for **each search point**

1. crop\_to\_neighborhood: crop raster grids to neighborhood extent

for **each neighborhood**

1. set random seed
2. establish\_ecotone\_transect:
   1. locate\_candidate\_transect\_lines:

for **each candidate line** until one is successful

1. orient\_transect\_line:
2. locate\_candidate\_transect\_band:
   * Extract band transect along elevation transect with a width of 200 cells
   * Calculate outline of maximal band transect along elevation transect and how to transform from original coordinates to band transect coordinates
3. determine\_ecotone\_linear\_extent: Apply definition of zone of ecological boundary to BSE and TF
4. trim1\_band\_to\_ecotone: Crop band transect to limits1\_ZoneEcolBoundary
5. Determine majority BSE and majority TF types within limits1\_ZoneEcolBoundary
6. determine\_ecotone\_linear\_extent: Apply definition of zone of ecological boundary to majority of BSE and majority of TF within limits1\_ZoneEcolBoundary

if successful

1. trim\_line\_to\_ecotone: Crop the linear transect to the ecological boundary zone
2. extract\_tband\_grids: Crop the band transect to the ecological boundary zone
3. trim2\_band\_to\_ecotone: Crop band transect to zone of ecol boundary and set start of 'my' coordinate system
4. Environmental gradient along transect

* Extract environmental conditions along transect band
* Quality/Homogeneity of environmental gradient along transect
  + Human footprint
  + Quality of aspect: local (RMSE), patches (clumps), and global (mean) aspect
  + Quality of elevation

1. Vegetation data along transect: Need to recalculate veg1, veg2, veg1density, and veg2density based on patches

* Quality: vegetation patch sizes

1. identify\_migration\_patches:

* AllMigration: all vegetation in transect
* OnlyGoodMigration
* all vegetation minus patches whose flowpath connects to beyond y-border indicating non-x-axis migration origin
* identify patches that 'drain' out either at y=1 or at y=200; assumption: most likely migration route = flowpath

1. calc.MigrationRoutes\_EstimateFlowpaths:
2. calc.Identify\_GoodvsBadMigration:
3. calc.RemoveBadMigration\_fromVeg:

end loop for each neighborhood

end loop for each search point