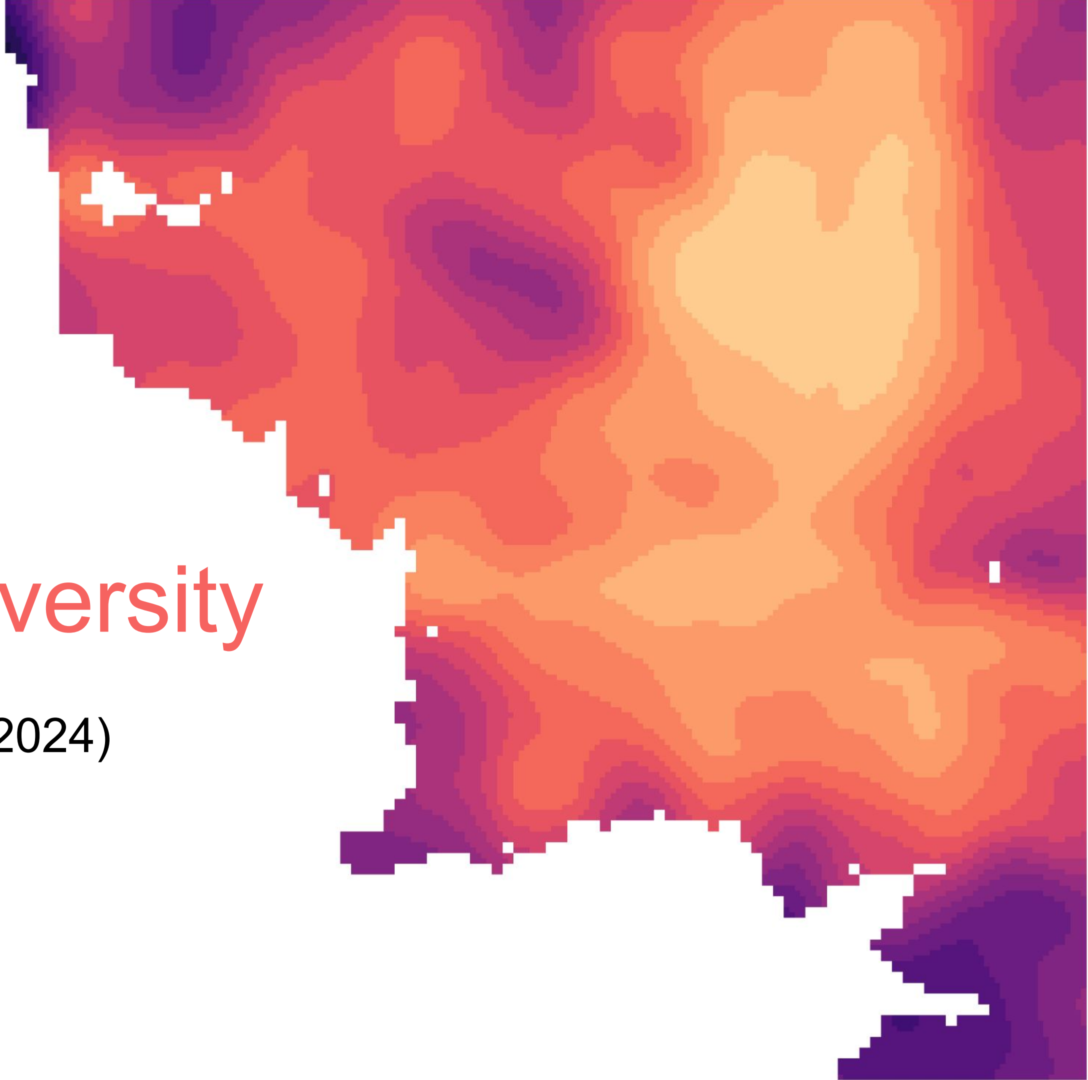
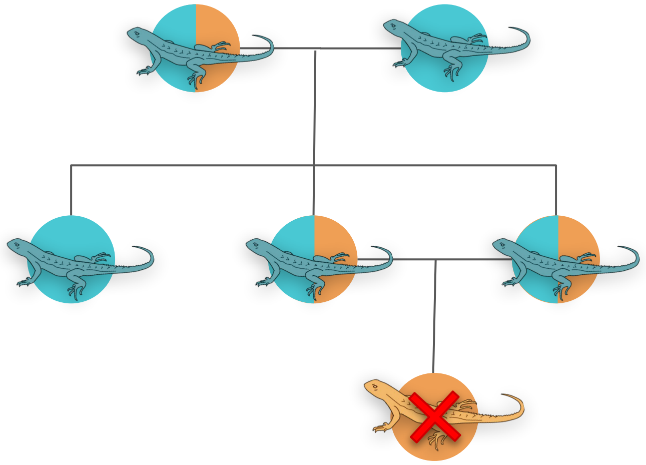
wingen:
Mapping genetic diversity

Anusha Bishop & Anne Chambers (2024)

