Simulating Influence on a Global Scale

A. Conrad Nied
Department of Computer Science
Boston University
anied@bu.edu

ABSTRACT

Infections and other forms of influence spread across more than nodes and edges, but the world as a whole. This project implements a data structure for visualizing forms of influence across the social network that is the globe. Models inspired by anthropology are applied to the program and form simulations. Anthropology is the study of humanity; more specifically it studies individuals and the relations between them, such as culture, religion, language, power, heritage and public health in biological anthropology. These topics can be translated to the nodes and edges of a social network. Problems can be transformed into algorithms from then. Thereby, this project uses anthropological intuition and software developed to display a global network as a map. Specific problems modeled are Disease Spread, Empire Building, Cultural Diffusion and Genetic Drift.

1. MOTIVATION

When analyzing large data sets, visualizations can be very helpful. The more visuals a data set lends itself to, the more it can be understood. Personally, I really enjoy studying maps and looking how they model information and its significance. Viewing socioeconomic factors on a map is analogous to viewing a social network. The interaction of geography and these factors happens on a network and real people are behind the data so political maps are basically very visually-appealing social networks.

After deciding I wanted to work with maps, I wrote a map visualization program. At this point it just drew a global population density map. The fun lay in designing algorithms to influence the map. Many models lent themselves easily to operating on a global map, the easiest being disease. Tweaking parameters and running the program to watch a disease ravage the world brings a sort of satisfaction.

Producing new models such as empires fighting over land, or culture diffusing and evolving is like playing a God game. However, there is no hand guiding the program once it has been told to run. The laws of nature (or more precisely, your algorithms) form the physics of each world you spawn when you click run. You are the Creator of your world and it is really fun to watch it grow.

2. THE NETWORK AND DATA COLLECTION

For this project, the social network is a map of the world. The points of data are analyzed on a Cartesian grid based on latitude and longitude. Depending on the resolution of the data, each pixel represents the area of half or a quarter of a degree on the map. The highest resolution grid is 1440 by 720 pixels, which equates to roughly one million data points, albeit a majority of them are unoccupied (usually being open ocean). Every pixel corresponds to a node on the social network. The network is completely connected, but the connections are weighted by their distance on the grid (with wrap around on the edges of the globe). Different algorithms take advantage of spatial locality and in general the network is treated as the 2D grid.

2.1 Localities

The first data collected combined the names, coordinates, and populations of significant localities, ranging from urban areas, counties, provinces, and entire countries. This data was compared to other factors such as HDI, political boundaries, and other significant geopolitical data but ultimately only population and location was significant for the algorithms employed.

The first source of population data was collection of localities with population statistics from the website World Gazetteer which mined the data from various media. This provided a launching pad for the information about significant urban areas and countries.

The coordinates of each locality were mined individually utilizing their names and finding it on Google Earth. This approach was exhausting but by scanning the globe for these data points, I started gaining an intuition over the relationships between these data points. Manually fining the coordinates also ensured that I matched the right ones, for example there are two cities with significant populations named Hyderabad, one in Pakistan the other in India.

Supplemental information on localities not included in the World Gazetteer dataset was found using Wikipedia. This included urban areas regionally significant but with little population such as Nuuk, Greenland and Pot Mosbi, Papua New Guinea.

Ultimately, 2361 localities were collected and processed. This data was used to form a preliminary population grid map and "urban area significance" factor, but ultimately most of this information was not used in the final algorithms. Intuition concerning these localities would later become significant in the Seeding algorithms discussed in section 4.1.

2.2 Population Grid

In the initial stages of the InfMap model, the locality information was used to model disease spread but there were far too many inconsistencies and problems that arose from using this sort of data. A population grid would be more useful to use for the algorithms and would be spatially fair.

This data set was provided by SEDAC, the Socioeconomic Data and Applications Center at Columbia University. Their data in Gridded Population of the World, version 3, was used. The dataset is a matrix of how many people live in each point. The points are provided at many resolutions, the ½° and ¼° data sets were used for this application.

This data set is overwhelmingly more accurate than using information on localities to derive population data. The source of this data is much more robustly documented and is not prone to as many errors based on the resolution of the data. The ¼° resolution data set has over one million data points, which provides for a much more accurate representation but weighs down the runtime of the program and requires restrictions on matrix calls and supplemental data.

