

“The work upon which this publication is based was funded in whole or in part through a grant awarded by the Southern Region, State and Private Forestry, U. S. Forest Service.”
Award No: 08-DG-11083150-010

Contents

1	Introduction	2
2	Installation	3
3	Using SODBuster	3
3.1	Launching SODBuster	3
3.2	Specifying Input Files	3
3.3	Selecting Output Reports	4
3.4	Performing the Computations	5
3.5	Multiple Scenarios	5
4	Sample Application of SODBuster.	5
5	Input File Formats	5
5.1	Parameter File (*.sdp)	6
5.2	Node Information File (*.sni)	6
5.3	Commodity Flow on Links Information File (*.sfl)	7
6	Model Output	7
6.1	Human Readable Text Report	7
6.1.1	Input File Names	9
6.1.2	Observed Infection Pattern	9
6.1.3	Source Infection Probabilities	9
6.1.4	Posterior Destination Probabilities	10
6.2	Create Maps	10
6.2.1	Sample Maps	11
6.3	CSV (TAB Delimited Values) Report	11

List of Figures

1	Opening Window for SODBuster	4
2	Sample SODBuster Network	5
3	Sample Parameter (*.sdp) File	8
4	Sample Node Information (*.sni) File	8
5	Sample Flow Information (*.sfl) File	8
6	Example of Observed Infection Pattern Report	9
7	Example of Source Infection Probabilities Report	9
8	Example of Posterior Destination Probabilities Report	10

9	Map Selection Window	12
10	Example of a Node Map	12
11	Small Scale Example of a Risk Map	13
12	Large Scale Example of a Risk Map	13

1 Introduction

This document describes the use of the SODBuster Software Package developed for the USDA Forest Service under the award cited above. The purpose of this software is to implement the mathematical model described in the article, “Probabilistic Techniques Used in SODBuster.” A PDF version of the article can be found at the following URL:

<http://www.stolaf.edu/people/mckelvey/SOD.dir/BaseEMTech.pdf>

The software was developed using the Java software development kit from Sun Microsystems. The PDF version for this user guide can be found at the following URL:

<http://www.stolaf.edu/people/mckelvey/SOD.dir/UserGuide.pdf>

The goal of the probabilistic model and implementing software is to give the USDA Forest Service an analytical tool to help focus scarce inspection resources on the early detection of *Phytophthora ramorum* outbreaks in those parts of North America where *P. ramorum*, the organism that causes Sudden Oak Death (SOD), is not yet endemic. This is accomplished by using partial survey results, along with commodity flow information, to create an ordered list of those sites presently not known to be infected. The list is ordered by likelihood of each site having recently become infected through the importation of infectious nursery stock.

The process of creating this list consists of several stages. In the first stage some subset of vulnerable sites, in the case of *P. ramorum* these sites will typically be areas east of the Rocky Mountains, are surveyed. The surveyed sites are categorized as being recently infected, very likely to be uninfected, or being a site for which infection status is uncertain. Sites with an uncertain infection status will be treated as though they were not surveyed. The combination of newly infected sites and recently certified clean sites is called an *infection pattern*.

Once newly infected and known clean sites are identified, known potential sources of infectious nursery stock are assigned probabilities of being sources of infectious nursery stock. In the terminology of probability theory this is a Bayesian process in which the probability of infectious exports assigned to each potential source is updated from some previous value based on the infection pattern observed. For example, those sources which happen to send a large amount of nursery stock to newly infected destinations will be assigned a high probability of exporting infectious materials because the new infections must have come from somewhere and the source sending materials to these destinations are good suspects. Similarly, sources that send large amounts of nursery stock to sites classified as known clean sites will be given a low probability of sending infectious exports because receiving these exports has not resulted in infection.

After the probabilities of exporting infectious materials have been updated attention moves to the unsurveyed recipients of nursery stock. For each unsurveyed recipient of nursery stock, called **destinations**, a probability is computed that this site has become recently infected. This probability is based on two characteristics of the destination, from which sources the destination’s nursery stock is sent and how much nursery stock comes from each source. If a given destination receives a significant amount of its stock from a high risk source, that destination will be assigned a relatively high probability of infection. Conversely, if a destination receives very little stock from high risk sources, it will be assigned a low risk of infection.

Once risks have been assigned to the unsurveyed destinations, inspection resources can be mobilized to high risk destinations with the aim of identifying those sites that are, in fact, infected and actions to eliminate the threat of introducing *P. ramorum* into forests currently free of Sudden Oak Death (SOD).

