**Assortative Mating Description**

Mate selection can be decided based on how similar individuals are in terms of their ‘HIndex’. The ‘HIndex’ (or admixture coefficient) is the proportion of *AA* genotype an individual has. If *AA*, then the ‘HIndex’ = 1.0. If *aa*, then the ‘HIndex’ = 0.0. The first F1 cross would produce a 0.5 ‘HIndex’. Then offspring receive the average ‘HIndex’ of their parents. This is the first step at understanding how much admixture is occurring. However, the ‘HIndex’ does not differentiate hybrid crosses (i.e., F1, F2, etc.) between parental back crosses (i.e., B1, B2, etc). For now, we will use the ‘Hindex’ idea to sort and find similar ‘species’ representative of their genotype with the assumption that phenotype is expressed as well. The models chosen follow M’Gonigle and FitzJohn 2010 Evolution with slight deviations of their models to represent hindex similarity.

In the PopVars.csv, there are two fields that define how individuals choose mate partners.

1. AssortativeMating\_Model (5 options: 1 – 5)
2. Option ‘1’: Random mating. Assortative mating factor ignored.
3. Option ‘2’: Strict self-preference mating. Here, the female can only have a mate partner with the same hindex. If within the females mate threshold distance there are no males with the same hindex, then this female will not have fertilized eggs.
4. Option ‘3’: Self-preference mating (e.g., M’Gonigle and FItzJohn 2010). The equation M’Gonigle and FitzJohn used was:

i,j = *ci,j \* fj*

Where *fj* denote the frequency of the males with genotype *j* (in this case hindex *j*) in the female’s patch or mating kernel. *i,j* is Kronecker’s Delta, which is equal to one when *i = j* and zero otherwise. i,j’s are standardized across all males. The user must specify the *c* parameter (see AssortativeMating\_Factor below). Note that the difference between option ‘2’ and ‘3’ is that when there exists a situation in which no males are within the females mate threshold distance with the same hindex, then the female will still have fertilized eggs from another male’s hindex. In other words, a very large c factor could be given, but mate partners still may not reflect self-preference mating.

1. Option ‘4’: Dominance-preference mating (e.g., M’GOnigle and FitzJohn 2010). This is similar to option ‘3’, but now the Kronecker’s Delta will only be zero when there is a ‘HIndex’ of 1.0 and 0.0. Hybrids can mate with any other ‘HIndex’ in this model.
2. Option ‘5’: Linear-preference mating (e.g., M’Gonigle and FitzJohn 2010). This assortative mating model assumes that mate selection occurs with higher probability the more similar you are in terms of ‘Hindex’. Following M’Gonigle and FitzJohn (2010), we derived the following equation to reflect similarity chosen based on ‘HIndex’:

i,j = (1 + (1 – abs(Female\_hindex – male\_hindex)) \* (*c* – 1)) \* *fj*.

1. AssortativeMating\_Factor ([1-100000])

The user specified *c* parameter for models 3 – 5 above. If *c* = 1, then random mating occurs and the mating model will be no different than using the option ‘1’ model. The larger the *c* factor the closer to strict self-preference mating.

**Psuedo Code for Assortative Mating**

DoMate(…) Module

# Select out only mature males and females

# --------------------------------------------------------------------

# Choose pairs for mating

# --------------------------------------------------------------------

DoSexual() Module

# Randomly grab a female

Female

# Get the female’s patch

Female\_Patch

# Get the female’s HIndex

Female\_HIndex

# Get potential patches males can come from and probability

Probarray # e.g., ([0.0, 0.75, 1.0, 0.25, 0.0])

# If the probarray does not equal 0

While sum(probarray) != 0.0:

# Select a patch the male will come from

Select\_Patch = w\_choice\_item(probarray)

# Select all the males in this patch

Patch\_Males

# If no males in the patch: move to next patch

# Else:

# Get the males Hindex

Males\_Hindex

# Assortative mating option ‘2’: Strict Self

If AMate\_Model == ‘2’:

# Check for match males

Select\_Males = Where(Female\_Hindex == Males\_Hindex)

# If there are no males with the same Hindex

Continue; move to next patch

# Else:

Return just males with same HIndex

Patch\_Males = Patch\_Males[Select\_Males]

# Assortative mating option ‘3’: Self-preference

elif AMate\_Model == ‘3’:

# Get the frequency of each males by their HIndex

Males\_HIndex\_fj

# Calculate probability

Males\_HIndex\_Prob = AMate\_C ^ (Female\_Hindex == Males\_Hindex) \* Males\_HIndex\_fj

# Standardize

Males\_HIndex\_Prob = Males\_HIndex\_Prob / sum(Males\_HIndex\_Prob)

# Draw one weighted

Select\_Males = w\_choice\_item(Males\_HIndex\_Prob)

# Return these males with the selected HIndex

Patch\_Males = Patch\_Males[Select\_Males]

# Assortative mating option ‘4’: Dominance-preference

elif AMate\_Model == ‘4’:

# Get the frequency of each males by their HIndex

Males\_HIndex\_fj

# Calculate probability for females with 1.0 (all males except 0.0)

if Female\_Hindex == 1.0:

Males\_HIndex\_Prob = AMate\_C ^ (Males\_Hindex > 0.0) \* Males\_HIndex\_fj

# Calculate probability for females with 0.0 (all males except 1.0)

elif Female\_Hindex == 0.0:

Males\_HIndex\_Prob = AMate\_C ^ (Males\_Hindex < 1.0) \* Males\_HIndex\_fj

# Calculate probability for hybrid females (all males)

Else:

Males\_HIndex\_Prob = AMate\_C ^ (Males\_Hindex <= 1.0) \* Males\_HIndex\_fj

# Standardize

Males\_HIndex\_Prob = Males\_HIndex\_Prob / sum(Males\_HIndex\_Prob)

# Draw one weighted

Select\_Males = w\_choice\_item(Males\_HIndex\_Prob)

# Return these males with the selected HIndex

Patch\_Males = Patch\_Males[Select\_Males]

# Assortative mating model option ‘5’: Linear-preference

elif AMate\_Model == ‘5’:

# Get the frequency of each males by their HIndex

Males\_HIndex\_fj

# Calculate probability

Males\_HIndex\_Prob = (1 + (1 – abs(Males\_Hindex – Female\_Hindex)) \* (AMate\_C – 1)) \* Males\_HIndex\_fj

# Standardize

Males\_HIndex\_Prob = Males\_HIndex\_Prob / sum(Males\_HIndex\_Prob)

# Draw one weighted

Select\_Males = w\_choice\_item(Males\_HIndex\_Prob)

# Return these males with the selected HIndex

Patch\_Males = Patch\_Males[Select\_Males]

# Returned Patch\_Males (Note if AMate\_Model == ‘1’), choose from all males and skips above)

Male\_Mate = random.smaple(Patch\_Males,1)

# Store mated pair

[Female, Male\_Mate]

# Tracking numbers