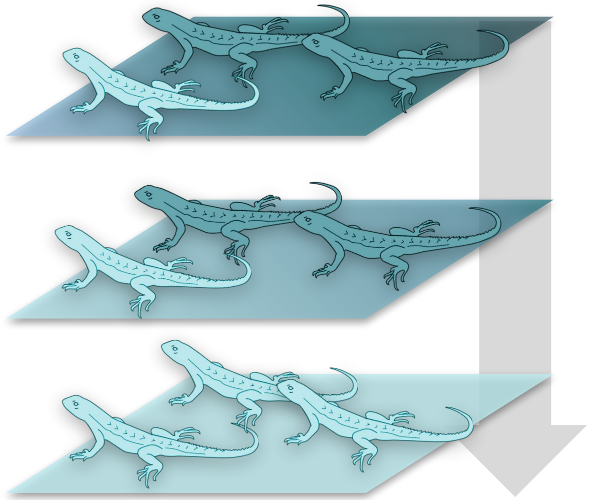


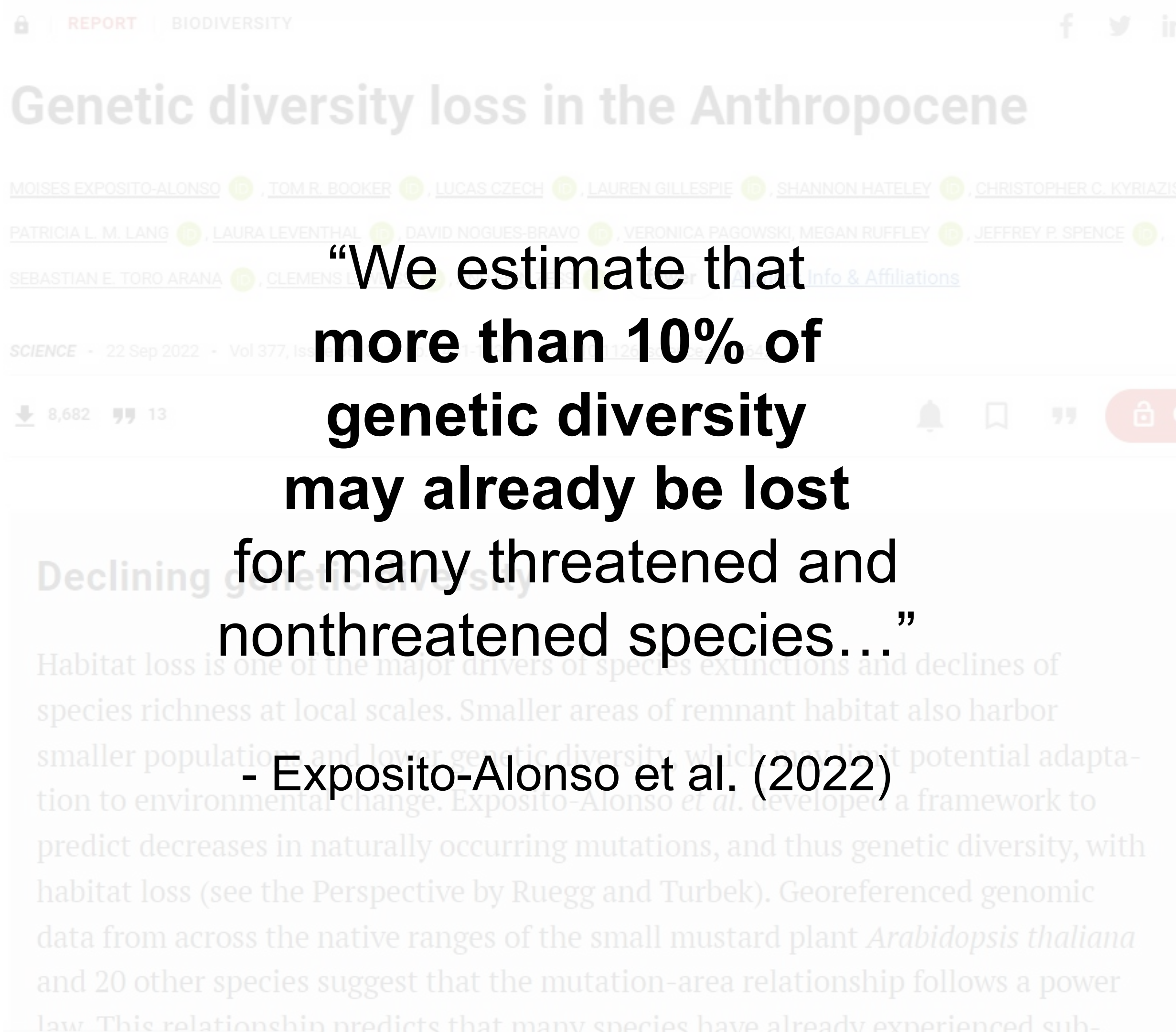
The importance of genetic diversity

Population Viability



Evolutionary Potential




REPORT | BIODIVERSITY

Genetic diversity loss in the Anthropocene

MOISES EXPOSITO-ALONSO, TOM R. BOOKER, LUCAS CZECH, LAUREN GILLESPIE, SHANNON HATELEY, CHRISTOPHER C. KYRIAZIS
PATRICIA L. M. LANG, LAURA LEVENTHAL, DAVID NOGUES-BRAVO, VERONICA PAGOWSKI, MEGAN RUFFLEY, JEFFREY P. SPENCE
SEBASTIAN E. TORO ARANA, CLEMENS … Info & Affiliations

SCIENCE · 22 Sep 2022 · Vol 377, Is…

8,682 13

Declining genetic diversity

Habitat loss is one of the major drivers of species extinctions and declines of species richness at local scales. Smaller areas of remnant habitat also harbor smaller populations and lower genetic diversity, which may limit potential adaptation to environmental change. Exposito-Alonso *et al.* developed a framework to predict decreases in naturally occurring mutations, and thus genetic diversity, with habitat loss (see the Perspective by Ruegg and Turbek). Georeferenced genomic data from across the native ranges of the small mustard plant *Arabidopsis thaliana* and 20 other species suggest that the mutation-area relationship follows a power law. This relationship predicts that many species have already experienced sub-

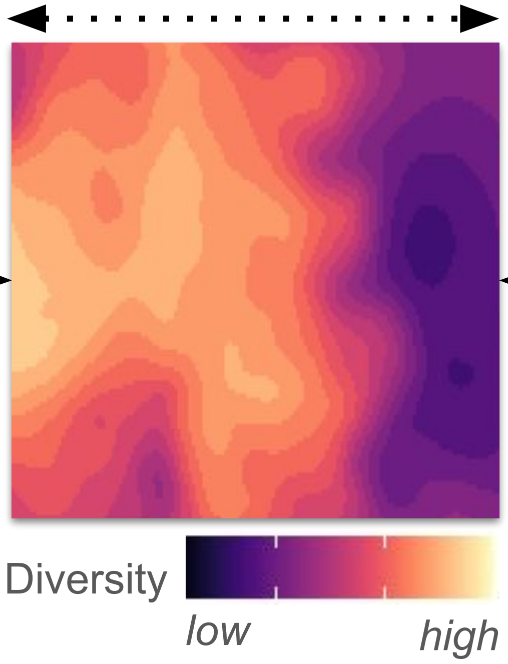
"We estimate that **more than 10% of genetic diversity may already be lost** for many threatened and nonthreatened species…"

- Exposito-Alonso et al. (2022)

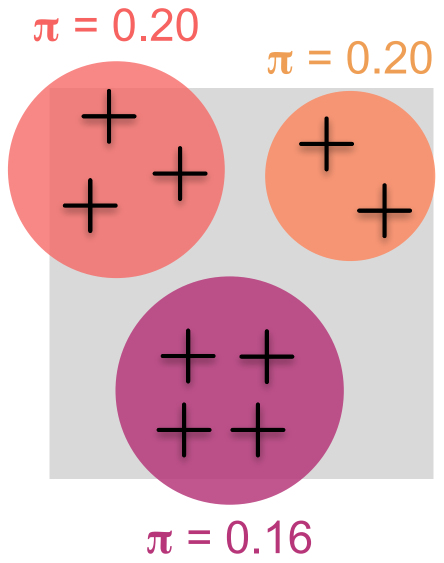
Understanding genetic diversity across landscapes

Determine **drivers of genetic diversity** patterns

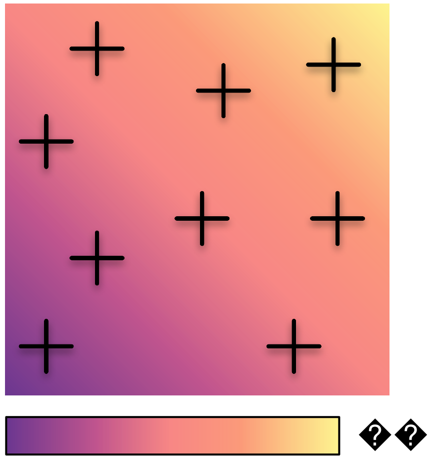
Protect valuable areas of **high genetic diversity**



Identify vulnerable areas of **low genetic diversity**



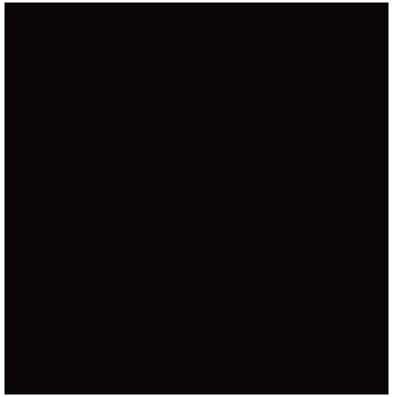
Traditional:
calculating genetic diversity by population



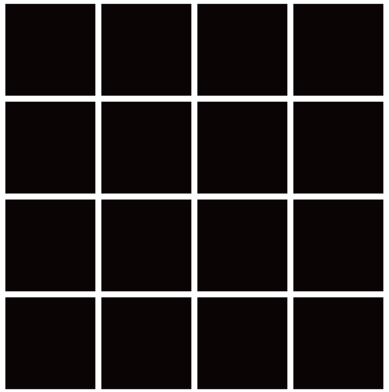
New:
calculating genetic
diversity <u>continuously</u>

wingen

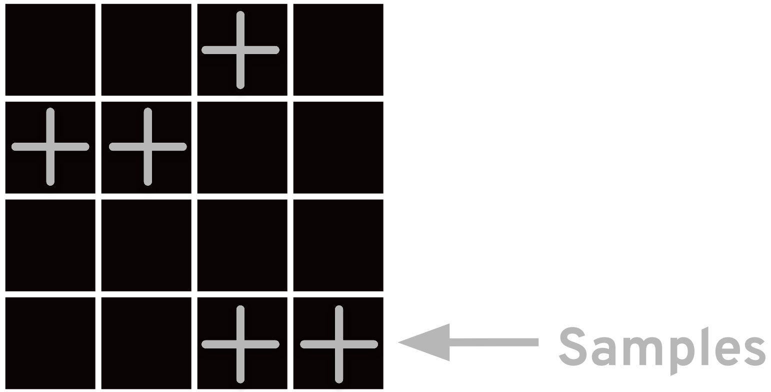
Continuous mapping of genetic diversity using moving windows

