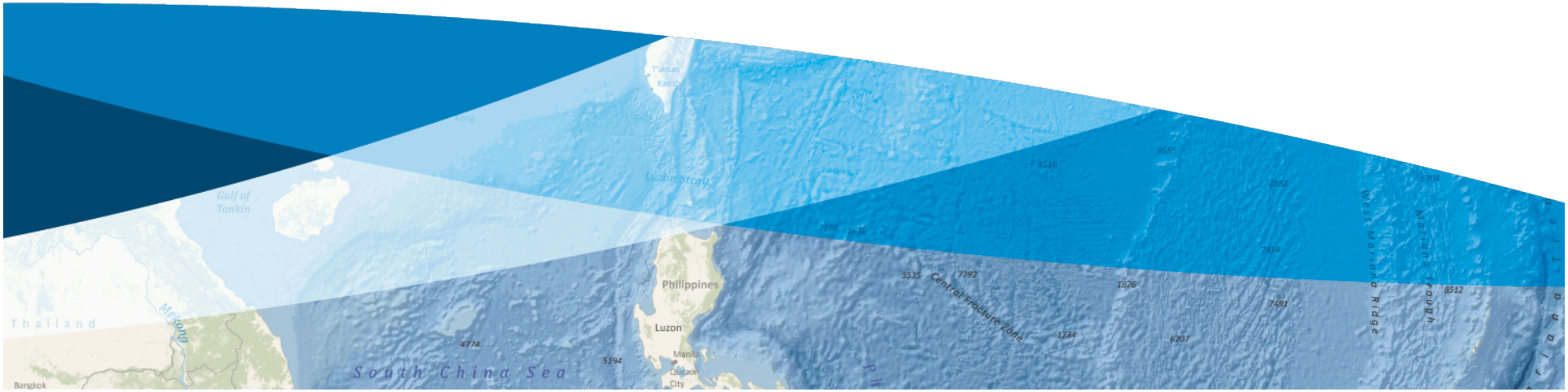
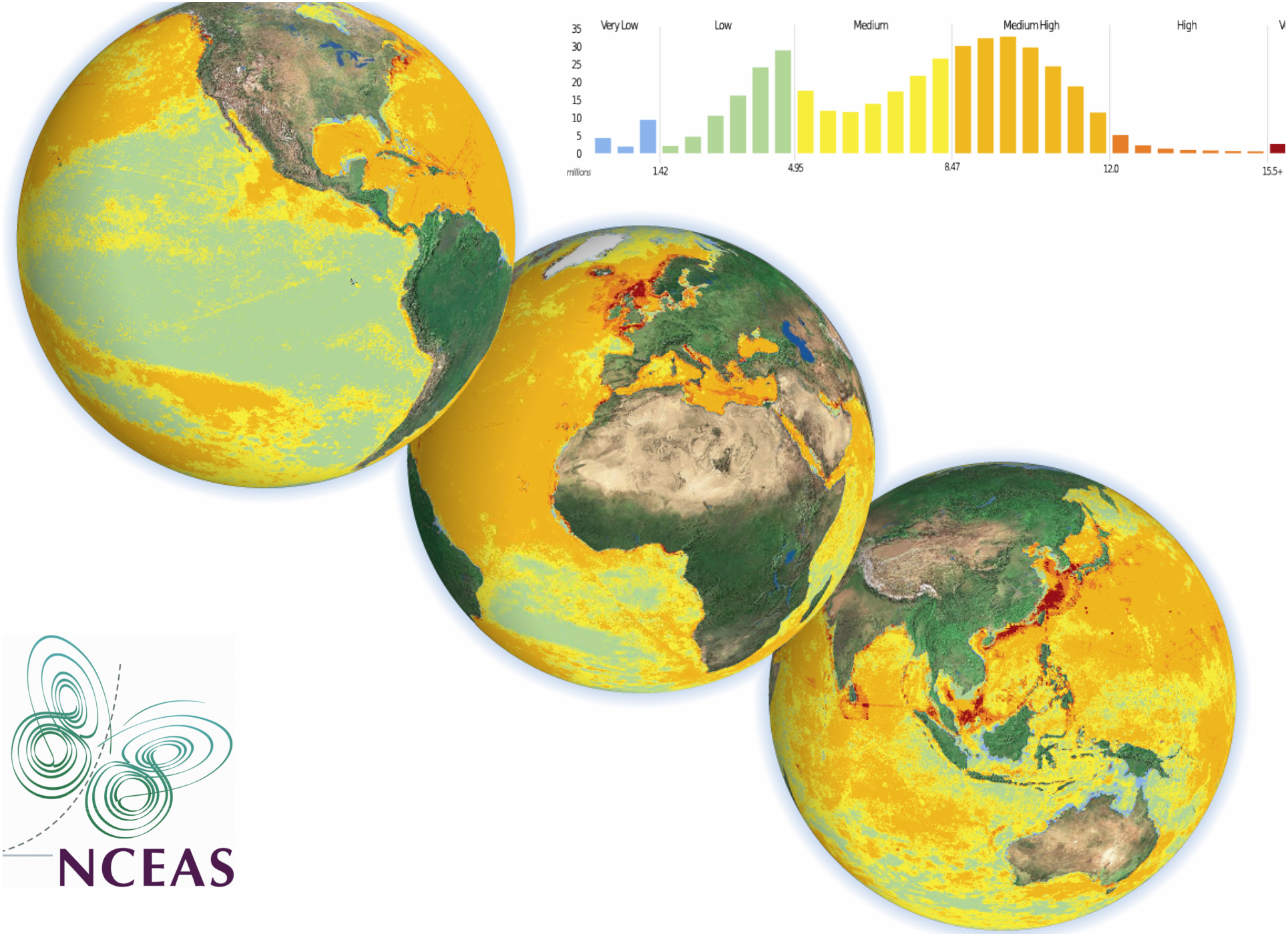
Deploying Python tools to GIS users

