TREE PROVENANCE DATABASES: ORGANIZATION & APPLICATIONS

MARCH 2020

OUTLINE

Database 1: Trial Results & Provenance Climate Data

Background
Database Organization
Database Application Example 1 □ Querying the Database
Database Application Example 2 □ Simple Transfer Distance Calculations in Python

Database 2: Test Site Information & Climate Data

Background
Database Organization
Database Application Example 3 □ Matching Test Site Climate to Future Climate

DATABASE 1: BACKGROUND

What does it contain?

- Trial results (height, survival, phenology data such as bud flush date)
- Provenance origin in decimal degrees
- Elevation at the provenance origin
- 65 Climate variables

Why do we want this information in a database?

- Difficult for researchers to comb through different sources to find usable data
- It is time-consuming and expensive to establish tree provenance studies
- We want to make sure available trial results are collected in one place so that researchers & forestry professionals save time and money

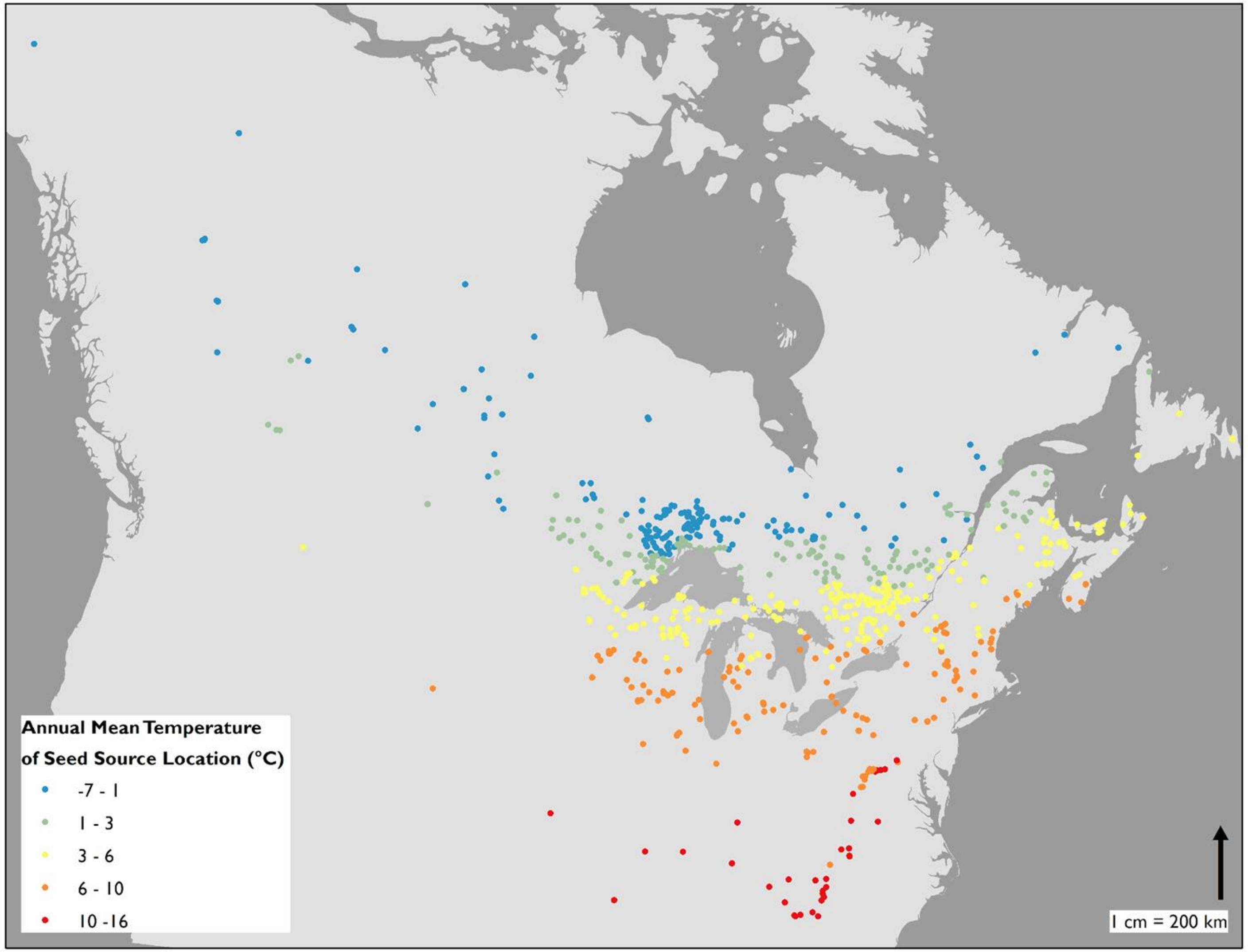
DATABASE 1: BACKGROUND

Species	Number of Test Sites	Spatial Range of Test Sites	Number of Provenances	Spatial Range of Provenances	Total Number of Variables Recorded
Jack Pine	7	ON, MN, NE, ME	132	NWT □ PEI, WI/MI □ NWT	24
Black Spruce	8	QC, NL, NB, ME	152	BC □ NL, WI/MI □ NL	15
White Spruce	9	ON, MN, ME	186	AK □ NL, SD □ AK	79
White Pine	26	ON, ME, VT, WV, PA, KY, MI, OH, MN, WI, MD, MA	167	MN □ NS, TN □ QC/ON	28
Red Oak	7	OH, MI, IN, IL, NE, KS, IN	32	MN □ ME, TN □ QC/ON	24
Tamarack	11	ON, NL, WV, MI	69	NWT □ PEI, MD □ NWT	25
Yellow Birch	1	WI	21	MN □ NS, GA □	4

DATABASE 1: BACKGROUND

How were the 65 climate variables generated?

- Obtained from existing climate models
- Estimates generated for each provenance
- Values represent long-term averages for the 1961-1990 period



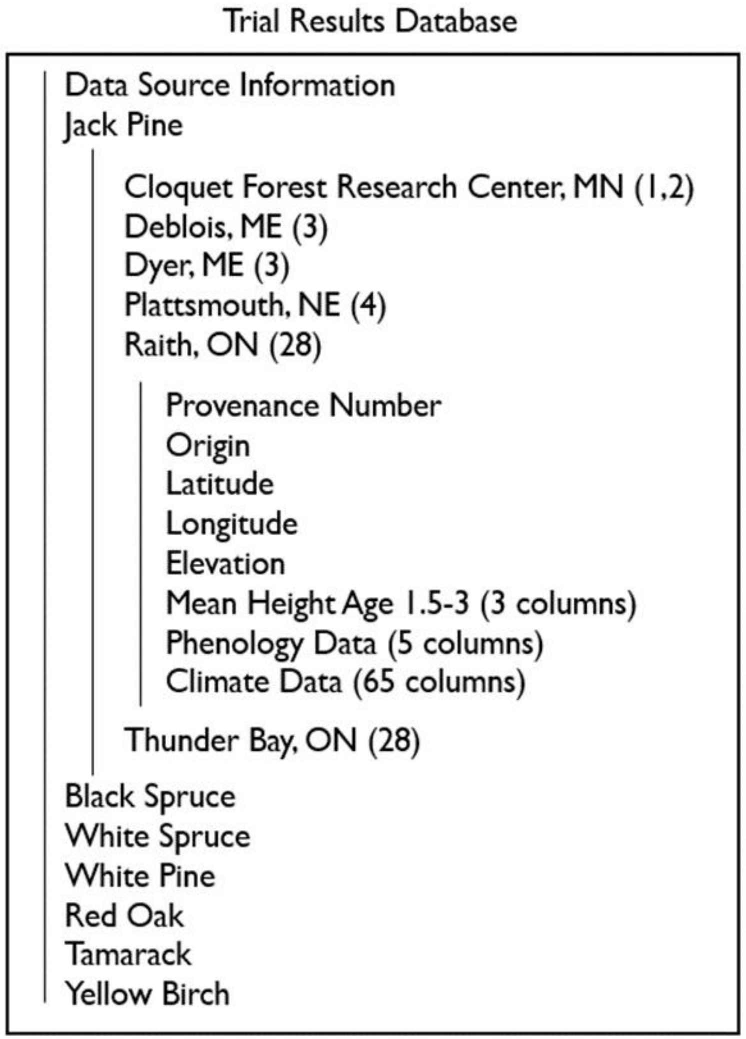
DATABASE 1: DATABASE ORGANIZATION

How is it organized?

- Organized by species, then by test site
- The number beside the test site indicates the data source(s)
- Data sources can be found under "Data Source Information"

What do you need to know?

- No data is always denoted by -9999
- All data is in standardized units
  - Location: decimal degrees
  - Elevation/height: metres
  - Phenology data: days since January 1
  - Climate data:
    - Temperature: °C
    - Precipitation: mm



Trial Results Database

- Data Source Information
- Jack Pine
  - Cloquet Forest Research Center, MN (1,2)
  - Deblois, ME (3)
  - Dyer, ME (3)
  - Plattsmouth, NE (4)
  - Raith, ON (28)
    - Provenance Number
    - Origin
    - Latitude
    - Longitude
    - Elevation
    - Mean Height Age 1.5-3 (3 columns)
    - Phenology Data (5 columns)
    - Climate Data (65 columns)
  - Thunder Bay, ON (28)
- Black Spruce
- White Spruce
- White Pine
- Red Oak
- Tamarack
- Yellow Birch

DATABASE 1: QUERYING THE DATABASE

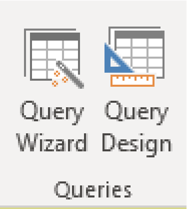
Problem: Find jack pine seed sources that have an origin climate similar to the conditions at a tree nursery:

- Mean annual precipitation (723 mm)
- Mean annual temperature (0.45°C)
- Elevation (480m)

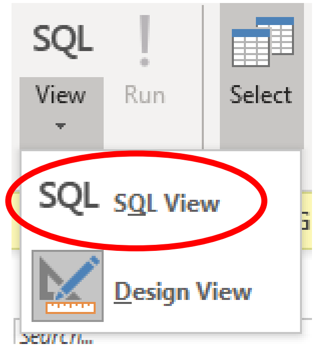
DATABASE 1: QUERYING THE DATABASE

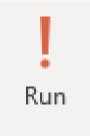
Solution: Select the provenances at the Ontario test sites for jack pine that have values for the climate variables within 10% of the values for the nursery site.

DATABASE 1: QUERYING THE DATABASE



```
SELECT X.[Provenance ID], X.Origin, cDBL(X.Latitude) AS Lat, cDBL(X.Longitude) AS
Lon

FROM (((

SELECT DISTINCT [Raith, ON (28)].[Provenance ID], [Raith, ON (28)].Origin,[Raith,
ON (28)].Latitude,[Raith, ON (28)].Longitude, [Raith, ON (28)].[Annual
Precipitation (mm)],[Raith, ON (28)].[Annual Mean Temperature (°C)],[Raith, ON
(28)].[Elevation (m)] from [Raith, ON (28)]

UNION

SELECT DISTINCT [Thunder Bay, ON (28)].[Provenance ID], [Thunder Bay, ON
(28)].Origin,[Thunder Bay, ON (28)].Latitude,[Thunder Bay, ON (28)].Longitude,
[Thunder Bay, ON (28)].[Annual Precipitation (mm)],[Thunder Bay, ON (28)].[Annual
Mean Temperature (°C)],[Thunder Bay, ON (28)].[Elevation (m)] from [Thunder Bay, ON
(28)]) AS X


FULL OUTER JOIN [Raith, ON (28)] A ON A.[Provenance ID] = X.[Provenance ID] )

FULL OUTER JOIN [Thunder Bay, ON (28)] B ON B.[Provenance ID] = X.[Provenance ID])


WHERE X.[Annual Precipitation (mm)] > (723-723*0.1) AND X.[Annual Precipitation
(mm)] <(723+723*0.1)

AND X.[Annual Mean Temperature (°C)] > (0.45-0.45*0.1) AND X.[Annual Mean
Temperature (°C)] <(0.45+0.45*0.1)

AND X.[Elevation (m)] > (480-480*0.1) AND X.[Elevation (m)] <(480+480*0.1);
```



