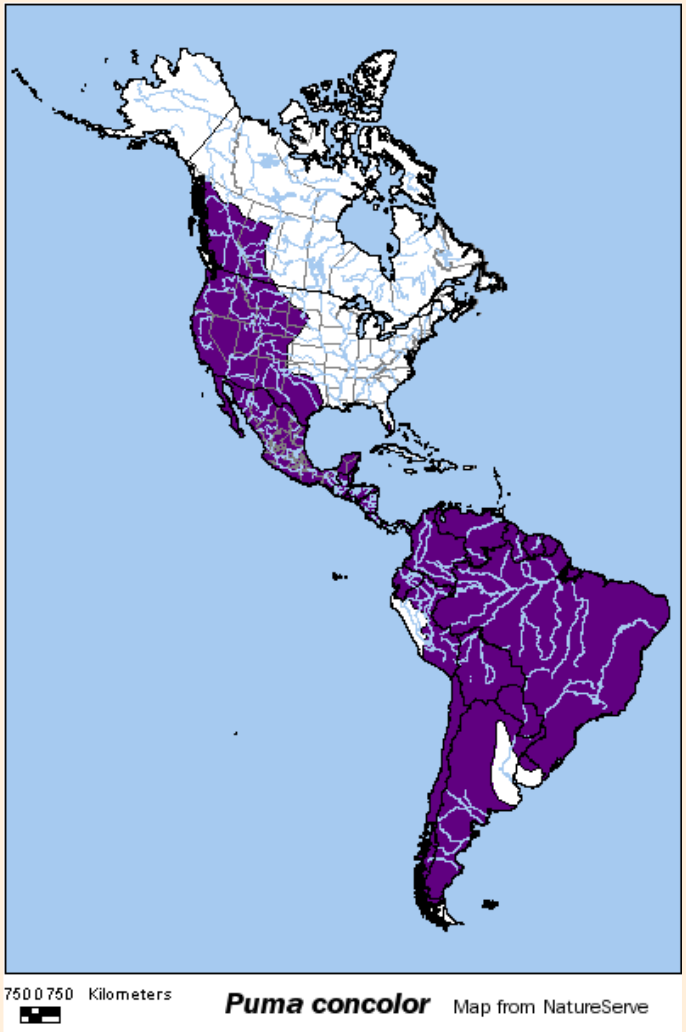
Species Distribution Modelling Using An Open Source Geospatial Software Stack

Allan D. Hollander
Information Center for the Environment
University of California, Davis
adhollander@ucdavis.edu

FOSS4G Conference — 26 September 2007

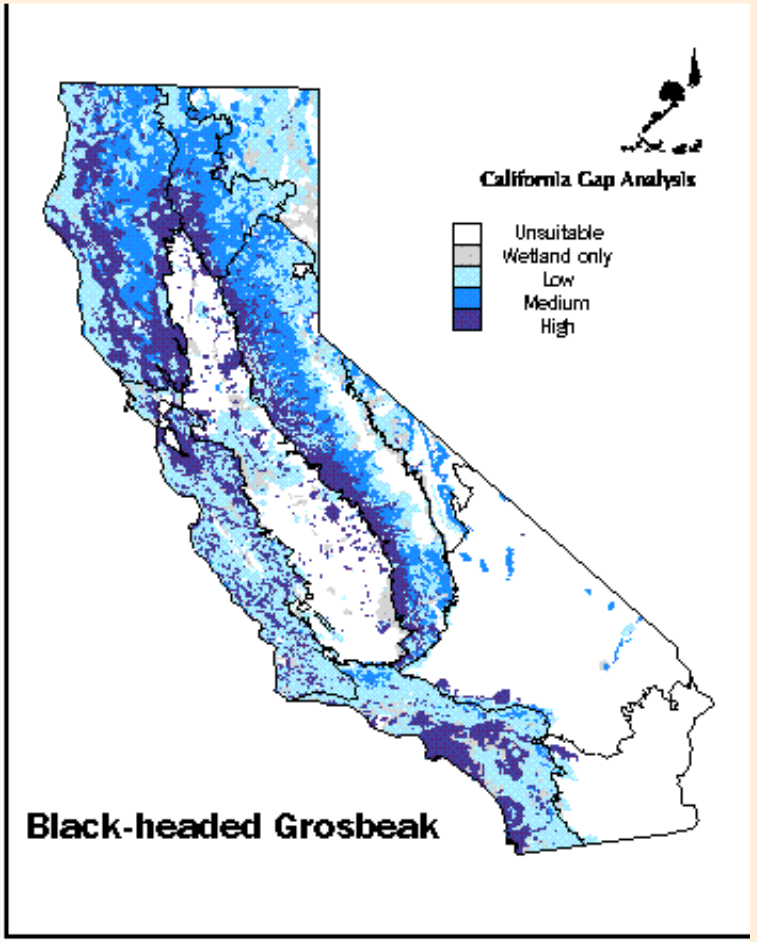
Why make range maps of species?

