-substrate-3 > 300
[ask n-of 30 modified-substrate-3
  [set next-gen-tag 1]]
[ask modified-substrate-3
  [set next-gen-tag 1]]

ifelse count modified-substrate-4 > 300
[ask n-of 30 modified-substrate-4
  [set next-gen-tag 1]]
[ask modified-substrate-4
  [set next-gen-tag 1]]
```

```
  ifelse count modified-substrate-5 > 300
  [ask n-of 30 modified-substrate-5
    [set next-gen-tag 1]]
  [ask modified-substrate-5
    [set next-gen-tag 1]]


  ask turtles
  [if not (next-gen-tag = 1)
    [die]]
  ]
 birth-mut-global ;; calls for birth counter
 birth-mut-counter-proc ;; calls for helper function to birth counter

;; allows implicit appearance and removal of molecules to simulation
 implicit-interaction-here-sub
 implicit-interaction-here-a-1-n
 implicit-interaction-here-a-2-n
 implicit-interaction-here-a-3-n
 implicit-interaction-here-a-4-n
 implicit-interaction-here-a-5-n
 implicit-interaction-here-substrate-2
 implicit-interaction-here-substrate-3
 implicit-interaction-here-substrate-4
 implicit-interaction-here-substrate-5
 implicit-interaction-here-a-1-r
 implicit-interaction-here-a-2-r
 implicit-interaction-here-a-3-r
 implicit-interaction-here-a-4-r
 implicit-interaction-here-a-5-r
 implicit-interaction-here-a-1-g
 implicit-interaction-here-a-2-g
 implicit-interaction-here-a-3-g
 implicit-interaction-here-a-4-g
 implicit-interaction-here-a-5-g
 implicit-interaction-here-sub-s


 implicit-interaction-gone-sub
 implicit-interaction-gone-a-1-n
 implicit-interaction-gone-a-2-n
 implicit-interaction-gone-a-3-n
 implicit-interaction-gone-a-4-n
 implicit-interaction-gone-a-5-n
 implicit-interaction-gone-a-1-g
```

```
  implicit-interaction-gone-a-2-g
  implicit-interaction-gone-a-3-g
  implicit-interaction-gone-a-4-g
  implicit-interaction-gone-a-5-g
  implicit-interaction-gone-a-1-r
  implicit-interaction-gone-a-2-r
  implicit-interaction-gone-a-3-r
  implicit-interaction-gone-a-4-r
  implicit-interaction-gone-a-5-r
  implicit-interaction-gone-substrate-2
  implicit-interaction-gone-substrate-3
  implicit-interaction-gone-substrate-4
  implicit-interaction-gone-substrate-5
  implicit-interaction-gone-sub-s



  tick-advance 1
  display



end

to move
  if patch-ahead (speed * tick-delta) != patch-here
    [ set last-collision nobody ]
  jump (speed * tick-delta)
  set heading random-float 360
end
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Collision Module
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Code is heavily derived from:
;; Wilensky, U. (1997). NetLogo GasLab Free Gas model.
;; http://ccl.northwestern.edu/netlogo/models/GasLabFreeGas.
;; Center for Connected Learning and Computer-Based Modeling, Northwestern
University, Evanston, IL.

to average
  set avg-speed  mean [speed] of turtles
  set avg-energy  mean [energy] of turtles
  set total-energy sum [energy] of turtles
end


to calculate-tick-delta
  ;; Tick-delta allows it so that the fastest particle will move
  ;; 1 patch at any given instant, as NetLogo does not allow floating
  ;; integer movement.
  ifelse any? turtles with [speed > 0]
    [ set tick-delta min list (1 / (ceiling max [speed] of turtles)) max-tick-delta ]
    [ set tick-delta max-tick-delta ]
end

to check-for-collision  ;; particle procedure

  if count other turtles-here = 1
  [
    ;; the following conditions are imposed on collision candidates:
    ;;   1. they must have a lower who number than my own, because collision
    ;;      code is asymmetrical: it must always happen from the point of view
    ;;      of just one particle.
    let candidate one-of other turtles-here with [myself != last-collision]
    ;; we also only collide if one of us has non-zero speed. It's useless
    ;; (and incorrect, actually) for two particles with zero speed to collide.
```

```
    ;; This initiates the actual collision function and records the last-collision
    if (candidate != nobody) and (speed > 0 or [speed] of candidate > 0)
    [
      collide-with candidate
      set last-collision candidate
      ask candidate [ set last-collision myself ]
    ]
  ]
end

;; implements a collision with another particle.
;;
;; THIS IS THE HEART OF THE PARTICLE SIMULATION, AND YOU ARE STRONGLY
ADVISED
;; NOT TO CHANGE IT UNLESS YOU REALLY UNDERSTAND WHAT YOU'RE DOING!
;;
;; The two particles colliding are self and other-particle, and while the
;; collision is performed from the point of view of self, both particles are
;; modified to reflect its effects. This is somewhat complicated, so I'll
;; give a general outline here:
;;   1. Do initial setup, and determine the heading between particle centers
;;      (call it theta).
;;   2. Convert the representation of the velocity of each particle from
;;      speed/heading to a theta-based vector whose first component is the
;;      particle's speed along theta, and whose second component is the speed
;;      perpendicular to theta.
;;   3. Modify the velocity vectors to reflect the effects of the collision.
;;      This involves:
;;        a. computing the velocity of the center of mass of the whole system
;;           along direction theta
;;        b. updating the along-theta components of the two velocity vectors.
;;   4. Convert from the theta-based vector representation of velocity back to
;;      the usual speed/heading representation for each particle.
;;   5. Perform final cleanup and update derived quantities.
to collide-with [ other-turtles ] ;; particle procedure
  ;;; PHASE 1: initial setup

  ;; for convenience, grab some quantities from other-particle
  let mass2 [mass] of other-turtles
  let speed2 [speed] of other-turtles
  let heading2 [heading] of other-turtles

  ;; since particles are modeled as zero-size points, theta isn't meaningfully
  ;; defined. we can assign it randomly without affecting the model's outcome.
  let theta (random-float 360)
```

```
;;; PHASE 2: convert velocities to theta-based vector representation

;; now convert my velocity from speed/heading representation to components
;; along theta and perpendicular to theta
let v1t (speed * cos (theta - heading))
let v1l (speed * sin (theta - heading))

;; do the same for other-particle
let v2t (speed2 * cos (theta - heading2))
let v2l (speed2 * sin (theta - heading2))



;;; PHASE 3: manipulate vectors to implement collision

;; compute the velocity of the system's center of mass along theta
let vcm (((mass * v1t) + (mass2 * v2t)) / (mass + mass2) )

;; now compute the new velocity for each particle along direction theta.
;; velocity perpendicular to theta is unaffected by a collision along theta,
;; so the next two lines actually implement the collision itself, in the
;; sense that the effects of the collision are exactly the following changes
;; in particle velocity.
set v1t (2 * vcm - v1t)
set v2t (2 * vcm - v2t)



;;; PHASE 4: convert back to normal speed/heading

;; now convert my velocity vector into my new speed and heading
set speed sqrt ((v1t ^ 2) + (v1l ^ 2))
set energy (0.5 * mass * (speed ^ 2))
;; if the magnitude of the velocity vector is 0, atan is undefined. but
;; speed will be 0, so heading is irrelevant anyway. therefore, in that
;; case we'll just leave it unmodified.
if v1l != 0 or v1t != 0
  [ set heading (theta - (atan v1l v1t)) ]

