

User manual

GMOseek software

15.5.2010

Introduction

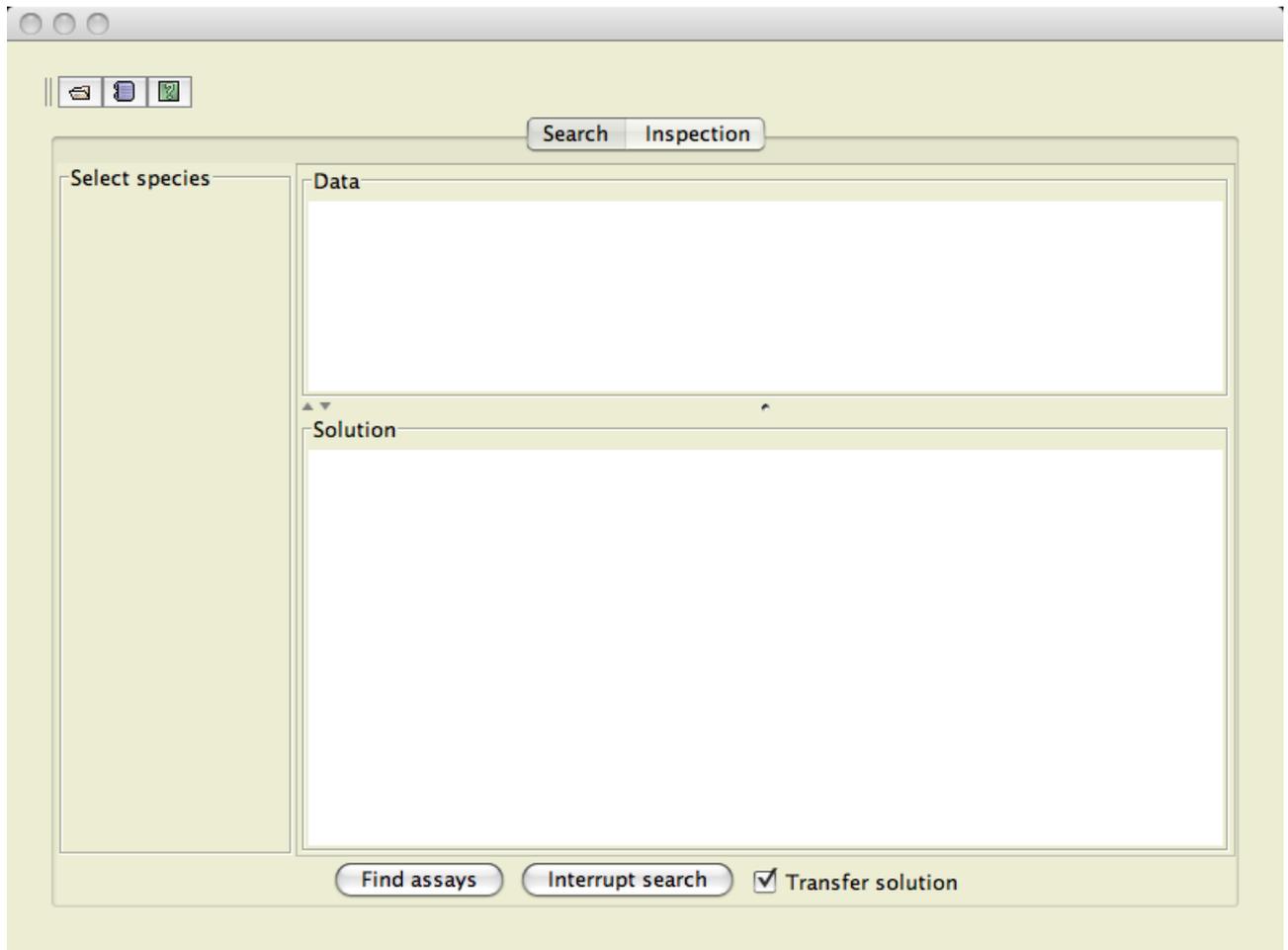
GMOseek software was developed within the GMOseek project. The main goal of the software is to provide optimal or near-optimal solutions of the two phase GMO testing process and to support the inspection of testing results.