2 Installation

This software was developed using the Java 2 (Version 1.5.0) Sun Microsystems Java Development Kit (JDK). To use this software a compatible version of the Java Runtime Environment (JRE) must be installed on your computer. The JRE is available for free download for almost all hardware configurations and common operating systems (Windows, Mac, Linux, Unix, etc.). If you need to install an appropriate JRE, please see the Sun Microsystems website for download instructions. As of the writing of this document, the appropriate URL is:

<http://www.java.com/en/download/index.jsp>

Once the JRE is installed on your computer, the installation of the SODBuster software consists of copying the file named `SOD3.jar` from the installation media to any folder of your choice on the host computer.

3 Using SODBuster

3.1 Launching SODBuster

On most operating systems there are two ways to launch the SODBuster program. On systems with graphical user interfaces (Microsoft Windows, Apple OS X, etc.) the easiest way to launch is to move into the folder containing the `SOD3.jar` file and double-click on this file. You can launch from other locations by creating short-cuts to the `SOD3.jar` file from anywhere you find convenient.

If you are using a command line interface, the usual command to launch the SODBuster program is:

```
java -jar SOD3.jar
```

The above instructions should be sufficient for the SODBuster software. If you care to examine the Java source code for the program, it resides within the `SOD3.jar` file along with an HTML document generated by the `javadoc` program that provides full documentation of the classes used by the `SOD3` program. The procedures for extracting these files from the `SOD3.jar` file are both standard and beyond the scope of this user's guide.

Upon launching, `SODBuster` displays its main window, as shown in Figure 1. This screen allows the user to select which input files (see section 5) should be used by the current run and which subsets of output (see section 6) should be created.

3.2 Specifying Input Files

The top four lines of this opening window deal with designating input files and subjecting the indicating files to syntax and consistency tests. The "Browse" button on each line opens a file selection window that allows you to select an input file through a series of clicks. The precise appearance of this selection window varies widely by operating system, but yours should look familiar to you if you are using your normal operating system.

In performing the file selection it is important to note that the process begins in your "Home" folder. Which folder is considered the "Home" folder depends on your operating system and the

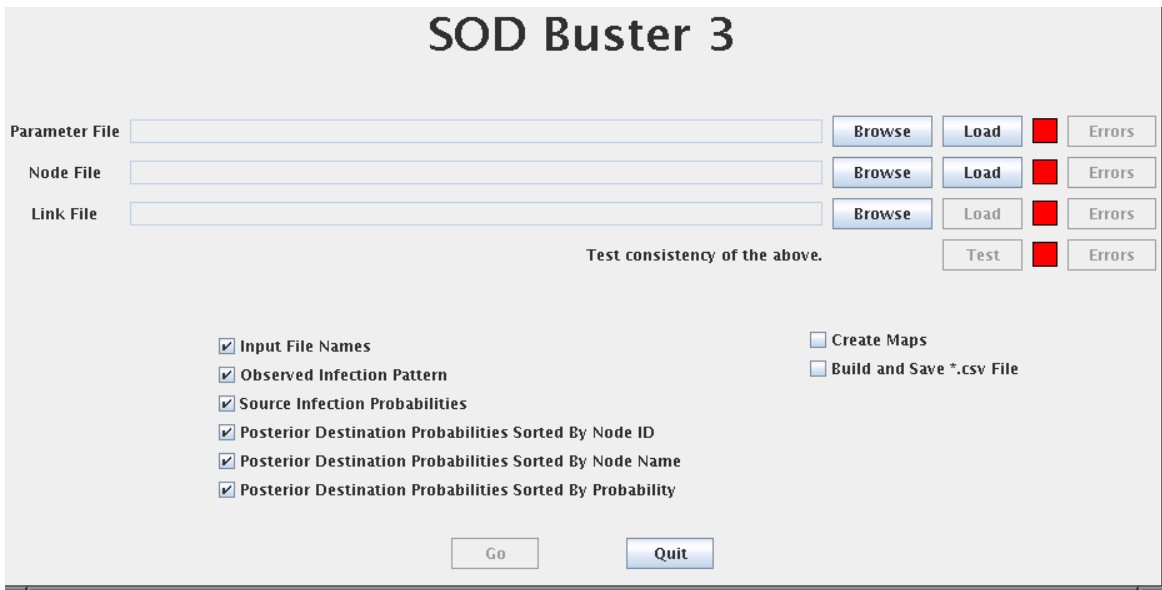


Figure 1: Opening Window for SODBuster

specific configuration of your computer. Be aware that you may have to click through several folders to find the input file you want to select. Fortunately, subsequent browsing in the same run of **SODBuster** will begin in the folder you initially selected, not back in the potentially distant home folder. It is also possible to put a short cut in your home folder that points to the folder containing input files. This technique can greatly reduce the number of click throughs you need to perform when running **SODBuster**.

