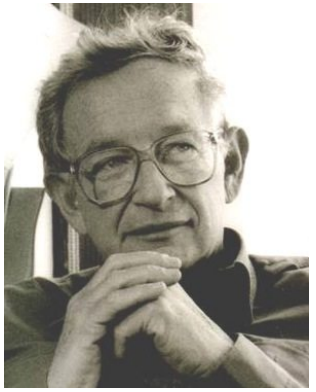
Structure in Complex Systems

Mark Newman

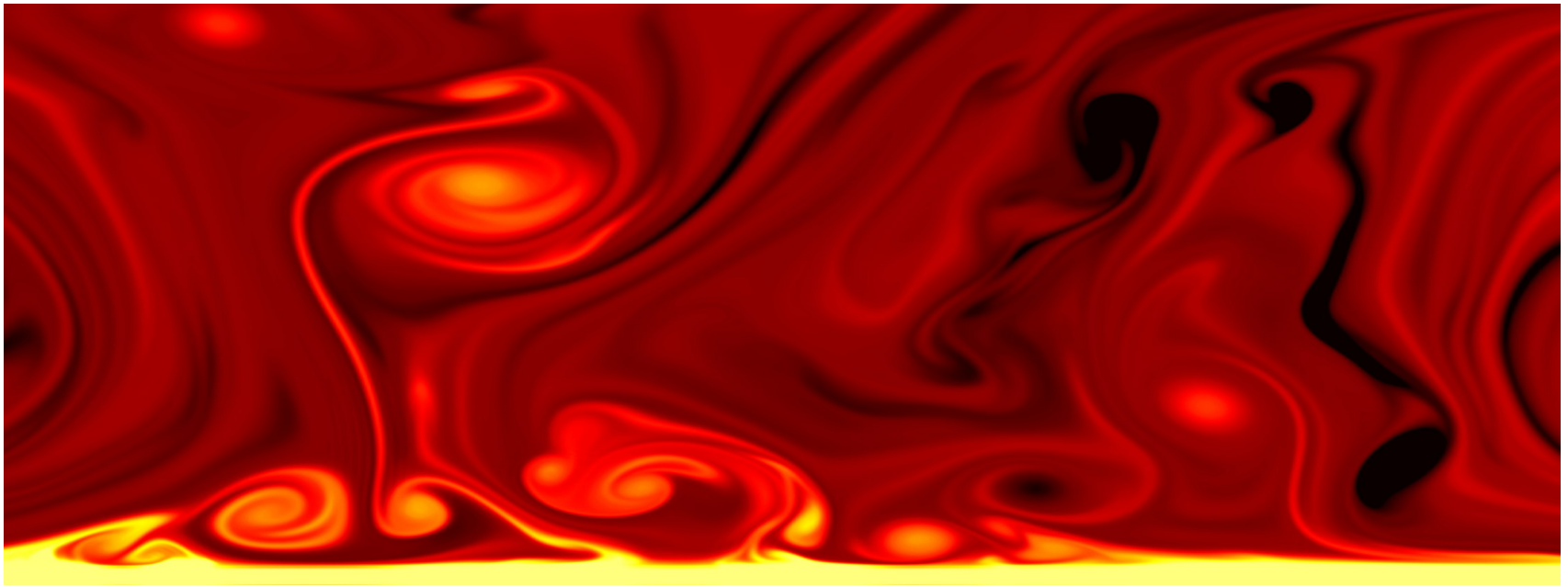
Department of Physics, University of Michigan
and
Rudolph Peierls Centre for Theoretical Physics

Complex Systems

- Systems of many similar parts
  - Atoms, molecules, cells, people, animals, computers, cars, trees, companies
- The behaviour of the whole is not just the sum of the parts
  - “Emergence”
- New and simple laws “emerge” from the combined behaviours of many parts
  - “More is different”, P. W. Anderson, 1972



Complex Systems



System	Emergent behaviours
Condensed matter	Solidity, conduction, sound
Fluid	Flow, turbulence
Ecosystem	Evolution
Market	Price setting, bubbles, crashes
Road traffic	Prevailing speed, traffic jams

Simple vs. complex models

Simple
Imparts understanding
Not quantitatively accurate

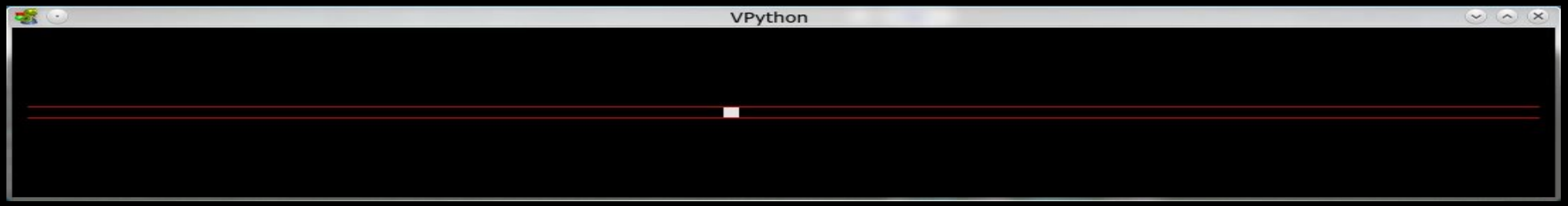
Complex
Little understanding
Good quantitative accuracy

- Two styles of simulations:
  - Simple models that teach us how the world works
  - Complex ones that make predictions
- Only in physics do models do both
- The "unreasonable effectiveness of mathematics" (Eugene Wigner)

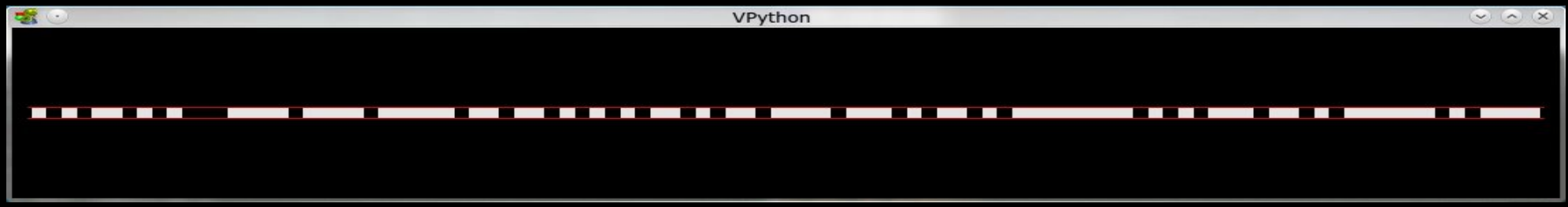
Traffic

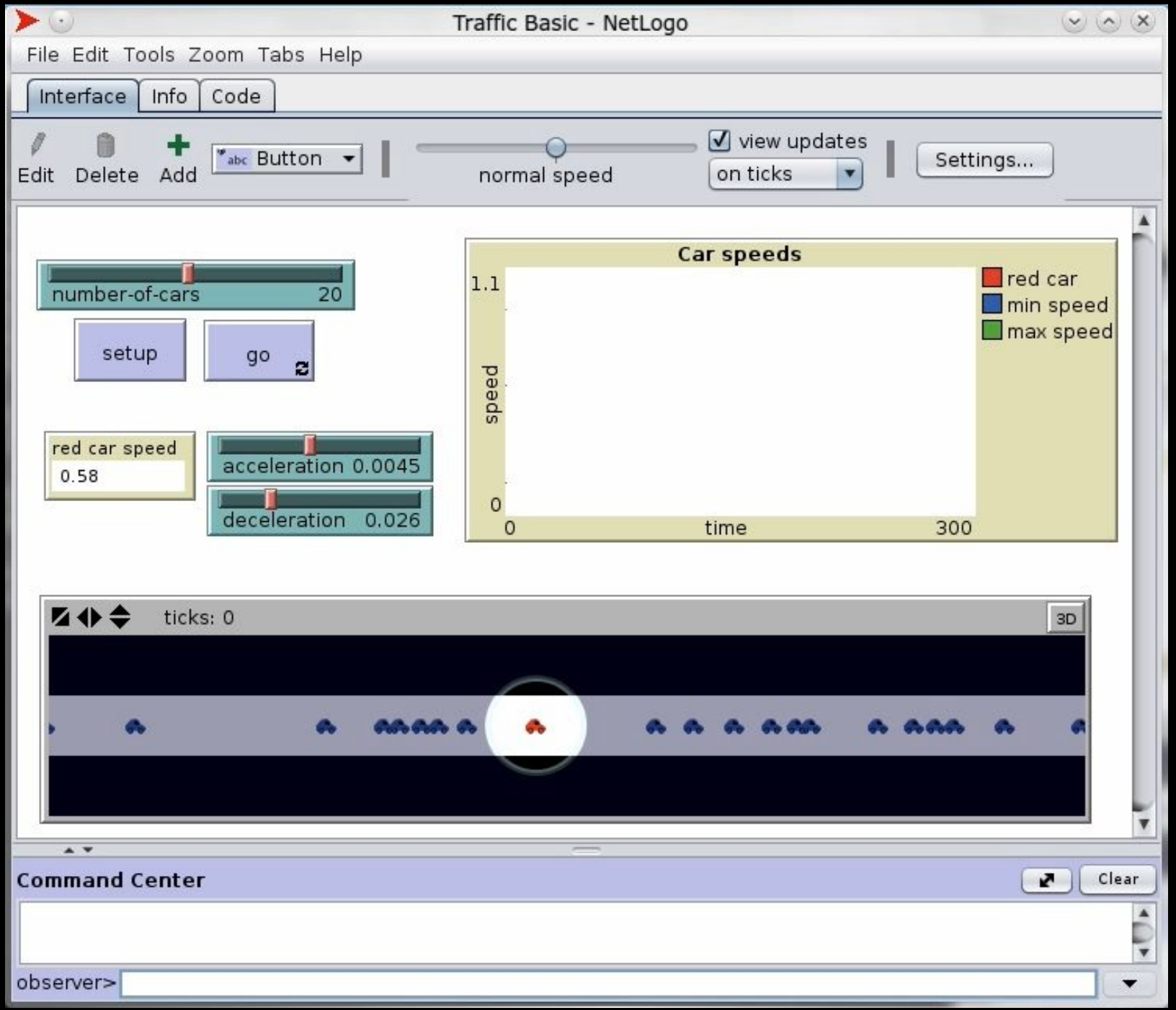
- The Nagel-Schreckenberg model:

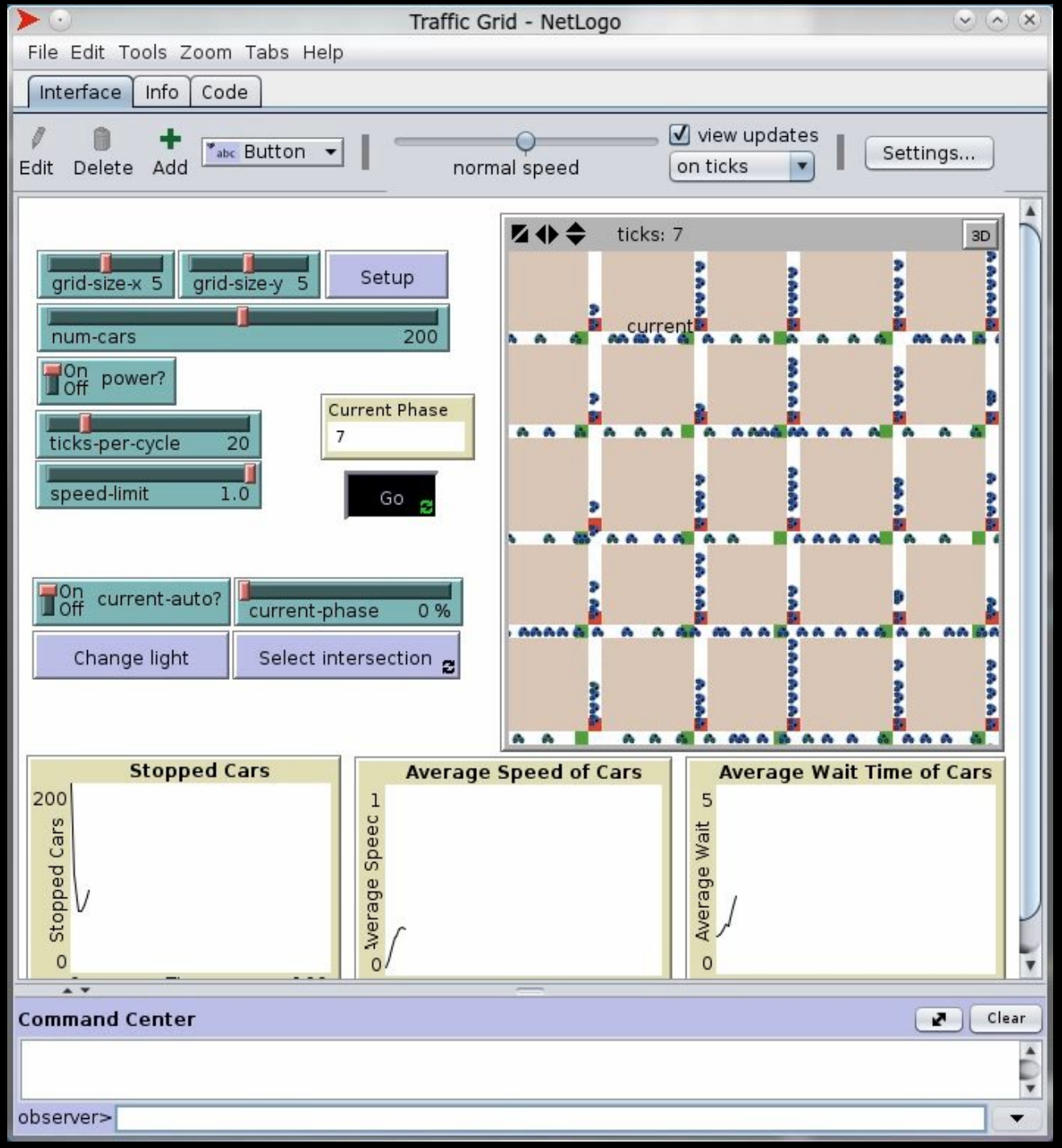
- This is a *cellular automaton*
- Highly unrealistic:
  - Cars sit in squares
  - They only go at one speed
  - They stop dead in their tracks


VPython


VPython


VPython


Traffic Basic - NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code
Edit Delete Add
Button
normal speed
view updates
on ticks
Settings...
number-of-cars 20
setup
go
red car speed
0.58
acceleration 0.0045
deceleration 0.026
Car speeds
1.1
0
speed
0
time
300
red car
min speed
max speed
ticks: 0
3D
Command Center
Clear
observer>


Traffic Grid - NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code
Edit Delete Add Button
normal speed
view updates
on ticks
Settings...
grid-size-x 5
grid-size-y 5
Setup
num-cars 200
On Off power?
ticks-per-cycle 20
speed-limit 1.0
Current Phase
7
Go
On Off current-auto?
current-phase 0 %
Change light
Select intersection
ticks: 7
3D
current
Stopped Cars
200
Stopped Cars
0
Average Speed of Cars
1
Average Speec
0
Average Wait Time of Cars
5
Average Wait
0
Command Center
Clear
observer>

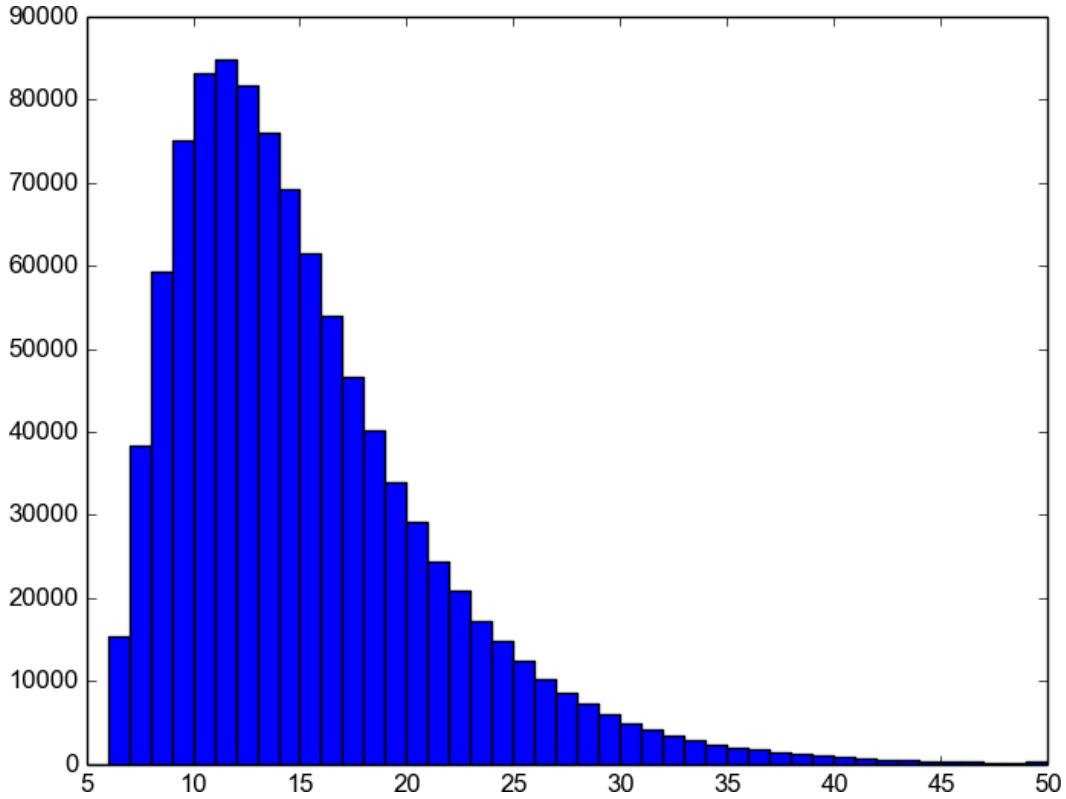
Monte Carlo simulation

- How many times will we have to roll a die before we get a 1?
  - Answer: *6 times on average*
- How many times will we have to roll a die before we get *every* number at least once?
  - Answer: *Um. . .*
- We can answer this question by *Monte Carlo simulation*

Program `dice.py`:

```
from random import randrange
s = {1,2,3,4,5,6}
m = 0
while s:
    n = randrange(6) + 1        # Roll a number
    s.discard(n)                # Remove from set
    m += 1
print(m)
```

```
% dice.py
14
% dice.py
18
% dice.py
11
% dice.py
16
% dice.py
19
% dice.py
8
% dice.py
8
% dice.py
10
% dice.py
19
```



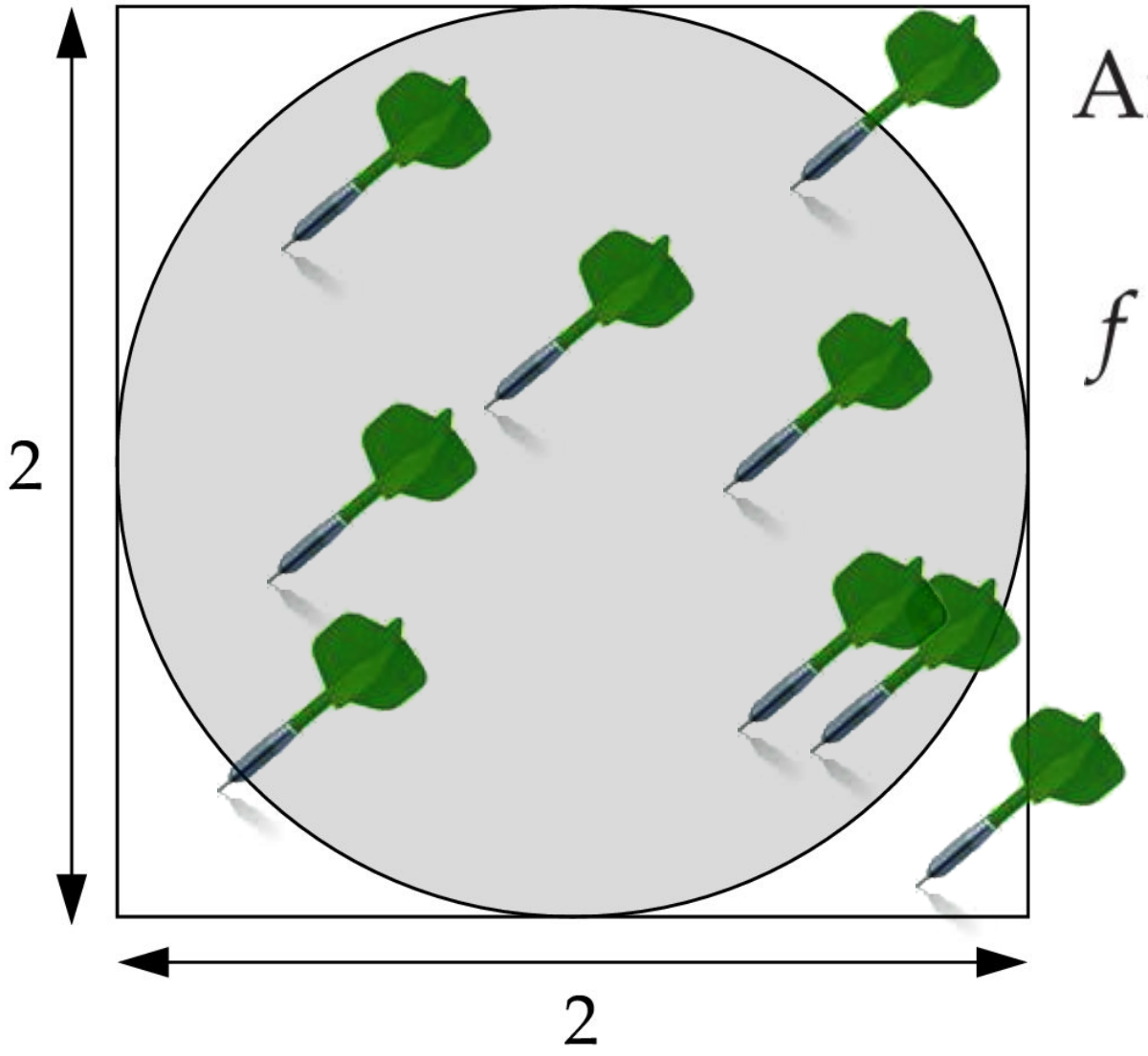
Monte Carlo simulation

- But this question also has an exact answer, which you can calculate using the tools of probability theory:

$$\sum_{k=1}^{6} \frac{6}{k} = \frac{147}{10} = 14.7$$

- So we have a problem that is fundamentally random, but also has an exact answer
- In the 1940s, Stanislaw Ulam and collaborators showed you could use this trick *in reverse* to solve all sorts of physics problems

