Ordering Samples along Environmental Gradients using Particle Swarm Optimization

Steven D. Essinger[1], Robi Polikar[2], Gail L. Rosen[1]

Drexel [1] Rowan University [2]

Bio-chemical Signal Processing Laboratory

Aim of study:

Without *a priori* knowledge of an underlying gradient nor the ideal conditions under which the sampled species thrive, can we order the sample sites such that the ordering corresponds to the increase/decrease of the gradient?

Ordination: Represent species/site relationships in low-dimensional space for visualization and interpretation.

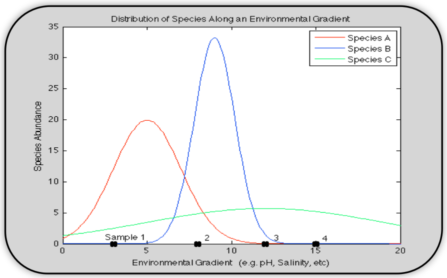
Environmental Gradient: Spatially varying environmental condition such as pH, salinity, moisture, temperature etc.

Environmental gradients affect the *presence/absence (i.e. abundance)* and *distribution* of species in the environment.

This is evident by observing that species prefer particular environmental conditions.

Think about humans and our sensitivity to temperature.

Typical species (e.g. bacteria) response curves to an environmental gradient.



Species samples are collected at various sites in the environment (e.g. soil, water, etc.). The abundance of each species obtained at each site is recorded in a community data matrix. Measurements of known environmental gradients may also be obtained at each site, each gradient mapped to a vector.

	Sample 1	Sample 2	…	Sample N
Species 1	4	7	…	0
Species 2	0	1	…	16
…	…	…	…	…
Species M	4	5	…	1

M >> N

pH =

4.1
4.3
…
8.35

A community ecologist has two sets of tools at their disposal for analyzing species/sample site relationships:

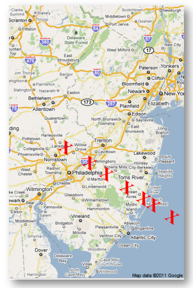
Direct Gradient Analysis (Constrained): Gradients have been measured. The scoring of sample sites is constrained to be linear combinations of the measured gradients.

Methods include: CCA, RDA

Indirect Gradient Analysis (Unconstrained): The gradients are unknown a priori. The sample sites are positioned by an ordering, such that samples that are maximally dissimilar are placed maximally apart in a resulting sequence or plot .

Methods include: PCA, CA (DCA), PCoA, NMDS

Hypothetical Example: Ordering land/water samples according to unknown environmental gradient (gradient hypothesized to be salinity)



X marks sample site: bacteria abundance data obtained via 16s rRNA/pyrosequencing

The spatial ordering of samples are:
PA, Del River, NJW, NJE, Bay, LBI, Atlantic

We wish to order these samples by the underlying gradient, which should be causing the species abundances to vary.

Expected output (if salinity is the gradient):
PA, NJW, NJE, LBI, Del River, Bay, Atlantic

Increasing Salinity

Empirical Example: Bacteria have been sampled along a pH gradient. Can we recover the sample order w. r. t. increase/decreasing gradient?

180m Hoosfield Acid Strip, UK

Controlled site so that pH is the only gradient of major influence

X1 X2 X3 X4 X5 … … … X20 X21 X22

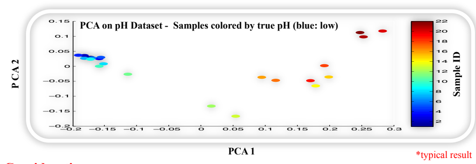
4.1 pH 8.3

X marks sample site: bacteria abundance data obtained via 16s rRNA/ pyrosequencing

The output of analysis should produce samples ordered 1 to 22 or 22 to 1

A typical approach to the problem: PCA Analysis

1. Standardize community data matrix (subtract mean, divide by std. dev.)
2. Compute symmetric correlation matrix [#_sites x #_sites]
3. Find orthogonal vectors of maximum variance (Eigen-analysis)
4. Keep two eigenvectors corresponding with the two largest eigenvalues
5. Plot PCA_1 vs. PCA_2:



*typical result

Considerations:

1. Arch Effect: Characteristic curve is an artifact of the method
   - obscures interpretation of PCA_2
2. Cannot resolve or detect multiple gradients due to arch effect
3. Effectively, most gradient influence is compacted in PCA_1 (highest variance) so we cannot say that PCA_1 is the gradient.
4. Assumes that the species are monotonically related to one another and the gradients.

Novel Approach to the Problem:

Fitness Function

Given a permutation of samples **x**, test the null hypothesis that the abundance of each species is randomly distributed across the samples.