GMO testing is performed in order to confirm the absence or to identify the presence of individual GMOs in food, feed and seed samples. The goal of laboratories is to perform testing as conveniently as possible while guaranteeing the accuracy of results. Different testing strategies can be used. The GMOseek software supports decision making in sample-testing to help choose cost-effective GMO testing strategies that are tuned to a sample at hand. The system also helps the interpretation of wet-laboratory results. The system can adapt to new situations on the market by changing the input data matrix, which includes the data about GMOs, methods (assays) for detecting GMOs and probabilities of GMO presence.

Software package

The software is packed and deployed as one file which is named *GMOseek.jar*. It can be placed at any location suitable for a user. The program can be executed by a double click on the file from a file manager or by `java -jar GMOseek.jar` from the command line. The software runs on any operating system which has java 1.5 or later installed.

The window below appears after execution is started.



Management of definitions

The toolbar consists of three icons representing commands *Open definition*, *Recent definitions*, and *Help*.



The command *Open definition* enables browsing a file system and selecting a file which contains a definition. A browsing filter is set to a directory of the last opened file and **.tab* as selector for *GMO table* file names. The filter can be changed to *All files*.

The command *Recent definitions* opens a list of recently opened definitions and one of them can be picked. This feature provides a quick switching among definitions.

The command *Help* opens a window containing this manual.

The format of a definition (GMO table) is the same as defined in the project GMOTrack. Both use a data matrix as a knowledge base to perform optimization and decision support. The main components of the data matrix are data about GMOs and the screening methods that detect them. This information is complemented with the information on GMO plant species and the probability of appearance of a GMO in a sample.

The input data matrix format is described below. The format needs to be followed strictly. A screenshot of the properly prepared data matrix is presented in the table below.

Table : Example of properly formatted input data

GMOname	species	probability	P-35S	P-TA29	P-nos	CP4-epsps	bar	Barstar	T-nos	T-35S	P-35S::bar
d	d	c	d	d	d	d	d	d	d	d	d
m	m	m									
RRS	soybean	0.54	1	0	0	1	0	0	1	0	0
Bt176	maize	0.11	1	0	0	0	1	0	0	1	1
MS1	oilseed rape	0.05	0	1	1	0	1	0	1	0	1
RF1	oilseed rape	0.01	0	1	1	0	1	1	1	0	1
RF2	oilseed rape	0.01	0	1	1	0	1	1	1	0	0
HCN92	oilseed rape	0.01	1	0	1	0	0	0	0	1	0

The GMOTrack input data matrix format is a tab-separated file. This means that columns are separated by a special character “tab” and that the filename extension should be “.tab”. The data matrix should be formatted according to the following constraints:

File head (first three lines):

