**Parameters** of the model are in bold. *Variables* (vectors/matrices) are in italics.

Step 1. Seed the model at the recruitment (pre-breeder) stage, just prior to competition for limited breeding sites

* Simulate **N\_local** individuals. The total number of individuals **N** is then **N = N\_local/(1-F)**, where **F** is the fraction that are non-local. We round **N** to a whole number, as **F** can take any value between 0 and 1. The number of nonlocal individuals is then **N\_nonlocal** = **N – N\_local**.
* Assign unique identifiers 1:**N** to these individuals. Also define a vector called **local** of length 1:**N**, with TRUEs for the local individuals (first chunk) and FALSEs for the non-locals (second chunk).
* Next we set up 3 different genotype matrices, where rows hold individuals and columns hold loci. Possible allelic values for each cell are 0 and 1.
  + The first genotype matrix corresponds to the loci that uniquely affect *ZSOFT* (*Genotypes-soft\_unique*)
  + The second genotype matrix corresponds to loci that uniquely affect *ZHARD* (*Genotypes\_hard\_unique*),
  + The third genotype matrix corresponds to loci that are shared between both traits (i.e., affect both; *Genotypes\_both*). Note that, in the Evolutionary Applications paper (Reed et al. 2024), we assume no shared loci between the traits (i.e. the option of a genetic correlation is turned off).
* We then define a parameter for the desired total number of loci affecting each quantitative trait, **num\_loci\_total** (assumed to be the same for *ZSOFT* and *ZHARD*). We then define a parameter for the number of shared loci, **num\_loci\_shared**, that affect both traits (overlap between them). This allows us to calculate the derived parameter, **num\_loci\_unique = num\_loci\_total - num\_loci\_shared**, i.e., the number of loci uniquely affecting each trait.
  + So if we, for example, set **num\_loci\_total** = 30 and **num\_loci\_shared** =15, we then have 15 loci that uniquely affect each trait, and 15 that overlap between them (leading to a genetic correlation).
  + If we instead wanted completely uncorrelated traits, we simply set **num\_loci\_shared** to 0.
* The number of rows for each of the above genotype matrices equals **N** (initial number of individuals).
  + The number of columns equals **2\*num\_loci\_unique** for the *Genotypes\_soft\_unique* and *Genotypes\_hard\_unique*matrices.
  + The number of columns equals **2\*num\_loci\_shared** for the *Genotypes\_both* matrix.
  + We multiply by 2 in each case because every individual carries two alleles per diploid locus. So the first two columns in each matrix correspond to Locus 1 (allele from inherited from “mum” and allele inherited from “dad”); columns 3 and 4 correspond to Locus 2…. etc.
* We now populate these genotype matrices with 1s and 0s.
  + In doing so, we need to distinguish between local individuals and nonlocals, as the allele frequencies can be different for each.
  + For the loci uniquely affecting the soft trait, we first define the total number of alleles across all local individuals and all loci, calculated as **N\_local\*num\_loci\_unique\*2**
  + To draw the actual alleles (1s and 0s), we draw from a random uniform distribution bounded by 0 and 1. If the number drawn is less than **p\_local\_soft\_unique** (the desired initial allele frequency for locals for the loci only affecting the soft trait), the allele = 0, and if greater than, the allele = 1.
  + We then temporarily store these alleles in a matrix with **N\_local** rows and **num\_loci\_unique\*2** columns, for later merging with the alleles from nonlocals.
  + We do the same now for the nonlocals, based on the **p\_nonlocal\_soft\_unique** parameter (the desired initial allele frequency for nonlocals for the loci only affecting the soft trait)
  + We store these alleles in a temporary matrix with **N\_nonlocal** rows and **num\_loci\_unique\*2** columns, and then we merge by rows (rbind) this matrix with the above one for the locals, to produce the *Genotypes\_soft\_unique* matrix.
  + Now we do the same for the loci unique to the hard trait, based on the (desired initial allele frequency) parameters **p\_local\_hard\_unique** and **p\_nonlocal\_hard\_unique**.
  + Then we do the same for loci that are shared between the soft and hard traits, based on the (desired initial allele frequency) parameters **p\_local\_shared** and **p\_nonlocal\_shared**.
  + We now have three matrices populated with 1s and 0s: *Genotypes\_soft\_unique*, *Genotypes\_hard\_unique* and *Genotypes\_both*
* Next we column-bind the *Genotypes\_soft\_unique* and the *Genotypes\_both* matrices to generate an overall *Genotypes\_soft* matrix of genotypes affecting *ZSOFT.* Likewise, we bind *Genotypes\_hard\_unique* and *Genotypes\_both* to generate an overall *Genotypes\_hard* matrix of genotypes affecting *ZHARD*.
* On top of these, we create an additional genotype matrix to store alleles for the neutral trait (*Genotypes\_neutral*). The number of rows equals **N**, and the number of columns equals 2, as we assume only a single bi-allelic neutral locus.
  + This neutral locus is diagnostic with respect to local versus non-local, so local individuals are all assigned 0/0 genotypes and non-locals are all assigned 1/1 genotypes. The frequency of the 1 allele at this neutral marker in subsequent generations then allows us to keep track of introgression. Note that in the Evolutionary Applications paper (Reed et al. 2024), we do not present any results in relation to this single neutral locus, as it was deemed surplus to needs for that paper.
* Before proceeding further, we need to define the initial (expected) additive genetic variances (VA), as the environmental variances are calculated based on these and the initial heritability (**h2­\_init**, assumed to be same for all traits and provenances). The formula for VA in cases like ours where every locus is assumed to contribute equally to the trait (see Kardos et al.) is VA = 2n\*a2\*p\*(1-p), where n = the number of loci, a = the additive genetic effect per locus, and p = the allele frequency. So we can apply this formula to calculate 6 different VA values, for: locals soft unique, nonlocals soft unique, locals hard unique, non-locals hard unique, locals shared, and nonlocal shared.
  + Total VA for the soft trait is then given as the sum of the VA contributed by the unique loci and VA contributed by the shared loci. Same goes for the hard trait.
  + The environmental variance, V­E, for each is then computed as VA/h2 - VA
  + This environmental variance is assumed to be constant across the simulations, and in generation two, every new offspring is by definition local. So from generation 2 onwards, we always draw environmental deviations for the soft and hard traits from VE(soft\_local) and VE(hard\_local), respectively.

