**Ideas: could possibly get rid of IDs altogether from hosts column or give all same ID (e.g., 1). Currently 1.9999 is an ID if lots of offspring created – this may slow down the model and increase memory usage.**

**Note: 2000 times steps in infection (~ 60 days, once per hour or something like that ) – near continuous…**

**Marked increase in heterozygoisty corresponds with onset of disease epidemic?**

**Changel log for 2.8**

1. Modified Susceptibilities’ to allow for a normal distribution. Will also add in Chi square or other leptokurtic distribution

**Change log for 2.7**

1. Changed Reproduction.R – decreased the standard deviation substantially to prevent normal distribution of genotypes from occurring to quickly (now much more uniform).
2. Version 2.6 never worked correctly - all infected individuals were immediately purged because all individuals had unique susceptibility (and the ChytridMortality was rounded down to the nearest individual). Chytrid mortality was rewritten to now sample each infected individual by its survival probability. If survival prob is 0.9, then 90% of the time that individual will survive – of course there is only a single trial for each individual – as would occur in the wild. Survivors are appended to hosts (and keep their infection). Those that die are removed from hosts object.

**Change log for 2.6**

1. Updated reproduction.R such that half the offspring get a susceptibility equal to their mom and half get a susceptibility equal to their dad (as before, expect that rather than get the exact susceptibility, it is pulled from a normal distribution that centers (has a mean, with 0.05 std dev) around either their mom or dad. This procedure prevents susceptibility from being completely fixed by the 50th generation (which is what happened in 2.5). Should not have any effect on neutral loci.
2. Also added proportion of polymorphic loci to output file.

**Change log for 2.5**

1. This version was updated to run faster by not sending the entire hosts object to Infection3.R. Only the three relevant columns are sent and matched back into hosts after infection is completed. Both Infection3.R and RunModel.R were edited to implement this change. System time reports a reduction in time from 8 seconds to ~ 0.7 seconds. Model is still slower than 2.2.
2. A bug was noted in 2.4 that resulted in the program crashing, likely because the number of columns in the host object was increased (with genotypes), but the if clauses in RunModel.R were not correspondingly adjusted. A fix was implemented for both 2.4 and 2.5.
3. Updated Chytrid.Adults.R to not take the ceiling when sampling for chytrid mortality. This kept chytrid in the system much longer (allowed for successful infection, but created weird artifact where at high chytrid mortality there was always one infected adult left over, when all infected adults should have died.
4. Updated Output.R to include expected heterozygosity as well.

**Change log for 2.4**

1. A substantial change to the model. The “hosts” object was modified to included multilocus genotypes for every individual (along with their susceptibilities). This change was implemented to measure “heterozygosity” directly, rather than “number of unique genotypes”, which declined through time at too fast of a rate.
2. Asexual reproduction was replaced with sexual reproduction. Half the individuals are chosen to be male and half are chosen to be female. Each individual has only 1 mate in a season. If the adult survives the season it may reproduce the next year (and it may also be assigned to a different sex). The genotypes of the two individual in the pair are aligned and each offspring is created iteratively, where one allele is randomly selected from the mother and one allele is randomly selected from the father at each locus. Each pair produces n.eggs worth of offspring (currently set to 11). Because there are 50 SNPs most offspring have completely unique genotypes. In terms of susceptibilities, each offspring is randomly assigned either the susceptibility from it’s mother or the susceptibility from it’s father (an average would drive all to the mean; taking the highest would inflate speed of evolutionary response; if there is no variation in susceptibility then this does not matter at all).
3. **Could modify output to spit out genotypic data at set intervals to calculate Fst between replicate populations.**

**Change log for 2.3**

1. Used as an intermediary to incorporating sexual reproduction and full genotypes. Never completely finished, but kept structure of original model, where genotypes were listed as numbers and kept separate from “hosts” objects. This became cumbersome and 2.4 was created to add each individual genotype to “hosts”.

**Change log for 2.2**

1. Fixed Metamorphosis.R to NOT sample with replacement, so that 100% of tads can metamorph in a single season. Had to also make some changes to RunModel.R to not break too early.
2. Changed host.suscepts == 1.1 to skip infection step (controls) as Cat suggested that 100% survival is a realistic possibility.

**Change log for 2.1**

1. No longer running “no disease” models. If host.suscepts == 1, then the infection step is simply skipped.

Code is if(n > 49 & mean(suscpetibilities[, 2]) == 1) {

[in RunModel.R]

Note that if variation in susceptibilities is introduced, then this wont work UNLESS additional if statements are added to the creation of variable susceptibilities (or ranges are used: range from 1 to 1).

