November 6th, 2025

Living Waters: Drilling Site Suitability Analysis

Outline

- Previous Work
- Study Area
- Project Objective
- Criteria
- Approach
- Code
- Results
- Future Work
- Acknowledgements
- Q & A

Before we arrived, there were extensive talks with the Mayans of Quisache to assess their needs.

Mayans provide local knowledge. For example, one potential new drilling area was prone to flooding.

Mayans value sharing food

Left A small child gathers water

Below Communal laundry cleaning area

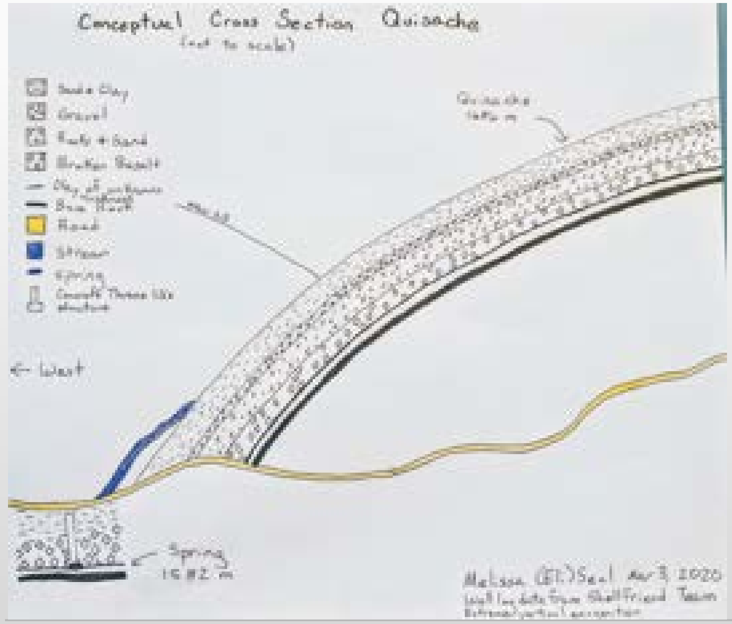
Drinking water must be carried a half-mile walk up steep terrain

