clustermax
Stata command to create clusters from point coordinates

Francisco Barba

October 18, 2018

Overview

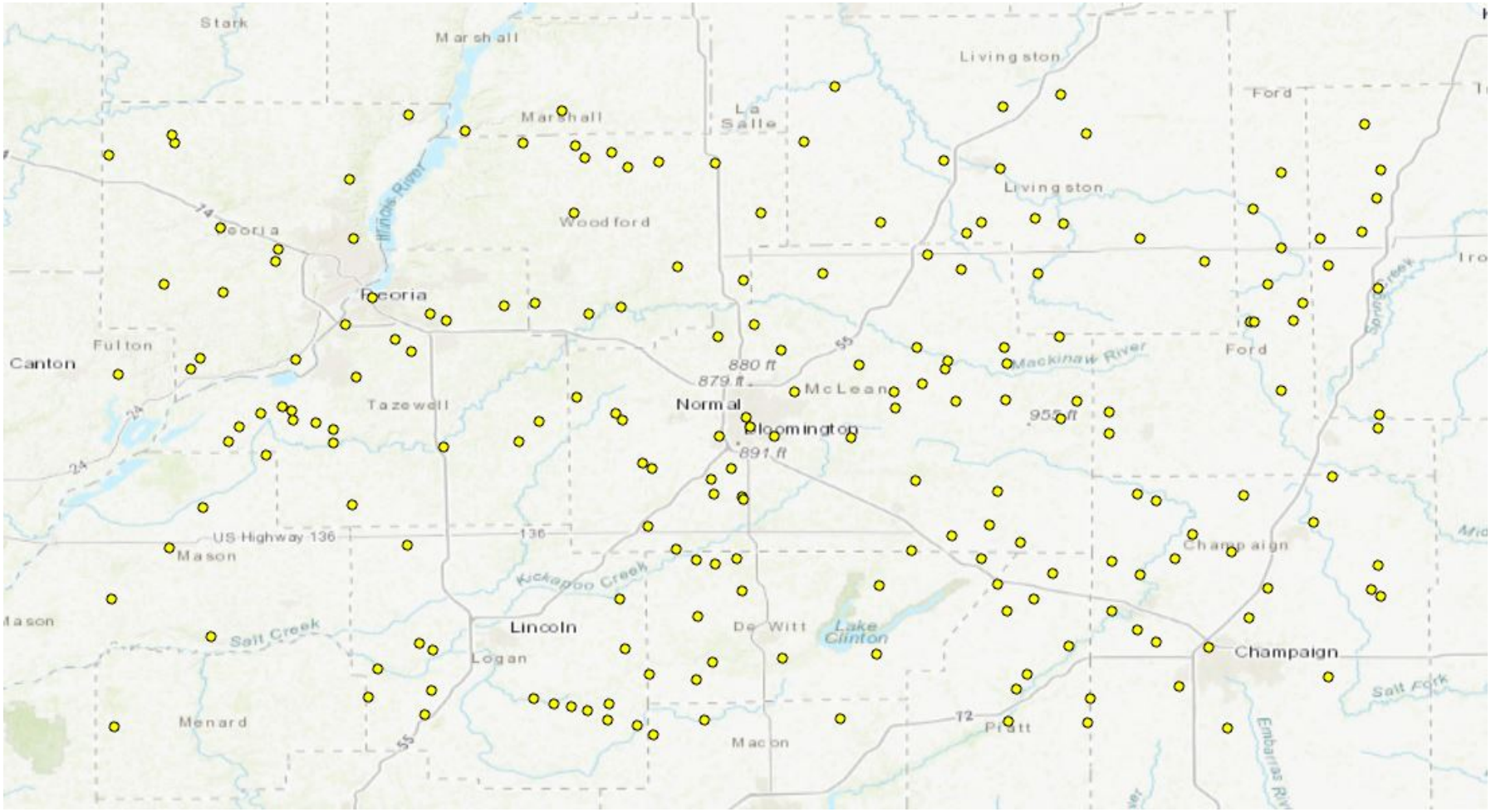
Overview

Examples

Generate 200 random points in Illinois, USA

```
. set seed 1
. set obs 200
. gen lat = 40 + runiform()
. gen lon = -90 + runiform()*2
```

Examples


Stark
Marshall
Livingston
Ford
La Salle
Marshall
Livingston
Woodford
Peoria
Fulton
Canton
Tazewell
Normal
McLean
Bloomington
880 ft
879 ft
891 ft
955 ft
Mackinaw River
Ford
US Highway 136
Mason
Champaign
Kickapoo Creek
Salt Creek
Lincoln
De Witt
Lake Clinton
Logan
Champaign
Salt Fork
Menard
Piatt
Macon

Examples

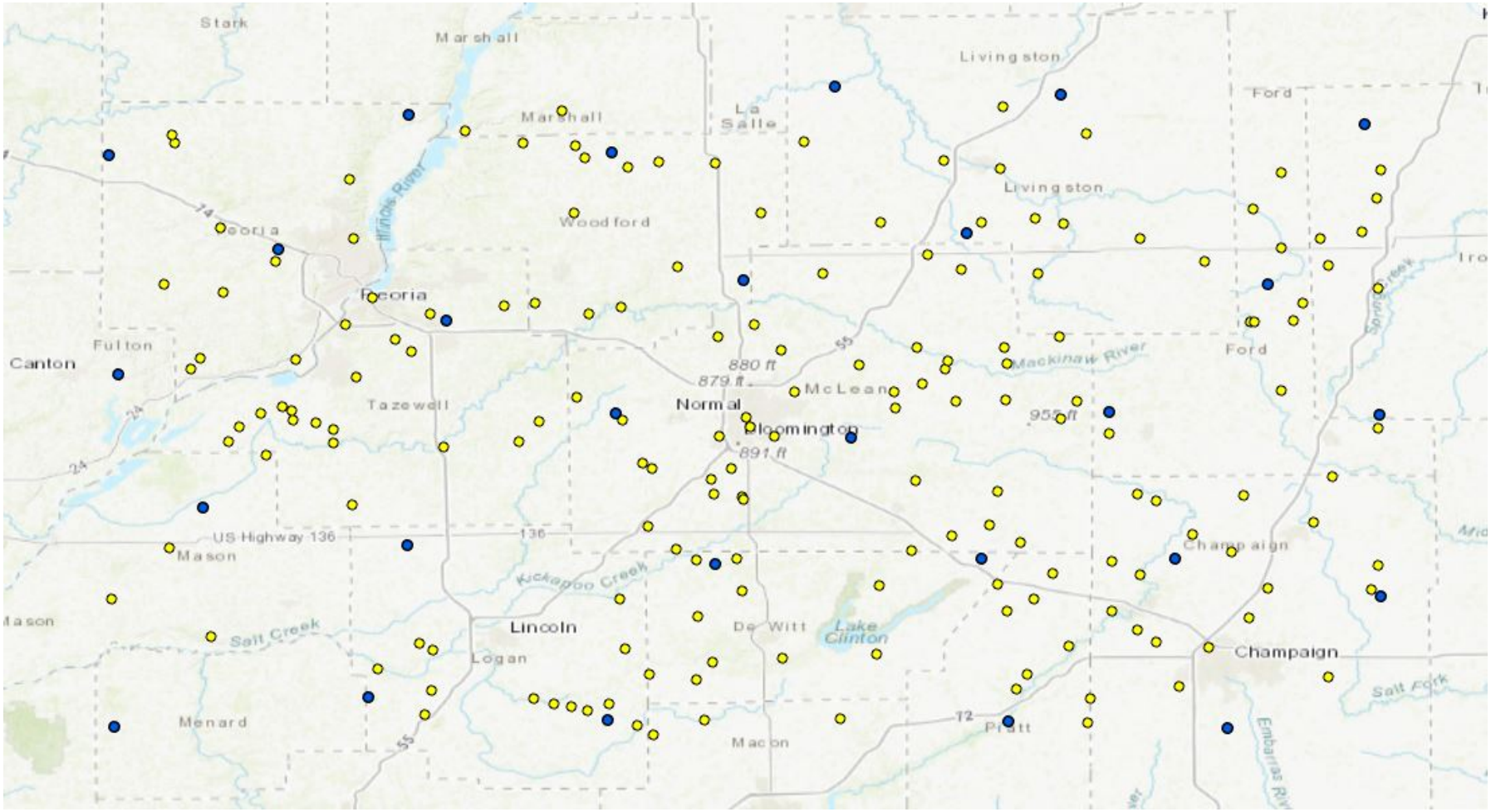
Question:

> How many points with a minimum mutual distance of 25km can be drawn from this map?

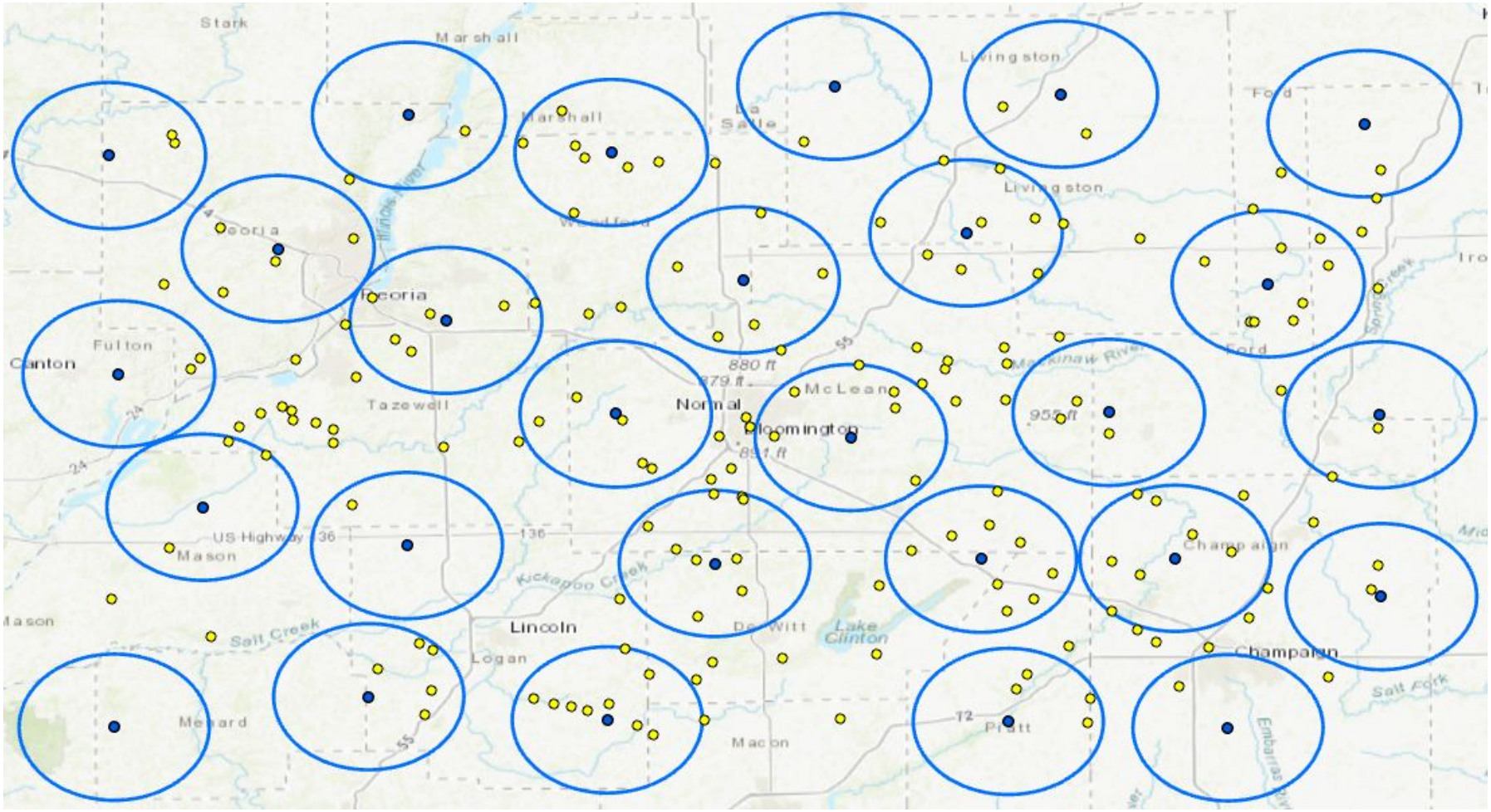
Run the command:

```
. clustermax lat lon, gen(cid) n(1) between(25)
seed(1)
```

Examples


Stark
Marshall
Livingston
Ford
La Salle
Peoria
Woodford
Fulton
Canton
Tazewell
Normal
McLean
Bloomington
880 ft
879 ft
891 ft
955 ft
Mackinaw River
US Highway 136
Mason
Champaign
Kickapoo Creek
Salt Creek
Lincoln
Logan
De Witt
Lake Clinton
Menard
Macon
Piatt
Champaign
Salt Fork