Shaun Walbridge / @scw
swalbridge@esri.com

https://github.com/scw/scipy2014-talk

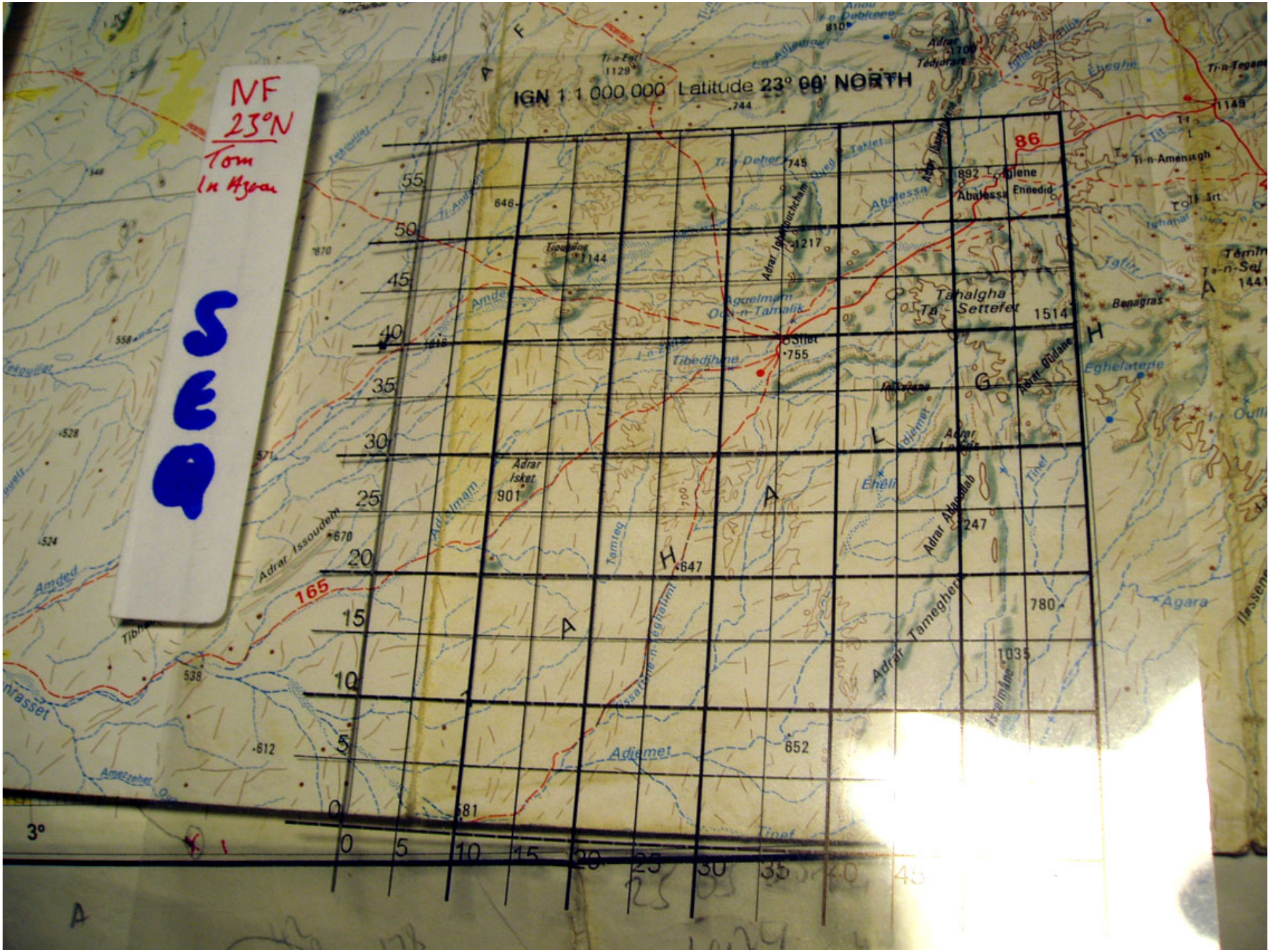




NCEAS

Halpern, 2008

- Dawn Wright, Esri Chief Scientist: Building up GIS for ocean science
- Multidimensional support, vertical datums, hybrid workflows
- Me: extensions for geomorphology, genetic analysis. Move scientific GIS users toward maintainable, reusable code and beyond the “one-off” code mentality


NF
23°N
Tom
In Ayar
SEQ
IGN 1:1 000 000 Latitude 23° 00' NORTH
Adrar Tédjorart
Ti-n-Tegeme
Ti-n-Amenugh
Ti-n-Sel
Benagras
Tanalgha
Ti - Settefet
Adrar Ouldane
Eghelatene
Adrar Isket
Adrar Issoudem
Adrar Tameghera
Adrar Abdoulab
Adiemet
Agara
Ameziehar
Amded
Tibedjhane
Agrelmam Ou-n-Tamalik
Ti-n-Deher
Abalessa
Ennedid
Tibet
Tamteq
Issarane-n-eghatimt
Issalmane
Ehelli
Tinet
Tafirt
Ihanar
Ilassene
Tibiri
Tezoulet
Tekoulet
Anrasset
86
165
1149
1514
1144
1217
745
892
755
901
670
647
247
780
1035
652
681
612
538
524
558
528
646
744
810
1129
1700
3°
0
5
10
15
20
25
30
35
40
45
50
55

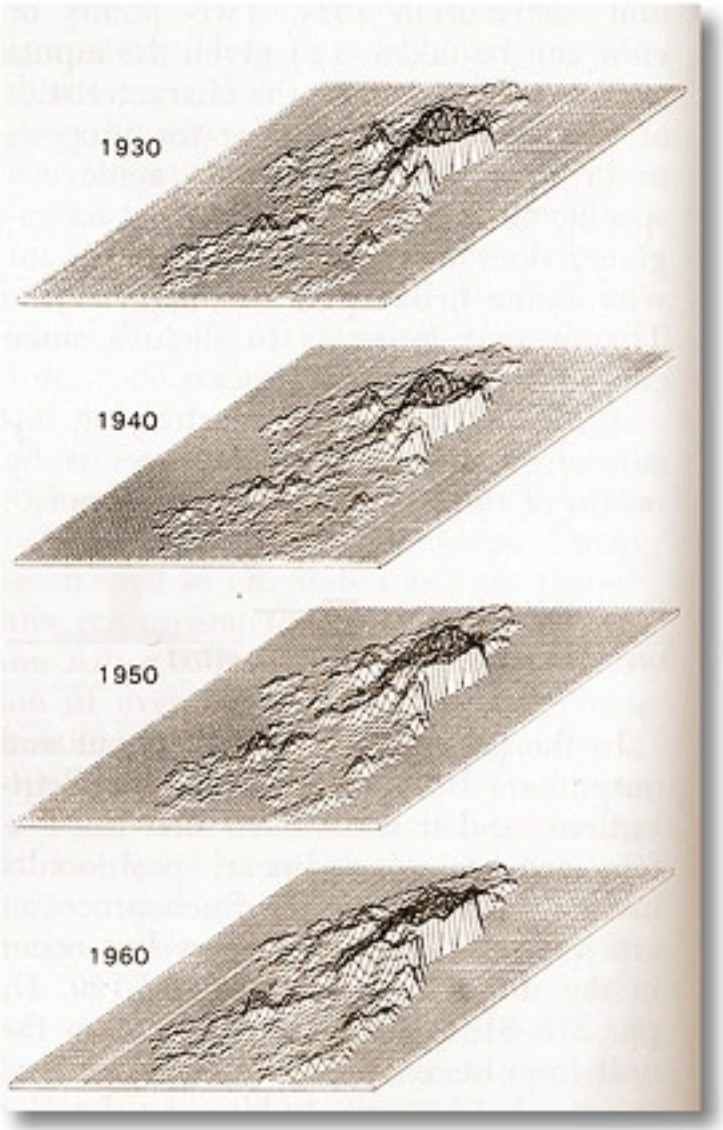
Background on GIS

- Software implementations preceded science (e.g. vs statistics)
- Blends both art and science in a cohesive model

- Modeling
- Prediction (geostatistics)
- Analysis (spatial statistics)
- Design
- Visualization

Representing Spatial Objects

- Continuous fields (raster)
  - Regular (GeoTIFF)
  - Irregular (Finite Element Model)
- Vector
  - Non-planar (Shapefile, GeoJSON)
  - Topological (E00, TopoJSON)
- Network
- Move between objects and a dimensional space:
  - 0D: Point
  - 1D: Line
  - 2D: Area
  - 3D: Solid
  - 4D: Space-time



How broadly used are the models?

Choices	Percentage
Simple Vectors (points, lines, areas)	*91.30%*
Rasters	*73.91%*
Vectors which include time or measures	*60.87%*
Point clouds (e.g., LiDAR)	*34.78%*
Raster Catalogs	*34.78%*
Raster Mosaics	*26.09%*
Multidimensional Data Structures (e.g., NetCDF, HDF)	*21.74%*
Other	
Irregular Grids	*17.39%*
Other	*13.04%*
Mesh Volumes	*8.70%*

Space is like Time

- How do we communicate about time?
- Easy enough: "meet me at 5pm at the statue of a guy riding a globe while reading a book."

Space is like Time

But accurately representing time at a fine scale requires:

- Timezones
- Historical data on time zone changes (some claimed under copyright!)
- High-resolution equipment
- So, even though time is 'easy' it's hard if you want it to be precise.

That said, tons you can do without the complexities of projections, geodesics, &c

Open Source Tools

- Primarily programmer- and CLI- oriented
- Great area of growth, want to bridge these tools to a lay audience who use GIS in existing complex workflows
- Hybrid solutions are common

GDAL PostGIS R GEOS Geometry Engine Open Source

Why GIS?