- Conservation of threatened species
- Modelling of invasive species
- Natural history interest



Approaches to Species Distribution Mapping

- Coarse-scale range maps
- Expert opinion models using habitat
- Statistical approaches with occurrence data



The Nature of Observation Data

- It is very unusual to have systematic surveys (e.g. Breeding Bird Survey).
- Most observations from very localized efforts (e.g. masters' theses, environmental impact reports).
- Museum records another source, but limited to collection efforts that are usually old.

Basics of statistical niche modelling - I

1. Collect observations of species together with absence data if possible.
2. Create a spatial stack of environmental data (elevation, climate, habitat type, etc.).
3. Overlay occurrence points on data stack to create data table of environmental factors by occurrence records.

Basics of statistical niche modelling - II

4. Use statistical model to distinguish presences from {pseudo}absences.
5. Apply statistical model to spatial stack to create predicted distribution map.

Choice of general vs. special-purpose tools

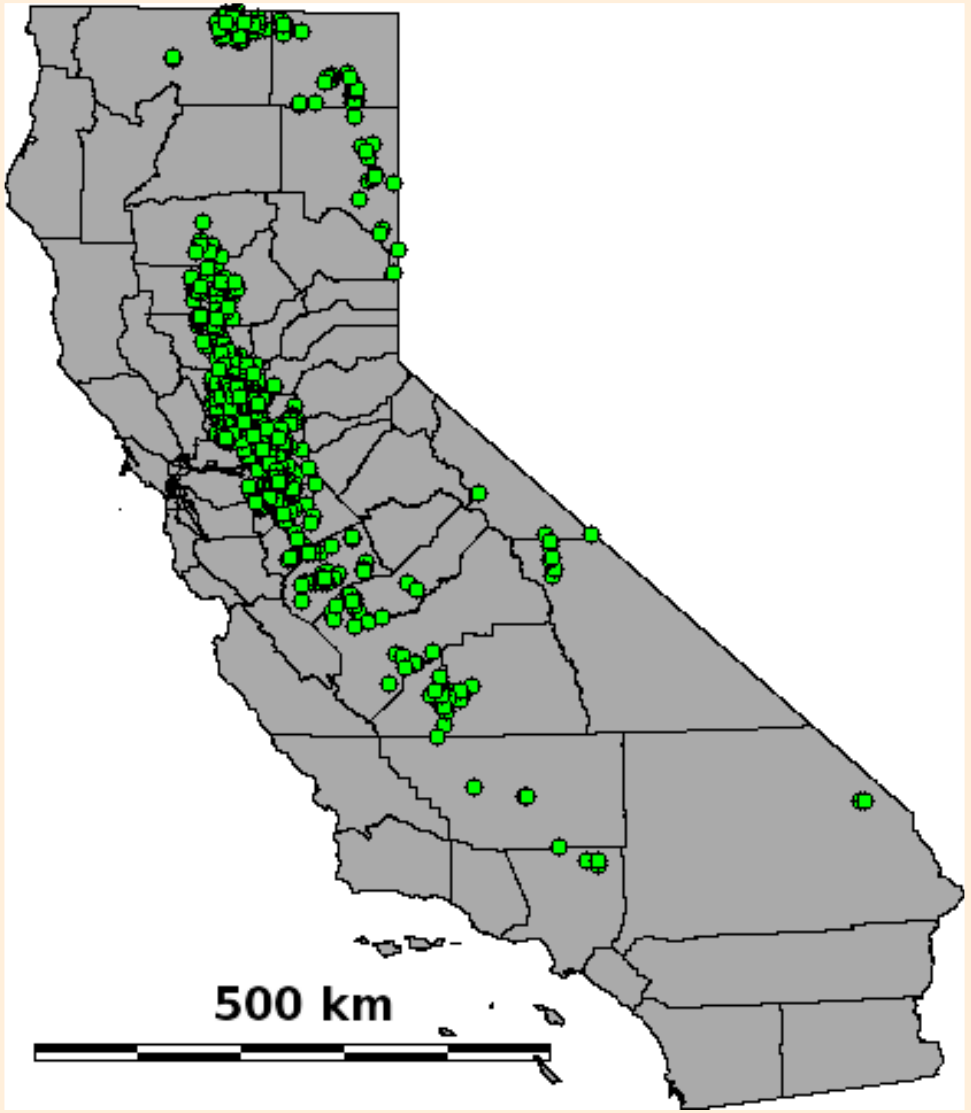
- General purpose: GRASS, R and friends
- Special purpose: OpenModeller, BioMapper, Species Analyst, DesktopGARP