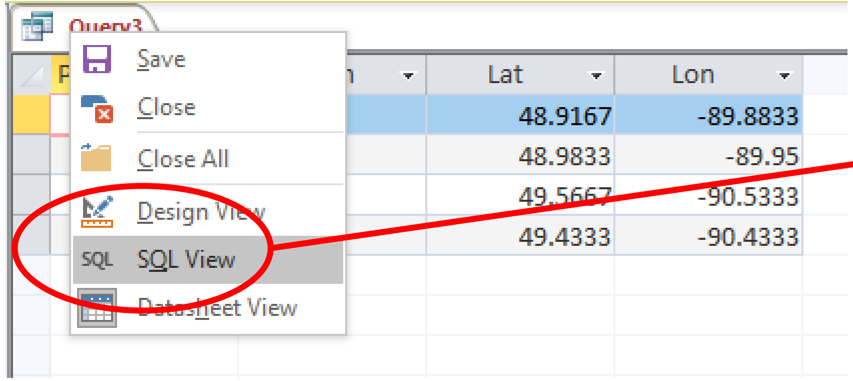


DATABASE 1: QUERYING THE DATABASE

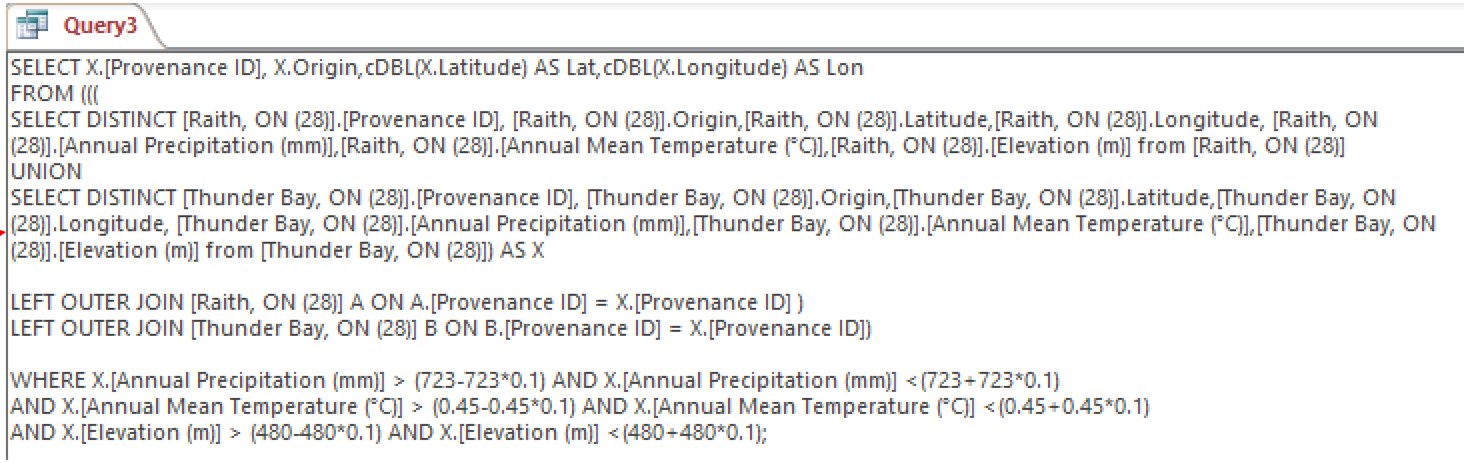
Answer:

Provenance ID	Origin	Latitude	Longitude
41	Ontario	48.92	-89.88
42	Ontario	48.98	-89.95
56	Ontario	49.57	-90.53
57	Ontario	49.43	-90.43

To modify selection criteria:



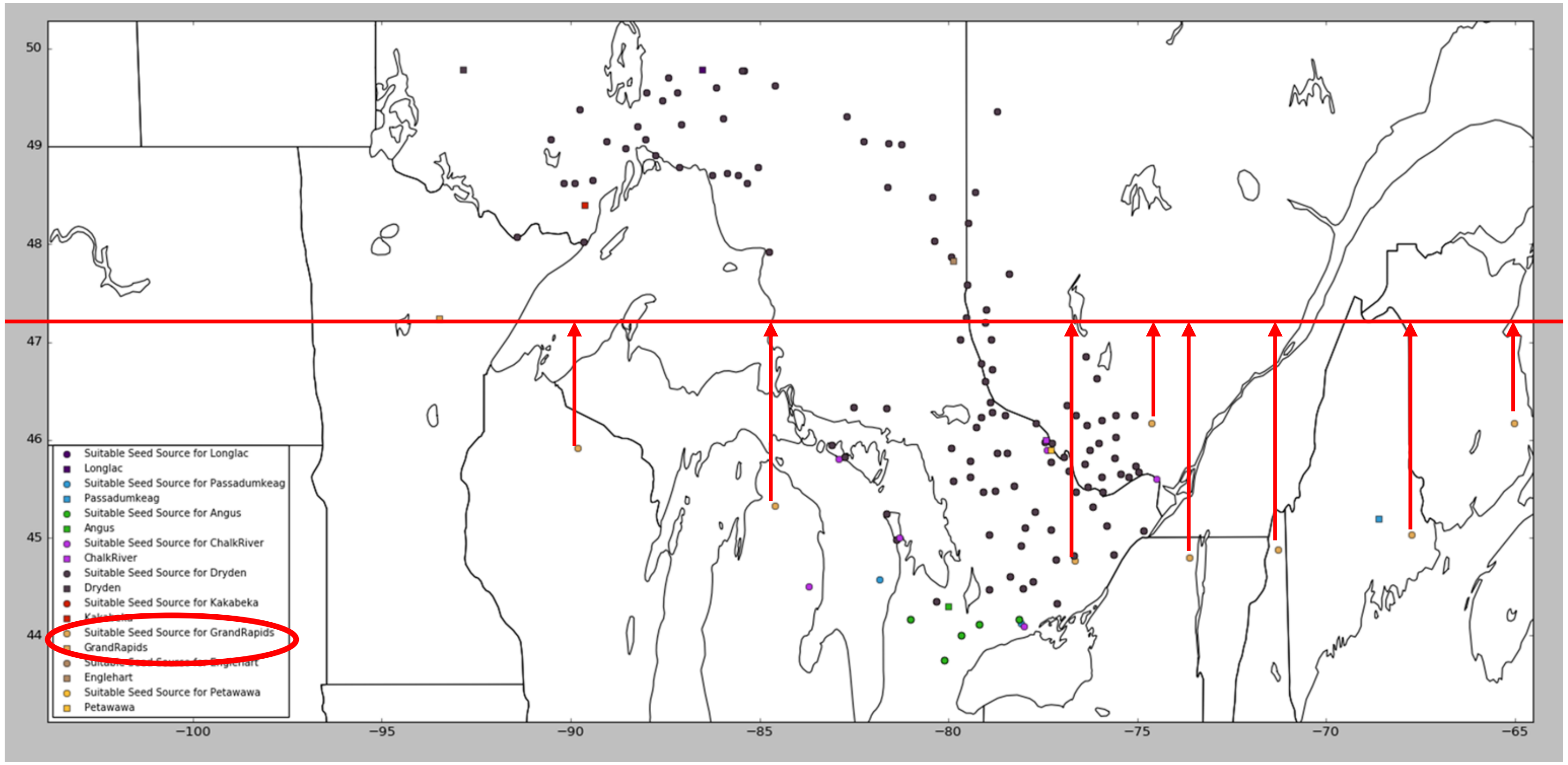
Lat	Lon
48.9167	-89.8833
48.9833	-89.95
49.5667	-90.5333
49.4333	-90.4333

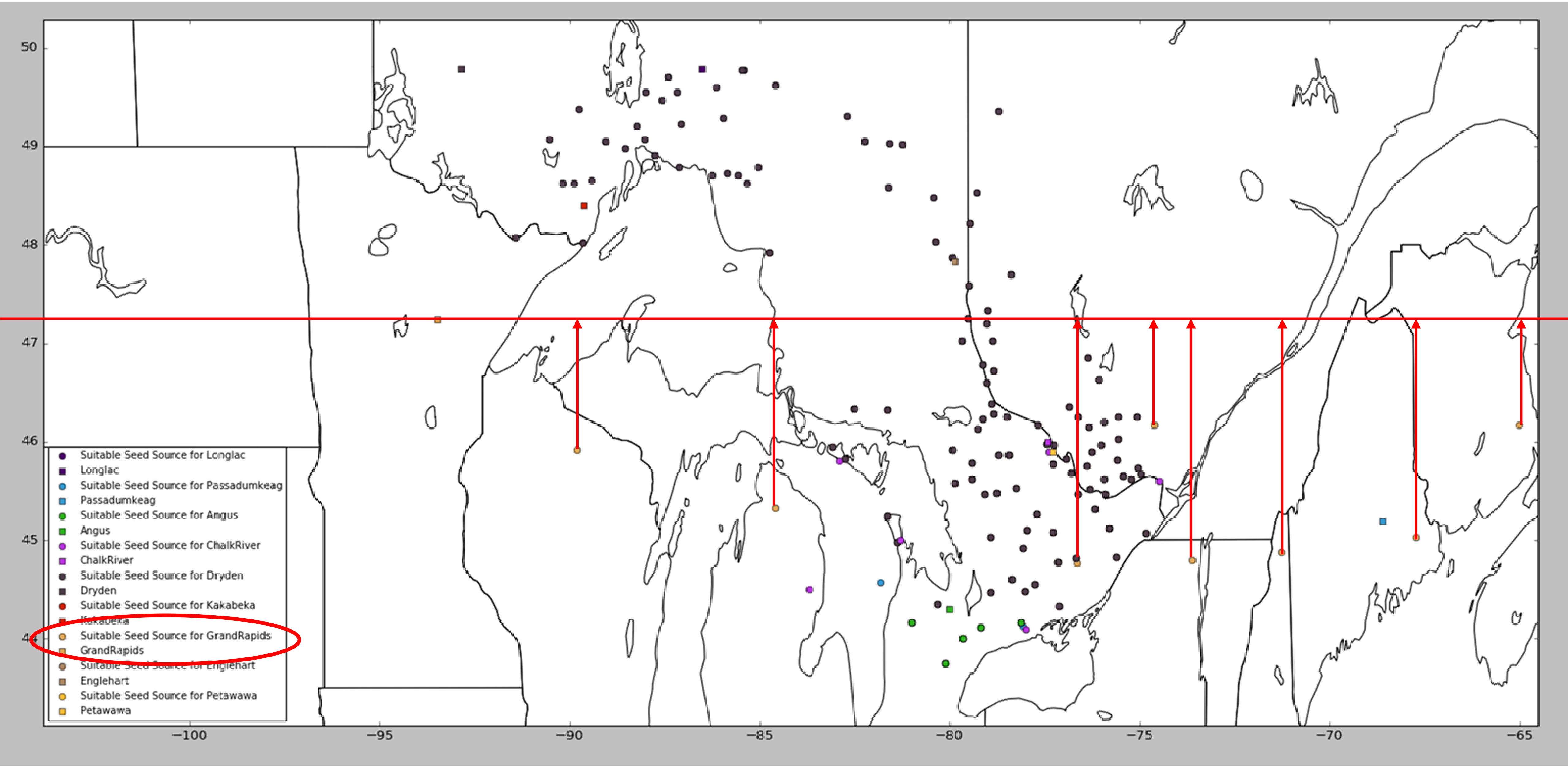


```
SELECT X.[Provenance ID], X.Origin,cDBL(X.Latitude) AS Lat,cDBL(X.Longitude) AS Lon
FROM (((
SELECT DISTINCT [Raith, ON (28)].[Provenance ID], [Raith, ON (28)].Origin,[Raith, ON (28)].Latitude,[Raith, ON (28)].Longitude, [Raith, ON
(28)].[Annual Precipitation (mm)],[Raith, ON (28)].[Annual Mean Temperature (°C)],[Raith, ON (28)].[Elevation (m)] from [Raith, ON (28)]
UNION
SELECT DISTINCT [Thunder Bay, ON (28)].[Provenance ID], [Thunder Bay, ON (28)].Origin,[Thunder Bay, ON (28)].Latitude,[Thunder Bay, ON
(28)].Longitude, [Thunder Bay, ON (28)].[Annual Precipitation (mm)],[Thunder Bay, ON (28)].[Annual Mean Temperature (°C)],[Thunder Bay, ON
(28)].[Elevation (m)] from [Thunder Bay, ON (28)]) AS X

LEFT OUTER JOIN [Raith, ON (28)] A ON A.[Provenance ID] = X.[Provenance ID] )
LEFT OUTER JOIN [Thunder Bay, ON (28)] B ON B.[Provenance ID] = X.[Provenance ID])

WHERE X.[Annual Precipitation (mm)] > (723-723*0.1) AND X.[Annual Precipitation (mm)] <(723+723*0.1)
AND X.[Annual Mean Temperature (°C)] > (0.45-0.45*0.1) AND X.[Annual Mean Temperature (°C)] <(0.45+0.45*0.1)
AND X.[Elevation (m)] > (480-480*0.1) AND X.[Elevation (m)] <(480+480*0.1);
```

DATABASE 1: TRANSFER DISTANCE CALCULATION

Problem: Perform rough calculations of northward successful seed transfer distance for each species using information from every test site included in the database. For simplicity, we will assume all trials are rangewide (otherwise, there are some statistical issues…).


