WHO Outbreak Toolkit Virtual Assistant

Stéphane Ghozzi
WHO/WHE/HIM/MDC and RKI/Signale
ghozzis@rki.de

31 October 2019

Contents

1 Introduction

This report describes the results of a consultancy in the MDC group of WHO, 50% part time from 1st May until 31st October 2019. Anne Perrocheau was the supervisor, with Lucas Deroo and Karl Schenkel contributing as well. As part of the Outbreak Toolkit project, it aimed to develop a prototypical tool to help the investigation of outbreaks of unknown origins.

The overall goal was to devise an automated approach to support the formulation of hypotheses and dynamically adapt investigation questionnaires according to evolving situation and information. A concept was proposed with three modules (see Fig. 1 for an illustration):

- Module **A** allows the entry of case data according to a given questionnaire;
- Module **B** analyses the case data and, by comparison with knowledge on diseases of interest ("references"), computes scores and produces visualisations;

- Module **C** adapts the questionnaire in function of the output of the analysis, in particular adding questions specific to the most probable diseases and remove questions specific to least probable diseases.

Since Module A is already tackled by existing tools such as Go.Data, and during the start of the consultancy no disease-specific questionnaires had been developed, it was decided to focus on Module B. The WHO Outbreak Toolkit Virtual Assistant is a prototype of this module.

For lack of a data set of confirmed cases, the **references** were defined as typical expectations for any disease of interest. They represent the expert knowledge that will help evaluate the situation.

2 Workflow and structure of the project

The steps of the project are:

1. import and process the data dictionary corresponding to the T1 questionnaire
2. generate references for fake diseases to check that everything works as it should
3. import real references as filled by experts
4. generate, from any set of references, case data (line lists) corresponding to different types of events
5. compute scores that measure the similarity of any one case to any one disease reference
6. visualise the proximity of cases and references

They are implemented in the programming language R and illustrated in the file "vignette.html". Moreover, a web application (in the framework Shiny, based on R) allows users to upload a data dictionary, references and case data, and visualise the results.

The data dictionary as it is used, as well as the references and case data, should be stored in a format easily readable and editable by non-informaticians. For this reason they are exported to simple tables in the CSV format. During the analyses in R, they are used mostly as matrices of lists, sometimes as data frames.

3 Processing the data dictionary

We need to know the list of variables in the T1 questionnaire, their types (e.g. date), the values they can take, and whether multiple answers to any given question are allowed. We use the version of the data dictionary as of 6 June 2019, specifically the file "dictionnary complet 04062019.xlsx".

A certain amount of ad-hoc processing is needed to bring it to a useable form: keep only relevant variables, remove duplicates, clean formats. (The pre-processing steps have been described in an internal document.)

Note that the following variables might need to be **converted** from a line list following the original data dictionary: all categorical data (category 0), "sign_bleed_site" (numeric categories), all the dates (format), "sign_temp" (degrees Celsius).

For the following variables, permissible values are added to the dictionary (based on own research or Anne's or Karl's suggestions):