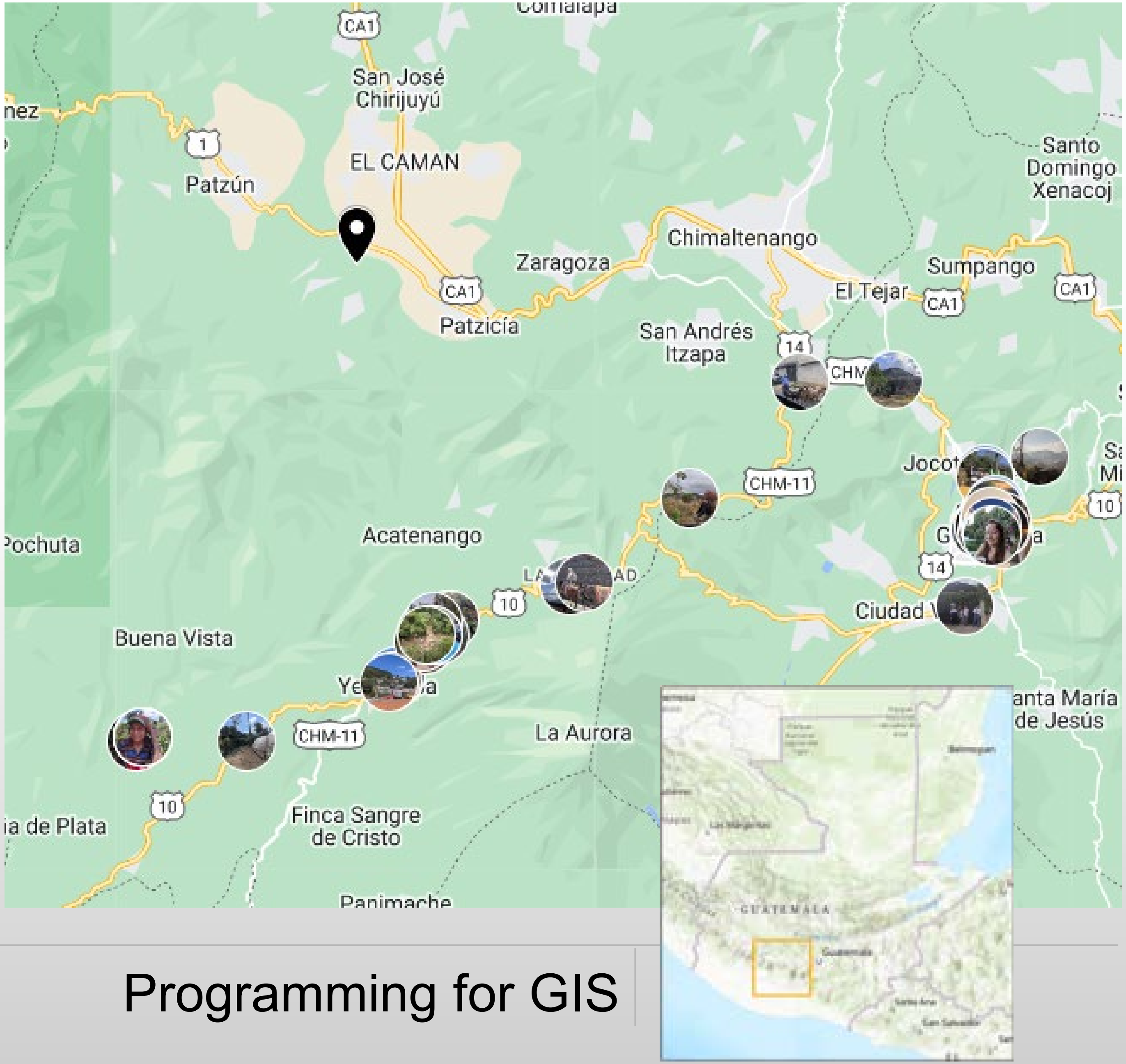
Previous work and motivation

- In March 2020 I worked on a survey trip in Quisache, Guatemala on a volunteer project with a geophysicist friend Berkeley Glass
- A previous well had been drilled there but it came back dry
- The well is intended to provide water for a school of 500 – 600 children
- Conducted a VES survey that shows a layer with a resisitivity consistent with water between 53 m and 85 m
- Previous well had been 50 m
- No use of GIS in original work and only using screenshots from google maps

Study Area

- Location of previous wells plus new location they are to begin drilling in January 2025
- Patzún and Patzicía, Guatemala





Programming for GIS

Criteria from MAGA Groundwater Potential Study

Rock types (lithology)