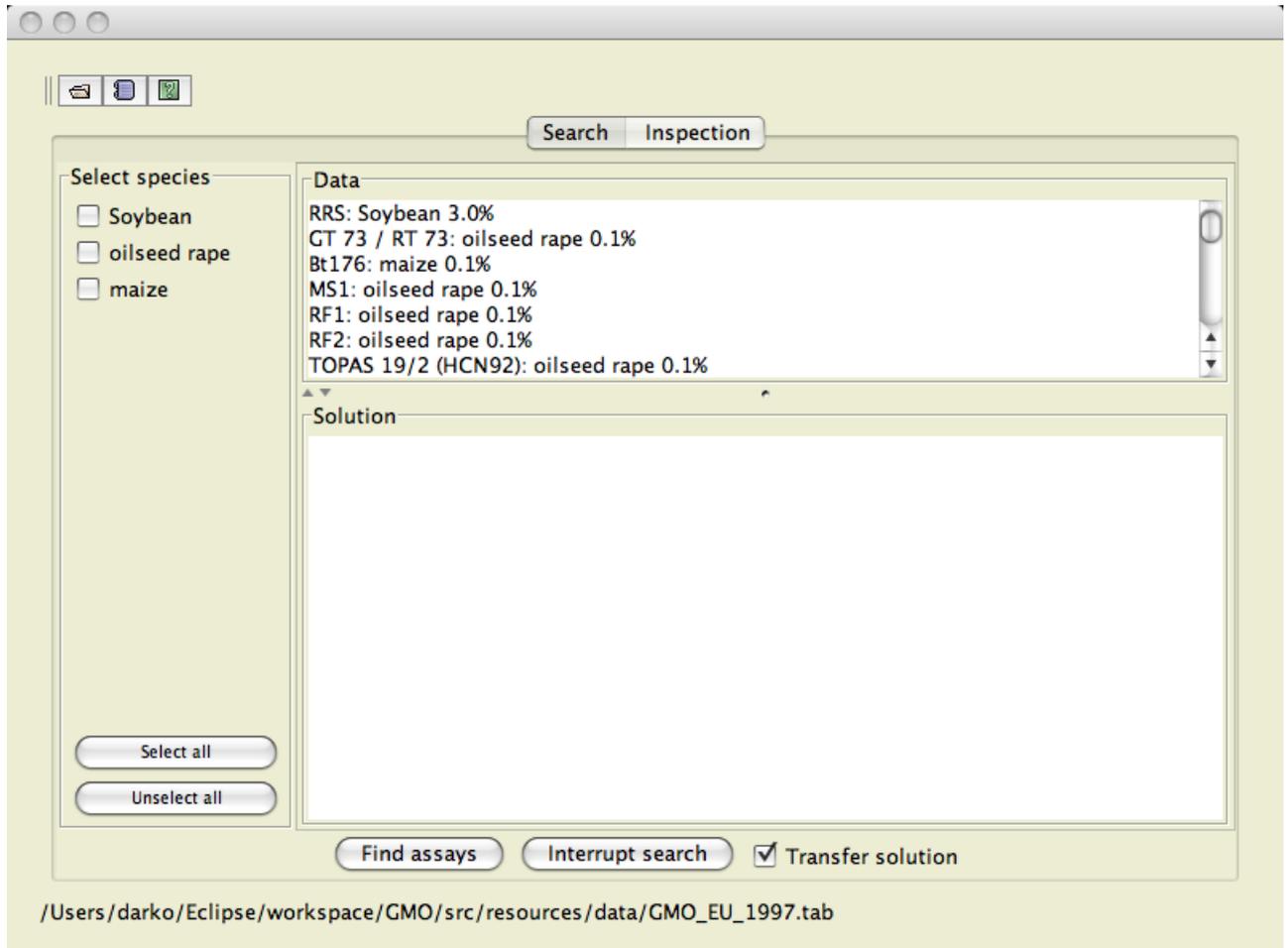
- The first line is reserved for column names:
 - First three are meta attribute names.
 - The first column name is “GMOname”.
 - The second column name is “species”.
 - The third column names are: "probability".
 - They are followed by assay names.
- The second line denotes the data type.
 - d for nominal values (all but the three probability columns)
 - c for numbers (the probability column).
- The third line denotes column type:
 - m for meta attributes (GMOname, species and probabilities)
 - empty for assays.

File body (from the forth line on):

- Each line of the file's body is one GMO.
 - The first column represents GMO names.
 - The second column must be species.
 - The third to the fifth columns contain probabilities.
 - The rest of the columns are the responses of the GMO to the assay:
 - 1 means positive response,
 - 0 means negative response.

Testing process

Once a definition is loaded, the similar content as in a window below appears.