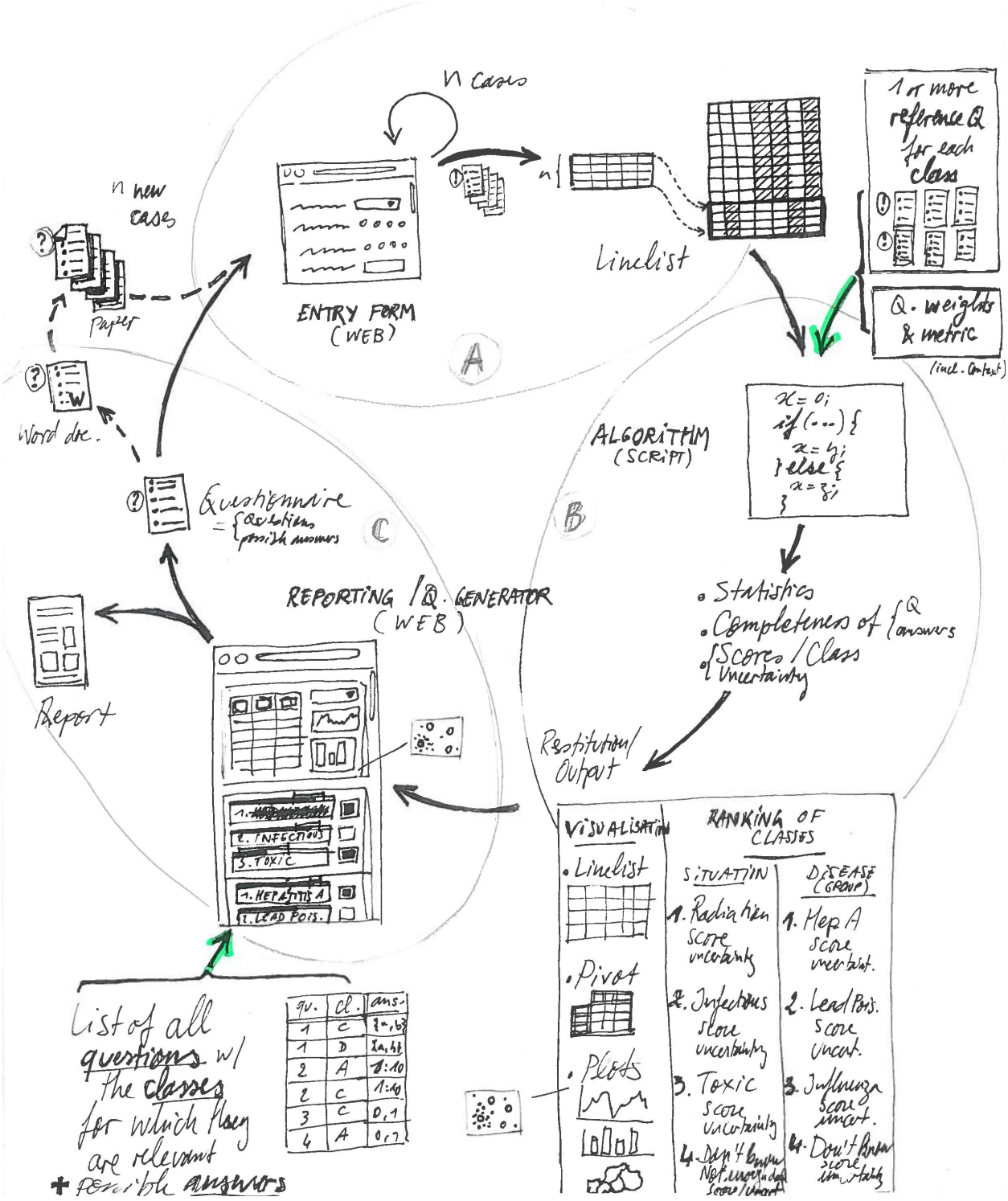

Figure 1: Concept of an automated adaptative tool for the supporting investigations of outbreaks of unknown origin. See the main text for details.

- gps_lat: -180, 180
- gps_long: -180, 180
- alt_sgpt___u_l_: 1, 2000
- ast_sgo___u_l_: 1, 2000
- lactate___mmol_l_: 0, 3
- haemoglobin____g_l_: 60, 180 (warning: haemoglobin often given in g/dl, i.e. the number is 10 times lower than in g/l)
- creatinine___umol_l_: 50, 300
- total_bilirubin___umol_l_: 0, 99
- potassium___mmol_l_: 0, 10
- wbc_count___x109_l_: 4, 20
- urea___mmol_l_: 0, 3
- platelets___x109_l_: 20, 500
- creatinine_kinase___u_l_: 10, 500
- sign_temp: 24, 44
- sign_hr: 40, 160
- sign_syst: 5, 25
- sign_dias: 2, 13
- sign_resprate: 8, 30
- sign_sat: 65, 100
- cond_malnut: 80, 500
- dates: NA

All variables of unknown permissible values are discarded, which includes all variables of *type* "string" (which should all be free text as categorical answers should be of type "int_categorical"). We also discarded the variables of unknown *type*.

We can discard the following variables that are linked to the interviewees and will not help identify the disease: "interview_date" and "telephone_number". We also discard "age_month", which is redundant with "age_year".

Further, two variables might indicate unequivocally the disease or absence thereof: "test1_result" and "classification". However at the moment in the data dictionary they do not indicate the disease, so that we cannot use them and they are discarded as well.

It remains 117 variables out of 235.

Lastly, we keep only the properties *name*, *type*, *values* and *multiplicity*, see Fig. 2.