Examples

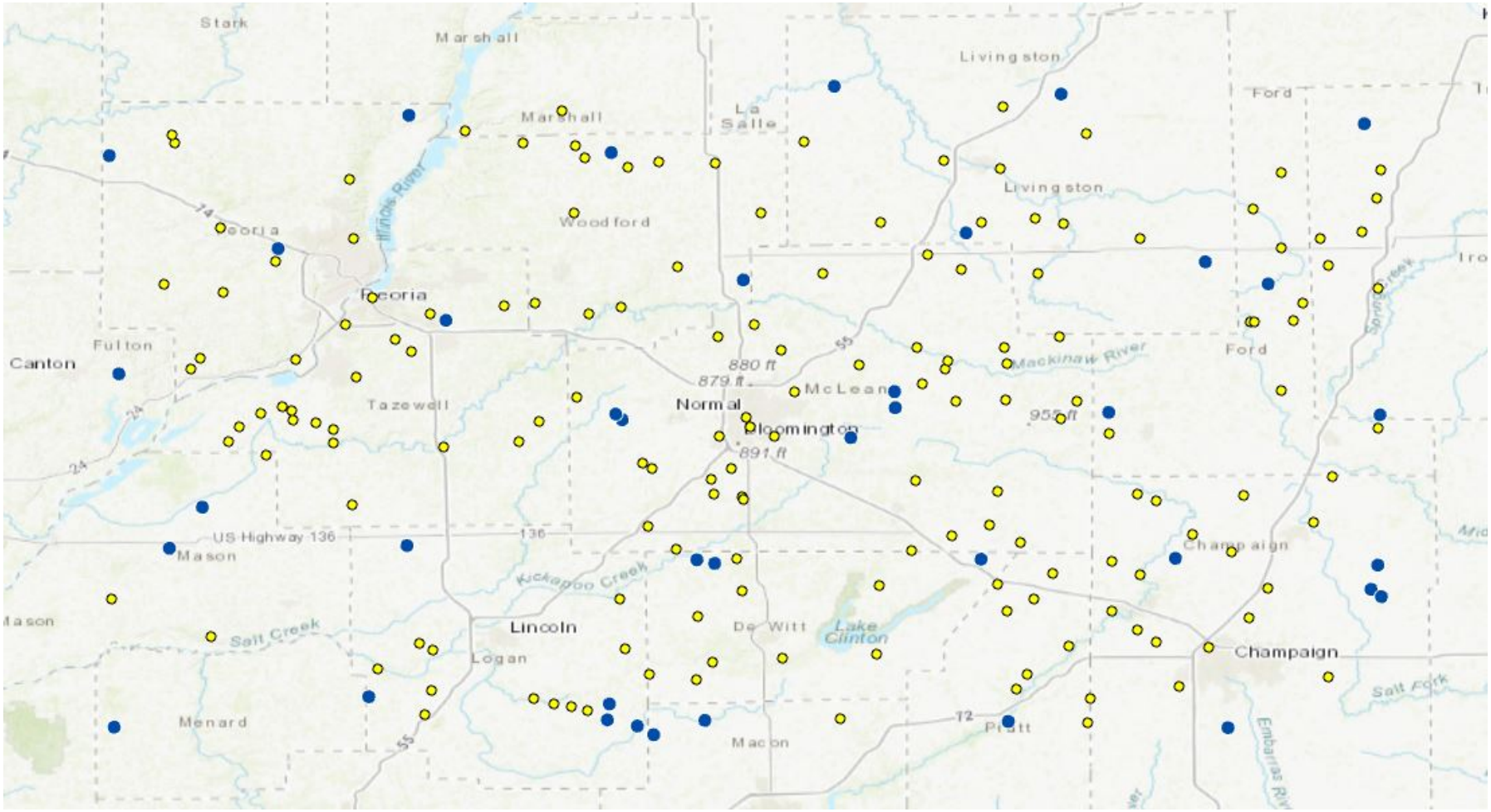

Stark
Marshall
Livingston
Peoria
Fulton
Canton
Tazewell
Normal
McLean
Bloomington
Mason
Lincoln
Logan
Champaign
Piatt
Macon
DeWitt
Lake Clinton
Salt Creek
Kickapoo Creek
US Highway 136
880 ft
955 ft

Examples

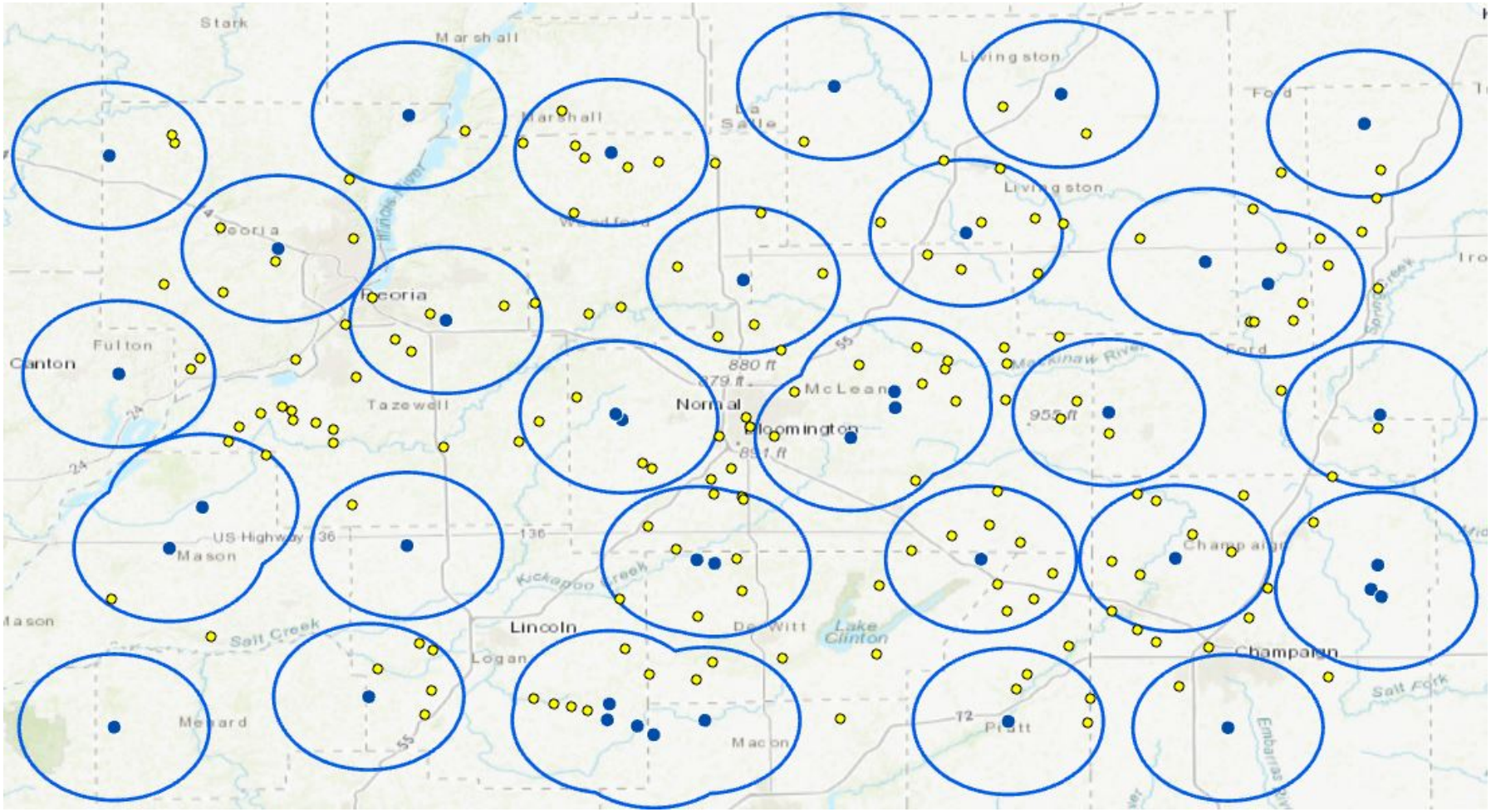
Can more points be added, given they satisfy the distance constraints?

```
. clustermax lat lon, gen(cid) n(1) between(25)
within(15) seed(1)
```