;; and do the same for other-particle
ask other-turtles [
  set speed sqrt ((v2t ^ 2) + (v2l ^ 2))
  set energy (0.5 * mass * (speed ^ 2))
```

```
   if v2l != 0 or v2t != 0
     [ set heading (theta - (atan v2l v2t)) ]
  ]
end
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Data-collection
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

;; Used for collecting data. File must be saved in a .csv file to be analyzed

```
;; opens file for the data to be saved into
to new-file
  let file user-new-file
  if is-string? file
```

```
  [
; if file-exists? file
;    [file-delete file]
    file-open file
    ;; Need to print the headers
    file-print csv:to-row
    (
     list "Ticks" "ID" "Compound name" "evo-tag" "generations" "performance-1"
     "performance-2" "performance" "reaction_total_type-1" "reaction_total_type-2"
     )
    write-to-file
  ]
end

;; obtains data from the turtles
to-report get-vals
  report
  (list ticks who compound-name evo-tag generations performance-1 performance-2
    performance
    rxn-1-counter rxn-2-counter
    )
end

;; allows turtle data to be written in the new file
to write-to-file

  ;; use SORT so the turtles print their data in order by who number,
  ;; rather than in random order
  foreach sort turtles [
    ask ? [
      file-print csv:to-row get-vals
    ]
  ]
  file-print ""  ;; blank line
end


;; A dead turtle will retain its ID, thus no function is necessary to compensate for a
'lost' ID from
;; a dead turtle. New turtles will have the next highest ID of all turtles prior to the
turtle's death.

;; Reports the number of reactions that the agent type has done
to-report rxn-1-counter
if breed = a-1-n
  [report sum [rxn-tot-1] of a-1-n]
```

```
if breed = a-2-n
  [report sum [rxn-tot-1] of a-2-n]
if breed = a-3-n
  [report sum [rxn-tot-1] of a-3-n]
if breed = a-4-n
  [report sum [rxn-tot-1] of a-4-n]
if breed = a-5-n
  [report sum [rxn-tot-1] of a-5-n]
if breed = a-1-g
  [report sum [rxn-tot-1] of a-1-g]
if breed = a-2-g
  [report sum [rxn-tot-1] of a-2-g]
if breed = a-3-g
  [report sum [rxn-tot-1] of a-3-g]
if breed = a-4-g
  [report sum [rxn-tot-1] of a-4-g]
if breed = a-5-g
  [report sum [rxn-tot-1] of a-5-g]
  report 0
end

to-report rxn-2-counter
if breed = a-1-g
  [report sum [rxn-tot-2] of a-1-g]
if breed = a-2-g
  [report sum [rxn-tot-2] of a-2-g]
if breed = a-3-g
  [report sum [rxn-tot-2] of a-3-g]
if breed = a-4-g
  [report sum [rxn-tot-2] of a-4-g]
if breed = a-5-g
  [report sum [rxn-tot-2] of a-5-g]
  report 0
end
```

```
;; --------------------
;; Enzyme Kinetics
;; --------------------

;; Use the Enzyme Kinetics equation to find parameter values

;; If you are adding a new reaction, copy and paste the "react" and
;; "form-product" functions, switching substrate-1-molecule for the substrate-1 in
"react"
;; and the product in "form-product" for whatever is relevant.
;; Follow format for the a-1-n-&-substrate-1 reaction
;; Make a new dissociation function, with changes to the "hatch" function and the
"breed-here"


;; -----------------------------
;; COMPLEX FORMATION FUNCTIONS
;; -----------------------------

;; Change the "substrate-1-here" to whatever the reactant for the
;; reaction is
;; This sets a tag for the enzyme to transform into the complex
;; For enzymes with multiple active sites, simply change "one-of" into "n-of"

to form-complex-a1-&-s
  let mol-1 one-of (other substrate-1-here)
  if mol-1 = nobody [stop]

  set rxn-reactant "complex"
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
```

```
  set rxn-time-dif ticks
end

to form-complex-a2-&-mod-a1
  let mol-1 one-of (other substrate-2-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end

to form-complex-a3-&-mod-a2
  let mol-1 one-of (other substrate-3-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end

to form-complex-a4-&-mod-a3
  let mol-1 one-of (other substrate-4-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end

to form-complex-a5-&-mod-a4
```

```
  let mol-1 one-of (other substrate-5-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;; Gain-of-function proteins
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;a-1s
to form-complex-a1s-&-mod-a2
  let mol-1 one-of (other substrate-3-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end


to form-complex-a1s-&-sub
  let mol-1 one-of (other substrate-1-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end
```

```
;; a2s
to form-complex-a2s-&-mod-a3
  let mol-1 one-of (other substrate-4-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end


to form-complex-a2s-&-mod-a1
  let mol-1 one-of (other substrate-2-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end
;;a-3s
to form-complex-a3s-&-mod-a4
  let mol-1 one-of (other substrate-5-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end
```

```
to form-complex-a3s-&-mod-a2
  let mol-1 one-of (other substrate-3-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end

;; a4s
to form-complex-a4s-&-sub
  let mol-1 one-of (other substrate-1-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end


to form-complex-a4s-&-mod-a-3
  let mol-1 one-of (other substrate-4-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end

;; a-5-g
```

```
to form-complex-a5s-&-mod-a-4
  let mol-1 one-of (other substrate-5-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end


to form-complex-a5s-&-mod-a-1
  let mol-1 one-of (other substrate-2-here)
  if mol-1 = nobody [stop]
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
  set rxn-time-dif ticks
end


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; a-#-r proteins
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to form-complex-a1r-&-s-s
  let mol-1 one-of (other modified-substrate-1-s-here)
  if mol-1 = nobody [stop]

  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-reactant "complex"
```

```
  set rxn-time-dif ticks
end


to form-complex-a2r-&-modified-substrate-2
  let mol-1 one-of (other modified-substrate-2-here)
  if mol-1 = nobody [stop]

  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-time-dif ticks
  set rxn-reactant "complex"
end

to form-complex-a3r-&-modified-substrate-3
  let mol-1 one-of (other modified-substrate-3-here)
  if mol-1 = nobody [stop]

  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]

  set rxn-time-dif ticks
    set rxn-reactant "complex"
end

to form-complex-a4r-&-modified-substrate-4
  let mol-1 one-of (other modified-substrate-4-here)
  if mol-1 = nobody [stop]

  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
```

```
  set rxn-time-dif ticks
  set rxn-reactant "complex"
end

to form-complex-a5r-&-modified-substrate-5
  let mol-1 one-of (other modified-substrate-5-here)
  if mol-1 = nobody [stop]

  set rxn-reactant "complex"
  ask self
  [
   set mass mass + [mass] of mol-1
   set radius radius + [radius] of mol-1
   set size radius
  ]
  ask mol-1 [die]
  set rxn-time-dif ticks
end

;; Binding time
;; enzyme procedure that controls the rate at which complexed turtles break apart
;; this affects the enzyme agent


;; ---------------------------
;; DISSOCIATION FUNCTIONS
;; ---------------------------

;; Need to make a dissociate function for every reaction
;; If the product is the same, no need to make a new dissociation function

to dissociate-a1-&-s [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
;; time difference to prevent binding to multiple substrates when in complex
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
          ;; sets characteristics of enzyme to normal
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-tot-1 rxn-tot-1 + 1
        set rxn-reactant nobody
        set performance performance + performance-1
```

```
;; makes product
        hatch-substrate-2 1 [substrate-2-char]
      ]]
  stop
end

to dissociate-a2-&-mod-a1 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-3 1 [substrate-3-char]
      ]]
  stop
end

