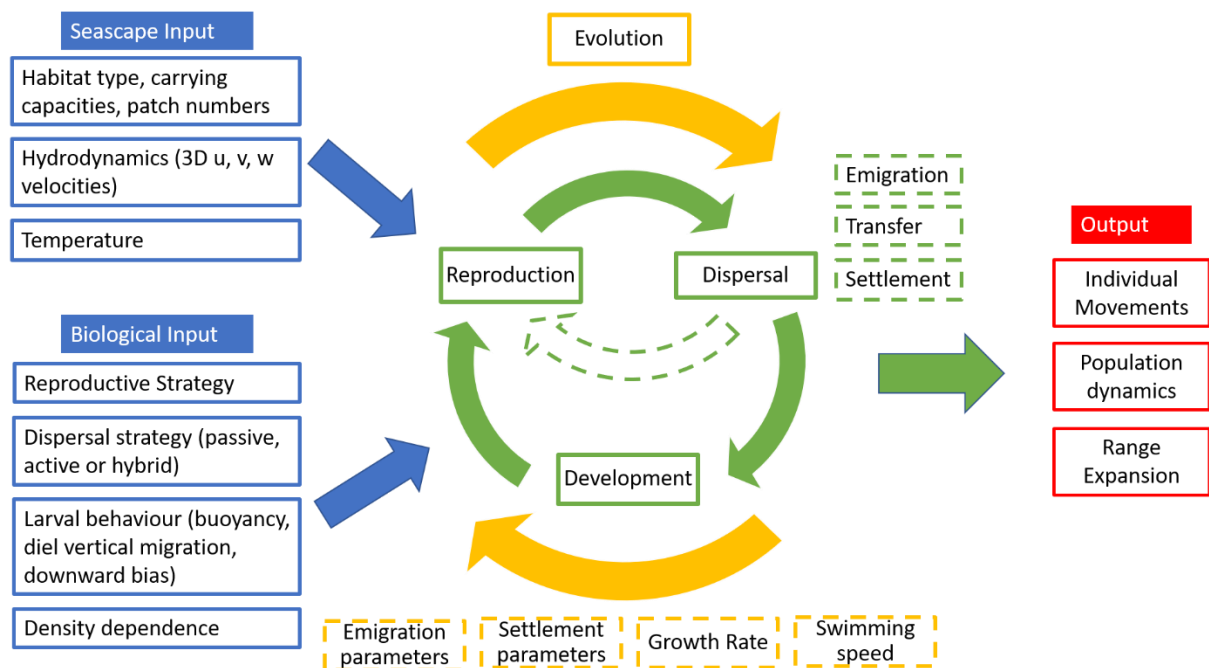# User Manual for MerMADE

The model for predicting Movement And Dispersal Evolution

in aquatic environments

Developed by Rebekka Allgayer

University of Aberdeen

Model Schematic

Seascape Input

Habitat type, carrying capacities, patch numbers

Hydrodynamics (3D u, v, w velocities)

Temperature

Biological Input

Reproductive Strategy

Dispersal strategy (passive, active or hybrid)

Larval behaviour (buoyancy, diel vertical migration, downward bias)

Density dependence

Evolution

Reproduction

Dispersal

Development

Emigration

Transfer

Settlement

Output

Individual Movements

Population dynamics

Range Expansion

Emigration parameters

Settlement parameters

Growth Rate

Swimming speed

How to use this user manual

Since MerMADE doesn't have a user interface yet, it works purely off of input files that contain all the data needed to tell it what you'd like to simulate. Therefore, I have structured this user manual by input file and as we go on, I give you all the information relevant to that input file. Note that there is some overlap between files and therefore it isn't always possible to completely compartmentalise, but I have done my best!
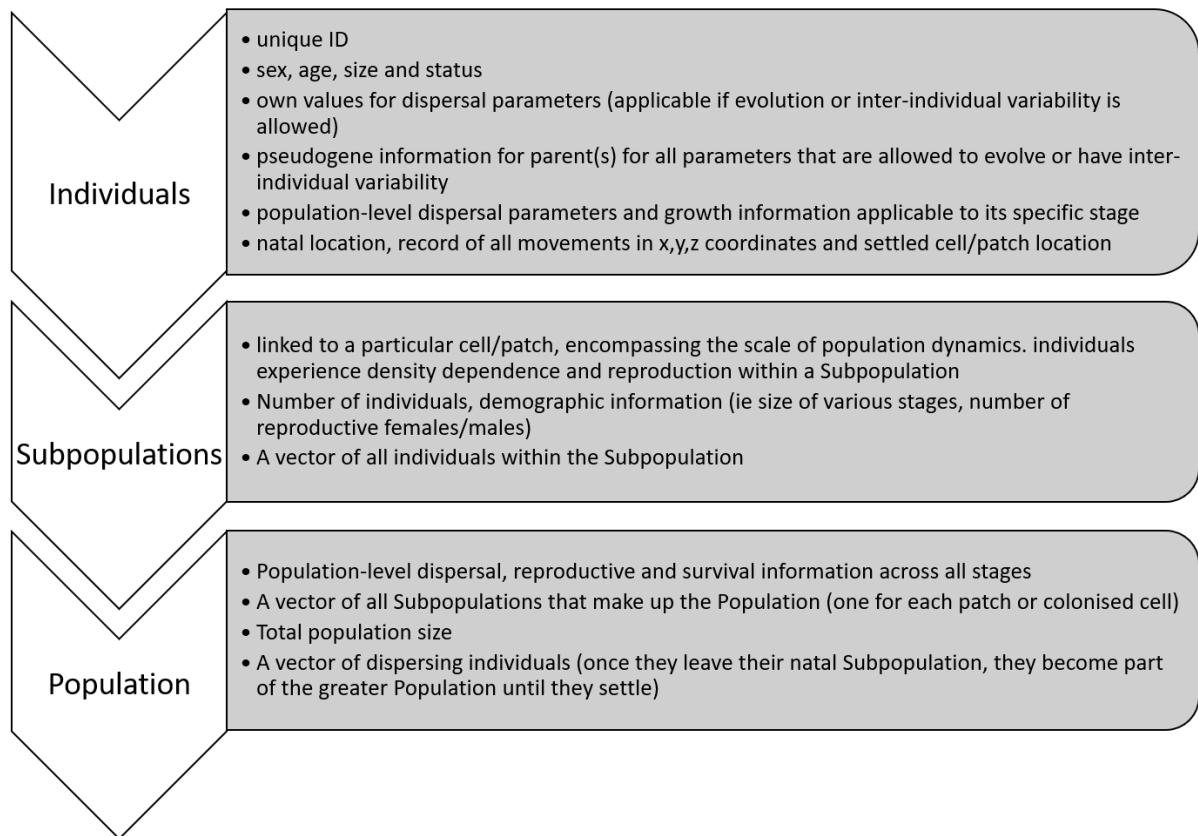
So, to use this user manual to build your simulation, simply take it one input file at a time and make your decisions as they come up.
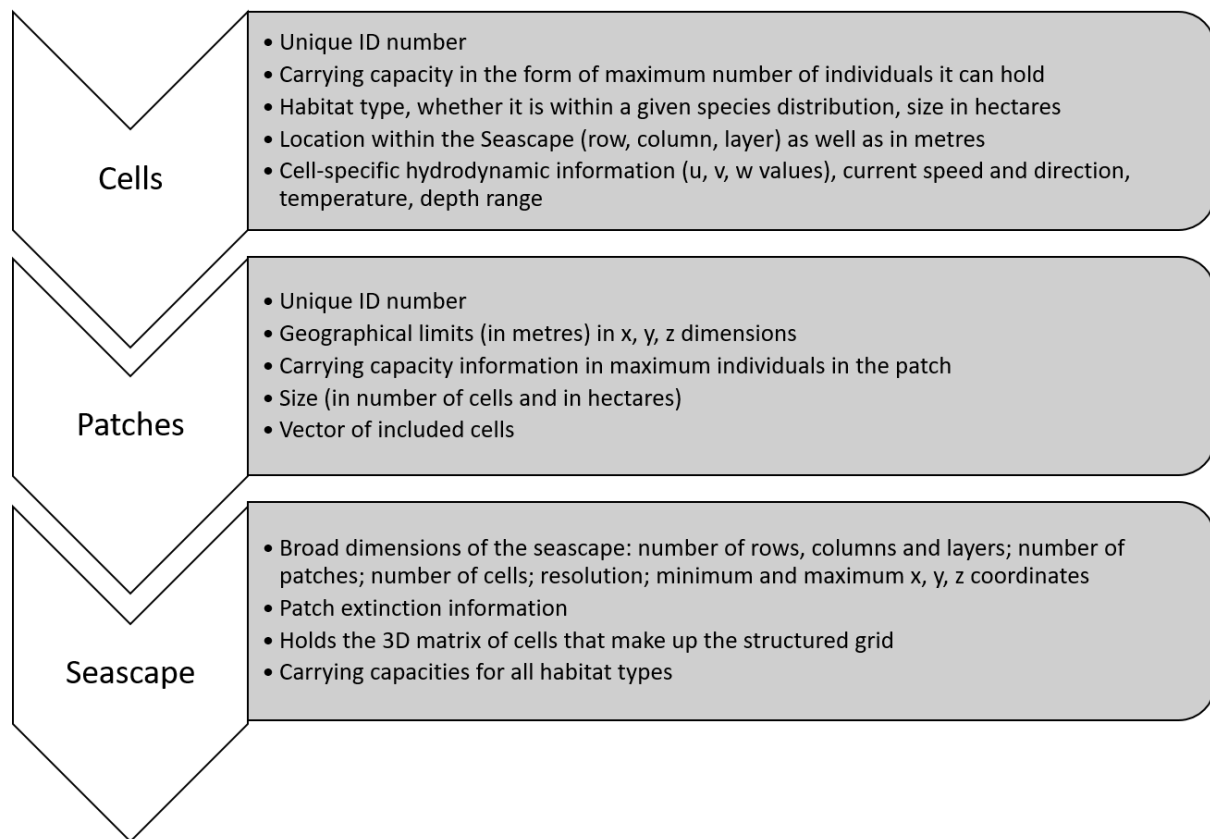
General information: How things work

Some general information about how the model is structured first, so you have some idea of what's going on behind the scenes. You can also jump straight to the Using MerMADE section. Good luck!

Structure of the Code

MerMADE is a standalone, individual-based, spatially explicit and stochastic dynamic model coded in C++. Its structure relies on the use of nested *classes*. For example, many instances of the *Individual* class make up a *Subpopulation* class, and several *Subpopulations* make up the overall *Population* class. Each class contains different information relevant to itself and its processes. This structure is necessary for the user to extract information at the relevant level and also for processes such as evolution to take place. Individuals must carry their own unique information so that their pseudogene expression is specific to them and their parentage as well as the parameter values that individual passes on to its own offspring.

Individuals

- unique ID
- sex, age, size and status
- own values for dispersal parameters (applicable if evolution or inter-individual variability is allowed)
- pseudogene information for parent(s) for all parameters that are allowed to evolve or have inter-individual variability
- population-level dispersal parameters and growth information applicable to its specific stage
- natal location, record of all movements in x,y,z coordinates and settled cell/patch location

Subpopulations

- linked to a particular cell/patch, encompassing the scale of population dynamics. individuals experience density dependence and reproduction within a Subpopulation
- Number of individuals, demographic information (ie size of various stages, number of reproductive females/males)
- A vector of all individuals within the Subpopulation

Population

- Population-level dispersal, reproductive and survival information across all stages
- A vector of all Subpopulations that make up the Population (one for each patch or colonised cell)
- Total population size
- A vector of dispersing individuals (once they leave their natal Subpopulation, they become part of the greater Population until they settle)

The Seascape is similarly structured, with *Cell* class objects making up *Patch* objects (if a patch-based model is being applied), and both making up the *Seascape* umbrella object. If the model is only cell-based then the *Patch* class is never instantiated and the *Seascape* only works on the cell level with *Subpopulation* classes operating on a cell scale. This nested structure is also useful for computational efficiency. Since every Cell carries its own hydrodynamic information, when and Individual enters a new cell, we do not have to search at the higher level of the *Seascape* for the relevant information. Calculations are run the minimum number of times possible and the cells store only their values.