4 Disease references

Disease references are represented in a very coarse, but hopefully still useful form: for each variable of the data dictionary, three properties are defined:

- a **weight**, a number between 0 and 1, measures the importance of the variable relative to the others in diagnostising the disease;
- **observations** are a range or a set of values within which one expects the the typical responses to the questionnaire in case the outbreak originates from the disease of interest;

name	label	type	format	values	multiplicity
report_date	Date case was reported (DD.	date	YYYY-MM-DD	NA	0
notification_facility_type	Case reported by:	int_categorical	NA	0=unknown;1=mobileteam;2=(	0
birth_date	Date of birth (DD/MM/YYYY):	date	YYYY-MM-DD	NA	0
age_year	Age at onset: years	int_ordinal	NA	0, 99	0
sex	Male Female	int_categorical	NA	1=male;2=female	0
occupation_hcw	Works in health facility:	int_categorical	NA	0=unknown;1=yes;2=no	0
gps_lat	GPS latitude:	float	point_decimal_no_grouping	-180, 180	0
gps_long	GPS longitude:	float	point_decimal_no_grouping	-180, 180	0
mass_gathering_yn	Did you participate in mass ga	int_categorical	NA	0=unknown;1=yes;2=no	0
mass_gathering_date	Date (DD/MM/YYYY):	date	YYYY-MM-DD	NA	1
no_hhd	How many household member	int_ordinal	NA	0, 100	0
sick_hhd_yn	Do you have any household m	int_categorical	NA	0=unknown;1=yes;2=no	0
sick_hhd_onset	date of onset (DD/MM/YYYY)	date	YYYY-MM-DD	NA	1
sick_comm_yn	Are you aware of people outsi	int_categorical	NA	0=unknown;1=yes;2=no	0
sick_comm_relat	Relationship	int_categorical	NA	1=friend;2=workmate;3=relativ	1
sick_comm_onset	date of onset (DD/MM/YYYY)	date	YYYY-MM-DD	NA	0
contact_sym_yn	Did you had any contacts wi	int_categorical	NA	0=unknown;1=yes;2=no	0
contact_sym_date	Date of last contact (DD/MM/Y	date	YYYY-MM-DD	NA	0
travel_hist_yn	Did you travel, outside your re	int_categorical	NA	0=unknown;1=yes;2=no	0
travel_hist_date	Traveling Date (DD/MM/YYYY	date	YYYY-MM-DD	NA	1

Figure 2: First rows of the processed data dictionary.

- the **extension** is the size of a confidence interval (by default the 50% confidence interval) for the variable in case of an outbreak of the disease of interest.

Any of them can take missing values, although the more filled, the better the analysis will work. Dates don't have *observations* and categories don't have *extensions*.

The *extension* is optional and is not used as such in the analysis. (The idea was to consider the statistics of whole event, not just case-based, but that was not implemented.)

The first rows of a fake reference are shown in Fig. 3.

4.1 Generating fake references

To test the analysis, references for fake diseases are randomly generated: They typically are quite distinct from one another and have all properties for all variable filled. In that sense, they show a best case scenario, and help gauge the most that can be expected of the approach. For each variable:

1. the **weight** of either 0, 0.5 or 1 is randomly assigned: 0 for 1/4th of the variables (rounded), 0.5 for half of the variables (rounded), 1 for the rest (about 1/4th of the variables);
2. the **values**
   - for "int_categorical" are drawn randomly a random number of times (between 1 and n-1 times), with:
     - the value "0" (corresponding to "unknown"), having been removed;
     - n the number of permissible values;
   - for "int_ordinal" are the sequence of integers between two numbers picked at random among the permissible values;

name	weight	observations	extension
report_date	0	NA	252
notification_facility_type	0.5	1, 4	NA
birth_date	0	NA	9
age_year	1	49, 97	45
sex	0.5	1	NA
occupation_hcw	1	1	NA
gps_lat	1	2.78, 165.12	69.16
gps_long	1	91.61, 136.61	35.26
mass_gathering_yn	1	2	NA
mass_gathering_date	0	NA	344
no_hhd	0	32, 88	26

Figure 3: First rows of a fake (randomly generated) reference.

- and "float" are two different numbers (the interval) drawn at random within the permissible values;
- for "date" are NA (no value);

3. the **extension**
   - for "int_ordinal" is drawn at random between 1 the size of the range of permissible values;
   - for "float" is drawn at random between (by default) 10% of size of the range and the size of the range;
   - for "date" is drawn at random between 1 and (by default) 400 days;
   - for "int_categorical" is NA.

For reproducibility, we set the random seeds while generating references.

The generated references can be saved in a CSV file, e.g. for later use in the web application.

4.2 Defining real references

More interesting are references for real diseases. Anne and Karl have filled five, all potentially producing jaundice syndromes, for:

- Hepatitis A
- Hepatitis E
- Leptospirosis
- Mushroom infection
- Yellow fever

These references have many properties missing, which results in a lower discriminatory power than the fake references. Moreover some are quite close to one another, especially Hepatitis A and

Hepatitis E.