to dissociate-a3-&-mod-a2 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-4 1 [substrate-4-char]
      ]]
  stop
end

to dissociate-a4-&-mod-a3 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
```

```
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-5 1 [substrate-5-char]
       ]]
  stop
end

to dissociate-a5-&-mod-a4 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-1 1 [substrate-1-char]
       ]]
  stop
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Gain-of-function
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;a-#-s

to dissociate-a1s-&-mod-a2 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
```

```
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-2 rxn-tot-2 + 1
        set performance performance + performance-1
        hatch-modified-substrate-2 1 [modified-substrate-2-char]
       ]]
  stop
end

to dissociate-a1s-&-substrate-1 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-2
        hatch-substrate-2 1 [substrate-2-char]
       ]]
  stop
end

;a2s

to dissociate-a2s-&-mod-a3 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-2 rxn-tot-2 + 1
        set performance performance + performance-2
        hatch-modified-substrate-3 1 [modified-substrate-3-char]
       ]]
  stop
end
```

```
to dissociate-a2s-&-mod-a1 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-3 1 [substrate-3-char]
       ]]
  stop
end


to dissociate-a3s-&-mod-a4 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-2 rxn-tot-2 + 1
        set performance performance + performance-2
        hatch-modified-substrate-4 1 [modified-substrate-4-char]
       ]]
  stop
end

to dissociate-a3s-&-mod-a2 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
```

```
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-4 1 [substrate-4-char]
       ]]
  stop
end

;;a4


to dissociate-a4s-&-sub [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-2 rxn-tot-2 + 1
        set performance performance + performance-2
        hatch-modified-substrate-5 1 [modified-substrate-5-char]
       ]]
  stop
end

to dissociate-a4s-&-mod-a-3 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-substrate-5 1 [substrate-5-char]
       ]]
```

```
  stop
end
;;a-5s


to dissociate-a5s-&-mod-a-4 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-2 rxn-tot-2 + 1
        set performance performance + performance-2
        hatch-substrate-1 1 [substrate-1-char]
      ]]
  stop
end



to dissociate-a5s-&-mod-a-1 [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-modified-substrate-1-s 1 [modified-substrate-1-s-char]
      ]]
  stop
end


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; a-#-r
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


to dissociate-a1r-&-s-s [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [

        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-tot-1 rxn-tot-1 + 1
        set rxn-reactant nobody
        set performance performance + performance-1
        hatch-modified-substrate-2 1 [modified-substrate-2-char]
      ]]
  stop
end

to dissociate-a2r-&-mod-a1s [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-modified-substrate-3 1 [modified-substrate-3-char]
      ]]
  stop
end

to dissociate-a3r-&-mod-a2s [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
```

```
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-modified-substrate-4 1 [modified-substrate-4-char]
      ]]
  stop
end

to dissociate-a4r-&-mod-a3s [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-modified-substrate-5 1 [modified-substrate-5-char]
      ]]
  stop
end

to dissociate-a5r-&-mod-a4s [bind-time orig-mass orig-rad]
  if rxn-reactant = "complex"
  [
  let rxn-dis? (ticks - rxn-time-dif)
  if bind-time < rxn-dis?
        [
        set mass orig-mass
        set radius orig-rad
        set size radius
        set rxn-time-dif 0
        set rxn-reactant nobody
        set rxn-tot-1 rxn-tot-1 + 1
        set performance performance + performance-1
        hatch-modified-substrate-1-s 1 [modified-substrate-1-s-char]
      ]]
  stop
```

```
end
;; Reaction File

;; Please copy and paste the first reaction as a template if you are making your own
reaction.
;; Change the name of the reactant (read: substrate-1) in the ifelse line and create a
new dissociate and form
;; -complex function
;; in the enzyme-kinetics file. Follow those instructions.
;; Otherwise, leave everything the same.

;; Add the rxn function to the designated place in the go function.
;; Remember, add a rxn-prob in the molecule-chars function as well!

;; For additions to rxns, change the substrate-1-here in second line of code and
;; rename the form-complex function

to a-1-n-&-sub-rxn
;; Complex formation

  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other turtle itself now
  set rxn-reactant one-of substrate-1-here
  ;; If no other turtle in area, then rxn-reactant = nobody
  if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-1-n-here != nobody [ask other a-1-n-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
  ;; sets rxn-reactant as the original turtle now
    [ ask rxn-reactant [ set rxn-reactant myself ]
      ;; rxn-type differentiates this reaction with other possible a-1-n reactions
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a1-&-s
    ;; Product formation. dissociate inputs = [ bind-time original-mass original-radius]
    ;; If a change is made here, please reflect the change in the above dissociate
function as well
end

;; a-2-n and substrate-2
to a-2-n-&-substrate-2-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-2-here
```

```
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-2-n-here != nobody [ask other a-2-n-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a2-&-mod-a1
end

;; a-3-n and substrate-3
to a-3-n-&-substrate-3-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-3-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-3-n-here != nobody [ask other a-3-n-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a3-&-mod-a2
end

;; a-4-n and substrate-4
to a-4-n-&-substrate-4-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-4-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-4-n-here != nobody [ask other a-4-n-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a4-&-mod-a3
end
```

```
;; a-5-n and substrate-5
to a-5-n-&-substrate-5-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-5-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-5-n-here [ask other a-5-n-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a5-&-mod-a4
end

;;;;;;;;;;;;;;;;;;;;;;;;;
;; A-#-G
;;;;;;;;;;;;;;;;;;;;;;;;;

;;a-1-g
to a-1-g-&-substrate-3-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-3-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-1-g-here [ask other a-1-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 2]
    [ set rxn-reactant nobody stop]
    form-complex-a1s-&-mod-a2
end

to a-1-g-&-sub-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-1-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-1-g-here [ask other a-1-g-here [ stop ] ]
```

```
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a1s-&-sub
end

;;a2s
to a-2-g-&-substrate-4-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-4-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-2-g-here [ask other a-2-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 2]
    [ set rxn-reactant nobody stop]
    form-complex-a2s-&-mod-a3
end

to a-2-g-&-mod-a1-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-2-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-2-g-here [ask other a-2-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a2s-&-mod-a1
end

;;a3
to a-3-g-&-substrate-5-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
```

```
  set rxn-reactant one-of substrate-5-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-3-g [ask other a-3-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 2]
    [ set rxn-reactant nobody stop]