Why GRASS and R?

Why I chose a general-purpose approach:

- General spatial data management needed
- Raster analysis capabilities
- May need to use own algorithms
- Assembled datasets will be used in other contexts

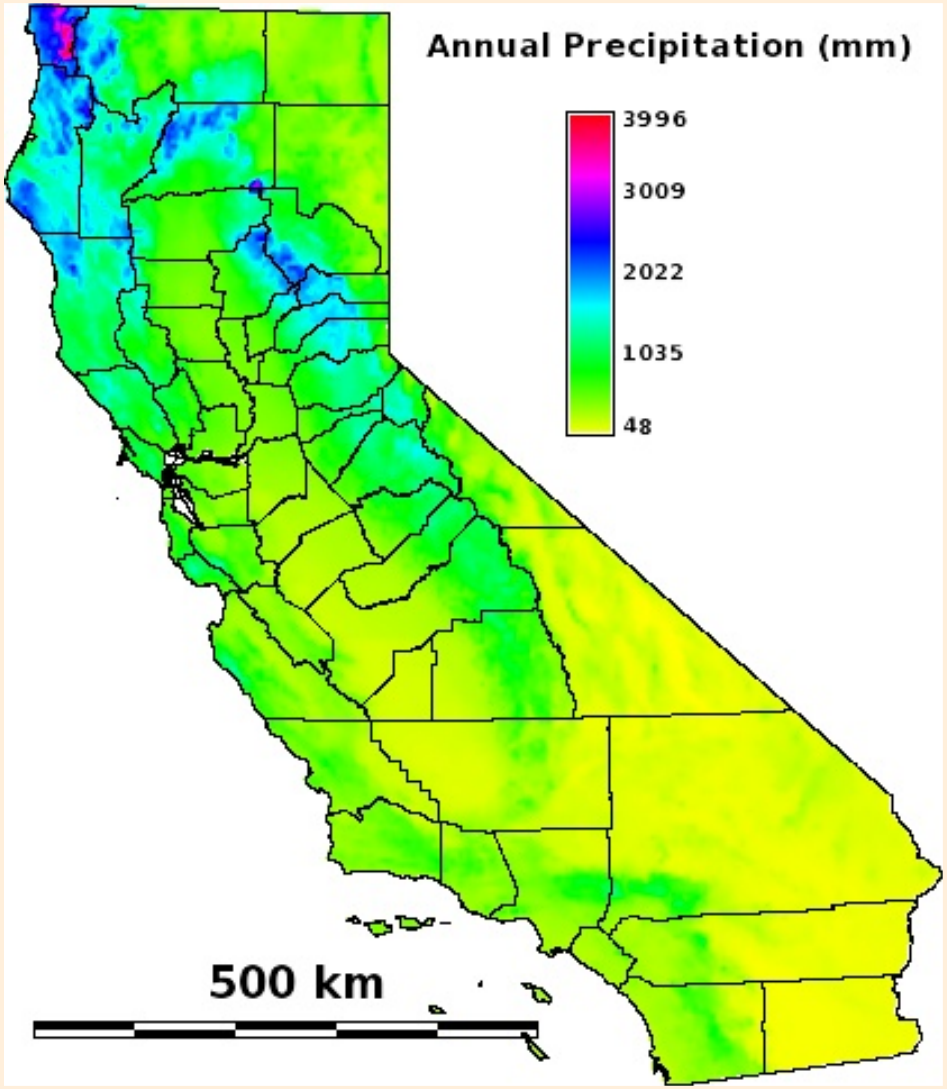
California observations of the Swainson's Hawk



Data from California Natural Diversity Data Base (CNDDB)

Environmental Data Layers

- Climate (precipitation, temperature, humidity)
- Elevation, slope, and aspect
- Soil characteristics
- Habitat characteristics



How to produce occurrence data table?