Wald-Wolfowitz Runs Test, w(x)

$$y = \sum_{i=1}^{M} w_i(\mathbf{x}),\ \{y\ \epsilon\ \mathbb{Z} : (0,m)\} \text{ and } \{w_i\ \epsilon\ \mathbb{Z} : (0,1)\}$$

$$\mathbf{x}_{correct} = \max_{\mathbf{x}}(y)$$

Null hypothesis (1) affirmative or (2) rejected for most species:

(1) Infer samples are incorrectly ordered
(2) Infer samples are correctly ordered

Particle Swarm Optimization

Inspired by the behavior of insects in a population.

We model each insect as a particle representing one solution to our optimization problem.

The collection of particles, referred to as the swarm, exist in the solution space.

Each particle has a position and a velocity. With each iteration of the PSO algorithm, a particle moves in the solution space based on its performance on the fitness function and that of the entire swarm.

The benefit of this behavior is that in many cases the optimization avoids local minima.

Optimization terminates once maximum number of iterations reached or a particle's fitness score passes predetermined threshold.

Hybrid PSO Algorithm Pseudocode

Input:
- Data matrix of species abundance: $[(\#\,of\,Species)\ x\ (\#\,of\,Samples)]$
- Fitness test: Wald-Wolfowitz
- m: # of particles for PSO
- k: # of iterations
- Free-parameters: $\{w, c_1, c_2\}$

Initialization:
1. Input seed, $\mathbf{x}_s$: random permutation of integers 1:n
2. Compute initial particle velocities, $\mathbf{v}_i^0$:
   a. Choose # of IM{} for each particle's initial velocity:
      - Drawn from: $U[n/4, n/2]$
   b. Select the node and displacement for each move:
      - Node from $U[1,n]$
      - Displacement from: $U[-n/3, n/3]$
3. Compute initial particle positions:
   $\mathbf{x}_i^0 = \mathbf{x}_s + \mathbf{v}_i^0$, i = 1:m
4. Set each particle best to initial:
   $\mathbf{p}_i = \mathbf{x}_i^0$
5. Run fitness test on all $\mathbf{x}_i^0$
6. Set global best position:
   $\mathbf{g} = \mathbf{x}_i^0$ s.t. $\max_{\mathbf{x}_i^0}(fitness)$

Iteration:

For iter = 1:k
For particle$_i$ = 1:m

A. Compute velocities: $\mathbf{v}_i^{k+1}$:
   1. $w\mathbf{v}_i^k$
   Compute r_1, r_2, then:
   2. $c_1 r_1(\mathbf{p}_i - \mathbf{x}_i^k)$ distance of current to particles best
   3. $c_2 r_2(\mathbf{g} - \mathbf{x}_i^k)$ distance of current to global best
B. Add velocities sequentially to $\mathbf{x}_i^k$:
   $\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1}$
C. Compute fitness of each $\mathbf{x}_i^{k+1}$:
   if $fit(\mathbf{x}_i^{k+1}) > \mathbf{p}_i$, then $\mathbf{p}_i = \mathbf{x}_i^{k+1}$
   if $fit(\mathbf{x}_i^{k+1}) > \mathbf{g}$, then $\mathbf{g} = \mathbf{x}_i^{k+1}$
D. Perform Local Enhancement on $\mathbf{g}$ using SOP-3 Exchange
E. Update $\mathbf{g}$

End
End

Definitions:
- $\mathbf{x}_s$: position (permutation) seed for all particles
- $\mathbf{x}_i^k$: position for particle i at iter k
- $\mathbf{v}_i^k$: velocity for particle i at iter k
- $\mathbf{p}_i$: best position (permutation) for particle i
- $\mathbf{g}$: best position (permutation) of all particles
- r_1, r_2: random values from U[0,1]
- Free parameters:
  - w: current particle
  - c_1: particle best
  - c_2: global best

Performance: Random Chance

Iterations	Fitness Best	Fitness Mean
5000	121	10
10000	122	10

Performance: Proposed Algorithm

Iterations	Fitness
20	111
50	149
100	140
200	212
300	201
500	203

Fitness Goal
338

For the pH dataset there are 22! (~10[21]) possible solutions to the problem

Challenges of the Problem:

- ✧ What do the gradients look like?
  Current methods cannot recover them.
- ✧ What is the true distribution of species w. r. t. each environmental condition?
  Arguments over the unimodal model.
- ✧ Determining number of gradients
  Current methods cannot resolve multiple gradients
- ✧ Issues dealing with sampling artifacts.
  Recovering rare species and correct abundance in samples
- ✧ Plus, Issues with Species/Species Interactions and Nonlinearities

*This work was supported in part by National Science Foundation award #0845827.