    form-complex-a3s-&-mod-a4
end

to a-3-g-&-mod-a2-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-3-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-3-g-here [ask other a-3-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a3s-&-mod-a2
end

;a4
to a-4-g-&-sub-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-1-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-4-g [ask other a-4-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 2]
    [ set rxn-reactant nobody stop]
    form-complex-a4s-&-sub
end
```

```
to a-4-g-&-mod-a3-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-4-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-4-g-here [ask other a-4-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a4s-&-mod-a-3
end

;; a-5-g

to a-5-g-&-mod-a4-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-5-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-5-g-here [ask n-of (count a-5-g-here - 1) a-5-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a5s-&-mod-a-4
end


to a-5-g-&-mod-a1-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of substrate-2-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if any? other a-5-g-here [ask n-of (count a-5-g-here - 1) a-5-g-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
```

```
    set rxn-type 2]
   [ set rxn-reactant nobody stop]
   form-complex-a5s-&-mod-a-1
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; A-#-r
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


to a-1-r-&-sub-s-rxn
;; Complex formation

  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other turtle itself now
  set rxn-reactant one-of modified-substrate-1-s-here
  ;; If no other turtle in area, then rxn-reactant = nobody
  if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-1-r-here != nobody [ask other a-1-r-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
  ;; sets rxn-reactant as the original turtle now
    [ ask rxn-reactant [ set rxn-reactant myself ]
      ;; rxn-type differentiates this reaction with other possible a-1-n reactions
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a1r-&-s-s
    ;; Product formation. dissociate inputs = [ bind-time original-mass original-radius]
    ;; If a change is made here, please reflect the change in the above dissociate
function as well
end

;; a-2-n and substrate-2
to a-2-r-&-modified-substrate-2-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of modified-substrate-2-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-2-r-here != nobody [ask other a-2-r-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
```

```
     set rxn-type 1]
   [ set rxn-reactant nobody stop]
   form-complex-a2r-&-modified-substrate-2
end

;; a-3-n and substrate-3
to a-3-r-&-modified-substrate-3-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of modified-substrate-3-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-3-r-here != nobody [ask other a-3-r-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a3r-&-modified-substrate-3
end

;; a-4-n and substrate-4
to a-4-r-&-modified-substrate-4-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of modified-substrate-4-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
  if other a-4-r-here != nobody [ask other a-4-r-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a4r-&-modified-substrate-4
end

;; a-5-n and substrate-5
to a-5-r-&-modified-substrate-5-rxn
  if rxn-reactant != nobody [stop]
  ;; value of rxn-reactant is the actual other reactant now
  set rxn-reactant one-of modified-substrate-5-here
 if rxn-reactant = nobody [ stop ]
  ;; If another enzyme is present, it stops the function of all but one enzyme
```

```
  if any? other a-5-r-here [ask other a-5-r-here [ stop ] ]
  ;; Determines the reactant for this reaction. random-float and rxn-prob determines
% of reaction success
  ifelse ((random-float 100) <= rxn-prob)
    [ ask rxn-reactant [ set rxn-reactant myself ]
      set rxn-type 1]
    [ set rxn-reactant nobody stop]
    form-complex-a5r-&-modified-substrate-5
en

;;;;;;;;;;;;;;;;;;;;
;;;;; Genome
;;;;;;;;;;;;;;;;;;;;

;; Set up separately from the numbers- function for ease of reading in module
to genome-char
  create-genome 1
ask genome
[
  set mass 10000000
  set radius 30
  set speed .00000001
  set last-collision nobody
  set rxn-reactant nobody
  set size radius
  set color 42
  set shape "starry"
  set compound-name "genome"
  set evo-tag []
  set r-tag []
  set next-gen-tag 1
  ]

end

;; Allows genome to produce proteins
to genome-produce
  ;; For a-1-n, allowing it to evolve into a Gain-of-function form upon
  ;; the evo-probability. The evo-tag provides a memory of this occurence
  ;; First case is when it did not evolve
  ifelse ((random-float 1000) >= a-1-n-evo-prob and not member? 1 evo-tag)
  [
    hatch-a-1-n 1
       [
         ;; Any changes with mutation probability occurs here
    set mut-prob-run 100
```

```
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-n-char-run
      ]
  set performance performance - 4]
;; Second case when it does evolve
[
  ;; Set a record that the evolution has occurred
  ask self [set evo-tag fput 1 evo-tag]
  hatch-a-1-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-g-char-run]
  set performance performance - 4
  ]
;; Third case is after it evolved
if member? 1 evo-tag
[ifelse ((random-float 100) >= a-1-g-evo-prob and not member? 1 r-tag)
  ;;
[hatch-a-1-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-g-char-run]
  set performance performance - 4
  ]
[
  ask self [set r-tag fput 1 r-tag]
  hatch-a-1-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-g-char-run]
  set performance performance - 4

  hatch-a-1-r 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
```

```
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-r-char-run]
  set performance performance - 4
  ]]
 if member? 1 r-tag
 [
  hatch-a-1-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-g-char-run]
  set performance performance - 4

  hatch-a-1-r 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-1-r-char-run]
  set performance performance - 4
  ]

;;;;;;;;;;;;;;;;;;;;;;;;
; a-2-
;;;;;;;;;;;;;;;;;;;;;;;;

ifelse ((random-float 1000) >= a-2-n-evo-prob)
[
  hatch-a-2-n 1
     [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-n-char-run
      ]
  set performance performance - 4]
[
  ;; Set a record that the evolution has occurred
  ask self [set evo-tag fput 2 evo-tag]
  hatch-a-2-g 1
  [
```

```
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-g-char-run
  ]
  set performance performance - 6
  ]
 if member? 2 evo-tag
  [ifelse ((random-float 100) >= a-2-g-evo-prob and not member? 2 r-tag)

 [hatch-a-2-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-g-char-run]
  set performance performance - 4
  ]
 [
  ask self [set r-tag fput 2 r-tag]
  hatch-a-2-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-g-char-run]
  set performance performance - 4

  hatch-a-2-r 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-r-char-run]
  set performance performance - 4
  ]]
   if member? 2 r-tag
 [
  hatch-a-2-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
```

```
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-g-char-run]
  set performance performance - 4

  hatch-a-2-r 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-2-r-char-run]
  set performance performance - 4
  ]
;;;;;;;;;;;;;;;;;;;;;;;;
; a-3-
;;;;;;;;;;;;;;;;;;;;;;;;

  ifelse ((random-float 1000) >= a-3-n-evo-prob and not member? 3 evo-tag)
[
  hatch-a-3-n 1
     [
      ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-3-n-char-run
      ]
  set performance performance - 4]
[
  ;; Set a record that the evolution has occurred
  ask self [set evo-tag fput 3 evo-tag]
  hatch-a-3-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-3-g-char-run
   ]
  set performance performance - 4
  ]
if member? 3 evo-tag
[ifelse ((random-float 100) >= a-3-g-evo-prob and not member? 3 r-tag)

