Real-Time Biosurveillance Program

T-Cube Web Interface

Quick Reference Manual

Revision 1.2 DRAFT
November 27, 2009

Edited by Maheshkumar Sabhnani and Artur Dubrawski

sabhnani@cs.cmu.edu, awd@cs.cmu.edu

Carnegie Mellon

Auton Lab

Table of Contents

1. Introduction

This manual is intended as a quick introduction for public health officials and epidemiologists – the end-users of T-Cube Web Interface (TCWI). It is relevant to the specific version of TCWI tailored to the requirements of the Real-Time Biosurveillance Program (RTBP), conducted as a pilot in India and Sri Lanka. The TCWI as well as the underlying data representation and analytic technologies have been developed by the Carnegie Mellon University Auton Lab, in Pittsburgh, Pennsylvania, the United States of America.

TCWI is a front-end for the public health databases collected through RTBP. It allows for visualization, statistical analysis and navigation through data. It uses modern computer science, statistics and machine learning technologies to support public health analysts and epidemiologists in their daily duties, including monitoring for disease outbreaks, outbreak investigations, and reporting.

The underlying technology provides the users of TCWI with unique abilities to very quickly navigate through and mathematically analyze large amounts of data even if it spans highly multidimensional spaces of multiple diseases, locations, symptoms, syndromes, and demographic factors. The users of TCWI benefit from it by becoming able to concurrently monitor many more hypotheses about the status of public health much more thoroughly than it was possible before. The data can be comprehensively mined for statistically significant increases of counts of specific subpopulations reporting specific symptoms, with respect to baselines inferred from historical trends. The results of such automated massive screenings are presented to the users in a form of the list sorted according to their statistical significance. The users can then navigate through the list and easily drill-down into details of the corresponding data for additional hints and explanations. A skilled operator can use TCWI to support maintenance of high levels of awareness of current epidemiological situation and of ongoing processes that affect monitored populations, enabling fast and reliable identification of emerging problems and creating opportunities for implementing focused and effective responses to crises.

The following sections briefly review the essential functionality provided by TCWI. The ideal user of TCWI should be fundamentally computer literate and be familiar with fundamental methods of analysis of public health data and with the objectives of such analyses.

While RTBP would provide their respective users with a web link (URL), other first-time users may access a demonstration version of the TCWI available on the Auton Lab site (http://www.autonlab.org/T-Cube/) for the purpose of hands-on experimentation with the features and functions described in this manual.

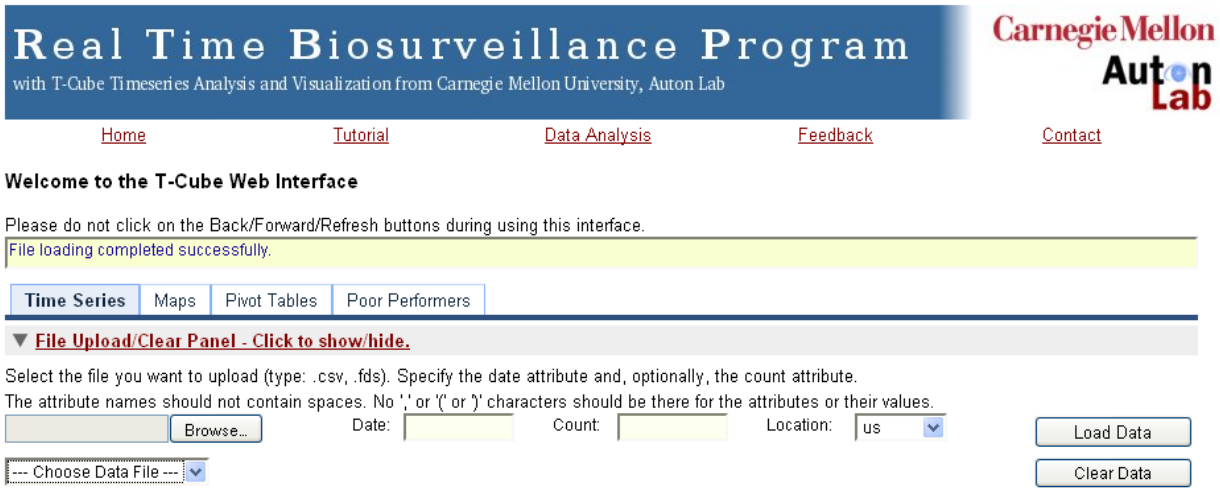

Figure 1: Front panel of the T-Cube Web Interface.

Figure 1 shows the starting panel of the TCWI with the project header and tabs for panels containing specific analytic components: **Time Series**, **Maps**, and **Pivot Tables** (note: tab labeled "Poor Performers" is irrelevant to the RTBP project). Each analytic component is described in a separate section of this manual: Time Series Analysis, Maps (Spatio-Temporal Analysis), and Pivot Tables, respectively.

Immediately below the header, there are links for specific sections of the TCWI. Figure 1 shows the **Data Analysis** section of the interface. In this manual, we will focus on the data analysis section. The **Feedback** section can be used to send direct feedback to the TCWI developers. The **Tutorial** section points to an older version of this manual and it will be updated soon. For now, please refer to this document as the current reference manual.

2. Time Series Analysis

Time series analysis component of TCWI supports both univariate (Analysis Panel) and multivariate temporal analysis (Massive Screening Panel). Each of the five panels under Time Series Analysis is described below.

2.1 File Upload/Clear Panel

This panel allows the users to select data for analysis. The data can be either loaded from disk on the local machine (advanced usage) or it can be preloaded on the server (typical usage).

Using Pre-loaded Data (Typical Usage)

To use a pre-loaded data, just click on the **Choose Data File** to expand a drop-down list of available data files and pick one for analysis. This tutorial uses **lk_flat_table** data as an example (at the moment of writing this revision of the manual we did not have access to an equivalent set from India, but the functionality of TCWI will be identical when used on data collected in India). This data contains a total of 69,000 reportable disease cases from multiple regions in the country of Sri Lanka collected starting December 16, 2006 and ending on July 10, 2009. It spans the following fields: **loc_name**, **age_grp**, **disease**, **gender**, **sign**, and **symptom**. Note that the original data contained weekly aggregates for each combination of **loc_name** and **disease**. We have semi-synthetically converted this data into daily resolution with more demographic attributes to match the scheme and level of detail of data being currently collected in through the RTBP project.

Loading External Data (Advanced Usage)

The users of TCWI can load their own data instead of using the pre-loaded data sets. This mode of usage is not expected or considered typical in the context of the RTBP project, since the relevant data will be prepared for use and frequently updated through automatic processes set up by the maintainers of the system. We explain it here for the sake of completeness.

TCWI accepts external files in the comma-separated values (csv) format. The first line of it contains names of the data dimensions. The subsequent lines contain the actual data, each record denoting either an individual disease case (in the purely transactional format of data), or it represent a cumulative count of disease cases matching the specific combination of values of dimensions represented in the record (the cumulative count format of data). A top fragment of an example of a TCWI-compatible file in the cumulative count format is shown below. It has four dimensions named *date*, *lk_city*, *disease*, and *count*. Each record reveals the number (*count*) of specific reportable *disease* cases reported in Sri Lanka region labeled with its main city (*lk_city*) on a given day (*date*). If the count dimension is not available, TCWI data loader will assume that each record represent a single case (that is that its count value equals 1), and so that the data is provided in the purely transactional format.

```
date,lk_city,disease,count
DEC-16-2006,Colombo,Dengue_fever,7
DEC-16-2006,Colombo,Dysentery,1
DEC-16-2006,Kandy,Dengue_fever,1
DEC-16-2006,Kandy,Dysentery,2
DEC-16-2006,Matale,Viral_Hepatitis,1
DEC-16-2006,Nuwara_Eliya,Viral_Hepatitis,1
DEC-16-2006,Galle,Dengue_fever,1
DEC-16-2006,Hambantota,Typhus_fever,1
DEC-16-2006,Matara,Dengue_fever,3
DEC-16-2006,Matara,Leptospirosis,1
DEC-16-2006,Vavuniya,Dengue_fever,1
DEC-16-2006,Kurunegala,Dengue_fever,2
DEC-16-2006,Kurunegala,Typhus_fever,1
DEC-16-2006,Puttalam,Encephalitis,1
```

To load an external data set, use the **Browse** button (cf. Figure 1) to interactively select the file to load from the disk on the local machine. Once the location of the file is specified (for instance: C:/mydata/myfile.csv), the user needs to identify which of its dimensions represents the date dimension and enter its name in the field next to **Date** (in the above example, the user would have entered “date”). Similarly, if the individual records of data represent multiple counts (as in the example above), the user needs to provide the name of the count dimension by entering it in the field next to **Count** (that field is named “count” in the above example**)**. Purely transactional data, in which each record represents a single unique event, does not require the count dimension and therefore the **Count** entry field can be left empty.

If the data contains a spatial dimension (in the example above, the spatial column was *lk_city*), then the user must specify the type of map to be loaded using **Location** drop-down list.

Once all required information for loading external data is specified, click on the **Load Data** button to load data into TCWI. Depending on the current workload of the TCWI server and the complexity and size of the dataset being loaded, this operation may take some time and patience is advised. The investment in waiting time will be returned many times by the resulting speed of responses to user and algorithm queries against the loaded data.

It is not a requirement, but a good habit, to clear computer memory of data that is no longer in use, before loading a new data set. To accomplish that click on **Clear Data** button at any time, and then proceed to loading new data as needed.