Cells
- Unique ID number
- Carrying capacity in the form of maximum number of individuals it can hold
- Habitat type, whether it is within a given species distribution, size in hectares
- Location within the Seascape (row, column, layer) as well as in metres
- Cell-specific hydrodynamic information (u, v, w values), current speed and direction, temperature, depth range

Patches
- Unique ID number
- Geographical limits (in metres) in x, y, z dimensions
- Carrying capacity information in maximum individuals in the patch
- Size (in number of cells and in hectares)
- Vector of included cells

Seascape
- Broad dimensions of the seascape: number of rows, columns and layers; number of patches; number of cells; resolution; minimum and maximum x, y, z coordinates
- Patch extinction information
- Holds the 3D matrix of cells that make up the structured grid
- Carrying capacities for all habitat types

Visualising the Seascape

MerMADE runs over a seascape which is represented as a layered stack of grids, with user-defined X and Y resolution in metres, with each layer accounting for a user-defined depth interval, also in metres. X and Y resolutions are required to be identical, so each cell is square in the XY plane, but the Z dimension may have a different resolution, most likely much finer than the XY, as this is more biologically relevant and representative. Therefore, each cell will in most cases be rectangular in the XY direction (Figure 1).

The appropriate resolution in the XY plane will depend on the species being modelled and the geographic location being represented. For example, if it is an active disperser and therefore has a swimming speed, represented by its step length (SL) in calculations, it would make little sense to make the XY resolution much smaller than the SL, unless fine-scale hydrodynamic information is critical, since one movement step will take the individual out of the present cell. The finer the resolution, the more computationally expensive the model becomes, as more memory is needed to store the additional cell-objects. It is also worth noting that the model allows for various levels of activity throughout the dispersal process so the resolution needed at the phase where hydrodynamics play the most influential role might be the deciding factor.
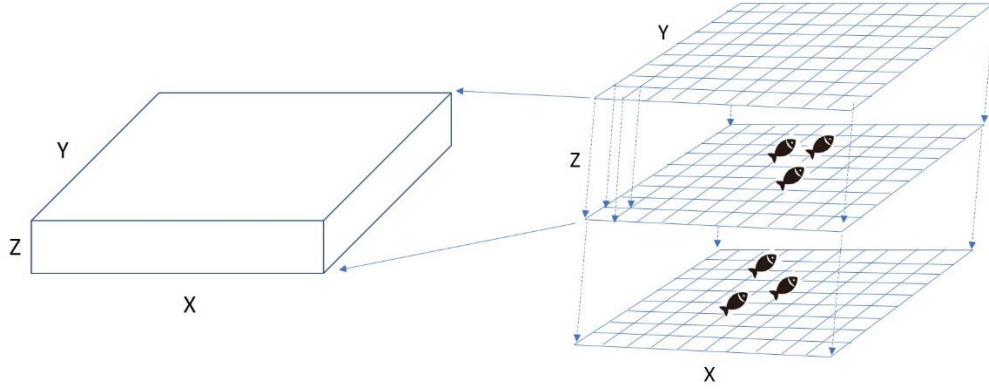
Figure 1: The seascape is represented by a layered stack of grids, where cells are square in the XY plane but can have a much finer resolution in the Z direction, most often creating a rectangular cell in the XY direction.

It is up to the user whether the top and bottom layers represent the surface and seafloor, respectively. If this is the case, then these boundaries will act reflectively, while the XY boundaries are always absorbing as there is no physical obstacle for movement (unless the last row or column is solid, in which case the individual will never reach the seascape's boundary, as they will settle or change direction). If, on the other hand, the seascape is a slice of water column, the top and bottom layers will also act as absorbing boundaries.

Representing hydrodynamic data

Each cell holds its hydrodynamic velocity values which are needed to calculate current velocity and direction: *u* (eastward velocity), *v* (northward velocity) and *w* (upward velocity) in a Cartesian vector format. In non-water cells (HT > 0), these values will be *-9999*, the NODATA value used for this model. These values are then used in the following equation to calculate the angle and magnitude of the current vector. Since this is a 3D vector, two angles are needed to describe its direction.

$$\varphi = \arctan(\frac{v}{u}) \tag{1}$$

$$\theta = \arccos(\frac{w}{r}) \tag{2}$$

Where *r* is the magnitude of the 3D vector, which also represents the current's speed in that cell, and is calculated by:

$$r = \sqrt{u^2 + v^2 + w^2} \tag{3}$$

Angles are handled in radians (0, 2π).
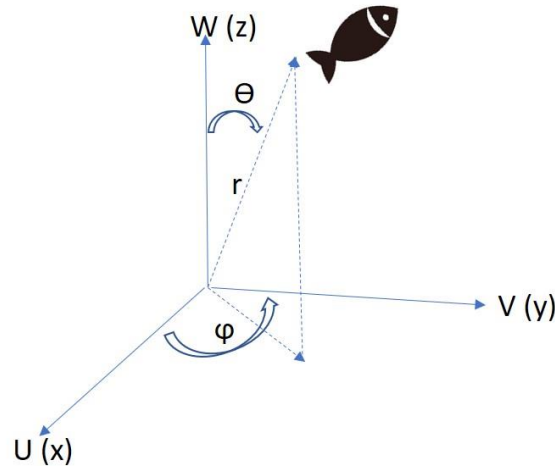
Figure 2: Eastward, northward and upward velocities (u, v, w respectively) are used to calculate the angle of the current in the XY plane (φ) and the Z plane (θ). Note that θ is measured from the Z axis, so θ = 0 points upward, therefore an individual making no depth change would be θ = 1.5708. Note also that φ is measured from the x axis, therefore φ = 0 would point eastwards.

Cell-based vs Patch-based

Depending on the study species, it might be appropriate to cluster cells into patches and have population dynamics operate on the scale of a patch (Bocedi et al. 2014, 2020, Malchow et al. 2020). This is referred to as a patch-based model and might be applicable for species whose habitat is inherently patchy (ie coral reefs, sandbanks, seagrass beds etc), often surrounded by less suitable or completely non-suitable habitat. Using this model type allows for post-settlement movement within the patch (though this is generally not explicitly modelled) as reproduction happens on a patch-level rather than only immediate neighbours in the same cell. During dispersal, settlement on cells outside of designated patches is not allowed.

Cell-based models, on the other hand, might be more applicable for species whose habitat is more scattered and not in recognisable patches. Reproductive partners are chosen from within the same cell, which is perhaps more realistic for completely sessile organisms that must mate with immediate neighbours, and post-settlement movement out of that cell is assumed not to exist. The choice of running a patch-based or cell-based model is also present in MerMADE's parent model, RangeShifter, and allows for two very distinct modes of visualising range expansion, for example. The relationship between dispersal success, potential dispersal distance, and habitat fragmentation is particularly clearly visualised in patch-based models where the odd patch of suitable habitat is no longer available to use as a stepping-stone during dispersal.

MerMADE can be applied in both a cell-based and a patch-based manner. Patches are 3D structures that can transcend layers and have irregular shapes (depending on resolution). When individuals are initialised, they are placed on the outside edge of their starting patches and no movement through solid structures is ever allowed within any phase of the model. The initialisation of individuals at the start of a simulation can either be done randomly by choosing a percentage of suitable habitat in a specific XYZ range, initialising from one specific patch, or by reading in an initial species distribution. In patch-based model simulations, a subpopulation is made up of all the individuals in the cells comprising that patch, while in cell-based models, each cell has its own mini-subpopulation for population dynamics.

How the Movement Model works

Movement through the seascape is always cell-based, in that individuals use hydrodynamic data in each cell to inform their movement. With the hydrodynamic information detailed above, we know in which direction the current is flowing and how quickly, but the dispersal method of the individual will determine how it reacts to this information. Passive dispersers will be highly influenced by the current while active dispersers may be able to make additional decisions that will dictate in which directions they swim. The calculated φ angle is then used as a mean in a circular Wrapped Cauchy Distribution to introduce stochasticity to the model. A concentration parameter ρ is used to indicate how closely sampling should adhere to the given mean of the distribution. For passive individuals, $\rho = 0.9$, but the ρ for active dispersers is user-defined $(0 - 0.9)$. A value $\rho = 0$ would indicate an entirely active disperser that is not influenced at all by current and increasing ρ represents increased attention to current directionality during decision-making. It is important to note that this does not represent a *proportion*, i.e. $\rho = 0.5$ does not mean that 50% of the time, the individual chooses the direction of the current. It means that the individual is equally influenced by current as by its own directional persistence, for example.

Using φ and θ, the coordinates of the new position can then be calculated using:

$$x = x_c + \text{SL} \left(\sin(\theta) \cos(\varphi) \right) \tag{4}$$

$$y = y_c + \text{SL} \left(\sin(\theta) \sin(\varphi) \right) \tag{5}$$

$$z = z_c - \text{SL} \left(\cos(\theta) \right) \tag{6}$$

where x_c, y_c, z_c are the coordinates of the current position, and SL is the step length or swimming speed of the individual. If the individual is passive, SL is the speed of the current, as they have no swimming power of their own. The fact that the XY resolution will most likely be different from the Z resolution makes scaling it by the SL for active individuals an interesting consideration, since the SL represents the swimming speed in the XY plane, i.e. an individual would not necessarily move the same distance in the vertical direction. This would therefore mean that any movement in the XY direction would be disproportionately large in the Z direction. However, this can't be avoided since multiplying by the same scalar is required to preserve the calculated angles. For actively dispersing individuals, we have therefore restricted θ to $(1.55, 1.6)$, so any movement is only moving slightly upwards or downwards (no change in depth would have $\theta = 1.5708$). The current is therefore informing the direction of movement in the XY plane and is informing "up" or "down" in the Z plane, but the individual retains decision-making power over the degree of depth change. No such restriction is placed on values of θ for passive individuals.

It is important to note that the distance to travel in the Z direction is subtracted from the current coordinate in Eqn. 6. This is due to the fact that the Z axis is reversed, since depth increases as individuals move downwards. This means any value of θ that would have resulted in a positive movement on a normal axis, now requires a decrease in the Z value, and vice versa.

Active individuals are also able to store past movements in memory, thereby increasing their directional persistence, avoiding overly laborious and inefficient movements. There is also functionality within the model for actively dispersing individuals to detect suitable habitat and change movement behaviour to move towards it once they have passed the minimum time/size for competency. Detection distance is user-defined in number of cells (making it resolution dependent), and can represent detection via sight, sound or olfactory cues. The ability of the individual to follow these cues and move towards suitable habitat depends on its dependence on the current, ρ.

Therefore, we take a weighted average of the two vectors: the direction towards suitable habitat and direction of current flow (Eqn. 7-10 using equations 4,5 and 6).

$$x_{average} = (1 - \rho)(x\_(suitable))) + \; \rho(x\_current) \qquad (7)$$

$$y_{average} = (1 - \rho)(y\_(suitable))) + \; \rho(y\_current) \qquad (8)$$

$$z_{average} = \; (1 - \rho)(z\_(suitable))) + \; \rho(z\_current) \qquad (9)$$

$$\varphi = \arctan\left(\frac{y_{average}}{x_{average}}\right) \qquad (10)$$

This method is also used to calculate the direction when memory is used as well as the direction of the current.