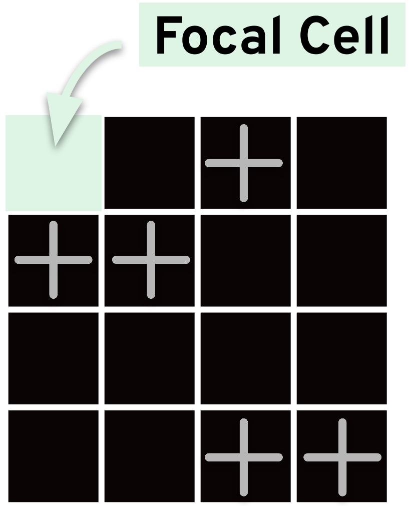


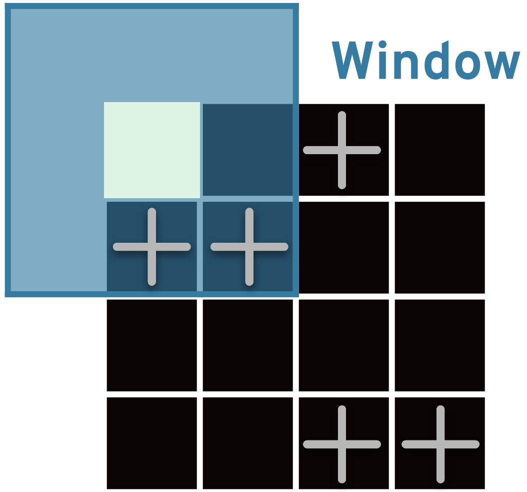
Example Landscape

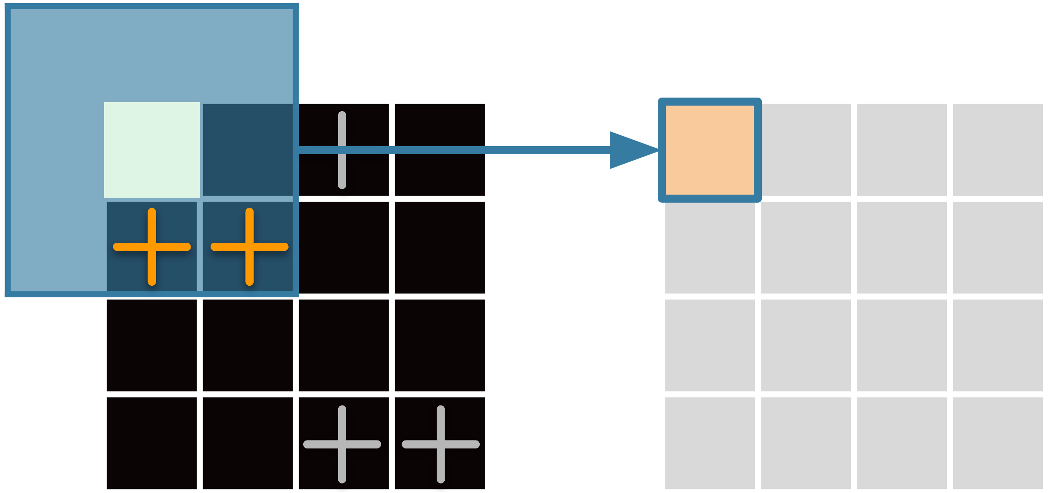


Example Landscape

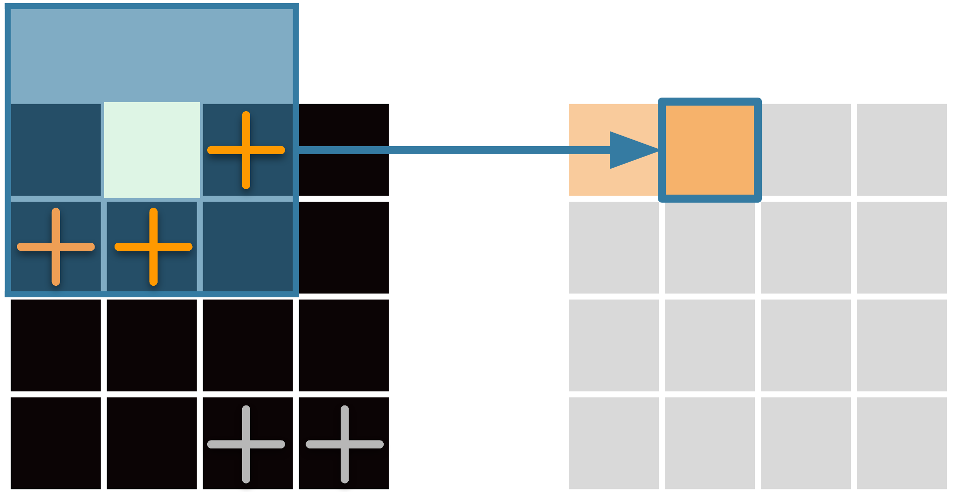

Samples


Focal Cell

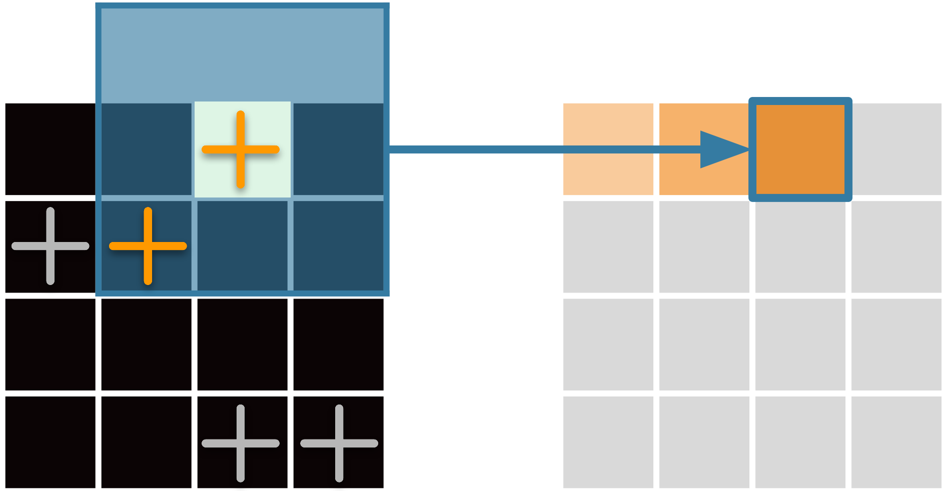

Window

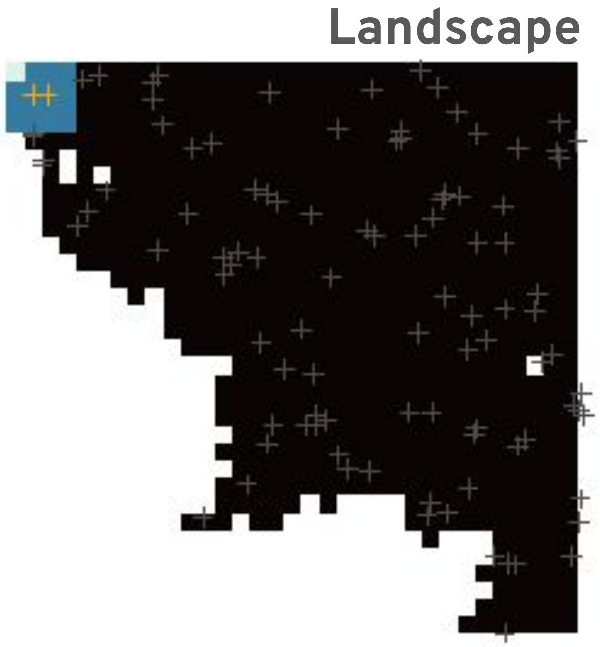


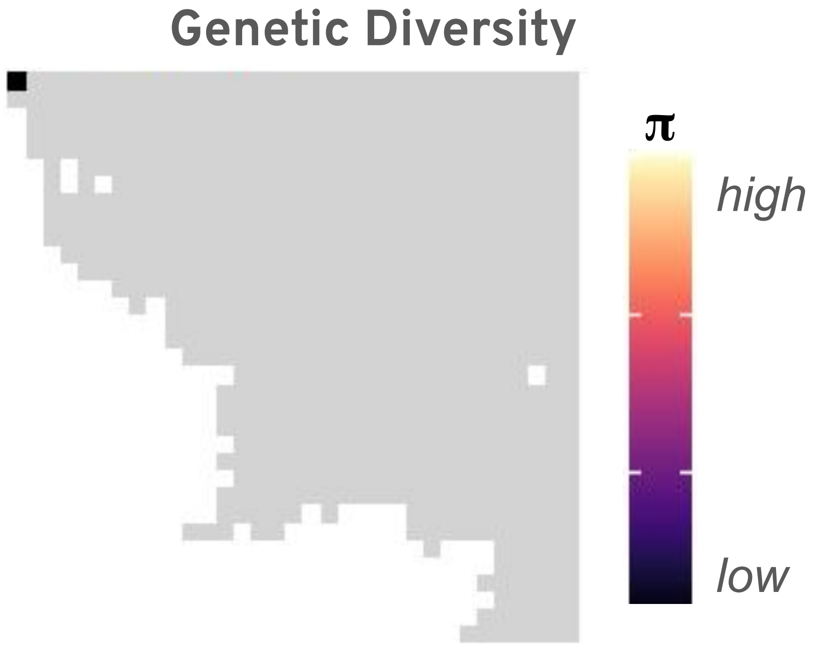
Genetic Diversity

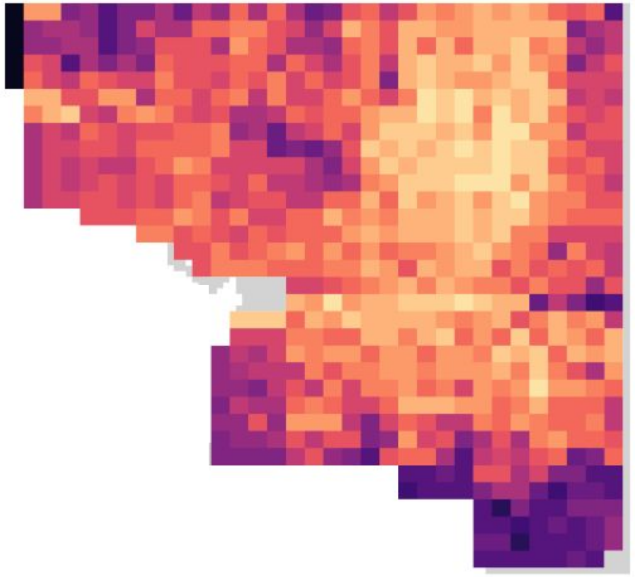


Genetic
Diversity

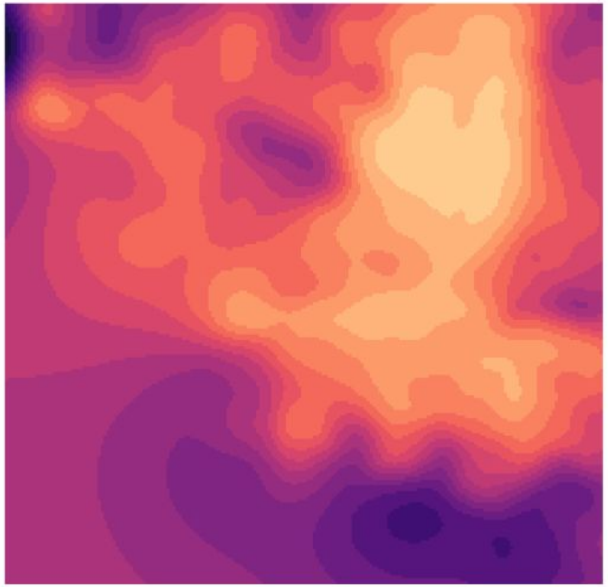

Genetic Diversity


Landscape

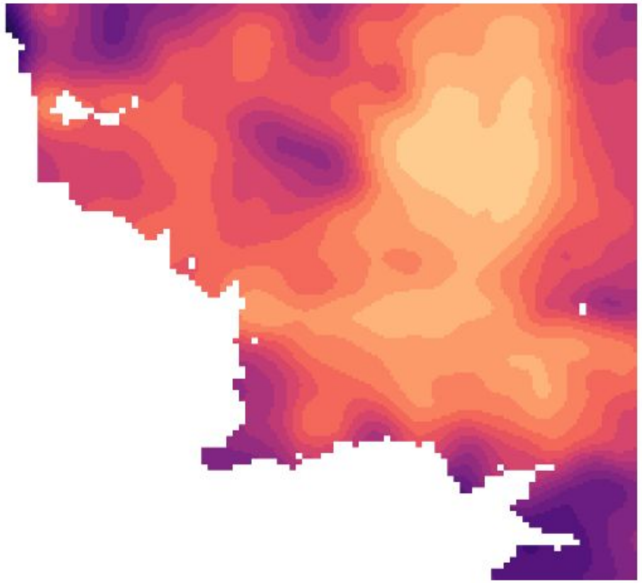

Genetic Diversity
π
high
low



Moving window
maps of π,
allelic richness,
heterozygosity,
and more…