Example: Calculating π



$$\text{Area, } A = \pi r^2 = \pi$$

$$f = \frac{\text{Area of circle}}{\text{Area of square}}$$

$$= \frac{\pi}{4}$$

```
from random import random
N = 1000000
hits = 0
for k in range(N):
    x = 2*random() - 1
    y = 2*random() - 1
    if x**2+y**2<1: hits += 1
print(4*hits/N)
```

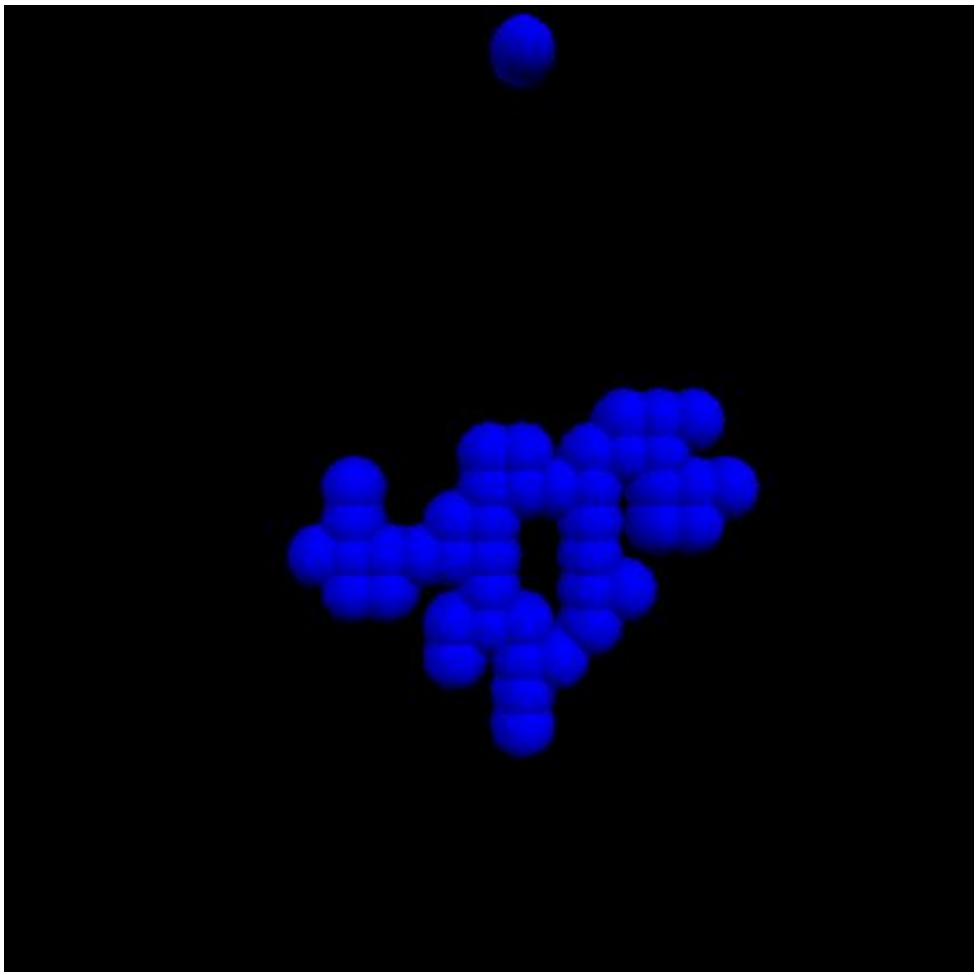
```
% pi.py
3.140852
% pi.py
3.141904
% pi.py
3.144476
% pi.py
3.1431
% pi.py
3.142596
% pi.py
3.139916
% pi.py
3.141704
```

A variant of this experiment, “Buffon's needle”, performed without the aid of a computer by dropping needles on a marked board, was a popular pastime among the 19th century intelligencia.

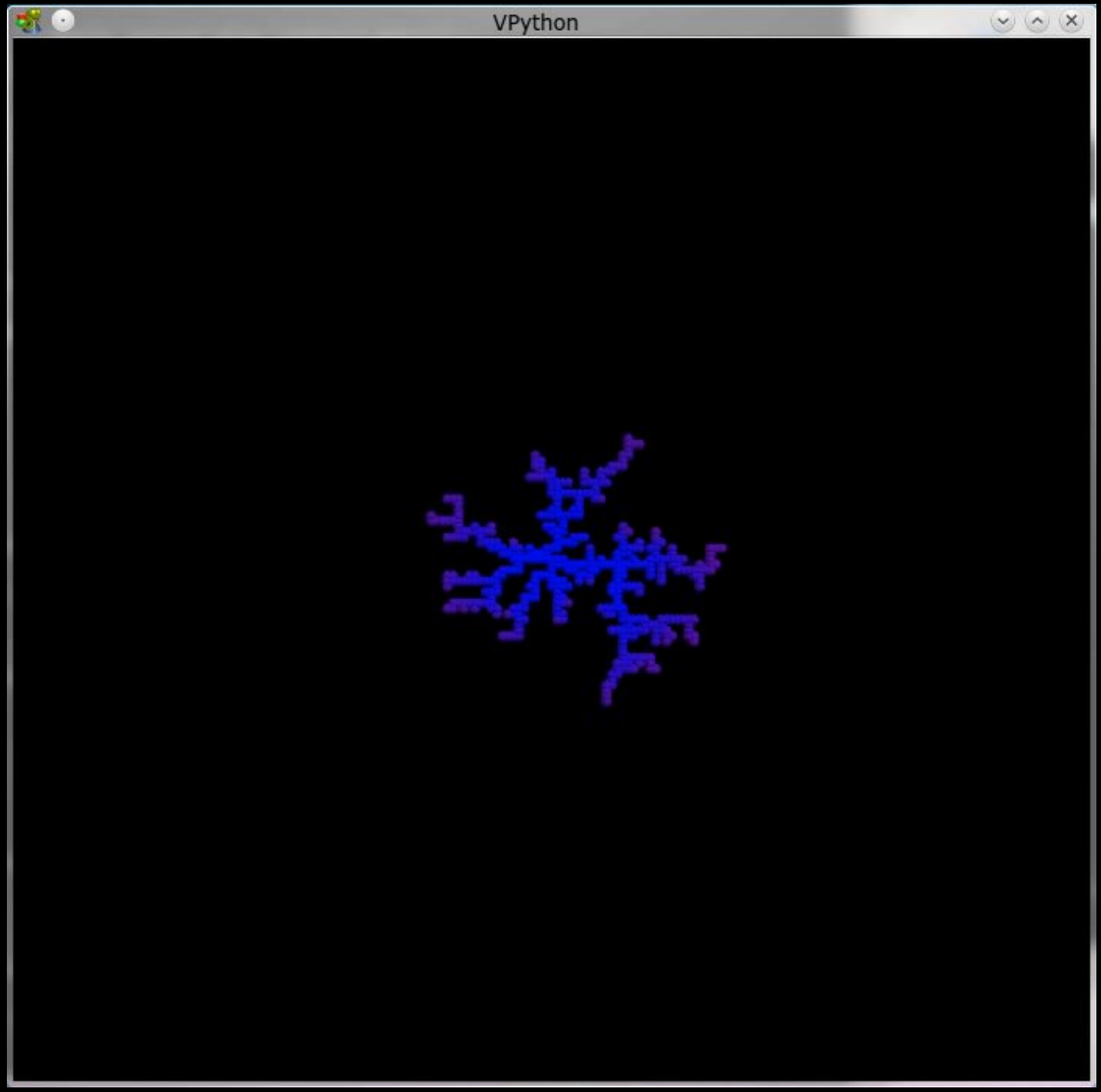
A particularly famous instance is the experiment by Mario Lazzarini in 1901, in which he dropped 3408 needles and calculated a value of π accurate to 6 decimal places.

Diffusion-limited aggregation

- Particles diffuse around until they stick to one another



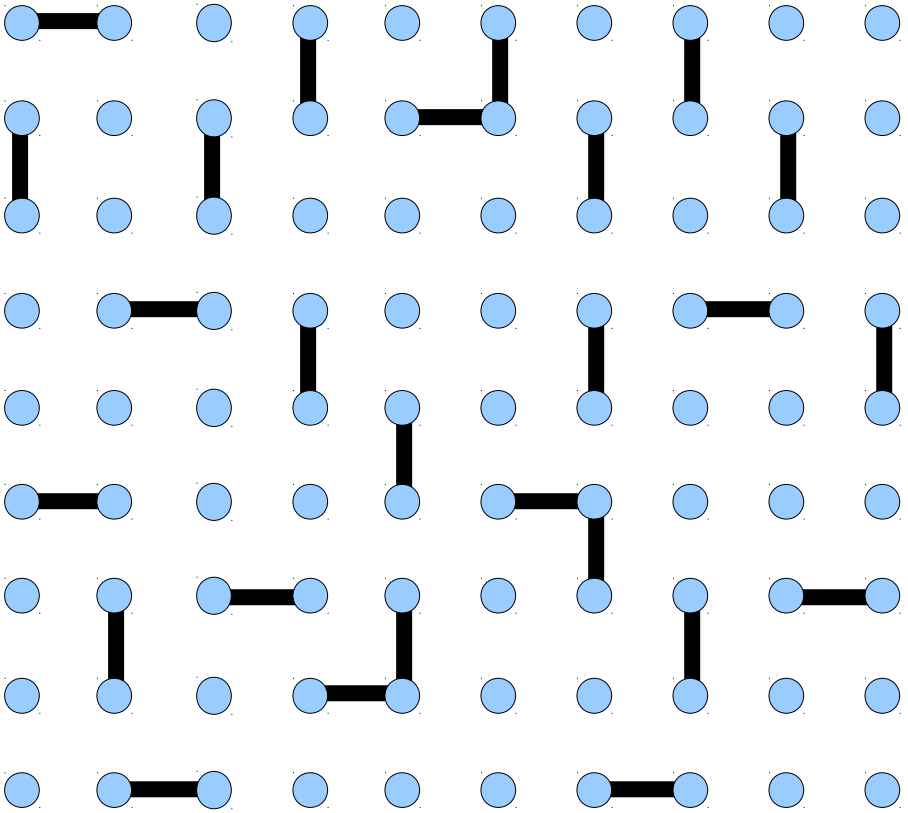
```
while maxr<L:

    # New particle
    theta = 2*pi*random.random()
    x = int((maxr+1)*cos(theta)+0.5)
    y = int((maxr+1)*sin(theta)+0.5)
    t += 1
    col = sin(colscale*t)**2

    # Start walking
    while True:

        # Check if the particle has moved out of range
        if x*x+y*y>4*maxr*maxr:
            break

        # Check if the particle has reached another particle
        if anchored[x-1,y] or anchored[x+1,y] \
           or anchored[x,y-1] or anchored[x,y+1]:
           anchored[x,y] = 1
           sphere(pos=[x,y,0],radius=1,color=[col,0,1-col])
           r = sqrt(x*x+y*y)
           if r>maxr:
               maxr = r
           break

        # Move the particle
        if random.randrange(2):
            if random.randrange(2):
                x += 1
            else:
                x -= 1
        else:
            if random.randrange(2):
                y += 1
            else:
                y -= 1
```