Additionally, active dispersers may have a downward bias on their movements, which allows them to find the seafloor or their depth boundary of the seascape, as the case may be, and to track depth changes in bathymetry. This is done on the assumption that organisms that are searching for a place to settle will do so on the seafloor or on the side of a protruding habitat mound (i.e. a sea mount or reef wall), rather than searching the water column for suitable habitat. Once suitable habitat is detected, this downward bias is lifted and the individual moves in the direction of the suitable cell. Passive dispersers do not have this bias as they are forced to move with the current, regardless of direction.

Due to an active disperser's potential to move more than one cell in a time step, if SL > XY resolution, each cell along the trajectory is assessed to avoid individuals moving *through* solid habitat, be it the seafloor or a raised patch. This is done by calculating the unit vector between the current position and the goal coordinates, scaling by the distance it will take to reach the next cell, and assessing each new cell encountered. If it is solid and unsuitable, the organism is turned away to find a path around it, moving in a new direction until it has reached the maximum distance to move in one timestep (its SL), when a new assessment is made and new goal coordinates are calculated. If the encountered cell holds suitable habitat, the individual assesses whether or not to settle in that cell and is deflected as described above if it decides not to settle.

Activity levels

Many organisms that use pelagic larval dispersal have both a passive and an active phase. This means that after a developmental period, the juvenile is then morphologically competent enough to be able to influence its movements and potentially detect and move towards suitable habitat. The model also allows this functionality, in that after a user-defined competency period is reached, a passively dispersing individual is assigned a swimming speed and a downward bias and searches the environment for suitable habitat. This also gives it the ability to assess a habitat patch or cell when it comes across one and resuspend and continue searching, should it decide not to settle. Before the competency period is reached, a passively dispersing individual is forced to settle on any solid habitat it is swept towards by the current, whether it is suitable or not, with a small degree of stochasticity. If the habitat turns out to be unsuitable, that individual dies as it is unable to resuspend.

Diel vertical migration

Alternatively, or perhaps additionally, individuals may undergo larval behaviour such as diel vertical migration (dvm) which affects their vertical distribution in relation to day and night cycles but does not give them the ability to influence their horizontal movement. A user-defined dvm range is used to determine upper and lower boundaries, and positions within the water column are then

sampled within these restrictions every 12 hours to simulate day vs. night, and we assume juvenile release occurs in the evening. We acknowledge that this is simplified and that, depending on time of year and geographical location, an even day-night split may not be accurate. Improvements to the accuracy of dvm within this model is an area for future development.

MerMADE therefore allows for four stages of dispersal activity: completely passive, passive but with diel vertical migration, active pre-competency, and active with settlement competence. The user can choose any combination of these activity levels, given suitable information on minimum times and/or sizes for when to switch to the next level. More detail on this can be found in the relevant sections of the Transfer File.

Inheritance, Inter-individual variation and evolution

Individuals carry a 'pseudogene' coding for the genotype of each trait, which juveniles will simply inherit from their parent(s) with 100% heritability. If the model is a female-only, asexual model, the species is considered to be haploid and the pseudogene is a single allele at that trait's locus on the chromosome. In the case of a sexual model, species are diploid and each trait is represented by two alleles held by the chromosome at each locus. How an individual inherits its parents' information depends on whether that trait is sex-dependent, meaning that the trait differs between the sexes. If not, then a juvenile inherits one allele from its mother and one from its father, expressing the arithmetic mean of the two values. If the trait is sex-dependent, then it inherits one allele coding for the female trait and one for the male trait from each parent, inheriting four values in all (whether the parent passes on the allele it received from its mother or father is random). Depending on the sex of the offspring, it will express the mean of the appropriate alleles for its phenotype (Figure 3). Individuals store their parent(s)' information exactly as it is visualised in Figure 3.
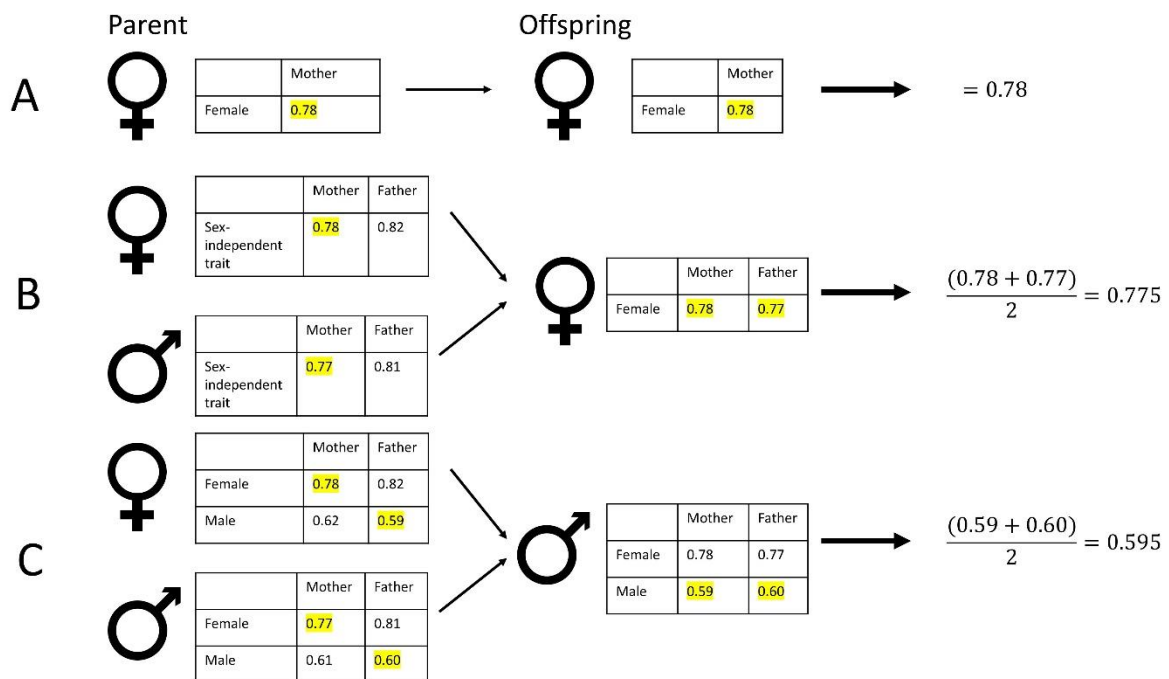
Figure 3: How an individual inherits values for each trait's pseudogene depends on whether the model is asexual (A), sexual with a sex-independent trait (B), or sexual with a sex-dependent trait (C).

These methods of inheritance become important if inter-individual variation and/or evolution are introduced into the model.

Inter-individual variation

In order to reflect the natural variation present within a population, MerMADE allows the user to introduce inter-individual variability (IIV) in certain traits: initial size of juveniles at release, emigration probability, ρ, SL, minimum age/size when switching between dvm, activity and competency levels in the dispersal phase, growth rate, and settlement probability. This variation is represented by a user-defined standard deviation around the mean value given to the model, and an individual's trait is then sampled from a normal distribution using these two values. There is no restriction on the number of parameters that IIV can be applied to, but it is important to note that the cause of resulting patterns may be unclear if too many parameters vary at once.

This IIV is applied at model initialisation, where individuals inherit values drawn from this normal distribution and store them as their parent(s)' genetic information. It is worth noting that if the user wishes to apply IIV, values for emigration, transfer, and settlement phases can't be stage-dependent (i.e only one stage is then allowed to disperse). This is to facilitate computational simplicity. If stage-dependence of dispersal is a requirement, no IIV is currently allowed and therefore no evolutionary functionality is available either (see next section).

After this initial variability, there is no further introduction of more and values in the population during the course of the model run are a product of which values get inherited.

Evolution

The absence of evolutionary functionality in models predicting species' responses to environmental change makes the assumption that traits are fixed and species can't adapt to their new conditions. However, this rules out a potential coping mechanism, making conclusions drawn about the capability of a species to cope with environmental change incomplete. MerMADE therefore includes evolutionary functionality in several traits: emigration probability, age/size at activity and competency levels, growth rate and settlement probability. Each of these traits is allowed to evolve given a mutation rate and size in response to given environmental conditions. This mutation occurs when individuals inherit their parent(s)' genetic information, with individuals storing the mutated genotype, ready to pass it onto its own offspring.

This differs from simple IIV because this allows for the possibility of traits appearing in the population due to mutation that weren't present at initialisation, while IIV only allows for shifts in frequency within the given range. Having traits able to evolve in all three stages of dispersal allows us to tease apart crucial factors that decide the success of dispersal, as well as the influence that other physiological processes might have.

Using MerMADE

This section walks you through step by step all the parameters in the input files. This is a more in-depth level of detail than the Descriptions.xls file included in your Tutorial Package, but that is also a helpful tool to refer to as you're building your input files.

Running the program

At present, MerMADE does not run from a GUI or interface of any kind. It is available as an executable which, when opened, will call for a Terminal Window to open with the following instruction:
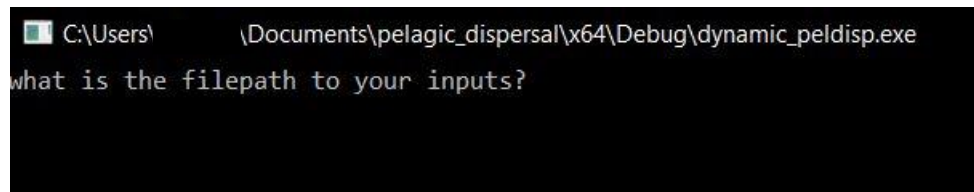
The filepath is the address of the directory in which all your input files are stored as well as the location in which the outputs of the model will be saved. This address will usually look something like this: C:\Users\Documents\pelagic_dispersal\MerMADE. This is the answer to the question prompted in the terminal window. Enter it by typing directly into the terminal window and press Enter.

Within the directory called MerMADE (in the example case, your name can be different!), there should be a folder titled Inputs and a folder titled Outputs, like so:
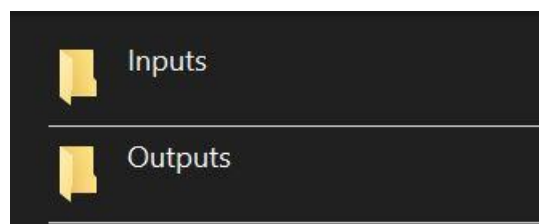
It is important that the folders be named in this way (with capital I and O) because otherwise the model will not know in which folder to look for the necessary documents. Please make sure that these names are correct.

At this point, MerMADE will automatically go through all the input files in specific sequence, following the instructions you have specified within them. These files should all be in *.txt format with headers across the top (with the exception of the Control File and the Dimensions File, see below). See example input files for more detail.

Let's go through the input files in the order MerMADE will read them. Note, however, that all input files and necessary parameters are described in more detail in the Descriptions.xls file included in your Tutorial Package. Example templates are also available there to guide you in structuring your inputs. Note that in the *.txt files, headers and values are separated by Tabs, not spaces!

Input Files

All the files in this section should be located in your Inputs folder in the working directory you specify when the terminal window pops up. You will notice references to a few key decisions that have to be made:

- Is your model cell-based or patch-based?

- Is your population asexual (female only) or sexual (both sexes)?
- Is your population stage-structured or are generations non-overlapping (individuals die after reproduction)?
- Is dispersal size-dependent (behaviour is linked to a growth function) or time-dependent (behaviour is linked to time)?

There is more information on the specifics of what these questions mean in the sections below but if you have an idea of which direction you'd like to go before diving into the input files, that will help make it all make sense!

Dimensions File

The Dimensions File (in this case called info_3d.txt, the filename is specified within the Seascape File, see Descriptions.xls for details) provides MerMADE with all the spatial information it needs to create the 3D raster that the entire model will operate on.