Examples


Stark
Marshall
Livingston
Ford
La Salle
Peoria
Woodford
Fulton
Canton
Tazewell
Normal
McLean
Bloomington
880 ft
879 ft
891 ft
955 ft
Mackinaw River
US Highway 136
Mason
Champaign
Kickapoo Creek
Salt Creek
Lincoln
Logan
De Witt
Lake Clinton
Menard
Macon
Piatt
Champaign
Salt Fork

Examples

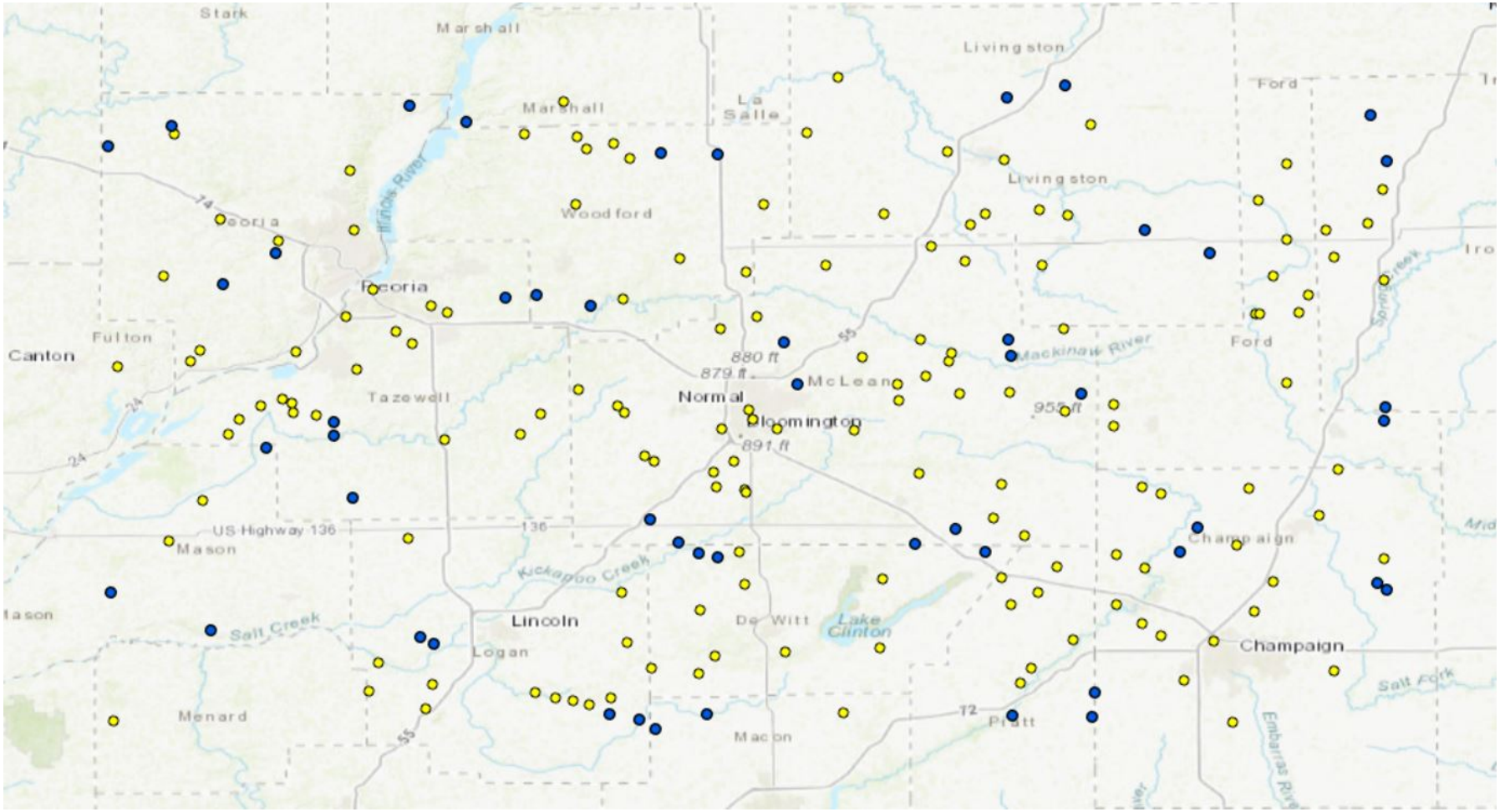

Stark
Marshall
Livingston
Peoria
Fulton
Canton
Tazewell
Normal
McLean
Bloomington
Mason
US Highway 136
Lincoln
Logan
Salt Creek
Lake Clinton
DeWitt
Champaign
Salt Fork
Macon
Piatt
Menard
880 ft
955 ft