Once you have used the file selection window to select an input file, you must click on the corresponding “Load” button to tell **SODBuster** to read the file and check it for syntax errors. If all is well, the line’s red box will change to green. If a problem is detected the box will remain red and the “Errors” button will activate. **SODBuster** attempts to generate helpful diagnostic messages if problems are encountered while loading input files. Clicking on an activated error button will show you these error messages.

You may have noticed that the “Load” button for the third input file is not initially activated. This is because the information for individual nodes must be read before link information can be properly processed. The “Load” button for the link file will be activated as soon as the node file has been successfully loaded.

Once all three input files have been successfully loaded, the last validation tests for the input data are performed when the “Test” button is pressed. If this final test is satisfied, the “Go” button on the bottom line of this window will be activated, indicating that **SODBuster** is ready to perform its calculations.

While great care has been taken to test the validity and consistency of input data, and to provide helpful diagnostic error messages when necessary, it is important to understand that not all erroneous conditions can be automatically detected. It is incumbent upon the user of this software to take care when assembling input files to ensure the accuracy and validity of the information represented in the files.

3.3 Selecting Output Reports

The lower half of the opening window consists of two columns of check boxes. The left hand column controls the contents of a text report created by **SODBuster** at the end of its run. This report will

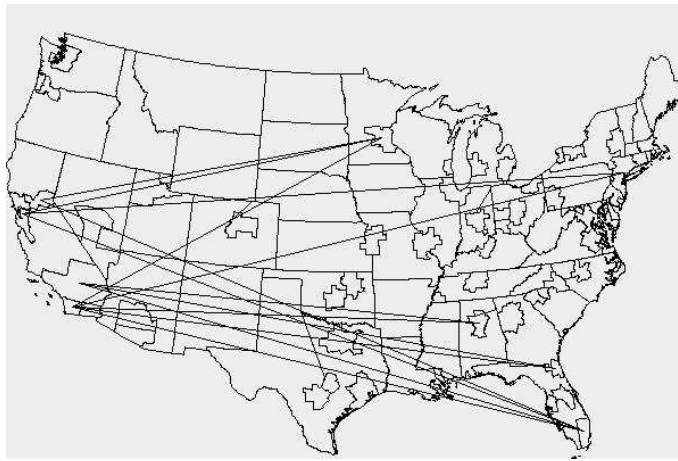


Figure 2: Sample SODBuster Network

appear on the computer screen when the computations are finished and will contain the information indicated by the checked boxes. By default, all the fields are selected, which can result in a longer than necessary text report. You should feel free to uncheck any boxes that refer to information you do not need.

The right hand column consists of two check boxes, both left unchecked by default. Checking the first instructs **SODBuster** to draw and display various maps of the United States showing the locations of nodes, transportation links and the relative risk of infection of destinations.

The second check box instructs the software to create a tab delimited file, known as a **csv** file, containing all the information the software uses and creates. This file is suitable for loading into a spreadsheet, allowing the user to perform whatever analysis is of interest.

3.4 Performing the Computations

The last line of the opening window allows for two choices. To quit the application altogether or to “Go,” meaning to perform the calculations and produce the requested reports, files and maps.

3.5 Multiple Scenarios

At any time it is possible to change the name of an input file. It may be necessary to click on several “Load” buttons to perform all the consistency tests necessitated by a change in an input file, but there is no need to quit and relaunch **SODBuster** in order to investigate a series of scenarios.

4 Sample Application of SODBuster.

Throughout this user guide samples of inputs and outputs are shown. These samples are derived from a very simple hypothetical test case of the program consisting of four sources and five destinations. The network used in this demonstration is shown in Figure 2. The sources are those nodes found in California. All nodes to the east of California are considered destination nodes in this fictitious scenario.