In this case, our seascape is made up of 267 rows and 229 columns of cells that have a resolution of 1500m in the horizontal direction (x,y). The lower left-hand x and y coordinate (in metres) is given for xllcorner and yllcorner, which tells MerMADE where to start calculating cell-specific coordinates from. There are 9 depth layers, each representing an interval of 10m and the minimum depth is 0, the surface. Specifying the NODATA_value is necessary if you have cells that contain no data and you want MerMADE to ignore them. This number can be anything as long as it is specified here but be careful in choosing this number as it should be a number very easily recognised to be nonsense. I STRONGLY advise against using 0 for this, as this could be a valid value for some parameters and this NoData value will be applied across all inputs and parameters. The number -9999 is commonly used and highly unlikely to correspond to a real value of any kind.

Now MerMADE knows how big the Seascape is, it can start populating the cells with real information, found in the Seascape File.

Seascape File

As the Seascape File is quite large, please see the dedicated sheet in the Descriptions.xls document for specific details on all parameters and the template provided for help in structuring this input file. In this section, I will cover the basics of what is contained in this file and things to watch out for.

The first section of the Seascape File is worthy of discussing in more details because of the Nhabitats parameter and the associated carrying capacity values. In this example, we have 3 habitat types: open water, often referred to as matrix, which is always K0; suitable habitat, K1; and unsuitable habitat, K2. The carrying capacity values tell the model how many individuals PER HECTARE can settle and reproduce in that habitat. In this file, there are three K values because I have three habitat types. If I had five, there would be five K values (an added K3, K4). This means that the number of

columns in your input file DEPENDS ON THE NUMBER OF HABITAT TYPES you have. Pay attention that these parameters match.

In the case above, only K1 has a K value. This is because K0, open water/matrix, is not suitable for settlement so it has K0=0. Similarly, K2 is solid substrate but it is not suitable for settlement (ie it's rocky when the organism needs sandy, or something similar). Therefore, 0 individuals per hectare are able to settle there.

Calculating K in individuals/hectare

The number of individuals per hectare DEPENDS ON YOUR RESOLUTION. 1 hectare= 100m x 100m= 10,000m^2 so in our case, where the resolution is 1500m in both the x and the y direction:

$$100m \times 100m = 10000m^2 = 1\ hectare$$

$$1500m \times 1500m = 2250000m^2 = 225\ hectares$$

So, each cell contains 225 hectares. If our carrying capacity is 0.022 individuals PER HECTARE then each cell can contain 4.95 individuals within it (because $0.022 \times 225 = 4.95$). Note that this value does not need to be a whole number because MerMADE will use it as a mean and draw from a probability distribution to add stochasticity to number of initialised individuals.

Using Super-individuals

In the marine environment, the number of individuals that are produced to eventually undergo dispersal can be enormous, far too many for MerMADE to model each one individually. It is common, therefore, to lump individuals into "super-individuals", meaning that instead of each individual being 1:1, each one now represents 1:1000 or 1:1million. Depending on the density the study species lives at in their preferred substrate, even this might result in too many individuals for MerMADE to handle, simply because of the complexity of the movement model and the number of calculations required. Play around with how you want to represent an "individual". This might seem like a huge simplification, and it is, but it is necessary unless you want to attempt to model millions and millions of individuals once your original population reproduces, most of which suffer mortality and don't complete their dispersal process anyway. This is a useful tool but remember to factor this into your above calculations of carrying capacity and individuals per hectare.

For example, if your usual density is 1 ind/m^2 (which for some sessile marine species is VERY low, think of barnacles) then your calculated K in ind/hectare would be 10,000 individuals. Each of our cells has 225 hectares, so this quickly becomes unmanageable because you have 2.25million individuals in once single cell. And if you think back to our Dimensions File, we have over 60,000 cells (though of course not all of them will be suitable habitat). You do the math.

But if we assume that 1 super-individual represents 10,000 individuals, then we reduce our numbers quite a bit, ending up with 0.0001 ind/m^2 = 1 ind/hectare = 225 ind/cell. This might still be quite a lot depending on the size of your overall seascape and the amount of suitable habitat present as well as your initialisation strategy (see relevant section) but is much more manageable, and you can always increase the super-individual ratio to match your needs.

LandFile_0.txt - Notepad

File Edit Format View Help

```
U_File          V_File          W_File          LandscapeFile   Abs_DepthFile   PatchFile
u_layers.txt    v_layers.txt    w_layers.txt    hab_layers.txt  sf_depth.txt    patch_layers.txt
```

The next few columns of the Seascape File indicate the locations of files containing information on habitat types, patch designation, hydrodynamic information, and absolute seafloor depth. We will go into each of these files individually below.

Patch extinction

The remainder of the Seascape File pertains to patch extinction functionality. In patch extinction simulations, the entire subpopulation of the chosen patch effectively experienced 100% mortality, leaving the patch empty before the next reproductive event. If this is irrelevant, then the values should remain -9 (except for a 0 under Patch_extinct to indicate False). If it is applicable, there are two methods of implementing patch-extinction over time.

You may want a random percentage of the patches to go extinct every year:

LandFile_0.txt - Notepad
File Edit Format View Help

Patch_extinct	P_ext_method	P_ext_p_prop	P_ext_patch	P_ext_i_prop	P_ext_int	P_ext_burnin
1	0	0.10	-9	1	5	15

In this example, 10% of patches will go extinct (with 100% mortality) every 5 years after an initial 15-year burn-in period.

On the other hand, you might want to have some control over which patch experiences extinction. In this case, we have specified that patch 10 should experience 100% mortality every 5 years after an initial 15-year burn-in period.

LandFile_0.txt - Notepad
File Edit Format View Help

Patch_extinct	P_ext_method	P_ext_p_prop	P_ext_patch	P_ext_i_prop	P_ext_int	P_ext_burnin
1	1	-9	10	1	5	15

These are all the parameters contained within the Seascape File.

Habitat Layers File

hab_layers.txt - Notepad
File Edit Format View Help
hab_0.txt
hab_1.txt
hab_2.txt
hab_3.txt
hab_4.txt
hab_5.txt
hab_6.txt
hab_7.txt
hab_8.txt

This file indicates the locations of the files which contain habitat type information for each layer of the seascape. In the model there must be at least two habitat types: water (HT=0) and suitable habitat (HT=1), but there can be as many habitat types as you like, each with their corresponding carrying capacity and therefore a range of "suitabilities" for settling dispersers. We assume 100% coverage of each cell of the specified habitat type.

We have 9 layers in our seascape, therefore we have 9 habitat type files. Each of these files contains 267 x 229 values (the number of rows and columns of the seascape), each value corresponding to the cell with that row and column number in the seascape raster within MerMADE.

Think about how these layers will look in MerMADE. These habitat layers are meant to be stacked to create 3D bathymetric profiles. Since we only have 3 habitat types (water, suitable and unsuitable), coastline and seafloor are designated by being habitat type 2. That means that the deeper we go (ie the high the layer number), the more cells are going to be type 2 because they are solid seafloor that is unsuitable for settlement. If we visualise this as a cross-section of seafloor, it looks something like this:

Depth						Layer
0	0	0	0	0	0	0
10	2	0	0	0	0	1
20	2	0	1	1	0	2
30	2	0	1	1	0	3
40	2	2	1	1	2	4
50	2	2	1	1	2	5

Here, blue-coloured cells are water with type 0, unsuitable habitat (ie land, seafloor) are orange and type 2, suitable habitat is green and type 1.
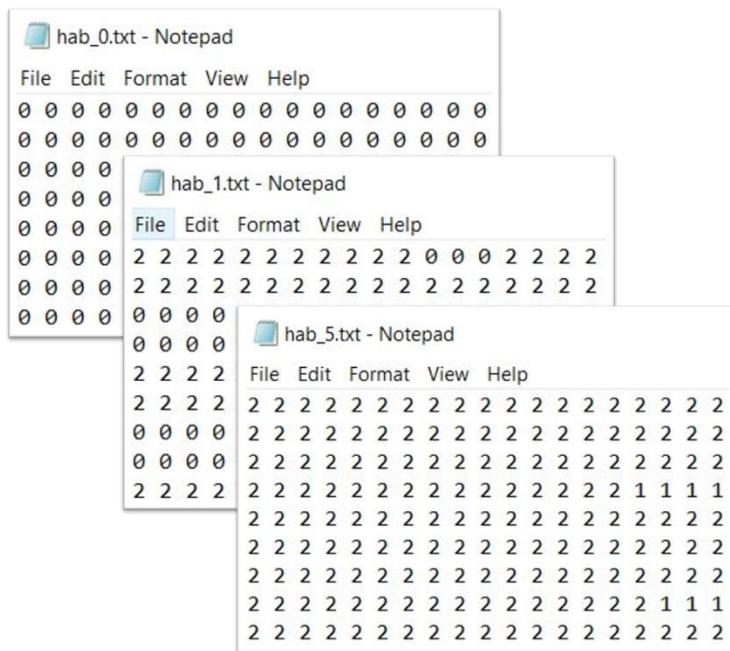
This is how this looks in the input files. Notice how, as the layers get deeper (0 at the surface, then 1, then 5) more and more of the cells are habitat type 2. Stacked, these will create a slope.

Patch Layers File

This file is only needed if the model being run is a patch-based model. The format of this file follows that of the habitat type file detailed above, except that this time, the 267x229 (in this case, remember this is the number of cells required for one layer of your seascape) values correspond to patch designation of each cell. Patch numbers should be sequential and start at 0 (ie if you have 5 patches, your patch numbers would be Patch 0, Patch 1, Patch 2, Patch 3 and Patch 4). Just as the
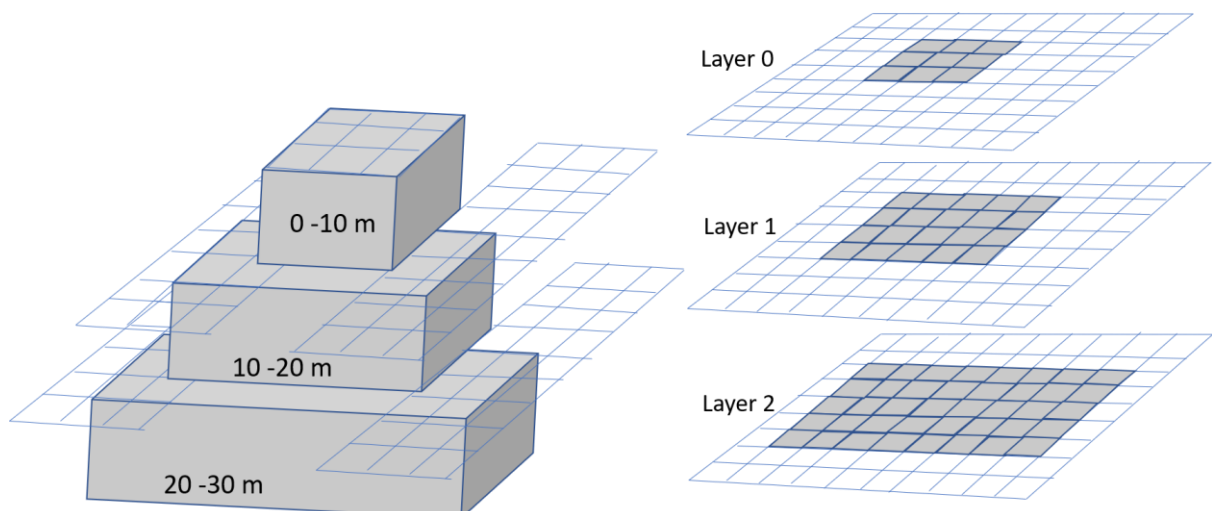
habitat file layers are stacked in MerMADE to create bathymetry, the Patch file layers are stacked to create 3D patches that dispersing individuals can settle on.

Absolute Depth File

This file is used to offset the issues of the coarse vertical resolution that will commonly be used. As you can see in the figure above, having a 10m vertical resolution makes patches very boxy and unnaturally shaped. The result can be that if an individual approaches a patch shaped like this instead of an accurately smooth slope, they will be deflected or settle incorrectly. Having the information of how deep the seafloor actually is allows MerMADE to know whether an individual can continue moving or needs to settle. This is a nuance of the model and I'm only detailing it here because it requires an input file, but as long as you can provide the data, you don't need to fully understand how it is being implemented. Note that **only one layer file** is required for this file because the absolute seafloor depth will not change with the layers. If the seafloor is at 62m then it will be 62m whether the individual is in layer 0 or layer 5. The format of this file is the same as the habitat type file and the patch file.