2.3 Generalizations

Many generalizations were made to accommodate the primary goals of the project.

The first generalization was that the algorithms would operate on a Cartesian 2D plan rather than the surface of a 3D sphere. In algorithms such as radial spread, influence spreads on a radius based on the distance in degrees between two points. One a sphere, longitude overestimates distances between two points as one moves away from the equator. Thereby, influence spreading around London should influence a wider 2D grid of degrees than influence spreading around Dakar. The effect of using the 2D plane is that is slows down lateral influence in higher latitudes. Although a system designed to counteract this effect was implemented, ultimately the trigonometric calculations and function calls were too costly for the runtime. Slowing lateral influence can be justified by the observation that in areas that receive less sunlight (as a consequence of latitude) are colder and slow down the spread of influence regardless of how many people live in the vicinity. This isn't a fantastic assumption,

but saves quite a bit of runtime and in the end may not significantly affect the results.

While collecting other metrics besides population it was decided that using additional socioeconomic factors would enhance the accuracy of the project, but be too hard to find an accurate data set and may take up too much memory and runtime. Details such as birthrate, wealth, cleanliness, and trade routes would really improve the accuracy of some simulations. Unfortunately, there were no uniform, reliable data sets of this information that correspond to the same grid used for population. Although this information could be collected for localities and be interpreted on the grid, the accuracy of this data would be diminished and this mapping would generalize potentially very heterogeneous areas such as the distribution of wealth in most countries. Storing the population grid in the Java buffer already took quite a bit of memory, so adding more million-entry matrices and calls to each element in the algorithms would also degrade the performance.

Overall, population can be generalized to serve as a catchall for these socioeconomic factors. Wealth, birthrate, dirtiness, commerce and many other factors that positively increase influence directly correlate to higher populations. Thereby, the algorithms directly use population to determine the spread of influence, but indirectly call on other socioeconomic factors associated with population density.

The last primary generalization is in defining large cities. In each algorithm, points on the map are seeded with the influencing entity. This corresponds to shipments of vaccines, formations of new empires, and arrivals of diseases to airports. Using information from the localities dataset, jumps to large cities were written in instead of specifically recording airport activity and less useful data.

3. MODELS

After designing the population density map, models were designed to implement algorithms over the set of data. They start out simple and add layers of complexity as the program developed. InfMap studies the effect of just one influencing entity (namely a disease) while CultMap studies the effects of many influencing entities that interact and spawn more influencing entities.

3.1 InfMap: Disease Spread

The simplest model using this information generates a disease and propagates on the grid of population. The disease spreads to areas of higher population quicker, reflecting the role of concentrated areas in epidemics, and ultimately infects as many people as accessible on the grid.

Pseudocode:

```
Seed large city
Iterate over time
   Probability to seed large city
   Iterate over data points
      Radially influence neighbors
      Jump to large city nearby
      Kill percentage of influenced
```

3.2 EmpMap: Empire Building

The empire building model was inspired from the disease model. Instead of one influence spreading, multiple influences spread and interact on the population grid. Each point on the grid can be influenced only to the maximum of how many people live there and can hold allegiance to an empire. Thereby, in border areas, over-influenced points are reduced in influence proportionally to how much the empire is influencing the area and its competitive advantage compared to the other empires.

Pseudocode:

```
Seed starting empires
Iterate over time
   At specific times, seed points
   Iterate over data points
      Radially influence neighbors
      Jump/Reinforce influence
   Forall over-influenced points
      Scale     influence     given
imperial competitive advantages
```

3.3 CultMap: Cultural Diffusion

Further derived from the code for EmpMap, the competing influencers now also evolve in this representation. In a tree structure, progenitor cultures influence their vicinity while spawning child cultures. These children gain influence and occupy new territory adjacent to the parent culture or when the competitive advantage of their parent culture decreases they may take over their natal lands.

Pseudocode:

```
Seed progenitor cultures
Iterate over time
   Iterate over data points
      Radially influence neighbors
      Jump influence of this
culture and subculture
   Forall over-influenced points
      Scale influence given
cultural competitive advantages
   Randomly adjust cultural
competitive advantages
```

3.4 GeneMap: Genetic Drift

This last model is similar to the CultMap, but the evolution of the influencing entities based on genetics rather than an arbitrary tree structure. This model was developed last and is much less robust than the other models, but future development could scientifically and accurately generate models of genetic spread.