5 Input File Formats

The **SODBuster** software uses plain text input files to provide information on the structure of the nursery stock shipping network, the a priori probability of infectiousness for each source, the results

of recent surveys of some subset of the destinations and the unit flow probability of infection, the vulnerability of destination nodes to infection by incoming infectious nursery stock. These parameters are provided by three different input files. The structure of each input file is indexed by the extension (the three characters after the final dot) appended to the file's name. Other than honoring the presence of the required extension, the names of the files are restricted only by the host computer's operating system.

The software imposes no limitations on the length of any input line.

The formats for these three types of input files are specified below. In each case user comments can be introduced into the input files by beginning a line with a # symbol. In this case the line is ignored by the program and the user can type any useful text onto this line without affecting the operation of the software.

The three input formats allow for completely blank lines. These are never required and are ignored if encountered. Blank lines can be used to increase readability and emphasize the structure of the input files.

5.1 Parameter File (*.sdp)

The parameter file must have a file name with extension `sdp`. It provides the SODBuster model with values for various parameters. Specifically it provides values for

Unit Flow Probability of Infection The probability that a single unit of flow from a source that is exporting infectious nursery stock will cause a new infection at the previously uninfected destination.

Newly Found Infected Sites The integer node ID numbers for those destinations that are classified as newly infected.

Known Clean Sites The integer node ID numbers for those destinations that are classified as known clean sites. These are sites that have been recently inspected for which the inspecting authority has significant confidence of noninfectedness.

A priori Source Infection Probabilities Each source node in the model must be given an *a priori* probability of infection. This is the model user's best approximation of the probability that a given source is exporting infectious nursery stock capable of causing an SOD outbreak at a destination node.

This text flat file should be in the following format (order of lines is unimportant):

```
UnitFlowProb=[unit flow probability]
NewlyFoundInfectedSites=[comma separated list of sink ID integers]
KnownCleanSites=[comma separated list of sink ID integers]
AprioriSourceInfectionProbabilities=[comma separated list of Source:Prob pairs]
```

The `Source:Prob` pairs consist of a source ID integer, a colon and a probability value.

Figure 3 shows an example of a complete, though small, SODBuster parameter file.

5.2 Node Information File (*.sni)

This file contains information on each node in the network. The name of the file must end with the extension `.sni`.

Each line corresponds to a single node and provides a numerical ID for the node along with a string name, the NAD83 Alber's projection coordinates for the node and a classification of the node as a source or vulnerable destination of possible SOD infection.

To be valid, the node names and node IDs must be distinct from all other node names and node IDs.

Each line has the following format (fields are delimited by commas):

Field 1: Node Name (string, any length okay)

Field 2: Node ID (non-negative integer)

Field 3: X Coordinate, Alber's Projection (floating point number, positive or negative okay)

Field 4: Y Coordinate, Alber's Projection (floating point number, positive or negative okay)

Field 5: Source/Destination status (one character, **s** for source, **d** for destination. The letter's case is unimportant, no quotation marks should be used.)

Figure 4 shows an example of a complete, though small, SODBuster node information file.

5.3 Commodity Flow on Links Information File (*.sfl)

This file contains information on the amount (weight) of nursery stock that flows between network nodes defined in the Node Information File. The SODBuster software ignores any link that has a first node that is classified as a destination node or a second node that is classified as a source node. Having such link represented in a *.sfl file is not an error, per se, but all such links are discarded as the file is read. This allows for the easy change of node classifications without having to make corresponding changes to the flow information file.

The file is considered to be invalid if any link has a zero flow or if a node ID appearing in the file carries neither a source or destination classification in the *.sni file. Each link should appear at most once in the *.sfl file.

Each line of the *.sfl file corresponds to a link. The fields are delimited by commas and are defined as follows:

Field 1: Node ID 1, the source node from which this flow originates (non-negative integer)

Field 2: Node ID 2, the destination node at which this flow terminates (non-negative integer)

Field 3: Amount of flow (strictly positive floating point number)

Figure 5 shows an example of a complete, though small, SODBuster link flow information file.

6 Model Output

As mentioned in section 3.3, the user has control over the output generated by a run of SODBuster. Broadly speaking, SODBuster produces three types of output: a human readable textual report, a series of map images and a textual report suitable for exporting into a wide variety of spreadsheet software. A single run of SODBuster can produce any or all of these products.

6.1 Human Readable Text Report

Each run of SODBuster generates a human readable report of the run's results. The report appears as the contents of a scrollable text window that will appear once the run's computations are complete. It is possible to save this report as a text file. If the text file is opened with a text editor or word processor using a fixed width font, the report will appear as a neatly formatted document. The structure is lost when proportional width fonts are used.