*FROM THIS POINT ON, WE START LOOPING OVER GENERATIONS. SO THE ABOVE BITS WERE ALL ABOUT SEEDING THE MODEL IN GENERATION 1.*

* We first build in an if statement that says “only run the remaining code (for the remaining generations in the loop) if the current pop size is >0”. If the pop has gone extinct, the loop stops and the remaining output metrics get an NA.
* The next step is to calculate the genotypic value (‘additive genetic merit’) of each (local or non-local) individual. This is done as follows, for *ZSOFT* and *ZHARD* separately:
  + Sum the number of 1 (trait-increasing) alleles across all loci for each individual.
  + For the soft-selected trait, multiply this sum by **a\_soft**, the additive effect for the soft trait (assumed to be the same for all loci and for each provenance).
  + For the hard-selected trait, multiply the corresponding genotypic values (sum of 1 alleles) by **a\_hard**, the additive effect for the soft trait (assumed to be the same for all loci and for each provenance).
  + So we end up with an **N** x 2 matrix of genotypic values **G**, where the rows are individuals and columns are the two quantitative traits. The first **NF** rows are non-locals, whilst the second **N(1-F)** rows are locals (only relevant in generation 1; thereafter everyone is a local).
  + The expected initial mean of these genotypic values for each group (provenance) equals n\*2\*a\*p, where n = number of loci affecting the trait, a = the additive effect, and p = initial allele frequency.
  + The expected initial variance of these genotypic values for each group (provenance), as already stated above, equals n\*2\*(a^2)\*p\*(1-p).
  + Thus if the additive effect (a) parameter is higher, both the mean and the variance will be higher, but the evolvability (variance ÷ mean^2) will be the same. This is the way Kardos et al. did it.
* Now we rescale the individual **G** values for each trait by subtracting off the initial mean expected if the population were composed (in generation 1) entirely of local individuals.
  + This requires first computing the average allele frequency for locals (for each trait) across the unique and shared loci, and using that to compute the expected mean genotypic value as per above equation (n\*2\*a\*p).
  + We do this for both the soft and hard traits, and subtract these references values off the current genotypic value of each individual, such that a value of 0 now corresponds to the reference value in generation 1. Any subsequent evolution will lead to a deviation away from 0 therefore.