- More people want to use spatial thinking than there are programmers – domain experts, not computer scientists
- Desktop GIS broadly is used, want to reach these users with tools we develop
- GUI with command line + API for experts
- Latency for many analytical applications means real-time on the web isn't yet feasible (user leaves if it takes ≥10s[1])

1. Nielsen, J. "Designing Web Usability". 1999.

Why ArcGIS?

ArcGIS was polled as #1 tool used for scientific analysis at the 2011 USGS Community for Data Integration Workshop[1]:

ArcGIS	65%
Open Source GIS	12%
Python	8%
Matlab	5%
R	5%
Other	5%

1. Courtesy Rich Signell, USGS Woods Hole

ArcGIS Python Environment

- Python added as scripting language at ArcGIS 9.2 (2004), ships with every install
- Now can execute~80% of ArcMap (100% if you use `ctypes` to connect with Arc Objects)
- Continue to improve support and integration with python

ArcGIS Python Environment

Out of the box w/ ArcGIS:

python™

matplotlib

NumPy

`xlrd + xlwt` (10.2)
`netcdf4`

netCDF

Simple Integration:

IP[y]: IPython
Interactive Computing

pysal
Python Spatial Analysis Library

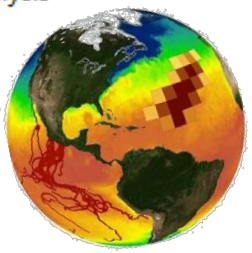
SciPy

- Marine Geospatial Ecology Tools
  - Connectivity Analysis
  - Conversion
  - Data Management
  - Data Products
  - Fishery Analysis
  - Oceanographic Analysis
  - Spatial Analysis
  - Statistics

pandas

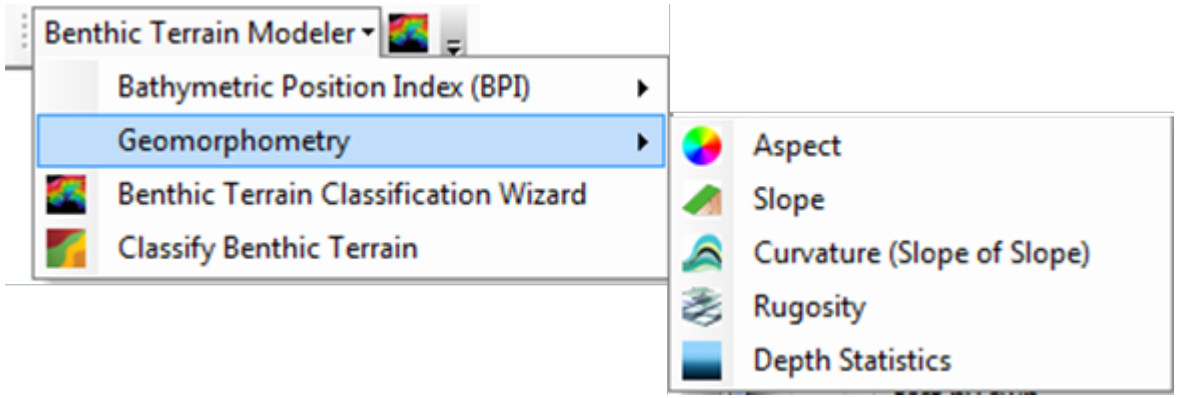
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