U, V, W Files

The format of all of these files is the same, therefore I will use the U File as an example that is applicable across the hydrodynamic vector types (including dynamic seascapes).

```
u_layers.txt - Notepad
File  Edit  Format  View  Help
u_0.txt
u_1.txt
u_2.txt
u_3.txt
u_4.txt
u_5.txt
u_6.txt
u_7.txt
u_8.txt
```

This might seem tedious, but this file contains the locations of the u (eastward) velocity information for each layer of the seascape.

In this example case, we have 9 layers, so we have 9 files of u-vector information. Each of these files contains a u velocity value for each cell in that seascape layer. The format of the u_0.txt etc files are the same as the habitat type files so I will not include an example here. If you are not using dynamic seascapes, this is all you need to do.

Dynamic Seascapes

If you are applying dynamic seascapes, then additional information is needed *on top of* what is provided above. The U File in the previous section is the status quo for model initialisation. For dynamic landscapes, you need to create a separate file. This file looks like this:

```
dyn_land.txt - Notepad
File  Edit  Format  View  Help
Sim      num      dyn_files       burn_in dyn_interval   hab     hydro   patches
0        5        dyn_files.txt   2       3               0       1       0
```

This indicates that there are 5 new sets of hydrodynamic information being read in, that the first new import will begin after a burn-in period of 2 years and after that will be read in every 3 years. At present, this functionality only exists for hydrodynamic information so ignore the "hab" and "patches" headers. Where the new hydrodynamic information can be found is detailed in dyn_files.txt:

Simulation	hab_files	patch_files	u_files	v_files	w_files
0	-9	-9	dyn_u.txt	dyn_v.txt	dyn_w.txt
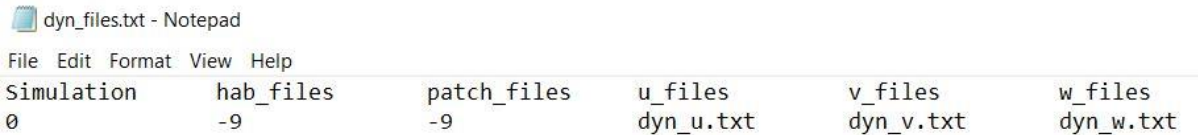
Similar to the Seascape File, it points to the location of .txt files containing *more* locations for the actual hydrodynamic information.

u_0_06.txt	u_1_06.txt	u_2_06.txt	u_3_06.txt	u_4_06.txt	u_5_06.txt	u_6_06.txt	u_7_06.txt	u_8_06.txt
u_0_08.txt	u_1_08.txt	u_2_08.txt	u_3_08.txt	u_4_08.txt	u_5_08.txt	u_6_08.txt	u_7_08.txt	u_8_08.txt
u_0_10.txt	u_1_10.txt	u_2_10.txt	u_3_10.txt	u_4_10.txt	u_5_10.txt	u_6_10.txt	u_7_10.txt	u_8_10.txt
u_0_12.txt	u_1_12.txt	u_2_12.txt	u_3_12.txt	u_4_12.txt	u_5_12.txt	u_6_12.txt	u_7_12.txt	u_8_12.txt
u_0_14.txt	u_1_14.txt	u_2_14.txt	u_3_14.txt	u_4_14.txt	u_5_14.txt	u_6_14.txt	u_7_14.txt	u_8_14.txt

In the first column, this file indicates where to find the u-vector hydrodynamic information for layer 0 can be found for the different years (in this case, 2006, 2008, 2010, 2012, 2014). You can name these files anything you like, you don't need to have the year in there because MerMADE will simply read them in sequentially at the time and interval you specified in the dyn_land file.

Species Distribution File

This file is only needed if you would like MerMADE to initialise individuals from an existing species distribution. This file is structured in the same way that the Habitat and Patch files are structured, except that each cell either contains a 0 (not in species distribution) or a 1 (in species distribution). For patch-based models, all cells within a patch that is within a species distribution should have the value of 1.

That is the end of the files that pertain to the seascape that MerMADE uses to run the model. Now we move onto the Individuals, Subpopulations and Populations that will make up your simulation.

Population File

This file contains the population-level information that is not specifically about dispersal or evolution. I would highly recommend looking at the Descriptions.xls document for more details on these parameters but I will cover them briefly.

Reproduction can be asexual (whereby the model assumes a female-only population as presence of males is not considered a limiting factor) or sexual, with the option of increasing complexity by including harem dynamics and polygamy. The standard sexual model simulates the broadcast spawning approach adopted by many aquatic organisms, where gametes are expelled into the water column and fertilisation—and therefore parentage—is not directly controlled by the parents. In this case, males are randomly assigned to females to pass on genetic material to their offspring, and a male can be chosen more than once as it is not sampling without replacement. This therefore ignores factors such as physical fitness in mating and assumes each male is equally likely to mate with a given female.

For asexual reproduction with non-overlapping generations, the number of offspring would be calculated by Maynard-Smith & Slatkin's (1973) model:

$$Poisson(\frac{Rmax}{1 + |Rmax - 1| \times \left(\frac{N}{K}\right)^{b_c}})$$

Rmax is the maximum growth rate, though this will only be obtained at very low densities, N is the number of individuals in the cell/patch, K is its carrying capacity, and b_c is the competition coefficient

which pertains to the type of density regulation, whether it's under-compensatory (bc < 1), compensatory (b=1) or over-compensatory (bc > 1). The number of offspring produced by each reproducing individual is drawn from this distribution to add a level of stochasticity to the system. In the case of sexual reproduction, MerMADE would multiply Rmax by 2 to account for both male and female offspring (since the above assumes we are only modelling females).

For stage-structured populations, parameters bc and Rmax are left without values (-9) as relevant parameters are specified in the StageStruct File (see below).

PopFile.txt - Notepad

File Edit Format View Help

Simulation	bc	Rmax	max_age	fish_mort	Nrepro	RepInt	Prep
0	-9	-9	10	0	1	0	1

In this case, the maximum age of our species is 10years, we are not applying any fishing mortality, individuals reproduce once a year and can reproduce consecutively (they do not need to skip reproductive periods before becoming fertile again) with 100% probability.

The next section pertains more to the larvae that are produced. Here, we have specified that they are release 5m above the location of the adults. This is to ensure that the larvae are expelled straight into the water column for dispersal. This value must be greater than 0 to move the disperser out of the natal cell in the first place. Larvae are 5.3mm in length at the start of dispersal (this needs to be specified if conducting a size-dependent dispersal model, see Transfer File for more details. If we were doing a time-dependent model, this value would be -9, irrelevant). We are not allowing inter-individual variation in juvenile size.

PopFile.txt - Notepad

File Edit Format View Help

Juv_rel	Juv_size	iiv_juv_size	juv_size_sd	SurvSched	Hsize	DepthFile
5	5.3	0	-9	0	-9	depthpref.txt

The DepthFile parameter is fairly basic at this stage (ready for more detailed implementation later). This file contains the depth preference information for the species.

depthpref.txt - Notepad

File Edit Format View Help

Simulation	DepthPref	DepthMin	DepthMax	DepthSex	MDepthMin	MDepthMax	FDepthMin	FDepthMax
0	0	0	170	0	-9	-9	-9	-9

This file indicates that depth preference is not sex-specific and that individuals can reside anywhere between 0-170m (this applies to both dispersal and settlement). With future development, these parameters could be temperature-dependent and therefore the position of species within the water column could shift with climate change, but this functionality is not yet implemented. Having sex-specific depth preferences could introduce interesting dynamics if the sexes only overlap at certain depths, for example, though this is a niche example.

The next section indicates that there is density dependence in fecundity but not in development or survival. Parameters such as fecundity and survival can be density dependent, therefore limited by the carrying capacity K associated with the habitat type of the cell or patch in question. The equation used for density dependent reproduction s:

$$\phi_i = \phi_{i,0} * e^{-bN_t}$$

where ϕ_i is the fecundity of stage i, $\phi_{i,0}$ is its maximum fecundity at low densities, b is the strength of density dependence, and N_t is the number of individuals in the local patch subpopulation. The value used for b depends on local patch carrying capacity K_i and can roughly be imagined as $\frac{1}{K_i}$, though this is only an approximation.

PopFile.txt - Notepad

File Edit Format View Help

FecDensDep	DevDensDep	DevDensCoeff	SurvDensDep	SurvDensCoeff
1	0	-9	0	-9

StageStruct File

This file is applicable if the population you are modelling is stage-structured with overlapping generations. This means you can assign different fecundity and survival rates for different age groups, for example. At present, this file is a placeholder for more information, and the only parameter you really need to pay attention to is the TransMatrixFile parameter, which indicates where MerMADE can find the transition matrix information for your population.

StageStructFile.txt - Notepad

File Edit Format View Help

Simulation	DepthStage	TransMatrixFile
0	0	transmat_sexual.txt

Transition Matrix File

This file can seem very daunting so I have chosen a complicated example so I can prepare you for the level of detail you might need if you choose to model a sexual, stage-structured population with 5 stages, as we have in this example. Remember that the basic structure of this Leslie matrix is

	Stage 0	Stage 1	Stage 2
Stage 0	0	0	ϕ_2
Stage 1	1	$\sigma_1(1-\gamma_1)$	0
Stage 2	0	$\sigma_1(\gamma_1)$	σ_2

Where σ_1, σ_2, are the survival rates for stages 1 and 2, respectively, and γ is the transition probability for Stage 1 to Stage 2. This is what an asexual transition matrix looks like.

As you can see from the image below, you need a column for the males and females of each stage, and a row for each as well (with the exception of stage 0, because they will always have to transition to stage 1, they can't remain stage 0 as this is specifically for the dispersing stage). Let's go through the calculations in detail.

In this case, we are using a survival rate of $\sigma_1 = 0.63$ for all settled stages. Referring to the template transition matrix, that means $\gamma_1 = 0.79$ because $\sigma_1(1-\gamma_1) = 0.1323$ and $\sigma_1(\gamma_1) = 0.4977$ for males and females of Stage 1. That means that an individual in Stage 1 has a 79% chance of transitioning to Stage 2. Repeat this calculation for the values associated with Stage 2, using a transition probability of 0.98 (so only 2% of Stage 2 individuals remain at Stage 2). Notice that the values associated with Stage 3 are slightly different. Here, we are assuming that 100% of Stage 3 individuals will transition to Stage 4, so the value of Stage 3 individuals that remain Stage 3 =0, and the value of Stage 3 individuals that transition to Stage 4 is 0.63 because the only thing acting on them is their survival rate. This can make it confusing when compared with Stage 4 individuals,

because there is no transition rate. They stay Stage 4 until they reach their maximum age of 10 years (see Population File), experiencing a survival rate of 0.63 every year.

```
transmat_sexual.txt - Notepad
File  Edit  Format  View  Help
Transition   0m    0f    1m       1f       2m       2f       3m    3f     4m     4f      MinAge  MinD  MaxD
0            0     0     0        0        1        3.7      1     4.55   1      12.05   0       -9    -9
1m           1     0     0.1323   0        0        0        0     0      0      0       0       -9    -9
1f           0     1     0        0.1323   0        0        0     0      0      0       0       -9    -9
2m           0     0     0.4977   0        0.0126   0        0     0      0      0       1       -9    -9
2f           0     0     0        0.4977   0        0.0126   0     0      0      0       1       -9    -9
3m           0     0     0        0        0.6174   0        0     0      0      0       2       -9    -9
3f           0     0     0        0        0        0.6174   0     0      0      0       2       -9    -9
4m           0     0     0        0        0        0        0.63  0      0.63   0       3       -9    -9
4f           0     0     0        0        0        0        0     0.63   0      0.63    3       -9    -9
```

It's worth mentioning that the value for Stage 0 to Stage 1 =1, and this is because we are assuming no annual mortality (dispersal-related mortality is applied elsewhere). This is for computational efficiency as this means we don't create individual objects in the model that aren't needed. To be clear, this is **not dispersal-related mortality**, which is often extremely high. That is detailed in the Transfer File section.