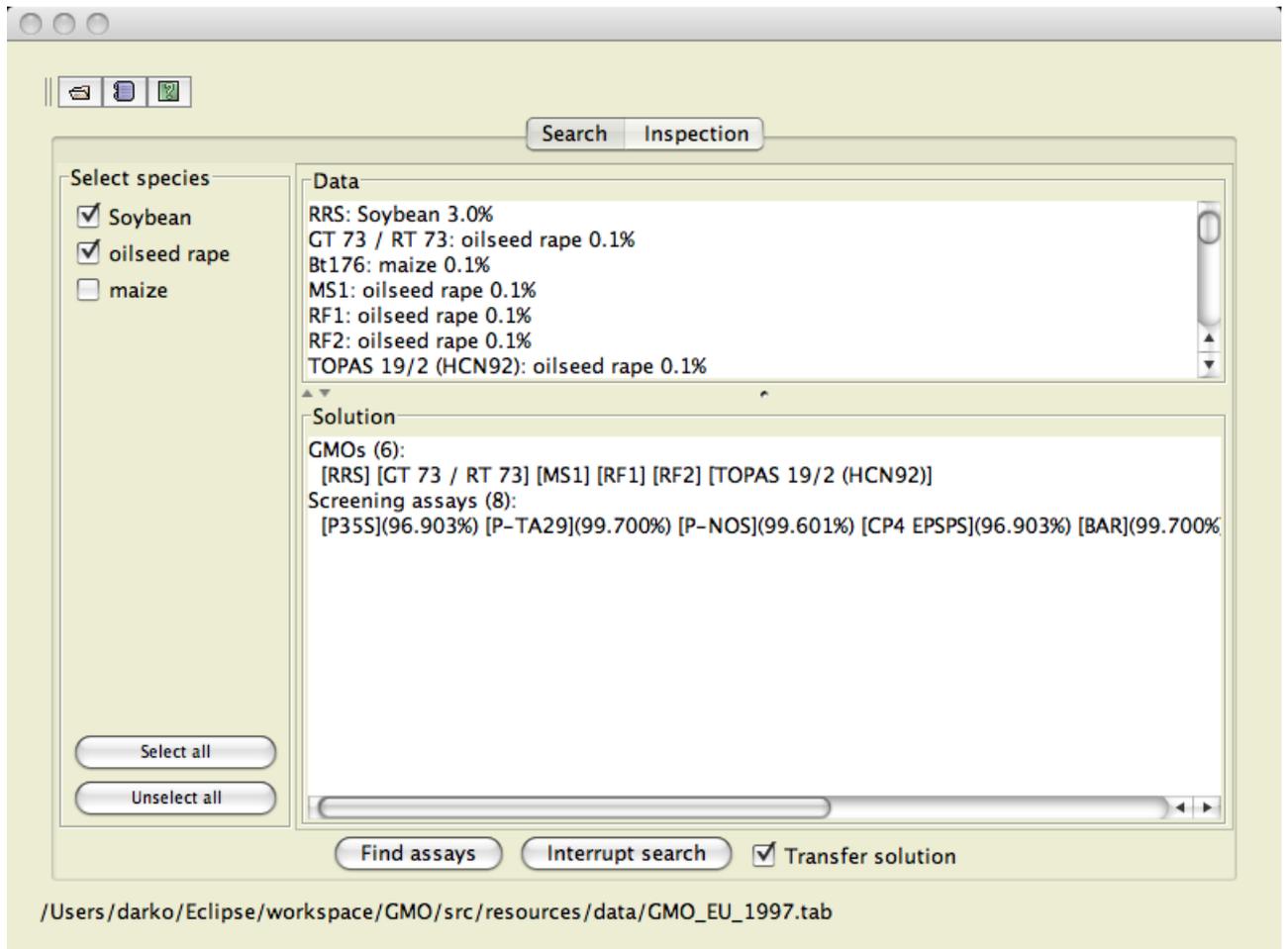


At the bottom of the window, a file name of the loaded definition is shown.

Search

The panel on the left side of the window contains all species available in the definition. Each of the species can be selected (and later deselected) by clicking on the check box. At the bottom of the left panel, there are two buttons which can speed up selections and deselections of the species.