Geomorphology

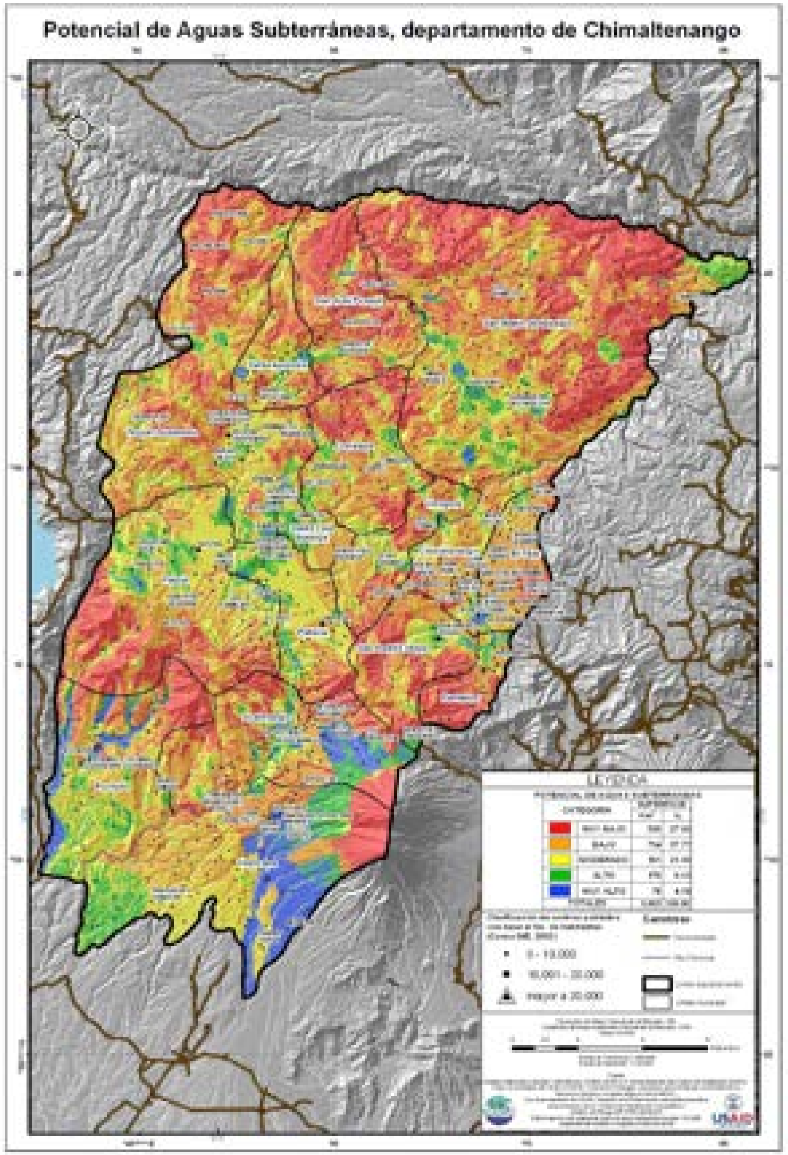
Geological structures

Rainfall

Terrain slope

Density of river currents

Aquatic bodies



Reconciling Project Objectives

Living Water Project Objective

A combined raster that informs engineers and project managers which locations are better candidates for drilling water wells based off water potential, slope and distance from streams

Programming Class Project Objective

Aggregate data and compare previous well log information to the suitability analysis

Write function to retrieve well information for the closest well depending on input coordinates

Show I can use Python for GIS to create a useful product.

Criteria from Living Waters for Drill Site Location

More than 30 m from river

Slope less than 15% preferred

Likely presence of water according to "*water potential*" data layer created by USAID and Guatemalan Ministry of Agriculture, Livestock and Food (*MAGA*)

Analysis Approach

Desktop Suitability Analysis → Code Jupyter Notebook → Query & Validate

Desktop Suitability Analysis

- Work through process
- Have results to compare to
- Learn ArcGIS Pro

Code Jupyter Notebook

- Scalable and customizable
- Can look at different areas
- Later, can fit the model to the well logs with "AB testing"

Query & Validate

- Yield and depth from nearest well using user-input coordinates
- Validate against original water potential data
- Validate against desktop suitability analysis

Validation: Compare Suitability Scores to Existing Well Locations

- Are existing wells located in areas with high modeled suitability scores?
- Is there a relationship between yields and modeled suitability scores?
- Rating success
  - ✓ Well can be dug – avoid cobblestones and boulders
  - ✓ Current Living Water (LW) standard is to yield 20 liters/minute (~5 gallons/minute)
  - ✓ But standard being updated to 110 liters/minute (~30 gallons/minute)

Code for site suitability analysis

Import libraries, set environment settings and check to make sure that it is in the correct workspace

```
import arcpy
from arcpy.sa import *
import os
arcpy.env.addOutputsToMap = False
from arcpy import env
arcpy.env.snapRaster = r"F:\GISC_2435\Projects\LivingWaters\LW_V10\DEM_Guatemala_SRTM_30.tif"
arcpy.env.cellSize = 30 #set to 30 m because this is the biggest cell size of my rasters
arcpy.env.workspace = r"F:\GISC_2435\Projects\LivingWaters\LW_V10\Inputs.gdb"
arcpy.env.scratchWorkspace = r"F:\GISC_2435\Projects\LivingWaters\LW_V10\Scratch.gdb"
arcpy.env.overwriteOutput = True

if arcpy.Exists("rivers_worldbank"):
    print("Your workspace is set correctly.")
else:
    print("Your workspace is wrong.")
```

Access data

```
DEM = arcpy.Raster(r"F:\GISC_2435\Projects\LivingWaters\LW_V10\DEM_Guatemala_SRTM_30.tif")
WP = arcpy.Raster("pot_a_sub_")
studyarea = "StudyAreaPandCh"
rivers = "rivers_worldbank"
```

Project layers into WGS 1984 UTM Zone 15N

```
arcpy.management.Project(rivers, "riversproject", 32615)
arcpy.management.ProjectRaster(WP, "wpproject", 32615, "", 30)
arcpy.management.ProjectRaster(DEM, "demproject", 32615, "", 30)
```

Clip rasters to study area

```
arcpy.analysis.Clip("riversproject", studyarea, "riversclip")
arcpy.management.Clip("demproject", studyarea, "demclip")
arcpy.management.Clip("wpproject", studyarea, "wpclip")
```

Code for site suitability analysis

Convert DEM to slope

```
outMeasurement = "DEGREE"
slopeclip = Slope("demclip", outMeasurement, "", "PLANAR", "METER")
```