The next thing to pay attention to is the fecundity. For the female columns, this is straight forward as it is the number of offspring a female can produce in each stage (not that these don't have to be whole numbers as the true number of offspring will be sampled from a Poisson distribution every reproductive cycle). The important thing to note is the value for males in each stage, since we still need a method of indicating that males are reproductively active, even if they don't produce any offspring directly. So here, Stage 1 males and females are not reproductively active, even though they are settled. This is sometimes referred to as a subadult stage or immature juvenile stage.

The next column details the minimum age at which they are able to transition into the next stage. This can get a little confusing and tricky. The minimum ages for Stage 0 and Stage 1 **have to be 0**, because dispersers never remain in Stage 0 after undergoing dispersal, they all transition. After this, you can decide how many years they remain in each stage before the transition probability is applied (though the survival rate will still be added each year). Here, individuals only stay in an age class for minimum one year before being able to transition, so they are able to reach their maximum fecundity by the age of 3.

The last two columns are for stage-specific depth preferences (as indicated in the Stage Struct File). Here, we don't have this implemented but you could add in minimum and maximum depth preferences for each stage if it was relevant.

Now we move on to the dispersal-specific input files.

Emigration File

Since dispersal is modelled in three distinct stages –emigration, transfer, and settlement—each phase has its own input file. Emigration is all about leaving the natal patch.

Density dependent emigration

In this example, we have not implemented density-dependent, stage- or sex-dependent emigration. The stage that emigrates in this case is Stage 0 with emigration probability 0.8.

Simulation	DensDep	StageDep	SexDep	IndVar	iiv_sd	EmigStage	Stage	Sex	EP	D0	alpha	beta
0	0	0	0	0	-9	0	0	-9	0.8	-9	-9	-9

Because you have the option of making emigration stage- and sex-dependent, the format of the input file can be slightly different. If you had stage-dependent emigration, for example, you would need a new line for each stage so the information following the Stage column would pertain to that specific stage. Similarly, if emigration were sex-dependent, a row per sex would be needed. Therefore, Simulation 0 could be comprised of several rows, depending on the combination that is relevant.

Simulation	DensDep	StageDep	SexDep	IndVar	iiv_sd	EmigStage	Stage	Sex	EP	D0	alpha	beta
0	0	1	1	0	-9	-9	0	0	0.2	-9	-9	-9
0	0	1	1	0	-9	-9	0	1	0.4	-9	-9	-9
0	0	1	1	0	-9	-9	1	0	0.6	-9	-9	-9
0	0	1	1	0	-9	-9	1	1	0.8	-9	-9	-9

In this example, emigration is stage-dependent *and* sex-dependent, so each stage has two rows dedicated to it: one for male (sex 0) and one for female (sex 1). **Note** that the male information is always given first so be aware that your rows are in the right order, or MerMADE will assign the values to the wrong sex. If you choose stage-dependent emigration, you need to include a row for each stage in your model, not only for the stages that emigrate. So, if you had 5 stages but only stage 0 and 1 emigration, you still need 5 rows. In the above example, we have not implemented density-dependent emigration, but these parameters could also be sex- and/or stage-dependent.

Transfer File

The transfer phase is the one with the most possible parameters so I will break it down as much as I can. This file will take the same format as the Emigration File above if transfer is sex- or stage-dependent in that you need a row per sex and a row per stage. The following example will not implement this but keep it in mind.

The first section is much like the Emigration File in that it specifies whether transfer parameters are sex- or stage-dependent and which stage/sex this row pertains to. The additional parameter of DistMort indicates whether individuals experience distance-related mortality (ie the further they travel from their natal patch, the higher the mortality probability will be). In this case, we don't have that implemented.

Simulation	StageDep	SexDep	DistMort	Stage	Sex
0	0	0	0	0	-9

The next section starts nailing down the behaviour of dispersers. **Pay attention** to the differences between size- and time-dependent dispersal in this input file, as the columns require different parameters corresponding to this distinction.

Time-dependent dispersal

In a Wrapped Cauchy distribution, ρ (rho) is the concentration parameter that indicates how tightly to follow the current (remember that the model works off of a Wrapped Cauchy distribution and uses the direction of the current as the mean in that circular distribution. ρ tells MerMADE how closely to stick to that mean). For time-dependent dispersal, the **Rho column in the input file is simply this concentration parameter** and can be any value between 0 and 0.9 (not 1, to maintain a little bit of variation). The closer to 0.9, the tighter the distribution and the closer the individual is to

passively following the current. On the flip side, the closer to 0, the more active the individual is assumed to be and can choose a wider range of directions, even going back on themselves if ρ < 0.5. In time-dependent dispersal, step length (SL) is the distance in cm/s that the individual can travel; it is its swimming speed.

Larval behaviour during dispersal

At this point, we have to talk about behaviour during dispersal. You have the following options: 1) completely passive for the entirety of dispersal (including for settlement, individuals are completely passive), 2) hybrid dispersers with a passive phase and an active phase (with the option of adding diel vertical migration in between), and 3) active dispersers which have a swimming speed from the beginning and have no passive phase at all. How you fill in the next section of this input file will depend on which of the above scenarios you choose.

1) For **passive dispersers**, $\rho = 0.9$, SL=1 and comp_SL=-9. This combination of values tells the model that this is a passively dispersing individual, following the current with $\rho = 0.9$, a swimming speed of 1 cm/s (which is negligible enough to pretty much ignore), and no active swimming speed (comp_SL) because it remains passive throughout the dispersal phase.

Transfile.txt - Notepad

File Edit Format View Help

Rho	Rho_sd	SL	comp_SL	SL_sd	min_active_time	min_active_size	active_sd
0.9	-9	1	-9	-9	-9	-9	-9

The Rho_sd column is for cases where inter-individual variation is implemented and this parameter can evolve. This would be the standard deviation for a normal distribution with ρ as the mean. Since individuals don't become active at any point, the value for min_active_time=-9 and we don't let either SL or active_time evolve so the standard deviation columns for those parameters remain -9 as well.

2) For **hybrid dispersers**, the Rho column needs a value other than 0.9 and it is **the value of ρ after disperser becomes active**. In this example, $\rho = 0.6$ **after** individual becomes active at 10 days (min_active_time), at which point the swimming speed (comp_SL) becomes 10 cm/s. Note that SL still needs to be 1 to show there is a passive stage before activity so that MerMADE automatically assigns $\rho = 0.9$ for the first 10 days of dispersal.

Transfile.txt - Notepad

File Edit Format View Help

Rho	Rho_sd	SL	comp_SL	SL_sd	min_active_time	min_active_size	active_sd
0.6	-9	1	10	-9	10	-9	-9

3) For **active dispersers**, the Rho column indicates the value of ρ for the entire dispersal phase. In this example, $\rho = 0.6$ and SL=10 cm/s immediately, because comp_SL=-9 (indicating there is no passive phase before activity) and min_active_time=0, indicating that on day 0, individuals are already active.

Transfile.txt - Notepad

File Edit Format View Help

Rho	Rho_sd	SL	comp_SL	SL_sd	min_active_time	min_active_size	active_sd
0.6	-9	10	-9	-9	0	-9	-9

Size-dependent dispersal

For size-dependent dispersal, Rho in the input file is actually a **slope** because how closely an individual sticks to the current is dependent upon its swimming ability, which develop as they grow. Therefore, for size-dependent dispersal, we put the slope m of the relationship between body length and ρ, following the format of $\rho = m(length) + 1$, for the distribution into this slot in the input file. In this example $m = -0.008$, which seems like a strange value, but that's because it is dependent upon the type of growth function you're using and how quickly individuals grow, what size they start at, as well as how long their pelagic larval duration is. This slope is usually negative, as the smaller an individual is, the closer its ρ would be to 1 (which is why we use 1 as the intercept).

The other difference is that for size-dependent dispersal, SL is a **multiplier** not an absolute value. In this example, this input file codes for a hybrid disperser (see above section for details on larval behaviour during dispersal), which means that after a passive period (because SL=1 and comp_SL has a value), the disperser has a swimming speed of 2 x length, since comp_SL=2. This change happens when the individual is 26 mm in length, as the min_active_size column states, meaning that when the individual begins actively dispersing, it will have a swimming speed of 52 mm/s (because size is measured in mm). MerMADE will then calculate the appropriate ρ for that size, which in this case would be 0.584 (because $\rho = m(length) + 1$).

Transfile.txt - Notepad

File Edit Format View Help

Rho	Rho_sd	SL	comp_SL	SL_sd	min_active_time	min_active_size	active_sd
-0.008	-9	1	2	-9	-9	26	-9

The same thought process for which columns to fill in for which larval behaviour combination applies here as applied to time-dependent dispersal, detailed above. The main differences are that Rho is a slope, SL is a multiplier of body size, and you need to indicate a minimum active size instead of time. Otherwise, they operate the same.

Diel Vertical Migration

The other aspect of larval behaviour we have not covered is diel vertical migration, which is what the next section of columns pertains to. The first two columns indicate buoyancy limits during dispersal. Here, individuals can move anywhere between 0-80m depth. Diel_vert=1 means that dispersers undergo diel vertical migration **which only happens if there is a passive stage.** This means that it only applies to passive and hybrid dispersers, **not** immediately active ones. The next parameter, dv_active, indicates whether diel vertical migration should continue past the passive stage in hybrid dispersers (dv_active=1) or whether it should cease as soon as active dispersal is reached (dv_active=1). This process is done very simply in MerMADE in that you define dv_range, which will be the depth range **below minimum** and **above maximum** for night and day, respectively. MerMADE operates simply on a 12-hr day/night cycle and larvae are released at night at the start of dispersal. If diel vertical migration is implemented, they will switch every 12 hours from being towards the surface (night) to being towards their maximum buoyancy limit (day).

In this example, dv_range=5. That means that at night, when individuals are at the surface, they can be anywhere between 0-5m depth. When they switch, they will be between 75-80m depth, since 80m is the maximum buoyancy limit.

File Edit Format View Help

buoy_min	buoy_max	diel_vert	dv_range	dv_active	min_dv_time	min_dv_size	min_dv_sd
0	80	1	5	0	-9	10	-9

Just like with disperser activity, the min_dv_time and min_dv_size columns code for the timing of when diel vertical migration should begin for time-dependent and size-dependent dispersal, respectively.

The final section of this input file codes for a few last things. The only ones you need to worry about are down_bias, memory and MortProb. Down_bias indicates whether, when individuals become active, they tend to follow the seafloor (or their lower buoyancy limit, whichever is shallower). This might apply, for example, if individuals seek suitable habitat or swim close to the bottom for safety as they do so, rather than staying up in the water column. This may or may not reflect the behaviour of your study species and you can control this via down_bias.

Memory is a parameter that serves to straighten movement paths. If an individual can remember its last few steps, then the direction of travel will be calculated from its earliest remembered position, not its most recent one, so it will tend to continue in a similar direction. If an individual can't remember its movements very well, it might have very circuitous paths and could double back on itself as it forgets where it has already been.

In this example, we have implemented down_bias, so the individual will track the seafloor, and we've given it a memory=3, which means it remembers 3 steps (or the last three hours as steps are calculated hourly).