VPython

Kaandorp et al. 2001

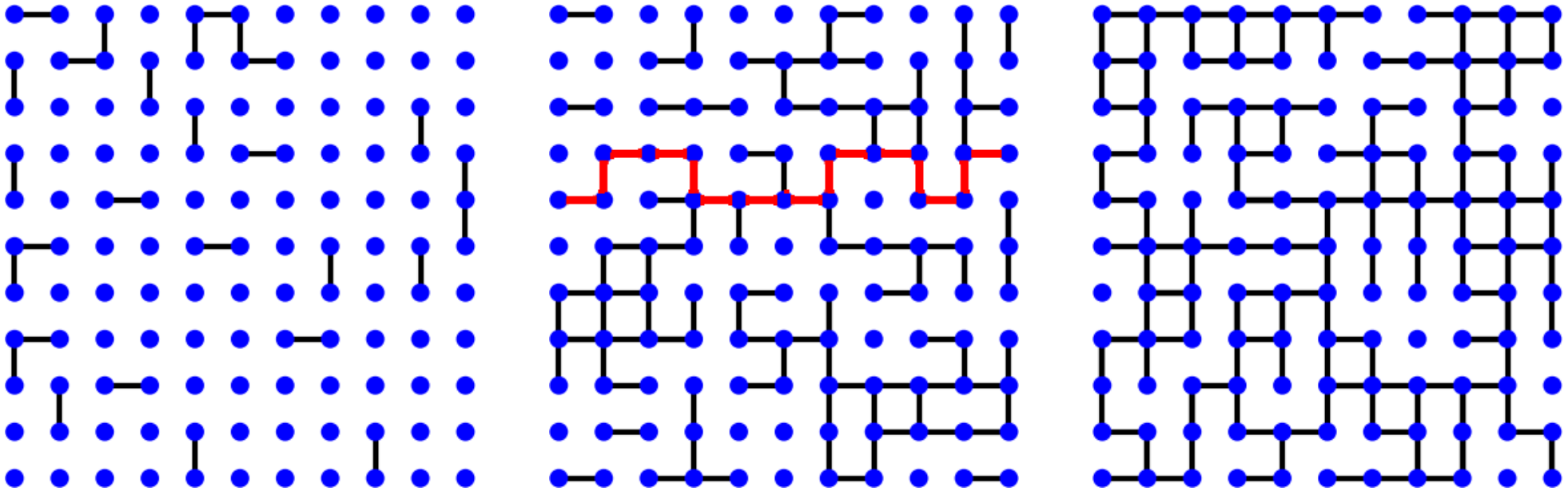
Percolation

- Add bonds to a lattice one by one, in random order:



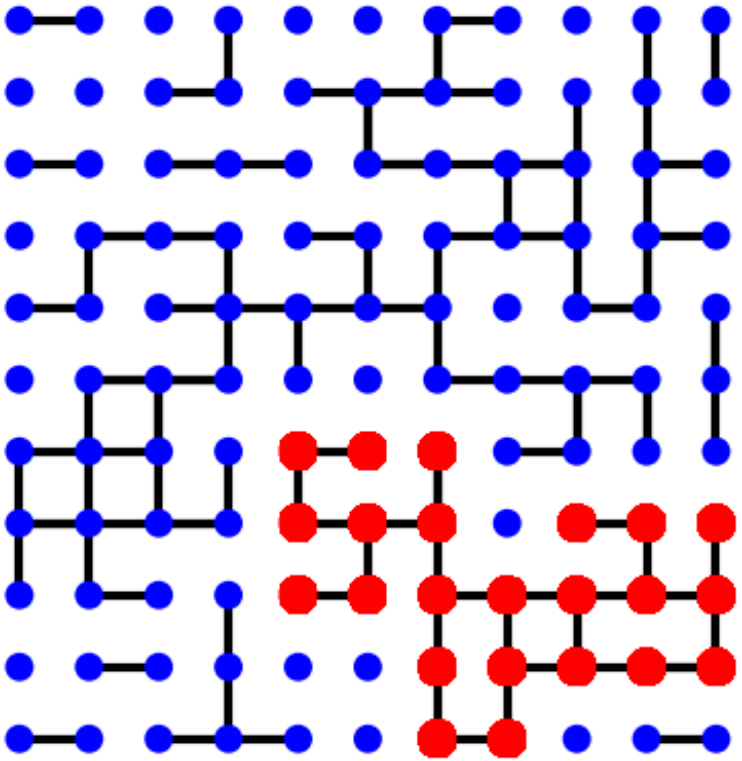
Percolation

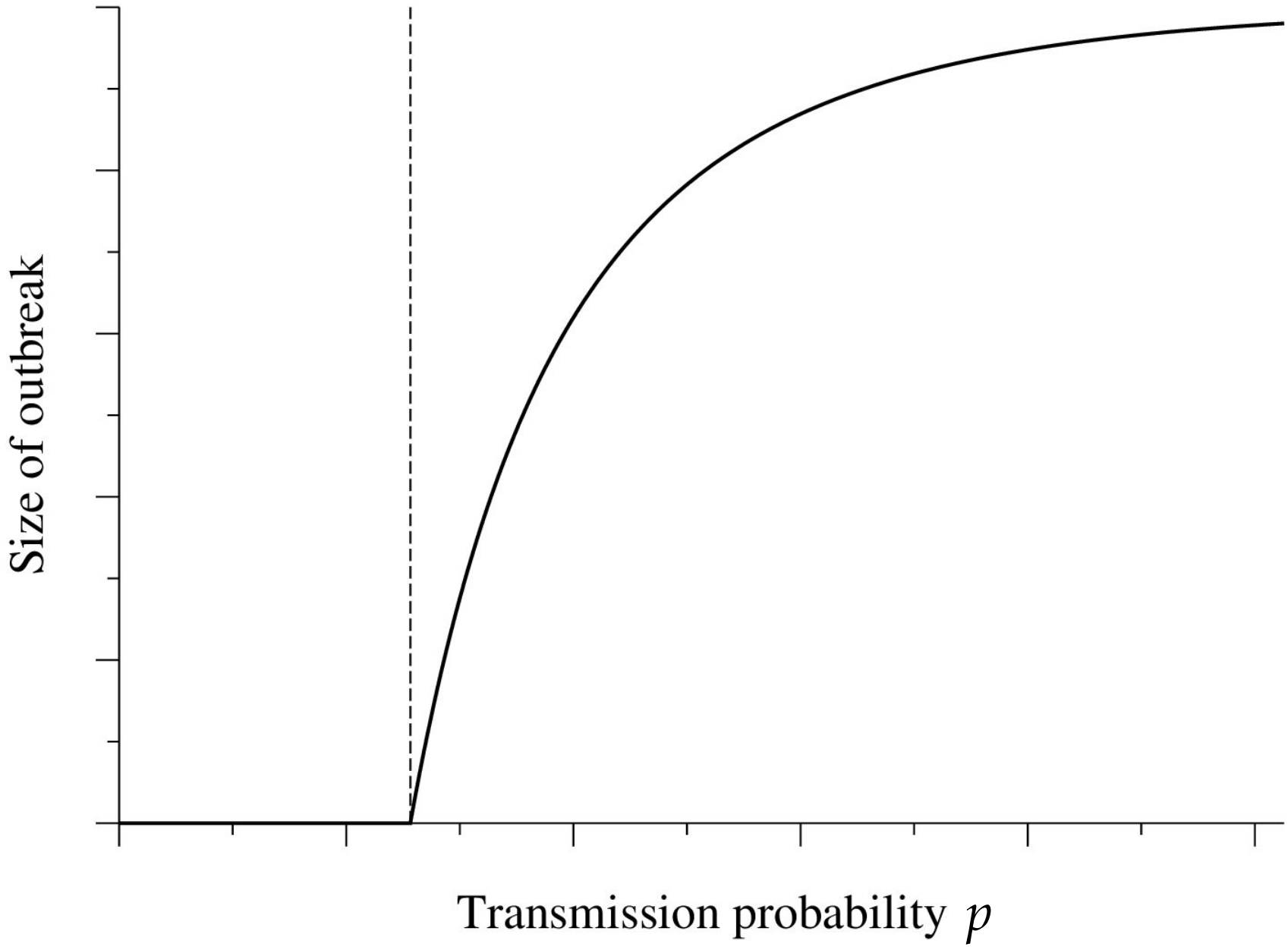
- Keep on going for long enough, and eventually a path forms across the lattice:



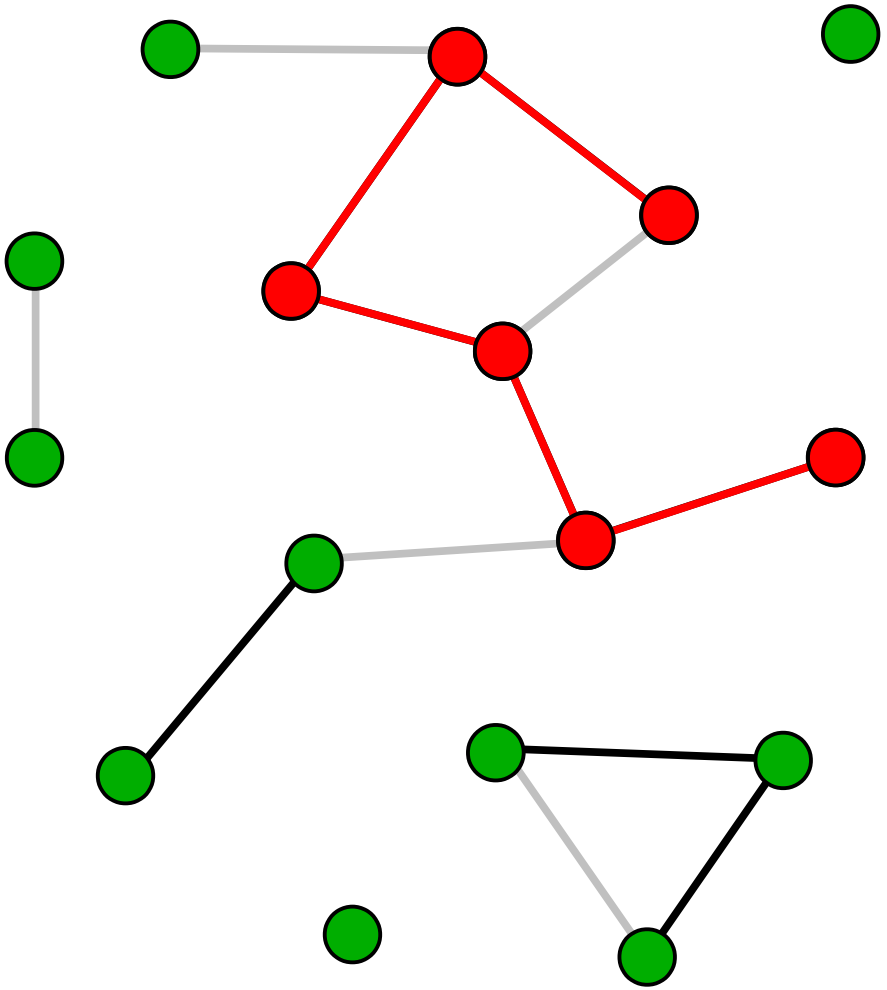
Percolation and epidemics

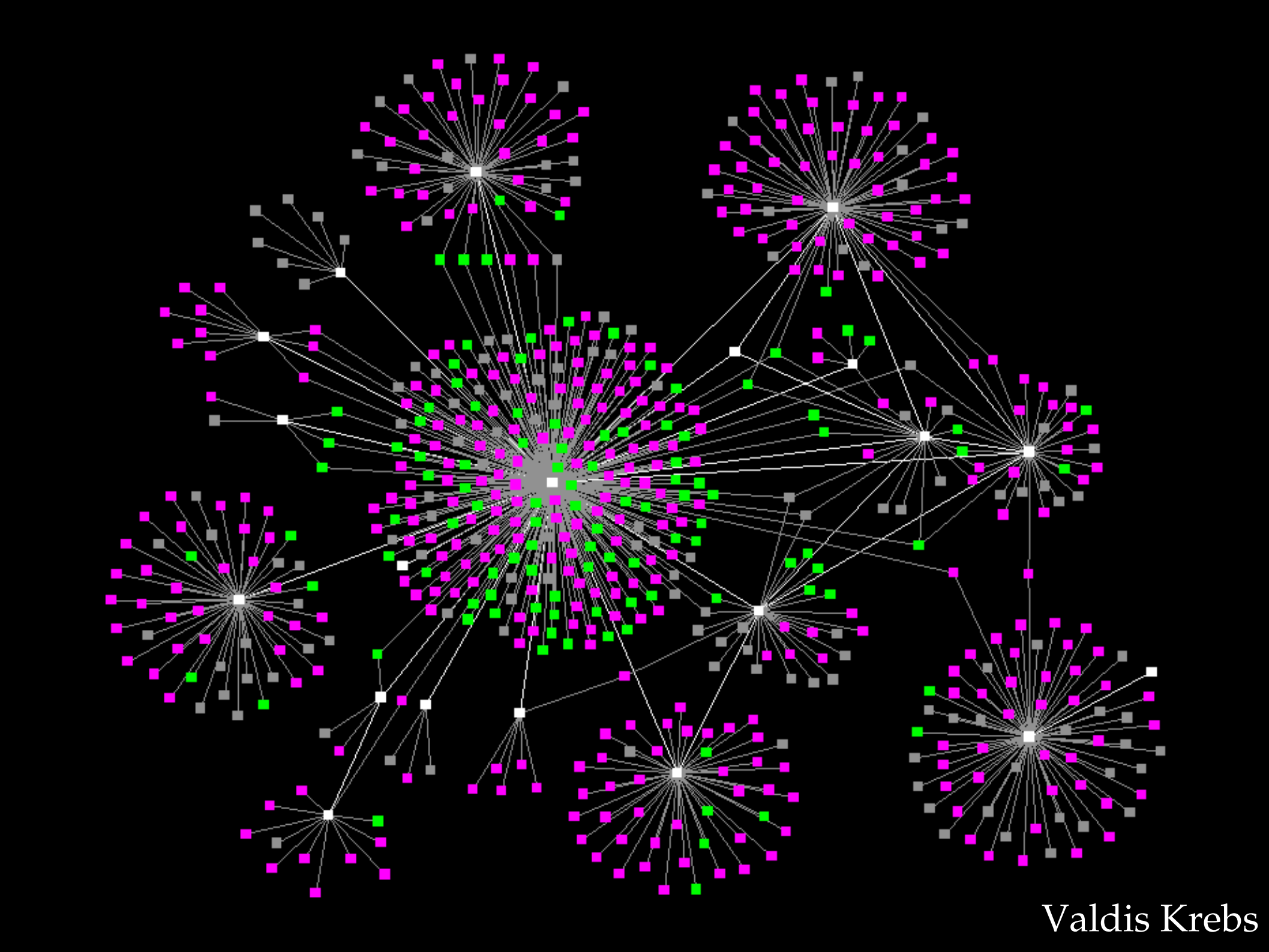
- Each site represents a person, who might have the flu
- If you have the flu, you pass it on to your friend with probability p