Examples

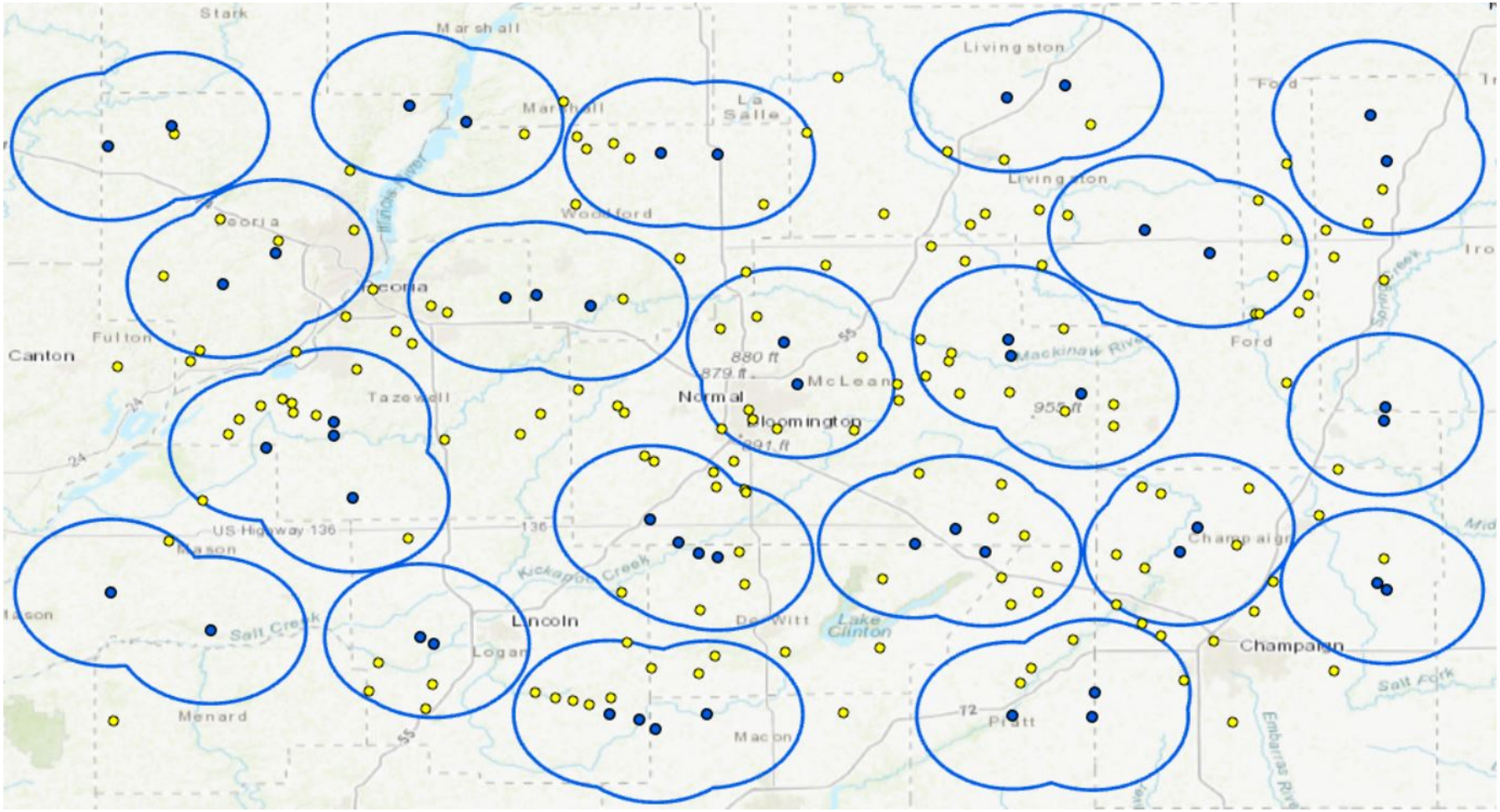
Two points per cluster:

```
. clustermax lat lon, gen(cid) n(2) between(25)
within(15) seed(1)
```

Examples


Stark
Marshall
Marshall
La Salle
Livingston
Ford
Livingston
Woodford
Peoria
Peoria
Fulton
Canton
Tazewell
880 ft
879 ft
McLean
Normal
Bloomington
891 ft
955 ft
Mackinaw River
Ford
US Highway 136
Mason
136
Champaign
Kickapoo Creek
Lincoln
De Witt
Lake Clinton
Champaign
Salt Creek
Logan
Salt Fork
Menard
Macon
Piatt
72
55

Examples

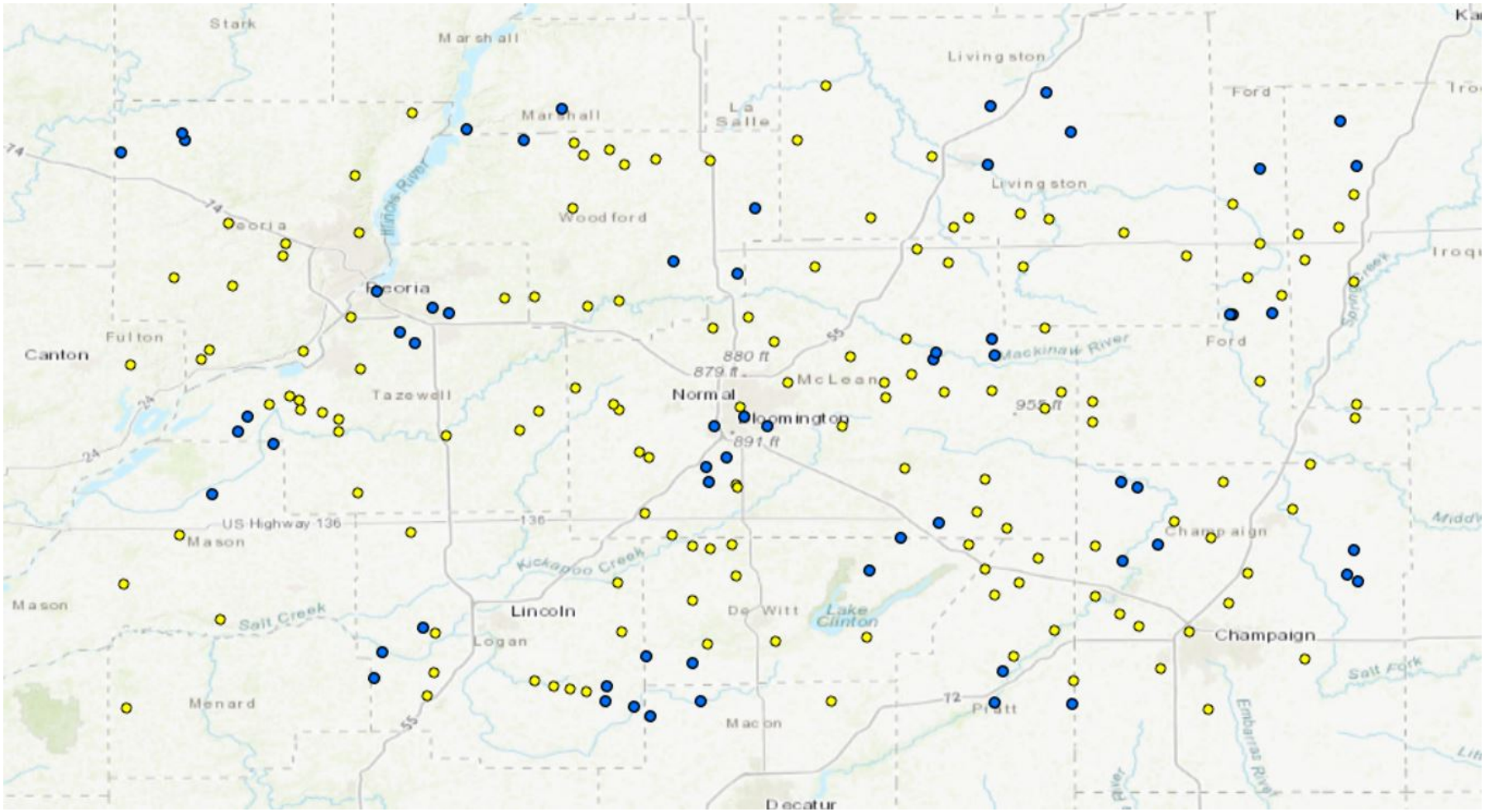

Stark
Marshall
Livingston
La Salle
Peoria
Fulton
Canton
Tazewell
Normal
McLean
Bloomington
880 ft
879 ft
991 ft
955 ft
Ford
Champaign
US Highway 136
Mason
Salt Creek
Kickapoo Creek
Lincoln
Logan
DeWitt
Lake Clinton
Champaign
Salt Fork
Menard
Macon
Piatt
72
55
136
24