50
49
48
47
46
45
44
−100
−95
−90
−85
−80
−75
−70
−65
● Suitable Seed Source for Longlac
■ Longlac
● Suitable Seed Source for Passadumkeag
■ Passadumkeag
● Suitable Seed Source for Angus
■ Angus
● Suitable Seed Source for ChalkRiver
■ ChalkRiver
● Suitable Seed Source for Dryden
■ Dryden
● Suitable Seed Source for Kakabeka
■ Kakabeka
● Suitable Seed Source for GrandRapids
■ GrandRapids
● Suitable Seed Source for Englehart
■ Englehart
● Suitable Seed Source for Petawawa
■ Petawawa


50
49
48
47
46
45
44
−100
−95
−90
−85
−80
−75
−70
−65
● Suitable Seed Source for Longlac
■ Longlac
● Suitable Seed Source for Passadumkeag
■ Passadumkeag
● Suitable Seed Source for Angus
■ Angus
● Suitable Seed Source for ChalkRiver
■ ChalkRiver
● Suitable Seed Source for Dryden
■ Dryden
● Suitable Seed Source for Kakabeka
■ Kakabeka
● Suitable Seed Source for GrandRapids
■ GrandRapids
● Suitable Seed Source for Englehart
■ Englehart
● Suitable Seed Source for Petawawa
■ Petawawa

DATABASE 1: TRANSFER DISTANCE CALCULATION

Solution: For each test site:

(A) Determine the provenances that performed better than average for each measured metric (height or survival at different ages)

(B) Determine the provenances that are south of the test site

(C) Cross-reference the successful and southern provenances

(D) Calculate the northward distance to the test site for each selected provenance

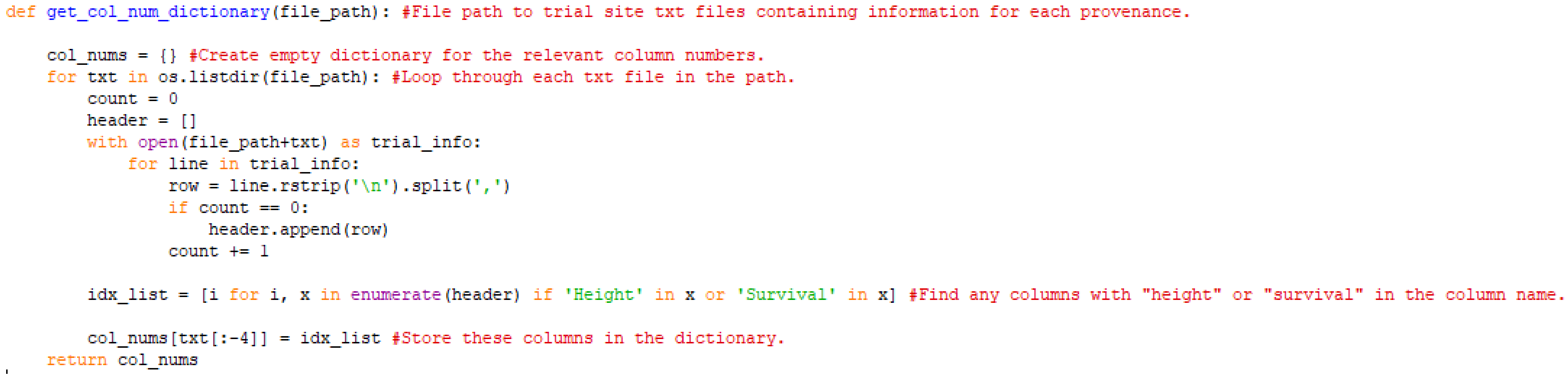
(E) Take the mean of these distances

… Extremely tedious to perform these calculations in SQL

DATABASE 1: TRANSFER DISTANCE CALCULATION

Step A: Select the Suitable Provenances

- Each study included in the database measured different variables
- Find the columns for each test site that contain relevant data

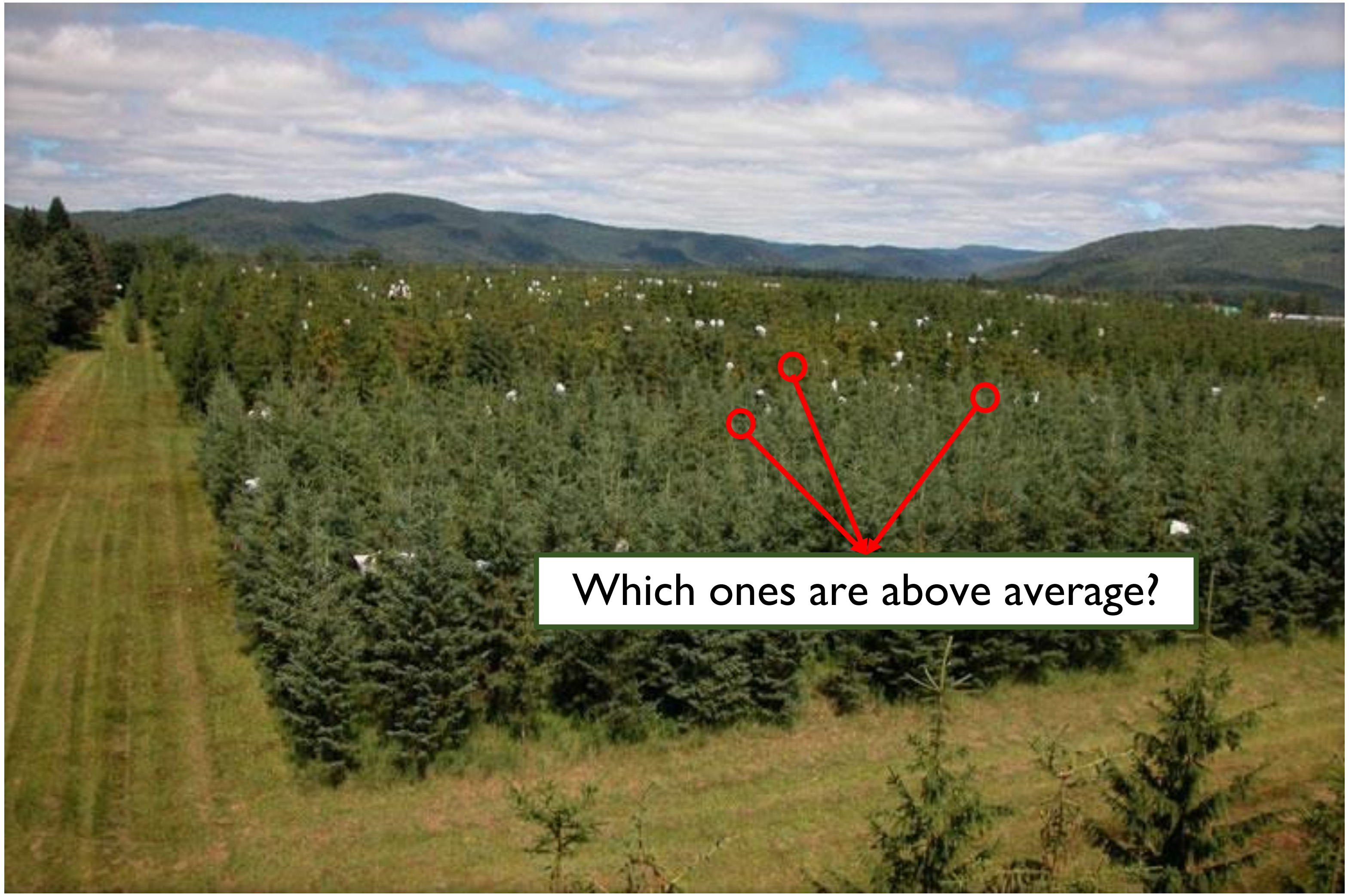


```
def get_col_num_dictionary(file_path): #File path to trial site txt files containing information for each provenance.

    col_nums = {} #Create empty dictionary for the relevant column numbers.
    for txt in os.listdir(file_path): #Loop through each txt file in the path.
        count = 0
        header = []
        with open(file_path+txt) as trial_info:
            for line in trial_info:
                row = line.rstrip('\n').split(',')
                if count == 0:
                    header.append(row)
                count += 1

        idx_list = [i for i, x in enumerate(header) if 'Height' in x or 'Survival' in x] #Find any columns with "height" or "survival" in the column name.

        col_nums[txt[:-4]] = idx_list #Store these columns in the dictionary.
    return col_nums
```

DATABASE 1: TRANSFER DISTANCE CALCULATION

Step A: Select the Suitable Provenances

1. Calculate the overall mean of each column
2. Loop through each column index (for each test site) and calculate the % Mean for each provenance
3. Select out the provenances that have a % Mean >= 100 for every relevant column
4. Return the provenance IDs of these provenances

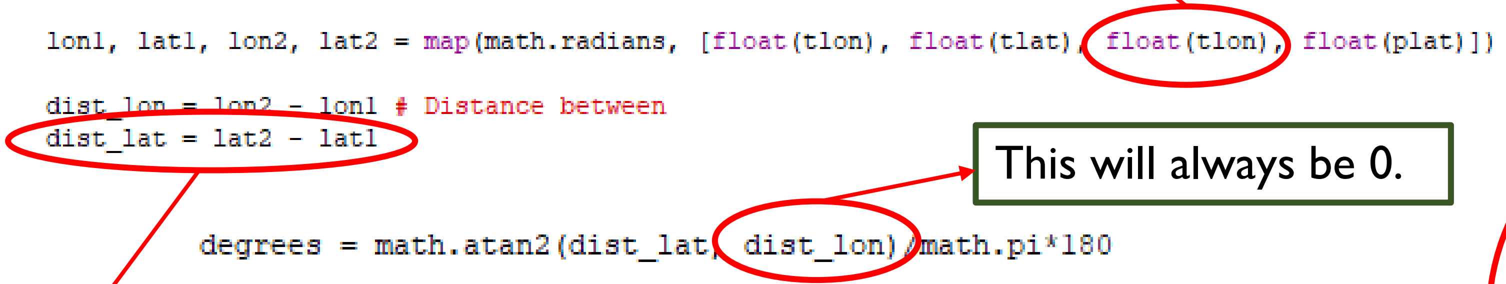


Which ones are above average?

DATABASE 1: TRANSFER DISTANCE CALCULATION

Step B: Determine which provenance origins are south of each test site

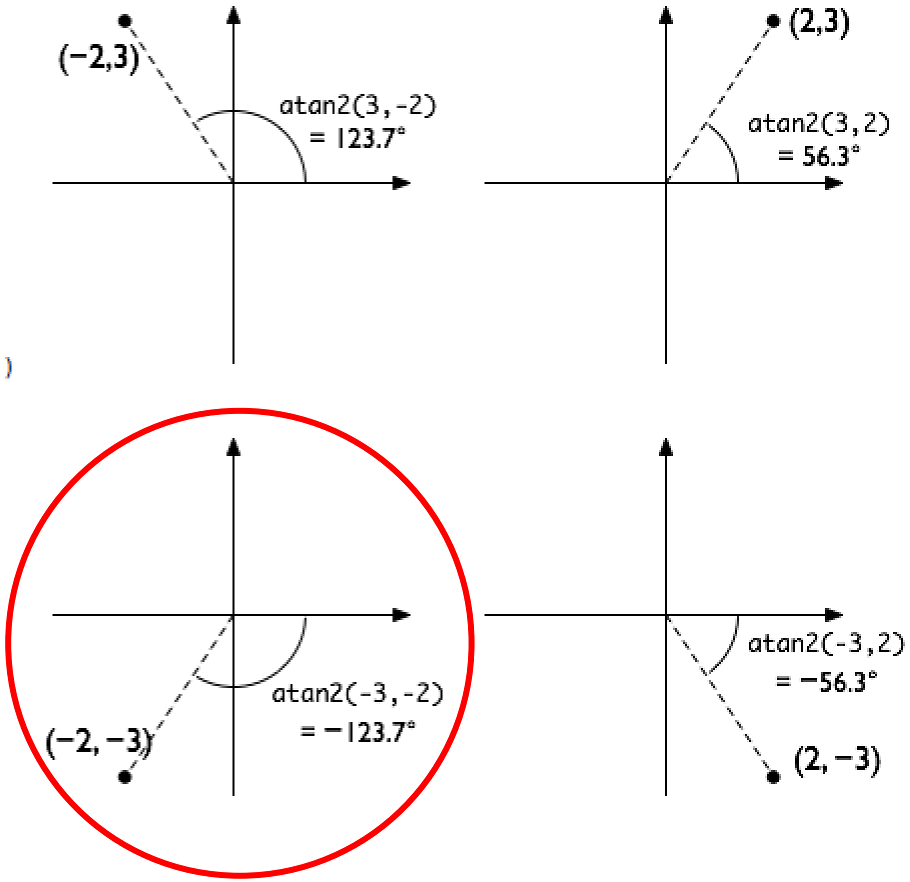
We can assume that the provenance and the test site have the same longitude because we are only interested in the northward distance.



```
lon1, lat1, lon2, lat2 = map(math.radians, [float(tlon), float(tlat), float(tlon), float(plat)])

dist_lon = lon2 - lon1 # Distance between
dist_lat = lat2 - lat1

        degrees = math.atan2(dist_lat, dist_lon)/math.pi*180
```

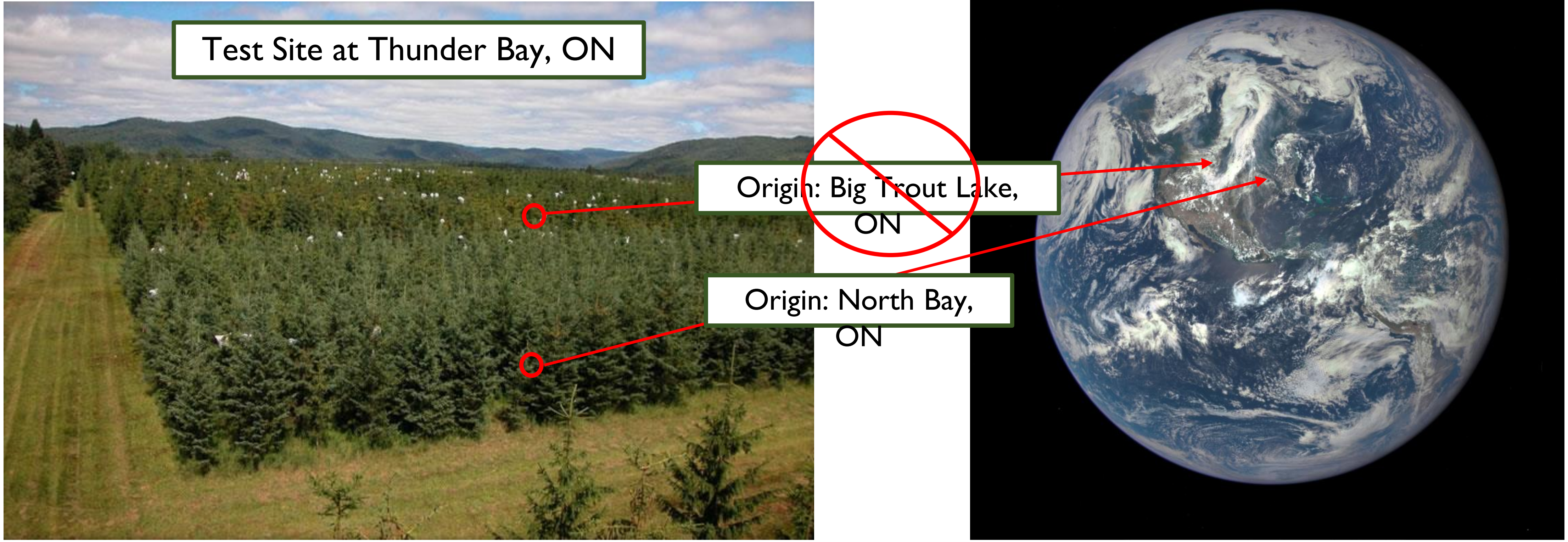
This will always be 0.