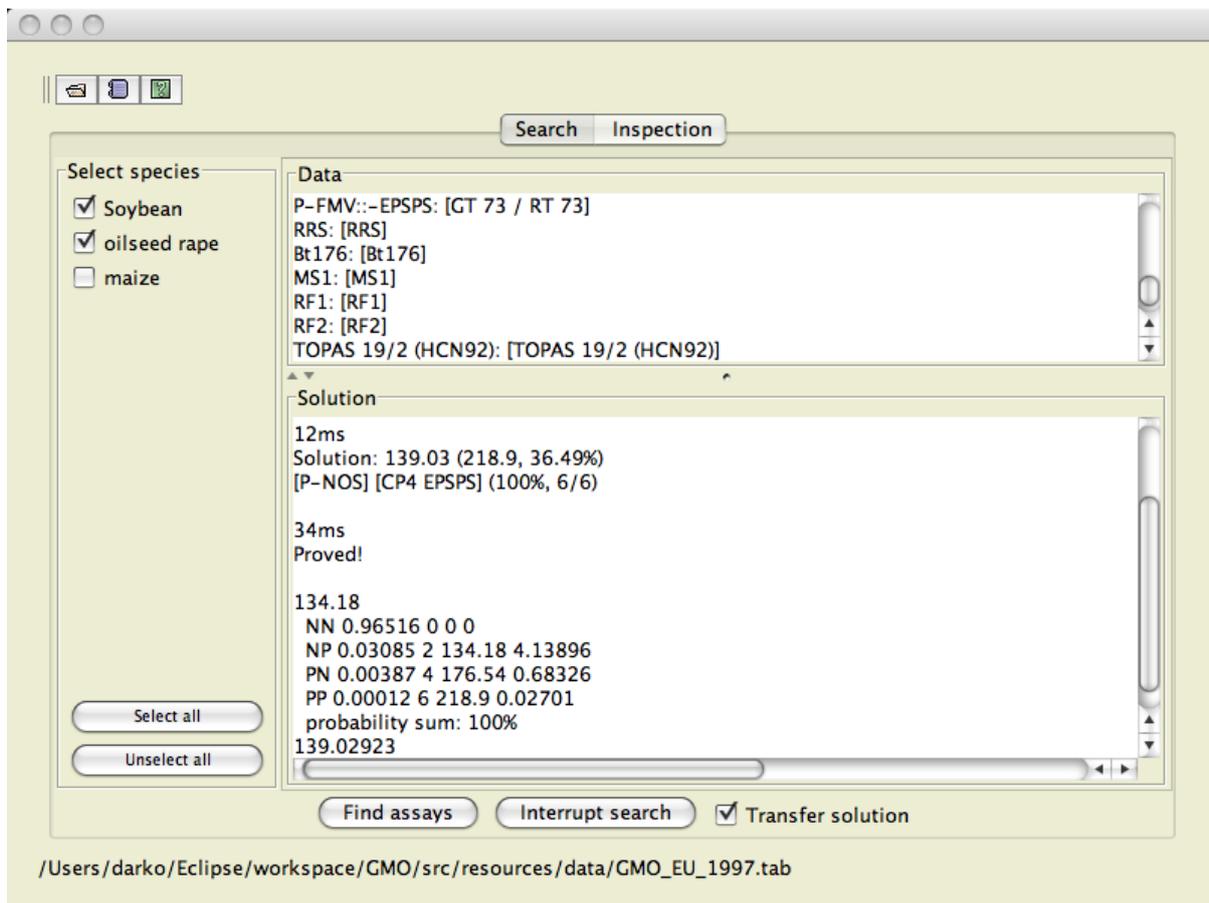
The central part of the window is divided into two text panels. The upper one contains information about the definition. The information is about chances of a GMO appearing in a sample (for instance, *RRS: Soybean 3.0%* means that it is a 3% chance that RRS is contained in Soybean) and about a GMO detection of a screening assay (for instance, *P35S: [Bt176] [TOPAS 19/2 (HCN92)] [RRS]* means that screening assay *P35S* can detect that one or more of *Bt176*, *TOPAS 19/2 (HCN92)*, *RRS* is present).