Examples

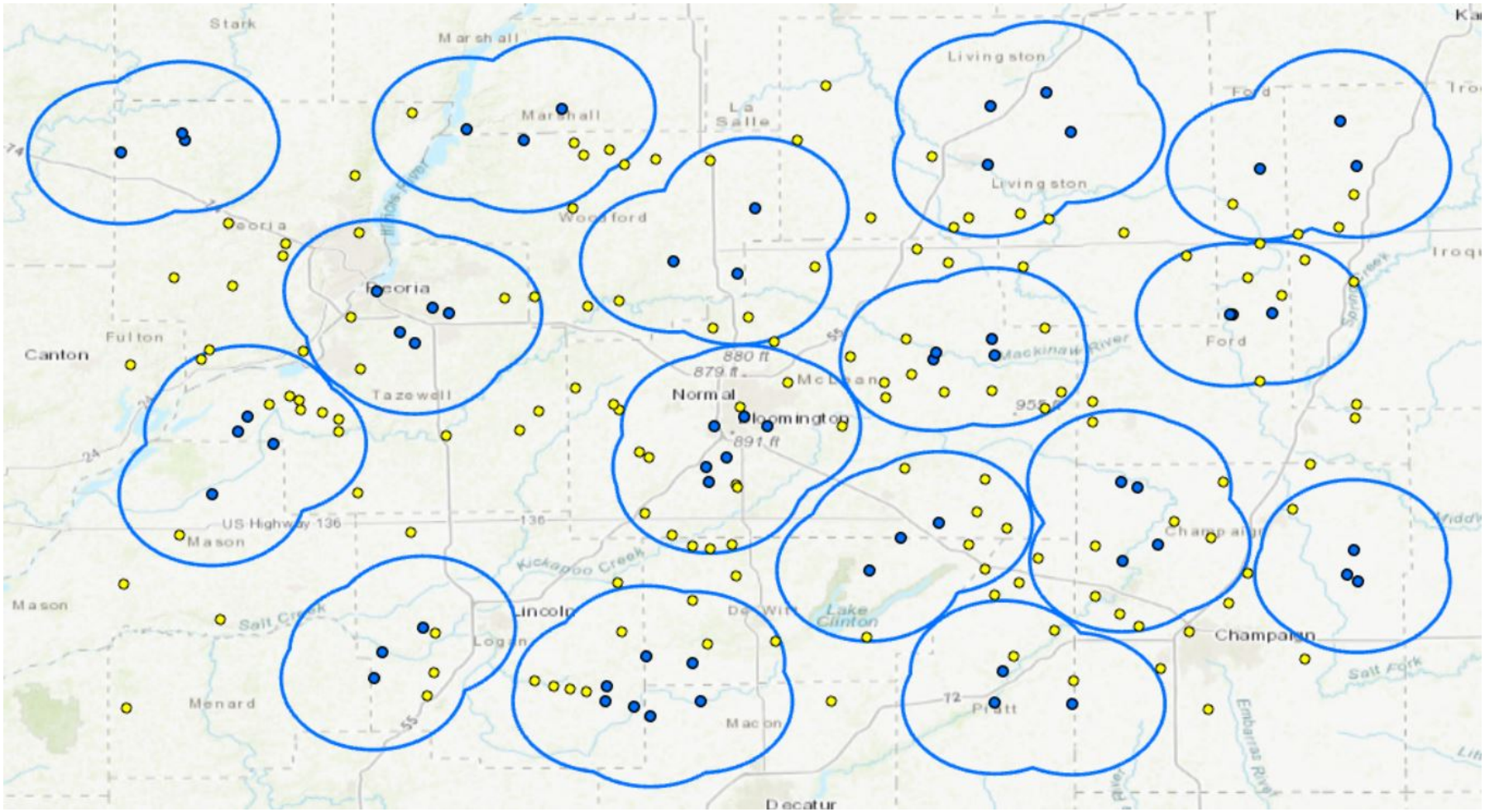
Three points per cluster:

```
. clustermax lat lon, gen(cid) n(3) between(25)
within(15) seed(1)
```

Examples


Stark
Marshall
Livingston
Ford
La Salle
Peoria
Woodford
Fulton
Canton
Tazewell
McLean
Normal
Bloomington
880 ft
879 ft
891 ft
955 ft
Mackinaw River
US Highway 136
Mason
Champaign
Kickapoo Creek
Lincoln
De Witt
Lake Clinton
Logan
Salt Creek
Menard
Macon
Piatt
Decatur

Examples


Stark
Marshall
Livingston
La Salle
Peoria
Woodford
Fulton
Canton
Tazewell
Normal
Bloomington
McLean
Ford
Mason
US Highway 136
Kickapoo Creek
Lincoln
Logan
Salt Creek
Lake Clinton
Champaign
Menard
Macon
Piatt
Decatur

Overview

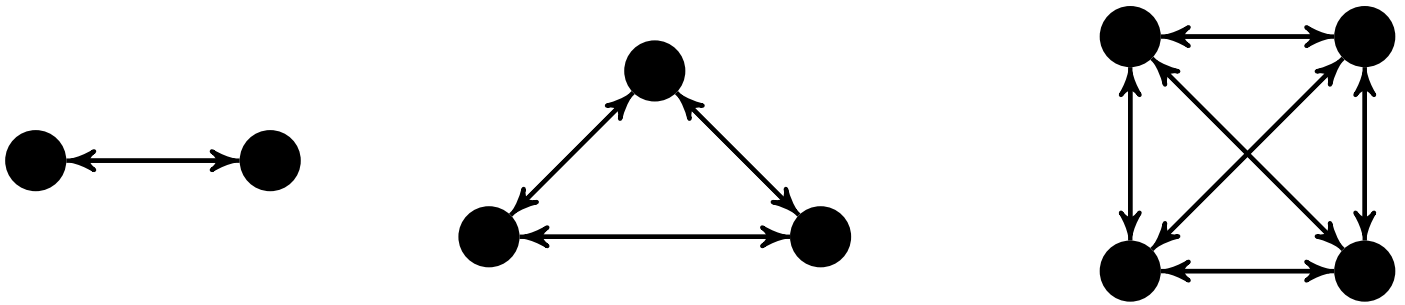
The Algorithm

1. Find groups of *n*() points mutually in *within*() distance
2. Resolve *between*() cluster conflicts
3. Join additional points