Create multiple buffers around the buffer in order to exclude the area near the river. This is because near rivers it tends to be cobblestones which is a layer that tends to end in well collapse and so that the wells will not flood when the water level is higher. Buffering at 50 m because when I buffer at 30 m which is the raster size, parts of the river still get included sometimes. I am less certain if the 100 m buffer should be included in this analysis.

```
distances = [50, 100]
bufferUnit = "meters"
arcpy.analysis.MultipleRingBuffer("riversclip", "rivermulti", distances, bufferUnit, "", "ALL")
```

Convert multiple ring buffer polygon layer to raster

```
arcpy.conversion.PolygonToRaster("rivermulti", "distance", "rivermultiraster")
```

Reclassify all layers. Areas that should always be excluded are set to zero

```
reclassriver = Reclassify("rivermultiraster", "Value", RemapValue([[50,0],[100,1],["NODATA",2]]))
wpreclassify = Reclassify("wpclip", "VALUE", RemapValue([[1,0],[2,1],[3,2],[4,3],[5,4]]))
slopereclass = Reclassify(slopeclip, "Value", RemapRange([[0,14,2],[14.01,30,1],[30,90,0]]))
```

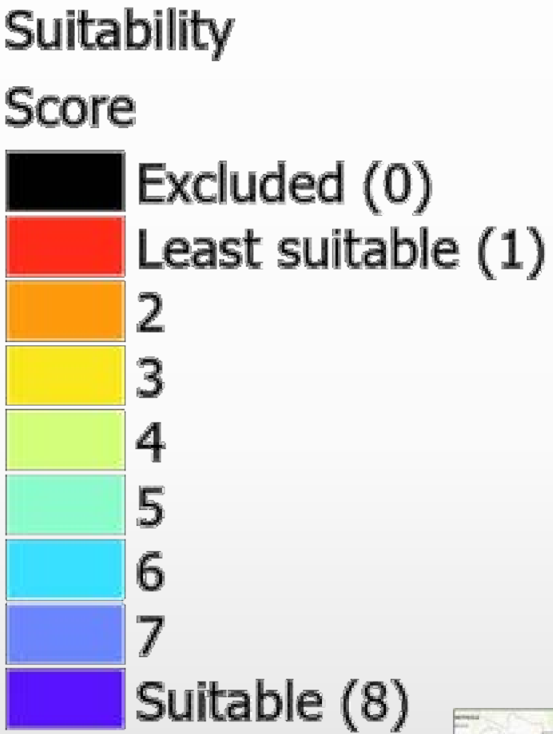
Raster math to create a combined raster that takes into account the combined criteria

```
drillinglocations =  reclassriver * wpreclassify * slopereclass
drillinglocations.save("DLinPandCh")
```