Selecting some of the species provides some additional information in the lower text panel. There is a list of possible GMOs (six possible GMOs in the example above) and a list of screening assays which can detect the possible GMOs. A probability of a negative result of a test is attached to each screening assay (in a case that the screening assay would be used alone).

In the lower part of the window, there are two buttons for handling a solution search. The command *Find assays* starts the search while the command *Interrupt search* forces an end of the search if the user is not willing to wait for a better solution anymore.

Launching a solution search by clicking on the *Find assays* button produces an additional output to the lower text panel.



The first three lines of the additional output reveal that the first intermediate solution was found in 12 milliseconds. Furthermore, the solution cost is 139.03 which is 36.49% of the cost of using event-specific assays only (218.9). And finally, the selected screening assays are *P-NOS* and *CP4 EPSPS*, followed by a percentage of GMO coverage (100%) and a ratio of a number of covered GMOs and a number of all GMOs (6/6). The first solution in the example happened to be the final solution. If some improvements of the solution would happen, additional output with the same format of information would be produced.

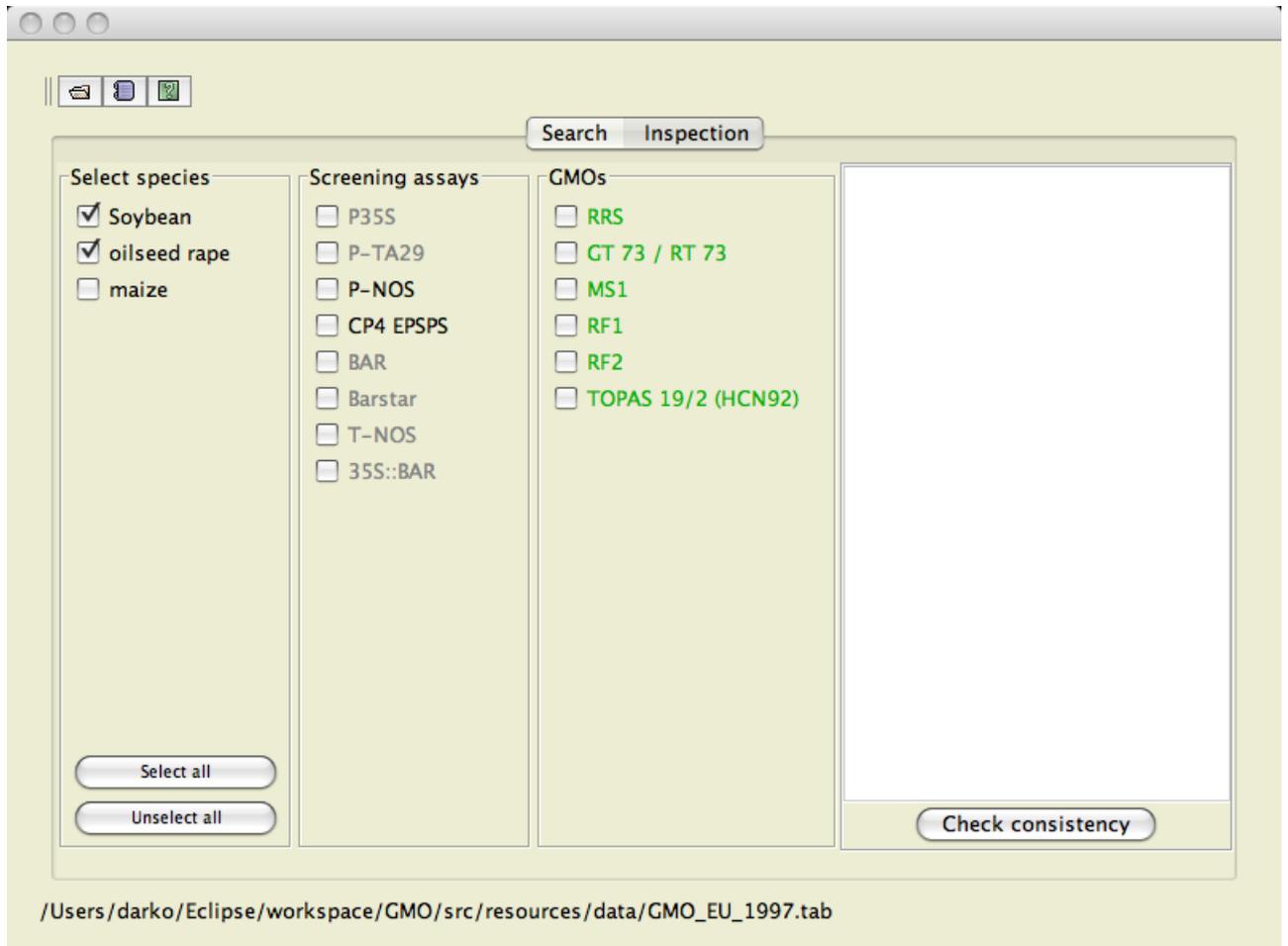
After the final solution, the time needed to prove an optimality of the solution is reported. An alternative to text *Proved!* is text *Interrupted!* if a solution search was interrupted.

