Lifemapper 2.0

Using and Creating Geospatial Data and Open Source Tools for the Biological Community

Aimee Stewart, CJ Grady, Dave Vieglais, Jim Beach
Natural History Museum
and
Biodiversity Institute
University of Kansas
Overview

• Overall Goals
• History
• Current version
• Implementation
• Future
Goals

• Archive of niche models
• Spatial data services (REST)
  – Data (OGC)
  – Spatial analysis against archive
  – On-demand modeling service
Lifemapper 1.0

- NSF funded
- Experimental app.
- Successful DC project
- Workflow steps
- Limited by
  - Data quality
  - Architectural decisions
Lifemapper 2.0

- Funded by NSF/EPSCoR
- Demo pipeline processing specimen data from GBIF cache
- Simpler, controlled architecture
- Cluster computing instead of DC
- Focus on services
Niche Modeling

- In vogue
- Algorithm
- Input Data
  - Environmental
  - Occurrence
- Basic process
- Limitations
- To what end?
• which species
• where does it live
• when did it live there
Western hemisphere fishes
Three Museums
Western hemisphere fishes
“Estimate the distribution of the Fringed filefish”

*Monacanthus ciliatus*
1. Identify algorithm
2. Assemble inputs
3. Compute model
4. Visualize results
Specimen Point Data → Prediction Algorithm (GARP) → Ecological Niche Model → Distribution Predicted for Native Region

Current Environment → Δ Climate → Δ Region → Environmental Coverages

Distribution After Climate Change → Distribution Predicted In Non-native Region

Ecological Niche Model Prediction Algorithm (GARP)
Implementation

• Pipeline
• Spatial Data Library (SDL)
• Cluster
• Open-source
• Python
Operation

• Pipeline
  – retrieves point data
  – constructs request
  – sends job to cluster by REST
• Cluster front end receives /schedules job
• Cluster nodes
  – retrieve environmental data
  – dispatch job to OM
• Pipeline
  – polls for status
  – retrieves and stores model/projection
QuickTime™ and a GIF decompressor are needed to see this picture.
Spatial Data Library (SDL)

- Layer metadata in PostgreSQL/PostGIS
- Expose with Mapserver with custom W*S
- Independent service - so could
  - be standalone
  - be one of multiple SDLs servicing pipeline
 SDL Data

• Environmental data
  – URL in job, retrieved via WCS by node
  – Caches on nodes for efficiency

• Point data
  – Could be REST or WFS URL

• Result data
  – Model (ruleset) stored on file system
  – Projection (raster map) registered in SDL
Cluster

- 64 node, 128 processors
- 2 TB storage
- NPACI Rocks
- Sun Grid Engine scheduler
- Exposes a HTTP REST service
  - Run openModeller (GARP or other algorithm)
  - Get status
  - Get result data
Workflow Controller (Pipeline)

- Could simply generate jobs
- Currently
  - Harvests from local GBIF mirror
  - Generates jobs per species
- Reproduce LM1 but
  - with refined data
  - scalable system
  - focus on services
Overall system

• Standalone system
  – only outside connection is REST service
  – easily moved to smaller/larger system
  – or combine multiple systems for failover
• Interface - easier than existing SOAP
• High throughput not rapid eval of single model
Implementation Status

• Core components operational
  – harvest data
  – generate jobs
  – output projections
  – store back to SDL

• No user interface yet

• W*S so existing viz solutions easy
What does the future hold?

- Fine-tune
- Data quality
  - taxonomic
  - spatial
- Multiple
  - algorithms
  - projection scenarios
- Analysis services
Acknowledgements

- **Funding**
  - NSF Award (EPSCoR 0553722)
  - Kansas Technology Enterprise Corporation

- **openModeller**
  - CRIA and more

- **Original GARP**
  - David Stockwell, SDSC

- **Environmental data**
  - Climate Research Unit
  - International Panel on Climate change
  - Normalization BDWorld, Tim Sutton, Pete Brewer

- **GBIF and contributing collections**
- **Lifemapper1 Team, especially Ricardo Pereira**
Questions?