Note that TCWI expects the external data to meet the above described format requirements (specifically, all dimensions should be categorical, except for the count and date dimensions), and it should not contain any missing values. The interface will signal an alert if an attempt is made to load an incompatible data set.

2.2 Query Selection Panel

Query selection panel incorporates drill-down functionality to filter data by selecting subsets of values of individual dimensions. It also enables interactive visualization of multiple time series. Below we briefly introduce two basic modes of its operation. This description is not comprehensive, but the functions not addressed here are intuitive and accessible via obvious user interactions.

2.2.1 Filtering Data by Selecting Subsets of Values of Individual Dimensions

Figure 2 shows an example view of the query selection panel involving the kind of data typically used within the RTBP project. Using value filtering, the user can select a subset of values for each dimension of data, and the corresponding time series labeled as **Current Query** will be automatically updated in the time series analysis window, as shown in Figure 5. For convenience, TCWI automatically visualizes the time series resulting from the all-inclusive query called **All Data**. It reveals the temporal distribution (by day) of all disease case records stored in the currently loaded data. Any other, more specific query, will result in a subset of **All Data**.

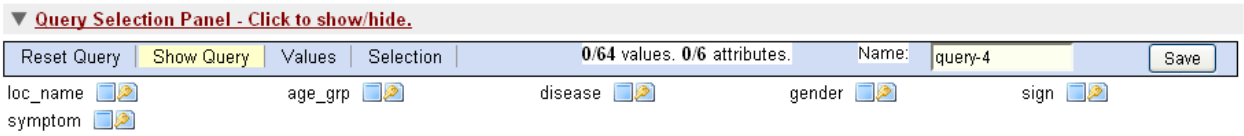

Figure 2: Starting view of the Query Selection Panel.

There are two modes in which the user can select the subsets of values of individual dimensions: **list view** and **tile view**, accessible by clicking on one of the small icons next to each dimension name (Figure 2). In the **list view** (see Figure 3) the values of the selected dimensions are presented in the check-list form and the user can select any subset of values. **Tile view** mode (Figure 4) offers more extensive view, in which the users can filter for specific values using regular expressions.

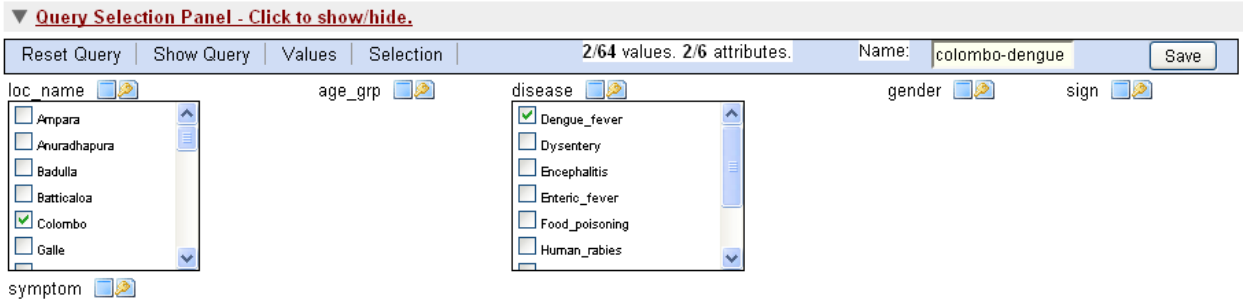

Figure 3: List view of the Query Selection Panel.

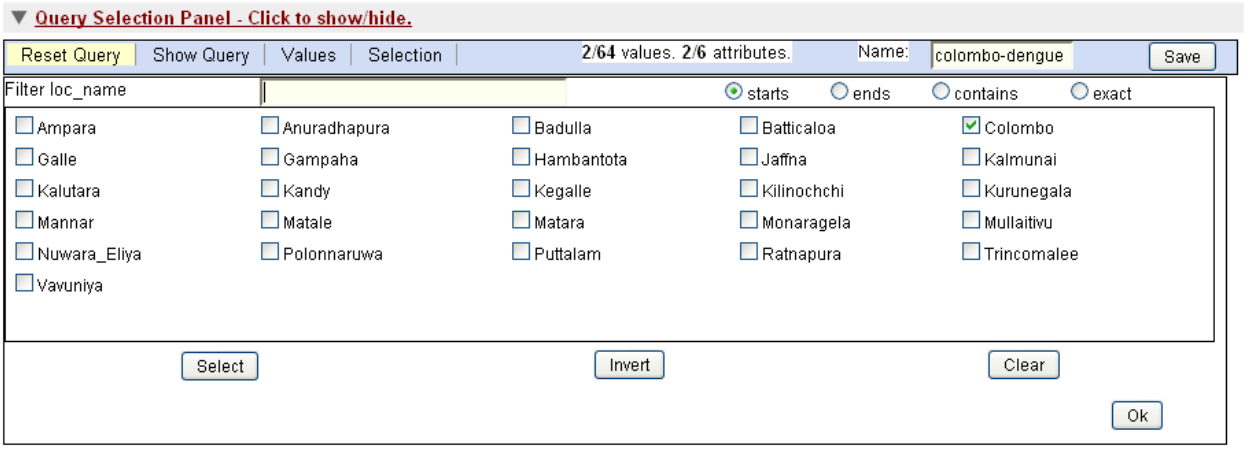

Figure 4: Tile view of Query Selection Panel.

Each time the user updates their choices in the query selection panel, the TCWI extracts the daily counts of disease cases corresponding to the user-selected set of values, and the resulting time series is shown in the visualization window under the name of **Current Query**. The user can optionally save the time series corresponding to the current query by giving it a unique name by entering it in the **Name** text box. The time series visualization window will then add it under the new name to the list of available series. For example, in Figure 5, the *red* time series corresponds to temporal distribution of counts of *Dengue fever* cases in *Colombo* and is named "colombo-dengue". It was obtained by manually selecting Colombo and Dengue_fever values of the *loc_name* and *disease* dimensions through the query selection panel. Similarly, the *green* time series corresponds to the total number of patients in all regions and diseases. The user has named this series "all-values". The **Current Query**, shown in *blue*, represents in this example the daily number of Dengue fever cases collected across all data – this matches the current selection of values and dimensions in the query selection panel.

Note that in the case of the *lk_flat_table* dataset, the dimensions named *sign* and *symptom* are available to the user in a composite form. .Each disease case can be associated with multiple signs and symptoms, and therefore each individual value of a sign as well as symptom variable is represented internally as a separate binary dimension. In order to avoid cluttering the TCWI panel with multiple binary dimensions of data, it collapses their groups together. When the user selects one value of such a dimension for their query, all records for which that value is present will be reflected in the result. If the user selects two distinct values of e.g. *symptom* the response to such query would include all records mentioning both of the two symptoms. This is different than in the case of regular (not composite) dimensions, where selecting two values leads to concatenating the records of data which match either of the selected values, so that the resultant is a sum of the two subsets matching the individual values. On the other hand, two or more values selected out of a composite attribute on the other hand would produce the joint part (the product) of the individual value matching subsets – only records matching both of them will be

reported. Currently, only *sign* and *symptom* are represented in the composite form. Future release of TCWI will allow the users to select one of the two types of representations for each qualifying attribute.

2.2.2 Visualization of Time Series

Figure 5 shows the Time Series Analysis panel used for visualization. It shows a couple of time series previously queried for, extracted, and named by the user. Their display status can be toggled on-and-off using the check boxes next to their names in the legend to the left of the graph.

The horizontal axis of the main time series visualization window denotes time at a daily resolution, arranged from the oldest (left) to the newest entries (right). The vertical axis is primarily used to reference the values of the daily counts corresponding to the displayed time series. Its secondary purpose is to reference the magnitude of alerts generated by the statistical algorithms for event detection, discussed later. The scale for daily counts is shown at the left side of the main time series window, while the scale for alerts is on the right. The scaling of counts axis can be controlled via the **Scaling** dropdown menu. The scale of the magnitude of alert signal can be toggled between linear and logarithmic display (note that no alert signal time series are shown in Figure 5, this functionality will be explained later).

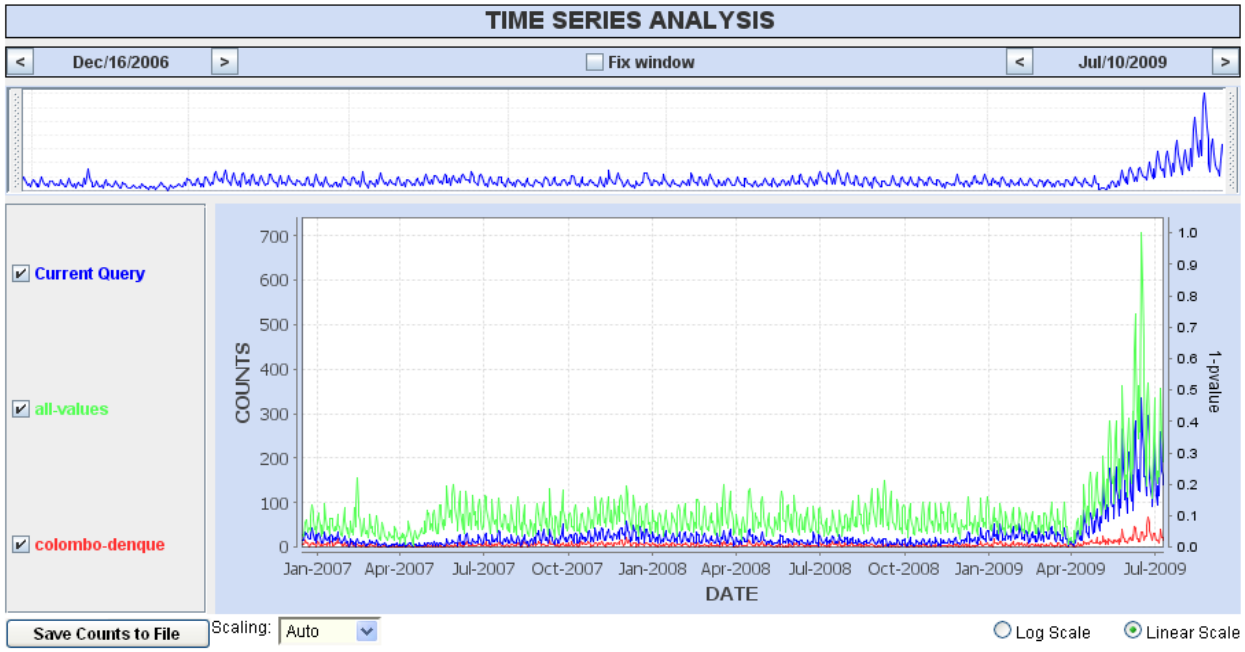

Figure 5: Visualization of time series.