In section 3.3 we discussed the mechanism for choosing which information will be included in the text report generated by SODBuster. Below we describe in a little more detail what each section of this report contains.

```
# sept25.sdp, the parameter file for a SODBuster test.

UnitFlowProb=0.01

NewlyFoundInfectedSites=6,8
KnownCleanSites=5,9
AprioriSourceInfectionProbabilities=1:0.1,2:0.2,3:0.3,4:0.4
```

Figure 3: Sample Parameter (*.sdp) File

```
# sept25.sni, the node information file for a SODBuster test.

CA Los A,1,-1860336.6924900000,1473280.66945000000,s
CA San D,2,-1909942.4422600000,1316587.29083000000,s
CA Sacra,3,-2121296.7190100000,2039721.30065000000,s
CA San J,4,-2244464.0445600000,1941463.48114000000,s

AL Birmi,5,845287.83937400000,1205035.99044000000,d
FL Jacks,6,1360113.96170000000,897392.03159800000,d
FL Miami,7,1555113.80044000000,472049.66795000000,d
MN Minne,8,188085.69660700000,2457567.21844000000,d
NY New Y,9,1821405.18805000000,2225019.18194000000,d
```

Figure 4: Sample Node Information (*.sni) File

```
# sept25.sf1 -- A SODBuster link flow information file for
# testing.

1,5,100.0
1,6,10.0
1,7,30.0
2, 5, 10.0
2,6,40.0
2,7,1.0
2,8,5.0
3,7,50.0
3,8,20.0
4, 7,35.0
4,8,25.0
4,9,15.0
2,9,3.0
```

Figure 5: Sample Flow Information (*.sf1) File

6.1.1 Input File Names

The full file names of the three input files used by this run of **SODBuster** are listed. This is useful to keep track of multiple reports when many runs are performed in a short period of time.

6.1.2 Observed Infection Pattern

This section lists the destination ID numbers and node names for the sources listed as newly discovered infected sites and newly certified clean sites. The Observed Infection Pattern section generated by our example is shown in Figure 6.

```
-----  
  
Newly discovered infected site node IDs:  
  
    6   FL Jacks  
    8   MN Minne  
  
Known clean site node IDs:  
  
    5   AL Birmi  
    9   NY New Y  
  
-----
```

Figure 6: Example of Observed Infection Pattern Report

6.1.3 Source Infection Probabilities

This section of the human readable report lists, for each source, the source's node ID number, the source's name and the given a priori probability of exporting infectious nursery and the updated (posterior) probability of exporting infectious nursery stock. The Source Infection Probabilities section of the report generated by our example data is shown in Figure 7.

```
-----  
  
Source Infection Probabilities  
  
Node ID      Node Name      A priori      Posterior  
  
    1         CA Los A      0.1000        0.0817  
    2         CA San D      0.2000        0.9629  
    3         CA Sacra      0.3000        0.4952  
    4         CA San J      0.4000        0.6346  
  
-----
```

Figure 7: Example of Source Infection Probabilities Report

6.1.4 Posterior Destination Probabilities

This section of the text report shows the probabilities of infection for every destination, sorted in various ways. These are the probabilities that are designed to assist managers when making inspection assignments to at-risk destinations. Each line in this section corresponds to a single destination and provides three pieces of information, the destination's node ID number, the destination's name and the computed probability that the destination is infected based on the observed infection pattern.

Every destination node is listed in these sections. For nodes with a precisely known status, destinations known to have been recently infected or destinations known to be clean, an asterisk appears next to the node ID number. These destinations have infection probabilities of either 1.00 (known recently infected) or 0.00 (known to be clean). The remaining destinations have not recently undergone definitive inspections so their status is not known with certainty, but the relative risk is represented by the reported probability of infection.

The Posterior Destination Probabilities Report section generated by our example is shown in Figure 8. In this example the destinations are sorted by probability of infection.

```
-----  
--Destination Nodes Sorted by probability of infection--  
  
(* indicates known infected or known clean node.)  
  
   ID      Node Name      P(Infected)  
  
   8*     MN Minne          1.0000  
   6*     FL Jacks          1.0000  
   7       FL Miami          0.3738  
   9*     NY New Y          0.0000  
   5*     AL Birmi          0.0000  
  
-----
```

Figure 8: Example of Posterior Destination Probabilities Report

6.2 Create Maps

SODBuster has the capability of producing several types of maps that provide visual support for the model's conclusions. The existing maps show both the scale of the problem being solved and the spatial distribution of the SOD threat to currently pristine areas of the United States.