The last part of the output represents a detailed breakdown of the cost of the final solution (either proved to be optimal or intermediate if a search was interrupted). It starts with a fixed cost of screening assays test (134.18), followed by all possible test outcomes (N stands for negative, P for positive) and the total cost (139.02923) which is the same as the cost reported in the last outputted solution.

A line with an outcome is composed of a combination of N and P, a probability for the combination to happen, a number of event-specific assays which should be tested in such a case, a cost of such an event-specific test, and a cost which is normalized with the probability. The line containing *probability sum: 100%* means that all combinations were considered. The probability sum can be lower than 100% (usually for something around 0.001%) because of one of the search space pruning techniques involved (other search space pruning techniques are complete). It could be argued that in such cases *Proved!* is misleading. However, because of the granularity of the search space, it is correct for all practical purposes.

Inspection

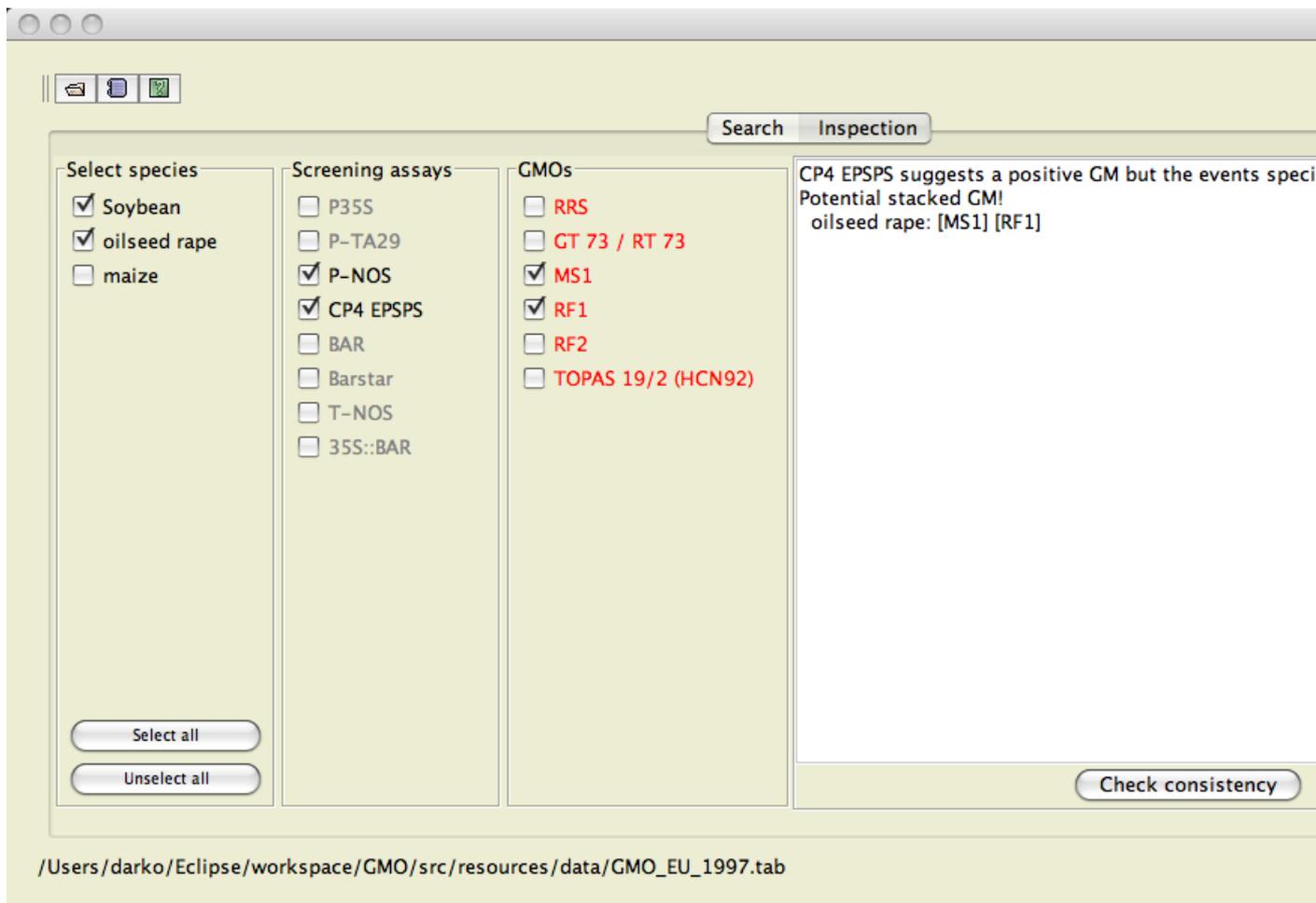
The previous sections of this manual were dedicated to the *Search* panel of the software. This section is dedicated to the *Inspection* panel of the software. It is independent of the *Search* panel and it can be used without prior use of the *Search* panel. However, when *Transfer solution* check box is ticked, the solution data is prearranged in the *Inspection* panel.



The left panel is the same as in the *Search* panel.