Pseudocode:

```
Seed progenitor genotype
Iterate over time
   Birth children based on parental
genotypes
   Migrate people based on
genotypic trends
      Use radial influence and jump
   Kill people based on deathrates
```

This algorithm is much more specific than the other influence paradigms. Migration uses shared algorithms, but births and deaths are specific to this program. The birthing algorithm uses allele frequencies and DNA recombination theory to determine the genetic composition of the generation's offspring (which includes some mutations. The killing algorithm simply uniformly reduces the population to 1-deathrate.

4. ALGORITHMS

Referenced in the pseudocodes of each model are many algorithms related to spreading influence. They are bolded in the pseudocode. Additionally, algorithms were developed for displaying the data on the maps provided in section five.

The primary intuition behind all of these algorithms is to keep the algorithm simple and generative, while reflect realistic expectations. Overly complex algorithms may be too specific to generate comparable results. Underlying the myriad of factors of influence in the real world there are simple equations that may not be precise, but they give a good picture.

4.1 Seeding

Since there are no preprogrammed areas of influence in any of the models, the first step is to seed points on the graph. This step simply adds z influence of w influencer to coordinate (x, y). Determining the parameters for each seeding is the important part of the algorithm.

A simple expert system was developed to match calls of particular localities. The top 200 or so localities based on geographic distinctness and population were added to this system. The name of the locality is put through a system of cases for first character, second character and so forth to find the coordinates for a given locality. For example infect("Calgary", 1000) would follow the case system for C, a, l, then g to find the coordinates for the city and differentiate it from Calcutta, Cairo, and Chicago.

Deciding where to seed is important. For InfMap, a random city is seeded with the disease and as time goes on additional random cities are seeded reflecting infected flights or breaches in containment. In EmpMap, each starting empire in relation in the graph is seeded from core cities of the empires such as the Celts starting in Paris, London, Barcelona, etc.

4.2 Radial Influence

The notably simplest algorithm is the radial spread of influence. Each data point iterates over its neighbor in a square. Influence is added to

each point in the neighborhood multiplied by a factor. In early versions of the program, this factor is determined by the real kilometer distance between two points and adjusted further, but the sheer number and complexity of these calls was punitive to the runtime. Later versions just use small factor such as 1/20 multiplied by a factor in a matrix relating to distance out.

+0.0	+0.1	+0.2	+0.1	+0.0
+0.1	+0.3	+0.5	+0.3	+0.1
+0.2	+0.5	+1.0	+0.5	+0.2
+0.1	+0.3	+0.5	+0.3	+0.1
+0.0	+0.1	+0.2	+0.1	+0.0

Figure 4.2.1: Radial influence matrix

The usage of the matrix is optional. Even if every cell in a radius of 2 gets the same amount of influence regardless of distance from the center, over time the influence will even out to a circular expanse (See image blurring algorithms). In models with competitive influence, using this distance matrix is preferred.

4.3 Jumping

With only seeding and radial influence, the simulations expand in more or less an even circle. This behavior is not terribly realistic, and although it may be for something like an asteroid impact, it is not for influence in a social network. The most important contribution to viewing this as a social network is the jumping algorithm.

The concept is that social influence will tend to nearby areas with larger populations. The algorithm thereby looks for the point in a certain radius around the originating point that has the highest population. In order to avoid every point from migrating to local maximum, such as the Northeast United States all trying to influence New York, only 20 random nearby points are searched. The point with the highest population of these random points has influence spread to it. Over time, local maxima such as New York will get more jumping influence sent to it, but runners up such as Boston and Philadelphia will get a percentage of these jumps to, correlated to their population in relation to their neighbors.

4.4 Weighing

In the models with multiple influencers, once an area's influence is maximized, excess influence has to be burned off. Strictly proportionally adjusting influence will cause some areas to adjust borders, but they stabilize quickly. In order to increase the dynamics of border cultural interaction, each culture is given a strength value. These strengths are set and stay there unless specific events happen as in the Roman Empire simulation, or they are constantly changing as in CultMap.

Weighing cultural strength allows for cultures to move into others and even for child cultures to replace their parent culture.