GDAL

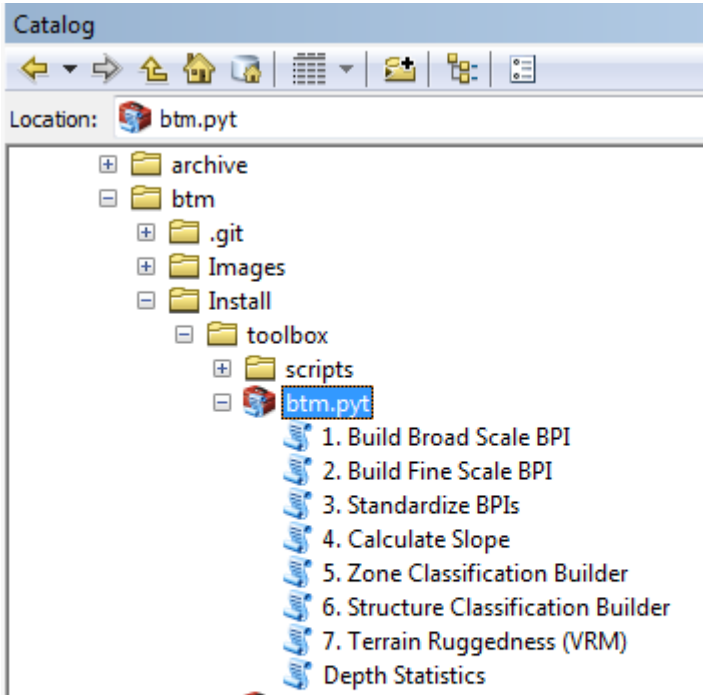


Extensions

Python Add-in

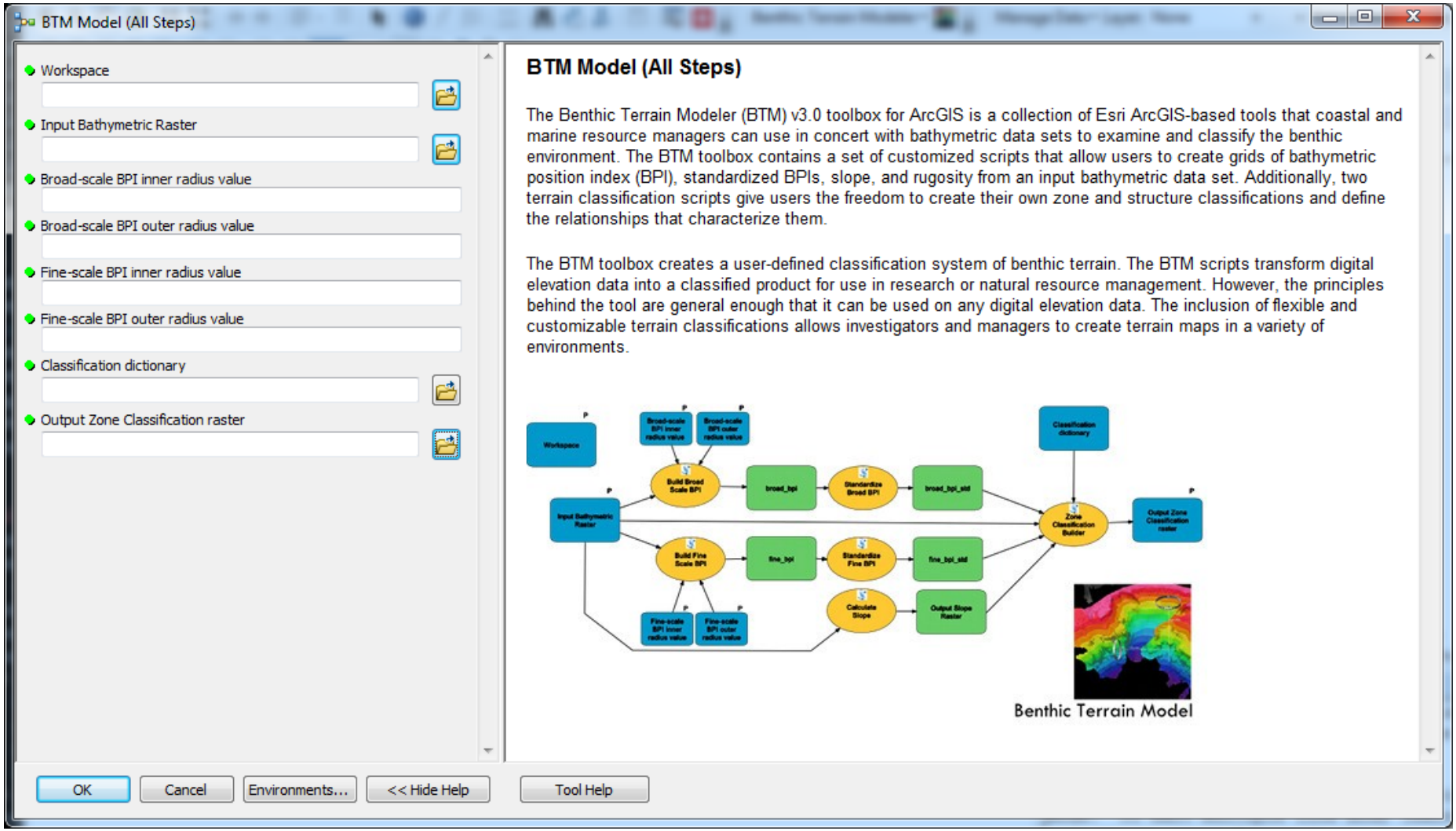


Python Toolbox



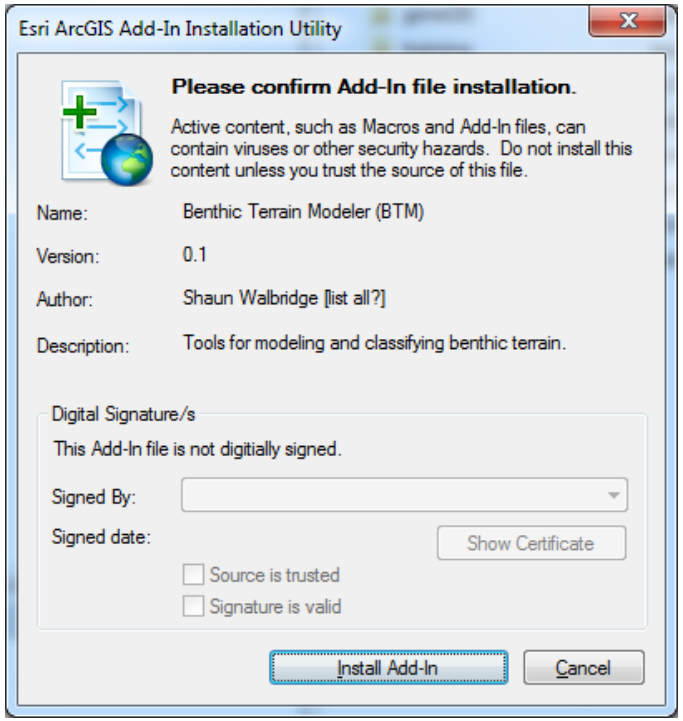
- .NET or Python
- Python toolboxes and Add-ins (10.1+)

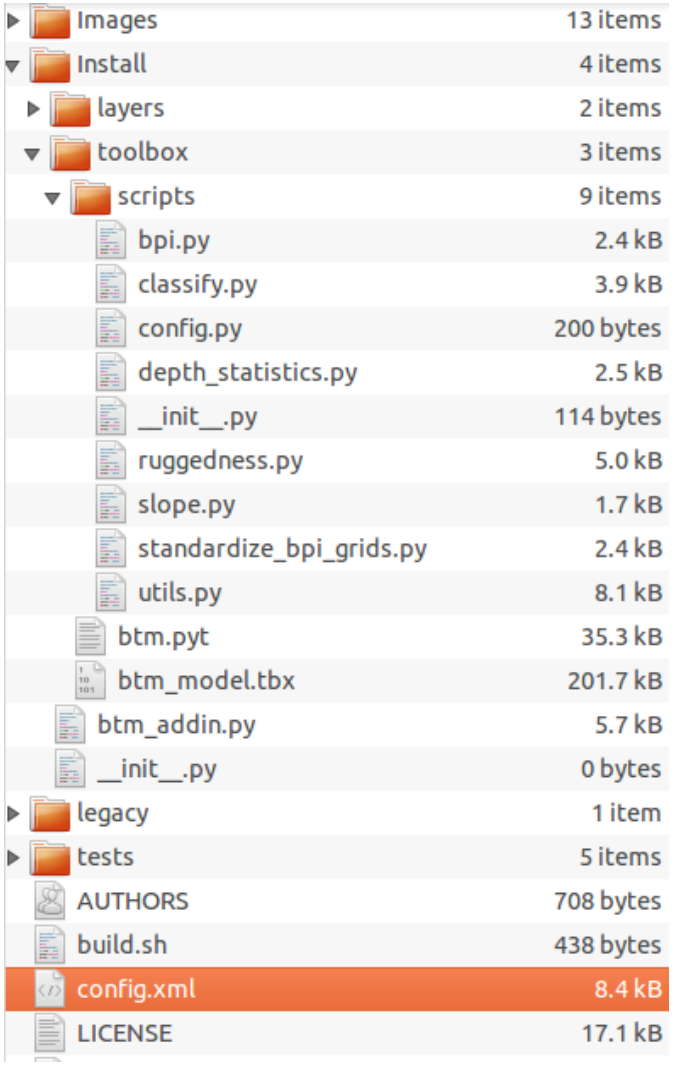
Combine tools in Models



Easy installation and creation

- Python scripts + Python toobox + layout XML
- Builds an installer

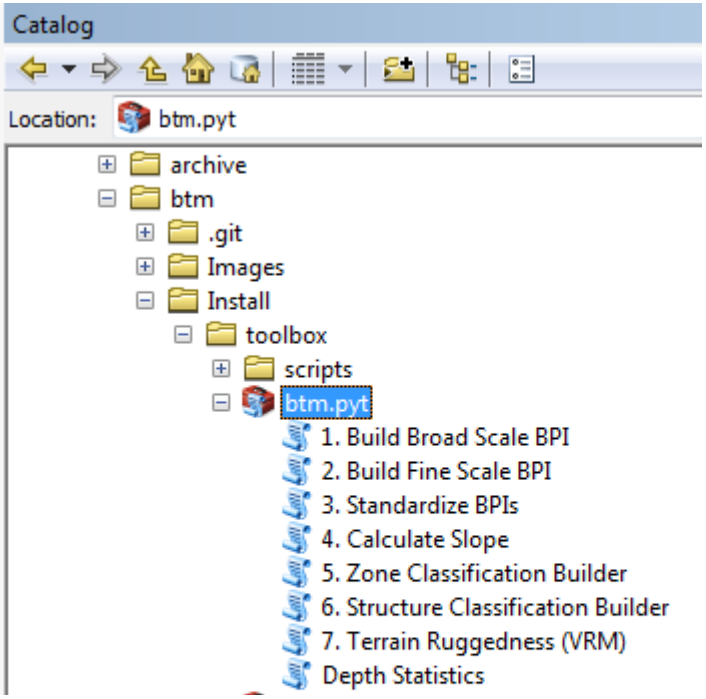




Python Toolboxes

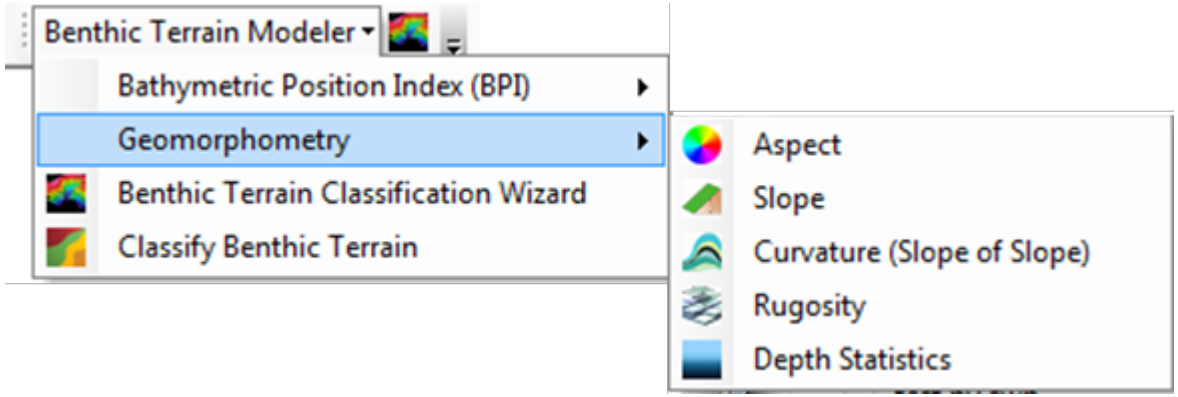
Create a Python class, and add one function per tool you'd like to create

