

Predict

analyse uncertainties from weather



Introduction

This application is an add-on to the CANDY model and can be used to predict the dynamics of selected variables from a given system state depending on different weather data that usually will represent several future possibilities. It is recommended to first read the candy manuals to better understand the usage of this tool.

The program can be used as a desktop application from the graphic user interface (GUI) or embedded in a script with several call-parameters including an auto-list option that control the detailed behaviour. In the directory where the program is located (the working directory) must also be the file `pg4candy.ini` (ini-file) with the connectivity information for the available databases and the general candy settings.

Using the GUI

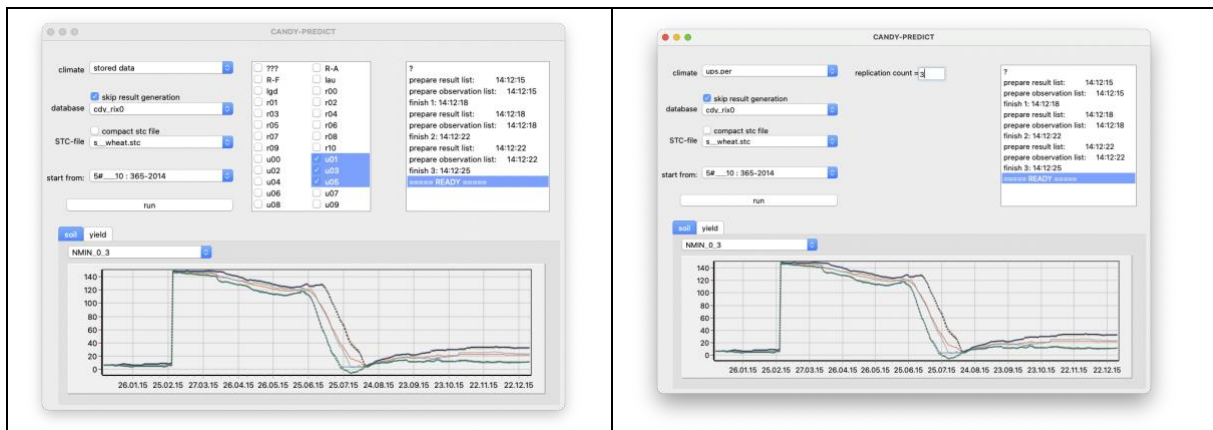


Figure 1: user interface with fixed climate data (left) or randomly generated data (right)

The required inputs contain the climate scenario, the candy database, the file containing system states and the selection of a certain record within this file which represents the initial time of the following model simulations with different weather data. Weather data may be selected from the database but must fit the date that is given by the initial system state. There is an explicit input of the number of replications required if the climate selection shows a generator file.

Start with manual input with the upper left combo-box (climate). It contains the entry “stored data” and all weather-generator files (*.per) located in the working directory. Next step is the selection of a database from the second combo-box. If the checkbox above this combo-box is activated, you can go directly to the analysis of the results from the previous run without repeating the simulations – this may be required if you rashly closed the program. After selecting the database, you may check the climate stations that should be included – if the option “stored data” was chosen. In the other case, if you selected a weather generator, it is necessary to specify the number of repetitions (i.e. randomly generated weather years). The third step is the selection of an STC-file containing the system state that you want to use as initial condition. This file may be compacted to contain only valid records if the appropriate checkbox is activated. After closing this combo-box the content of the selected stc-file is analysed. All available initial conditions are listed in the last combo-box (start from) in a format like 5#__10 : 365-2014 where

- 5 is the record number of this state-record required to find it at run-time
- __10 is the specification of plot (1) and subplot (0) in a string of length 5
- 365 is the day number; the state is valid at the end of this day
- 2014 is the year