4.5 Color Gradient

After calculating influence for each round, coloring the map had to be done. The right balance of distinct colors and showing the most information had to be found.

For the single influence InfMap, the entire color spectrum can be taken advantage of. Fatalities, the Infected, and the Healthy populations determine how much red, green and blue each pixel had. One early iteration of the coloring protocol led outbreaks to look like starbursts in a nebula. Eventually the visually appealing solution was found in using the raw numbers of fatalities, infected and healthy, calculating their absolute influence on a 0-255 scale and their percentage of total influence in the point and combining the absolute and relative measurements to show influence in every point.

Usage of absolute color measurements gives a sense of population density while displaying the color, so one can identify areas in which the influencer is likely to go to.

In CultMap and the later stages of EmpMap, color gradients were introduced to highlight areas of only marginal, not complete influence. In these demos, each pixel is colored

in a winner takes all approach (or a top 3 winners). Because each empire has a completely different color in color space (see section 4.6) fading between two random colors would look too chaotic, but the percentage of control of the dominate culture can be shown. Adjusting the brightness of each pixel according to relative influence, a smoother map is produced and areas of high competition that may turn over can be easily identified.

4.6 Color Hashing

When presenting many different entities on the same map, using multiple colors can help distinguish between different entities. This issue first came from modeling localities influencing their immediate vicinity. Every locality had to be assigned a different color to look distinct. In EmpMap, the number of influencers is fixed so specific colors can be assigned to them, but in CultMap there are many, as much as 32 thousand cultures on the map at once.

In order to solve this problem, a hash function was developed. The function takes the ID of each culture and finds a deterministic but seemingly random red, green, and blue value given the ID. Each time a pixel is colored with the culture's ID the same color is produced, but located distinctly in the color space. Using a hash function ensures good variation between colors for each culture.

5. SCENARIOS

5.1 InfMap: Disease

This model brought together many of the algorithms and design decisions for how to use the data set and how to infect people. In the end, this model the most visually complete. In the figures, red corresponds to fatalities, green corresponds to infections, blue corresponds to healthy people, and black/dark red areas represent extinction with light shading to reflect the population that was once there. In the particular visuals shown, the disease starts in New York and after mostly wiping humans off of the Americas starts spreading in Lagos and Stockholm to cover most of Afrasia.

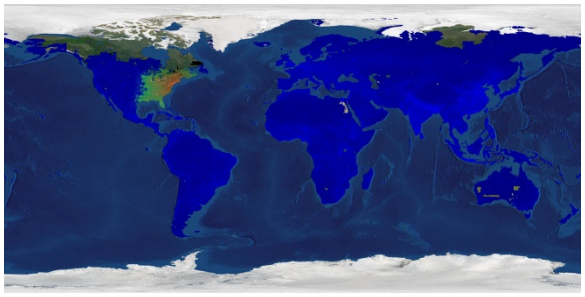

Figure 5.1.1: InfMap Disease at t=43

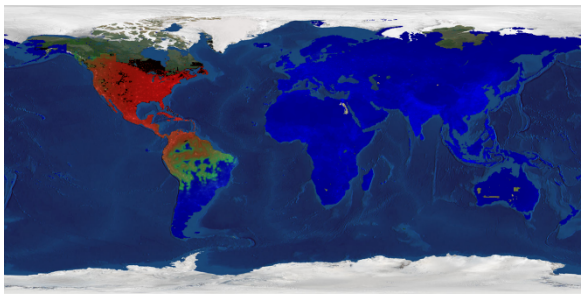

Figure 5.1.2: InfMap Disease at t=119

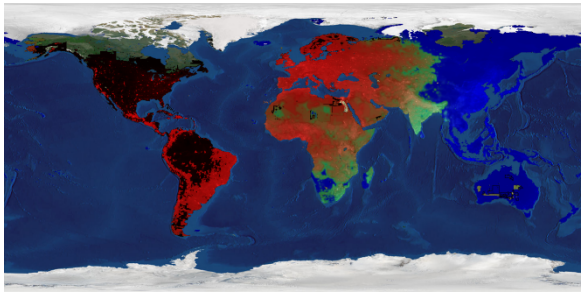

Figure 5.1.3: InfMap Disease at t=227

5.2 InfMap: Vaccine

As proposed by Evimaria Terzi, a vaccine is distributed during the infection in order to inoculate people from the disease. Although more sophisticated algorithms could be used, the vaccine was modeled from the disease itself. The spread basis for the vaccine is different in which it prefers to jump around urban areas, spreading the vaccine to the most cities possible, but not being well distributed within the city. The 50 seeding cities are given the vaccine at t=10 and as the disease develops, the vaccine is distributed to stop it. The colors are adjusted so that red is fatalities, green is healthy but vulnerable people and blue is immune people. In the end of the simulation, the more red an