[hatch-a-3-g 1
```

```
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-3-g-char-run]
 set performance performance - 4
 ]
[
 ask self [set r-tag fput 3 r-tag]
 hatch-a-3-g 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-3-g-char-run]
 set performance performance - 4

 hatch-a-3-r 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-3-r-char-run]
 set performance performance - 4
 ]]
 if member? 3 r-tag
[
 hatch-a-3-g 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-3-g-char-run]
 set performance performance - 4

 hatch-a-3-r 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-3-r-char-run]
```

```
  set performance performance - 4
  ]
;;;;;;;;;;;;;;;;;;;;;;;;
; a-4-
;;;;;;;;;;;;;;;;;;;;;;;;


   ifelse ((random-float 1000) >= a-4-n-evo-prob and not member? 4 evo-tag)
[
  hatch-a-4-n 1
     [
       ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-4-n-char-run
      ]
  set performance performance - 4]
[
  ;; Set a record that the evolution has occurred
  ask self [set evo-tag fput 4 evo-tag]
  hatch-a-4-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-4-g-char-run
  ]
  set performance performance - 4
  ]
if member? 4 evo-tag
[ifelse ((random-float 100) >= a-4-g-evo-prob and not member? 4 r-tag)

[hatch-a-4-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-4-g-char-run]
  set performance performance - 4
  ]
[
  ask self [set r-tag fput 4 r-tag]
  hatch-a-4-g 1
```

```
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-4-g-char-run]
 set performance performance - 4

 hatch-a-4-r 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-4-r-char-run]
 set performance performance - 4
 ]]
 if member? 4 r-tag
[
 hatch-a-4-g 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-4-g-char-run]
 set performance performance - 4

 hatch-a-4-r 1
 [
 ;; Any changes with mutation probability occurs here
 set mut-prob-run 100
 ;; Any changes with the birth-mut-prob should happen here
 birth-mut-action
 a-4-r-char-run]
 set performance performance - 4
 ]

;;;;;;;;;;;;;;;;;;;;;;;;
; a-5-
;;;;;;;;;;;;;;;;;;;;;;;;


   ifelse ((random-float 1000) >= a-5-n-evo-prob and not member? 5 evo-tag)
[
 hatch-a-5-n 1
```

```
    [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-5-n-char-run
      ]
  set performance performance - 4]
[
  ;; Set a record that the evolution has occurred
  ask self [set evo-tag fput 5 evo-tag]
  hatch-a-5-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-5-g-char-run]
  set performance performance - 4
  ]
if member? 5 evo-tag
[ifelse ((random-float 100) >= a-5-g-evo-prob and not member? 5 r-tag)

[hatch-a-5-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-5-g-char-run]
  set performance performance - 4
  ]
[
  ask self [set r-tag fput 5 r-tag]
  hatch-a-5-g 1
  [
  ;; Any changes with mutation probability occurs here
  set mut-prob-run 100
  ;; Any changes with the birth-mut-prob should happen here
  birth-mut-action
  a-5-g-char-run]
  set performance performance - 4

  hatch-a-5-r 1
  [
  ;; Any changes with mutation probability occurs here
```

```
   set mut-prob-run 100
   ;; Any changes with the birth-mut-prob should happen here
   birth-mut-action
   a-5-r-char-run]
   set performance performance - 4
   ]]
   if member? 5 r-tag
  [
   hatch-a-5-g 1
   [
   ;; Any changes with mutation probability occurs here
   set mut-prob-run 100
   ;; Any changes with the birth-mut-prob should happen here
   birth-mut-action
   a-5-g-char-run]
   set performance performance - 4

   hatch-a-5-r 1
   [
   ;; Any changes with mutation probability occurs here
   set mut-prob-run 100
   ;; Any changes with the birth-mut-prob should happen here
   birth-mut-action
   a-5-r-char-run]
   set performance performance - 4
   ]
end



;;; HERE IS THE ALTERED MUT-ORB RUN!!!!! ;;;
;; This is for changing the global variables mutation-gamma-dist
;; and changing the global counter for length of mutations
to birth-mut-global
if birth-mut-counter >= 1
[
;; setting new lambda conditions...?
]
if birth-mut-counter >= 6
[set birth-mut-counter 0
  ;; setting it back to normal

]
end

to birth-mut-counter-proc
```

```
  if birth-mut-counter >= 1
[set birth-mut-counter (birth-mut-counter + 1)]
end

;; This is turtle specific for all the newly produced proteins, and allows a
;; greater chance of mutations occuring within the birth mut period
to birth-mut-action
if birth-mut-counter >= 1
[set mut-prob-run 100]
if birth-mut-counter >= 6
[set mut-prob-run 0
  set birth-mut-counter 0]
end

;; High parameter = high likelihood of evo
to genome-evo
set a-1-n-evo-prob 0
set a-2-n-evo-prob 0
set a-3-n-evo-prob 0
set a-4-n-evo-prob 0
set a-5-n-evo-prob 0
end

to genome-r
set a-1-g-evo-prob 0
set a-2-g-evo-prob 0
set a-3-g-evo-prob 0
set a-4-g-evo-prob 0
set a-5-g-evo-prob 0
end
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Molecule Characteristics
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Assuming independent mutations


;; NUMBER OF AGENTS
;; Change the initial number of agents HERE
;; NOTE: If the radius is changed, please change the radius in
;;      the CHARACTERISTICS OF AGENTS functions as well!
to number-of-agents
 create-substrate-1 300
 [
 setxy random-xcor random-ycor
 set radius 3
 ]
 create-a-1-n 30
 [
 setxy random-xcor random-ycor
 set radius 3
 ]
 create-a-2-n 30
   [
 setxy random-xcor random-ycor
 set radius 3
 ]
 create-a-3-n 30
 [
 setxy random-xcor random-ycor
 set radius 3
 ]
 create-a-4-n 30
 [
 setxy random-xcor random-ycor
 set radius 3
```

```
]
create-a-5-n 30
[
setxy random-xcor random-ycor
set radius 3
]
create-substrate-2 300
[
setxy random-xcor random-ycor
set radius 3
]
create-substrate-3 300
[
setxy random-xcor random-ycor
set radius 3
]
create-substrate-4 300
[
setxy random-xcor random-ycor
set radius 3
]
create-substrate-5 300
[
setxy random-xcor random-ycor
set radius 3
]
create-a-1-g 0
[
 set radius 3
 setxy random-xcor random-ycor
]
create-a-2-g 0
[
 set radius 3
 setxy random-xcor random-ycor
]
create-a-3-g 0
[
 set radius 3
 setxy random-xcor random-ycor
]
create-a-4-g 0
[
 set radius 3
 setxy random-xcor random-ycor
]
```

```
 create-a-5-g 0
 [
  set radius 3
  setxy random-xcor random-ycor
 ]
 create-a-1-r 0
 [
  set radius 3
  setxy random-xcor random-ycor
 ]
 create-a-2-r 0
 [
  set radius 3
  setxy random-xcor random-ycor
 ]
 create-a-3-r 0
 [
  set radius 3
  setxy random-xcor random-ycor
 ]
 create-a-4-r 0
 [
  set radius 3
  setxy random-xcor random-ycor
 ]
 create-a-5-r 0
 [
  set radius 3
  setxy random-xcor random-ycor
 ]
 create-modified-substrate-1-s 0
 [
  set radius 3
  setxy random-xcor random-ycor
  ]
end

;; CHARACTERISTICS OF AGENTS
;; Change the characteristics of agents HERE
;; NOTE: If the radius is changed, please change the radius in
;;       the number-of-agents function as well!

to substrate-1-char
 ask substrate-1[
 set mass 1
 set radius 3
```

```
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "substrate-1"
  set color 4
  set shape "circle"
  ]
end

;;a-#-n

to a-1-n-char
  ask a-1-n[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  set rxn-reactant nobody
  kinetic
  set size radius
  set compound-name "a-1-n"
  set shape "square 2"
  set color 15
  set rxn-prob 60
  set performance-1 10
  ]
end

to a-2-n-char
  ask a-2-n[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-2-n"
  set shape "square 2"
  set color 25
  set rxn-prob 60
  set performance-1 -1
  ]
end

to a-3-n-char
```

```
  ask a-3-n[
  set mass 1
  set radius 4
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-3-n"
  set shape "square 2"
  set color 35
  set rxn-prob 60
  set performance-1 10
  ;;allo-state-a-3-n
  ]
end

to a-4-n-char
  ask a-4-n[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-4-n"
  set shape "square 2"
  set color 45
  set rxn-prob 60
  set performance-1 -1
  ]
end

to a-5-n-char
  ask a-5-n[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-5-n"
  set shape "square 2"