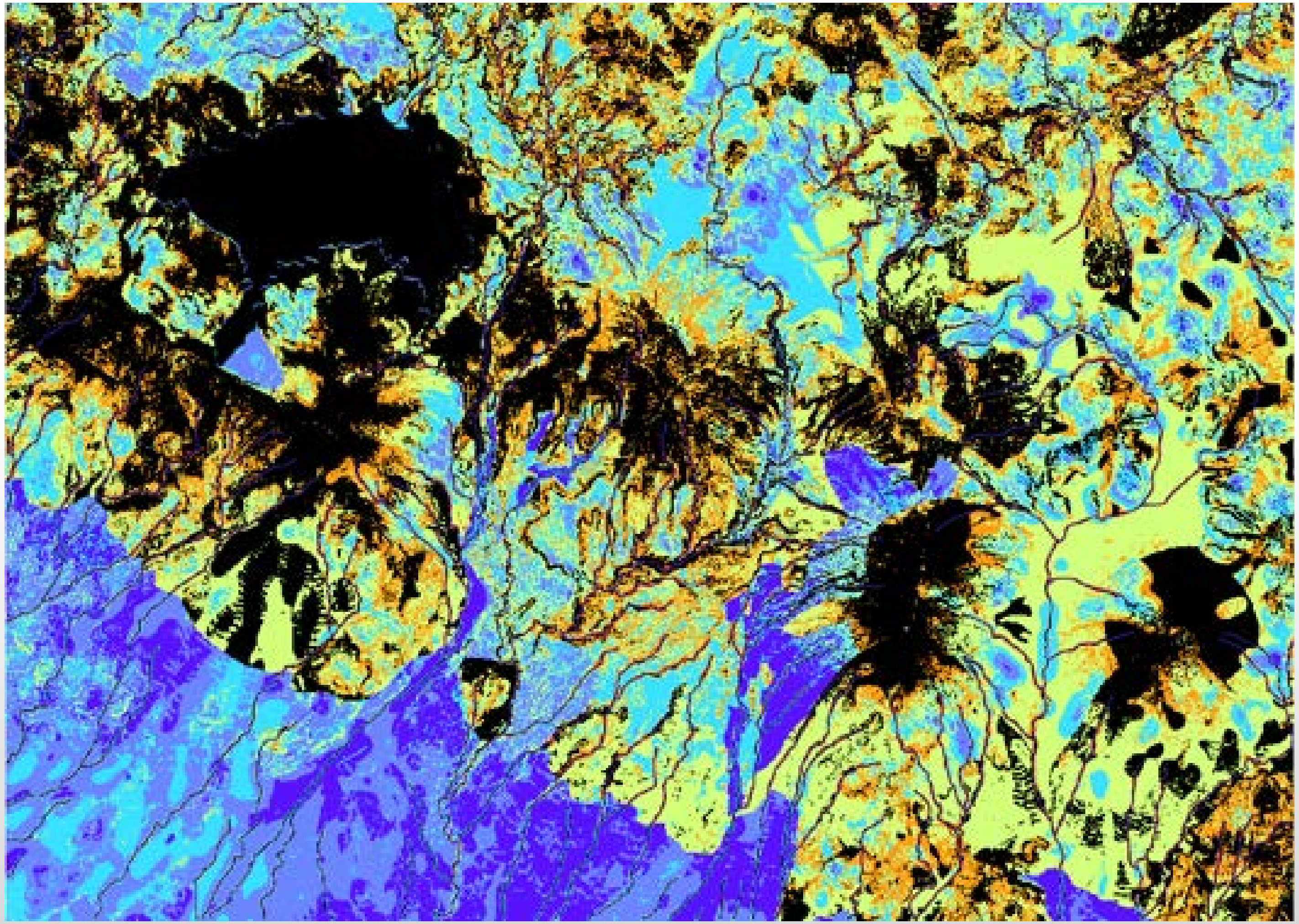
```
dlreclass = Reclassify(drillinglocations, "Value", RemapValue([[6,5],[8,6],[12,7],[16,8]]))
dlreclass.save("DrillingLocations")
```

```
arcpy.management.AddField(DLReclasify, "Location_Rating", TEXT, "", "", 16, "Location Rating")
```