area, the more people died before they could be inoculated. The disease is more virulent in this scenario and infected many more cities in the beginning.

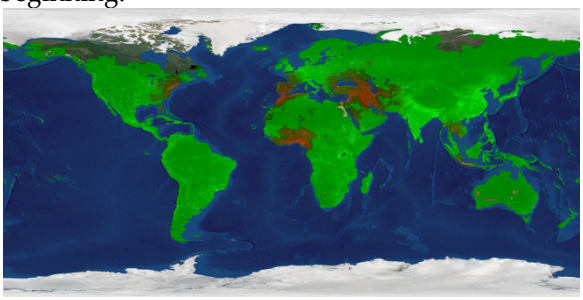

Figure 5.2.1: InfMap Vaccine at t=39

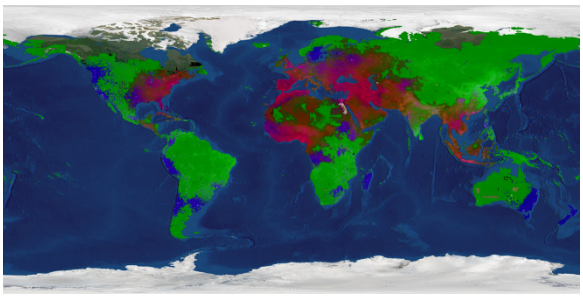

Figure 5.2.2: InfMap Vaccine at t=59

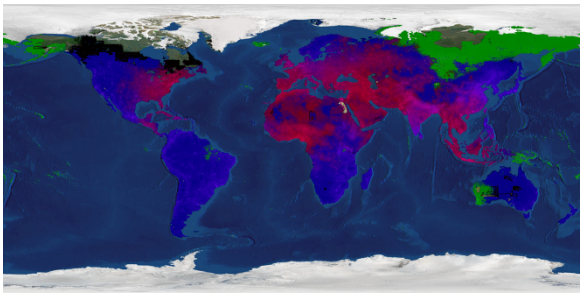

Figure 5.2.3: InfMap Vaccine at t=79

5.3 EmpMap: Roman Empire

This model starts with the surrounding area being seeded with the rival empires at the time of the rise of the Roman Empire. At t=100, the empire begins in Rome and conquers based on the comparative imperial advantage it has against the other empires on the map. At some times, the empire is encouraged to invade particular areas by seeding an important regional city with Roman influence. At t=600 the Roman advantage is substantially reduced and the Barbarians come back down to attack. By no means would I call this model historically accurate, but it is fun to watch.

Red = Rome
Black = Barbarians
Green = Celts
Blue = Carthaginians
Cyan = Persians/Seleucids
Yellow = Egyptians