If the provenance (lat2) is south of the test site (lat1), this will be NEGATIVE.



DATABASE 1: TRANSFER DISTANCE CALCULATION

Step C: Cross-reference



Test Site at Thunder Bay, ON

~~Origin: Big Trout Lake, ON~~

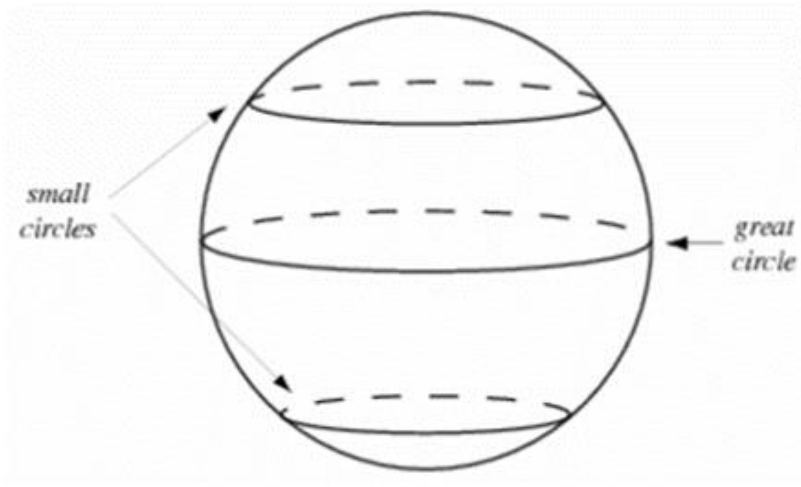
Origin: North Bay, ON

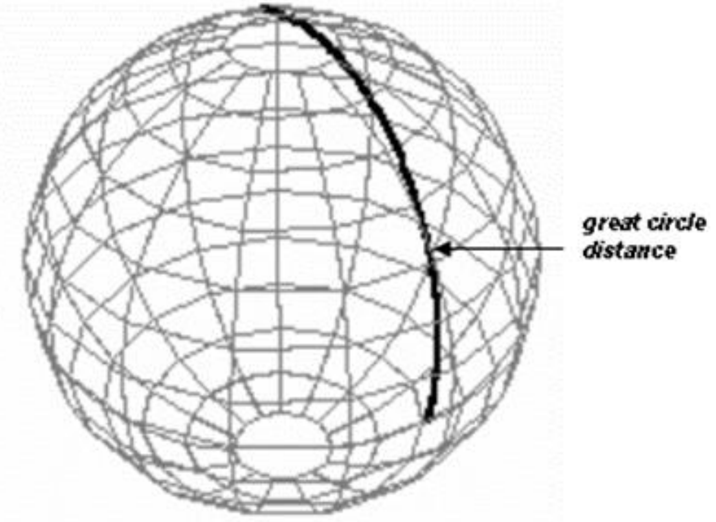
Even though the Big Trout Lake provenance performed better than average, it will be excluded because it is north of the test site.

DATABASE 1: TRANSFER DISTANCE CALCULATION

Step D: Calculate the northward distance

Use the Haversine Formula to calculate the northward distance between the provenances to their corresponding test sites

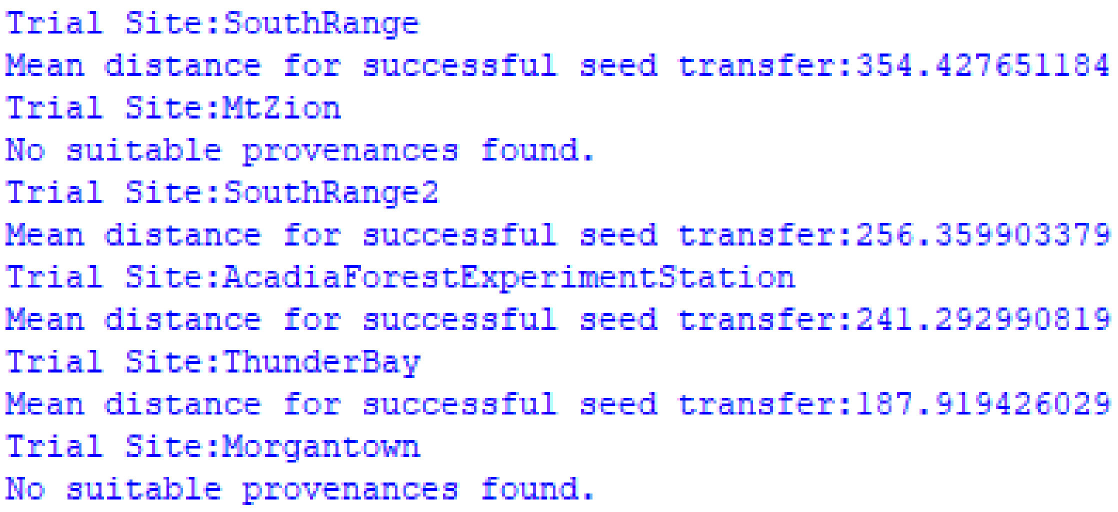




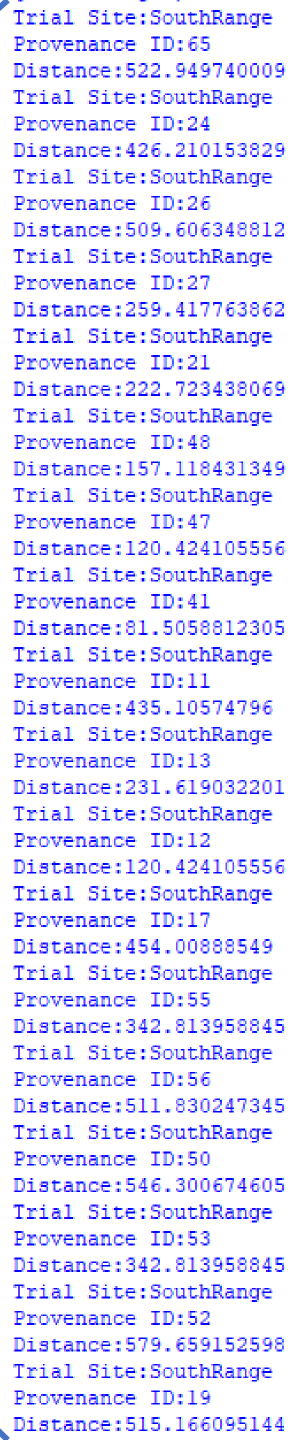
$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

DATABASE 1: TRANSFER DISTANCE CALCULATION

Step E: Take the mean of these distances



```
Trial Site:SouthRange
Mean distance for successful seed transfer:354.427651184
Trial Site:MtZion
No suitable provenances found.
Trial Site:SouthRange2
Mean distance for successful seed transfer:256.359903379
Trial Site:AcadiaForestExperimentStation
Mean distance for successful seed transfer:241.292990819
Trial Site:ThunderBay
Mean distance for successful seed transfer:187.919426029
Trial Site:Morgantown
No suitable provenances found.
```



```
Trial Site:SouthRange
Provenance ID:65
Distance:522.949740009
Trial Site:SouthRange
Provenance ID:24
Distance:426.210153829
Trial Site:SouthRange
Provenance ID:26
Distance:509.606348812
Trial Site:SouthRange
Provenance ID:27
Distance:259.417763862
Trial Site:SouthRange
Provenance ID:21
Distance:222.723438069
Trial Site:SouthRange
Provenance ID:48
Distance:157.118431349
Trial Site:SouthRange
Provenance ID:47
Distance:120.424105556
Trial Site:SouthRange
Provenance ID:41
Distance:81.5058812305
Trial Site:SouthRange
Provenance ID:11
Distance:435.10574796
Trial Site:SouthRange
Provenance ID:13
Distance:231.619032201
Trial Site:SouthRange
Provenance ID:12
Distance:120.424105556
Trial Site:SouthRange
Provenance ID:17
Distance:454.00888549
Trial Site:SouthRange
Provenance ID:55
Distance:342.813958845
Trial Site:SouthRange
Provenance ID:56
Distance:511.830247345
Trial Site:SouthRange
Provenance ID:50
Distance:546.300674605
Trial Site:SouthRange
Provenance ID:53
Distance:342.813958845
Trial Site:SouthRange
Provenance ID:52
Distance:579.659152598
Trial Site:SouthRange
Provenance ID:19
Distance:515.166095144
```

DATABASE 2: BACKGROUND

What does it contain?

- Mean, standard deviation, minimum, maximum values for the period between the test site establishment year and the measurement year(s)
- Multiple tables were used if there was more than one measurement year, such as 1978-1988, 1978-1992, etc.

Why do we want this information in a database?

- Helpful for:
  - Calculating future successful species-specific seed transfer distances
  - Improving nursery and plantation productivity

DATABASE 2: BACKGROUND

Species	Number of Test Sites	Earliest Establishment Date	Latest Establishment Date	Earliest Measurement Date	Latest Measurement Date
Jack Pine	7	1942	1988	1950	1990
Black Spruce	8	1973	1977	1979	1985
White Spruce	9	1962	2003	1966	2004
White Pine	26	1958	1965	1958	1979
Red Oak	7	1962	1964	1973	1975
Tamarack	11	1962	1986	1963	1992
Yellow Birch	1	1972	1972	1974	1976

DATABASE 2: BACKGROUND

How were the climate variables generated?

- Mean, standard deviation, minimum, maximum calculated for establishment – measurement period
- Estimates generated for each test site
- Values calculated & formatted automatically using a Python script

1 – Obtain the value for each climate variable for each year between establishment & measurement for the test site

2 – For each variable, calculate mean & standard deviation, and find the minimum & maximum values

3 – Print to a txt file in the correct format

4 – Upload to the database

DATABASE 2: DATABASE ORGANIZATION

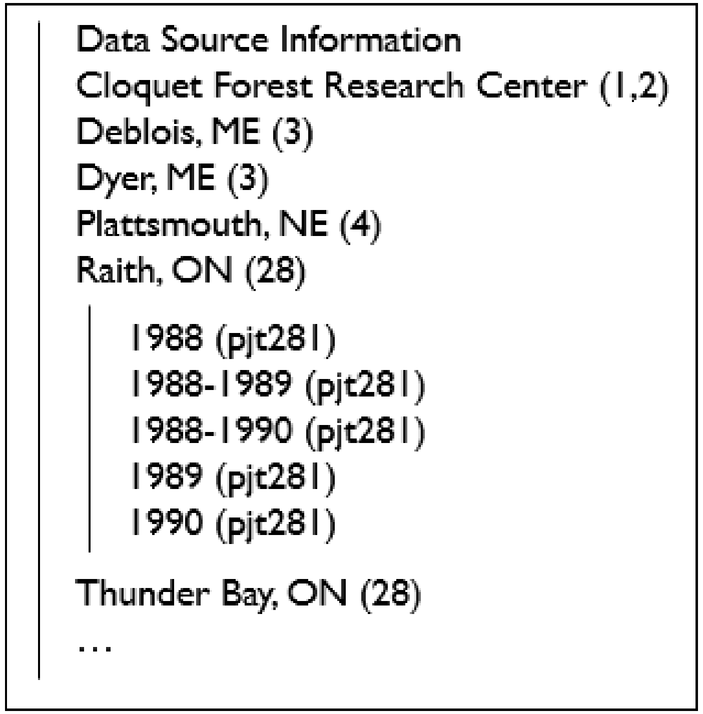
How is it organized?

- Organized by test site, then by year(s)
- The code beside the year(s) indicates the test site (to avoid multiple tables with the same name □ causes problems when constructing queries)

What do you need to know?