Can embed the execution, or connect to another script



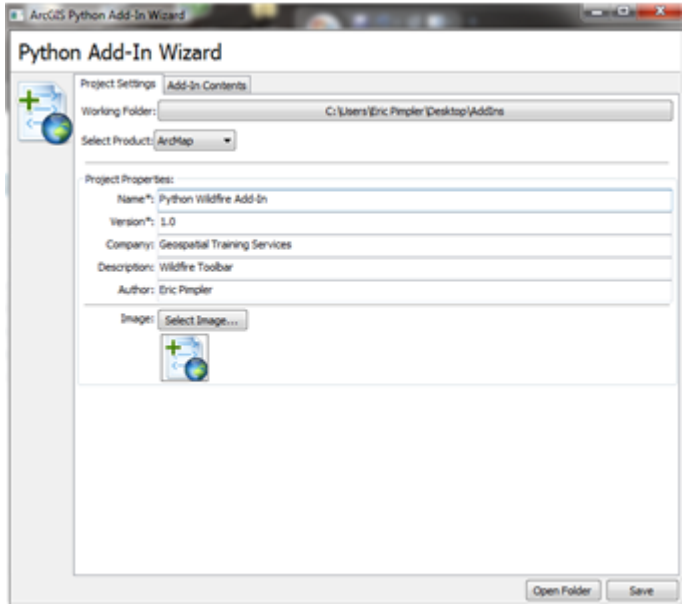
https://github.com/scw/scipy2014-talk/blob/master/maxent.pyt

Python Add-in



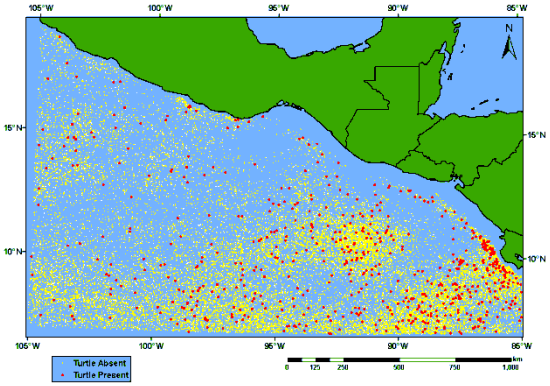
Buttons
Tool Palletes
Combo Boxes
Menus
Toolbars