Figure 5.3.1: EmpMap Roman Empire t=116

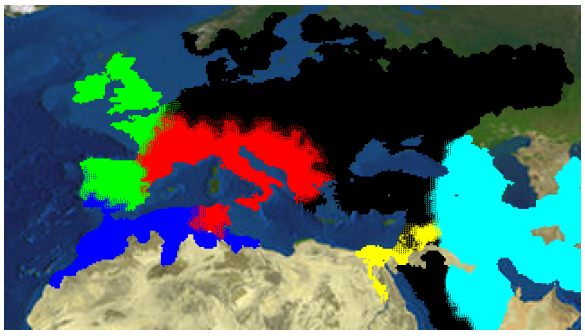

Figure 5.3.2: EmpMap Roman Empire t=288

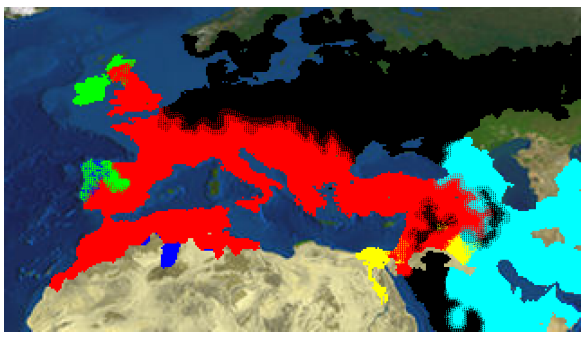

Figure 5.3.3: EmpMap Roman Empire t=600

5.4 CultMap: Latin America

This model does not reflect historic reality, but is an interesting examination of the spread of the cultures. Note that for this model and all other CultMaps, when the number of cultures reaches a the maximum short integer, no more cultures can be produced so some cultures at the leaf position of the binary tree mature without competitive children and dominate vast areas of land (this can be analogous to a rise in nationalism). Progenitor cultures are seeding in México City, Bogotá, and São Paulo

Figure 5.4.1: CultMap Latin America at t=55

Figure 5.4.2: CultMap Latin America at t=135

Figure 5.4.3: CultMap Latin America at t=499

5.5 CultMap: Europe

In this map, culture is seeded in Donetsk, Ukraine, close so where Proto-Indo-European is thought to have originated. The slides provided show the status of the world map far into the simulation. At this moment many cultural units have formed, resembling possible nations. Although the boundaries do not reflect actual European states, some correlate to states or linguistic areas such as West Germanic at t=1119, Celtic Ireland and Scotland at t=1209 and 1333, Belarus throughout the simulation, mainstream Romance languages at t=1209, and other possible interpretations.

Figure 5.5.1: CultMap Europe at t=1119

Figure 5.5.2: CultMap Europe at t=1209

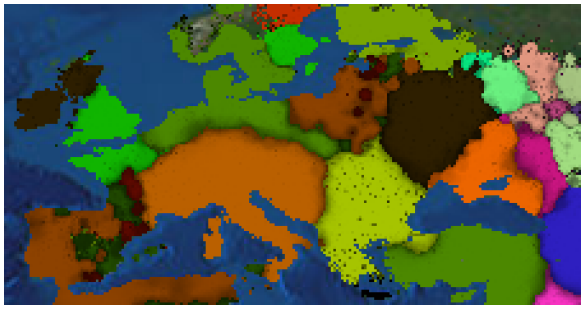

Figure 5.5.3: CultMap Europe at t=1333

5.6 CultMap: Proto-Indo-European

This map models the historic evolution of Proto-Indo-European on a Eurasian scale. The selected frames show the transformation of most of Europe from a yellow-colored culture to a red-colored one (its child culture) but also the rise and fall of marginal cultures such as in Iberia/France and Central Asia. Russia also turns over to a child culture and two very large cultures in the Middle East interact.