  set color 55
  set rxn-prob 60
  set performance-1 10
  ]
```

```
end

;mod-a-#-n

to substrate-2-char
  ask substrate-2[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "substrate-2"
  set shape "circle 2"
  set color 65
  ]
end

to substrate-3-char
  ask substrate-3[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "substrate-3"
  set shape "circle 2"
  set color 85
  ]
end

to substrate-4-char
  ask substrate-4[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "substrate-4"
  set shape "circle 2"
  set color 105
  ]
end
```

```
to substrate-5-char
  ask substrate-5[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "substrate-5"
  set shape "circle 2"
  set color 125
  ]
end

;; a-#-s

to a-1-g-char
  ask a-1-g[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-1-g"
  set shape "box"
  set color 15
  set rxn-prob 60
  set performance-1 10
  set performance-2 10
  ]
end

to a-2-g-char
  ask a-2-g[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-2-g"
  set shape "box"
  set color 25
  set rxn-prob 60
  set performance-1 -1
```

```
  set performance-2 -1
  ]
end


to a-3-g-char
  ask a-3-g[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-3-g"
  set shape "box"
  set color 35
  set rxn-prob 60
  set performance-1 10
  set performance-2 10
  ]
end


to a-4-g-char
  ask a-4-g[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-4-g"
  set shape "box"
  set color 45
  set rxn-prob 60
  set performance-1 -1
  set performance-2 -1
  ]
end


to a-5-g-char
  ask a-5-g[
  set mass 1
  set radius 3
  set speed init-speed
```

```
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-5-g"
  set shape "box"
  set color 55
  set rxn-prob 60
  set performance-1 10
  set performance-2 10
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Related enzymes
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to a-1-r-char
  ask a-1-r[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-1-r"
  set shape "box"
  set color 15
  set rxn-prob 60
  set performance-1 10
  ]
end

to a-2-r-char
  ask a-2-r[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-2-r"
  set shape "box"
  set color 25
  set rxn-prob 60
  set performance-1 -1
  ]
```

```
end


to a-3-r-char
  ask a-3-r[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-3-r"
  set shape "box"
  set color 35
  set rxn-prob 60
  set performance-1 10
  ]
end


to a-4-r-char
  ask a-4-r[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-4-r"
  set shape "box"
  set color 45
  set rxn-prob 60
  set performance-1 -1
  ]
end


to a-5-r-char
  ask a-5-r[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set compound-name "a-5-r"
```

```
  set shape "box"
  set color 55
  set rxn-prob 60
  set performance-1 10
  ]
end

;;;;;;;;;;;;;;;;;;
;;  Modified    ;
;;  a-#-s init   ;
;;;;;;;;;;;;;;;;;;
;; made to initiate upon the setup of the simulation, for debugging

to modified-substrate-2-char-i
  ask modified-substrate-2[
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-2"
set shape "triangle 2"
set color 105
  ]
end

to modified-substrate-3-char-i
  ask modified-substrate-3 [
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-3"
set shape "triangle 2"
set color 125]
end

to modified-substrate-4-char-i
  ask modified-substrate-4 [
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
```

```
kinetic
set size radius
  set compound-name "modified-substrate-4"
set shape "triangle 2"
set color 11]
end

to modified-substrate-5-char-i
  ask modified-substrate-5 [
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-5"
set shape "triangle 2"
set color 31]
end


to modified-substrate-1-s-char-i
  ask modified-substrate-1-s [
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-5"
set shape "triangle 2"
set color 8]
end

;;;;;;;;;;;;;;;;;;
;;  Modified     ;
;;  a-#-s    ;
;;;;;;;;;;;;;;;;;;

;; mod-a-#-s

to modified-substrate-2-char
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
```

```
kinetic
set size radius
  set compound-name "modified-substrate-2"
set shape "triangle 2"
set color 105
end

to modified-substrate-3-char
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-3"
set shape "triangle 2"
set color 125
end

to modified-substrate-4-char
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-4"
set shape "triangle 2"
set color 11
end

to mod-a-5-g-char
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "mod-a-5-g"
set shape "triangle 2"
set color 51
end

to modified-substrate-5-char
set mass 1
set radius 3
```

```
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-5"
set shape "triangle 2"
set color 31
end


to modified-substrate-1-s-char
set mass 1
set radius 3
set speed init-speed
set last-collision nobody
kinetic
set size radius
  set compound-name "modified-substrate-5"
set shape "triangle 2"
set color 8
end
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Molecule Characteristics-Genome-related
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Assuming independent mutations
;; This module holds all information about the molecules
;; produced from the genome. Thus, it holds possibilities of mutations etc.
;; and retains memory of these changes

;; CHARACTERISTICS OF AGENTS
;; Change the characteristics of agents HERE
;; NOTE: If the radius is changed, please change the radius in
;;       the number-of-agents function as well!

to substrate-1-char-run
  ask substrate-1[
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  set size 3
  kinetic
  set color 9.9
  set shape "square 2"
  set compound-name "substrate-1"
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; A-1-N Performance mutator ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; creates a mutator according to normal distribution
to a-1-n-mutation-performance-1?
;; random number generator to see if it hits the necessary probability
```

```
  if ((random-float 100) <= mut-prob-run)
;; normal distribution
  [set performance-1 random-normal a-1-n-performance-1 1]
end

to a-2-n-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-2-n-performance-1 1]
end

to a-3-n-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-3-n-performance-1 1]
end

to a-4-n-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-4-n-performance-1 1]
end

to a-5-n-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-5-n-performance-1 1]
end

to a-1-g-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-1-g-performance-1 1]
end

to a-1-g-mutation-performance-2?
  if ((random-float 100) <= mut-prob-run)
  [set performance-2 random-normal a-1-g-performance-2 1]
end

to a-2-g-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-2-g-performance-1 1]
end

to a-2-g-mutation-performance-2?
  if ((random-float 100) <= mut-prob-run)
  [set performance-2 random-normal a-2-g-performance-2 1]
end

to a-3-g-mutation-performance-1?
```

```
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-3-g-performance-1 1]
end

to a-3-g-mutation-performance-2?
  if ((random-float 100) <= mut-prob-run)
  [set performance-2 random-normal a-3-g-performance-2 1]
end

to a-4-g-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-4-g-performance-1 1]
end

to a-4-g-mutation-performance-2?
  if ((random-float 100) <= mut-prob-run)
  [set performance-2 random-normal a-4-g-performance-2 1]
end

to a-5-g-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-5-g-performance-1 1]
end

to a-5-g-mutation-performance-2?
  if ((random-float 100) <= mut-prob-run)
  [set performance-2 random-normal a-5-g-performance-2 1]
end

to a-1-r-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-1-r-performance-1 1]
end

to a-2-r-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-2-r-performance-1 1]
end

to a-3-r-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-3-r-performance-1 1]
end

to a-4-r-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
```

```
  [set performance-1 random-normal a-4-r-performance-1 1]
end

to a-5-r-mutation-performance-1?