The template "reference_template.csv" for filling references is provided (generated from the data dictionary). In filling a reference, any cell can be left out (leaving the value NA). Categories are given as numbers. Ranges and sets are numbers separated by a comma.

4.3 Importing references

References can be imported in the analysis from the CVS files. Then each reference as for name the file name with underscores replaced by a blank and the extension removed, e.g. the file "Hepatitis_A.csv" will produce the reference "Hepatitis A".

5 Generating case data (line lists)

The last input needed are case data. Since none were available in the format of the data dictionary, and to test that the analysis performs as expected as well as gauge its potential usefulness, case data were randomly generated.

For each of the two sets of references (fake and real), five scenarios were considered:

1. cases are random, i.e. their values are drawn uniformly at random from the allowed values as defined in the data dictionary
2. all cases are drawn exactly from one reference
3. half of the cases are drawn from one reference, half from another one
4. all cases are drawn exactly from one reference, but noise is added (see below for the definition of noise)
5. half of the cases are drawn from one reference, half from another one, with noise added to all cases

For dates, the minimum and maximum dates have to be given as parameters.

Noise indicates that each case had for each variable a (by default) 50% chance not to be drawn from a reference, but from the data dictionary, i.e. to be completely random.

We don't generate missing observations.

We don't take into account possible correlations between variables, e.g. between "mass_gathering_yn" and "mass_gathering_date" or between "sick_hhd_yn" and "sick_hhd_onset", so that here we can have contradictory observations.

The **noise** can have a value between 0 and 1. For each case, for each variable: noise = 0: the observations are drawn from the reference, with all "date" within the extension (this is automatically the case for "int_ordinal" and "float" and doesn't to "int_categorical"); noise = 1: the observations are drawn from the data dictionary (and extension is ignored). For intermediate noise values, we draw each answer from either the reference (probability 1-noise) or from the data dictionary (probability noise). For the first line list, we use noise = 1 and no reference list, for the line lists 4 and 5 we use a (by dafault) 50% noise.

When **multiple** answers are possible, we set the maximum number of answers to (by default) 3 or, for categories, the minimum of 3 and the number of possible answers. The number of answers is

case_id	report_date	notification_f	birth_date	age_year	sex	occupation_h	gps_lat	gps_long	mass_gatheri	mass_gatheri
danger_fever_event_1	2015-02-05	4	2018-08-20	54	2	1	-10.561609198	-118.40378069	2	2017-09-13
danger_fever_event_2	2019-09-01	4	2017-12-21	47	2	1	-16.694285074	-62.322222565	2	2015-08-08
danger_fever_event_3	2019-04-22	1	2015-02-11	44	2	1	6.1287991879	-102.70033583	2	2015-09-04, 20
danger_fever_event_4	2019-10-18	1	2015-06-10	44	2	1	1.6473386705	-100.35858407	2	2016-05-15, 20
danger_fever_event_5	2018-07-12	4	2017-12-24	43	2	1	-18.524208949	-40.380729289	2	2015-01-04, 20
danger_fever_event_6	2018-12-01	4	2018-10-24	58	2	1	-6.7130254994	-127.62705966	2	2017-01-02, 20
danger_fever_event_7	2017-05-14	1	2016-05-25	64	2	1	32.523126923	-140.13485732	2	2015-05-30, 20
danger_fever_event_8	2016-01-02	1	2017-05-17	44	2	1	-21.088938099	-19.647448508	2	2018-02-26
danger_fever_event_9	2018-01-03	1	2018-04-11	44	2	1	-7.3504254324	-112.05122689	2	2018-03-15, 20
danger_fever_event_10	2019-09-15	3	2015-10-15	59	2	1	-20.667682803	48.591987626	2	2015-10-22, 20
danger_fever_event_11	2015-08-23	4	2017-07-22	55	2	1	-15.869003509	-137.3709156	2	2016-03-16, 20
danger_fever_event_12	2015-05-25	3	2017-07-31	44	2	1	5.0164222981	-106.83533791	2	2015-07-01

Figure 4: First rows and columns of a randomly generated line list

drawn uniformly at random between 1 and the maximum. All multiple observations for a given case and a given variable are drawn either from a reference or from the data dictionary.

To take **extension** into account, we proceed as follows:

1. pick a lower bound among the permissible values lower than the maximum minus the extension
2. set the upper bound at lower bound plus extension resp. the highest observations lower than the lower bound plus extension for "float" resp. "int_ordinal" or "date"
3. identify the set of permissible values within the bounds
4. for each case, for each of the multiple values: draw an observation either within resp. outside the bounds with probability the size of the confidence interval resp. 1 minus the size, relative to the size of the range of allowed values. If the variable is "int_ordinal" or "date", for that case, remove the value drawn from the set; stop when the attempt to draw an observation fails.

