

# Supplementary Information

## S1 Genetic Algorithm

Here we describe, in detail, the genetic algorithm used to search for optimal mixes of plants. The algorithm begins with a randomly created initial “population” of  $N$  plant mixes. Each of these mixes contains a subset of  $k$  plants drawn from the list of all available plants. Each subsequent iteration then consists of “selection”, “recombination”, and “mutation”. While we require that each plant mix contain the same number of plants, we can also find the optimum size of a plant mix by comparing model runs across different plant mix sizes.

We let  $f$  denote our optimality criteria. By evaluating  $f(M_i)$  and  $f(M_j)$ , where  $M_i$  and  $M_j$  are two mixes of plants, we can directly compare the quality of different mixes (e.g.,  $f$  is a function that assigns numeric scores to mixes of plants based on the restoration-specific criteria).

### S1.1 Selection

To select the next generation of mixes, a list of  $s \cdot N$  mixes is created by sampling from the current population, with replacement. The plant mixes in this new list are then ranked and sorted according to the selection criteria function,  $f$ . The best  $N$  mixes are then used to create the next generation. Increasing  $s$  intensifies the strength of selection. However, it also reduces the amount of variation that is maintained throughout the optimization which can possibly slow the rate of convergence.

## S1.2 Recombination

A fraction  $\psi$  of the population is selected to “reproduce sexually”. For each sexually reproducing plant mix, a mate is chosen, at random, from the population of mixes used in the previous generation (choosing individuals from the previous generation as mates helps maintain variation). Any plants present in both mixes are necessarily present in the recombinant. The plants that are not common to both mixes are then ordered as though they are genes along a chromosome (how exactly they are ordered does not matter). Beginning with the focal parent, and travelling along the “chromosome”, recombination events then occur between adjacent plants with probability  $r$  to form a new “recombinant” list. The resultant plants are then combined with those that were common to both parents to produce the full recombinant mix. These recombinant plant mixes are then combined with the fraction  $(1 - \psi)$  of mixes that did not undergo recombination to complete the production of the pre-mutation offspring generation.

## S1.3 Mutation

Plants within each mix “mutate” with probability  $\mu$ . A mutation event replaces a plant in the mix with another plant, chosen at random, from those not already in the mix. Increasing  $\mu$  increases the rate of production of novel mixes. However, it also increases the probability that a good mix will be mutated to one that is not as good.

## S2 Objective functions

### S2.1 Pollinator visitation

To maximize pollinator visitation, we use the plant-pollinator interaction data to first construct an  $m \times n$  “visitation matrix” (denoted  $V$ ). The rows of  $V$  correspond to the pollinator species, the columns to the potential plant species, and the  $(i, j)$  entry indicates the number of records of pollinator species  $i$  visiting plant species  $j$ . We compute the visitation score of plant mix  $M$ , denoted  $f_V(M)$ , by subsetting the visitation matrix to those plants present in  $M$  and then summing all matrix elements. Consequently, for a given mix size  $k$ ,  $f_V$  is maximized by the  $k$  plants that received the most visits across the data-set.

### S2.2 Pollinator richness

To compute the richness score of a plant mix  $M$ , denoted  $f_R(M)$ , we identify how many rows in the subsetting visitation matrix described above for visitation contain a non-zero entry. In other words, for a mix of a given size,  $f_R$  is maximized by the plant mix that is visited by the greatest number of pollinator species.

### S2.3 Floral phenology

One could use floral phenology to select a plant mix in multiple ways, and exactly how should depend on the data available and the goals of the restoration. Here we provide two representative approaches.

The first requires a record of calendar dates on which each visitation occurred. We use these to infer the duration of a flower’s “bloom period”, defining it for each plant species as the range of dates spanning all recorded visits across all pollinators. Alternatively, if direct phenological measurements of bloom were available, we could instead use those

or, alternatively, one might wish to use percentiles instead of extremes. However, a full evaluation of how one computes the bloom period of a plant species is not our goal here. Once we have computed bloom periods, we then classify a pollinator as “supported” by a mix if the bloom periods of the included plants visited by that pollinator collectively cover the flight season of that pollinator. The phenology score for a particular mix, denoted by  $f_T(M)$ , is then equal to the number of pollinators whose flight seasons are fully supported. Here, the subscript T is used to indicate that this metric is based on the timing of interactions.

The second method is less data intensive, requiring bloom periods for the candidate plant species but not visitation records. First, we split the season into some number of time intervals (e.g., months or weeks). For plant species  $i$ , we define the matrix  $x$  where  $x_{i,j} = 1$  if plant species  $i$  flowers during time period  $j$  and 0 otherwise. Ideally, we want to select plant mixes that have many species blooming in each interval and few intervals with nothing blooming. If we let the vector  $y$  denote the number of species blooming in interval  $j$  (specifically,  $y_j = \sum_i x_{i,j}$ ), then we can accomplish this by computing a modified geometric mean of the elements of  $y$ . Specifically, we define

$$f_B(M) = \sqrt[n]{\prod_{j=1}^n (y_j + 1)} \quad (S1)$$

where  $n$  denotes the number of time intervals or, equivalently, the length of the vector  $y$ . Unlike the arithmetic mean, the geometric mean penalizes heavily mixes with small values, thus favoring mixes that promote season-long evenness of flowering. However, because the geometric mean is multiplicative, it equals zero when any single entry is zero. Thus we add 1 to each entry in the above product. The subscript B is used here to indicate that this metric is based on bloom periods.

Neither of these objective functions account for uncertainty in our knowledge of the

timing of plant-pollinator interactions. For example, it is possible that our estimated flight seasons or bloom periods for some species are too long or too short, or some phenologies may change from year to year. One could accommodate such uncertainty by modifying these functions appropriately. For example, in the case of  $f_B$ , we could require that pollinator flight seasons be buffered on either side by some number of intervals. Optimizing  $f_T$  does not require any pollinator information, however, because it simply finds the mix of plants that provides the most even floral sequence. In this case, one could shorten the estimates of plant bloom periods by a given percentage before optimization, which would force selection of mixes that have greater bloom overlaps.

## **S2.4 Composite objective functions**

While the functions introduced above have all been constructed to evaluate single criterion, construction of objective functions that simultaneously maximize multiple criteria can be accomplished easily by combining these simpler functions. For example, to maximize both pollinator visitation and richness, the required objective function could be constructed by multiplying together the appropriate simple objective functions, giving  $f_{VR}(M) = f_V(M) * f_R(M)$ . By combining multiplicatively, the component objective functions are weighted equally and thus the composite objective function will be maximized by plant mixes that maximize this product. Thus, the units of measurement for each component objective function need not be comparable. More complex methods for combining these objective functions could also be constructed, should one want to weight visitation more heavily than richness, for example.