1 Method

To simplify reality, neutral models assume individuals are ecologically equivalent. This does not necessarily mean individuals have identical traits, it means trait variation is independent of species identity. The models consist of a community and proceed in discrete, uniform time steps. In each step, a randomly-chosen individual dies. With probability v, they are replaced by a new species. With probability 1 - v, they are replaced, via dispersal, by the offspring of another randomly-chosen individual. For convenience, the models make a zero-sum assumption: Birth and death balance (dead individuals are immediately replaced), so the number of individuals is constant (Hubbell 2001; Rosindell et al 2008, 2011).

I begin with a two-dimensional, spatially explicit version: Individuals occupy cells in a grid, representing positions in space. When individuals reproduce, offspring disperse according to a dispersal kernel - a probability distribution of dispersal distances. In a basic model, death, birth, speciation, and dispersal rates do not vary across individuals.

I add an altitudinal temperature gradient that drives variation in death, birth, and dispersal, as predicted by Metabolic Theory. To explore the effect of area, I vary the number of individuals in a cell. Finally, I envision the community consists of guilds; individuals in a guild have the same body size. Each guild is a separate simulation. Among guilds, dispersal ability and the total number of individuals differ, as predicted by allometric scaling. So, temperature drives variation within simulations, whereas body size drives it across simulations. Initally, the model is fully neutral: In a simulation, species are equivalent - individual variation is independent of species identity. However, as explained later, species have a thermal optimum, so the model moves away from neutrality.

1.1 The Model's Geometry

I use a cone's surface as a model of a mountain. Unfurled, a cone's surface is a circle sector. The polar coordinates, r and θ, describe position on the mountain. The circle centre (mountain tip or cone apex) is the origin, (0, 0). The radial coordinate, r, is the radial distance from the circle centre - how far down the mountain a point is. The angular coordinate, θ, is the angle from the x-axis.

In silico, I represent the cone's surface as a square array (grid of cells). Rows in the array are altitudinal bands, and columns, positions along a band. Row indices correspond to radial positions, and column indices, to angular positions. The array's top edge is the cone's apex (mountain tip), so has radial coordinate 0. The number of rows and columns does not set the size of the model mountan, it sets the spatial resolution (number of positions). individuals can occupy Though the array is depicted as a flat square, it forms a cone: The left and right edges connect, and, going up the mountain, each cell represents an increasingly narrow area.

The cone has three parameters: base radius (x), height (h), and slant height (s). Slant height is the distance along the cone's lateral (curved) surface from the apex to the base. Measured in metres, these set the size of the model mountain, and the area grid cells represent. If c is the ratio of s and x:

$$\frac{s}{x} = c \tag{1}$$

$$s = cx \tag{2}$$

$$x = \frac{s}{c} \tag{3}$$

R is the ratio of h and x:

$$\frac{h}{x} = R \tag{4}$$

The area, A, of a cone's lateral surface is:

$$\pi x s = \pi c x^2 \tag{5}$$

Using Pythagoras' theorem:

$$h^2 + x^2 = s^2 \tag{6}$$

To convert between metres and number of cells:

$$cx \text{ metres} = T_r \text{ cells} \tag{7}$$

$$1 \text{ m} = \frac{T_r}{cx} \text{ cells} \tag{8}$$

$$\frac{cx}{T_r} \text{ m} = 1 \text{ cell} \tag{9}$$

A key advantage of the model is it expresses area and distance as proportions. This means I need not pick absolute values for the cone's dimensions and simplifies the model greatly. By varying the parameters' relative values, I can simulate a variety of mountain topologies. I can explore the generality of diversity gradients, and the comparative importance of ecological mechanisms, across mountains. While this is hard with real-world experiments, it is simple and tractable here.

1.2 Population Density

- an intro to metabolic theory will precede this
- justify/cite values of metabolic parameters - body mass and temperature exponents

- discuss birth/death scaling/maps - context will be scaling of population growth and size (Rmax and K)