Hopefully this leads to observations distributed as desired within the extension. However that means that the actual multiplicity might be smaller than one determined above for "int_ordinal" or "date".

Fig. 4 shows the first rows and columns of such a line list.

Same as for the references, random seeds are set and line lists can be exported to and imported from CSV files.

6 Scores

For each reference and for each case, scores are computed that measure the similarity between the two. Specifically:

For a given reference r, a given case c and a given variable v with a reference value (i.e. is not "date"), we define the agreement between reference and case S^1_{rcv} to be 1 if at least one answer of the case is in the set or the range of the reference, 0 if all are outside of it and 0.5 if there are no observations or the observation is "unknown". From there we can define a score S^1_{rc} for a case to correspond to the reference by using the weights w_{rv}: $S^1_{rc} = \sum_v w_{rv} S^1_{rcv} / \sum_v w_{rv}$.

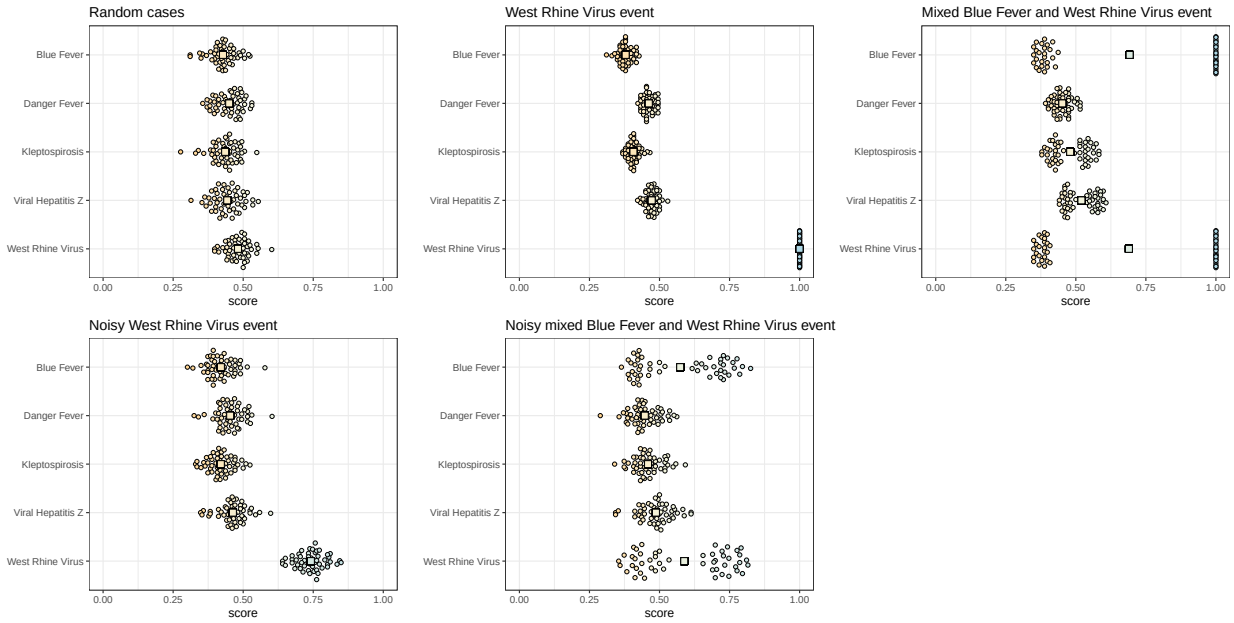

Figure 5: Score distributions for different events for the five fake references. Each dot represents one case, the squares show the average scores.

That way, cases that corresponds exactly to one reference have a score of 1 for that reference. Random cases are expected to have scores around 0.5.

Scores for the series of scenarios described above are shown in Fig. 5 for fake references and Fig. 6 for real references.

The distribution of scores can help identify the events. However, the approach reaches its limit with real references and a noisy mixed event, where one cannot distinguish a clear pattern.

7 2 dimensional visualisation

One can try to directly visualise the relative proximity of cases as well as between cases and references. However, cases are 117 dimensional data (one dimension for each variable). Thus we need to reduce their dimensionality before hoping to recognise patterns. For that we use the t-SNE algorithm.

References are represented by (be default) 10 cases drawn exactly from the corresponding reference.