File Edit Format View Help

down_bias	DP	Memory	MortProb	Slope	InflPoint
1	0	3	0	-9	-9

MortProb is the mortality that dispersers suffer during dispersal. Generally speaking, larvae dispersing pelagically will incur a very high dispersal-related mortality (which is why the number of larvae produced is often so high). This number might well be >0.9 in many cases. Remember this is **not annual mortality**, this is additional and only happens during dispersal. Mortality is applied at every timestep, which is hourly. Often, mortality probabilities will be given daily, so remember to divide by 24 to make this hourly.

DP is a remnant of implementation of directional persistence but is no longer a valid parameter as memory fulfils the same purpose (but you still need this column as this has not been erased from the source code yet so MerMADE will get confused if the column is missing). Similarly, Slope and InflPoint pertain to DistMort, distance-dependent mortality, which isn't able to be implemented at present.

Growth File

This file is only necessary if you are modelling size-dependent dispersal, as it contains information for the growth function applied during dispersal.

This format should look fairly familiar now. The first section codes for simulation number in case you are running several simulations in one batch, whether growth is stage-dependent or sex-dependent and which stage/sex this line

File Edit Format View Help

Simulation	Stage_Dep	Sex_dep	method
0	0	0	1

pertains to. There will need to be a line per sex and a line per stage if either of these are true.

The Method parameter has two options: linear (method=0) or Gompertz model (method=1).

The linear method is available if you want to implement a very simple growth pattern over time. It is simply

$$l_t = m(t) + l_{t=0}$$

Where l_t is length at time t, m is the slope of the relationship, t is time **that has already passed** (in days), and $l_{t=0}$ is length **at the start of dispersal** (which is column b in this input file). Again, m_sd allows the slope of this relationship to evolve with inter-individual variability.

The Gompertz method is available if your growth relationship is slightly more complex. **Note** that this is a **modified** Gompertz relationship taken from Regnier et al 2021 (in prep). This equation is as follows:

$$l_t = l_{t=0} + l_{inf} \left(e^{-e^{-G_K(t-T_i)}}\right)$$

where $l_{t=0}$ is the **size at hatching**, l_{inf} is the maximum length an individual can reach, where this function reaches an asymptote, G_K is a growth parameter, t represents the **time left for dispersal** until PLD is reached, and T_i is the inflection point or earliest date of settlement. In this example, we have said that an individual can reach 67mm, earliest date of settlement is 53 days, and the growth parameter is 0.03696889 (note that this was a value specific to sandeels, available in Regnier et al 2021, in prep). This relationship will look something like this:

Growth only occurs during dispersal, so the model stops tracking an individual's size after settlement.

Settlement File

This file codes for the third and final phase of dispersal: settlement. The first section of this input file operates much the same as the above detailed files in that you decide whether settlement is stage, sex-, or density dependent (or a combination thereof) and you indicate which sex and stage each line of the file pertains to.

The next section indicates timings, behaviours and probabilities. FindMate tells MerMADE what you whether you want the individuals to only settle if there is a potential mate of the opposite sex already established and settled in that cell/patch. This is of course only applicable in sexual models where both sexes are present. If this parameter is 0 (no), as it is below, then only habitat type (therefore suitability) is used to make settlement decisions. Otherwise, if it's yes (1), then a disperser won't settle unless an individual of the opposite sex is present.

PLD is the pelagic larval duration (in days), which is the maximum amount of time dispersers have to settle. Once they pass this allotted time, all dispersers that haven't settled die and are taken out of the simulation. Comp_time is the competency period (also in days) after which dispersers are

competent enough to be able to settle (for time-dependent dispersal). Before this point, they may be active (see section above) but even if they come across suitable habitat, they are not able to settle. For size-dependent dispersal, the parameter comp_size indicates the minimum size individuals need to reach (in mm) to be considered competent.

S0 is the settlement probability the individuals have when they come across suitable habitat (or suitable habitat with a potential mate, if that is a requirement you set). Here, it is 1 because we want them to settle with certainty, but this can be any value between 0-1. AlphaS and BetaS are only applicable if settlement is density dependent, they are the scaling factor and the inflection point of the density dependence relationship between number of individuals already present and settlement probability of newcomer.

$$ p_s = \frac{S_0}{1 + e^{-(bN_i - \beta_s)*\alpha_s}} $$

where b represents the strength of density dependence used for population dynamics equations, S_0 is the maximum settlement probability, N_i is the number of individuals at Patch i, α_s is the slope of the reaction function and β_s its inflection point. As an example, we chose values for S_0, α_s, and β_s that produced this relationship between density and p_s, using $b = \frac{1}{75}$, that looks something like this:
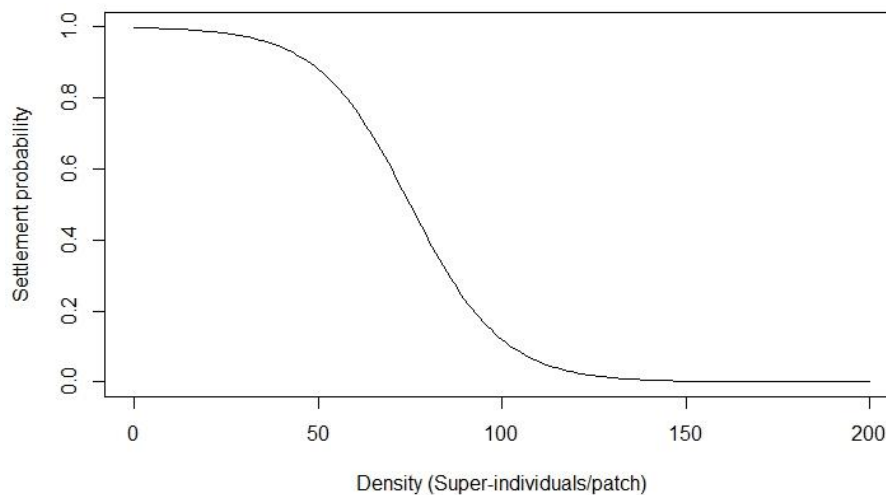
The remaining parameters in this input file code for the behaviour of seeking out suitable habitat. the buffer parameter (0=no, 1=yes) tells MerMADE whether individuals are able to detect suitable habitat from a distance. If not, then the rest of these parameters are -9 and individuals will have to stumble across a suitable site by chance. If yes, then buffer_xycells tells MerMADE from how far away in the horizontal dimension an individual can detect this (in cells). Since this is in cells, the actual distance depends on your horizontal resolution. Here, an individual can detect suitable habitat from 3 cells away in the horizonal dimension, and since in this example we are working with a resolution of 1500m, that means individuals can start swimming in the direction of suitable habitat from 4.5km away. The same principle applies for the buffer_zlayers parameter, but since the vertical resolution will most likely be much finer (ie in our case it's 10m), we need to code it separately. Here, we've said 2 layers so that individuals don't simply swim over it but will dive down to meet it from 20m above. The buffer_cond parameter operates similarly to the FindMate parameter in that it tells MerMADE whether individuals will

SettFile.txt - Notepad

File   Edit   Format   View   Help

```
buffer   buffer_xycells   buffer_zlayers   buffer_cond   buffer_thresh
1        3                2                0             -9
```

consider a site suitable at a distance if it is simple the correct habitat type (=1), whether there are conspecifics present (=2) and whether there are *enough* conspecifics present (=3), meeting a threshold that is determined by the buffer_thresh parameter.

If the decision to settle is made, the individual is incorporated into the local population. We assume that, once settled, individuals do not move from their settlement site. In case such as sessile species, this is biologically accurate, but we acknowledge that mobile species may move around their new patch or cell. However, since population dynamics occur on the scale of the cell or patch, we do not believe this will have a significant effect on overall trends. Offspring of these settled individuals are given their mother's settlement coordinates from which to start their dispersal journey.

That is the last of the files coding for dispersal parameters. Take a breath.

Evolution File

This section is only relevant if you want to include evolution *by mutation* for dispersal parameters. This goes beyond adding inter-individual variability in each parameter, which is achieved by indicating a standard deviation in the respective file. Over time, that population might well shift towards the boundaries of the values you've allowed, but will not change much beyond that. Allowing evolution by mutation opens the floor for a wider spectrum of change within your population.

The dispersal parameters that are allowed to evolve are:

- Emigration probability (both density-dependent and density-independent)
- Minimum size/time for active dispersal
- Minimum size/time for diel vertical migration
- Growth parameters (slope and intercept for linear, maximum size, growth factor and settlement date for Gompertz)
- Minimum size/time at competency
- Settlement probability (both density-dependent and independent)

In this input file, each of these parameters has the same associated columns: does it evolve (answered with a 0 or 1), if yes, what is the mutation size and mutation rate/probability. This means that if a mutation occurs (with some probability), how much is each parameter allowed to change. This needs to be specified for each parameter because the scales of each are so variable that each needs its own mutation size. For each group of parameters (ie emigration or growth parameters), the same mutation probability applies. I have included an example of this set up for a couple sets of the parameters, and hope you can extrapolate from that to the rest of the input file.


```
EvolutionFile.txt - Notepad
File  Edit  Format  View  Help
Simulation      EP_evo  EP_mut_size     EP_mut  alpha_mut_size  beta_mut_size
0               1       0.05            0.1     -9              -9
```

Initialisation File

Now you have given the model all the information it needs to run. The only thing left is to tell it where to start. The initialisation file tells the model how many individuals to initialise at the very start of the model, how to do so and where to put them.

SeedType has three options: free initialisation (0), from species distribution (1) and from a specific patch (2). This tells MerMADE what information to use to place the very first individuals in the seascape. The following columns pertain to free initialisation (SeedType=0).

InitFile.txt - Notepad
File Edit Format View Help

Simulation	SeedType	FreeType	FreeProp	minX	maxX	minY	maxY	minZ	maxZ
0	0	1	-9	-9	-9	-9	-9	-9	-9

FreeType has two options: random, given a percentage of suitable patches/cells (0) and all suitable patches/cells (1). If FreeType=0, then the proportion of suitable patches/cells is coded for in FreeProp. This can be any number between 0-1 (although if it is 1, then you might as well have chosen FreeType=1). You can designate this further by indicating *which section* of the seascape this proportion should be of. The minimum and maximum x, y, and z coordinates you enter will tell the model that a proportion of *that* section of the seascape should be initialised. If you want to apply the proportion to the entire seascape, then keep these minimum and maximum values =-9. **Note** that x and y coordinates are in **columns and rows** and that counting starts in the top left corner of the seascape. This is **not** the coordinate in metres but in **columns and rows**. It might feel counter-intuitive to give the x coordinate in columns but remember that this the correct association. The z coordinate is then in **layers**. In this example, we have started it with free initialisation, all suitable patches and we have included the entire seascape in the initialisation.

If we had chosen SeedType=1, we would fill in the next columns which code for initialising from species distribution. SpType operates like FreeType and has two options: initialise a randomly chosen percentage of the patches/cells within the distribution (0) or initialise all cells/patches within the distribution (1). SpProp is then the proportion of patches/cells for SpType=0. In this example,

InitFile.txt - Notepad
File Edit Format View Help

SpType	SpProp	InitDens	IndsHa	WhichPatch
0	0.7	1	-9	-9

we've chosen to initialise 70% of the species distribution.

Now that we have established *where* the individuals will be initialised, we need to tell MerMADE *how many* of them we would like. The next parameter, InitDens, has three options: initialise at full carrying capacity (0), half carrying capacity (1) or a specified density in individuals/hectare (2). This pertains to the carrying capacity of the cell/patch that is chosen to be initialised, which already carries this information from the Seascape file. If we've said we'd like to initialise at a specified density, then IndsHa is where we would indicate that density.

The last option for SeedType was to initialise a specified patch, which goes under the WhichPatch column and the density information applies to that patch as well. Here, you can only initialise one specific patch and you simply indicate its Patch ID (which patches that in the Patch file).