Each cell in the array has n individuals; n is a function of body mass and area. Population density (number of individuals per unit area) should decline with body mass as $M^{-0.75}$, if resource supply is constant. This is because individual resource demand depends on metabolic rate, which increases with body mass as $M^{0.75}$. Observations in animals and plants support this (Enquist et al 1998; Damuth 1987). A mountain base covers more area than the top. The model mountain is a cone, but, in silico, it is a square array. So, going up the mountain, each cell in the array represents an increasingly narrow area. If A_c is cell area, the number of individuals in a cell is:

$$A_c M^{-0.75}$$

1.2.1 Area of a Grid Cell

A key advantage of the model is it expresses area in relative terms. This means I need not worry about x's absolute value and greatly simplifies the model. The edge of an altitudinal band is a circle round the cone's surface. Knowing this circle's radius, you can get the area of a grid cell. Imagine the band's edge is the cone's base. The slant height is the band's radial position (distance from apex). Thus, to get the radius:

$$x' = \frac{s'}{c}$$

As row index corresponds to radial position:

$$x' = \frac{I_r}{c}$$

Convert I_r to metres, as x' is in m:

$$\begin{aligned} x' &= \frac{I_r c x}{c T_r} \\ &= \frac{I_r x}{T_r} \end{aligned}$$

A cone's surface area (excluding the base) is:

$$\pi x s = \pi c x^2$$

When a cone is cut by two planes parallel to the base, the shape between the planes is called a frustum. An altitudinal band is the surface of a frustum; the band's edges are the planes. The area of an altitudinal band, A_f, is:

$$\pi c b^2 - \pi c t^2 = \pi c(b^2 - t^2)$$

b and t are the base and top radii of the frustum. Using row index (I_r) and equation 3 to express b and t:

$$A_f = \pi c\Bigg(\Big(\frac{(I_r+1)x}{T_r}\Big)^2 - \Big(\frac{I_r x}{T_r}\Big)^2\Bigg)$$

Then, the area of one cell in an altitudinal band is: divide by the array's width (in # cells) to get

$$\frac{\pi c\Big(\big(\frac{(I_r+1)x}{T_r}\big)^2 - \big(\frac{I_r x}{T_r}\big)^2\Big)}{T_\theta}$$

Thus, cell area is unitless, and instead expressed in terms of x, keeping the model tractable.

1.3 Dispersal

Individuals do not move, but species disperse via birth and death (when an individual reproduces, its offspring fills a gap vacant due to a death). An individual's chance of being chosen to reproduce depends on its birth rate, dispersal ability, and distance from the destination (vacant position). In other words, it is the net probability of birth and dispersal. The challenge is that, across space, birth and dispersal rates vary.

Imagine the origin of a dispersal kernel as being centered on the start point; the kernel describes the distribution of destinations. In a sense, in neutral models, the kernel is backwards, as it centered on the destination (position vacant due to death). From the kernel, the models pick a random distance and direction away from the vacant position. This picks the parent whose offspring occupies the vacancy. While convenient computationally, this only works if dispersal (and birth) rates are fixed. To vary dispersal rate, I must amend this usual algorithm. The solution is a set of 'dispersal maps'.

Before running a simulation, I calculate the probability of dispersing from every cell to the destination - a discrete probability distribution, which I call a dispersal map. As each cell is a potential destination and probability depends on distance, there is a map per cell. Upon death, the model uses the maps to immediately pick a random parent. Dispersal maps are an elegant solution because they capture, in a single step of the model, variation in two traits, across three factors (temperature, body size, and area).

clarity - multiply birth and dispersal maps - net probability

The model uses a standard normal distribution (mean = 0, standard deviation = 1) as a dispersal kernel. The distribution's scale parameter, σ, determines its spread - the likelihood of long-distance dispersal events. It has thin tails, meaning it predicts lower rates of long-distance dispersal than fat-tailed kernels (which curve away from the x-axis). Being a phenomenological kernel, it does not capture complex dispersal behaviour (Clobert et al 2012). But, it is a simple start point to introduce metabolically-driven (deterministic) variation to dispersal.

1.3.1 Metabolic Effect

- again, I will elaborate on/cite the Metabolic Theory in the literature

To apply a metabolic effect to dispersal, I multiply a distance, drawn from the kernel, by a body-size and temperature dependent parameter, y:

$$y = B_0 M^{\alpha} e^{\frac{-E}{kT}}$$

y is proportional to mean dispersal distance, and increases with body mass and temperature - be more specific

- will explain why separate dispersal into horizontal/vertical and that I multiply them

1.3.2 Horizontal Dispersal

Going up a mountain, the distance round it decreases (the base covers more area than the top). However, in silico, the system is a square array - horizontal dispersal distance must be adjusted. Towards the top, individuals should be more likely to disperse among cells, within a row (ignoring the effect of temperature). Also, they should be more likely to complete a revolution round the mountain. So, dispersal ability is unaffected by area, but, as area reduces going up, so does the distance among cells.