  if ((random-float 100) <= mut-prob-run)
  [set performance-1 random-normal a-5-r-performance-1 1]
end

to a-1-n-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  set rxn-reactant nobody
  kinetic
  set size radius
  set shape "square 2"
  set color 15
  set rxn-prob 60
  a-1-n-mutation-performance-1?
  set compound-name "a-1-n"

end

to a-2-n-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color brown
  set rxn-prob 60
  a-2-n-mutation-performance-1?
  set compound-name "a-2n"

end

to a-3-n-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
```

```
  set shape "square 2"
  set color orange
  set rxn-prob 60
  a-3-n-mutation-performance-1?
  set compound-name "a-3-n"
end


to a-4-n-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color 45
  set rxn-prob 60
  a-4-n-mutation-performance-1?
  set compound-name "a-4-n"
  ;;allo-state-a-3-n
end


to a-5-n-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color 55
  set rxn-prob 60
  a-5-n-mutation-performance-1?
  set compound-name "a-5-n"
  ;;allo-state-a-3-n
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Gain-of-function enzymes
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to a-1-g-char-run
  set mass 1
  set radius 3
```

```
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "box"
  set color 15
  set rxn-prob 60
  a-1-g-mutation-performance-1?
  a-1-g-mutation-performance-2?
  set compound-name "a-1-g"
  ;;allo-state-a-3-n
end

to a-2-g-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "box"
  set color 25
  set rxn-prob 60
  a-2-g-mutation-performance-1?
  a-2-g-mutation-performance-2?
  set compound-name "a-2-g"
  ;;allo-state-a-3-n
end

to a-3-g-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "box"
  set color 35
  set rxn-prob 60
  a-3-g-mutation-performance-1?
  a-3-g-mutation-performance-2?
  set compound-name "a-3-g"
  ;;allo-state-a-3-n
end

to a-4-g-char-run
```

```
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "box"
  set color 45
  set rxn-prob 60
  a-4-g-mutation-performance-1?
  a-4-g-mutation-performance-2?
  set compound-name "a-4-g"
  ;;allo-state-a-3-n
end

to a-5-g-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "box"
  set color 55
  set rxn-prob 60
  a-5-g-mutation-performance-1?
  a-5-g-mutation-performance-2?
  set compound-name "a-5-g"
  ;;allo-state-a-3-n
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; A-#-r run
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to a-1-r-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  set rxn-reactant nobody
  kinetic
  set size radius
  set shape "square 2"
  set color 15
  set rxn-prob 60
```

```
  a-1-r-mutation-performance-1?
  set compound-name "a-1-r"

end


to a-2-r-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color brown
  set rxn-prob 60
  a-2-r-mutation-performance-1?
  set compound-name "a-2r"

end

to a-3-r-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color orange
  set rxn-prob 60
  a-3-r-mutation-performance-1?
  set compound-name "a-3-r"
end


to a-4-r-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color 45
  set rxn-prob 60
```

```
  a-4-r-mutation-performance-1?
  set compound-name "a-4-r"
  ;;allo-state-a-3-n
end


to a-5-r-char-run
  set mass 1
  set radius 3
  set speed init-speed
  set last-collision nobody
  kinetic
  set size radius
  set shape "square 2"
  set color 55
  set rxn-prob 60
  a-5-r-mutation-performance-1?
  set compound-name "a-5-r"
  ;;allo-state-a-3-n
end


to global-mutator-means
set a-1-n-performance-1 15
set a-2-n-performance-1 -1
set a-3-n-performance-1 15
set a-4-n-performance-1 -1
set a-5-n-performance-1 15

set a-1-g-performance-1 15
set a-2-g-performance-1 -1
set a-3-g-performance-1 15
set a-4-g-performance-1 -1
set a-5-g-performance-1 15

set a-1-g-performance-2 15
set a-2-g-performance-2 -1
set a-3-g-performance-2 15
set a-4-g-performance-2 -1
set a-5-g-performance-2 15


set a-1-r-performance-1 15
set a-2-r-performance-1 -1
set a-3-r-performance-1 15
set a-4-r-performance-1 -1
```

```
set a-5-r-performance-1 15
end




;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Implicit probabilities
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; Implicit probabilities refer to the likelihood of agents disappearing or
; appearing within the simulation based on randomness to reflect flux
; or side-reactions

;; Implicit-gone reflects molecules disappearing from simulation
;; Implicit-here reflects molecules appearing in simulation

;; Low probability-parameter means low likelihood of the event occuring

to implicit-probability
set  implicit-prob-gone-a-1-n 0
set  implicit-prob-gone-a-2-n 0
set  implicit-prob-gone-a-3-n 0
set  implicit-prob-gone-a-4-n 0
set  implicit-prob-gone-a-5-n 0
set  implicit-prob-gone-substrate-2 0
set  implicit-prob-gone-substrate-3 0
set  implicit-prob-gone-substrate-4 0
set  implicit-prob-gone-substrate-5 0
set  implicit-prob-gone 0
set  implicit-prob-gone-sub-s 0

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Evolved molecules         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

set implicit-prob-here-a-1-g 0
set implicit-prob-here-a-2-g 0
set implicit-prob-here-a-3-g 0
set implicit-prob-here-a-4-g 0
set implicit-prob-here-a-5-g 0
set implicit-prob-here-a-1-r 0
set implicit-prob-here-a-2-r 0
```

```
set implicit-prob-here-a-3-r 0
set implicit-prob-here-a-4-r 0
set implicit-prob-here-a-5-r 0

;;;;;;;;
; end  ;
;;;;;;;;

set implicit-prob-here-a-1-n 0
set implicit-prob-here-a-2-n 0
set implicit-prob-here-a-3-n 0
set implicit-prob-here-a-4-n 0
set implicit-prob-here-a-5-n 0
set implicit-prob-here-substrate-2 0
set implicit-prob-here-substrate-3 0
set implicit-prob-here-substrate-4 0
set implicit-prob-here-substrate-5 0
set implicit-prob-here 0
set implicit-prob-here-sub-s 0

end

;;;;;;;;;;;;;;;;;;
;; a-#-n
;;;;;;;;;;;;;;;;;;

;; Implicit Sub
to implicit-interaction-gone-sub
;; random number generator for probability
 if ((random-float 100)
   <= implicit-prob-gone) and not (one-of substrate-1 = nobody)
;; kills off and removes the molecules
  [ask one-of substrate-1 [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-sub
  if ((random-float 100) <= implicit-prob-here) and not (one-of substrate-1 =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-substrate-1 10
       [substrate-1-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end
```

```
;; Implicit a-1-n
to implicit-interaction-gone-a-1-n
 if ((random-float 100)
   <= implicit-prob-gone-a-1-n) and not (one-of a-1-n = nobody)
  [ask one-of a-1-n [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-1-n
  if ((random-float 100) <= implicit-prob-here-a-1-n) and not (one-of a-1-n =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-1-n 10
       [a-1-n-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-2-n
to implicit-interaction-gone-a-2-n
 if ((random-float 100)
   <= implicit-prob-gone-a-2-n) and not (one-of a-2-n = nobody)
  [ask one-of a-2-n [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-2-n
  if ((random-float 100) <= implicit-prob-here-a-2-n) and not (one-of a-2-n =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-2-n 10
       [a-2-n-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-3-n
to implicit-interaction-gone-a-3-n
 if ((random-float 100)
   <= implicit-prob-gone-a-3-n) and not (one-of a-3-n = nobody)
  [ask one-of a-3-n [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-3-n
```

```
  if ((random-float 100) <= implicit-prob-here-a-3-n) and not (one-of a-3-n =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-3-n 10
       [a-3-n-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-4-n
to implicit-interaction-gone-a-4-n
 if ((random-float 100)
   <= implicit-prob-gone-a-4-n) and not (one-of a-4-n = nobody)
  [ask one-of a-4-n [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-4-n
  if ((random-float 100) <= implicit-prob-here-a-4-n) and not (one-of a-4-n =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-4-n 10