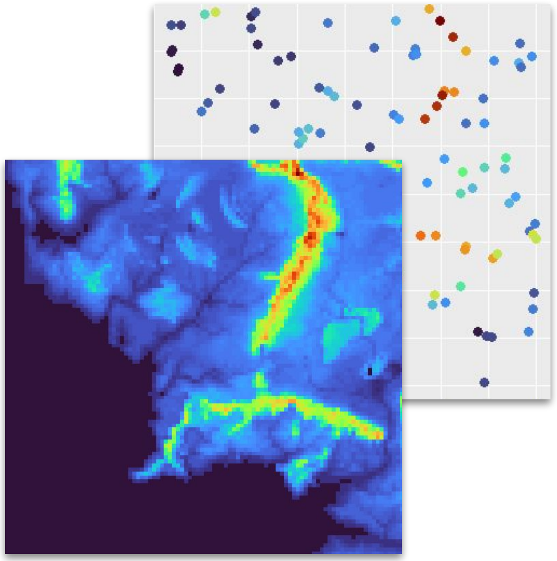


Kriging to produce interpolated maps of genetic diversity



Masking to exclude undersampled regions

Generalized window functions



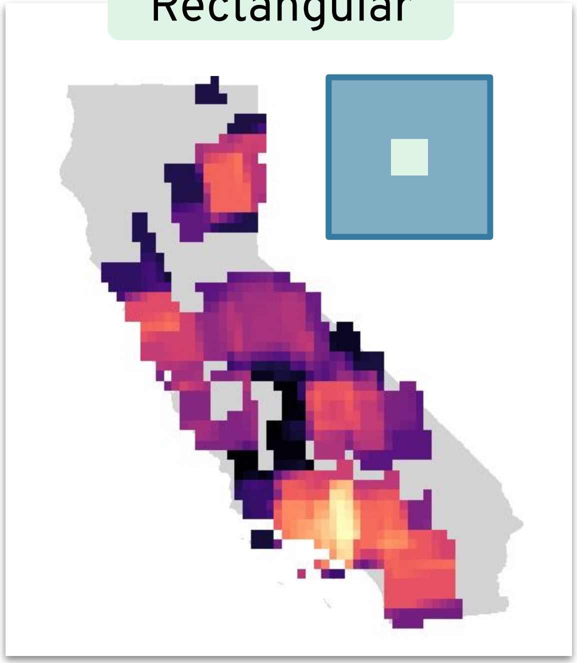
Input non-genomic data and use custom statistics:

- ✓ Phenotypic data
- ✓ Environmental data
- ✓ Anything else that can be formatted as a matrix or dataframe…

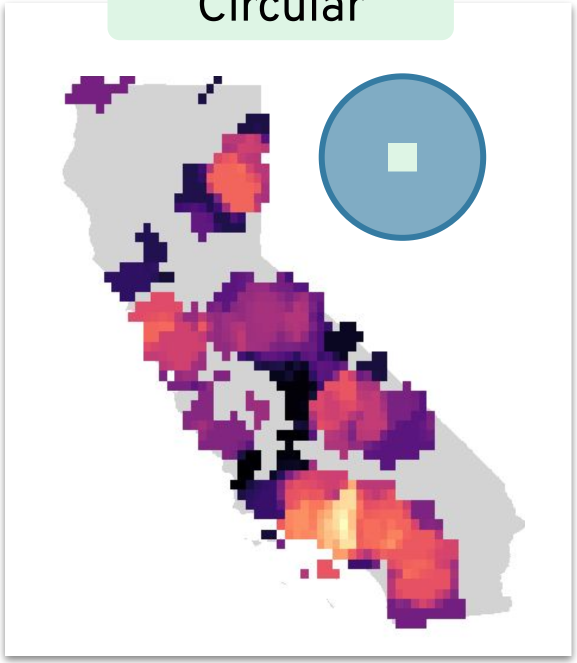
Window options

original

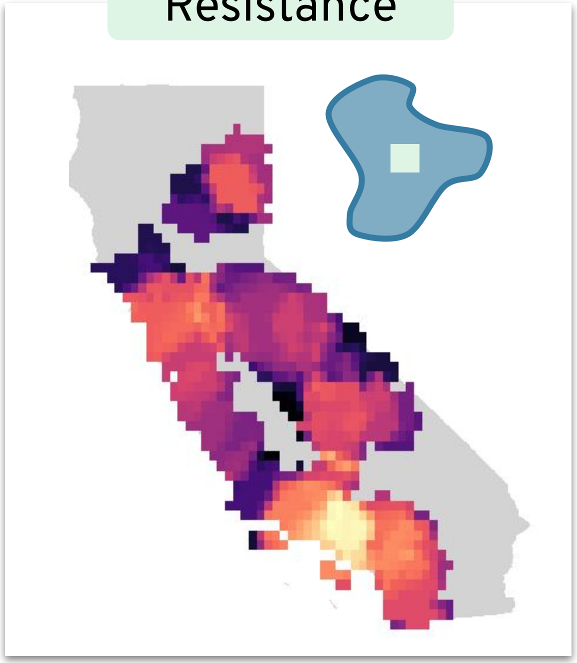
Rectangular



Circular



Resistance



EXERCISES

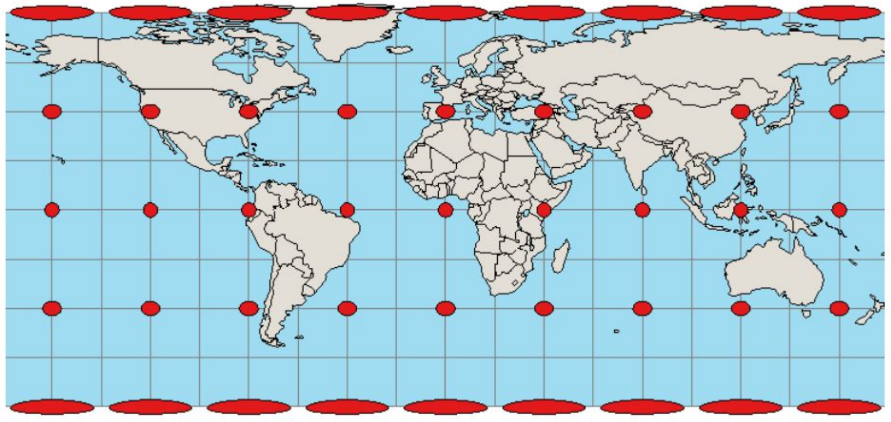
Crash course: **coordinate projections**

Why can't we just used unprojected longitude/latitude?