The model expresses distance in relative terms. This maximises the model's relevance (the simulated mountain can be any size) and minimises complexity (I need not pick absolute values for the cone's dimensions). An altitudinal band is a circle round the cone's surface. So, horizontal dispersal (left to right or vice versa) occurs along an arc of a circumference. The horizontal distance to the destination, expressed in number of cells and as a proportion, is:

$$\frac{n_\theta}{T_\theta}$$

The distance in metres (d_θ) is:

$$d_\theta = \frac{n_\theta}{T_\theta} 2\pi x'$$

You can obtain, in metres, the radius, x', of an altitudinal band, from the band's row index (see equation X):

$$x' = \frac{S_r x}{T_r}$$

However, area reduces with increasing altitude: So, I use the radius of the altitudinal position halfway between the dispersal event's start and end:

$$x' = \frac{(S_r + E_r)x}{2T_r}$$

$$d_\theta = \frac{n_\theta}{T_\theta} \frac{2\pi(S_r + E_r)x}{2T_r}$$

$$= \frac{n_\theta x \pi (S_r + E_r)}{T_\theta T_r}$$

The dispersal kernel is a normal distribution. To add a metabolic effect to dispersal, I multiply variates of the normal distribution by y, a body-size and temperature dependent parameter. Instead of randomly picking distances from the kernel, I want the probability of dispersing a known distance, d_θ. So, d_θ is the product of y and a variate of the normal distribution. d_θ (horizontal distance in metres to the destination)

$$d_\theta = yV$$

$$\frac{d_\theta}{y} = V$$

(V is a variate of the normal distribution). By evaluating at V the normal distribution's probability density function, you get the probability of dispersing d_θ metres.

So, in summary, the probability of dispersing horizontally, round the mountain, from one column to another is:

$$P\Big(V = \frac{n_\theta x \pi (S_r + E_r)}{T_\theta T_r y}\Big)$$

This depends on distance to the destination (an arc, or proportion, of a circumference), body size, and temperature. It decreases as V increases, as V is a variate of the standard normal distribution (mean $= 0$). Probability increases as body size and temperature increase, and distance reduces. In other words, big individuals, and those in hot places or close to the destination, have a higher chance of reaching the destination.

1.3.3 Revolutions Round the Mountain

1.3.4 Vertical Dispersal

Dispersal up and down the mountain occurs between the apex and base, along the cone's lateral (curved) surface. Unlike the left and right edges, the bottom and top ones are not joined - individuals cannot disperse across them. *implications?*

So inds can only disperse in one direction: up, if they are below the dest, or down, if they are above it. slant height

The vertical distance to the destination, expressed in number of cells and as a proportion, is:

$$\frac{n_r}{T_r}$$

The distance in metres (d_r) is:

$$d_r = \frac{n_r}{T_r}cx$$

cx ($= s$) is the cone's slant height (distance between apex and base). Like d_θ, vertical distance (d_r) is the product of y and a variate of the normal distribution:

$$d_r = yV$$

$$V = \frac{d_r}{y}$$

$$= \frac{n_r cx}{T_r y}$$

The probability of dispersing up or down the mountain, from one row to another is:

$$P(V = \frac{n_r cx}{T_r y})$$

Like d_θ, it depends on distance to the destination (a proportion of the slant height), body size, and temperature.

-conclude dispersal maps - how they're used, their pros (speed) and cons (memory intensive)

1.4 Thermal Optima and the Neutrality Assumption

survival probability

cite Huey, Levin, Angilletta

1.5 Parameter Space Analysis

-summarise model steps and what traits vary within/across simulations -body mass and abundance values

- explicit

-fixing variables (area, temperature)

-parameters - steepness, temp gradient

-will summarise in a table

-sensitivity analysis

Symbol	**Definition**
s	cone's slant height
x	radius of cone's base
c	$\frac{s}{x}$, ratio of s to x
T_r	array's height in number of cells (number of rows) - analogous to the cone's slant height
T_θ	array's width in number of cells (number of columns) - analogous to the circumference of the cone's base
I_r	row index - equal to the distance from the cone's apex in number of cells

Table 1: *Parameters of the Model's Geometry.* The model mountain is the lateral (curved) surface of a cone. In silico, I represent the cone's surface as a square array (grid of cells).