When the Create Maps checkbox is selected before a SODBuster run, a tabbed map window appears after the computations are complete. This window is presented in Figure 9.

Currently there are four map types implemented in SODBuster. They are:

Empty This shows an empty map of the continental United States and shows the regions into which the country has been divided for the sake of nursery stock flow data aggregation. This map is the base layer to which the other map types add information.

The careful observer will notice a few stray straight lines. Texas and Colorado exhibit the most egregious examples. These lines occur when a region has holes in it. The drawing

algorithm used in producing these maps draws a given region without raising the “pen.” The straight lines are caused by the leap from the outer boundary of a region to an inner boundary.

It would have been possible to fix this problem, but substantial amounts of time would have been required. Given the limited time available for software development, it was determined that fixing this minor problem was a low priority in terms of the overall project goals.

All Links This map shows all the links carrying strictly positive amounts of nursery stock from a source to a destination. This map is a particularly powerful demonstration of the magnitude of the problem solved by this software.

All Nodes This map shows the locations of all the sources (in red) and destinations (in green) used in the current **SODBuster** run.

Dest Risk This map shows all destination nodes as red discs. The **area** of each disc is proportional to the destination’s probability of infection. Destinations that are known clean or recently discovered to be infected are excluded from this map.

Destinations with unusually high probabilities of infection are denoted by a large red disk with a yellow cross in the center. The areas of these high risk discs are fixed, not in proportion to their risk. The definition of “unusually high” risk is any probability in excess of four standard deviations above the mean of all reported source probabilities.

All four map types are available after a given run of **SODBuster**. To see a map, click on the appropriate tab. Every map shown can be saved as an image in GIF format. Clicking the “Save Map” button starts the saving operation for the map currently being shown. It is recommended that the user provide a file name ending with the extension `.gif`.

The “Close” button closes the map tab window permanently.

6.2.1 Sample Maps

Figure 9 shows an example of the empty map. The link map corresponding to the small scale example used throughout this guide has already been shown in Figure 2.

Figures 10 and 11 show the node map and risk map for our example problem. Since the example problem has only one destination with unknown infection status, the risk map shown in Figure 11 is not particularly enlightening. A more typical risk map is shown in Figure 12. This map shows the fictitious results from a full scale test of the **SODBuster** software.

6.3 CSV (TAB Delimited Values) Report

The text file created by this option is meant to be read by other computer applications, primarily spreadsheets that can import TAB delimited text files. All the information used or created by **SODBuster** is included in this text file; all the input values are reported as well as all the probabilities computed by the **SODBuster** software. A forest manager or researcher can use this data in any desired manner. The reason for this report option is to allow the results of **SODBuster** to be utilized in ways that may not have been identified at the time of this program’s creation.

The CSV file is created and saved automatically whenever the appropriate check button in the opening window is checked. The newly created CSV file is saved into the folder that holds the parameter input file. The name of the CSV file is the same as the name of the parameter input file except the extension `.sdp` is removed and replaced by the extension `.csv`. For example, if the parameter input file was named `sod2020.sdp`, the CSV file created by this run of **SODBuster** would be named `sod2020.csv`.

One important consequence of this naming scheme occurs when several **SODBuster** runs are undertaken using the same parameter input file. In this case, every run will overwrite the previously