- Additional test site information, such as establishment and measurement year(s), can be found under "Data Source Information" in the table "Guide to Test Sites"

Test Site Locations Database



Data Source Information
Cloquet Forest Research Center (1,2)
Deblois, ME (3)
Dyer, ME (3)
Plattsmouth, NE (4)
Raith, ON (28)

- 1988 (pjt281)
- 1988-1989 (pjt281)
- 1988-1990 (pjt281)
- 1989 (pjt281)
- 1990 (pjt281)

Thunder Bay, ON (28)
…

DATABASE 2: FUTURE CLIMATE

Problem: Finding where provenances might grow successfully in 2050 and visualize these areas. We have the following information:

(A) Climate data at the test sites in the establishment – measurement years

(B) Climate projection raster for Ontario in 2050

(C) Interpolated long-term climate averages for Ontario & surroundings

(D) Tree provenance height & survival data for a variety of test sites

DATABASE 2: FUTURE CLIMATE

Solution: For each test site:

(A) Obtain the mean MAP, MAT, and JMMT for the period between test site establishment and LAST measurement year

(B) Find the best-performing provenances

(C) Obtain the1961-1990 MAP, MAT, and JMMT at the provenance origin

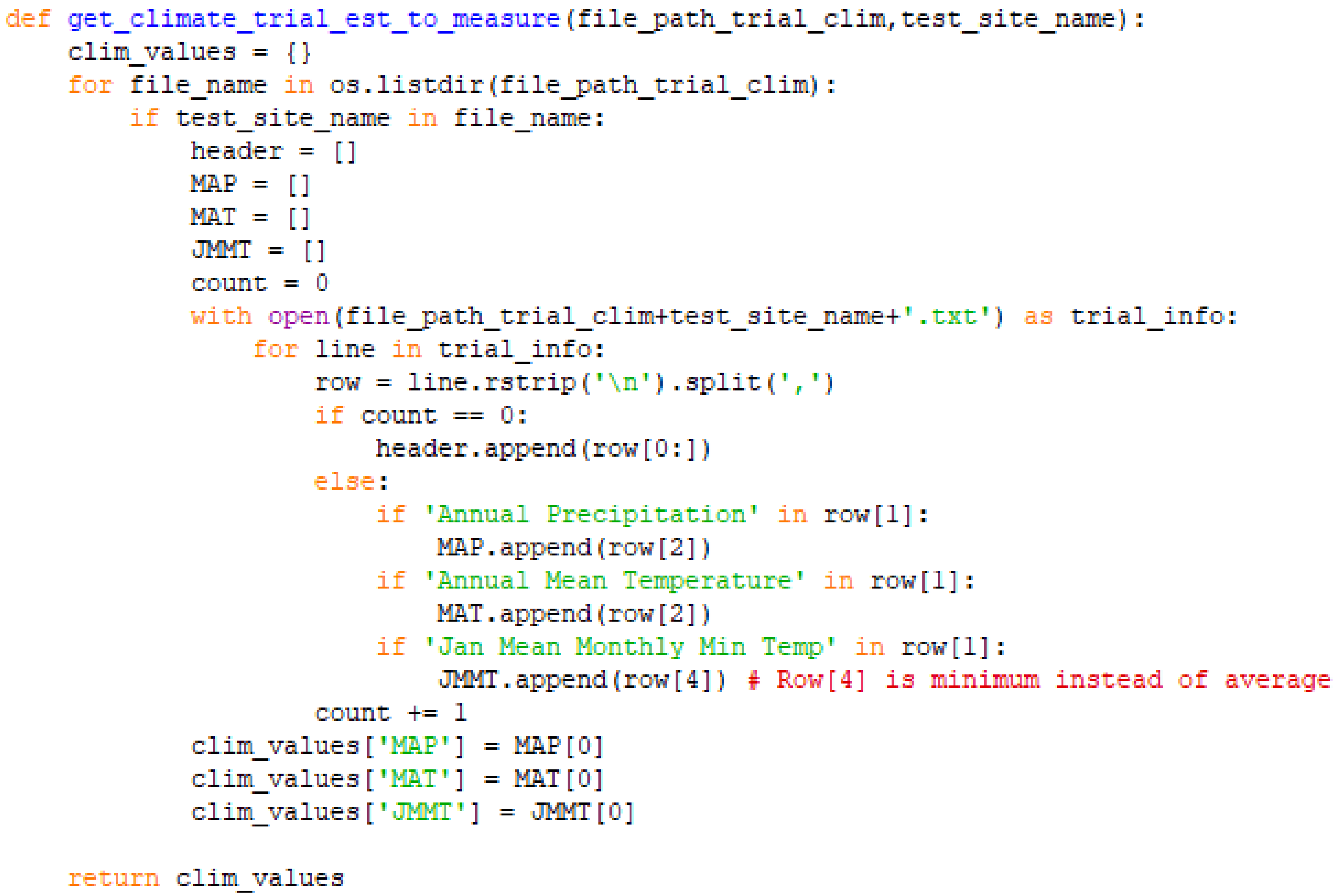
(D) Select the areas where the long term climate is similar to the climate at the provenance origin

(E) Select the area in Ontario where the 2050 climate is similar to the climate at the test site

(F) Create a map of the areas

DATABASE 2: FUTURE CLIMATE

(A) Obtain the mean MAP, MAT, and JMMT for the period between test site establishment and last measurement year

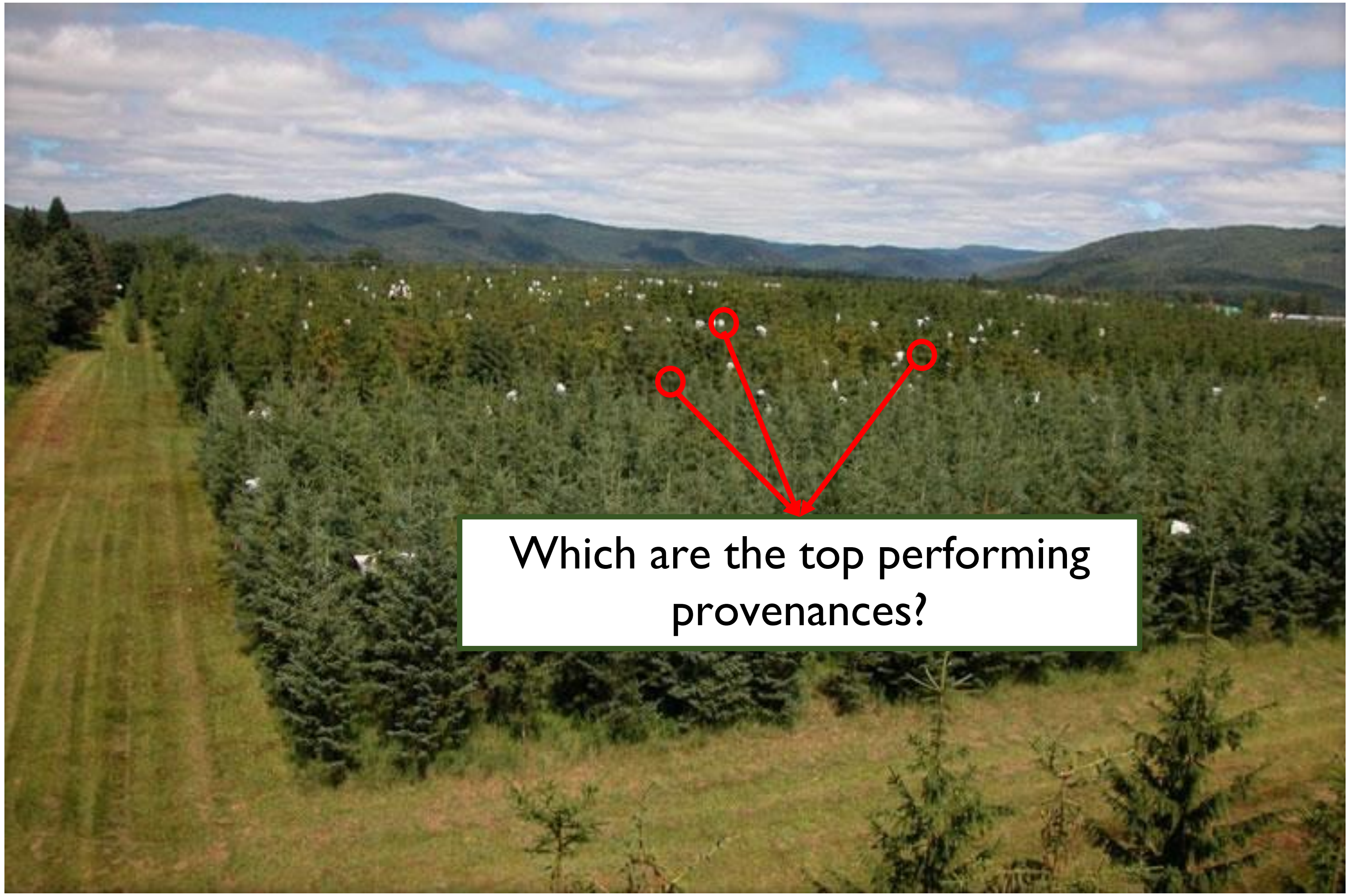


```
def get_climate_trial_est_to_measure(file_path_trial_clim,test_site_name):
    clim_values = {}
    for file_name in os.listdir(file_path_trial_clim):
        if test_site_name in file_name:
            header = []
            MAP = []
            MAT = []
            JMMT = []
            count = 0
            with open(file_path_trial_clim+test_site_name+'.txt') as trial_info:
                for line in trial_info:
                    row = line.rstrip('\n').split(',')
                    if count == 0:
                        header.append(row[0:])
                    else:
                        if 'Annual Precipitation' in row[1]:
                            MAP.append(row[2])
                        if 'Annual Mean Temperature' in row[1]:
                            MAT.append(row[2])
                        if 'Jan Mean Monthly Min Temp' in row[1]:
                            JMMT.append(row[4]) # Row[4] is minimum instead of average
                    count += 1
            clim_values['MAP'] = MAP[0]
            clim_values['MAT'] = MAT[0]
            clim_values['JMMT'] = JMMT[0]

    return clim_values
```

DATABASE 2: FUTURE CLIMATE

Step B: Select the Suitable Provenances

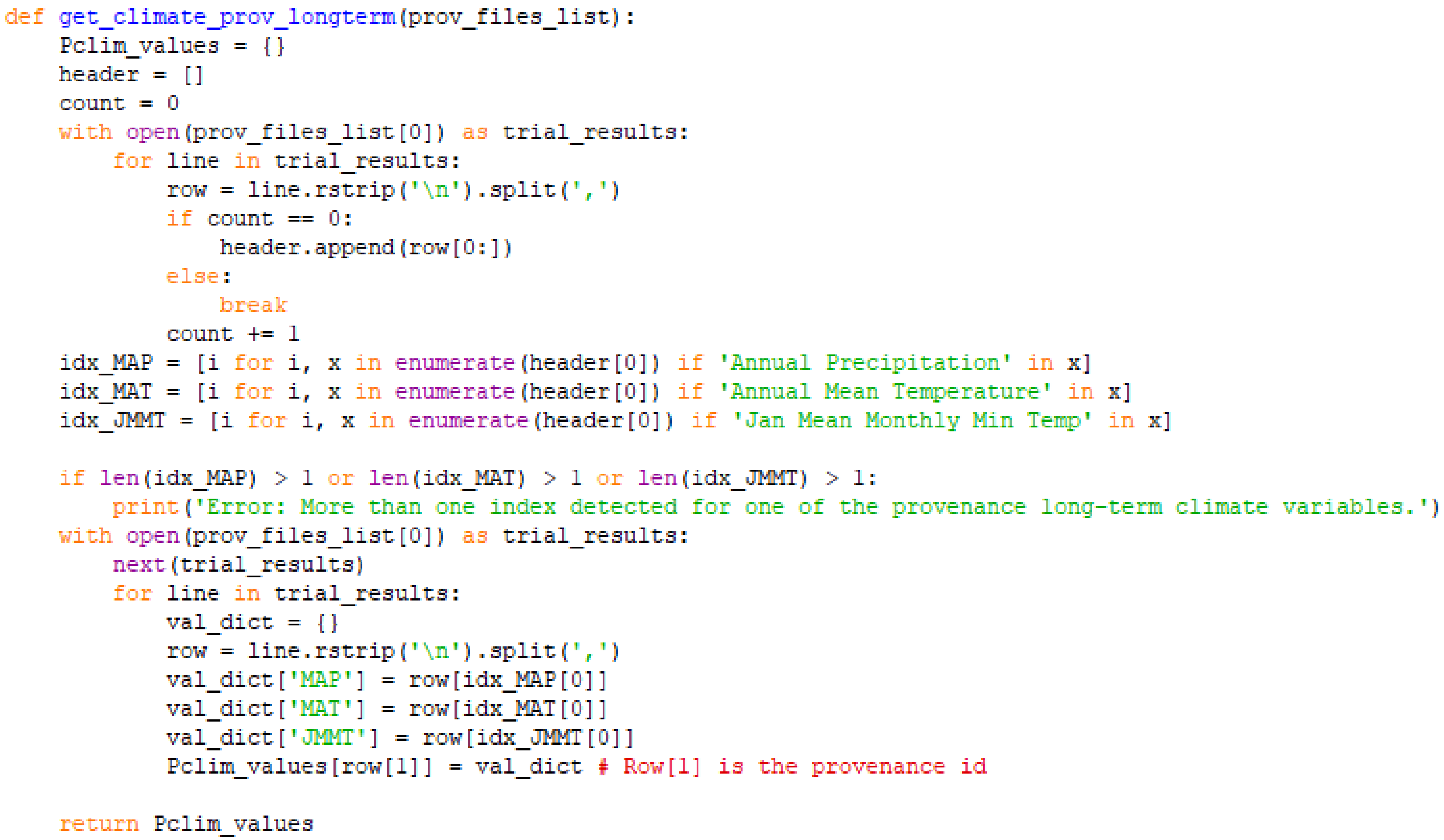
1. Calculate the overall mean of each column
2. Loop through each column index (for each test site) and calculate the % Mean for each provenance
3. Select out the top *X* provenances for each column
4. Cross reference the top provenances
5. Return the common provenance IDs