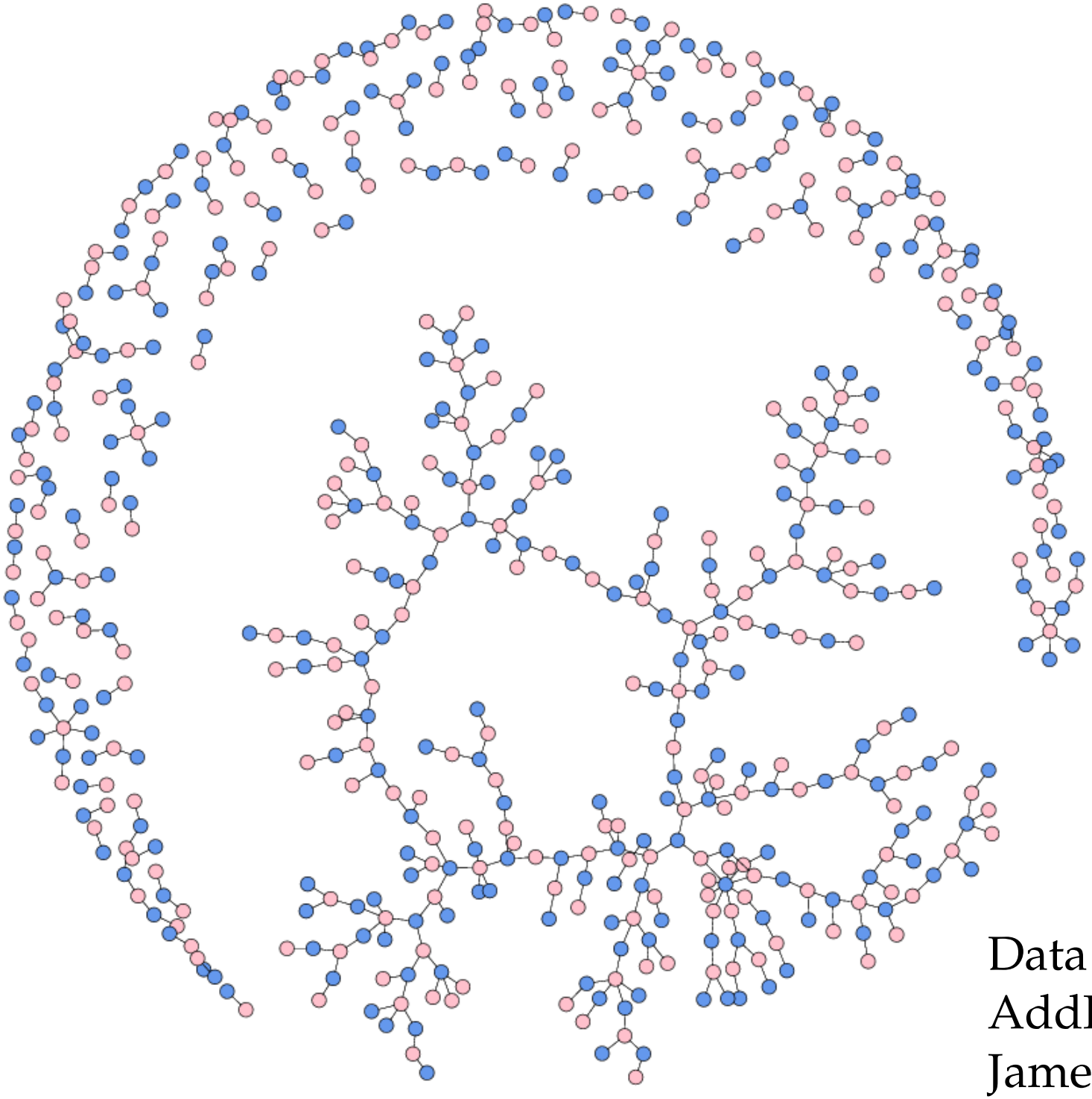



Size of outbreak
Transmission probability p

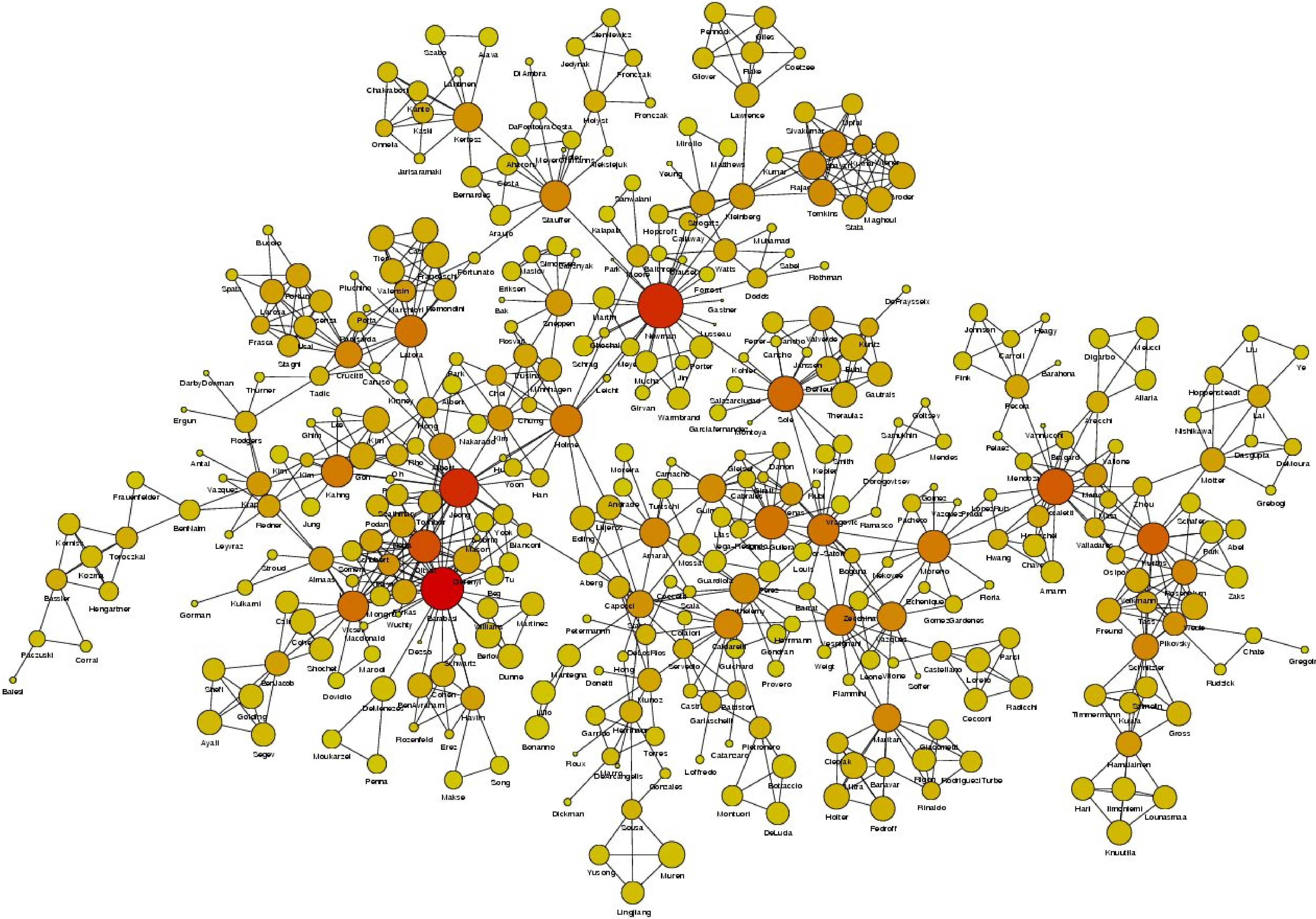
Spread of a disease



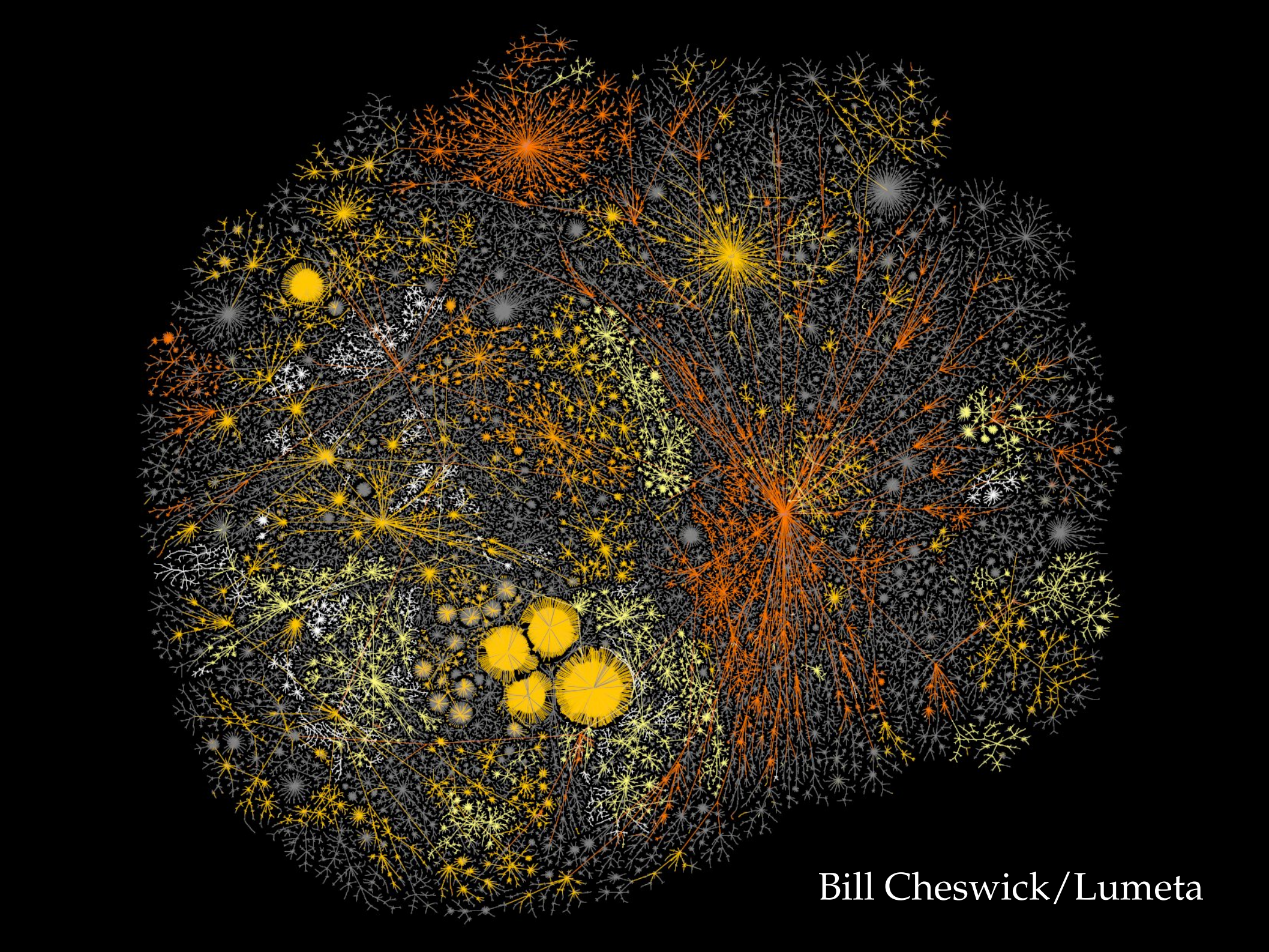

Valdis Krebs

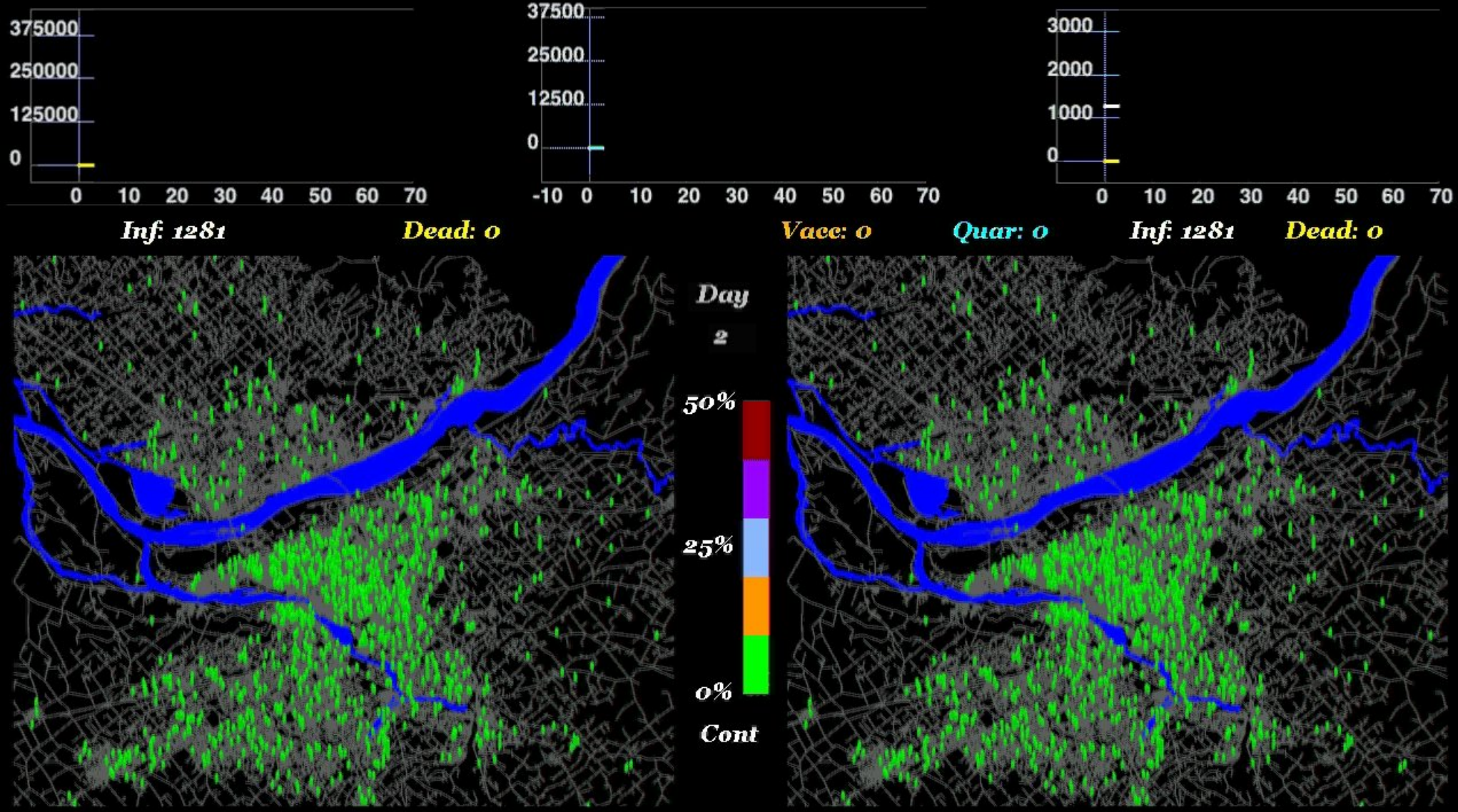


Data from
AddHealth via
James Moody



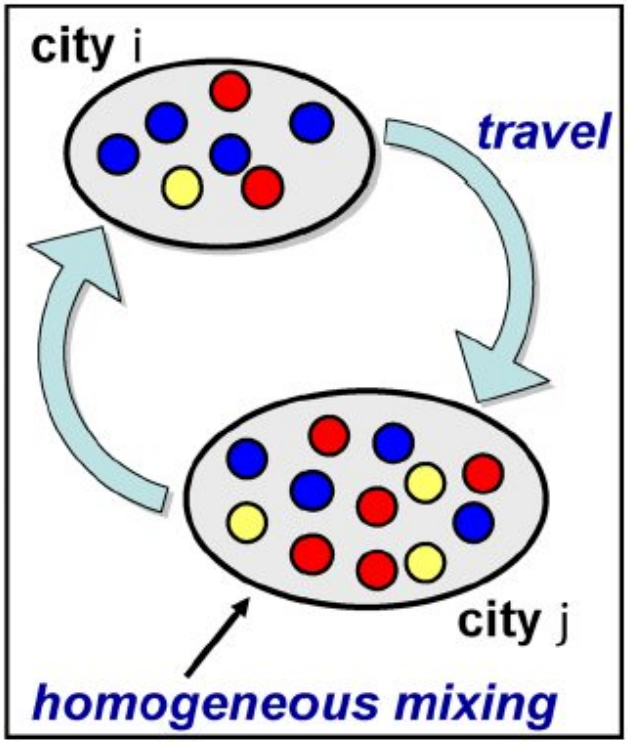
Szabo
Alava
Stenlewicz
Pennock
Giles
Glover
Flake
Coetzee
Lawrence
Jedynak
Fronczak
Di Ambra
Lahtinen
Chakraborti
Kanto
Kaski
Onnela
Jarisaramaki
Kertesz
Bernardes
Costa
Aharony
Araujo
DaFontouraCosta
MeyerOrtmanns
Holyst
Fronczak
Aleksiejuk
Stauffer
Mirollo
Matthews
Young
Sivakumar
Upfal
Kumar
Rajag
Broder
Tomkins
Stata
Maghoul
Sarwalani
Kalapala
Hopcroft
Strogatz
Callaway
Kleinberg
Muhamad
Watts
Sabel
Rothman
Dodds
Moore
Forrest
Gastner
Newman
Lusseau
Girvan
Warmbrand
Porter
Jin
Muchn
Schrag
Leicht
Martin
Bak
Erikson
Maslov
Sneppen
Rosvall
Trusina
Minnhagen
Chung
Holme
Buccio
Spata
Larosa
Frasca
Usai
Stagni
Pluchino
Valensin
Marchiori
Remondini
Fortunato
Porta
Bargsarda
Crucitti
Latora
Caruso
Tadic
Thurner
DarbyDowman
Ergun
Rodgers
Antal
Vazquez
Frauenfelder
BenNaim
Leyrwaz
Redner
Jung
Kahng
Goh
Kim
Park
Albert
Hong
Nakarado
Kinney
Jeong
Yoon
Han
Yook
Bianconi
Mason
Tu
Barabasi
Oltvai
Macdonald
Derenyi
Vicsek
Mongru
Wuchty
Almaas
Stroud
Kulkarni
Gorman
Kornis
Toroczkai
Kozma
Hengartner
Bassler
Paczuski
Corral
Balesi
Shochet
Dovidio
Marodi
Shefi
BenJacob
Golding
Ayali
Segev
Moukarzel
Penna
DeMenezes
BenAvraham
Cohen
Rozenfeld
Erez
Havlin
Makse
Song
Schwartz
Dezso
Williams
Berlow
Dunne
Martinez
Moreira
Camacho
Andrade
Lilljeros
Edling
Aberg
Amaral
Capocci
Colaiori
Servedio
Caldarelli
DeLosRios
Mossa
Guardiola
Guimera
Petermann
Mantegna
Lillo
Bonanno
Donetti
Hong
Munoz
Garrido
Herrmann
Torres
Gonzales
Roux
Dickman
Sousa
Yusong
Muren
Lingjiang
Castri
Garlaschelli
Battiston
Catanzaro
Loffredo
Pietronero
Montuori
Bottaccio