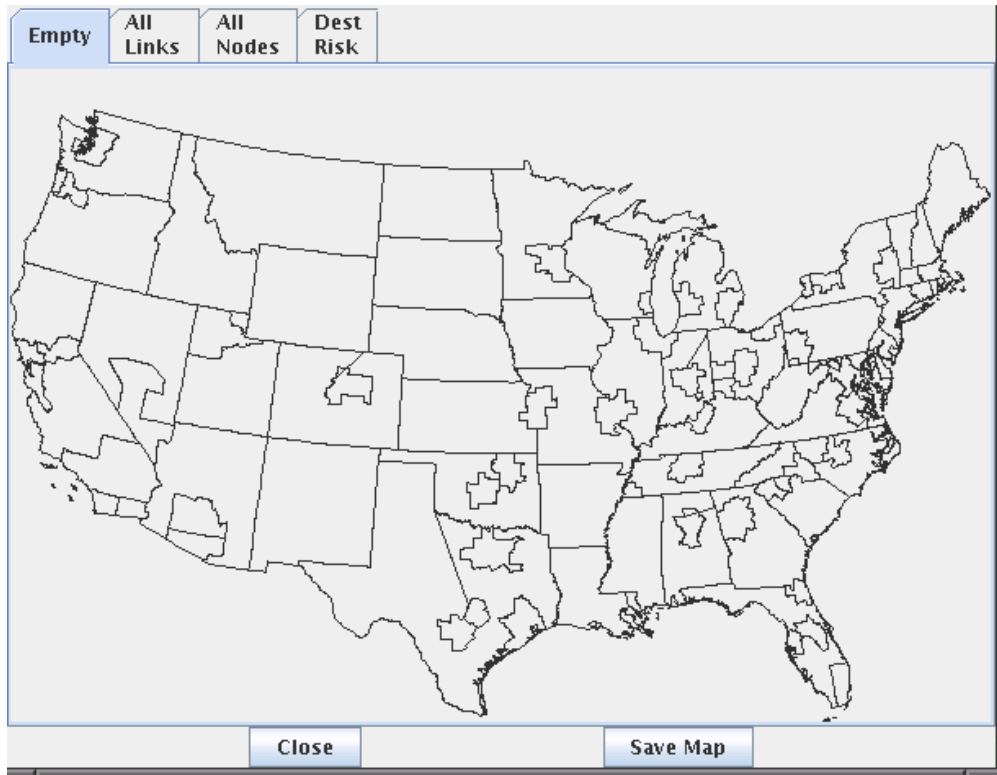


Figure 9: Map Selection Window

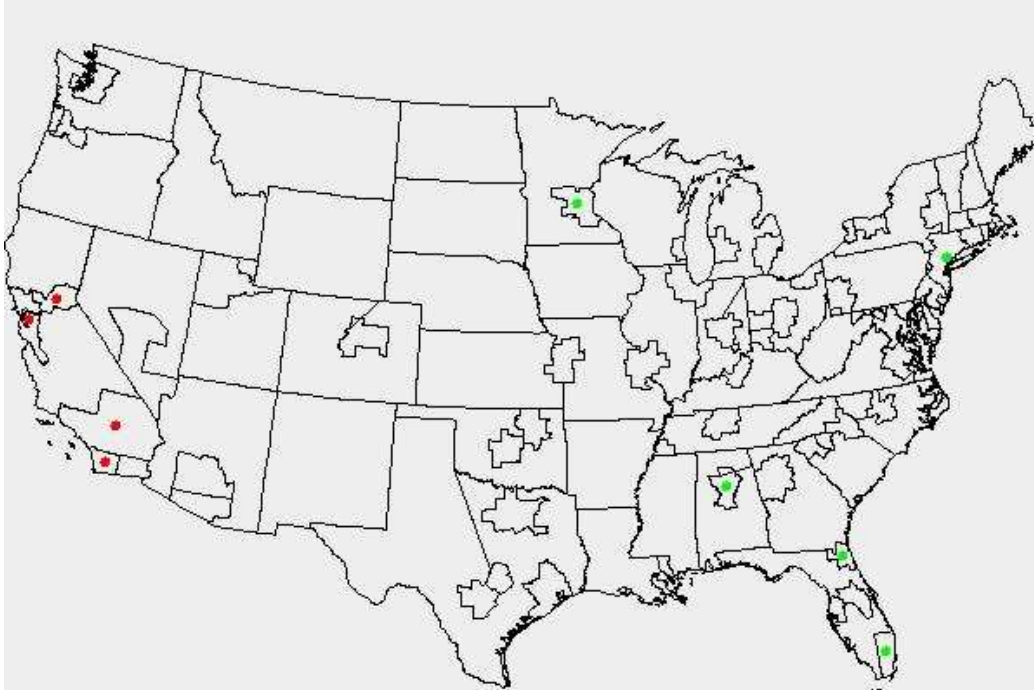


Figure 10: Example of a Node Map

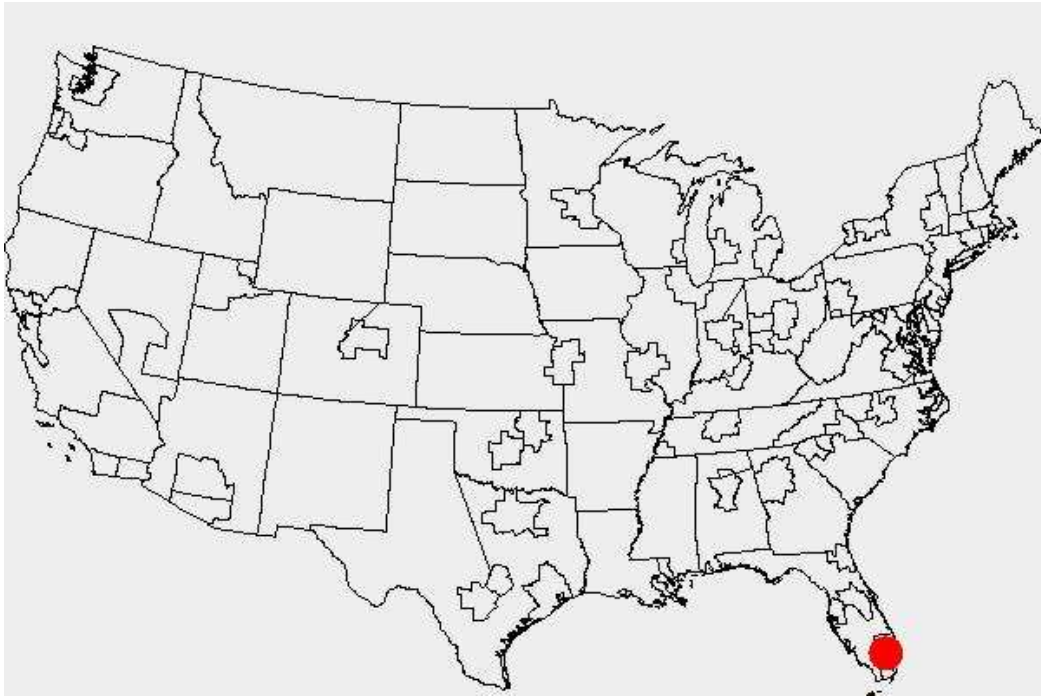


Figure 11: Small Scale Example of a Risk Map