Problem for our windows:

Longitudes are closer together the further you move away → the size of the window will change based where we are in globe

One latitude unit does not equal one longitude unit → the window will be rectangular when we want it to be square



Unprojected Latitude and Longitude (EPSG 4326)

https://michaelminn.net/tutorials/gis-projections/index.html

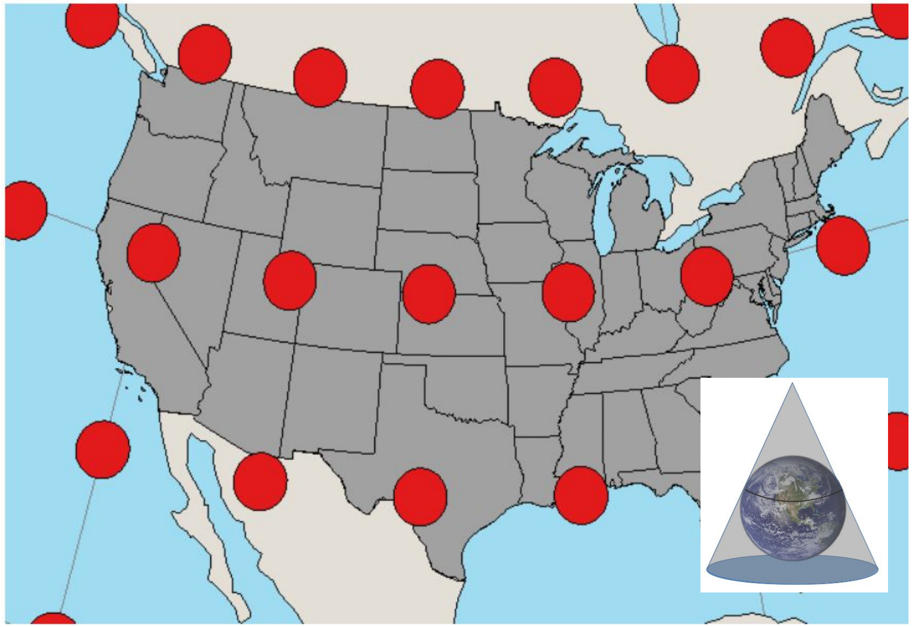
Crash course: **coordinate projections**

Equal-area projections

The size of any area is in proportion to the size on the earth

No projections are perfect, all have a some kind of distortion:

1. Conformal - preserve angles (shape), distort areas and distances
2. Equal area - preserve area, distort angles (shape)
3. Equidistant - preserve distances, but only from certain points/lines



North America Albers Equal Area Conic Projection (ESRI 102008)

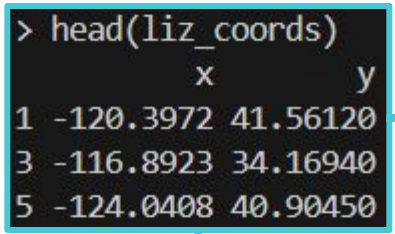
https://michaelminn.net/tutorials/gis-projections/index.html

1. Reprojecting coordinates

Try running:
`plot(coords_longlat)`
`plot(coords_proj)`

`st_as_sf()` *converts our coordinates into a sf (spatial) object*

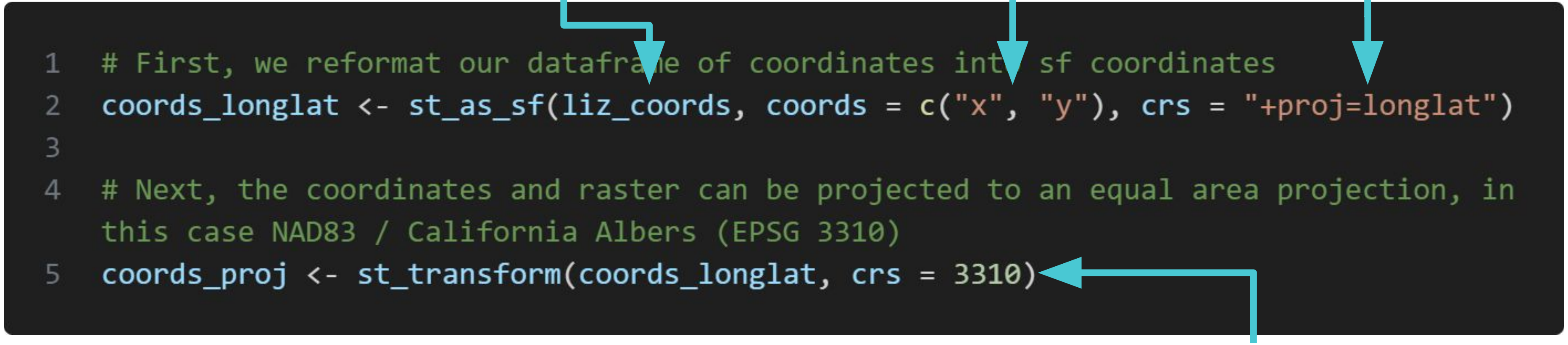
`liz_coords`



```
> head(liz_coords)
          x        y
1 -120.3972 41.56120
3 -116.8923 34.16940
5 -124.0408 40.90450
```

`coords` = the names of the columns with x/y coordinates

`crs` = the original CRS (longitude/latitude)

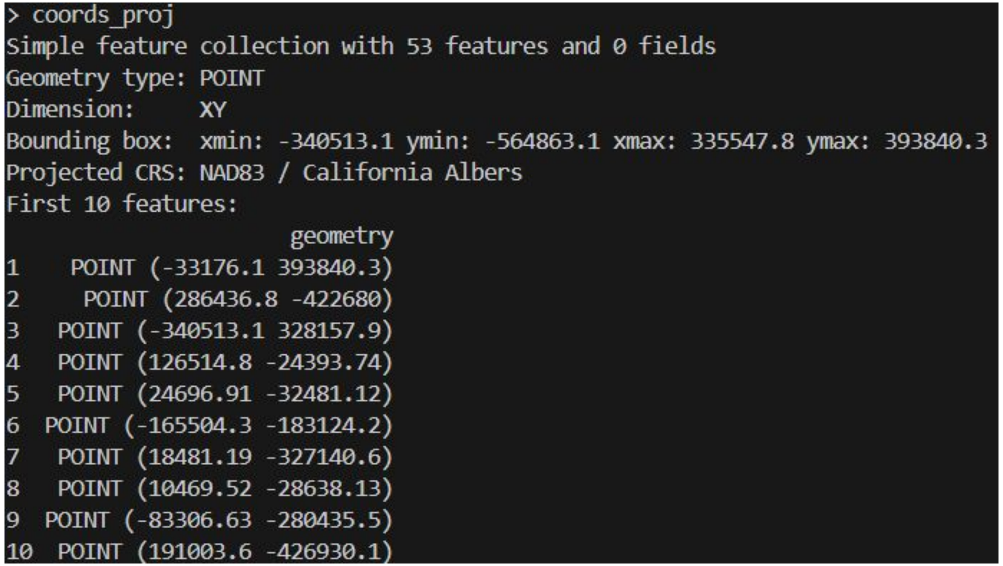


```
# First, we reformat our dataframe of coordinates into sf coordinates
coords_longlat <- st_as_sf(liz_coords, coords = c("x", "y"), crs = "+proj=longlat")

# Next, the coordinates and raster can be projected to an equal area projection, in this case NAD83 / California Albers (EPSG 3310)
coords_proj <- st_transform(coords_longlat, crs = 3310)
```

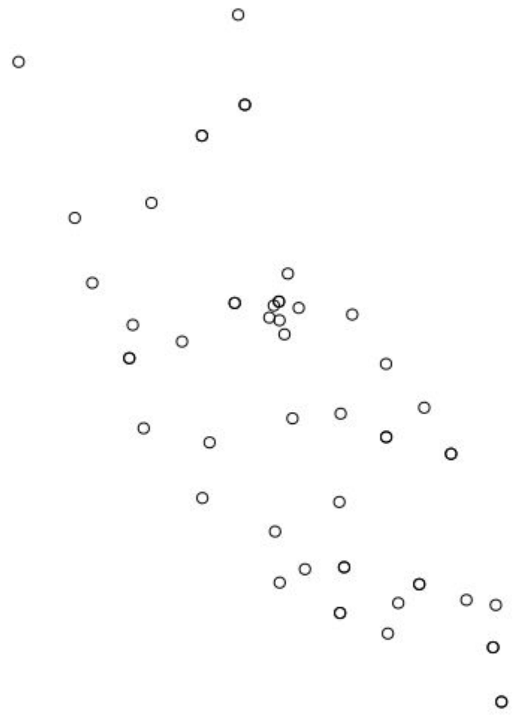
`st_transform()` *projects our coordinates to a new CRS*