**Model steps:**

1. Set the following variables:
   * **plength, pwidth** (Set dimensions of pond (length and width, here a rectangle)
   * **n.species** (Set number of species (fixed at 1 for this paper)
   * **k.tads , k.adults** (carrying capacity of tads and adults, respectively)
   * **n.tads, n.adults** (starting population sizes of tads and adults, respectively; usually set to k.tads, k.adults)
   * **adult.survival.var** (density independent variation in adults)
   * **tadpole.survival.var** (density independent variation in tadpoles above and beyond that of the adults; usually set to 0)
   * **proportion metamorph** (the proportion of tadpoles that metamorph each year; chosen randomly – to be changed in 2.0)
   * **r** (intrinsic rate of population growth, classic r from ecology)
   * **days** (not really days, but rather then number of discrete steps each time an infection event occurs; should really be continuous, but difficult to model: currently set at 2000)
   * **threshold** (the threshold for disease dispersal, higher == more dispersal)
   * **host.suscepts** (NOT susceptibility, rather the percent mortality due to chytrid)
   * **range.suscepts** (the within population range of host susceptibilities, modeled with a uniform distribution)
   * **n.generations** (total number of generations; number of generations pre-chytrid is set in RunModel.R)
   * **n.replicates** (number of replicates for each parameter set)
   * **n.genotypes** (number of initial unique genotypes)
   * **n.eggs** (number of eggs produced per frog) (does each frog produce eggs)
2. *Replicates*: creates replicates of parameter sets. Certain variables are paired with each other, such that not all combinations are created. The ith element in k.adults is always paired with the ith element in k.tads.
3. *PondSetup*: sets up Cartesian space for model run, initializes positions of individuals, outputs a **hosts** data frame, with the following columns: [1] individual, [2] species, [3] genotype, [4] stage (tadpole=1; adult=2), [5] infected (no=1; yes=2), [6] x position, y position. All individuals are initially uninfected.
4. *Suscpetibilities*: creates susceptibilities by sampling from host.suscepts-range.suscepts to host.suscepts+range.suscepts. Stored in suscpetibilities data frame where col 1 = genotype and col 2 = susceptibility. If range.suscepts = 0, then all values = host.suscepts. Genotypes are initially split equally across all individuals (tads + adults). This means that genotypes are NOT split equally among ADULTS at the beginning of a run, but rather among adults and tadpoles. This creates variation in genotype frequency of adults (and tads to a lesser extent) at the beginning of a run. This variation will be higher at smaller adult population sizes relative to tadpole population sizes. This feature could be modified, but we decided that it would be more realistic to have variation in genotype frequency than to have equal genotype frequencies.
5. Begin iterating through n.generations.
6. Isolate adults and count number of current adults (new n.adults)
7. Calculate number of adults in next generation (Nt1) with following equation:

* Nt1 = n.adults + (r x n.adults x (1 – (n.adults/k.adults)))

1. *Reproduction*: Reproduction is frequency dependent based on the frequency of each genotype in the adults in the current generation. Creates n.adults (# current adults) x n.eggs new tadpoles and adds to *hosts*.
2. *HostDispersal*: Randomly shuffles both tadpoles and adults throughout the pond by randomly sampling from a near continuous distribution of x and y values and adds new positions to *hosts*. # must occur because new tads assigned 0,0 initially.
3. *WinterTadpoles*: Separates adults and tadpoles from hosts. Creates winter tadpole mortality by subtracting k.tads from n.tads and adding additional variation by including tadpole.survival.var (sampled from a normal distribution). Randomly samples tadpoles to reach desired tadpole mortality. Survivors remain in hosts. Dead tadpoles are removed from hosts. Number of surviving tads given by:
   1. *N*.tads = (n.tads-tadpole.mortality)\*(n.adults/k.adults)
4. If n.generations = 50, then a single infected individual is introduced into the population (a tadpole).
5. *Infection3*: If n.generations >= 50. For d in days: dispersal, creates huge data frame where all x and y positions are sampled and recorded for x days. Places a single sensor in the middle of the pond. Next calculates absolute value of distances from all individuals (at every point in time) to the sensor. Next calculates which individuals (x and y coordinates) were <= the threshold distance to the sensor; called contacts for the number of contacts to sensor. Calculates total number of contacts and multiplies that by the total number of infected individuals (prior to calling this function) and creates that many new infected individuals. (e.g., if the sensor says that there were 10 contacts and there are 10 infected individuals, then there will be 100 newly infected individuals. In short, this function creates one infected individual that always sits right in the middle of the pond as other individuals move past it in space and time. We then calculate how many individuals would have gotten infected by this one individual and scale up to all the infected individuals already in the population. We do this because it is MUCH more computationally tractable.
6. *AdultEmmigration*: splits hosts into adults and tadpoles (called hosts). Simulates migration of adults out of the pond.
7. *Metamorphosis*: Sample tadpoles (randomly) by proportion.metamorph. Convert sampled tadpoles to adults. Remove new adults from tadpoles. New adults DO NOT clear infection (can easily be changed within this function).
8. *WinterAdults*: Create winter adult mortality via: rnorm(1, (n.adults-Nt1), adult.survival.var): see step 7 for Nt1; adult.survival.var introduces density independent mortality. Sample randomly for survivors. This could result in some old adults by chance, but this probably realistic correct?
9. *ChytridAdults*: If n in n.generations >=50 then chytrid mortality occurs on adults. Infecteds are matched to their genotype-specific mortalities. Mortality of infected occurs and depends on “susceptibility”. Chytrid mortality does not depend on age of the adult.[could stand one more debug].
10. Adults and tadpoles are combined again (to create new hosts data frame)
11. *Output*: output is collected for a wide range of responses. See Output.R
12. End of generation. Repeat until n.generations is reached or one of many if statements recognizes population extinction.

**Note**: When analyzing the data, there may be scenarios where an infected individual was introduced, but infection never took off. Should these scenarios be analyzed separately?

How would this proportion metamorphing look like? Sexual reproduction? Realistic values of r?