7.1 Distance based

We define a distance between cases based on all 117 variables and apply the t-SNE algorithm to the resulting distance matrix. The distance measure has to be the same for all cases and references, meaning we cannot consider different weights for different references. Instead, to still take into account that certain variables might have more importance overall than others, we consider weights averaged over all references: $w_v = \langle w_{rv} \rangle_r$ for each variable v.

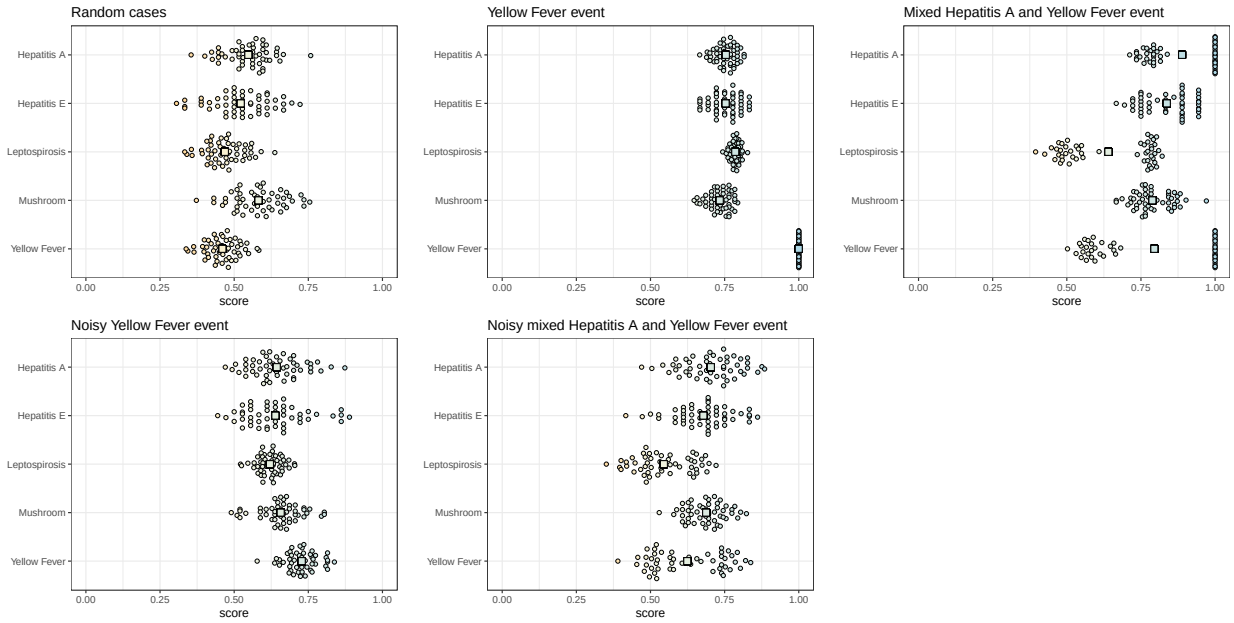

Figure 6: Score distributions for different events for the five real references. Each dot represents one case, the squares show the average scores.

The variable-wise distance is defined differently for quantitative and categorical variables. Important is that each variable set has the same size, which we set to 1: two cases have, for any variable, a distance between 0 and 1. For quantitative variables we take the minimum difference between any pair of (multiple) observations, normalized to a maximum of 1; for categorical variables, the distance is 0 if both cases share any observation, it is 1 else; if any case has a missing observation, the distance is 0.5.

With d_{ij} the distance between case i and case j, and w_v the weight for variable v averaged over all references:

$$d_{ij} = \sqrt{\sum_v w_v \, d_{ijv}^2 / \sum_v w_v}$$

with

$d_{ijv} = \min_{m,n} |v_{im} - v_{jn}| / \max_{i,j}(\min_{m,n} |v_{im} - v_{jn}|)$ if variable is *int_ordinal*, *float* or *date*
$d_{ijv} = 1 - \delta(|\{v_{im}\}_m \cap \{v_{jn}\}_n|)$ if variable is *int_categorical*
$d_{ijv} = 0.5$ if any answer is missing

where v_{im} is the m-th observation of variable v for case i, and $\delta(x) = 0$ if $x = 0$, 1 else.

Smallest distance is 0 and largest distance is 1.

Note that though the extension is not taken into account in computing distances, the space covered by reference cases is indicative of it.

The computation of distance is relatively expensive, with about 5 minutes needed for 10 events of 50 cases and 10 reference cases each.