Defined by an XML for layout + Python code

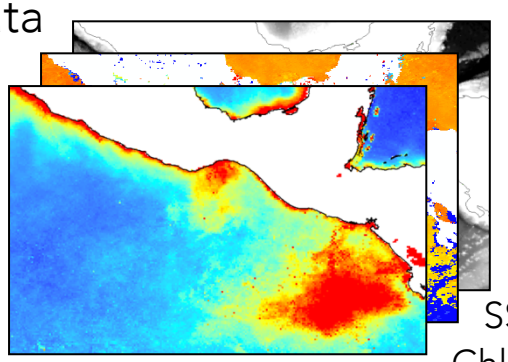


Predictive Species Distribution Modeling

Point observations of species



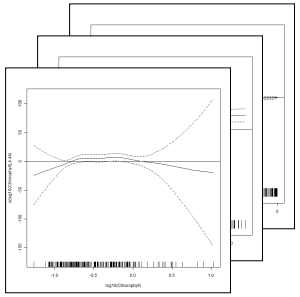
Gridded environmental data

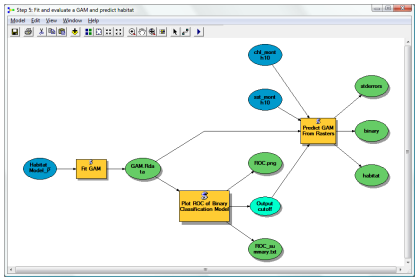


Bathymetry
SST
Chlorophyl

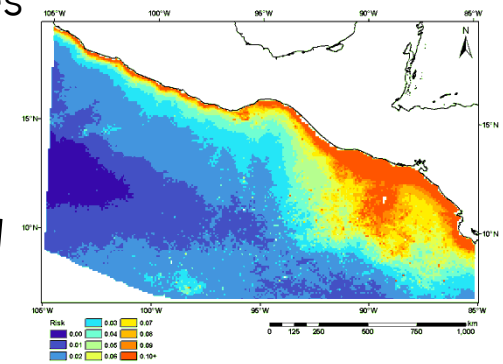
Predictive model

$$g(\mu) = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$$

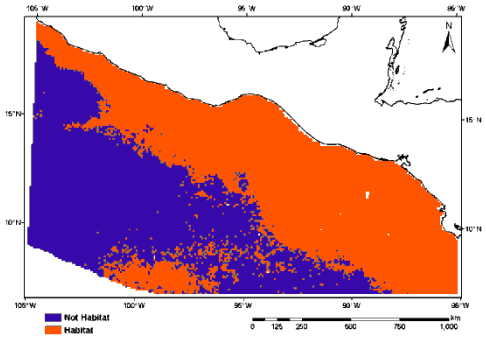




Probability of occurrence predicted from environmental covariates



Binary classification

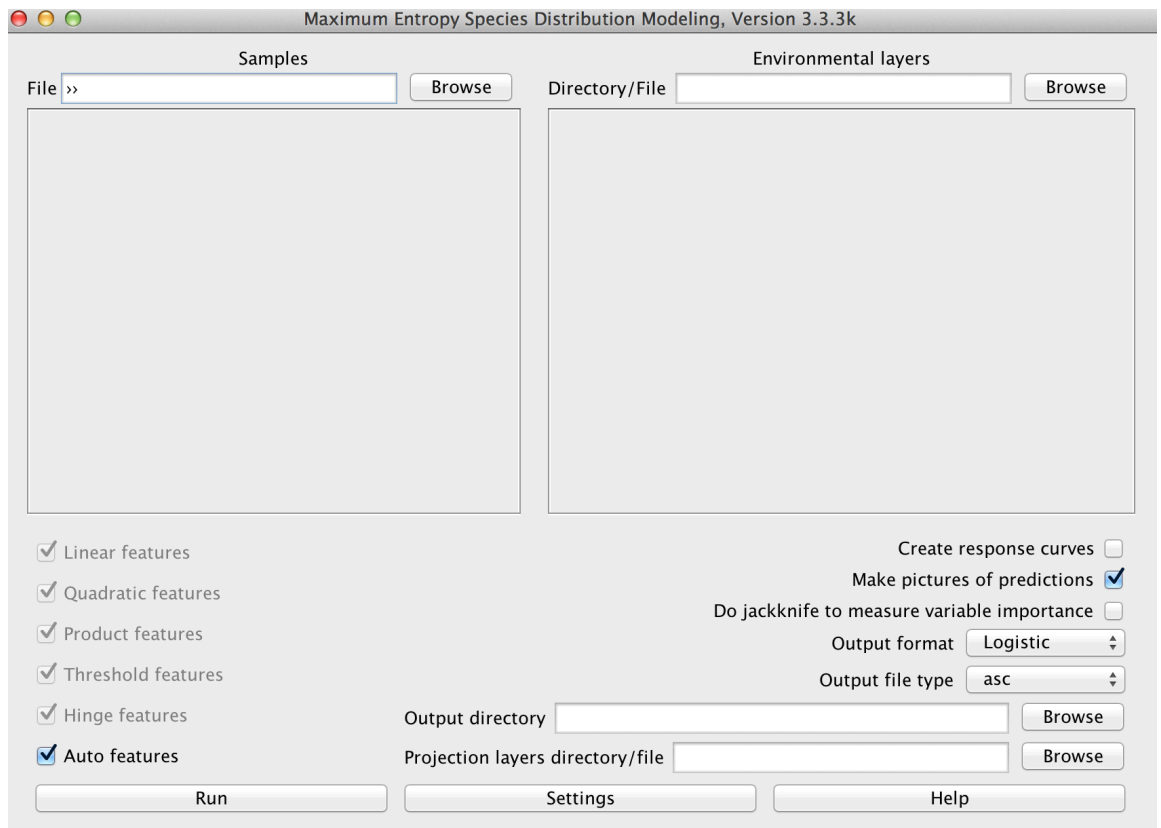


J. Roberts, Duke U

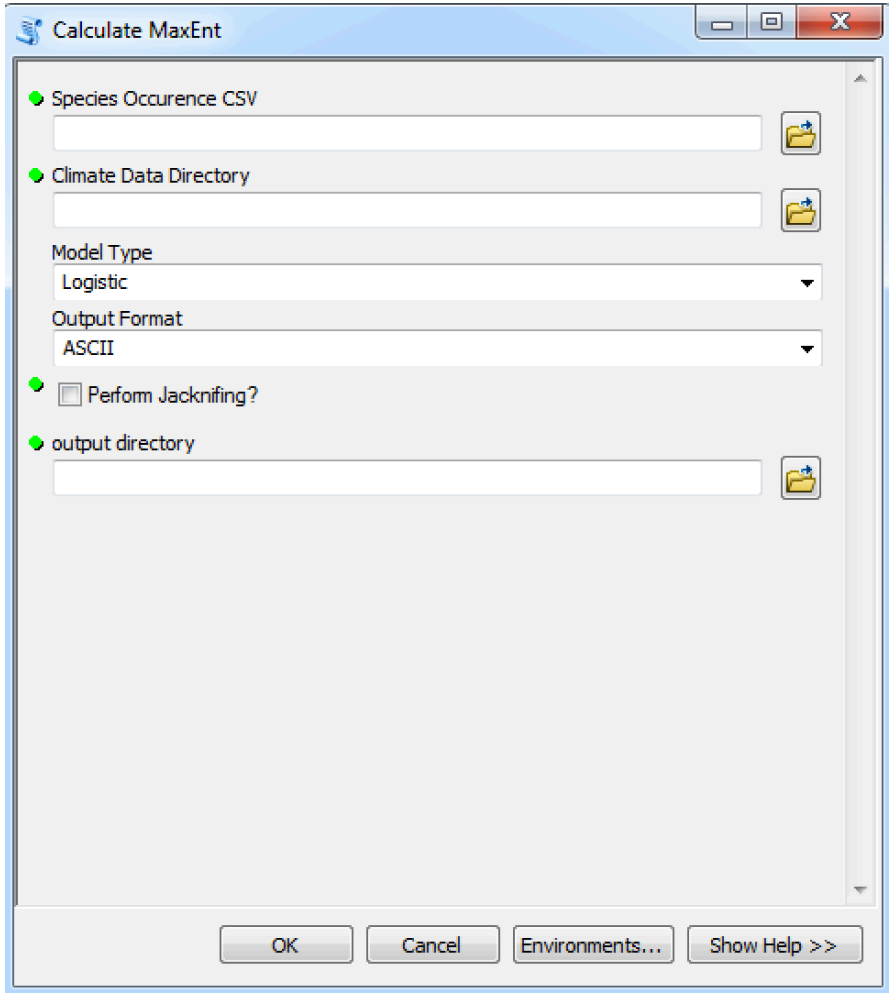
Maximum Entropy Modeling

Machine-learning approach for determining maximum likelihood estimations, particularly useful for presence / absence data with zero-inflation problems.

$$H(X) = -\sum_{k \geq 1} p_k \log p_k.$$



MaxEnt Toolbox




```python
import arcpy

class Toolbox(object):
    """ Our example toolbox which initializes a single tool."""
    def __init__(self):
        self.label = 'Maximum Entropy'
        self.alias = 'maxent'
        self.tools = [calc_maxent]

class calc_maxent(object):
    """ An example tool which calculates the Maximum Entropy of point observations. """
    def __init__(self):
        self.label = u'Calculate MaxEnt'
        self.canRunInBackground = False

    def getParameterInfo(self):
        input_csv = arcpy.Parameter()
        input_csv.name = u'Species Occurence CSV'
        input_csv.displayName = u'Species Occurence CSV'
        input_csv.parameterType = 'Required'
        input_csv.direction = 'Input'
        input_csv.datatype = 'Feature Layer'

        climate_dir = arcpy.Parameter()
        climate_dir.name = 'Climate Data Directory'
        climate_dir.displayName = 'Climate Data Directory'
        climate_dir.parameterType = 'Required'
        climate_dir.direction = 'Input'
        climate_dir.datatype = 'Folder'

        model_type = arcpy.Parameter()
        model_type.name = 'Model Type'
        model_type.displayName = 'Model Type'
        model_type.direction = 'Input'
        model_type.datatype = 'String'
        model_type.value = 'Logistic'
        model_type.filter.list = ['Logistic', 'Cumulative', 'Raw']

        output_format = arcpy.Parameter()
        output_format.name = 'Output Format'
        output_format.displayName = 'Output Format'
        output_format.direction = 'Input'
        output_format.datatype = 'String'
        output_format.value = 'ASCII'
        output_format.filter.list = ['ASCII', 'MXE', 'BIL', 'GRD']

        jacknife = arcpy.Parameter()
        jacknife.name = 'Perform Jacknifing?'
        jacknife.displayName = 'Perform Jacknifing?'
        jacknife.direction = 'Input'
        jacknife.parameterType = 'Required'
        jacknife.datatype = 'Boolean'

        output_dir= arcpy.Parameter()
        output_dir.name = u'Output_directory'
        output_dir.displayName = u'output directory'
        output_dir.parameterType = 'Required'
        output_dir.direction = 'Output'
        output_dir.datatype = 'Folder'

        return [input_csv, climate_dir, model_type, output_format, jacknife, output_dir]

    def updateParameters(self, parameters):
        validator = getattr(self, 'ToolValidator', None)
        if validator:
            return validator(parameters).updateParameters()

    def execute(self, parameters, messages):
        # run related python script with selected input parameters
        import run_maxent

        run_maxent.main(input_csv=parameters[0].valueAsText, climate_dir=parameters[1].valueAsText,
                model_type=parameters[2].valueAsText, output_format=parameters[3].valueAsText,
                jacknife=parameters[4].valueAsText)
```

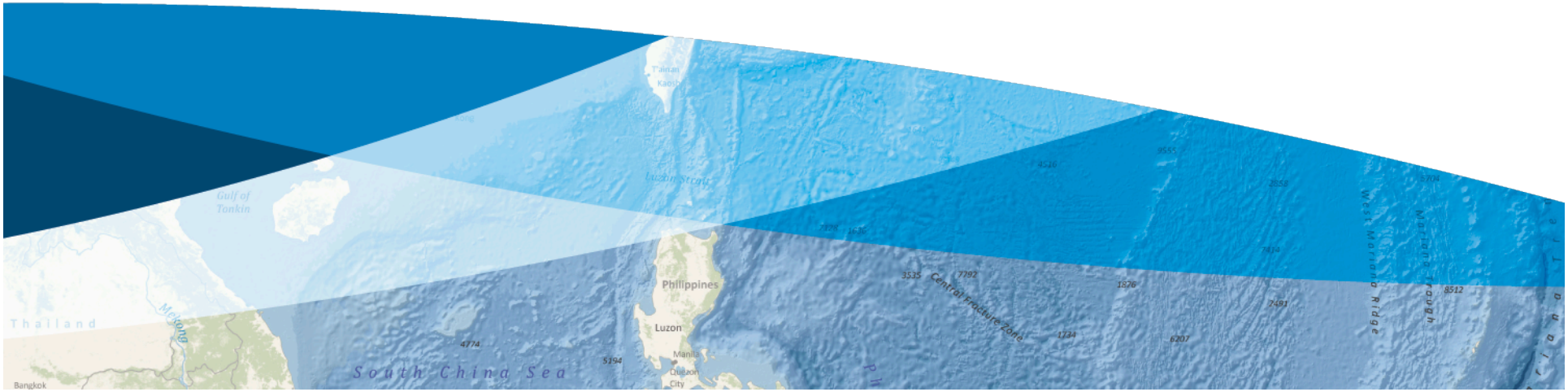
MaxEnt Toolbox and Add-in

- Shows both the model and add-in, and provide instructions on its creation:
  - https://github.com/scw/scipy2014-talk/
- With some minor shim work, share your tools with a new broad userbase