       [a-4-n-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end
;; Implicit a-5-n
to implicit-interaction-gone-a-5-n
 if ((random-float 100)
   <= implicit-prob-gone-a-5-n) and not (one-of a-5-n = nobody)
  [ask one-of a-5-n [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-5-n
  if ((random-float 100) <= implicit-prob-here-a-5-n) and not (one-of a-5-n =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-5-n 10
       [a-5-n-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;;;;;;;;;;;;;;;;;
;; a-#-s
```

```
;;;;;;;;;;;;;;;;;

;; Implicit a-1-g
to implicit-interaction-gone-a-1-g
 if ((random-float 100)
   <= implicit-prob-gone-a-1-g) and not (one-of a-1-g = nobody)
  [ask one-of a-1-g [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-1-g
  if ((random-float 100) <= implicit-prob-here-a-1-g) and not (one-of a-1-g =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-1-g 10
       [a-1-g-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-2-g
to implicit-interaction-gone-a-2-g
 if ((random-float 100)
   <= implicit-prob-gone-a-2-g) and not (one-of a-2-g = nobody)
  [ask one-of a-2-g [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-2-g
  if ((random-float 100) <= implicit-prob-here-a-2-g) and not (one-of a-2-g =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-2-g 10
       [a-2-g-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-3-g
to implicit-interaction-gone-a-3-g
 if ((random-float 100)
   <= implicit-prob-gone-a-3-g) and not (one-of a-3-g = nobody)
  [ask one-of a-3-g [die]
    set performance performance - 1]
  stop
end
```

```
to implicit-interaction-here-a-3-g
  if ((random-float 100) <= implicit-prob-here-a-3-g) and not (one-of a-3-g =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-3-g 10
       [a-3-g-char
        setxy random-pxcor random-pycor]
   set performance performance + 1]
end

;; Implicit a-4-g
to implicit-interaction-gone-a-4-g
 if ((random-float 100)
   <= implicit-prob-gone-a-4-g) and not (one-of a-4-g = nobody)
  [ask one-of a-4-g [die]
   set performance performance - 1]
  stop
end
to implicit-interaction-here-a-4-g
  if ((random-float 100) <= implicit-prob-here-a-4-g) and not (one-of a-4-g =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-4-g 10
       [a-4-g-char
        setxy random-pxcor random-pycor]
   set performance performance + 1]
end

;; Implicit a-5-g
to implicit-interaction-gone-a-5-g
 if ((random-float 100)
   <= implicit-prob-gone-a-5-g) and not (one-of a-5-g = nobody)
  [ask one-of a-5-g [die]
   set performance performance - 1]
  stop
end
to implicit-interaction-here-a-5-g
  if ((random-float 100) <= implicit-prob-here-a-5-g) and not (one-of a-5-g =
nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-5-g 10
       [a-5-g-char
        setxy random-pxcor random-pycor]
   set performance performance + 1]
end
```

```
;;;;;;;;;;;;;;;;;
;; a-#-r
;;;;;;;;;;;;;;;;;


;; Implicit a-1-r
to implicit-interaction-gone-a-1-r
 if ((random-float 100)
   <= implicit-prob-gone-a-1-r) and not (one-of a-1-r = nobody)
  [ask one-of a-1-r [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-1-r
  if ((random-float 100) <= implicit-prob-here-a-1-r) and not (one-of a-1-r = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-1-r 10
       [a-1-r-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-2-r
to implicit-interaction-gone-a-2-r
 if ((random-float 100)
   <= implicit-prob-gone-a-2-r) and not (one-of a-2-r = nobody)
  [ask one-of a-2-r [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-2-r
  if ((random-float 100) <= implicit-prob-here-a-2-r) and not (one-of a-2-r = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-2-r 10
       [a-2-r-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-3-r
to implicit-interaction-gone-a-3-r
 if ((random-float 100)
   <= implicit-prob-gone-a-3-r) and not (one-of a-3-r = nobody)
  [ask one-of a-3-r [die]
    set performance performance - 1]
```

```
  stop
end
to implicit-interaction-here-a-3-r
  if ((random-float 100) <= implicit-prob-here-a-3-r) and not (one-of a-3-r = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-3-r 10
       [a-3-r-char
        setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-4-r
to implicit-interaction-gone-a-4-r
 if ((random-float 100)
   <= implicit-prob-gone-a-4-r) and not (one-of a-4-r = nobody)
  [ask one-of a-4-r [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-4-r
  if ((random-float 100) <= implicit-prob-here-a-4-r) and not (one-of a-4-r = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-4-r 10
       [a-4-r-char
        setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; Implicit a-5-r
to implicit-interaction-gone-a-5-r
 if ((random-float 100)
   <= implicit-prob-gone-a-5-r) and not (one-of a-5-r = nobody)
  [ask one-of a-5-r [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-a-5-r
  if ((random-float 100) <= implicit-prob-here-a-5-r) and not (one-of a-5-r = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-a-5-r 10
       [a-5-r-char
        setxy random-pxcor random-pycor]
    set performance performance + 1]
end
```

```
;;;;;;;;;;;;;;;;
;; Modified products
;;;;;;;;;;;;;;;;

;; Implicit substrate-2
to implicit-interaction-gone-substrate-2
 if ((random-float 100)
   <= implicit-prob-gone-substrate-2) and not (one-of substrate-2 = nobody)
  [ask one-of substrate-2 [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-substrate-2
  if ((random-float 100) <= implicit-prob-here-substrate-2) and not (one-of
substrate-2 = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-substrate-2 10
       [substrate-2-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end
;; Implicit substrate-3
to implicit-interaction-gone-substrate-3
 if ((random-float 100)
   <= implicit-prob-gone-substrate-3) and not (one-of substrate-3 = nobody)
  [ask one-of substrate-3 [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-substrate-3
  if ((random-float 100) <= implicit-prob-here-substrate-3) and not (one-of
substrate-3 = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-substrate-3 10
       [substrate-3-char
         setxy random-pxcor random-pycor]
    set performance performance + 1]
end
;; Implicit substrate-4
to implicit-interaction-gone-substrate-4
 if ((random-float 100)
   <= implicit-prob-gone-substrate-4) and not (one-of substrate-4 = nobody)
  [ask one-of substrate-4 [die]
    set performance performance - 1]
  stop
```

```
end
to implicit-interaction-here-substrate-4
  if ((random-float 100) <= implicit-prob-here-substrate-4) and not (one-of
substrate-4 = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-substrate-4 10
       [substrate-4-char
        setxy random-pxcor random-pycor]
    set performance performance + 1]
end
;; Implicit substrate-5
to implicit-interaction-gone-substrate-5
 if ((random-float 100)
   <= implicit-prob-gone-substrate-5) and not (one-of substrate-5 = nobody)
  [ask one-of substrate-5 [die]
    set performance performance - 1]
  stop
end
to implicit-interaction-here-substrate-5
  if ((random-float 100) <= implicit-prob-here-substrate-5) and not (one-of
substrate-5 = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-substrate-5 10
       [substrate-5-char
        setxy random-pxcor random-pycor]
    set performance performance + 1]
end

;; sub-s
to implicit-interaction-here-sub-s
  if ((random-float 100) <= implicit-prob-here-substrate-2) and not (one-of
substrate-2 = nobody)
  ;; Must add the molecule characteristic here so it has characteristics after hatching
  [create-modified-substrate-1-s 10
       [modified-substrate-1-s-char
        setxy random-pxcor random-pycor]
    set performance performance + 1]
end
to implicit-interaction-gone-sub-s
 if ((random-float 100)
   <= implicit-prob-gone-sub-s) and not (one-of modified-substrate-1-s = nobody)
  [ask one-of modified-substrate-1-s [die]
    set performance performance - 1]
  stop
end
```