The middle panel of the check boxes contains all screening assays. The assays obtained from the solution search are black while the others are gray. The status of each assay can be changed by clicking on the right mouse button. The selected assays (either obtained by a solution search or selected manually) represent screening assays for an actual test. After the actual test is performed, the result of the test is entered by ticking check boxes of positive screening assays.

Let's assume that in our experiment the both screening assays were positive (a probability for this to happen is 0.012% as can be seen in the *Search* panel). Therefore, *P-NOS* and *CP4 EPSPS* have to be checked.



All uncovered GMOs (in this example all GMOs are uncovered) which are listed in the right panel of the check boxes are colored red. A red colored GMO indicates that it should be tested with an event-specific test. Again, after the actual test of event-specific assays is accomplished, a result is entered by ticking check boxes of positive event-specific assays.

Let's assume that *MS1* and *RF1* were positive and therefore they are ticked. To conclude the inspection, a *Check consistency* command can be performed. As can be seen in the window above, two warnings are listed in text panel. The first one alerts inconsistency between results of the screening assays test and the event-specific test. The second one alert a potential stacked GMOs because two of the GMOs are located in the same specie. One more warning is possible. It alerts an inconsistency between results of screening assays.

User's environment

A software execution can be exited by a click on a close icon of a window or a similar action which depends on an operating system. A state of a user's environment within a program is saved in a file named *gmo.dsk*. It is located in the same directory as *gmo.jar*. When the program is started again, the last working definition is automatically loaded and a list of recently used definition is established.