Click the run-button to start a sequence of candy simulations that will all start at the selected date and with the system state of this data resulting from a previous candy simulation. The simulation interval is limited to the end of the year where the currently growing crop is harvested. A listbox indicates the progress of the simulations. After the last message (READY) the results can be checked for soil- and yield-parameters. Uncertainty of soil parameters is represented with several curves for maximum,

minimum, arithmetic mean (orange), central value (red), and mean \pm standard deviation (dotted lines). Yield results are shown as set of columns and as cumulative probability. All simulation results are contained in table *rs_predict* and are available for individual analysis. The content of *rs_predict* will be deleted before each new run. The list of results is fixed and contains elements of nitrogen and water in soil as well as yield of main- and by-product. The *predict* simulation records results from the list named *predict* in table *cdy_rslt_lst*. Its standard content is shown here:

<i>Res_id</i>	<i>Res_name</i>	<i>Unit</i>
112	<i>NMIN_0_3</i>	<i>kg/ha</i>
113	<i>NMIN_3_6</i>	<i>kg/ha</i>
114	<i>NMIN_6_10</i>	<i>kg/ha</i>
108	<i>SoilWat_0_3</i>	<i>mm</i>
109	<i>SoilWat_0_10</i>	<i>mm</i>
115	<i>SW_PRFL_STC</i>	<i>mm</i>
104	<i>BAT</i>	<i>d</i>
102	<i>N_leach</i>	<i>kg/ha</i>
...and if <i>rs_at_hrv=1</i> then also:		
21	<i>by-product yield</i>	<i>dt FM/ha</i>
22	<i>main-product yield</i>	<i>dt FM/ha</i>

Mind: res_id 21 and 22 are not included in tables *cdy_rslt* and *cdy_rslt_sel*

Following parameter settings within the ini-file are important for the proper work this tool:

randseed=0 : to cause complete randomization of weather generation

rs_at_hrv=1 : to include natural yield of main- and by-product in result recording

Furthermore, it is recommended to avoid recording stc-records by setting *stc_record=0*.

Calling from a script, the application understands the following parameters:

parameter	meaning	example
DB	Candy database	DB=demo_db
W	Weather data	W=ups.per W=u0% W="(‘u01’;u02’;u03’)"
ST	File of system states	ST=s__wheat.stc
SRN	Record number of initial state	SRN=5
GO	Starts immediately the simulation run	GO
AUTO	alternative to start the processing of a complete task list given as a table in schema <i>predict</i> (here table <i>predict.auto_list</i>)	AUTO=auto_list
PPQ	A pre-processing sql script started before each run of an autolist record	PPQ=preproc.sql
POQ	A post-processing sql-script started after each run of an autolist record	POQ=postpro.sql

Result of a single call can be found in table `rs_predict` where the column `id` is just a running number and `obj_id` refers to the sequence of weather data sets

```
CREATE TABLE public.rs_predict (  
    id serial4 NOT NULL,  
    "date" timestamp NULL,  
    res_id int4 NULL,  
    obj_id int4 NULL,  
    value float8 NULL  
);
```

After an auto-run the results of all individual runs are collected in table `predict.rs_autolist` where the column `pid` refers to a certain record in the auto-list table

```
CREATE TABLE predict.rs_autolist (  
    pid int4 NOT NULL,  
    "date" timestamp NULL,  
    res_id int4 NULL,  
    obj_id int4 NULL,  
    value float8 NULL  
);
```

The structure of an auto-list table contains all required input information as described above but may contain additional columns if required. Here is the least necessary structure:

```
CREATE TABLE predict.auto_list (  
    pid int4 NULL,  
    stc_file varchar(12) NULL,  
    s_reco int4 NULL,  
    wgen varchar(7) NULL,  
    nrep int4 NULL,  
    simtag int4 NULL  
);
```

Like in the CANDY system, the `simtag` column displays the current processing state.

- 0 : record has still to be processed
- 1 : processing of this record is finished
- -1: record is currently being processed

This way, it is possible to create pre- and post-processing sql-scripts that directly relate to the currently handled record.