The user can change the temporal range of time series currently being viewed in the main window by either changing the dates on the top of the window (Dec/16/2006 and Jul/10/2009 in Figure 5), or by dragging with the mouse the bars right below the dates. The upper smaller time

series window always displays the full temporal range of the cumulative daily counts of all records in the current data. The movable bars in it depict the current selection of the temporal field of view of data being displayed in the main window. The users can fix the current width of the temporal field of view by checking the **Fix window** box. When it is checked, the field of view can be dragged left and right by dragging the right bar with the mouse.

The users can save the time series currently in the field of view to a local csv (comma separated values) file using **Save Counts to File** button. Saved files can then be further processed by external software. An upcoming revision of TCWI will enable the users to look up the raw data and save its subset corresponding to a query to a disk file. It will allow performing follow-up analyses outside of TCWI.

The appearance of the individual time series plots can be adjusted by mouse-clicking on their respective names in the legend. The users can change the color, thickness, type, and stroke of each plot (Figure 6).

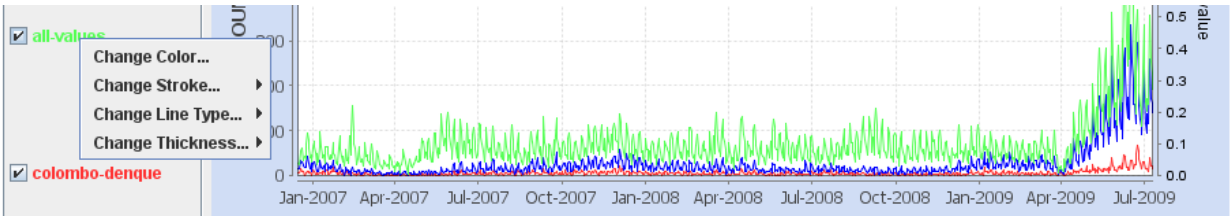

Figure 6: Changing the appearance of the time series plots.

2.3 Analysis Panel

This panel provides an interface with a number of the univariate statistical analysis algorithms that can be applied to any of the currently visualized time series. The currently implemented analytic algorithms are briefly explained in the following subsections.

The **Analysis Panel** can be activated by selecting one of the time series listed by name in a small window under the **Select Target** heading (Figure 7). The panel implements a dialog allowing the users to:

1. Select the time series for analysis (from the list under **Select Target** header),
2. Select the method of analysis (**Choose Method**),
3. Set parameters of the analysis (the set of parameters is specific to each particular algorithm),
4. Indicate whether the result should be stored separately, or the analysis is just an update of prior result (**Create new** vs. **Update**),
5. Assign or reassign a name to the result of the analysis (**Series name**),
6. Select the option to automatically update the result upon a change to the target time series (**Auto update on target change**).

The Analysis Panel also allows the users to remove the selected time series from the list of currently considered and visualized series by using the **Delete Series** button, or to change its

name (**Rename** button and text entry window). Note that the **Current Query** and **All Data** series cannot be deleted or renamed.

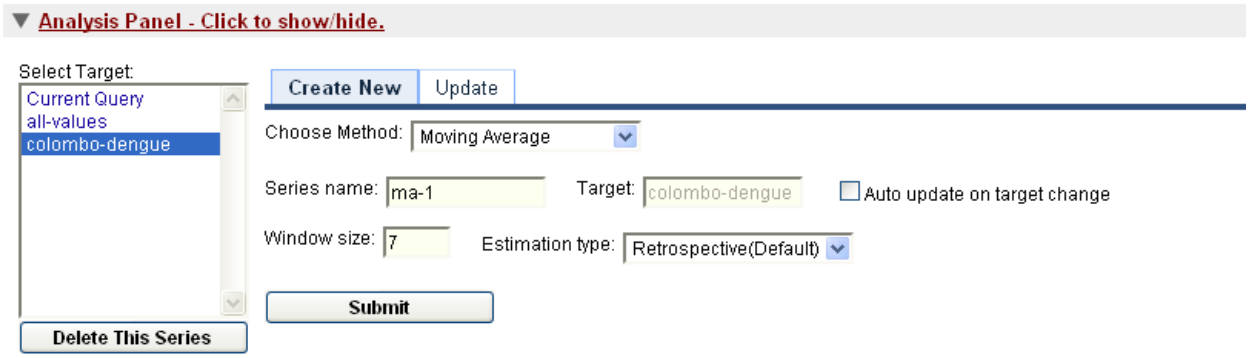

Figure 7: Analysis Panel

2.3.1 Time Series Modeling and Forecasting Functions

TCWI currently implements a small set of select univariate forecasting functions. They are useful in smoothing noisy data, and in predicting the counts of disease which should be observed on a given day using historical data. A forecast can be useful under assumption that the modeled processes are predictable and that the selected forecasting algorithm can capture their dynamics. In health surveillance applications, forecasting functions are often used to compute baselines for detection of statistically significant discrepancies between the predicted and actually observed counts of disease cases.

Moving Average

Moving average is a simple and popular method of smoothing "bumpy" time series. Given e.g. a time series of daily counts of disease cases selected for smoothing, for a search time step (each day in our case) the moving average algorithm computes the average daily count over the past few days (excluding "today"). The extent of smoothing can be controlled by choosing the number of days to average over (**Window size**). Note that in most practical applications related to public health it is advisable to select window sizes among multipliers of 7 days. That accounts for typically strong day-of-the week effects present in data.

The **Window size** is the only parameter of the basic moving average algorithm. The basic algorithm is offered as default among other variants and named **Retrospective (Default)** in the **Estimation type** pull-down menu.

Alternatively, the moving averages can be computed for each day using only data observed on the same weekdays from the past (that is, if the analysis is performed for a Tuesday, only data from Tuesdays it the past will be considered in computing the forecast for this Tuesday). That can be accomplished by selecting **Moving Average (day of week)** from the **Estimation type** list. In this context, the **Window size** parameter still determines of how many such days from the past should be taken into consideration in computing the daily moving average estimates. This

algorithm is often useful in practice to model longer-than-a-weeklong trends in data which is subjected to a strong day-of-the-week bias.

Regression

The forecast produced using the day of week algorithm explained above can be correct if the modeled time series is stationary (that is, if its daily counts distribution does not change over time, except due to random fluctuations). Running the moving average algorithm using **Regression (day_of_week)** estimation type allows to account for linear non-stationarities (trends) in data. For each day, it takes the daily counts from the number of past identical weekdays (specified under **Window size**), fits a linear regression model to this data, and sets the today's value of the resulting time series to the value predicted from that model. Running the moving average algorithm using **Regression (day_of_week, last_week_mean)** extends that approach to take into account the mean count of the last week of data as a separate covariate in the linear model.

Note that both moving average and regression-based algorithms can be used to forecast expected counts of disease cases pertinent to time series selected by the users for monitoring. In turn, these forecasts can be compared against the currently observed actual volumes of disease cases to verify if they may be significantly excessive with respect to expectations, and therefore indicative of a potential disease outbreak. The extent of discrepancy between the actual and forecast daily counts can be easily computed using time series arithmetic operations (Section 2.3.2) by subtracting the original target time series from the time series resulting from application of one of the moving average algorithms.

Moving Sum

Moving Sum algorithm is analogous to the above described moving average, except that daily counts selected for consideration are simply added together instead of being averaged.

Linear Trend

Linear trend method fits a linear regression model to the target time series. Users can optionally use day of week feature to compute a set of seven models, each independently dealing with data from one day of the week. The users can also de-trend their time series by computing the linear trend and subtracting it from the target time series. A de-trended time series is then created and it can be used in further processing.

Arithmetic Operations

The users can perform fundamental arithmetic operations (add, multiply, divide, and subtract) on any two target time series. The result becomes a new time series added to the Time Series Analysis panel. An example use scenario of this functionality is to compare results of a forecast against the true data. To experiment with that, create a forecast time series using moving average

function with an estimation type of your choice. Then, apply **Subtract** operator selecting the forecast series as the **Target** (this can be accomplished by clicking on the forecast time series name on the list under **Select Target** header), and choose the original time series for which you have computed the forecast as **Target 2**. Hitting the **Submit** button will produce a time series of the daily forecast errors.