GRASS approach — `v.what.rast` command

— doesn't handle raster stack, only single layers

— appends results into attribute table of the vector that is being overlaid on the raster

- Both these limitations can be scripted around, but there's an alternative that is specialized for this problem

StarSpan

http://starspan.casil.ucdavis.edu/

- Specialized for fast sampling of rasters using vector geometries
- Open source BSD-style license
- C++ application, requires GDAL and GEOS.
- Can work with any raster and vector layers recognized by GDAL and OGR
- Command line interface

Sampling the raster data stack

- We don't have absence data, so we cheat, and use random points as ‘pseudoabsences’.
- We use a bash script to loop through list of rasters in GRASS, sampling data values using StarSpan. We loop through twice, once for presences, once for pseudoabsences.
- The output is a four-column table, with point id, presence flag, name of the raster, and its value at point location.

The R software suite

- Freeware computing environment easily used for statistical analyses
- Extremely important in statistical research communities
- GPL licensing
- Most functionality available through importable packages, of which there are about 1000 in the standard distribution.

R-Spatial and spgrass6

- There are a large number of spatial packages in R, e.g. geostatistics, point pattern analyses
- R-spatial project is an effort to integrate these, providing uniform set of classes and methods for points, lines, polygons and grids.
- spgrass6 is an interface between GRASS6 and R
- Important spgrass6 commands: `gmeta6`, `readRAST6`, `writeRAST6`, `readVECT6`, `writeVECT6`

Modelling the species distribution in R

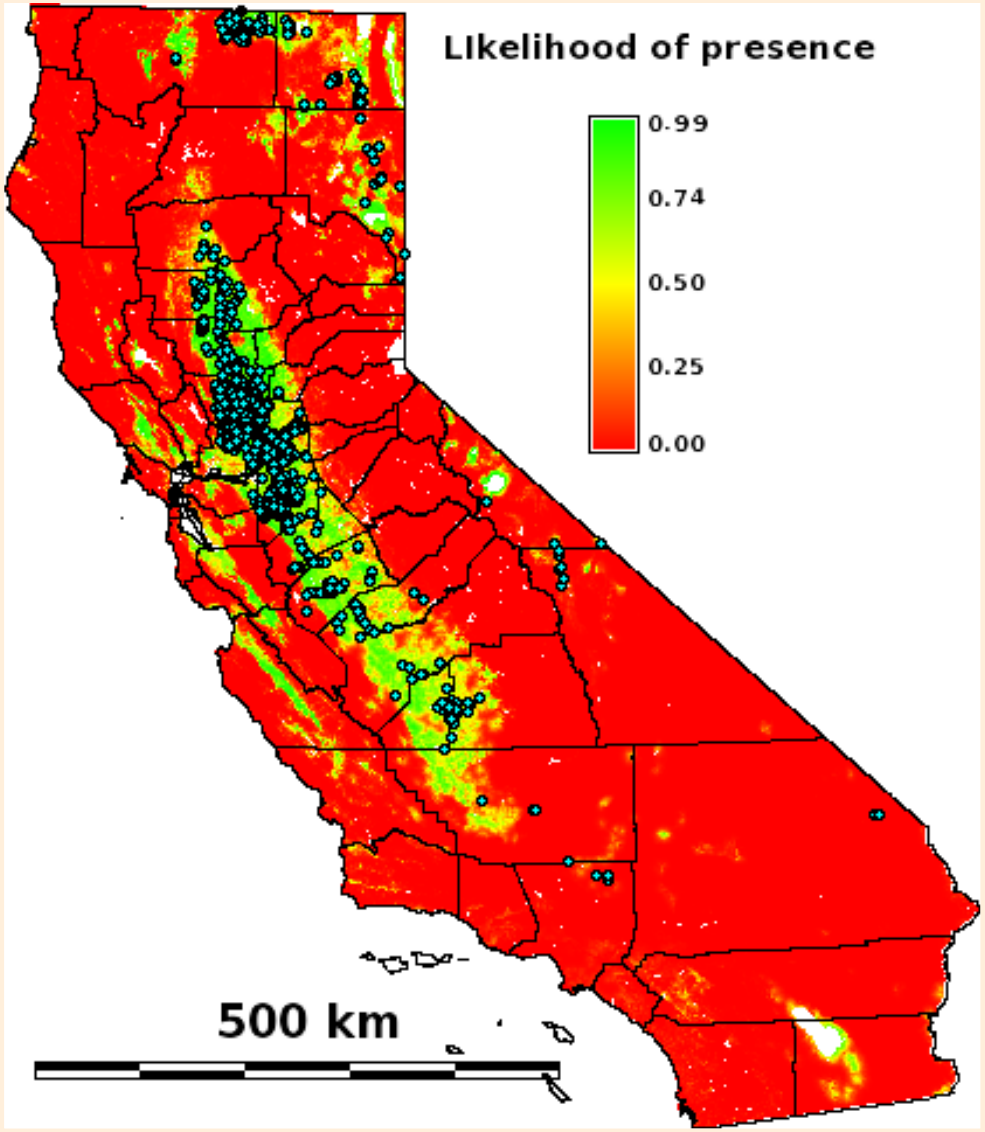
1. Load 4-column table from StarSpan run into R and recast into data frame.
2. Decide on model type (e.g. logistic regression, classification tree, random forest).
3. Generate model from data frame.
4. Load rasters into R with `readRAST6`.
5. Predict new raster in R.
6. Write new raster to GRASS with `writeRAST6`.