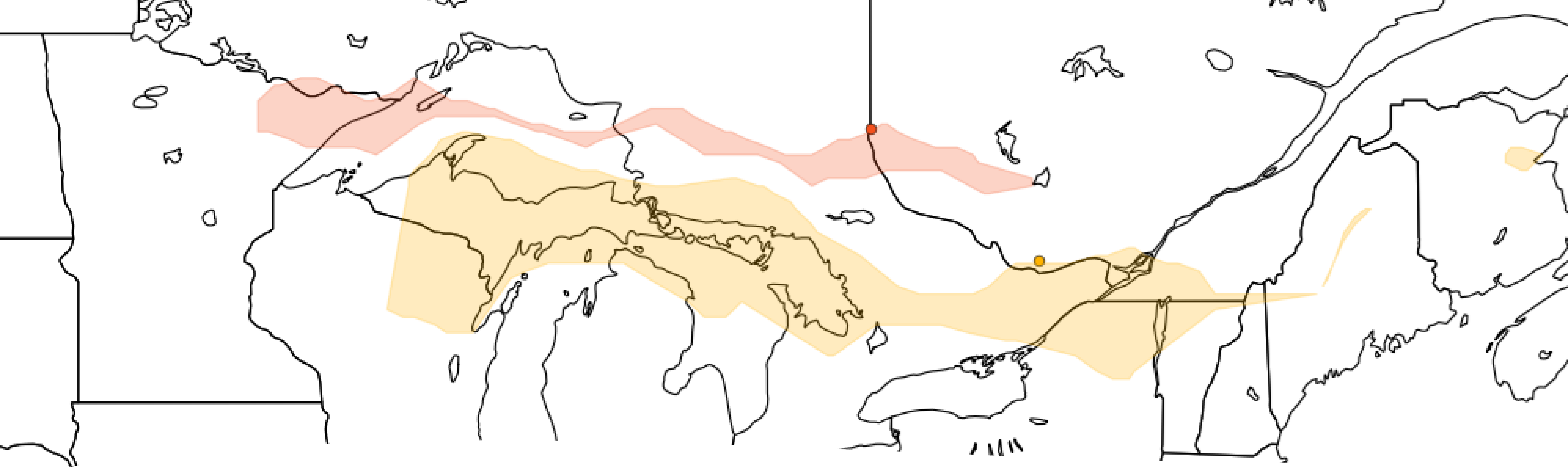
Which are the top performing provenances?

DATABASE 2: FUTURE CLIMATE

(C) Obtain the1961-1990 MAP, MAT, and JMMT at the provenance origin



```
def get_climate_prov_longterm(prov_files_list):
    Pclim_values = {}
    header = []
    count = 0
    with open(prov_files_list[0]) as trial_results:
        for line in trial_results:
            row = line.rstrip('\n').split(',')
            if count == 0:
                header.append(row[0:])
            else:
                break
            count += 1
    idx_MAP = [i for i, x in enumerate(header[0]) if 'Annual Precipitation' in x]
    idx_MAT = [i for i, x in enumerate(header[0]) if 'Annual Mean Temperature' in x]
    idx_JMMT = [i for i, x in enumerate(header[0]) if 'Jan Mean Monthly Min Temp' in x]

    if len(idx_MAP) > 1 or len(idx_MAT) > 1 or len(idx_JMMT) > 1:
        print('Error: More than one index detected for one of the provenance long-term climate variables.')
    with open(prov_files_list[0]) as trial_results:
        next(trial_results)
        for line in trial_results:
            val_dict = {}
            row = line.rstrip('\n').split(',')
            val_dict['MAP'] = row[idx_MAP[0]]
            val_dict['MAT'] = row[idx_MAT[0]]
            val_dict['JMMT'] = row[idx_JMMT[0]]
            Pclim_values[row[1]] = val_dict # Row[1] is the provenance id

    return Pclim_values
```



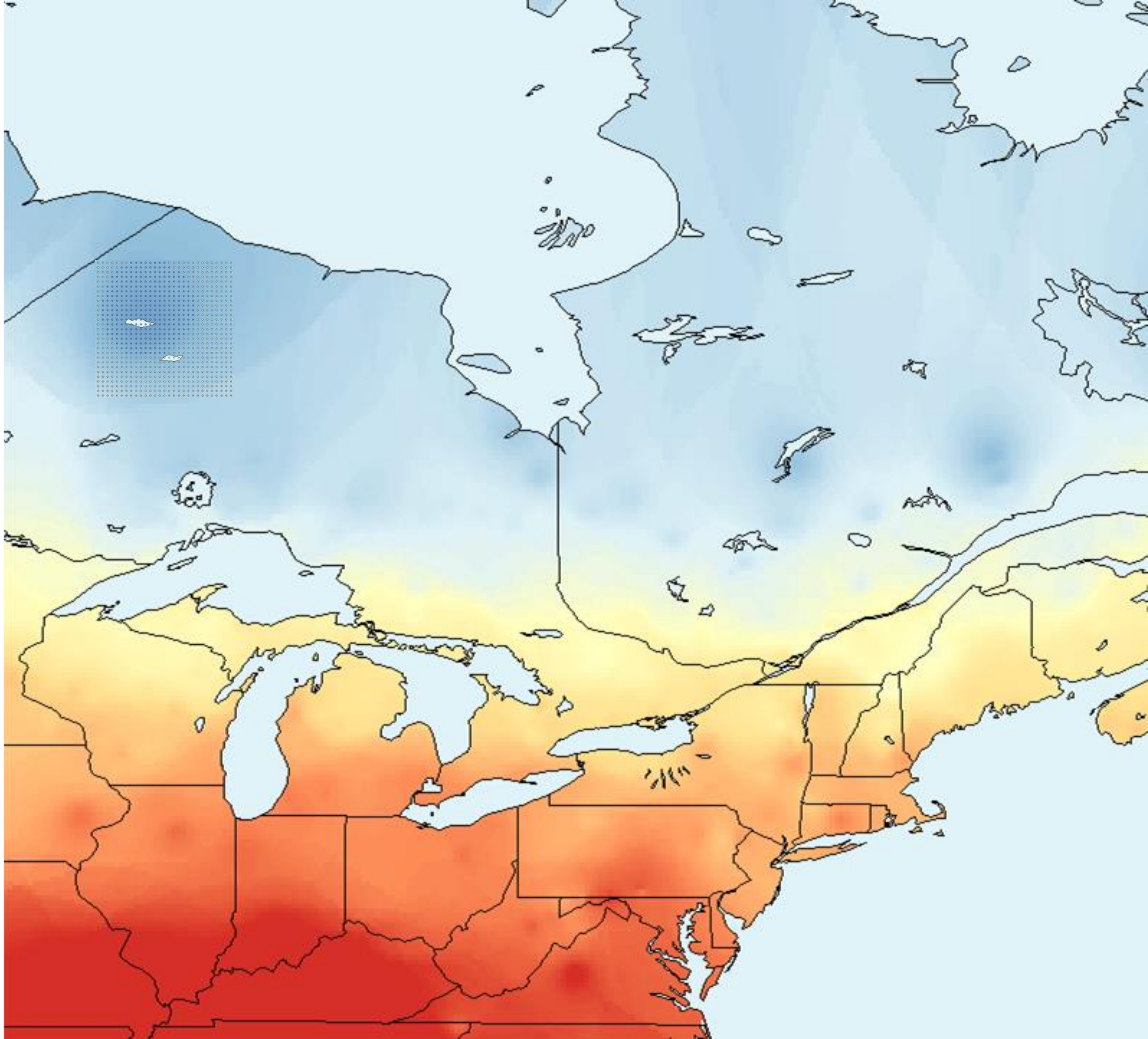
DATABASE 2: FUTURE CLIMATE

(D) Select the areas where the long term climate (1961-1990) is similar to the climate at the provenance origin

DATABASE 2: FUTURE CLIMATE

(D) Select the areas where the long term climate (1961-1990) is similar to the climate at the provenance origin

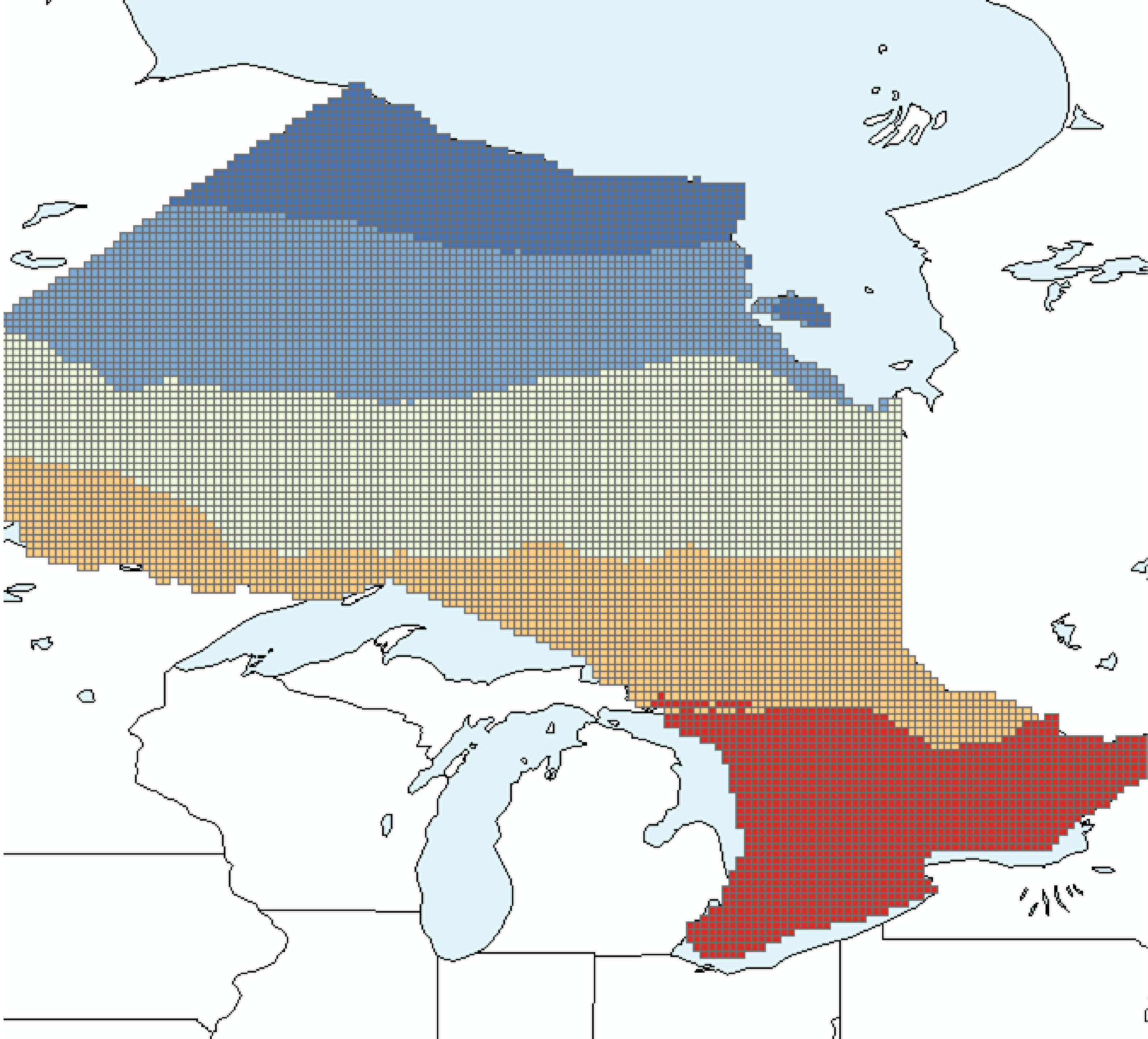
- First, select the areas (cell centroids of the climate surfaces) that are similar (such as having values within 10%) for MAT, MAP, and JMMT individually
- Select where these areas overlap

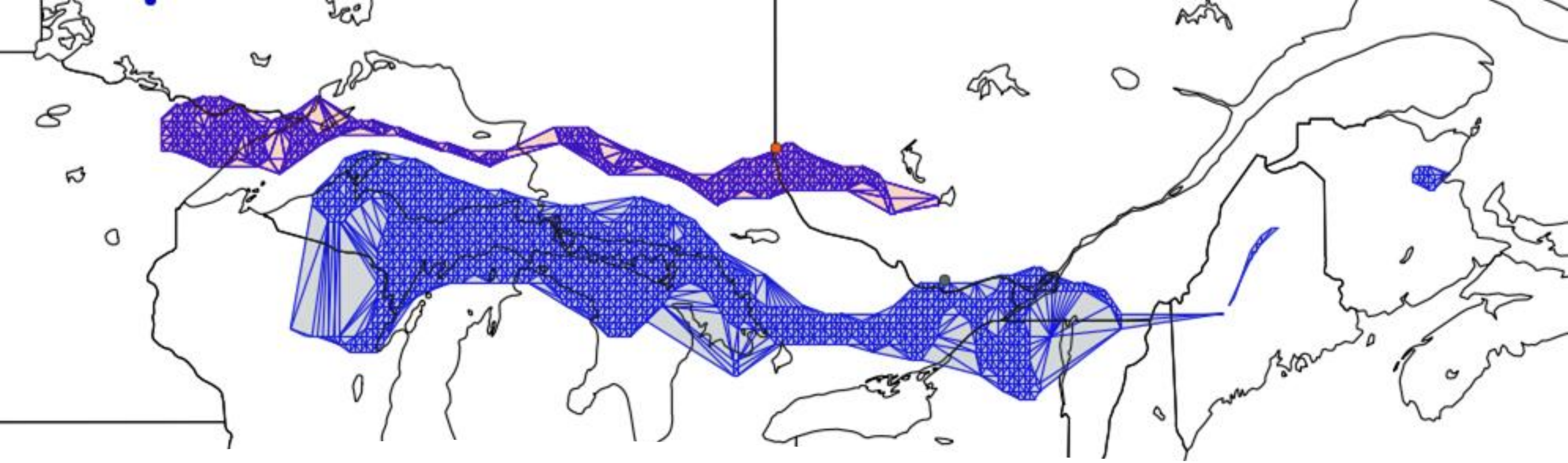


DATABASE 2: FUTURE CLIMATE

(E) Select the area in Ontario where the 2050 climate is similar to the climate at the test site