The last thing to specify is some demographic information. Now we know *how many*, we want to know *what kind* of individuals. The first thing, if you are running a sexual model, is to indicate what

InitFile.txt - Notepad
File Edit Format View Help

PropMales	InitAge	PropStage1	PropStage2	PropStage3	PropStage4
0.5	1	0	0	0.5	0.5

proportion of the starting population is male. It is not uncommon to find gender biases in populations. The next question is how old are the individuals you're initialising? InitAge has two options: lowest possible age (0) and randomised (1). If randomised, then any age between 1 and the

max age you indicated in the Population File will be chosen. If lowest possible age is chosen, this will also depend on whether your population is stage structured or not. If not, then they

This only applies to stage-structured populations and the **number of columns depends on the number of stages you have**. Note that we have 5 stages in our model but you **can't initialise Stage 0** because those are the dispersers and they are produced when the individuals that you do initialise reproduce. Therefore, we have 4 columns for Stages 1-4. Here, we've indicated that we only want to initialise individuals in the older two stages, 50-50 split between the two of them. The InitAge parameter will then assign ages appropriate for these stages (based on the age information entered in the transition matrix).

Control File

This is the last file you need to make (even though it is the first file that MerMADE will open) because you have now made all the decisions necessary so you can tell MerMADE up front what your simulations will look like.

The Control File is what tells MerMADE the highest level information of what structure the model run will take so that it can create the necessary objects to the specified size, ready to be filled with the information in upcoming input files. This basically gives answers to questions like "How big is the seascape?" so MerMADE can create enough cells in the right configuration, and "What kind of information is needed?" so it knows to look for a species distribution file if necessary, for example, but if it isn't needed then it won't waste time searching. It also contains all the names of the input files that the model will read. BE SURE THESE NAMES MATCH THE ONES IN YOUR FOLDER. If they don't, MerMADE will not know where to look and your simulation will fail. The files can be named anything you like, as long as these names are given correctly in the Control File **in this exact order**. MerMADE reads in the input sequentially so if your input files are out of sequence, it will assign the wrong information to the wrong parameters. If you run into trouble with your simulations, this is a good place to start double checking your inputs.

Finally, the Control File tells MerMADE what outputs are expected from the model run so that it can create the files, make sure to keep track of the needed information throughout, and output them at the correct intervals. Notice here that no/yes questions are often answered with 0/1, respectively. Where this is needed is explicitly stated in the Descriptions.xls file included in your Tutorial package.

In this example case, we are running a single patch-based simulation with only 1 replicate for 2 years. There are 3 habitat types and we are not initialising from a species distribution. The population is stage structured with 5 stages, sexual reproduction (as opposed to asexual or complex reproduction involving harems) with size-dependent dispersal and we are not implementing temperature-dependence. Note that this simulation will have dynamic seascapes so the dyn_landfile is provided. We want MerMADE to provide all possible types of output files (see section on output files) every year.

That's all the input files done!

Output Files

In the Control File, you indicated what kinds of outputs you would like MerMADE to create from the simulations. This section shows you what those outputs look like and what they can be used for.

Population File

This output file contains population level metrics about individuals and parameters. For each replicate, and each year within that replicate, and each reproductive season within each year, and for each cell/patch that has an active subpopulation in it, you get the following information:

- Number of individuals in that cell/patch
- Number of females and males present (this will be -9 if the reproductive method is asexual)
- If evolution of dispersal parameters is implemented, population means of those parameters

In this example, it's a patch-based model so for each patch (patches 0-4 shown here), we see the above-mentioned information. Here, we have allowed emigration probability to evolve so the

output file has shown us the mean emigration probability for each patch. Note that this is only in Year 0 so not much mutation has occurred, the differences in means reflect the inter-individual variation present at initialisation. If we scrolled down in the output file for later years, we would see the change in mean emigration probability over time.

Analyses of this output file could include changes in gender biases (for example, if there is sex-dependent dispersal or survival rates), changes in overall population size as well as spatial patterns of subpopulation size change, range shifts as new areas are colonised and some cells/patches go extinct, and of course, the evolution of dispersal parameters.

Range File

This output file indicates the extents of the occupied range of the species over time.


```
Range.txt - Notepad
File  Edit  Format  View  Help
Rep     Year    Rep_Season    NOcc    OccupSuit    Min_X       Max_X       Min_Y    Max_Y       Min_Z    Max_Z
0       0       0             104     1            5.8615e+06  6.2545e+06  697000   1.039e+06   10       90
0       1       0             104     1            5.8615e+06  6.2545e+06  697000   1.039e+06   10       90
0       2       0             104     1            5.8615e+06  6.2545e+06  697000   1.039e+06   10       90
```

Fore each reproductive season in each year in each replicate, you get the following information:

- NOcc is the number of occupied cells/patches within the seascape
- OccupSuit is the proportion of available suitable cells/patches that are occupied
- The minimum and maximum X, Y, Z coordinates (in metres)

Here, we initialised all available patches within the seascape so there was no increase in occupied patches possible, therefore the NOcc is 104 for all years, the OccupSuit is 1 (100% occupied) and the minimum and maximum x, y, and z values don't change. However, if you were to initialise only a portion of the seascape or you initialise from a species distribution and there are unoccupied patches present, these numbers may fluctuate.

Analyses of this output may include progression of range shifts, comparing simulations with different dispersal abilities for which allows the species to colonise new suitable habitat most effectively, or comparing reintroduction simulations for which enables establishment the best.

Individual File

One of the great benefits of working with IBMs is that you can get individual-level information for every individual in your simulation if you choose to. This output file gives us information on:

- Where each individual started
- Whether it underwent dispersal and if so where it ended up and how far it travelled
- What was its cause of death
- What was its size and evolving dispersal parameter value at time of death

The first few columns of this file are as seen in the above detailed output files: Rep, Year, and RepSeason so I have not included them again here, but they are present.

Ind_ID	Sex	Status	Starting_patch	Natal_X	Natal_Y	Natal_Z	Ending_patch	X	Y	Z
0	0	6	0	871262	6.21166e+06	30	0	871262	6.21166e+06	30
1	1	6	0	870163	6.20703e+06	30	0	870163	6.20703e+06	30
2	0	6	0	872712	6.21064e+06	30	0	872712	6.21064e+06	30
3	1	0	0	875894	6.2119e+06	30	0	875894	6.2119e+06	30
4	0	0	0	873522	6.21044e+06	30	0	873522	6.21044e+06	30
5	1	0	0	873270	6.21173e+06	30	0	873270	6.21173e+06	30
14970	0	2	0	870932	6.20739e+06	30	0	870932	6.20739e+06	30
14971	1	2	0	870932	6.20739e+06	30	0	870932	6.20739e+06	30
15000	1	2	0	870436	6.20772e+06	30	0	870436	6.20772e+06	30
15012	1	3	2	863238	6.19812e+06	30	0	869455	6.20898e+06	30
28981	0	3	72	871582	6.19753e+06	30	0	873151	6.20947e+06	30
28993	0	3	72	862594	6.1912e+06	30	0	870937	6.20793e+06	30

Ind_ID is the unique ID number of the individual the line pertains to. Sex is the gender of the individual (0=male, 1=female). Status tells you what fate that individual met in terms of dispersal and mortality. There are the following status options:

0. Initialised individual but has not met any fate yet (this is usually from the original initialisation, therefore it couldn't disperse, and that individual hasn't suffer mortality yet)
1. Disperser (emigrated as part of the emigration probability)
2. Local recruit (product of the emigration probability, if EP=0.8 then 20% of offspring will be local recruits)
3. Disperser that settled in suitable habitat
4. Disperser that was forced to settle on unsuitable habitat and died (this mainly applies to passive individuals that have no ability to influence their destination)
5. Disperser that died by dispersal-related mortality (as specified in the Transfer File)
6. Individual died by annual mortality (as defined in the transition matrix)
7. Individual died by exceeding max age (lived a full life until the maximum age was reached)
8. Disperser that died by being absorbed by the seascape boundary (since there are no physical boundaries, open water edges of the seascape are absorbing not reflective and dispersers are lost)
9. Disperser that was forced to settle on suitable habitat (also only applies to passive individuals to distinguish between individuals that just "got lucky" by hitting suitable habitat and dispersers that go through the full dispersal process)
10. Dispersers that had not found suitable habitat by the time PLD was reached
11. Dispersers that died by suffering fishing mortality

If these statuses seem out of order and tedious to you, that is because I added new ones as I needed them. Some of the distinctions might also seem pedantic but they are very helpful in understanding exactly how individuals are dispersing and where they might be running into difficulties. True, this was especially true when I was debugging the model but they are still useful when running various trials and tests on dispersal parameters with your species.

DistMoved	Size	EP
0	5.3	0.788138
0	5.3	0.856478
0	5.3	0.746937
0	5.3	0.865456
0	5.3	0.852185
0	5.3	0.810165
0	5.3	0.761075
0	5.3	0.800753
0	5.3	0.888815
31522.3	8.53221	0.775846
26493.3	8.90814	0.841442
37211.6	9.74335	0.831343

The output file then tells you in which cell/patch the individual started, its exact x, y, z coordinates at birth and those data for where it ended as well. The next columns tell you how far it moved and any parameters that had inter-individual variation or were allowed to evolve. Note that the distance moved is not simply calculated from the starting and ending coordinates (ie Euclidian distance) but takes into account every step the individual made during dispersal, which may have had a very circuitous path).

Note that the individual ID numbers might seem like they jump around, but that is because the data is written by going through the patches, so will include individuals that originated there and then the newcomers from potentially several generations later. This is why, for example, the Ind_ID column could jump from ID 5 to ID 14970.

Analyses of this output would most likely be for constructing connectivity matrices over time

Individual Movement File

This output file is only necessary if you want to recreate the movement paths (in R, for example). Each individual (in each reproductive season, year and replicate, as above) is dedicated three lines of this output: one for the sequences of X coordinates, one for the Y and one for the Z (all in metres). Each triplet of coordinates shows a step that that individual took on its dispersal path. The individual below, individual 947, was female and began her journey at location 64307, 634959,70. Her first step was to move upwards 7m, then incrementally moved in all three dimensions for the remainder of the dispersal phase.


```
Indiv_Mov_rep0.txt - Notepad
File  Edit  Format  View  Help
Rep     Year    Rep_Season      Ind_ID  Sex     Status  XYZ_coords
0       0       0               947     1       3       643207  643207  643170  643130  643081  643037  642988  642974  642991  642883
0       0       0               947     1       3       634959  634959  634948  634937  634927  634916  63490   634894  634882  634878
0       0       0               947     1       3       70      63      63.0172 63.0345 63.0517 63.0689 63.0862 63.1034 63.1206 63.131
```

Be careful at what frequency you output this information. Depending on the number of individuals in your model and how long the PLD is, this file can get HUGE. This is why there is an output file **per replicate** while the other outputs have all the replicates in one file. Even this way, these files can easily reach >500mb in size, which programs will struggle to open and will even take R a while to import, so be aware of this. Outputting this information is incredibly valuable for visualising the exact paths your individuals are taking to disperse, but it is not necessary to have this information yearly in most cases.

Tutorials

Example 1: Passive, non-overlapping generations, cell-based

Set up

Input files

Seascape

Population dynamics

Emigration

Transfer

Settlement

Initialisation

Results

Example 2: hybrid, stage structured, patch-based, fishing mortality

Example 3: active, stage structured, patch-based, dynamic seascapes, local patch extinction

References

Bocedi, G. et al. 2014. RangeShifter: A platform for modelling spatial eco-evolutionary dynamics and species' responses to environmental changes. - Methods Ecol. Evol. 5: 388–396.

Bocedi, G. et al. 2020. RangeShifter 2.0: An extended and enhanced platform for modelling spatial eco-evolutionary dynamics and species' responses to environmental changes. - bioRxiv: 1–26.

Malchow, A. K. et al. 2020. RangeShiftR: An R package for individual-based simulation of spatial eco-evolutionary dynamics and species' responses to environmental change. - bioRxiv in press.