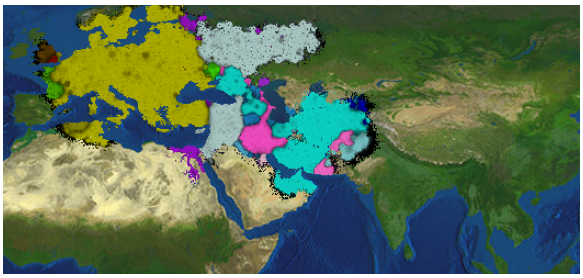

Figure 5.6.1: CultMap Eurasia at t=259

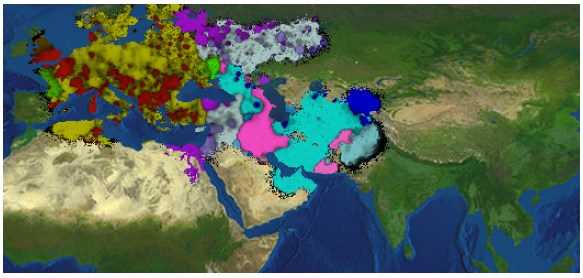

Figure 5.6.2: CultMap Eurasia at t=271

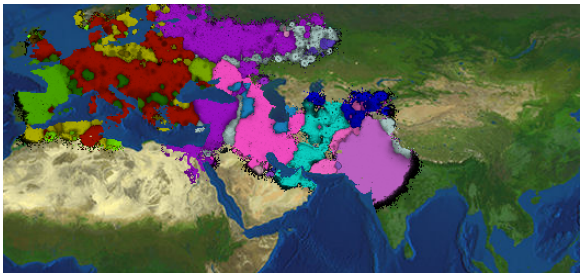

Figure 5.6.3: CultMap Eurasia at t=315

5.7 GeneMap: DRD4

The last modeling program is not complete, but shows promising results. This map displays genotypes. It begins at the dawn of time for humanity. At first, all humans have the same allele in their DRD4 gene named 4R, corresponding to green on the map. A mutation happens and the allele 7R is formed (represented by red). In modern humans, people with this allele are associated with ADHD, schizophrenia, and novelty seeking. Some biological anthropologists theorize that the presence of 7R in population groups encouraged some individuals to lead the migrations further and are ultimately responsible for populating the globe. The 7R allele is more common in areas away from Africa so a model confirming this assumption would have 4R present everywhere, but more 7R representation in the farther areas of the map. Another mutation 2R is shown in blue.

Although there is plenty of genetics intuition written into the model, there are many human evolutionary problems not addressed so the model is incomplete. The 4R allele at around t=108 stops functionally expanding because the 7R gene is favored, but realistically both genes should coexist just one favored a little more in extreme areas of the map. If the evolutionary algorithms are improved then this model could back up or refute claims that the 7R allele encouraged human migration.

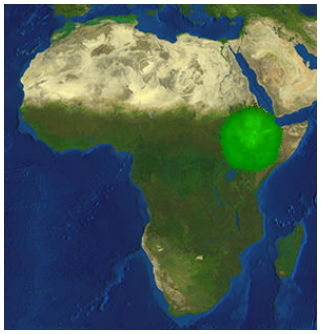

Figure 5.7.1: GeneMap Africa at t=48

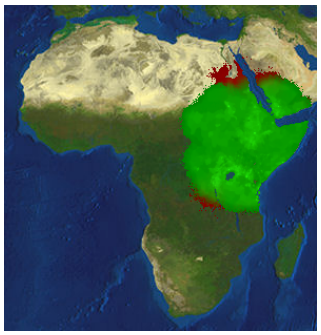

Figure 5.7.2: GeneMap Africa at t=108

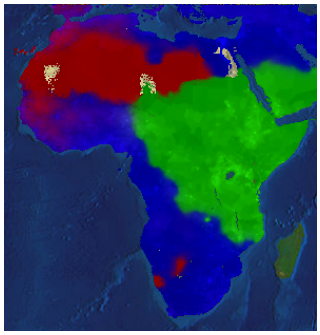

Figure 5.7.3: GeneMap Africa at t=279

6. EMERGENT BEHAVIOR

Once the parameters are set, the algorithms are configured, and the run button is pressed the program runs on its own with entities influencing on their own basis. Over time interesting behaviors began to emerge.

6.1 Population Corridors

Of course in the program, influence seeks cities and areas of larger population. This added up to forming corridors of high population. Influence travels down these population corridors before it starts to influence the surrounding area.

In a simulation of InfMap just looking at South America, a disease is started in Lima, Peru. The western side of the Andes is well populated, but the eastern side is dense jungle. People do live in the jungle mostly along major rivers, especially the Amazon. When the disease spreads from Peru it follows these populated corridors north and south on the Andes and along the Amazon River.