- First, select the areas (cell centroids of the 2050 climate surfaces) that are similar (such as having values within 10%) for MAT, MAP, and JMMT of the suitable provenance
- Select where these areas overlap





DATABASE 2: FUTURE CLIMATE

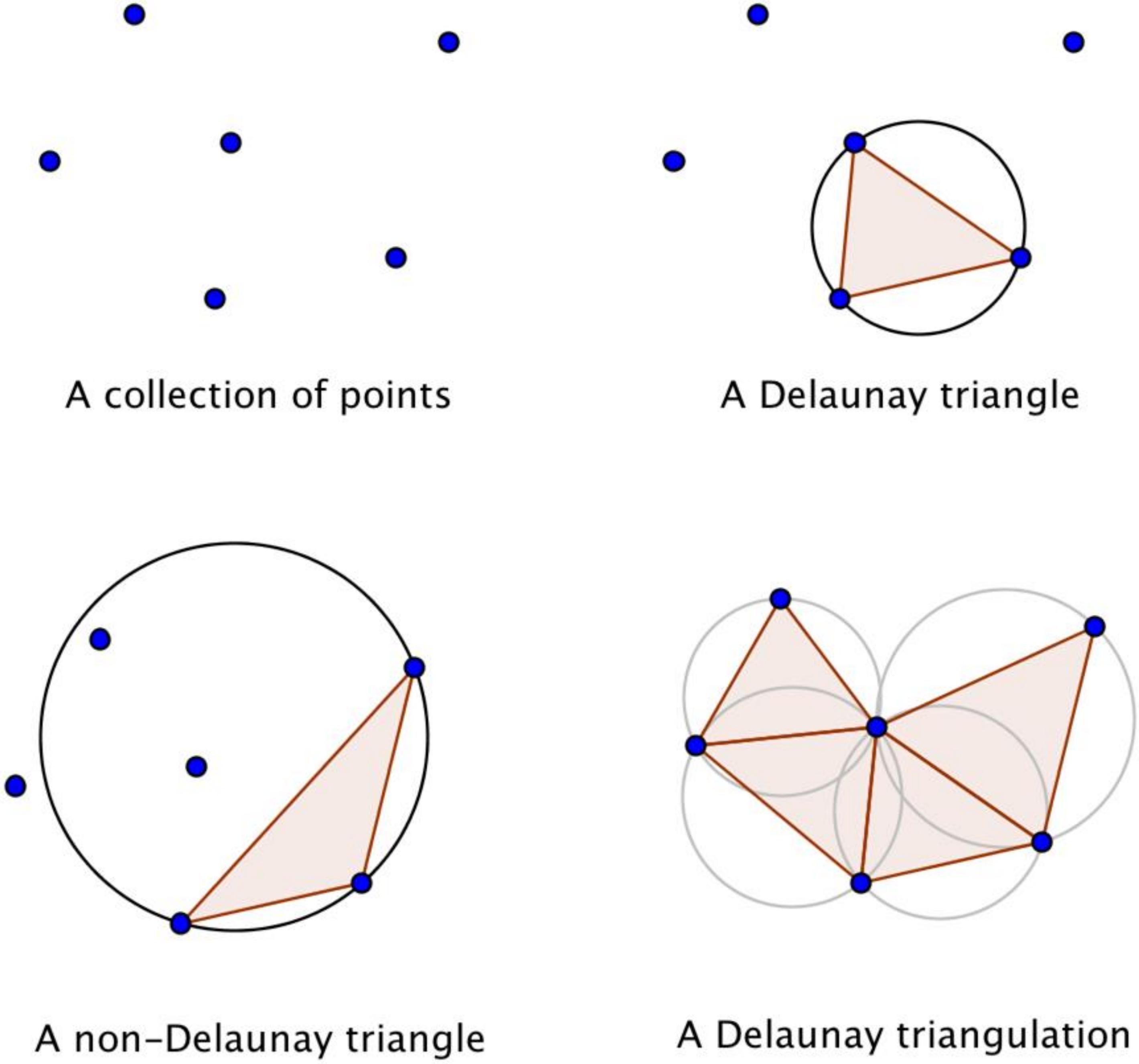
(F) Create a map of the areas

- We have a set of points for two areas:
  - Area matching the 2050 test site climate
  - Areas matching the selected provenance(s) long-term climate
- We need to turn these points into areas
- How? Delaunay Triangulation

DATABASE 2: FUTURE CLIMATE

(F) Create a map of the areas

- What is Delaunay Triangulation? It's a method to cluster points based on creating triangles from sets of points, with the goal of maximizing the smallest angle of the triangle
- Why use it?
  - Just drawing a boundary around all selected points will create too large an area, need to split into clusters
  - Avoid computationally intensive machine learning methods that would require the scikit-learn package



A collection of points

A Delaunay triangle

A non-Delaunay triangle

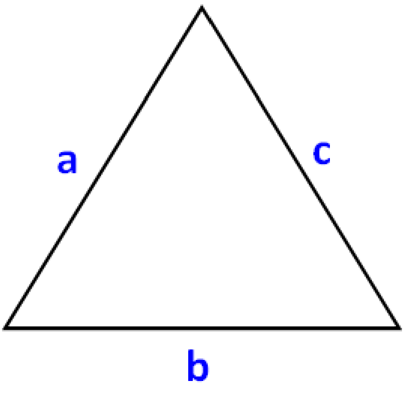
A Delaunay triangulation

DATABASE 2: FUTURE CLIMATE

(F) Create a map of the areas

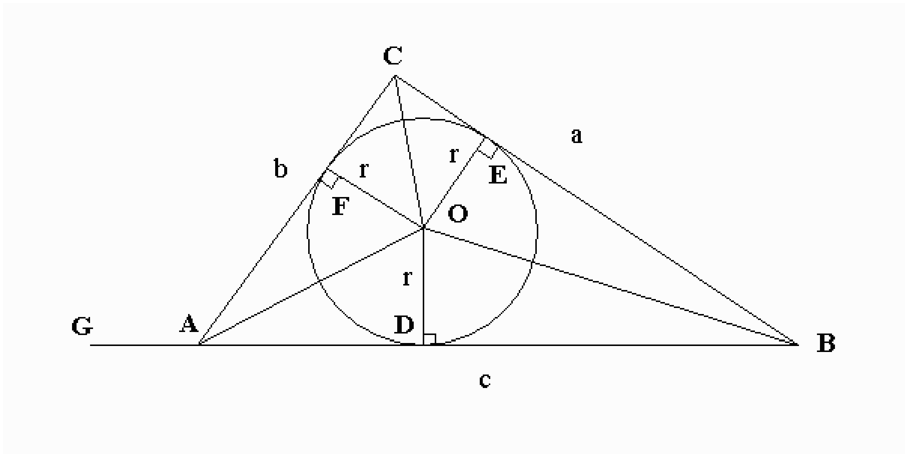
How is Delaunay Triangulation implemented?

- The program takes in all the points for the similar area
- It creates an array of the points and sends the array to the Delaunay function from scipy.spatial
- This built-in function triangulates ALL the points for us
- It sends us back an array of the vertices of each triangle that was formed
- The program loops through each set of vertices
- It calculates the length of the sides of each triangle and uses this to obtain the area of the triangle using Heron’s Formula
- This can be used to get the radius of the triangle (r)
- We can use the radius to filter out the triangles with a point that is too far away to be part of the area