Suitability Scores from Jupyter Notebook code



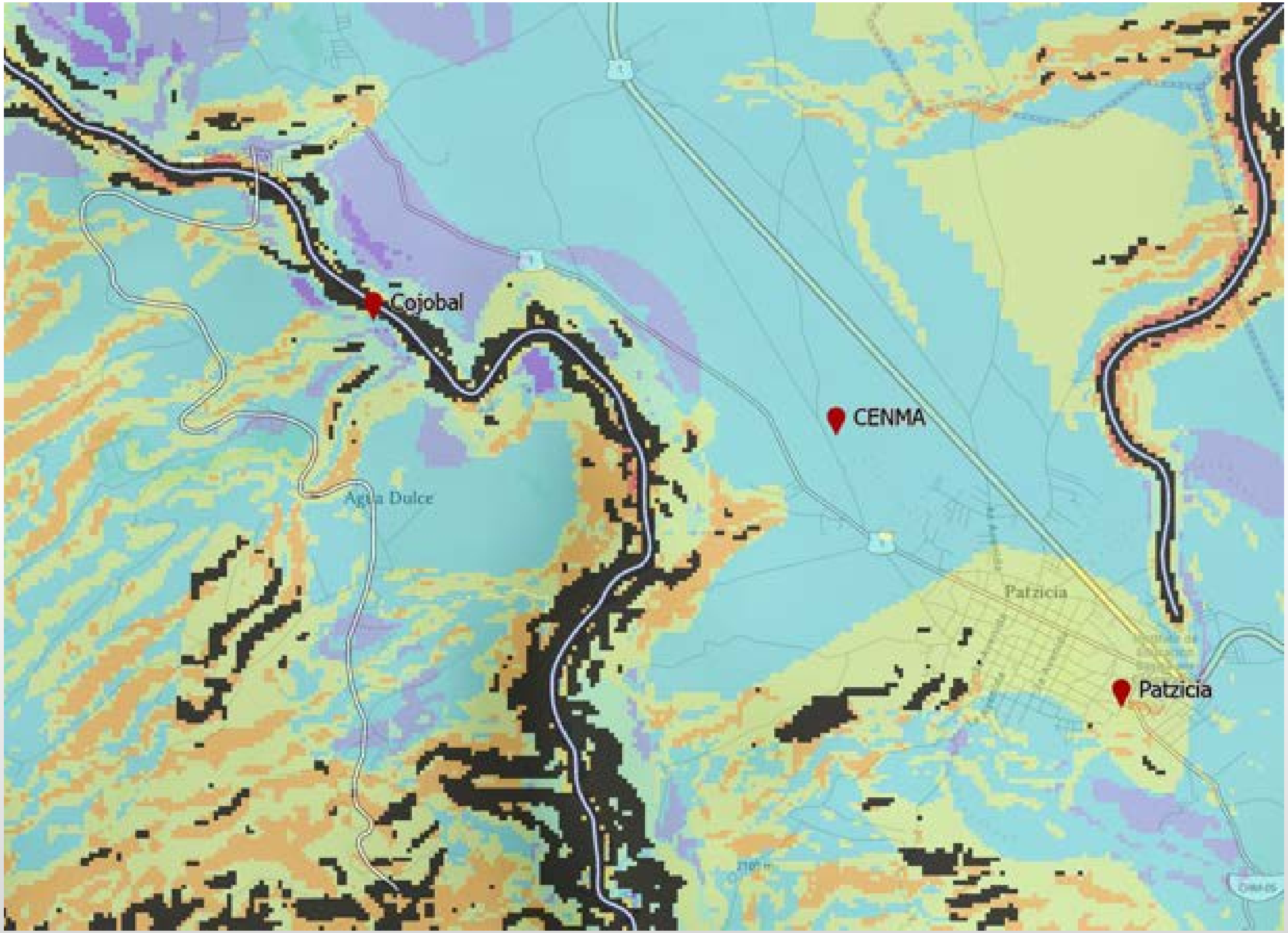
Lakes symbolized as black as you can't drill in a lake either



New drill sites with site suitability overlay

Suitability Score

- Excluded (0)
- Least suitable (1)
- 2
- 3
- 4
- 5
- 6
- 7
- Suitable (8)