2.3.2 Temporal Anomaly Detection Functions

Temporal Scan

Temporal scan is a bi-variate algorithm for detecting anomalies in time series. It therefore requires the users to specify two time series as inputs: the **targe**t and the **baseline**. It is sensitive to statistically significant changes in counts of target time series that cannot be statistically explained by the corresponding changes in the baseline.

Fundamental procedure of the temporal scan algorithm executed for one period (day) of analysis aggregates four sets of counts. One set is the sum of target time series counts observed during the period identified as current (user-selectable **Temporal scan window size**). The other set is the sum of target counts corresponding to the period of reference. The remaining two aggregates are the identical counterparts computed for the baseline time series. The results of aggregation can be put in a 2-by-2 contingency table such as the one shown in Table 1.

Table 1: An example of a contingency table obtained for a single test in the Temporal Scan procedure.

Counts	Current	Reference
Target	23	847
Baseline	95	11,550

Table 1 shows example results of aggregation of target and baseline counts for one day of analysis. Looking at the proportion of target and baseline counts observed during the period of reference, and comparing it to the same proportion observed currently, one can notice that currently it is substantially elevated. TCWI uses Fisher's exact test of significance to quantify the extent of surprise in the observed increase. The result is a p-value – an estimate of the probability that the observed counts can be a result of a random fluctuation of data. The lower the p-value, the slimmer the chance for the observed elevated current counts to be just accidental. The p-value computed for data in Table 1 is close to $8*10^{-8}$, a very small number indicating that the observed increase is highly significant and very hardly explainable as an effect of the random chance.

Temporal scan algorithm executes the above prescribed procedure independently for all days in the scope of analysis. The interface offers a few different methods of computing the reference counts (selectable from the **Estimation type** pull down list). Most **Estimation type** options have

been introduced above in Section 2.3.1 in the context of forecasting functions. The **Retrospective (Default)** method uses all the data outside the "Current" period as "Reference". The **Prospective** method aggregates as reference the counts over the number of days, specified in the **Estimation window size** dialog, that immediately precede the current window (excluding the current window). Forecasting-like estimation types explained in Section 2.3.1 are also allowed as the methods of reference counts estimation in temporal scan. In addition, the **Univariate** variant does not require separate time series to be designated baseline. Instead, temporal scan uses the global average of target series counts as the outside counts of reference, turning the algorithm into a univariate anomaly detector. It may become handy when the baseline events are rare, their frequency not exceeding the counts of the target series, and the attainable reference information may therefore be rendered unreliable.

The users can execute temporal scan aiming at any kind of departure from expected, or strictly constrain the detector to alert for either increased or decreased counts. That is controlled with the **Scan option** through which the users specify whether to use a **Two-sided** test (detects all discrepancies, without differentiating between increases and decreases of activity), or **Upper tail (**focuses on monitoring for increased activity), or **Lower tail** (targets decreases in activity).

Users can also define the threshold of sensitivity of the alerting procedure. It is set to 0.05 by default, but it can be manually adjusted. The days for which the p-value obtained from temporal scan procedure was lower than that threshold are considered days of alert. The users can also apply False-Discovery-Rate (FDR) algorithm to automatically adjust the sensitivity threshold, guarding against the multiple hypothesis effects. The FDR-selected thresholds are usually more conservative than the manually selected ones, leading to fewer alerts being raised. Note that FDR does not affect the individual p-values computed using the Fisher's test. It only selects an alternative threshold to pick potentially fewer of results from the top of the ranked list of sorted from the most to the least statistically unusual.

Figure 8 shows the results of executing the temporal scan algorithm using the familiar *colombo-dengue* as the **target** and *all-values* as the **baseline** time series. We expect it to produce low p-values on days when temporal changes in the number of dengue fever cases reported in Colombo region could not be explained by similar fluctuations in all kinds of disease activity recorded across the country of Sri Lanka (other scenarios may involve comparing Colombo cases of dengue against the counts of all other reportable diseases recorded in Colombo, or the Colombo dengue activity against the temporal distribution of country-wide aggregates of dengue cases).

The result of executing temporal scan is a time series of p-values. After completion of the computations, this time series is automatically visualized in the time series visualization window. The plot depicts complement of p-values (=1.0 - p-value) for all days on which the resulting p-value was lower than either the user-selected or, when activated, the FDR-based sensitivity threshold. In Figure 8, days like that are marked with blue spikes, and there is no signal shown

on days when the p-values were greater or equal to the threshold. It is often convenient to toggle the display of the p-values to the logarithmic scale to see more minute differences between the individual alerts.

Here are a few recommendations for setting temporal scan parameters. Temporal scan window size should be set to a multiplier of 7 days to remove weekly trends from data. Until substantial amounts of historical data is collected, we recommend using Prospective estimation type with estimation window size set to 35 days (5 weeks) and temporal scan window size set to 7 days. Longer scan windows enable detection of slower growing or generally longer-term problems. Longer estimation window sizes lead to taking into account more historical data when making forecasts. That will be a desirable strategy when the data collection process and therefore the data quality have stabilized. To reduce the false positive rate, always apply FDR correction. Given that the most common use of TCWI analytics would involve detecting unusual increases in disease case counts, we recommend performing primarily the upper tail tests. The lower tail tests would detect unusual decreases in certain disease counts. That may be useful in monitoring effects of responses to public health threats or in evaluating effects of longer term policy changes. . Also, use the last day scan option to find alerts related specifically to the latest date in the data.

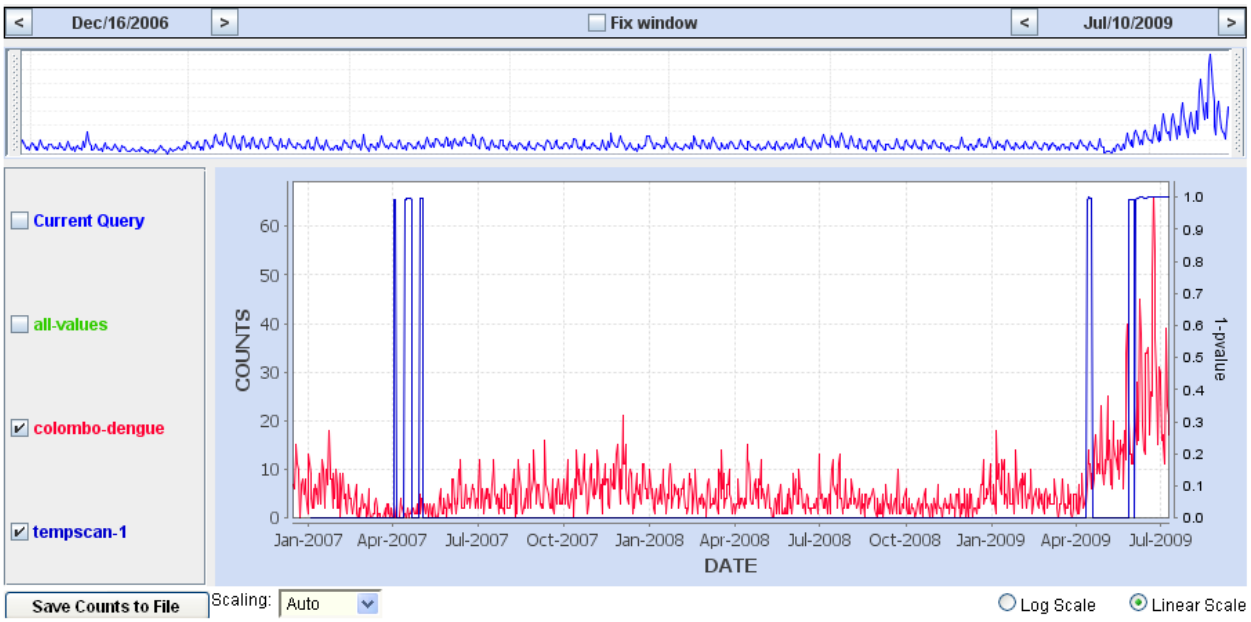

Figure 8: Example result of executing temporal scan on Colombo dengue fever data.

CUSUM

CUSUM (Cumulative Sum) control chart is a popular method for monitoring temporal processes primarily for changes in level. It is robust against short-term and presumably random fluctuations. It is being widely used in the practice of disease surveillance. TCWI implements the

basic uni-variate CUSUM algorithm. For each consecutive time step, it computes the excess of target time series counts over the expectation. The expectation is computed over the number of past days specified using **Cusum Period (in days)** dialog, as a sum of the mean counts over that period. If the counts observed today exceed the sum of mean and *K* standard deviations (*K* is the tolerance multiplier setting specified by the user), the cumulative sum is incremented with the value of the observed difference and the algorithm moves to processing the data from the next time step. Initially, the cumulative sum is set to zero. Whenever the cumulative sum exceeds a user-selected threshold, *H* standard deviations, and whenever the sum goes below zero, it is set back to zero, and the accumulation process starts over at the next time step. As a result, TCWI currently produces and displays a time series of the cumulative sum. The upcoming version of software will also produce a time series of alerts marking the days when the cumulative sum exceeded the running threshold. The optimal settings of the CUSUM algorithm parameters depend on the amount of variance in data and of desired sensitivity. Greater values of the tolerance multiplier and threshold lead to fewer alerts and therefore to detecting only more spectacular changes in the level of counts of disease. Selecting longer periods of aggregation (via estimation window parameter) typically leads to more stable estimates of variance used to determine the running threshold and tolerance, but that may also lead to undesirable suppression of sensitivity of CUSUM to temporally local fluctuations of variance in data. It is advisable to experiment with a few different sets of parameter values to empirically determine the most suitable set-point.