DeLuca
Provero
Gondran
Barthelemy
Pastor-Satorras
Vespignani
Weigt
Flammini
Leone
Vazquez
Zecchina
Barrat
Moreno
Echenique
GomezGardenes
Floria
Castellano
Loreto
Parisi
Cecconi
Radicchi
Maritan
Cieplak
Banavar
Giacometti
Fedroff
Holter
Rinaldo
RodriguezIturbe
Dorogovtsev
Samukhin
Goltsev
Mendes
Vragovic
Ramasco
Pacheco
Gomez
LopezRuiz
Sole
Ferrer
Cancho
Valverde
Kurtz
Kohler
Janssen
Gaulrais
Theraulaz
Salazarciudad
Garciafernandez
Montoya
DeFraysseix
Smith
Kepler
Johnson
Hoagy
Carroll
Fink
Barahona
Pecora
Pelaez
Vannucchi
Mendoza
Boccaletti
Hwang
Chavez
Amann
Digarbo
Meucci
Allaria
Arecchi
Valladares
Vallone
Zhou
Osipov
Kurths
Schafer
Park
Abel
Zaks
Volkmann
Rosenblum
Tass
Weule
Freund
Pikovsky
Schnitzler
Chate
Gregoire
Rudzick
Timmermann
Salmelin
Kujala
Gross
Hamalainen
Hari
Ilmoniemi
Lounasmaa
Knuutila
Liu
Ye
Hoppensteadt
Lai
Nishikawa
Dasgupta
DeMoura
Grebogi
Motter


Bill Cheswick/Lumeta

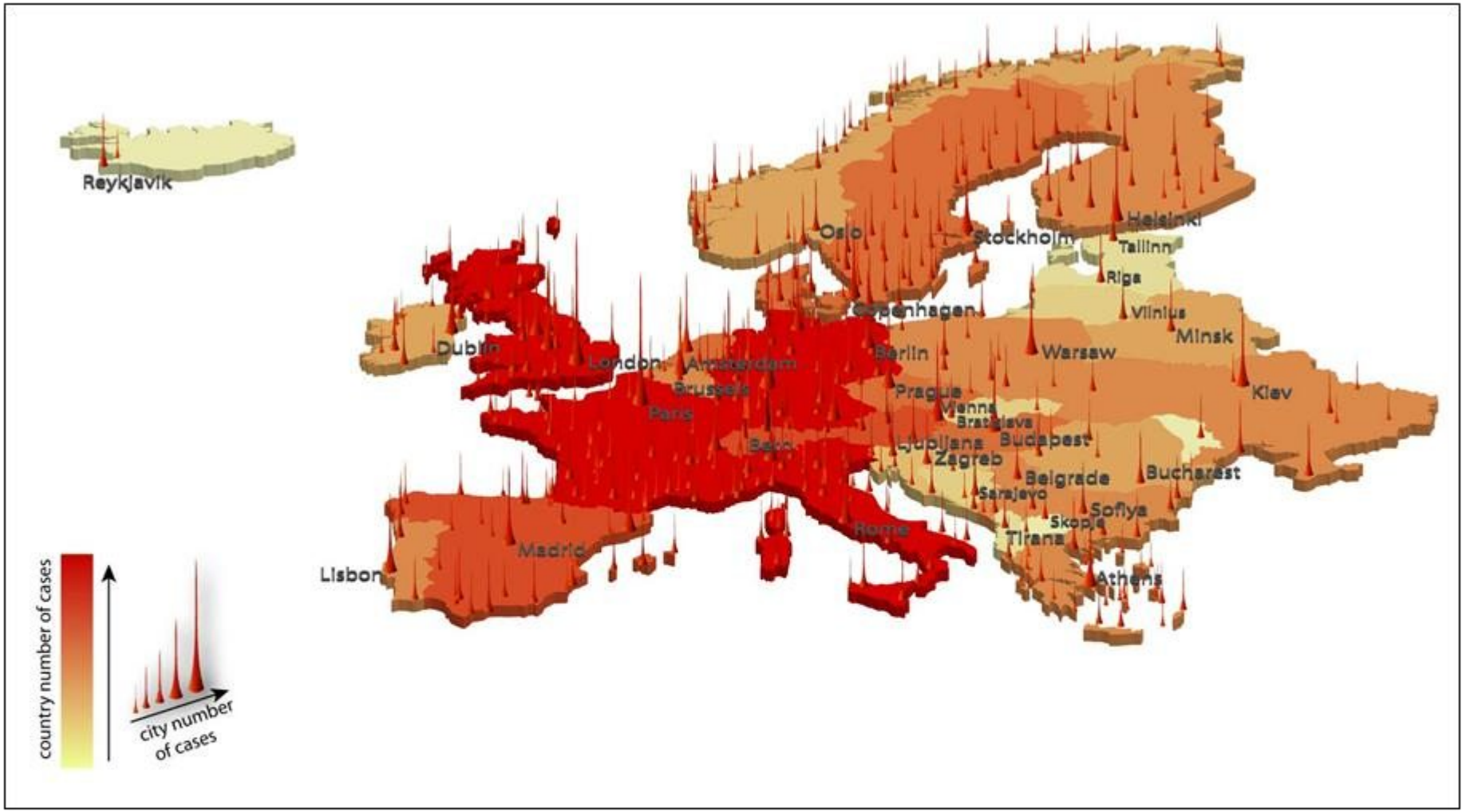


EpiSims (Eubank *et al.*)

Effects of airline travel



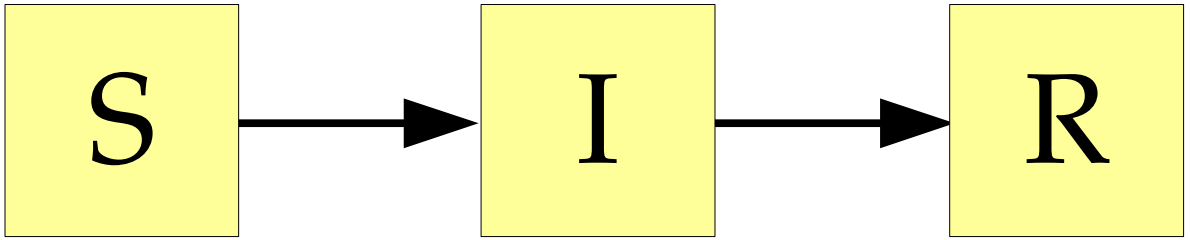
Colizza *et al*. 2007



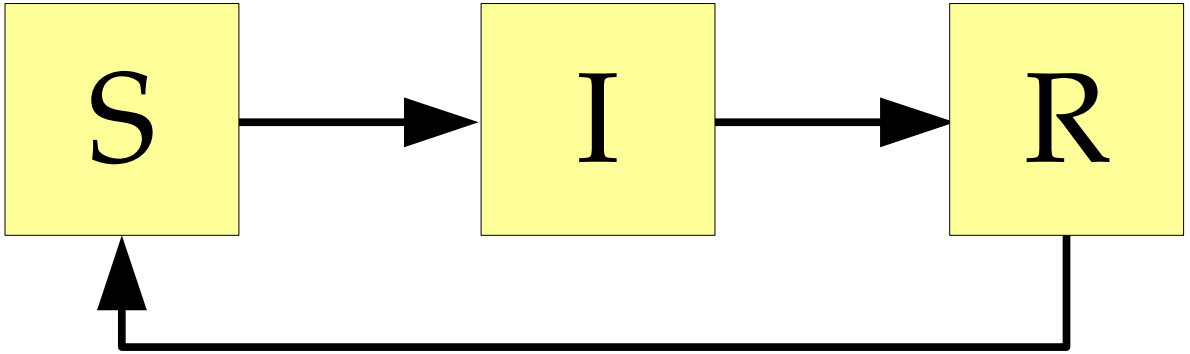
Simulated outbreak of a pandemic influenza in Europe. The epidemic starts in Hanoi, Vietnam, at the beginning of October with R_0=1.9 and no containment interventions are considered. The snapshot refers to the situation in Europe on March 1 of the following year. Countries are shown with a different color corresponding to the average number of cases observed within the country by March 1, from cream (10^3 cases) to red color (10^6 cases). The local situation within each city is represented by peaks whose height scales logarithmically with the number of cases reported in the city, from 10^2 cases to 10^5 cases. Simulations consider 3100 urban areas worldwide (513 in Europe) while for the sake of visualization the map shows only a set of major European cities.

