InteRpretations: Soil interpretations outside of NASIS

Joseph R Brehm and Colby W Brungard
New Mexico State University, Department of Plant and Environmental Science
3/23/2021
Introduction: Overview

NASIS interpretations provide key information on expected soil behavior
• Hydrologic Soil Group: Run-off potential
• Valley Fever Habitat: Range of a deadly pathogen
• Dwellings With Basements: Limitations to construction

These interpretations are only accessible through NASIS
• Limits user access
• Prevents outside data sources (e.g. DSM products)
Introduction: Goals

We proposed to develop a prototype interpretations engine that produces soil interpretations using digital soil maps of key soil properties.

Interpretations working group chose 5 interpretations:
• Hydrologic Soil Group (HSG)
• Soil Vulnerability Index (SVI)
• Wind Erodibility Group & Index (WEG/WEI)
• Dwellings With Basements (DWB)
• Valley Fever Habitat (VF)
Methods: Overview

• We have translated internal NASIS workings to a suite of R scripts, designed for multiple levels of proficiency

• At its most basic, the engine can run in as few as two steps: Load data, run function

• The InteRpretations engine will function with
  • Any data source
  • Raster, vector, and non spatial data
Validation

• Interpretations outside NASIS validated against NASIS output

• One targeted interpretation did not achieve parity: Wind Erodibility Group/Index
  • In NASIS, this is a complex block of code with no intermediate steps
  • Lack of intermediates limits our ability to bug hunt
What raster data sources would be useful here?
  - Looking for soils, topography, and climate

What are some variables that would benefit from higher spatial resolution?
Methods: Workflow 1 (HSG & SVI)

Calculations (HSG and SVI) are simple functions that make use of widely available data

• Step 1: Acquire data (processed, resampled, and in the right units)
  • HSG needs: limiting ksat, depth to restriction, and depth to water table
  • SVI needs: HSG, slope, and Kw soil erodibility factor

• Step 2: Run the single line function
  • hsg_calc(inputBrick) or svi_calc(inputBrick)
Study Area: Grand County, Utah
Input for Hydrologic Soil Group
(Grand County, Utah)

Source: POLARIS Properties

Source: gSSURGO
Input for Soil Vulnerability Index (Grand County, Utah)

**Kw Factor**
Source: POLARIS Properties

**Slope (%)**
Source: DEM

**Primary HSG**
Source: InteRpretations
• Interpretations (VF and DWB) make use of complex, precise soils data
  • Some data doesn’t exist outside of NRCS e.g. depth to water table
  • Some is complex derived data e.g. from USDA in-lieu-of textures

• We extract this complex data, provided with the engine
  • Starting dataset can immediately be used
  • Includes the simpler data, e.g. mean annual precip
  • Immediately sub in new data sources, e.g. PRISM mean annual precipitation
Methods: Input

• NASIS-extracted data creates the baseline

• Any NASIS data may be substituted by data from another source
  • PRISM
  • POLARIS / Soil Grid
  • DEM products, e.g. slope
  • Interpolations from your own field data, etc.
Methods: Workflow 2 (VF & DWB)

- Step 0: Advanced users extract baseline raster data
  - This process is very slow. To do: run over CONUS

- Step 1: Acquire baseline data
- Step 2 (Optional): Substitute rasters
  - `inputBrick$slope <- newSlope`
- Step 3: Run the interpretation R function
  - `vf_calc(inputBrick)` or `dwb_calc(inputBrick)`
Input for DWB & VF Interpretations
(Grand County, Utah)

• Unlike calculations, these have many, complex input variables
  • Dwellings with Basements: 17
  • Valley Fever: 15

• Get ready...
VF Example Input: Grand County, Utah
Valley Fever Habitat

(This interpretation is more affected by NA data than the rest)
Other benefits

• By moving functions out of NASIS, the interpretations code is more easily accessible

• Interpretation code can be modified by any user, e.g. for sensitivity analysis
Acknowledgements

• Authors of the soilDB and aqp packages, and other starting code
  • Especially Dylan Beaudette, Jason Nemecek, and Stephen Roecker

• The interpretations working group
  • Bob Dobos, Suzann Kienast, Colby Brungard, Jim Thompson, Jason Nemecek, Jessica Philippe, Stephen Roecker, Travis Nauman, Skye Wills, and Zamir Libohava
Questions?

• What raster data sources would be useful here?

• What are some variables that would benefit from higher spatial resolution?
Technical Details Overview

- Code structure
  - HSG/SVI: data tree
  - VF/DWB: access evaluations by NASIS ID

- Accessing NASIS properties

- Properties to raster
Code Structure

- HSG & SVI: decision trees with a navigator function
  - Uses `data.tree`
  - Tree splits are defined in a `.rdata` file
  - Trees are written in R, can be edited there

```
tr.hsg <- Node$new(
  name = "Site",
  nextvar = "rl",
  nextlogical = "rl < 50; rl >= 50 & rl < 100; rl >= 100 | is.na(rl)"
)
```

- Navigator evaluates the statements contained in “nextlogical”, then proceeds to the node (1-3) associated with the true logical
- Results in a slow function. Raster math would have been faster.
Code Structure

- DWB & VF: approximated fuzzy logic equations
  - Evaluations are extracted from NASIS via XML
  - Functions are called by evaluation ID (hard coded)
  - Functions are fit (on the fly) to data specified in the XML
  - All other processing is hard coded
  - Will NEVER achieve 100% parity

```r
outdata$fuzzsar <- evalbyeid(42999, indata$sar, sig.scale = 5) %% replace_na(0)
outdata$fuzzgypsumcontent <- evalbyeid(42991, indata$gypsumcontent) %% (0.5) %% replace_na(0)
outdata$fuzzvf <-
  sort(
    sort(outdata$fuzzwrd) *
    sort(outdata$fuzzwatergatheringsurface) *
    sort(outdata$fuzzom) *
    sort(outdata$fuzzsurfscat) *
    outdata$fuzzclimate *
    outdata$fuzzchem) / 0.95
```
Accessing NASIS Properties

- Representative values are extracted from NASIS web reports
  - Identify properties used in the interpretation, specified in a NASIS export
  - Get RVs for all relevant soil components

- NASIS web reports round all values to integer
  - This causes some problems, e.g. albedo
  - Workaround: gSSURGO, sometimes direct sometimes derived
Making Rasters

• Property RVs are then attached to SSURGO map unit polygons
  • coiid != cokey, so a hacked name-based join is used
  • Area majority component is assigned to the polygon
    • If you run the interp on the RV data frame then do your spatial join later you can vary this

• Rasterize!
Questions? (Again)