$$\text{Area of Triangle} = \sqrt{s\,(s-a)\,(s-b)\,(s-c)}$$

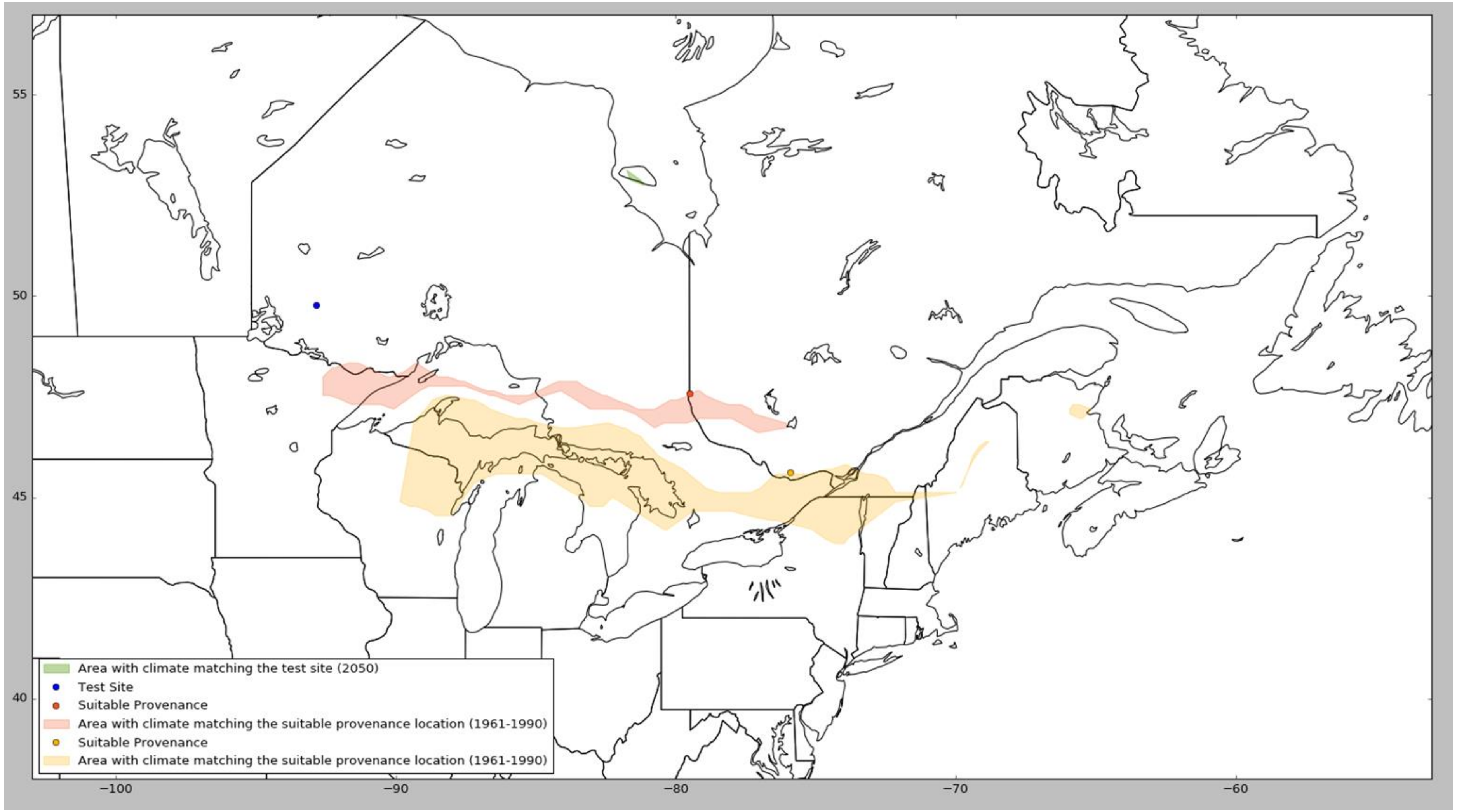
$$s = \frac{a+b+c}{2}$$



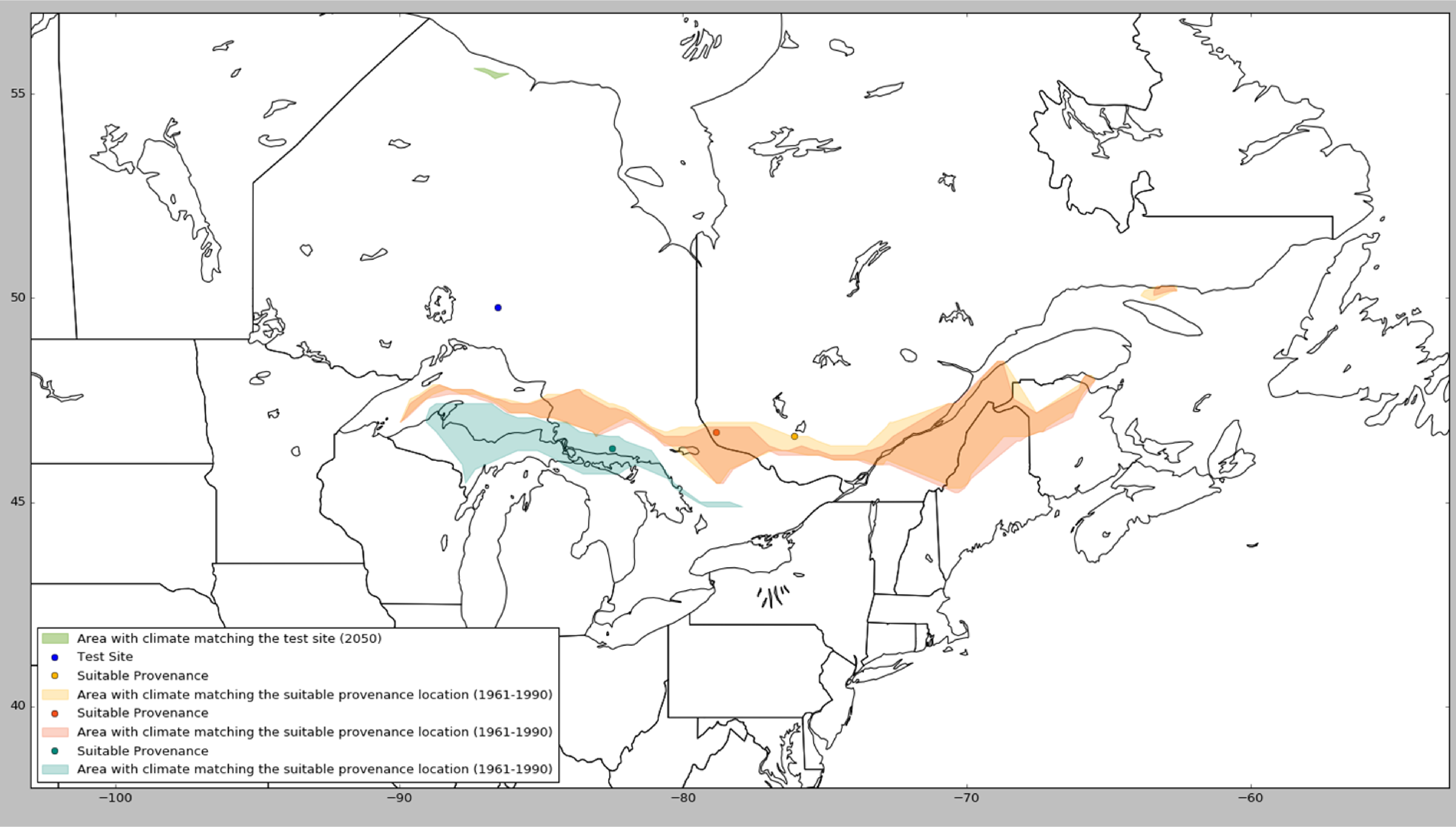
DATABASE 2: FUTURE CLIMATE

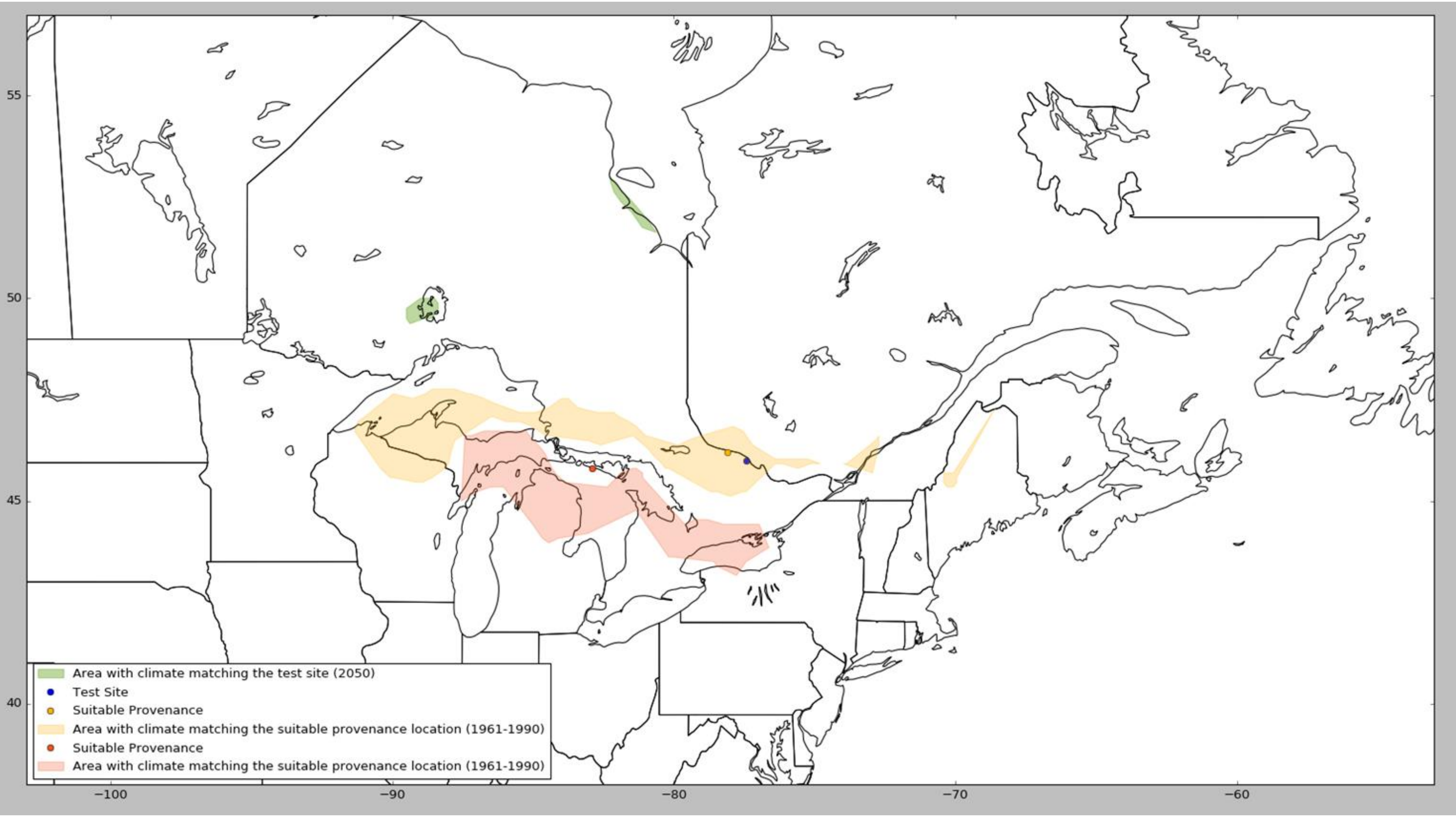
(F) Create a map of the areas

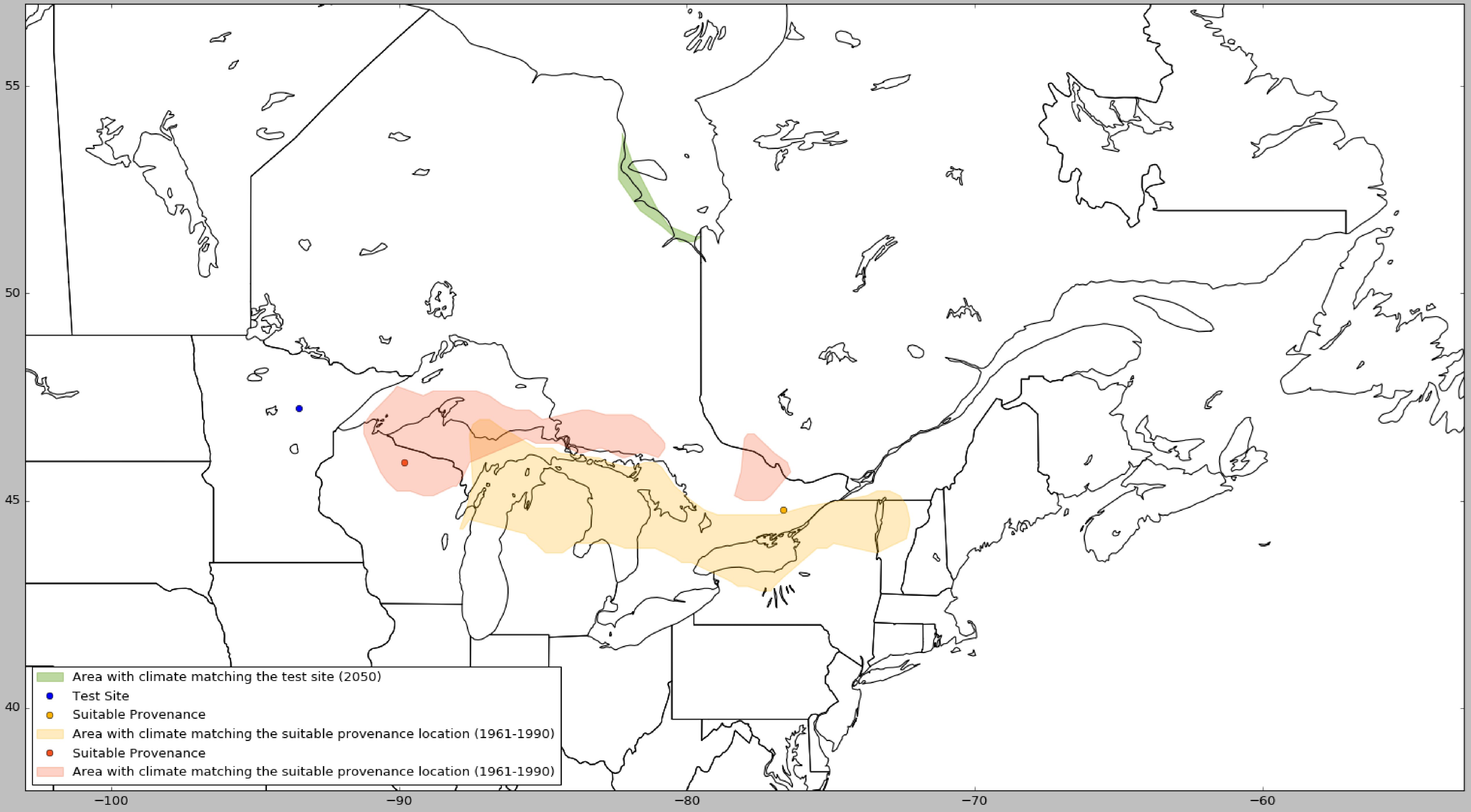
- Use Matplotlib to plot the spatial objects – North America, test sites, seed source origins
- Plot around the edge of the Delaunay Triangulations



55
50
45
40
−100
−90
−80
−70
−60
🟩 Area with climate matching the test site (2050)
● Test Site
● Suitable Provenance
🟥 Area with climate matching the suitable provenance location (1961-1990)
● Suitable Provenance
🟨 Area with climate matching the suitable provenance location (1961-1990)


55
50
45
40
−100
−90
−80
−70
−60
🟩 Area with climate matching the test site (2050)
● Test Site
● Suitable Provenance
🟨 Area with climate matching the suitable provenance location (1961-1990)
● Suitable Provenance
🟥 Area with climate matching the suitable provenance location (1961-1990)
● Suitable Provenance
🟦 Area with climate matching the suitable provenance location (1961-1990)


55
50
45
40
−100
−90
−80
−70
−60
🟩 Area with climate matching the test site (2050)
● Test Site
● Suitable Provenance
🟧 Area with climate matching the suitable provenance location (1961-1990)
● Suitable Provenance
🟥 Area with climate matching the suitable provenance location (1961-1990)


55
50
45
40
−100
−90
−80
−70
−60
🟩 Area with climate matching the test site (2050)
● Test Site
● Suitable Provenance
🟨 Area with climate matching the suitable provenance location (1961-1990)
● Suitable Provenance
🟥 Area with climate matching the suitable provenance location (1961-1990)

REFERENCES

Images:

1- NRCAN. National Research Forests – Valcartier [jpg]. Retrieved from: https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/forest/NationalReseachForests/Valcartier.jpg

2- Russell, Nancy. (2018). Government nursery expanding 'essential' tree selection [jpg]. Retrieved from: https://i.cbc.ca/1.4653106.1525786786!/fileImage/httpImage/image.jpg_gen/derivatives/16x9_780/pei-tree-nursery-7.jpg

3- National Geographic. Blue marble earth [jpg]. Retrieved from: https://www.nationalgeographic.com/content/dam/news/2015/07/21/bluemarble/01bluemarbleearth.jpg

4- Université Libre de Bruxelles. Great circle distance [png]. Retrieved from: https://dev.ulb.ac.be/ceese/ABC_Impacts/glossary/images_glossary/gcd.png

5- Haversine Formula [png]. Retrieved from: https://user-images.githubusercontent.com/2789198/27240436-e9a459da-52d4-11e7-8f84-f96d0b312859.png

6- NRCAN. 2056. Retrieved from: https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/cfs/assets/image/2056

7- NRCAN. Tallying_trees_regen_measurements_875 [jpg]. Retrieved from: https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/forest/NationalReseachForests/Tallying_trees_regen_measurements_875px.jpg

8- Mathmunch. (2013). Delaunay Triangulation [png]. Retrieved from: https://mathmunch.files.wordpress.com/2013/05/delaunay-triangulation.png

9- Onlinemath4all. Heron's Formula [png]. Retrieved from: https://www.onlinemath4all.com/images/heronsformula.png

10- UBC Math. Fig7 [gif]. Retrieved from: http://www.math.ubc.ca/~cass/courses/m309-01a/goon/fig7.gif