Change Scan

The Change Scan method is similar to temporal scan in that it uses the 2-by-2 contingency to accumulate counts of disease cases and to evaluate statistical significance of the observed departures from the expected. The main difference is that Change Scan compares counts observed before and after the day of analysis, while temporal scan focuses on counts “inside” and “outside” the current period of analysis. Therefore, the change scan is well suited for detecting days of change of the level of the monitored time series, similarly to CUSUM algorithm. However, unlike CUSUM, Change Scan is bi-variate – it allows using baseline time series as the reference of comparison against the monitored target. It also allows for estimating reference counts in various ways, accessible through the **Estimation type** selector, and it can be set to focus on detecting either positive or negative changes, or both kinds of them, analogically to temporal scan.

Peak Analysis and Range Analysis

Peak analysis method can be used to explain peaks observed in the target time series. User specifies the peak date and the algorithm screens the data for dimensions and their specific values which jointly contribute to at least 90% of the counts observed in the peak. It is often useful in explaining characteristics of data that might correspond to a disease outbreak.

Range Analysis performs similar computations, but with respect to a range of consecutive dates, instead of a single “peak” day.

2.4 Massive Screening Panel

The above described forecasting and anomaly detection methods are useful and applicable when the users know a priori which specific time series queries are of interest. This includes the above examples of monitoring of dengue cases in Colombo. However, when dealing multidimensional data, the number of possible unique projections of it onto subsets of dimensions and subsets of their values (and therefore the number of the extractable time series), can be very large. Analyzing them a one-by-one in a sequence could become a quite tedious task.

Massive screening approach applies to exactly such scenarios in which the analysts need to concurrently monitor multiple time series. It exhaustively checks all individual time series resulting from conjunctive queries fitting in the user-selectable scope of search, and reports the findings in the form of a list of time series sorted according to the statistical significance of temporal anomalies found in them. The elements of the resulting list are clickable links to results visualized in the Time Series Analysis window. The suggested usage pattern is to first execute a massive screening of the desired subset of data, and then to inspect results starting from the most statistically surprising, one-by-one, possibly drilling down the data for further explanations using interactive visualization capabilities of the Query Selection Panel, or the explanatory analysis functionality such as Peak or Range Analysis functions described above.

Massive screening provides the analysts with the ability of comprehensive monitoring of multidimensional data without having to make many assumptions regarding the expected impact of anticipated disease outbreaks. It allows for bringing their attention to specific patterns in data such as subpopulations of patients who appear in recent data at unusually high frequencies, which might otherwise go unnoticed.

Implementation of massive screening in TCWI uses the previously described methods of temporal scan and change analysis as the core anomaly detection methods.

The users select **Start Date** and **End Date** parameters to specify the period of analysis, and **Scan window sizes** to specify temporal granularity of the “Current” windows in the scan. **Pvalue threshold** parameter is used to decide what level of statistical discrepancy warrants an issuance of the alert. The higher the p-value threshold, the higher the sensitivity of event detection procedures, and the more alerts would be generated. The lower the p-value threshold, the more conservative filtering of the results, and the fewer the alerts. **Estimation type** selector allows the users to define the method of estimating “Reference” counts in the 2-by-2 contingency table used to determine p-values. **Scan Option** allows the users to select the type of the significance test (two-sided vs. one-sided upper- or lower-tail). **Last Day Only** flag is used to complete the

analysis only for the selected **End date**. This can be used for prospective surveillance when the users are only concerned if there are any alerts today.

Current implementation of the massive screening algorithm allows the users to pick up to three dimensions and any subsets of their values for screening. The algorithm will try every query within the scope of this selection, derived as a conjunction of individual values taken from 1, 2, or 3 dimensions (if the users selected 3 attributes for screening).

There are a few additional parameters that could be applied to control massive screening algorithm, but they are not fully applicable to the current shape and contents of the RTBP datasets. We will keep monitoring their utility as more data becomes available and either include full description in the next revision of this manual, or eliminate the corresponding functionality from TCWI.

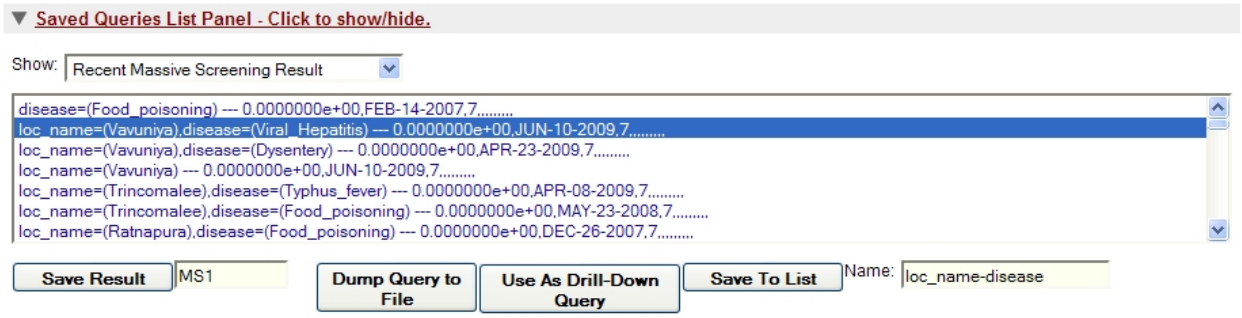

Figure 9: Saved Queries List Panel

Figure 9 shows the result of running temporal scan using default parameter values, except for the **Scan option** set to **Upper tail**, and selecting “loc_name” and “disease” as dimensions for analysis. The list appears under **Saved Queries List Panel** described below. The top alert is for disease “Food_poisoning” on February 14, 2007. The p-value of the Fisher’s exact test conducted for that day was lesser than the numeric precision of the computer representation. Equally extremely significant score was associated with a few more results shown on the top of the list.

The second result shows an alert about Viral_hepatitis in Vavuniya that would have been issued on June 10th 2009. Clicking on it brings up the corresponding data to the time series visualization panel. Figure 10 shows the screenshot presenting that result as well as the complete set of parameters used to arrive at it. The time series legend has been expanded by adding three items: **MS Baseline** is the temporal distribution of the data corresponding to query used as the baseline in the temporal scan procedure (in this example, we chose **All Data** to serve as the baseline, but the users can select any other baseline query to use in massive screening). The **Result Query** depicts time series of the target that is counts of viral hepatitis cases reported in Vavuniya. The 7-day long temporal scan window corresponding to the alert of June 10th is highlighted with

yellow background. Zooming in onto the period of interest we can see the results in more detail (Figure 11). We can see the number of the hepatitis cases ramping up significantly in early June.

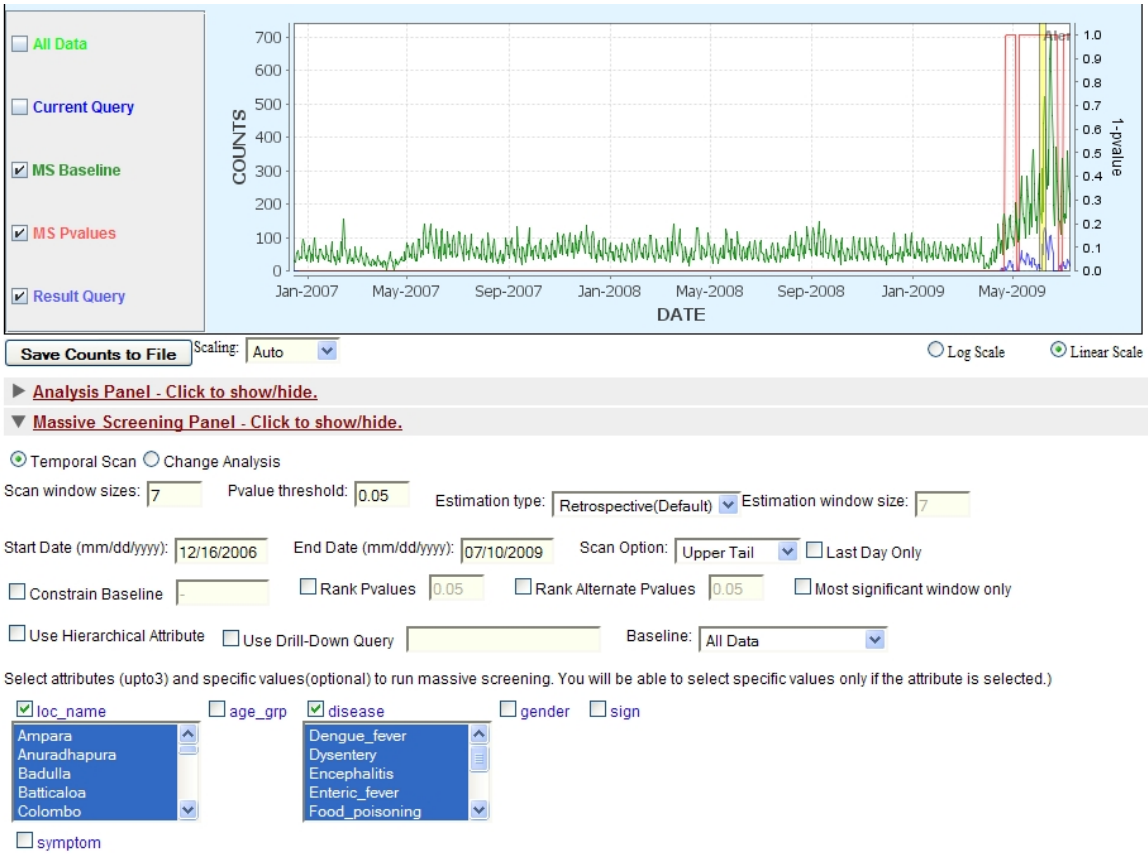

Figure 10: Inspecting one result of the massive screening with temporal scan.