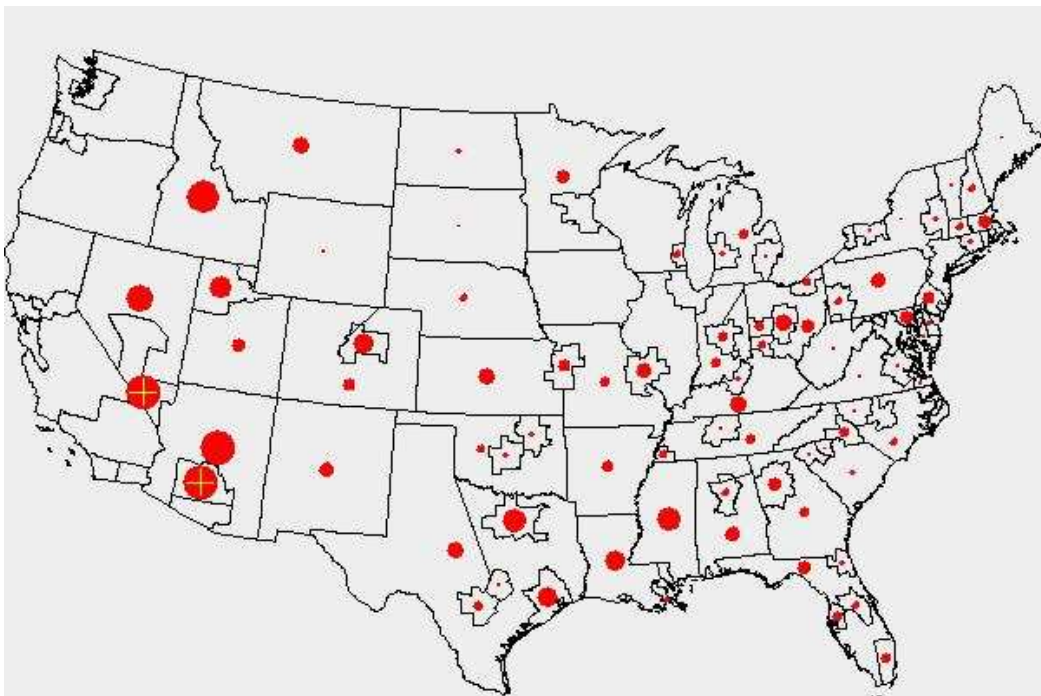


Figure 12: Large Scale Example of a Risk Map

created CSV file. If a user wants to maintain all the CSV files, it is necessary to rename a CSV file before a subsequent program run overwrites it.