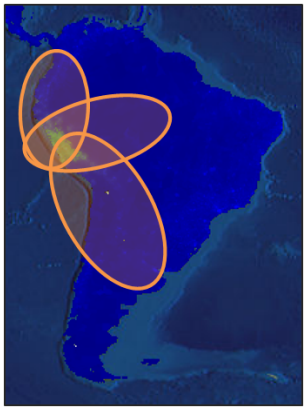

Figure 6.1.1: InfMap South America at t=17

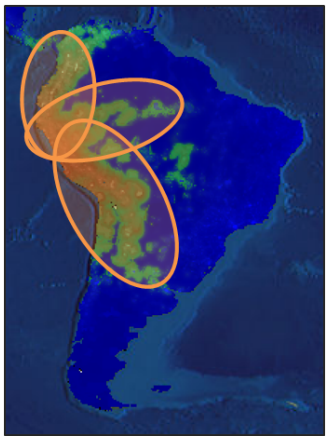

Figure 6.1.2: InfMap South America at t=46

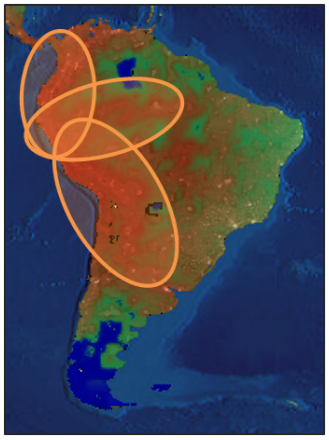

Figure 6.1.3: InfMap South America at t=76

6.2 Semi-Real States

In CultMap the boundaries of almost real states form and organize. Even successor cultures form into the molds of states that succeeded old ones.

The highlighted example is in an iteration of CultMap in the Near East the states of Turkey/Anatolia and Syria/Assyria almost match boundaries the states have held in historic times.

Figure 6.2.1: CultMap Near East at t=196

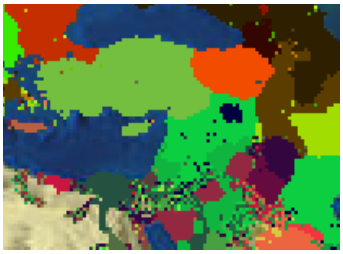

Figure 6.2.1: CultMap Near East at t=214

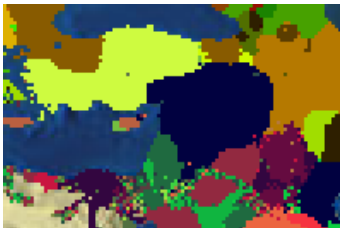

Figure 6.2.1: CultMap Near East at t=260

7. CONCLUSION

The mapping software is a great tool to visualize data on a global scale. Coupling the visualization with algorithms to model anthropological problems leads to many interesting scenarios.

InfMap is a visually complete study of the spread of disease across the world. With the current parameters, the disease is far more virulent than realistic, but the spread and devastation of the disease can be viewed. Models improved from this could accurately predict the spread of diseases such as swine flu or avian flu.

EmpMap is a good step into incorporating multiple competing empires vying to dominate their vicinity. Although many parameters such as seeding locations and empire weights are tweaked too much to fit the data, real events such as the Roman invasion of Dalmatia happened naturally given the algorithms and more natural behavior may be designed to form a self-evolving empire.

CultMap is a fascinating cultural simulator and can be used to create potential maps and pedigrees in history. State formation and cultural diffusion on the binary tree model seems to work. Additional considerations to decrease the randomness at which cultures have strengths should be implemented, as well as supplemental algorithms in which better model descendent culture evolution and cultural interaction forming entities such as Pidgins and Creoles.

GeneMap is underdeveloped but may shed light on theories proposed by people studying human migration. Further revising and incorporation of genetic research could create a realistic model for the emergence of DRD4 mutations useful in scientific study.

All in all, these models can be specified to more models and scenarios and watching evolutionary behavior on them is quite a joy.

REFERENCES

Data Set Sources:

Google Earth <http://earth.google.com/> coordinates of localities. Accessed 2010.01-03

SEDAC, Socioeconomic Data and Applications Center: <http://sedac.ciesin.columbia.edu/gpw/> population grid. Accessed 2010.03

World Gazetteer <http://world-gazetteer.com/> list of urban areas. Accessed 2010.01

Wikipedia <http://en.wikipedia.org/wiki/> supplemental information on urban areas. Accessed 2010.01-03

Some of the DRD4 research I used for another project that inspired me on this one:

Chang, FM, JR Kidd, KJ Livak, AJ Pakstis, and KK Kidd. "The world-wide distribution of allele frequencies at the human dopamine D4 receptor locus." Human Genetics 98, no. 1 (Jul 1996): 91-101.

McAuliffe, Kathleen. "They Don't Make Homo Sapiens Like They Used To." Discover, February 09, 2009.

Roussosa, Panos, Stella Giakoumaki, and Panos Bitsios. "Cognitive and emotional processing in high novelty seeking associated with the L-DRD4 genotype." Neuropsychologia 47, no. 7 (June 2009): 1654-1659.