`crs` = the <u>new</u> CRS

Output: sf object



```
> coords_proj
Simple feature collection with 53 features and 0 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -340513.1 ymin: -564863.1 xmax: 335547.8 ymax: 393840.3
Projected CRS: NAD83 / California Albers
First 10 features:
                     geometry
1    POINT (-33176.1 393840.3)
2     POINT (286436.8 -422680)
3   POINT (-340513.1 328157.9)
4   POINT (126514.8 -24393.74)
5   POINT (24696.91 -32481.12)
6  POINT (-165504.3 -183124.2)
7   POINT (18481.19 -327140.6)
8   POINT (10469.52 -28638.13)
9  POINT (-83306.63 -280435.5)
10  POINT (191003.6 -426930.1)
```



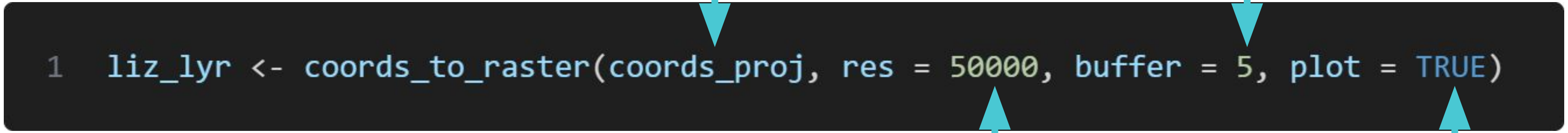
2. Creating raster from coordinates

See what happens when you change the value of **`res`**

`coords_to_raster()` *creates a raster from our projected coordinates*

our projected coordinates

`buffer` = the number of cells to add around the coordinates as a buffer

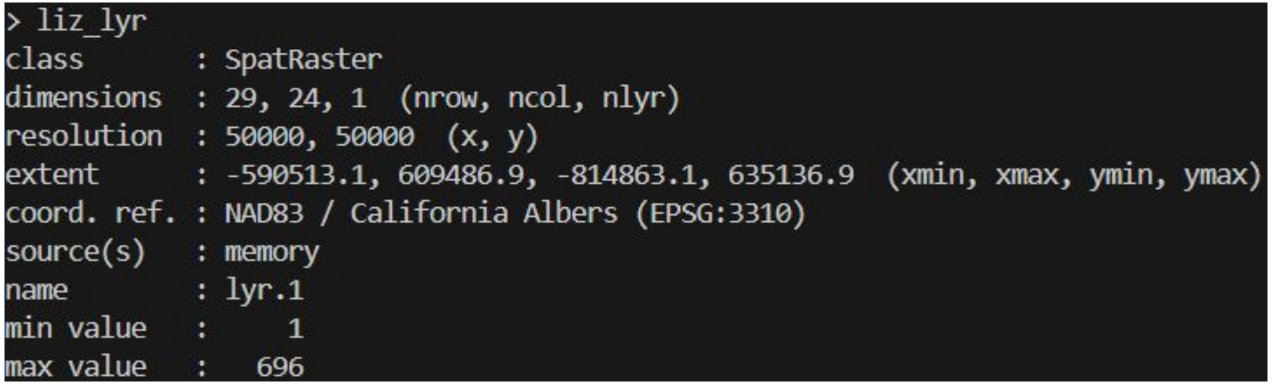


```r
liz_lyr <- coords_to_raster(coords_proj, res = 50000, buffer = 5, plot = TRUE)
```

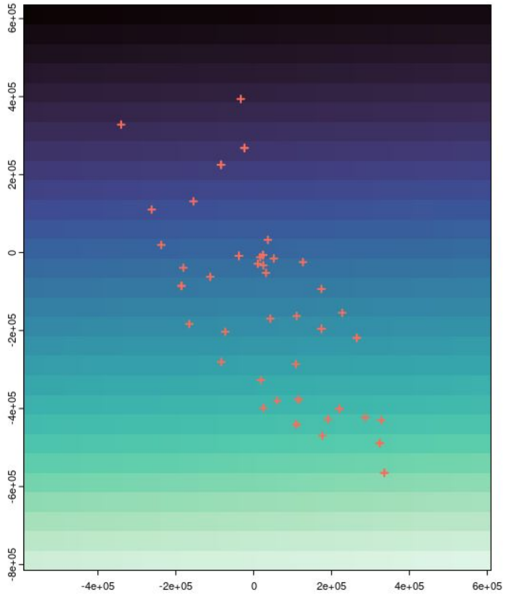
`res` = desired resolution for the output raster (the units are based on the coordinates; in this case it is in meters)

`plot` = whether to plot the resulting raster

Output: SpatRaster



```
> liz_lyr
class       : SpatRaster
dimensions  : 29, 24, 1  (nrow, ncol, nlyr)
resolution  : 50000, 50000  (x, y)
extent      : -590513.1, 609486.9, -814863.1, 635136.9  (xmin, xmax, ymin, ymax)
coord. ref. : NAD83 / California Albers (EPSG:3310)
source(s)   : memory
name        : lyr.1
min value   :     1
max value   :   696
```



Previewing window

See what happens when you change the value of **`wdim`** and **`fact`**

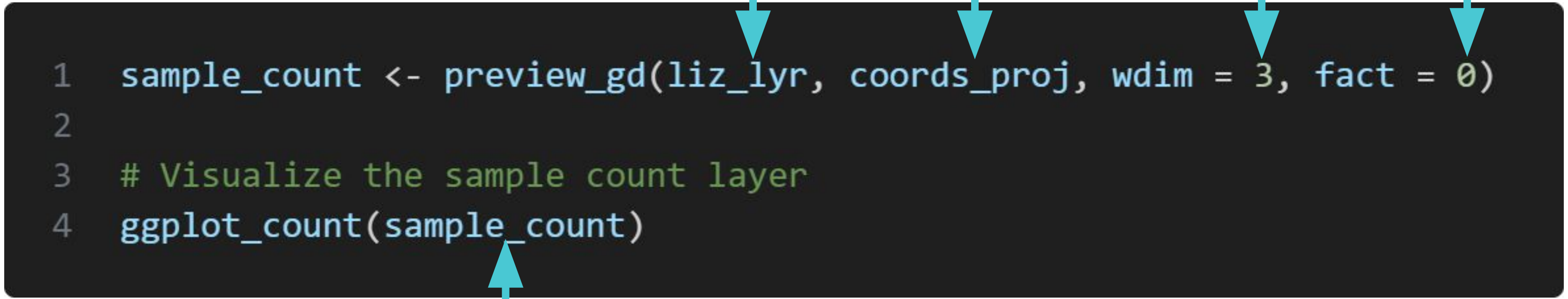
`preview_gd()` *previews the moving window and sampling counts*

raster

coordinates

`wdim` = window dimensions in cells

`fact` = aggregation factor for the raster (0 = no aggregation)

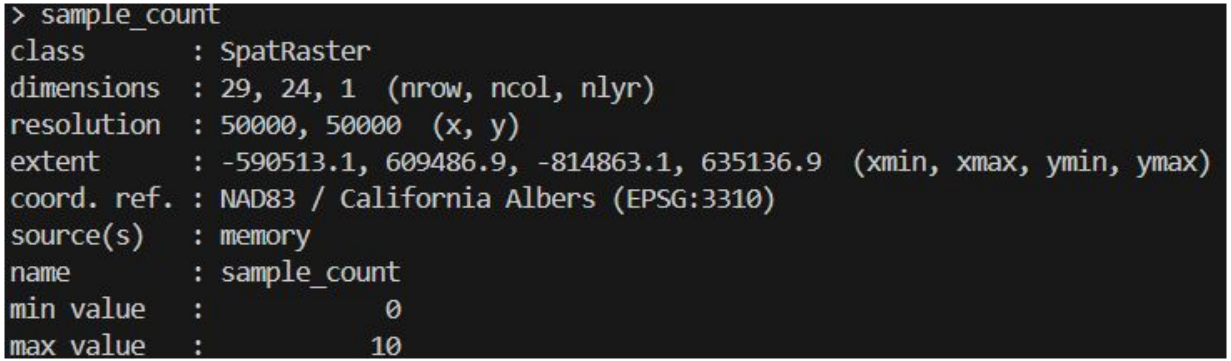


```
sample_count <- preview_gd(liz_lyr, coords_proj, wdim = 3, fact = 0)

# Visualize the sample count layer
ggplot_count(sample_count)
```

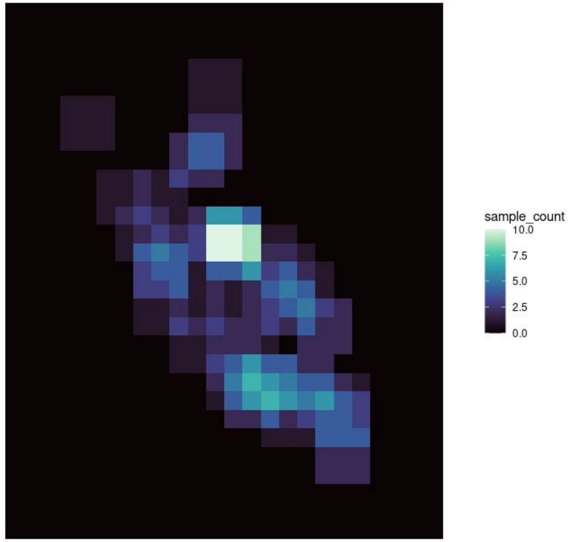
raster where values are sample counts in the window