The Algorithm

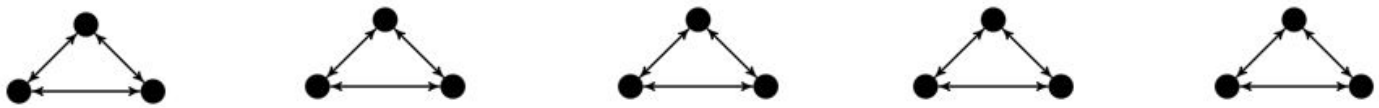
1. Find groups of *n*() points mutually in *within*() distance
2. Resolve *between*() cluster conflicts
3. Join additional points

The Algorithm



The Algorithm

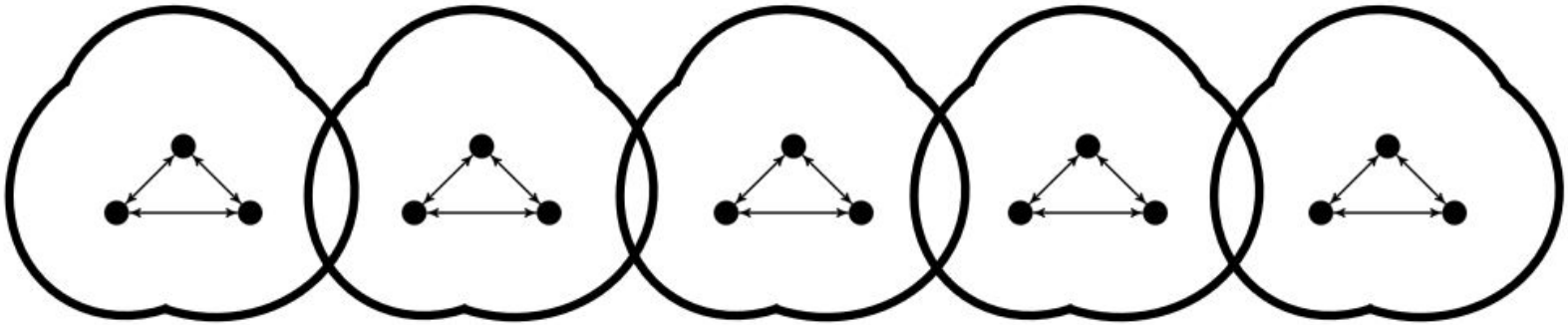
The Algorithm



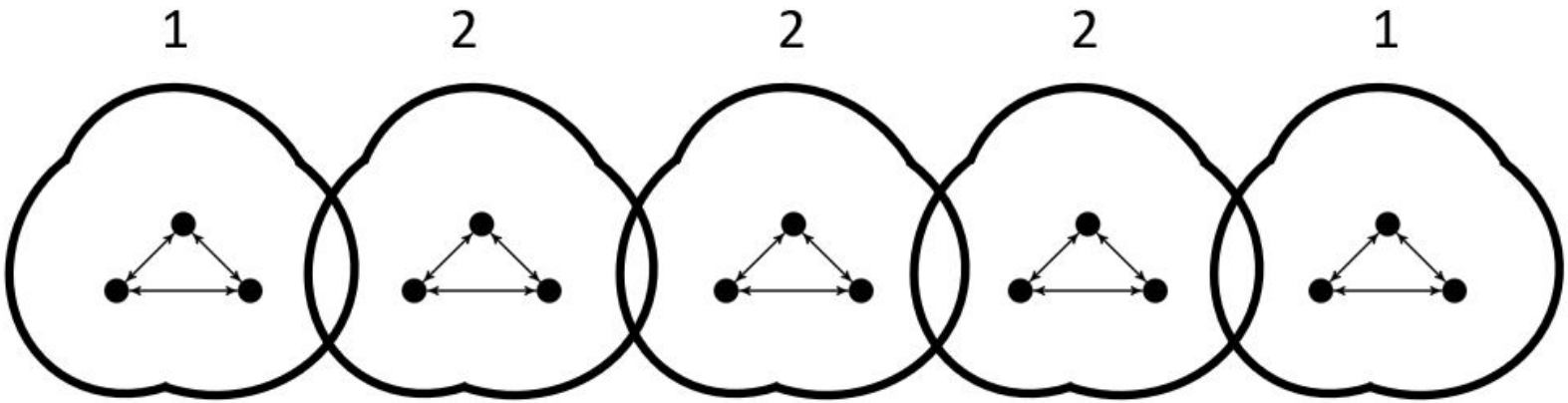
The Algorithm

1. Find groups of *n*() points mutually in *within*() distance
2. Resolve *between*() cluster conflicts
3. Join additional points