* Now we draw environmental deviations for each individual for each quantitative trait and add them to the genotypic values (additive genetic merit) to arrive at the phenotypic value for that individual.
  + These environmental deviations are, by definition, centred on a mean of zero
  + Their variance equals the environmental variance**.**
  + So we can now draw our environmental deviations for the soft-selected trait for locals and non-locals from normal distributions of mean 0 and variances **VE\_locals\_soft** and **VE\_nonlocals\_soft**, respectively.
  + And now we can define our phenotypic values for *ZSOFT* as the sum of the genotypic value and environmental deviation.
  + We do the same for the hard selected trait.

*Step 2: Subject these initial individuals to soft selection*

* Rank the individuals from top to bottom based on their soft-selected trait, *ZSOFT*
* Define **K** as the number of breeding sites (habitat slots).
* If the number of pre-breeders > **K**, then select only the top **K** individuals from the right-tail of the *ZSOFT*distribution (based on their ranks). Kill off the other individuals.
* If the number of pre-breeders < **K**, then everyone gets a breeding site.
* In scenarios where soft selection is turned off (via the parameter **soft\_switch**, which is either “off” or “on”), just randomly select **K** breeders if the number of pre-breeders is > **K**.

*Step 3: Random mating and reproduction*

* We do not distinguish males and females, and we assume random hermaphroditic mating based on a classic Wright-Fisher model. This is pretty common in evolutionary models, and shouldn’t cause problems with reviewers, even though we notionally model a salmonid (not much to be gained by modelling separate sexes, if we don’t model any sexual dimorphism in our traits of interest). So each individual has an equal chance of becoming a parent, and each individual can produce more than 1 offspring (or no offspring). This guarantees an approximately Poisson distribution of offspring number per parent.
* The total number of offspring to be produced equals *Nb*\***k**, where *Nb* is the current number of breeders (*Nb*<=K) and **k** is fecundity (number of offspring per parent).
* Set up 5 new empty matrices to store the offspring genotypes for each quantitative trait (separate matrices for the “unique” loci for each trait, plus a matrix for the “shared” loci), plus one for the neutral trait. So the number of rows equal to the total number of offspring (*Nb*\***k**), and the number of columns equals twice the number of loci for each case.
* To generate the first offspring, randomly draw its two parents from the vector of candidate parent IDs. Then, for each locus, flip a coin to choose which allele each parent passes on to that offspring. Remember, the first two columns of the parental genotype matrices for the quantitative traits correspond to the two alleles at locus 1, the next two columns correspond to the two alleles at locus 2, etc.
* The way we actually implement the above bit in the model is using the GAMETES function, which randomly chooses n alleles (where n=number of loci; one allele drawn per locus) and packages them together into *Nb*\***k** haploid gametes (where *Nb*\***k** = the desired number of offspring). We then combine these gametes to simulate fertilisations, such that for each locus, each new offspring gets a gamete from the “mum” and a gamete from the “dad” (but remember true sexes are not modelled).
* Repeat this for all offspring, until the offspring genotype matrices have been populated with 1s and 0s (the inherited alleles).