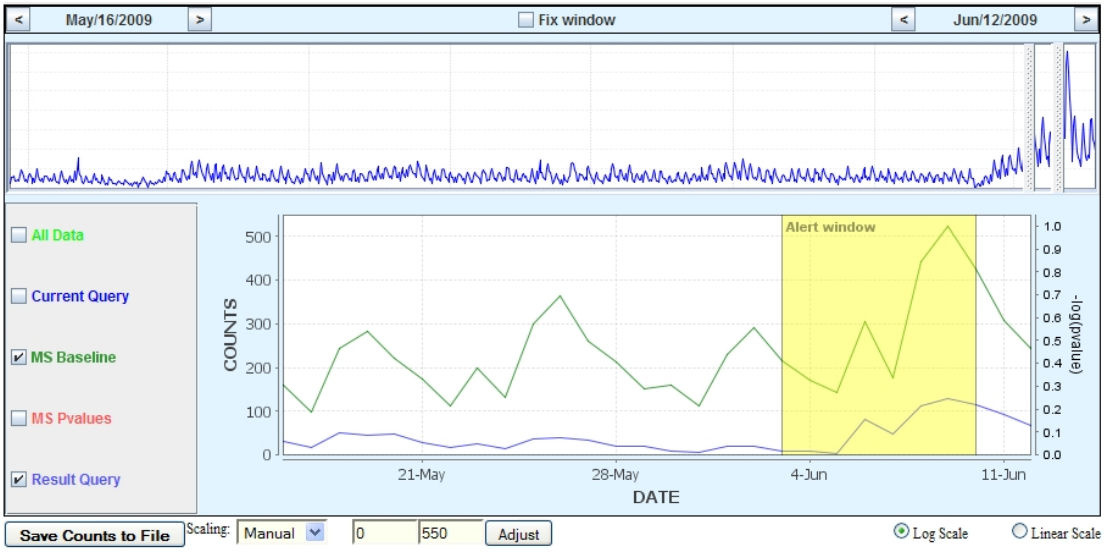

Figure 11: Zooming in on the results from Figure 10.

2.5 Saved Queries List Panel

This panel allows manipulation of the results generated by massive screening. **Save To List** button saves the results for future retrieval with a specified name. **Save Query** button saves the query as a .csv file on a local disk. **Use As Drill-Down Query** button allows to stage the next run of massive screening within the scope of data corresponding to the specific item on the finding list. If we do that with our hepatitis in Vavuniya finding, the **Use Drill-Down Query** selector in the Massive Screening Panel will be automatically populated with the query corresponding to our finding. Now, let us unselect the disease and loc_name dimensions from the massive screening list, and instead select age_grp and hit the **Run Screening** button again. Now, the algorithm will screen through all age groups of reported hepatitis cases from Vavuniya and sort them by the most statistically significant increase in the corresponding patient counts. Figure 12 shows the time series of patients over 45 years old, which happens to be the age group most significantly affected in this outbreak of viral hepatitis in Vavuniya region.

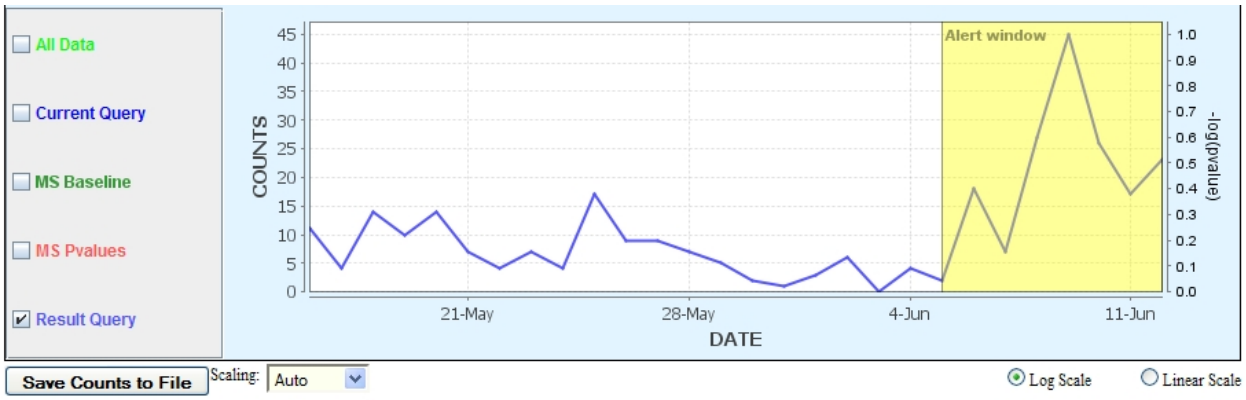

Figure 12: Drill down using another run of masive screening reveals that people older than 45 years seem to be the age group most spectacuarly affected by the June 2009 viral hepatitis outbreak in Vavuniya

3. Spatio-Temporal Analysis

The **Maps** component of TCWI can be used to detect and analyze spatio-temporal patterns in data. **Maps** panel can process data that contains a spatial attribute. In the discussed lk_flat_table data it is named “loc_name” and its values correspond to individual regions of Sri Lanka. In order to initialize the **Maps** panel, specify the spatial attribute name and click on **Load Map** button to view the map. Once the map is loaded, **Clear Map** resets the **Maps** panel.

3.1 Map Visualization

Figure 13 shows the geographic distribution of all disease cases over the complete period of time covered in the lk_flat_table data on the background of the Sri Lanka map. Figure 14 presents the corresponding temporal distribution of the visualized disease cases. Center of each circle corresponds to a Sri Lanka region, while its radius indicates the total number of patients for the date range shown in Figure 14. Try changing the date range using slider bars in time series window. Rendering of data on the map will be updated to reflect the volume of disease reports per region corresponding to the adjusted temporal scope. The legend in the bottom left part of the map display window presents the size to volume correspondence used in the map diagram.

3.2 Time Series Visualization

Temporal distribution of data is shown right under the map display. Its operation is analogous to the same panel under the Time Series tab. The green line shows the 7-day moving average. Animation is a new function which allows the users to watch spatial and temporal changes in the displayed data. Adjust values of **Scan window size** and **Slide window by** parameters (Figure 14) to adjust the resolution and speed of animation.

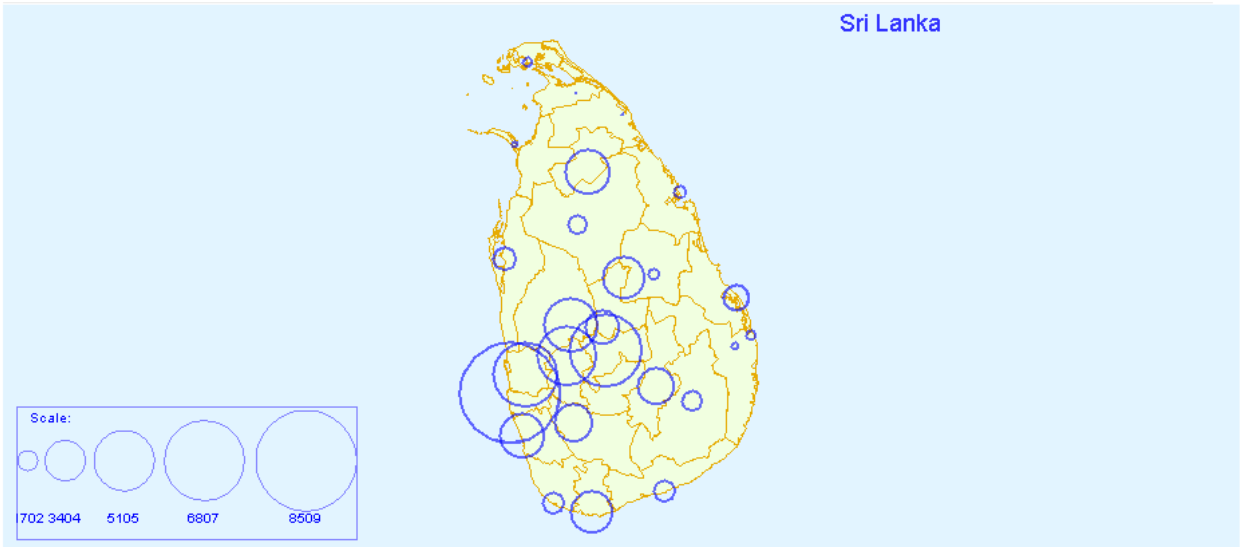

Figure 13: Map visualization.