Thank You

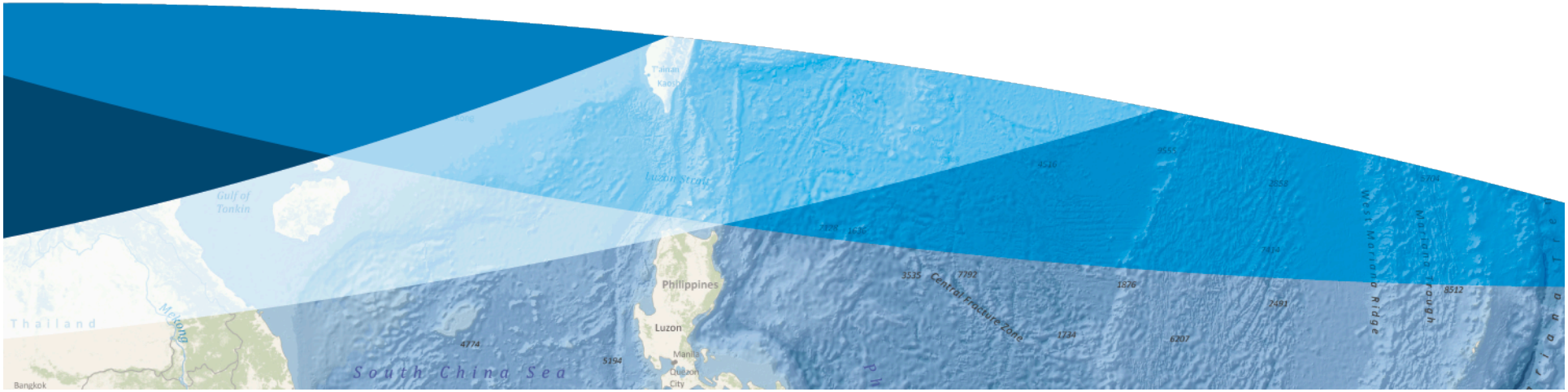
Shaun Walbridge
twitter: @scw
swalbridge@esri.com