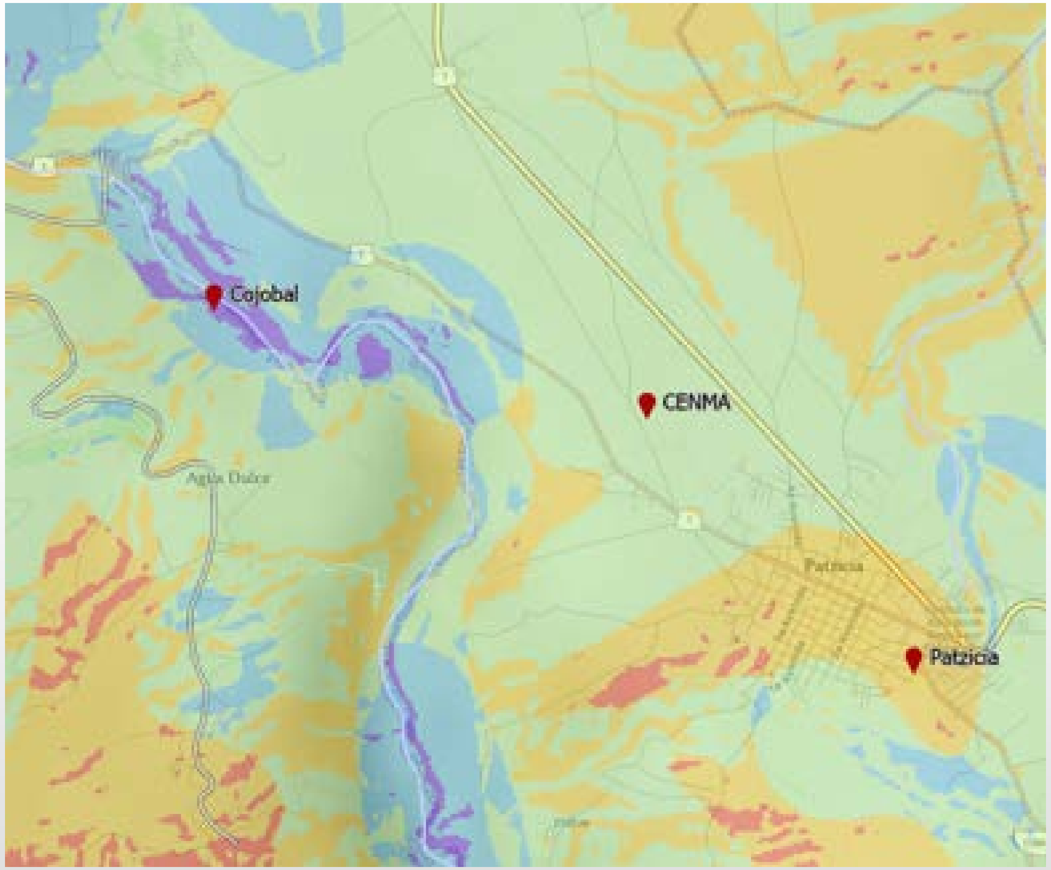


Site Suitability Analysis

Water Potential Map

Low Potential (1)

High Potential (5)



Validation:
Comparing suitability scores to well logs

Awaiting well log information from Living Waters

Most logs are still on paper documents and will need to be digitized
For now, I produced dummy well log data for a dry run

```
outGDB = r"F:\GISC_2435\Projects\WellDistance\DummyWell.gdb"
conFC = r"F:\GISC_2435\Projects\WellDistance\DummyWell.gdb\StudyAreaPandCh"

arcpy.management.CreateRandomPoints(outGDB, "dummywells", conFC, "", 100, 1000)
```

```
arcpy.management.DefineProjection("dummywells", 32615)
```

```
arcpy.management.CalculateField("dummywells", "depth", "random.randint(0,150)","PYTHON", "import random", '
arcpy.management.CalculateField("dummywells", "yieldgpm", "random.randint(0,50)","PYTHON", "import random",
```

```
ExtractValuesToPoints("dummywells", r"F:\GISC_2435\Projects\WellDistance\DummyWell.gdb\DLinPandCh", "dummywp")
```

```
wpdf = pd.DataFrame.spatial.from_featureclass(r"F:\GISC_2435\Projects\WellDistance\DummyWell.gdb\dummreclass")

wpdf
```

	OBJECTID	CID	depth	yieldgpm	POINT_X	POINT_Y	RASTERVALU	SHAPE
0	1	1	16	5	678708.634	1590473.2702	0	{"x": 678708.6339999996, "y": 1590473.27019999...
1	2	1	71	42	748895.8572	1636474.6715	7	{"x": 748895.8572000004, "y": 1636474.67149999...
2	3	1	0	14	743068.7456	1635540.3237	0	{"x": 743068.7456, "y": 1635540.3236999996, "s...
3	4	1	125	7	718345.0353	1623414.8758	5	{"x": 718345.0352999996, "y": 1623414.87580000...
4	5	1	39	17	720810.5997	1629739.5445	6	{"x": 720810.5997000001, "y": 1629739.54450000...
...	...	...	...	...	...	...	...	...
95	96	1	104	3	735694.327	1609749.3219	4	{"x": 735694.3269999996, "y": 1609749.32190000...
96	97	1	48	32	697474.4874	1611872.2137	2	{"x": 697474.4874, "y": 1611872.2137000002, "s...
97	98	1	141	27	717827.2648	1606436.8128	0	{"x": 717827.2648, "y": 1606436.8127999995, "s...
98	99	1	90	22	679495.5021	1601990.5033	6	{"x": 679495.5021000002, "y": 1601990.5033, "s...
99	100	1	145	9	686739.538	1626867.3101	8	{"x": 686739.5379999997, "y": 1626867.31010000...

100 rows × 8 columns

Validation: Comparing suitability scores to dummy well logs

```
wpdf['RASTERVALU'].value_counts()
```

```
RASTERVALU
4    26
0    23
2    16
7    11
6    11
8     9
5     4
Name: count, dtype: int64
```

23% of random locations were excluded (suitability = 0)
9% were rated as the best locations to drill (suitability = 8)

```
df5 = len(wpdf[wpdf["yieldgpm"]>=5])
print(df5)
```

```
87
```

87% pass the lower threshold yield of 5 gpm

```
wpdf[wpdf["yieldgpm"]>=30]
df30 = len(wpdf[wpdf["yieldgpm"]>=30])
print(df30)
```

```
39
```

39% pass the upper threshold yield of 30 gpm

Warning! This is dummy data

Validation: Compare to USAID and MAGA's Water Potential layer

```
wpdf['RASTERVALU'].value_counts()
```

```
RASTERVALU
2    32
1    24
3    20
5    15
4     9
Name: count, dtype: int64
```