`ggplot_count()` *plots the sample count rasters (you can also use* ***`plot_count()`*** *or* ***`just plot()`****)*

Output: SpatRaster

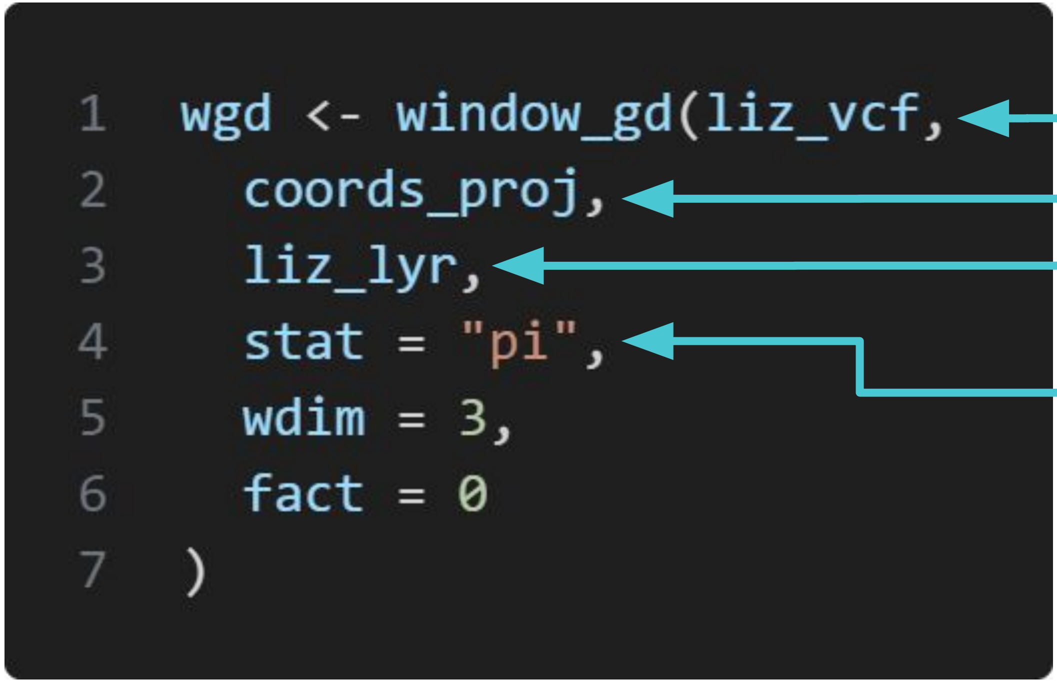


```
> sample_count
class       : SpatRaster
dimensions  : 29, 24, 1  (nrow, ncol, nlyr)
resolution  : 50000, 50000  (x, y)
extent      : -590513.1, 609486.9, -814863.1, 635136.9  (xmin, xmax, ymin, ymax)
coord. ref. : NAD83 / California Albers (EPSG:3310)
source(s)   : memory
name        : sample_count
min value   :            0
max value   :           10
```



Run the moving window

`window_gd()` runs the moving window analysis



```
wgd <- window_gd(liz_vcf,
  coords_proj,
  liz_lyr,
  stat = "pi",
  wdim = 3,
  fact = 0
)
```

VCF

Coordinates

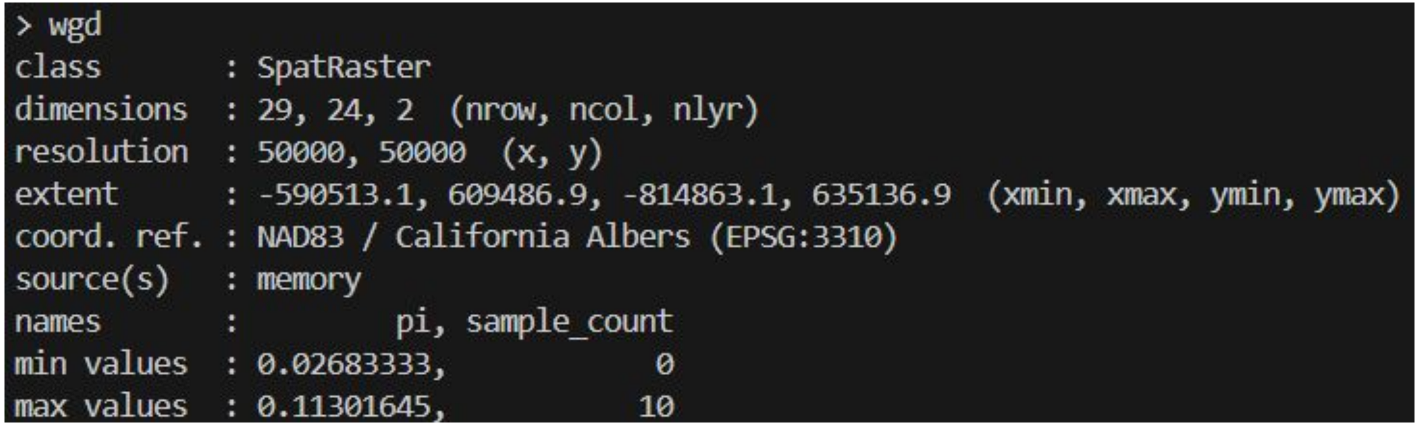
Raster

`stat` = statistic to calculate

*Checkout other arguments you can change by running **`?window_gd`***

See what happens when you change the value of **`wdim`**, **`fact`**, and **`min_n`** (min # of samples in a window).

Output: SpatRaster



```
> wgd
class       : SpatRaster
dimensions  : 29, 24, 2  (nrow, ncol, nlyr)
resolution  : 50000, 50000  (x, y)
extent      : -590513.1, 609486.9, -814863.1, 635136.9  (xmin, xmax, ymin, ymax)
coord. ref. : NAD83 / California Albers (EPSG:3310)
source(s)   : memory
names       :         pi, sample_count
min values  : 0.02683333,            0
max values  : 0.11301645,           10
```