*Step 4: Subject the offspring to hard selection*

* Now generate the phenotypes for Z­HARD for each offspring individual by computing its genotypic value (as per above) and adding an environmental deviation drawn from **VE\_locals\_hard** (as per above).
  + Note that all individuals now in this next generation are, by definition, locals, because they were born in the single wild environment considered.
  + Also note that we fix **VE\_locals\_hard** at its initial value as per Generation 1. This means that the realised heritability each generation might deviate from the initial 0.5 input value, because the realised genetic variance each generation will change owing to the level of introgression of non-local alleles, and erosion of genetic variance over time owing to selection and drift, whilst the (expected) environmental variance each generation will be constant.
* Now run the offspring through the hard-selection filter. So the expected survival *Wi* each individual is a function of its phenotype *ZHARD*, via a Gaussian fitness function defined by 3 parameters: *Wi* = **W­\_max**\*exp[-(*ZHARD* – **Theta**)/(2**\*Omega**2)]
  + **Theta** is the optimum value for ZHARD, and is set to 0, i.e. the reference level (=initial expected mean ZHARD for in Generation 1 for the local individuals).
  + **W\_max** is the maximum survival when *ZHARD* = the optimum.
    - To generate a stable population (prior to introgression), **W\_max** can be set to e.g., 1.1/**k**. So if **k** = 2 (each parent produces on average 2 offspring), then we set **W\_max** = 0.55. This ensures that an expected 1.1 offspring per parent survive the hard selection filter, i.e., recruits per spawner = 1.1, giving some reproductive excess (but not too much, as otherwise you’ll never see negative demographic effects of intrusion!).
  + **Omega** is the width of the fitness function, measured in phenotypic standard deviation units of *ZHARD*. Moderate selection here would correspond to an omega of about 5 (i.e., 5 times the expected initial standard deviation of *ZHARD* in this first generation of hard selection, which will vary across runs depending on the level of intrusion of non-locals and how different their mean *ZHARD* is relative to locals… so we will have to just choose a reasonable value for omega that applies across all these scenarios, i.e., **Omega** will be constant across runs but **Omega** ÷ standard deviation of *ZHARD* will not be constant).

*Step 5: Cycle back to Step 2, i.e., the survivors of hard selection are then the recruits (pre-breeders) that must compete for K limited breeding sites.*

* Back we go to the start. The survivors of hard selection now immediately graduate to pre-breeder status, and their genotypic matrices are now passed to the soft-selection filter.
* At this pre-breeder stage each generation, we calculate summary statistics of interest on a per-generation basis, as model output:
  + The phenotypic means and variances of *ZSOFT* and *ZHARD*
  + The genotypic variances for each. This then allows us to calculate the realised heritability for each.
  + The realised genetic correlation between *ZSOFT* and *ZHARD* (correlation between individual genotypic values for each)
  + The allele frequency at the neutral (introgression marker) locus
  + The number of pre-breeders *N*.
  + The realised mean survival through the hard-selection filter (number of offspring just after divided by number of offspring just before).

Table 1. List of parameters. Does not include derived parameters, which can be computed based on these core parameters.

|  |  |  |
| --- | --- | --- |
| **Parameter number** | **Symbol** | **Meaning** |
| 1 | **N\_local** | Initial number of locals (as pre-breeders) |
| 2 | **F** | Fraction of all pre-breeders that are intruding non-locals (applied only in generation 1) |
| 3 | **num\_loci\_total** | Total number of loci affecting each quantitative trait (*ZSOFT* and *ZHARD­*­­) |
| 4 | **num\_loci\_shared** | Number of loci affecting both traits |
| 5 | **p\_local\_soft\_unique** | Initial frequency of 1 allele for local individuals for loci unique to *ZSOFT* |
| 6 | **p\_nonlocal\_soft\_unique** | Initial frequency of 1 allele for non-local individuals for loci unique to *ZSOFT* |
| 7 | **p\_local\_hard\_unique** | Initial frequency of 1 allele for local individuals for loci unique to *ZHARD* |
| 8 | **p\_nonlocal\_hard\_unique** | Initial frequency of 1 allele for non-local individuals for loci unique to *ZHARD* |
| 9 | **p\_local\_shared** | Initial frequency of 1 allele for local individuals for loci shared by *ZSOFT* and *ZHARD* |
| 10 | **p\_nonlocal\_shared** | Initial frequency of 1 allele for non-local individuals for loci shared by *ZSOFT* and *ZHARD* |
| 11 | **a\_soft** | Genotypic effect per locus for *ZSOFT* |
| 12 | **a\_hard** | Genotypic effect per locus for *ZHARD* |
| 13 | **h2\_init** | Initial heritability (assumed to be the same for both of traits) |
| 14 | **K** | Number of breeding slots (carrying capacity) |
| 15 | **soft\_switch** | Turns soft selection on or off |
| 16 | **k** | Fecundity (offspring per parent) |
| 17 | **W\_max** | Maximum height of hard-selection function (max offspring survival when *ZHARD* coincides with optimum |
| 18 | **Theta** | Optimum value for *ZHARD* |
| 15 | **Omega** | Width of fitness function, which determines strength of stabilising selection. |