Results for the series of scenarios described above are shown in Fig. 7 for fake references and Fig. 8 for real references.

Figure 7: 2 dimensional visualisations of different events for the five fake references. Each grey dot represents one case, the colored dots represent the reference cases.

Figure 8: 2 dimensional visualisations of different events for the five real references. Each grey dot represents one case, the colored dots represent the reference cases.

Figure 9: 2 dimensional visualisations of different events for the five fake references. Each grey dot represents one case, the coloured dots represent the reference cases.

Even in the worst-case scenario (noisy mixed event with real references), clear patterns can be recognised.

7.2 Score based

Rather than computing distances, we can take a score representation: each case gets coordinates its scores for each of the 5 references in a set. This is much faster than the distance-based approach, but doesn't lead to good results as soon as noise is added, and thus is not a helpful approach, at least note with the scores used here. Results are shown in Fig. 9 and 10.

8 Web application

A Shiny-based web application allows one to interactively see how the analysis works by manually uploading the three inputs need (a data dictionary, a set of references, a line list). On top of two graphs similar to the ones shown above, it includes a time series (the weekly case count), a map, a line list with scores, and a pivot table. A screenshot of the application is shown in Fig. 11.

All graphics are interactive: One can hover to see the case id, zoom in and out, sort, filter, etc.

It coded in the file "app.R" and can be run on any computer with R and the necessary libraries installed.

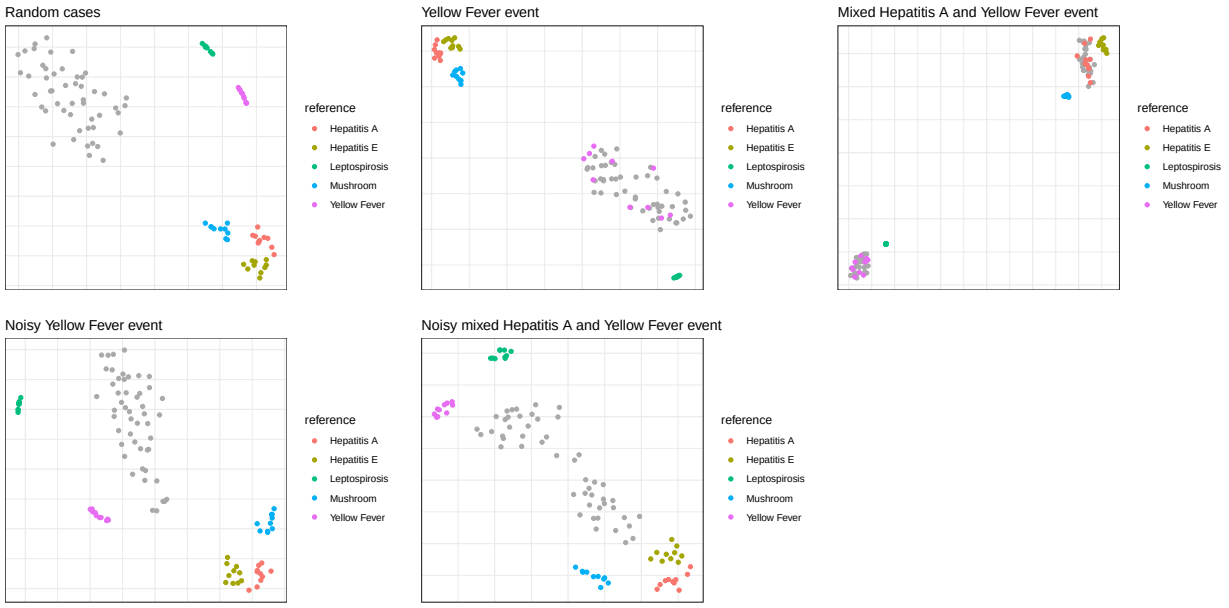

Figure 10: 2 dimensional visualisations of different events for the five real references. Each grey dot represents one case, the coloured dots represent the reference cases.

9 Conclusion and outlook

Already the simple approaches described here are promising and seem able to deliver useful insights.

While the 2d visualisations discriminate better, the scores attribute single numbers to each case and their values for one reference don't depend on the other reference being considered at the same time.

The usefulness for potential users can be probed by letting them play with the web application: Are the *functionalities* interesting? Can they picture themselves using such a tool? In which situations?

The next step should be to test the approach with *real case data*. Also interesting would be more complete references, as well as a more diverse set of real references.

To go farther, one could first *improve and evaluate the approach*:

- reducing the computation time of distances;
- distinguishing between suspected, confirmed and excluded cases;
- applying machine learning (classification) on with real, confirmed cases to quantify the patterns one might observe; and, to some extent, complement or replace the references.