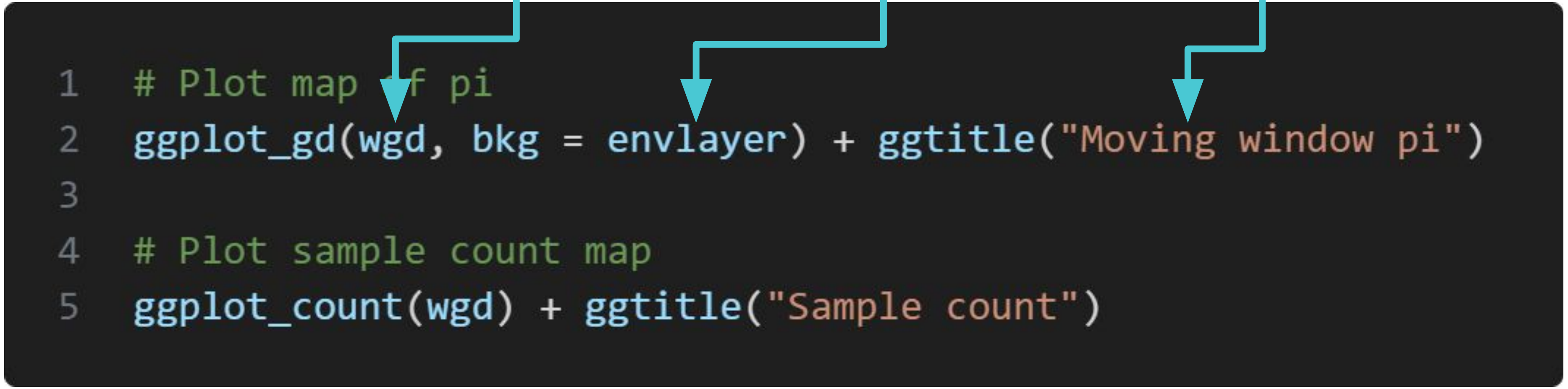
Plot results

ggplot_gd() *plots the moving window raster*

Raster

bkg = raster or other spatial object to use for background

This plot can be modified using **ggplot2** functions

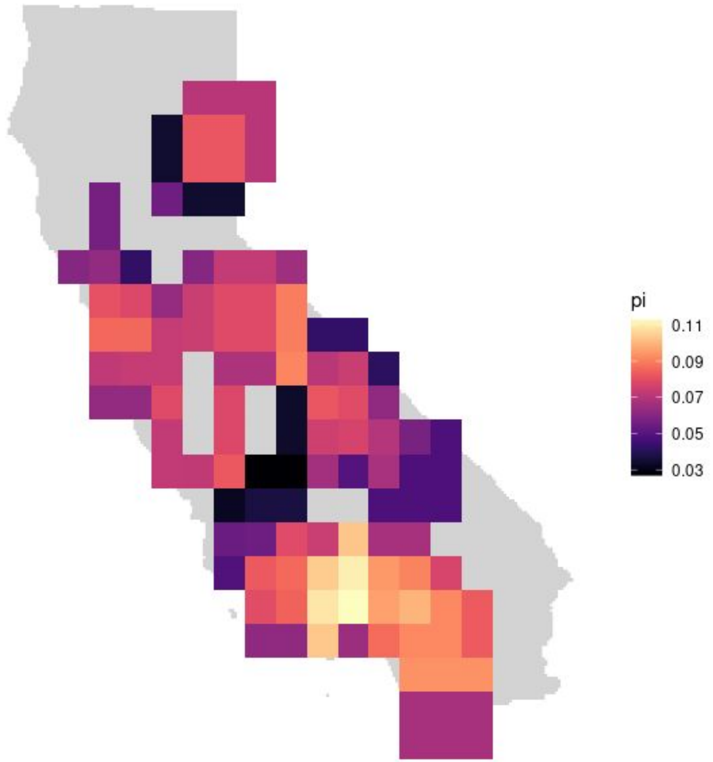


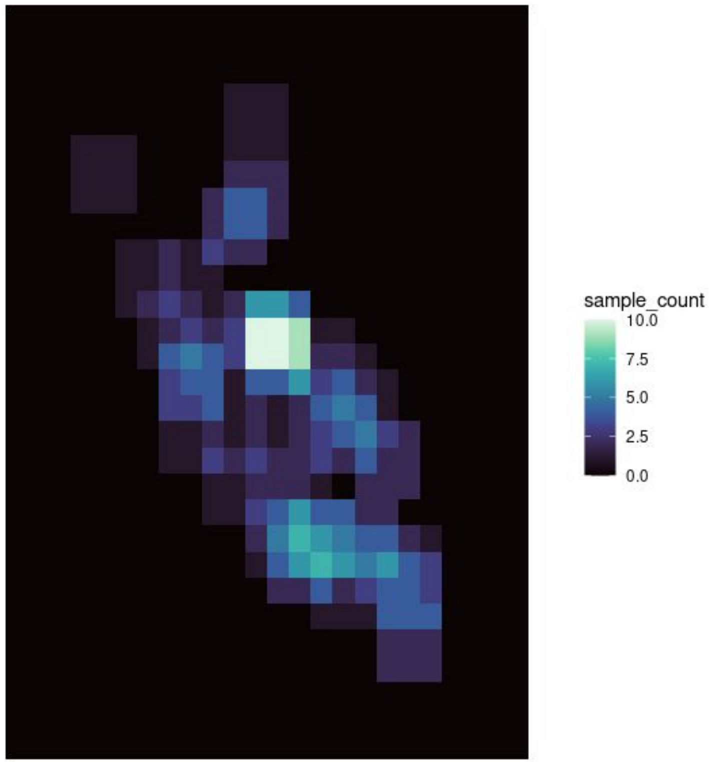
```r
# Plot map of pi
ggplot_gd(wgd, bkg = envlayer) + ggtitle("Moving window pi")

# Plot sample count map
ggplot_count(wgd) + ggtitle("Sample count")
```

ggplot_count() *plots the sample count layer window raster*

Try changing the color scheme by running **ggplot_gd() + scale_fill_viridis_c(option = "inferno")** (other options to try are "turbo", "viridis", "plasma", "rocket", etc.)

Output: ggplots





Krige results

`index` = which raster layers to krige (here we use the first (diversity) and the second (sample count) layers)

Moving window raster

Raster to interpolate across

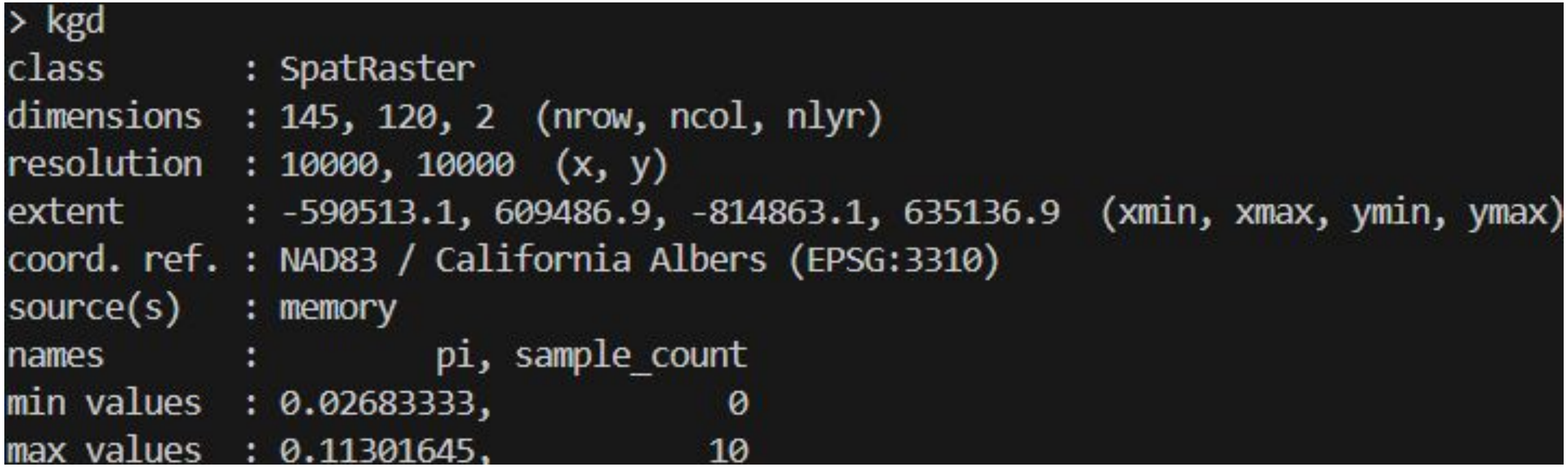
```r
kgd <- krig_gd(wgd, index = 1:2, liz_lyr, disagg_grd = 5)
```

`krig_gd()` *produce a spatially interpolated (smoothed) map*

`disagg_grd` = factor by which to disaggregate the raster used for interpolation (**`liz_lyr`**). This increases the resolution of the smoothed layer.

See what happens when you change **`disagg_grd`**

Output: SpatRaster



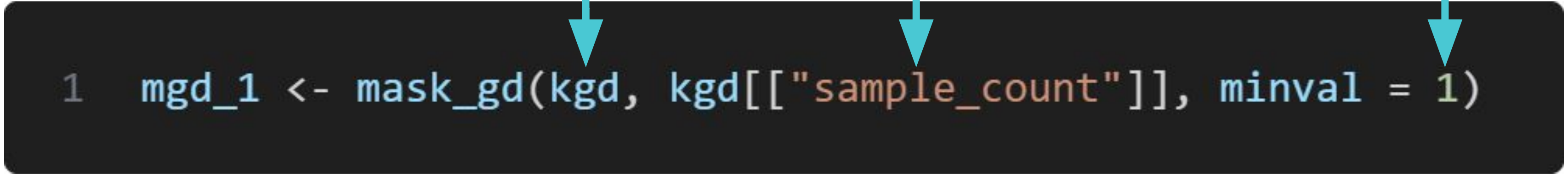
```
> kgd
class       : SpatRaster
dimensions  : 145, 120, 2  (nrow, ncol, nlyr)
resolution  : 10000, 10000  (x, y)
extent      : -590513.1, 609486.9, -814863.1, 635136.9  (xmin, xmax, ymin, ymax)
coord. ref. : NAD83 / California Albers (EPSG:3310)
source(s)   : memory
names       :         pi, sample_count
min values  : 0.02683333,            0
max values  : 0.11301645,           10
```

Mask results

Moving window or kriged raster to mask

Raster to mask with

`minval` = minimum value of the masking raster, below which values are replaced with NA (masked)



```r
mgd_1 <- mask_gd(kgd, kgd[["sample_count"]], minval = 1)
```

`mask_gd()` *mask the moving window or kriged rasters using another raster or spatial object*

You can also use a spatial object (e.g. a vector/polygon) for masking

If no minval is provided, areas of the masking raster that are NA will be masked

See what happens when you change **`minval`** or use **`wgd[[“sample_count”]]`** instead https://thewanglab.github.io/algatr/articles/wingen_vignette.html

Exercises

1. Load the example dataset
2. Create inputs
    a. Project coordinates
    b. Create a raster layer from your coordinates using `coords_to_raster()`
3. Preview moving window map using `preview_gd()`
4. Create moving window map using `window_gd()`
5. Krige moving window map using `krig_gd()`
6. Mask moving window map using `mask_gd()`