Some R code

```
> lmformula <- presence ~ annprecip + dem100m +
  janmintemp + julmaxtemp + julprecip + rhsummer
  + slope100m + statsgoclay + statsgoloam +
  statsgoom + statsgoph + statsgosand + statssilt

> logit1 <- glm(lmformula, family="binomial",
  data=specname)

> gridtemp <- predict(logit1, califbigdf,
  type="response")

> califgtemplate@data$annprecip <- gridtemp

> writeRAST6(califgtemplate,
  c(paste(specsummaryfile, "g", sep="")))
```

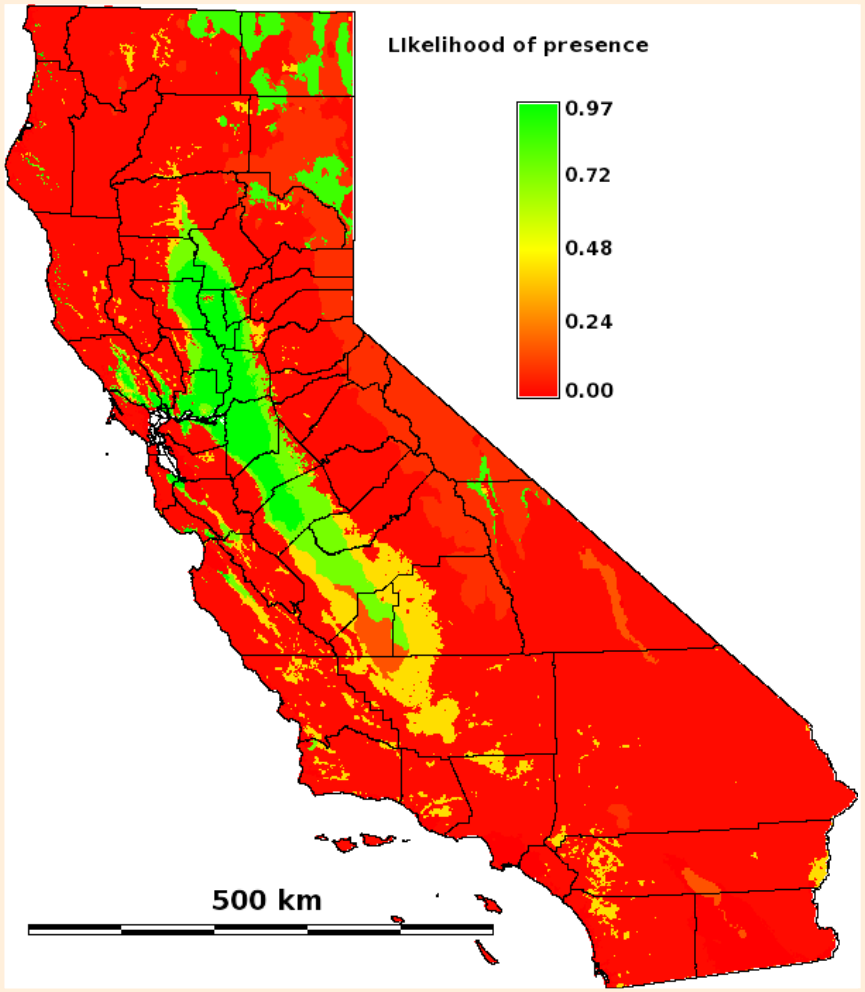
We now have a predicted distribution map in GRASS!