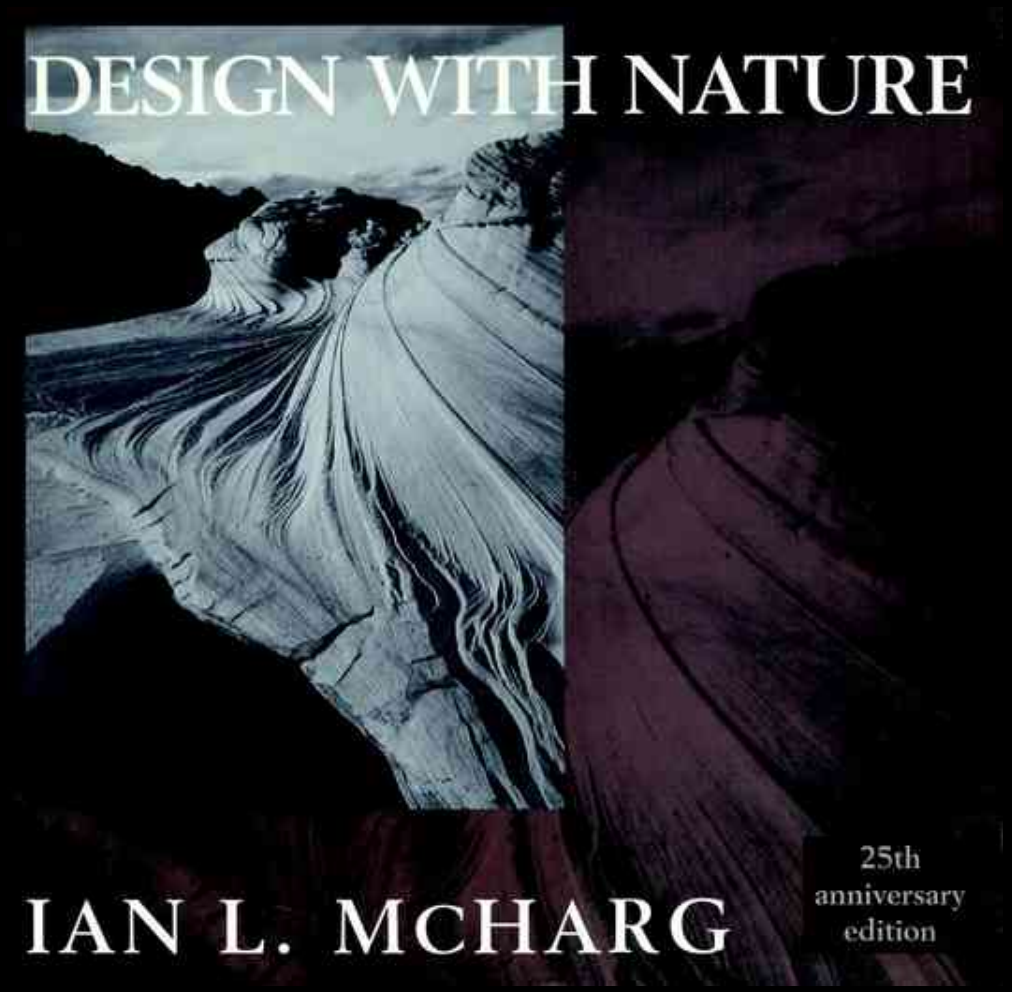
https://github.com/scw/scipy2014-talk



Thank You

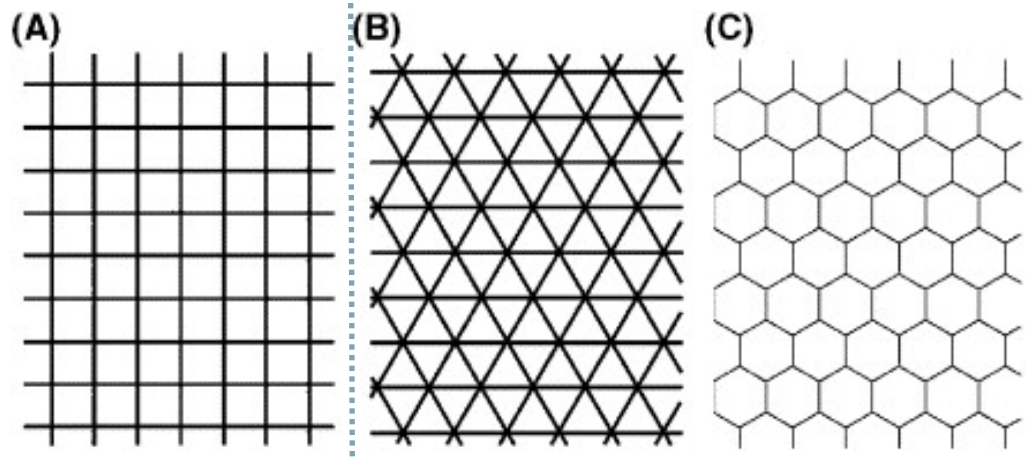
Extras start on next slide




DESIGN WITH NATURE
IAN L. MCHARG
25th
anniversary
edition

Tessellations

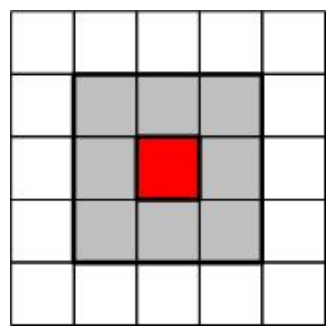
Descartes and friends



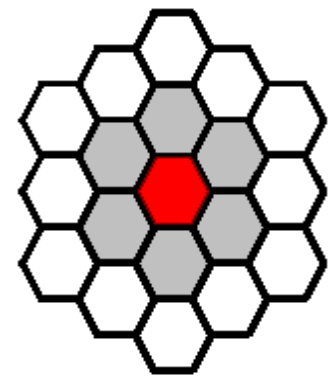
Neighborhoods

Focal operators

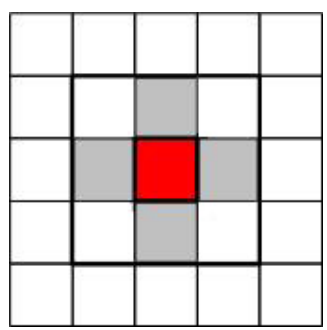
Can also use Zonal operations (use features for zones)



Moore, *r*=1



Hexagonal, *r*=1



Von Neumann, *r*=1

Existing Extensions

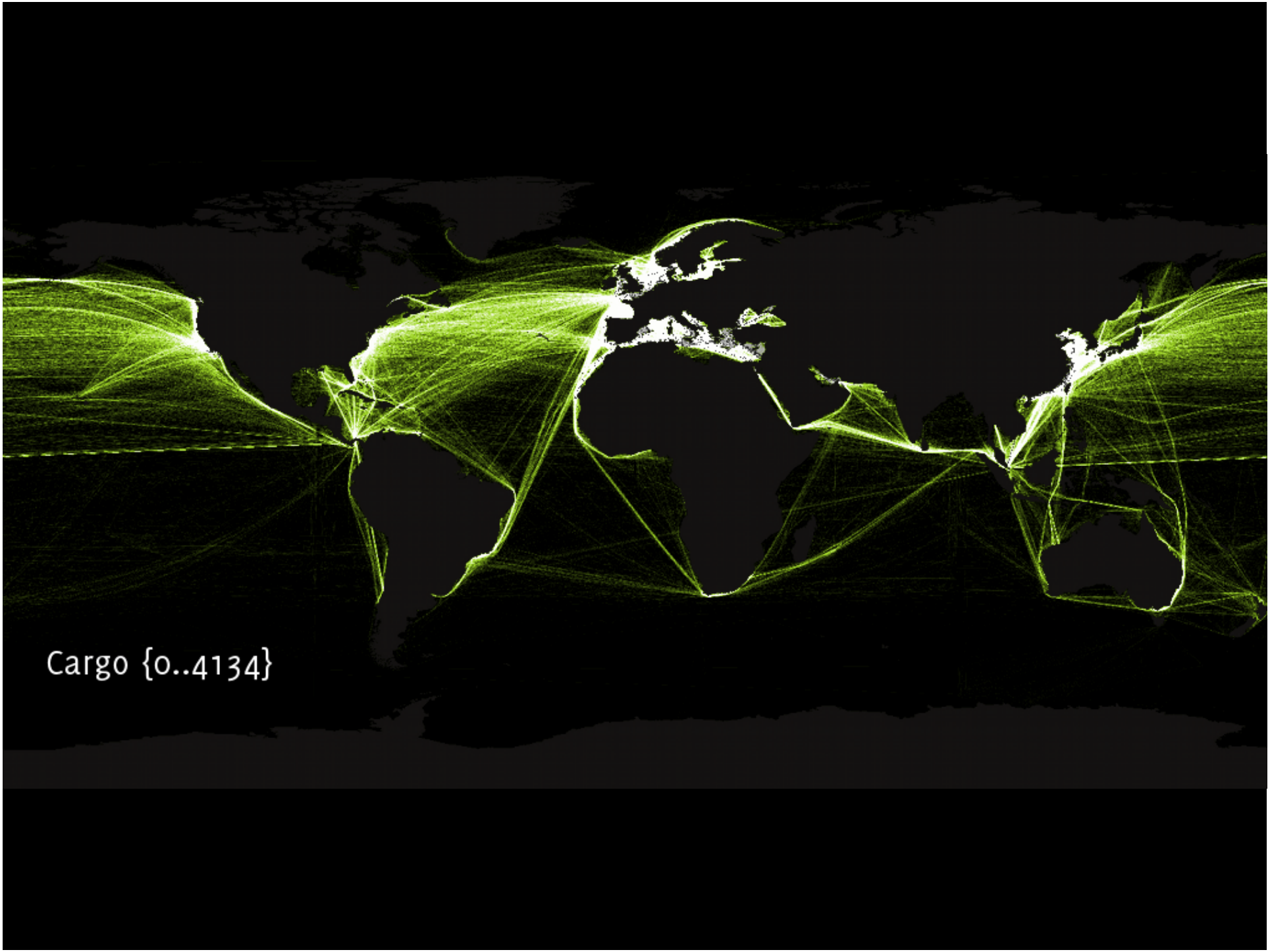
- MGET (Duke University): Integrates R, Python, MATLAB
- GME: R and statistical analysis
- CircuitScape: graph-based movement
- LASTools: LIDAR processing
- Benthic Terrain Modeler: geomorphology
- Esri2Open: export features to web-friendly formats (e.g. TopoJSON, SQLite)

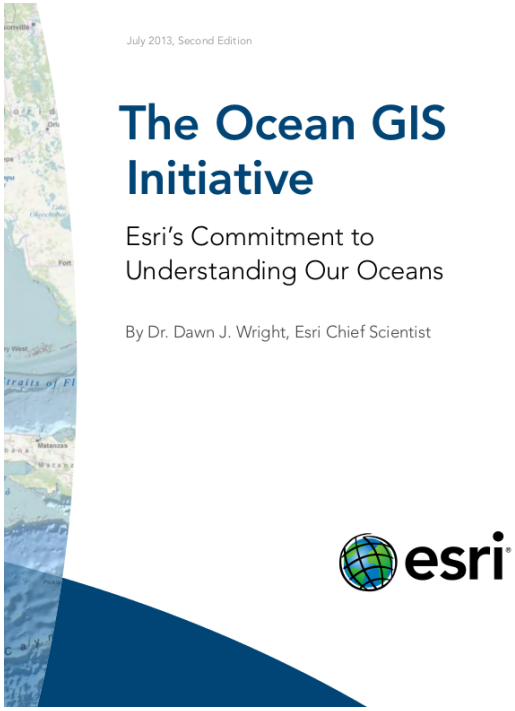
Why Not? (XXX)

- Viable and useful open source GIS components for SCIENTIFIC ENDS

- QGIS
- SAGA
- GRASS GIS
- R

- Other commercial: ENVI, MATLAB, …


Cargo {0..4134}



- Dawn Wright: my boss / awesome person
- Building up ocean science as a core part of what Esri does
- Part of broader Esri efforts to support the needs of ocean science – ocean basemap, multidimensional support, vertical datums, improving ArcGIS for scientific needs

Chicago Transportation (1959)

FIGURE 7—TOTAL PERSON TRIP DESTINATIONS

The destinations of 10,212,000 person trips, on the average weekday, are distributed throughout the Study Area as shown in this model. The highest blocks in the model represent 144,000 trip destinations per quarter square mile grid, the lowest blocks 5,000, the shaded areas less than 5,000 but more than 2,500.