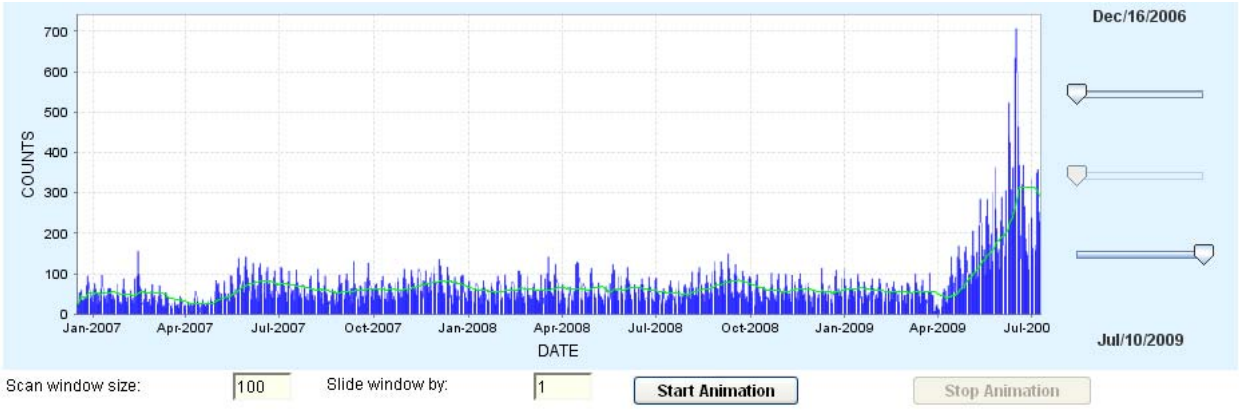

Figure 14: Time series view under the Map tab.

3.3 Attribute Selection Panel

The time series in Figure 14 shows the total daily counts of all disease cases. To display the time series corresponding to specific values of specific dimensions of data use the **Attribute Selection Panel**.

Functionality of this panel is generally similar to that of **Query Selection Panel** described before. Activating of the **Auto Filte**r modifies the dialog of the user-driven dimension-value selection such that only the actually present in data combinations of dimensions and their values are available for selection, given the already selected subsets. Selecting one of the dimensions and hitting the Submit/Reset button leads to splitting the currently visualized data into time series and circles on the map, separately colored for each of the individual values selected for this dimension.

Figure 15 shows how to display time series corresponding to *Dysentery* patients aging up to *5* years old recorded in regions of *Badulla* and *Batticaloa*. Note that after selecting the values, the user must press “Submit/Reset button” to update the time series and map visualizations.

Figure 15: Attribute Selection Panel under the Map tab.

3.4 Spatial Scan

TCWI implements a variant of the Multivariate Bayesian Spatial Scan algorithm for spatio-temporal analysis of public health data. This spatial scan method computes the overall probability of a disease outbreak anywhere in the scope of data selected by the user in the **Attribute Selection Panel** dialog, separately for each day in the scope. It is reported as the Spatial Scan Global score in the upper right corner of the map display window. The displayed value corresponds to the current last day of analysis. The users can inspect the specific values of the global score for other days by moving the end date slider bar located to the right of the time series window. The daily global scores are visualized with a red line in the upper part of the time series diagram.

The spatial scan algorithm also computes for each day, the probability of an outbreak occurring at each geographic location. The results are visualized on the map as circles filled with a color depending on the value of the respective probability estimate.

Figure 16 shows the dialog for setting the parameters controlling the spatial scan algorithm. **Temporal window size** acts as a smoothing parameter and it must be set to a value lower than 7. This value should reflect the expected disease outbreak duration. **Max group size** is the anticipated maximum number of regions affected by an outbreak. Based on initial experiments performed using *lk_flat_table* data, we recommend setting the **Temporal window size** to 7 and the **Max group size** to the value approximately equal 25% of the total number of distinct locations represented in the data.

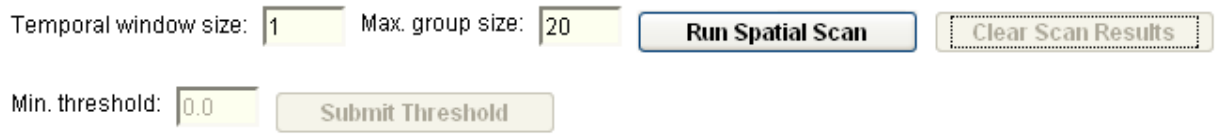

Figure 16: Selecting the Spatial Scan parameters

Figures 17 and 18 show results of running the spatial scan against counts of Leptospirosis cases using the end date of August 5, 2008. The global score (0.9612) in the top right of Figure 17 indicates that the chance of a Leptospirosis outbreak anywhere in the country on that day exceeds 96%. The shaded circles centered at each region (heat map legend is placed at the top left side of Figure 17) depict spatial distribution of outbreak probability. The blue circles still indicate the total number of Leptospirosis cases in each region. In Figure 18, the red time series plot depicts the temporal distribution of past global. Once the review of results of the spatial scan run is complete, the users can **Clear Scan Results** using the button shown in Figure 16.

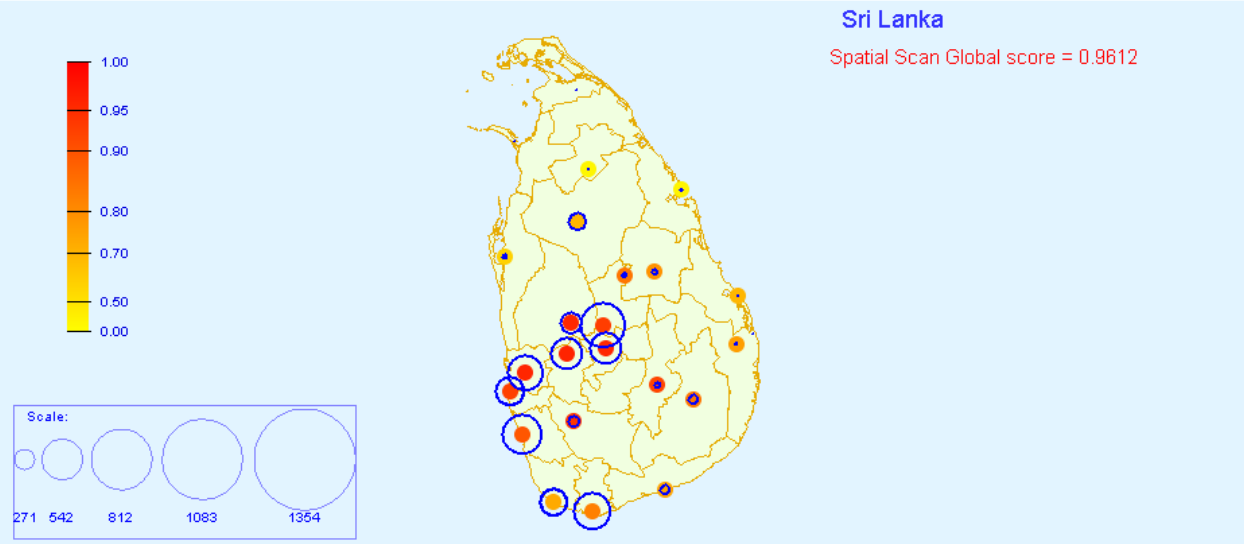

Figure 17: Detecting an outbreak of Leptospirosis with Spatial Scan (map view).

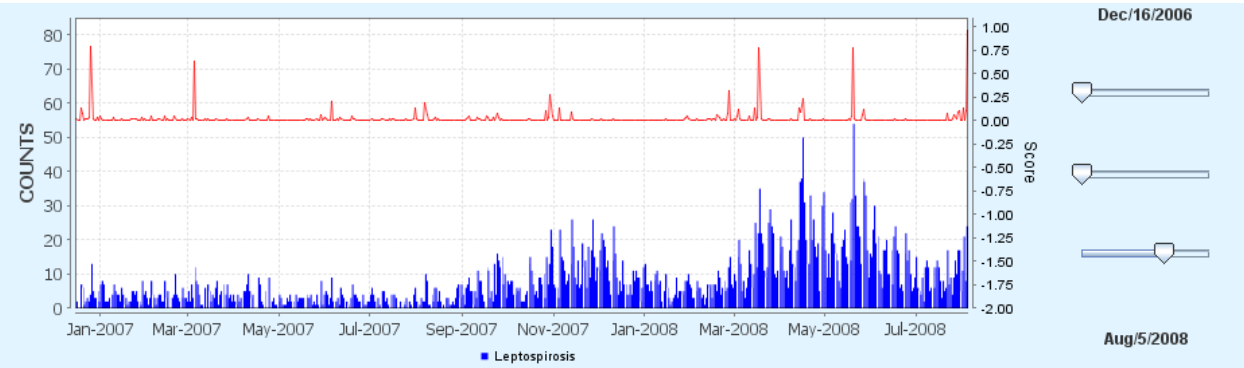

Figure 18: Detecting an outbreak of Leptospirosis with Spatial Scan (time series view).

4. Summarization of Data with Pivot Tables

Pivot Tables tab offers an interactive data summarization capability. TCWI provides an efficient algorithm for aggregating multidimensional data of counts of events (such as the numbers of reported disease cases) into a two-dimensional matrix view. Figure 19 depicts an example pivot table computed for the lk_flat_table data using two dimensions: *disease* and *age_grp*. Each cell in the table shows the total number of recorded cases of specific disease (row) among patients in the specific age group (column). The table also shows the marginal sums (by row, by column, and the total number of cases).