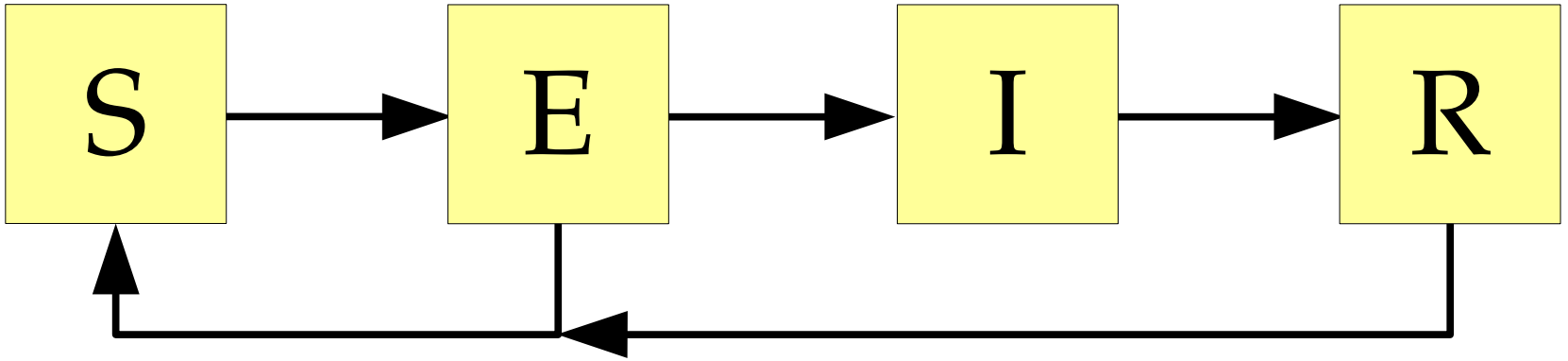
Colizza *et al*. 2007

- These are all SIR diseases:

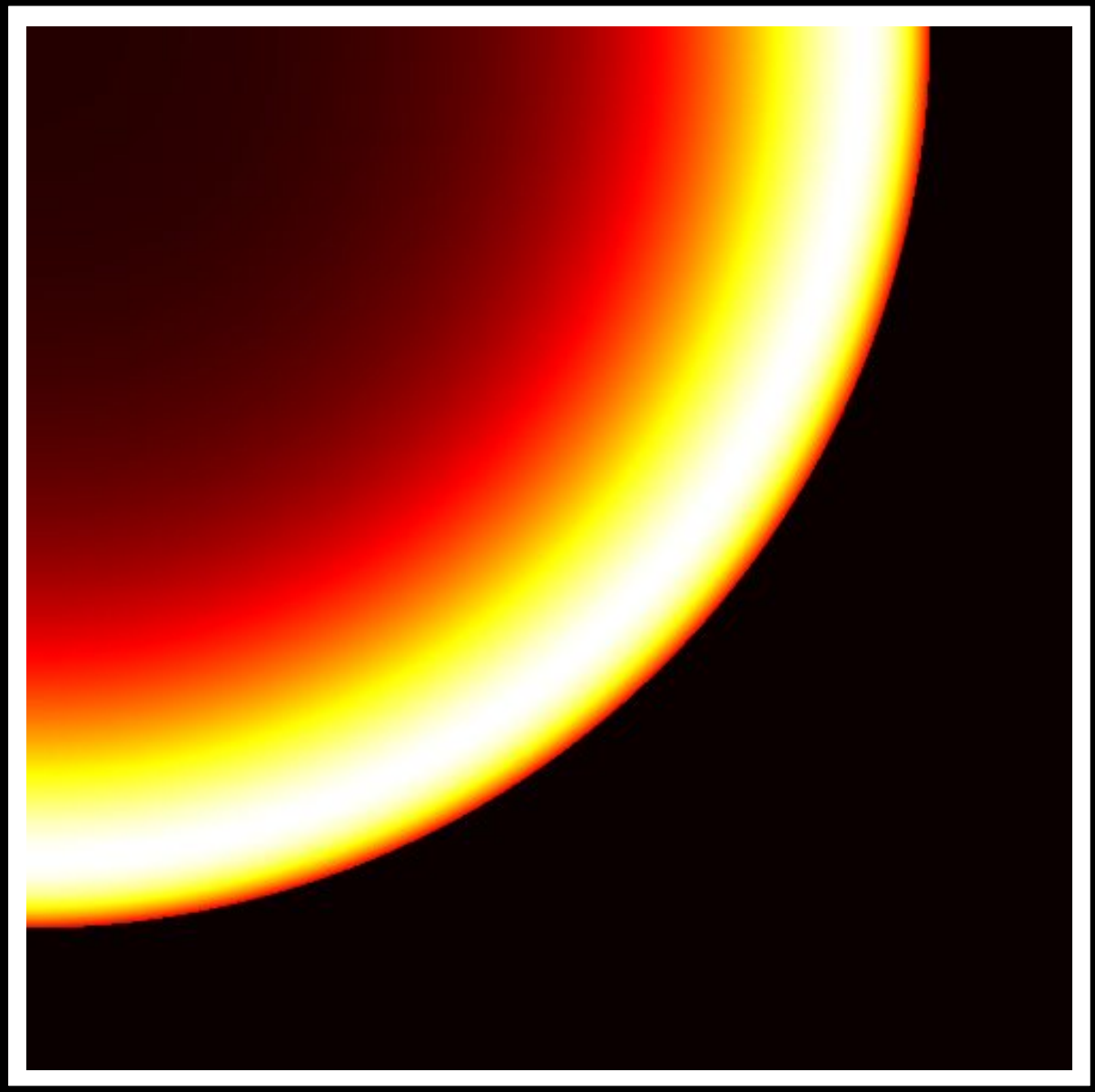


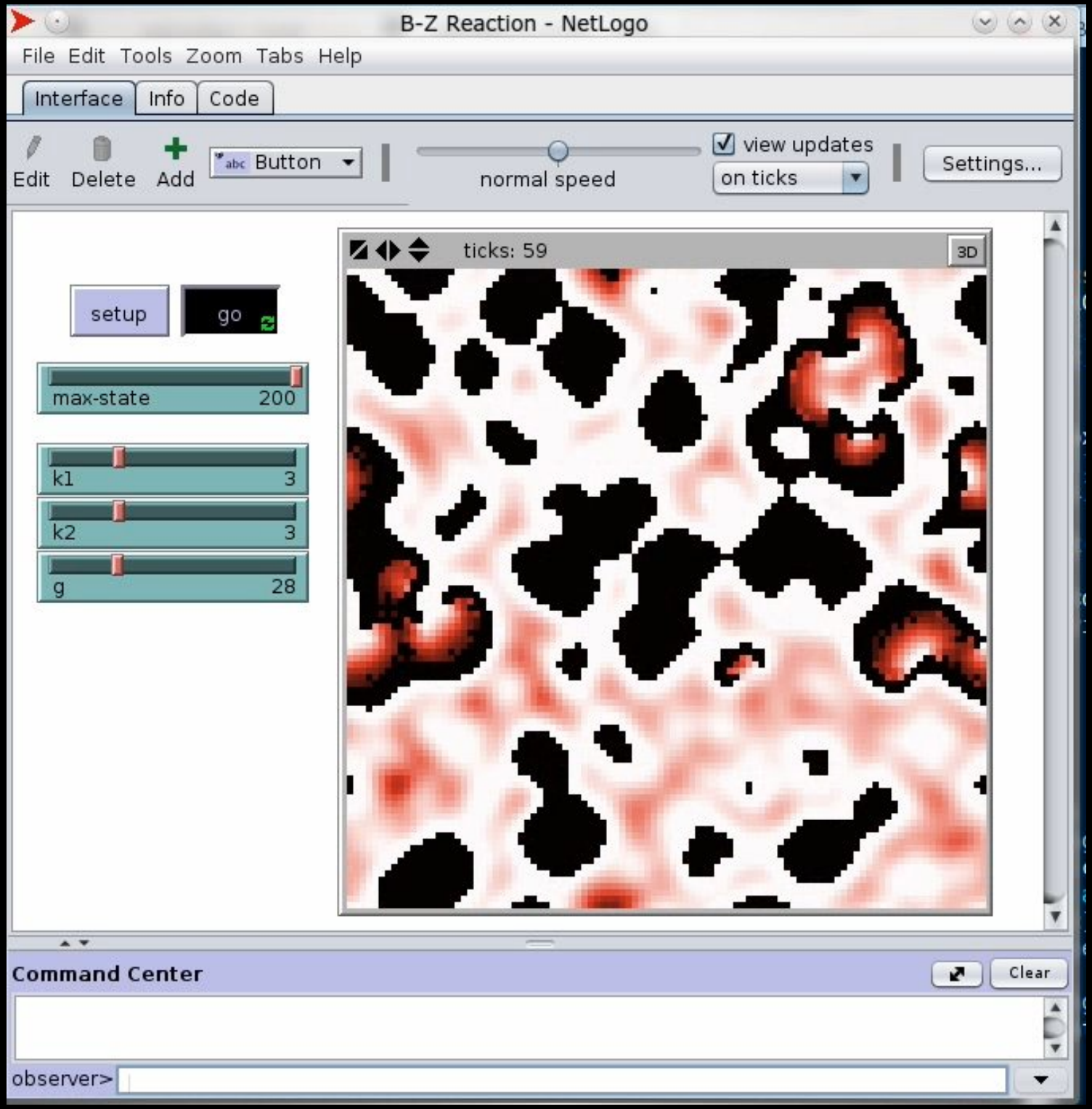
- There are other kinds, SIRS, SEIR, SEIRS:





Waves of disease




B-Z Reaction - NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code
Edit Delete Add Button
normal speed
view updates
on ticks
Settings...
setup
go
max-state 200
k1 3
k2 3
g 28
ticks: 59
3D
Command Center
Clear
observer>

Summary

- Complex systems can show simple emergent behaviours
- Randomized simulations are a powerful tool for understanding these behaviours
- Simple models give us new scientific understanding of physical, biological, and social processes
- More elaborate simulations can allow us to study specific situations in detail – how a change in road layout will affect traffic flow, or how people will move about a building