23% of random locations were excluded
9% were rated as the best locations to drill

```
df5 = len(wpdf[wpdf["yieldgpm"]>=5])
print(df5)
```

```
93
```

For the dummy data, 87% pass the lower threshold

```
wpdf[wpdf["yieldgpm"]>=30]
df30 = len(wpdf[wpdf["yieldgpm"]>=30])
print(df30)
```

```
39
```

For the dummy data, 39% pass the upper threshold

Warning! This is dummy data

Locate nearest well to user-specified x,y and return depth to water and yield

```
xcoord = input("Type x coordinate: ")
ycoord = input("Type y coordinate: ")
print(xcoord)
print(ycoord)
```

```
706000
1614861
```

```
def calculated_distance(xcoord, ycoord, wellX, welly):
    xSquared = pow(wellX - float(xcoord), 2)
    ySquared = pow(welly - float(ycoord), 2)
    squareRoot = (xSquared + ySquared)**(0.5)
    return squareRoot

distance = calculated_distance(float(xcoord), float(ycoord), df['POINT_X'], df['POINT_Y'])
print(distance)
```

```
0     36600.273536
1     48033.377568
2     42446.746978
3     15018.944296
4     20993.450167
          ...
95    30131.085427
96     9034.224297
97    14520.713572
98    29464.183244
99    22696.186435
Length: 100, dtype: float64
```

Locate nearest well to user-specified x,y and return depth to water and yield

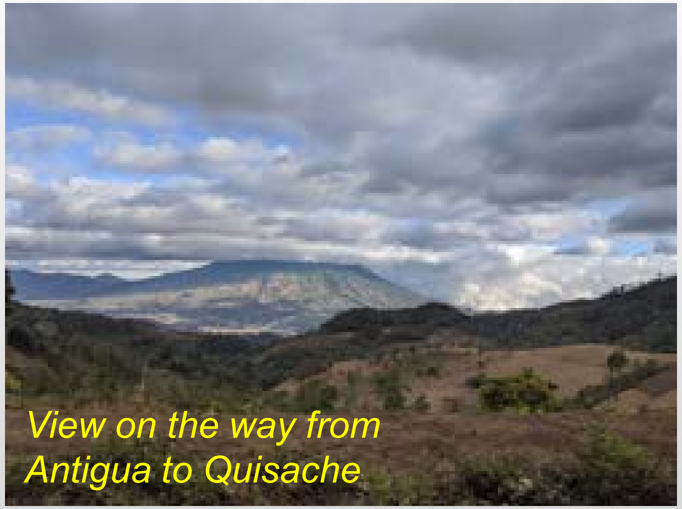
```
df = pd.DataFrame.spatial.from_featureclass("dummywp2")

df
```

```
df['distance']= distance
```

```
min_index = df['distance'].idxmin()
rowmin = df.loc[min_index]
print(rowmin)
```

```
OID                                                     46
CID                                                      1
depth                                                   27
yieldgpm                                                30
POINT_X                                         706241.665
POINT_Y                                       1621015.1933
SHAPE       {'x': 706241.665, 'y': 1621015.1932999995, 'sp...
distance                                       6158.936365
Name: 45, dtype: object
```



View on the way from Antigua to Quisache

Future Work

- Consider adding in buffer around lakes
- Digitize well logs and plug them into comparison analysis
- Add in other factors like NDVI and fracture traces to see how it compares to yields to water potential map
- AB Testing of model
- Follow-up to see if new wells dug using suitability scores from this analysis are more likely to be successful in completion and yields

View over city of Antigua

Lessons Learned

- How to work with a client to define objective and scope
- How to locate GIS-ready data from authoritative international sources
- Scientific and business Spanish
- File mismanagement is a great way to drive yourself nuts
- Don’t ever point a geoprocessing tool at source data, only at copies of source data
- Jupyter Notebook does not actually autosave, autosave before running any command and maybe even more frequently

Updates

- The first well that got dug on based on our recommendations came up dry
- Still awaiting well logs
- Lava tubes!

Acknowledgements

- Living Waters International including Berkeley Glass, Rob Pettigrew and Chris Hough
- Daniela Samanta Santos Lopez for sending me the water potential data from MAGA
- Sally Holl for the crash course on suitability analysis
- My programming professor Stephen Bond

melissa.k.seal@gmail.com

GIS Portfolio

Art Portfolio