Start Over >> disease -- age_grp

Attributes: gender loc_name sign symptom

Rows: disease

Columns: age_grp

	0-1	1-5	15-19	20-24	25-29	30-34	35-39	40-45	6-14	Above_45	Total
Dengue_fever	1260	1342	1310	3937	2609	2648	2591	2664	1366	6693	26420
Dysentery	665	713	715	2146	1427	1376	1434	1395	729	3623	14223
Encephalitis	25	23	23	72	49	61	47	62	29	132	523
Enteric_fever	167	196	166	495	350	322	350	355	157	852	3410
Food_poisoning	175	138	137	435	323	343	298	356	174	765	3144
Human_rabies	3	1	3	14	11	3	9	6	1	17	68
Leptospirosis	427	523	475	1384	1025	973	965	984	501	2420	9677
Typhus_fever	131	96	128	359	249	268	241	234	125	603	2434
Viral_Hepatitis	460	459	453	1377	887	919	960	901	501	2290	9207
Total	3313	3491	3410	10219	6930	6913	6895	6957	3583	17395	69106

Figure 19: An example 2-way Pivot Table

Construction and editing of pivot tables can be done by dragging and dropping icons with the individual dimension names between the “Attributes”, “Rows” and “Columns” lists. The dimensions in the “Rows” and “Columns” lists are used to create rows and columns of the table, respectively, while those which remain in the “Attributes” list are ignored. Multiple attributes can be put in “Rows” and “Columns” to create nested tables. To move from table shown in Figure 19 to the one shown in Figure 20, just drag the icon denoting “gender” to the “Row” list. Now, the rows are split by disease and gender, and each cell of the table contains counts of either male or female patients diagnosed with a specific disease, and belonging to a specific age group.

The users can always clean the current table and start over by clicking the **Start over** link (Figure 19).

Left clicking on any data cell of the table will produce a pop-up window with a time series plot of the data represented by that cell. The temporal range of data can be adjusted by changing the

dates at the bottom of the page. Figure 21 shows the time series for male patients aged 30-34 years old having been diagnosed with typhus fever.

Attributes: loc_name | sign | symptom

Rows: disease | gender

Columns: age_grp

		0-1	1-5	15-19	20-24	25-29	30-34	35-39	40-45	6-14	Above_45	Total
Dengue_fever	Female	633	646	693	1961	1327	1322	1300	1359	686	3397	13324
	Male	627	696	617	1976	1282	1326	1291	1305	680	3296	13096
Dysentery	Female	333	365	371	1074	683	647	755	729	346	1776	7079
	Male	332	348	344	1072	744	729	679	666	383	1847	7144
Encephalitis	Female	13	13	12	32	34	29	28	35	19	55	270
	Male	12	10	11	40	15	32	19	27	10	77	253
Enteric_fever	Female	88	91	91	224	172	154	164	163	87	382	1616
	Male	79	105	75	271	178	168	186	192	70	470	1794
Food_poisoning	Female	82	71	64	210	158	166	161	179	98	392	1581
	Male	93	67	73	225	165	177	137	177	76	373	1563
Human_rabies	Female	1	1	3	6	4	2	4	3	1	8	33
	Male	2	0	0	8	7	1	5	3	0	9	35
Leptospirosis	Female	207	273	243	692	491	466	506	472	233	1203	4786
	Male	220	250	232	692	534	507	459	512	268	1217	4891
Typhus_fever	Female	70	54	67	158	120	129	121	111	67	305	1202
	Male	61	42	61	201	129	139	120	123	58	298	1232
Viral_Hepatitis	Female	217	227	211	667	442	453	448	456	243	1145	4509
	Male	243	232	242	710	445	466	512	445	258	1145	4698
Total		3313	3491	3410	10219	6930	6913	6895	6957	3583	17395	69106

Figure 20: Example 3-way pivot table.

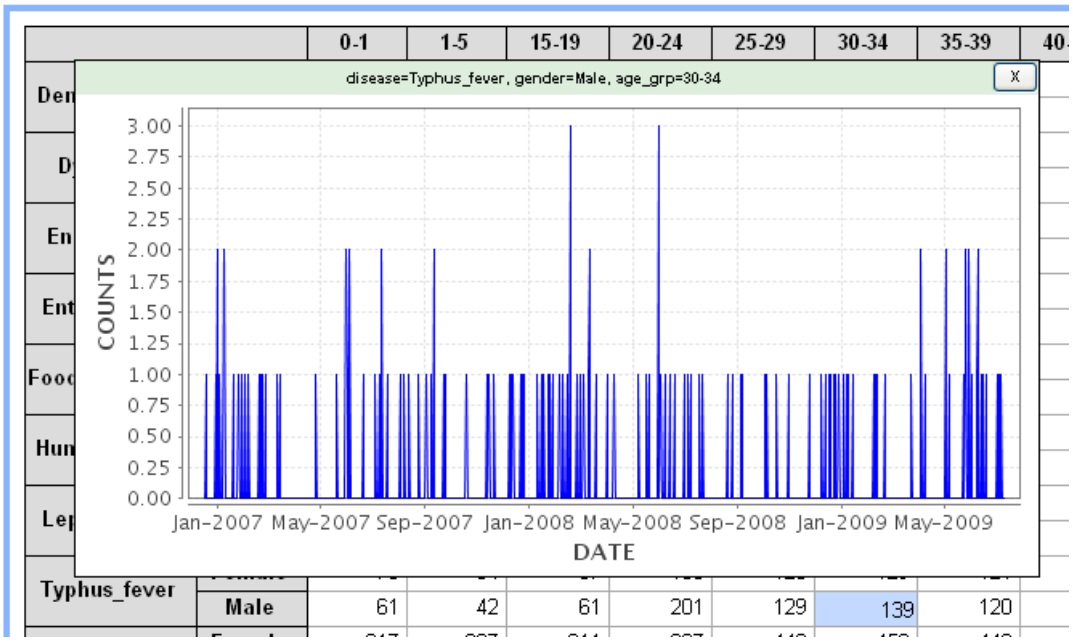

Figure 21: Example Time Series view under Pivot Table tab.

Left-clicking on any row or column name of a pivot table will produce a pie chart visualizing the distribution of data split according to the values of the counterpart row or column dimensions. Figure 22 shows the distribution of age groups for female patients diagnosed with dengue fever.

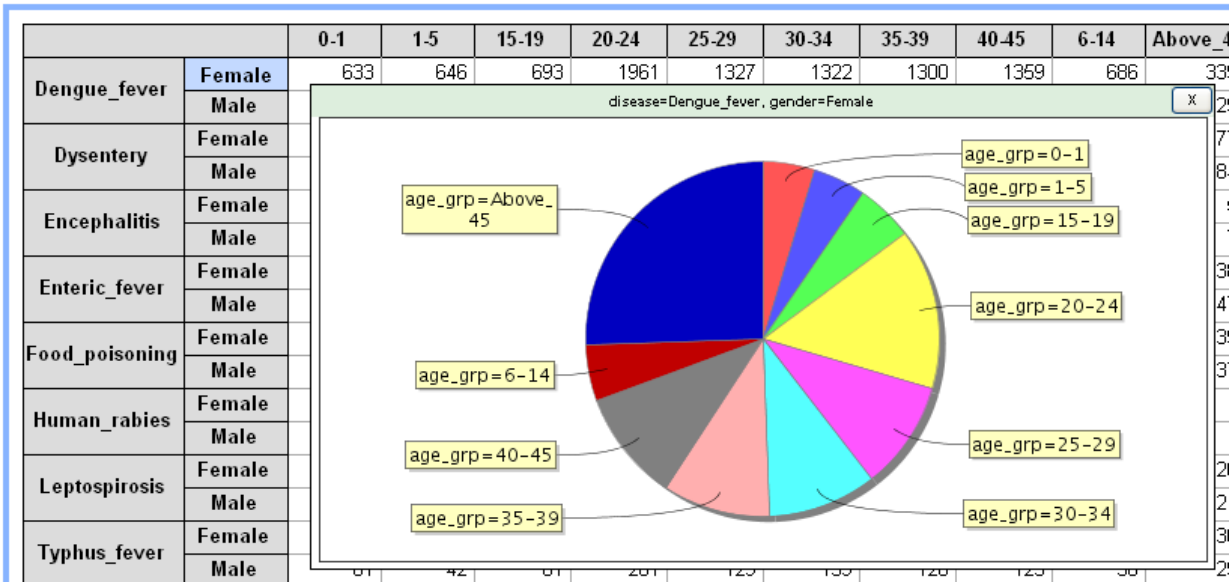

Figure 22: Example pie chart under the Pivot Table tab.

Each cell in a pivot table reflects the aggregate counts of disease cases computed over a specific period of time. It is shown and it can be edited at the bottom of the page. If either or both of the dates are changed, TCWI will quickly re-compute counts for all cells in the current pivot table.

5. Future Work

We have mentioned above a few minor extensions of the capabilities of TCWI scheduled for inclusion in the next release of the software (expected in mid-December 2009). The next release is also intended to include the following major extensions:

- Ability to browse the original transactional data. We are working on an interface that will allow users to see the individual original records of disease cases loaded to TCWI, besides the aggregated counts shown currently in the form of time series. This will support detailed drill downs and other basic database operations not easily accessible through TCWI.

- Background screening and alerting capability. Currently, TCWI offers a variety of analytic functions available for interactive use. The background screening and alerting will enable setting up scripts for user-determined analyses to be automatically executed at user-defined periods of time. It is of practical importance to execute screening for emerging patterns at regular intervals in order to maintain situational awareness among stakeholders of RTBP, even if the TCWI operators are not always present at their consoles to invoke the analyses manually. The results of scheduled analyses will be stored for inspection by the users, and the automatically generated alerts will be made available for distribution to the designated recipients.