Moreover, one could envision including the analysis in an existing such as Go.Data, so that real case data would automatically be available, in the right format.

Lastly, ideas for future *collaborations* on the WHO Outbreak Toolkit Virtual Assistant include:

- making an R package and sharing it with community, e.g. the R epidemic consortium (RECON);
- working with a Cambridge University working group on a similar topic;
- working with a WHO Medical expert group, medical experts at the Antwerp University, the READY project (USAIDS), the RKI.

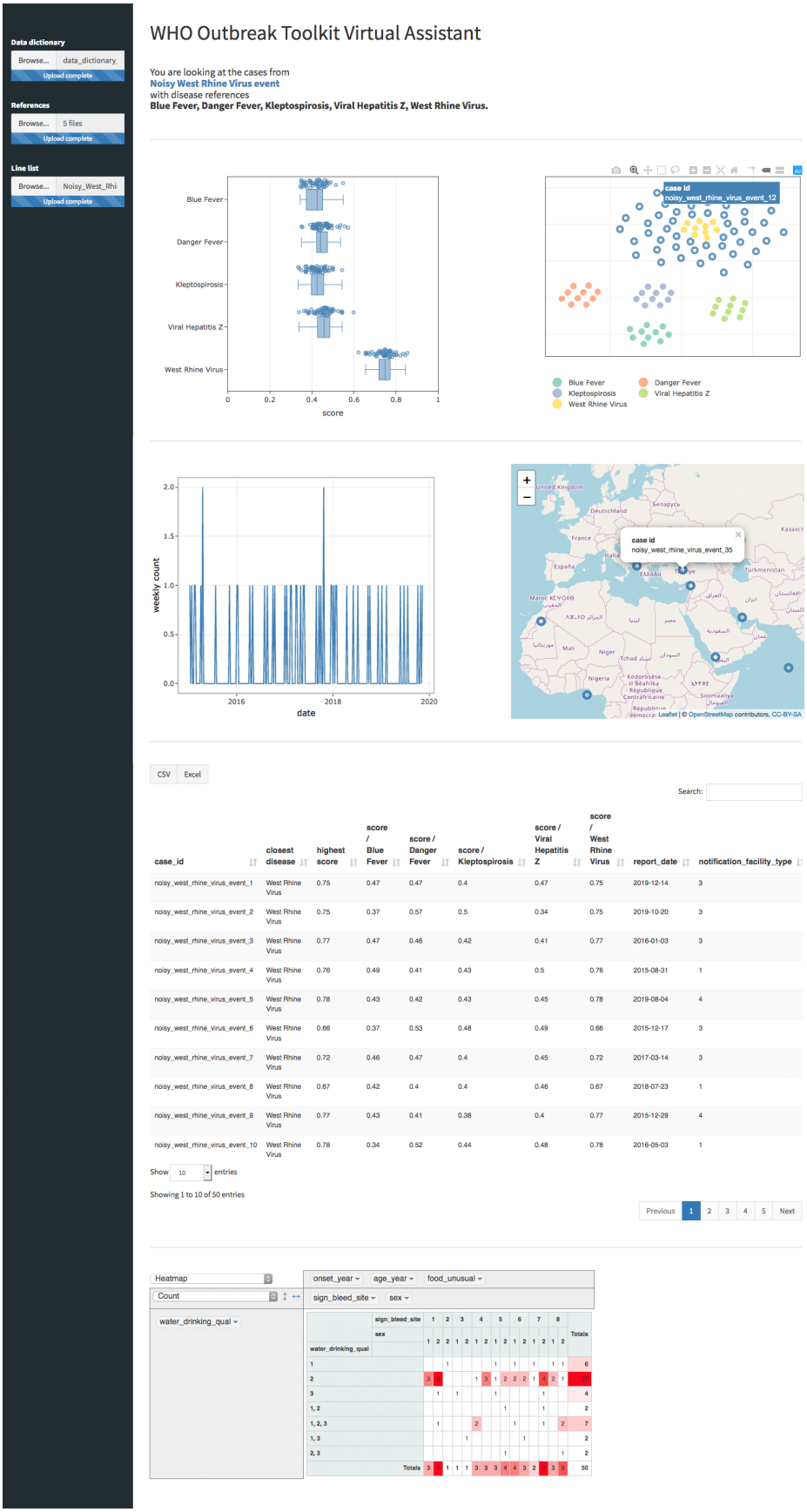

Figure 11: Screen shot of the web application.