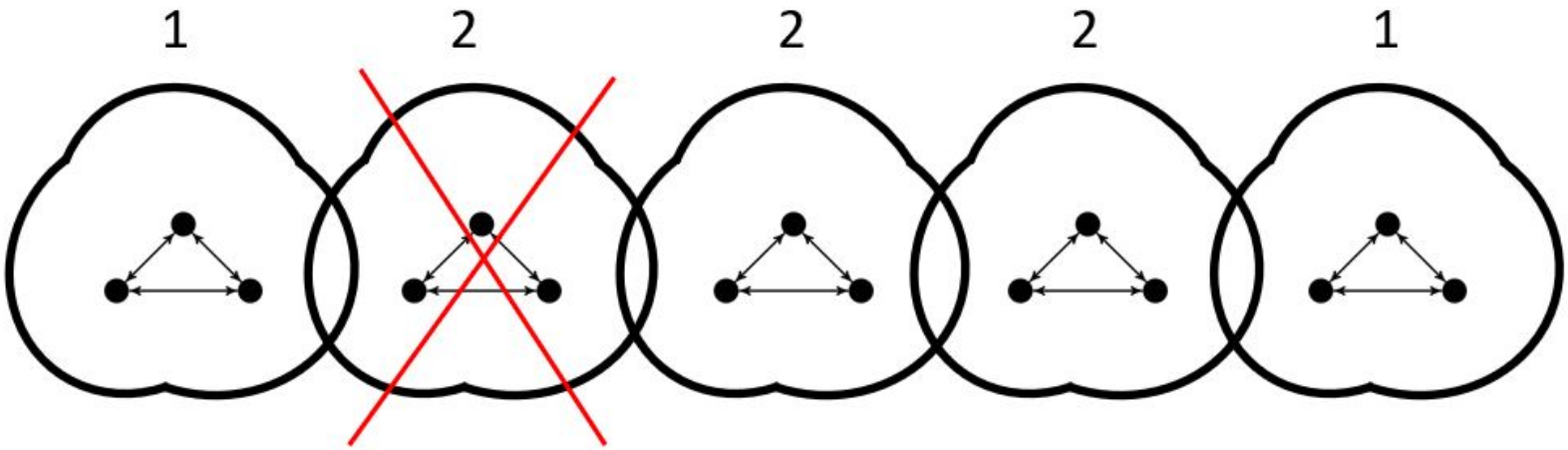
The Algorithm



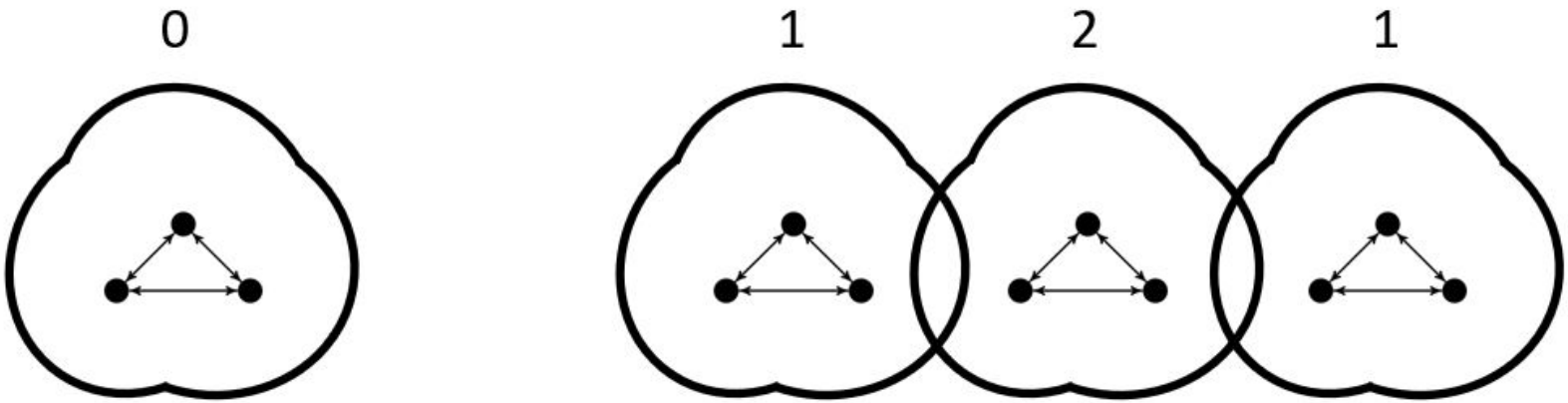
The Algorithm



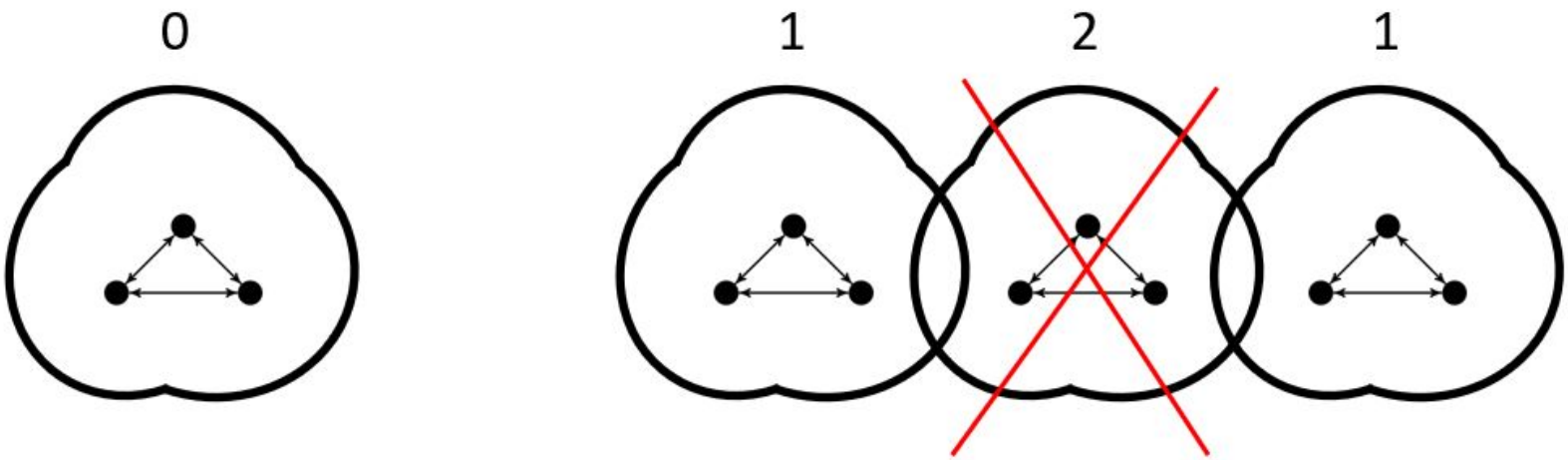
The Algorithm



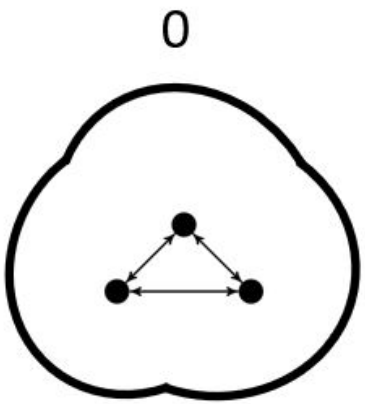
The Algorithm

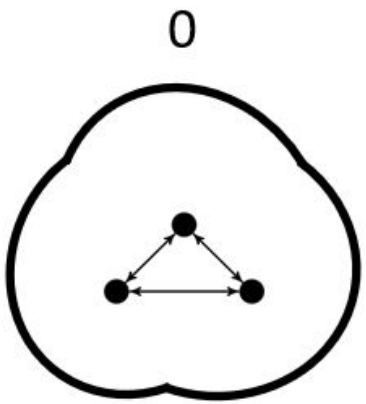


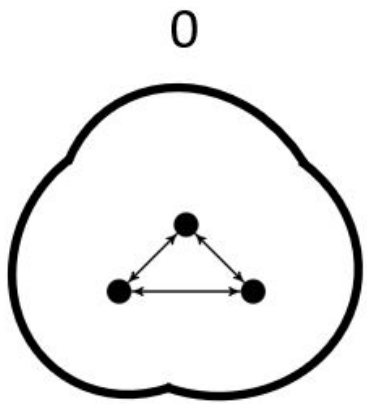
The Algorithm



The Algorithm







The Algorithm

1. Find groups of *n*() points mutually in *within*() distance
2. Resolve *between*() cluster conflicts
3. Join additional points

Overview

Further Improvements

Additional options

- Add option to import own distance matrix (travel time, social distance, etc.)
- Add within-cluster stratification conditions (at least 1 male and 1 female, etc.)

Download and Related Literature

Stata ado-file and helpfile:

```
https://github.com/fmbarba
```

Related Literature:

Jade Benjamin-Chung, Benjamin F Arnold, David Berger, Stephen P Luby, Edward Miguel, John M Colford Jr, Alan E Hubbard (2018)

Spillover effects in epidemiology: parameters, study designs and methodological considerations

International Journal of Epidemiology, 2018, 332-347