Predicted distribution of Swainson's Hawk modelled using logistic regression.

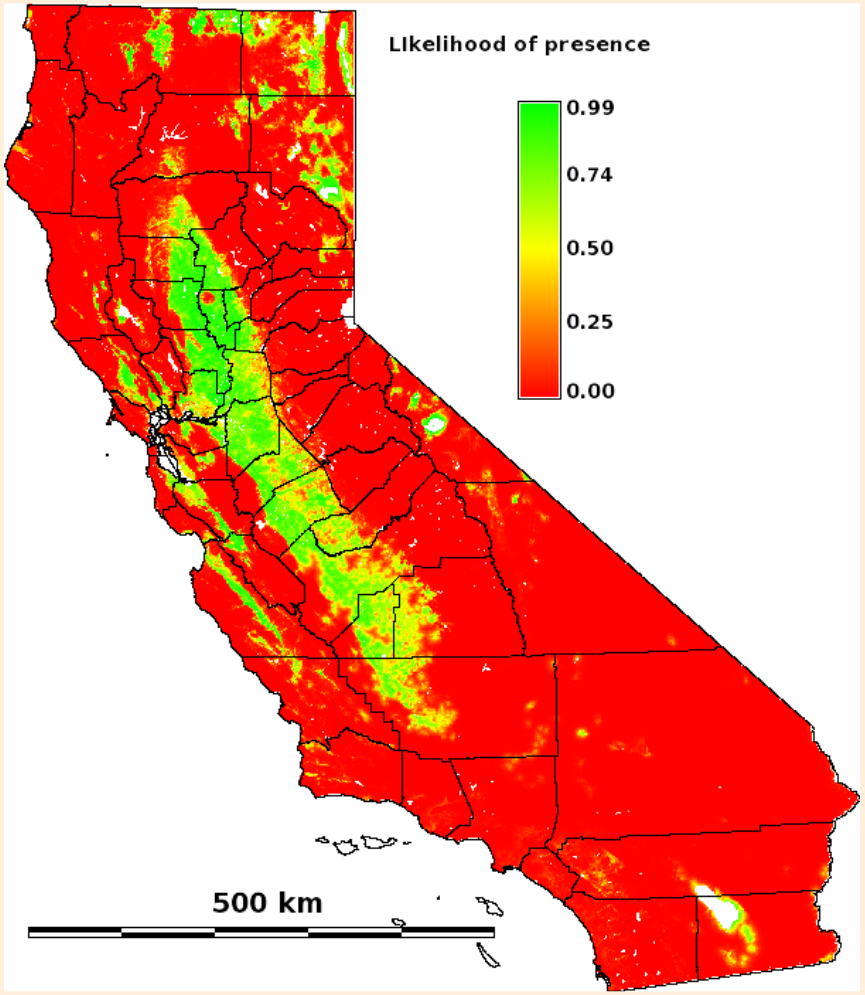


But how do we assess differences between models?

Classification tree



Logistic regression



Future directions

- Making species distribution data available in Semantic Web-friendly manner (Linking Open Data project).
- Making species distribution modelling functionality into a web service with PyWPS?
- Workflow improvements with Kepler??

Acknowledgements

- Information Center for the Environment, Department of Environmental Science and Policy, University of California, Davis
- California Department of Transportation
- National Science Foundation
- National Biological Information Infrastructure, US Geological Survey