
Table of Contents

The toggle switch example	1
Generate and fit a family of models	1
Execute OED for model discrimination	3
Generate new pseudo experimental data	4
Second iteration: generate and fit a family of models	5
Second iteration of OED for model discrimination	6

The toggle switch example

This toggle switch example was generated with a Hill kinetics model trained to a subset of the data provided in Lugagne et al. 201x. The example is based on pseudo experimental data which means that all data is generated by the model we have previously trained. This is a standard procedure that serves as a means of benchmarking the algorithm.

```
clear variables;  
clc;  
close all;  
noise_pseudo_data=0.05;
```

A number of settings is defined in the default configuration file. The assertiveness of these settings is problem dependent. Settings such as the number optimization solver settings, the experiments considered, or number of models (sparsity cases) have default values that should be tailored to the problem.

```
SBL_config_defaults;
```

As an example we modify `sbl_config.exp_idx` for considering only experiments 1 and 2.

```
sbl_config.exp_idx=1:2;
```

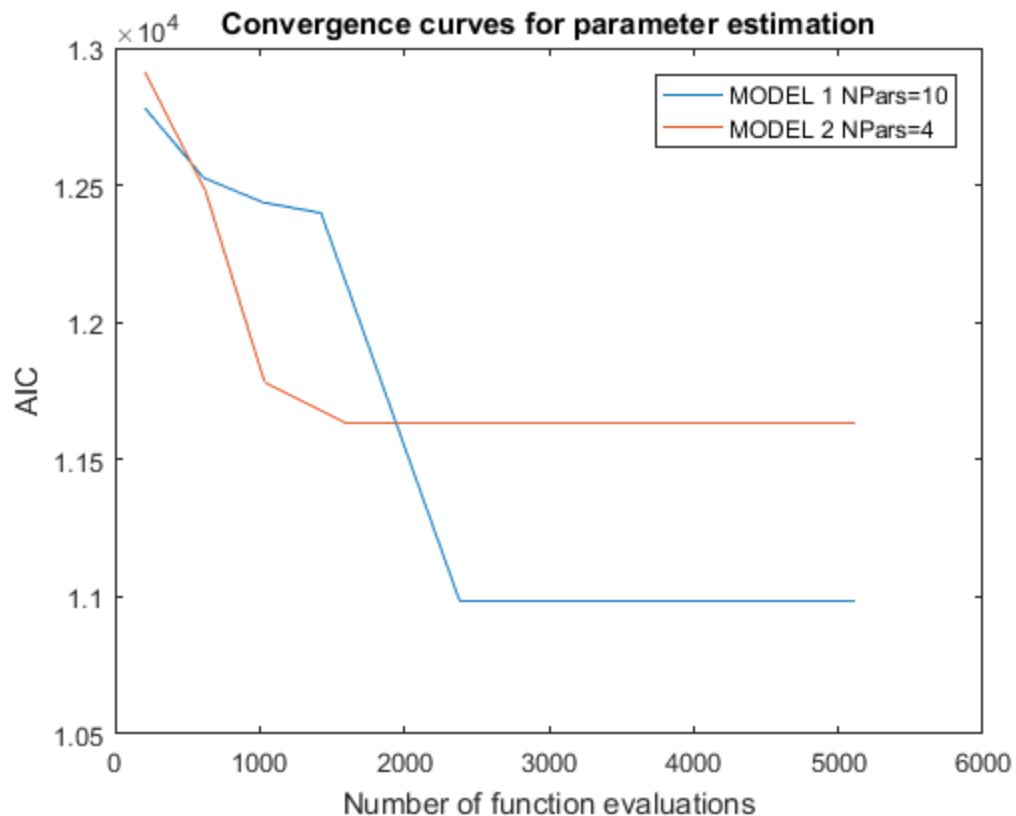
Generate and fit a family of models

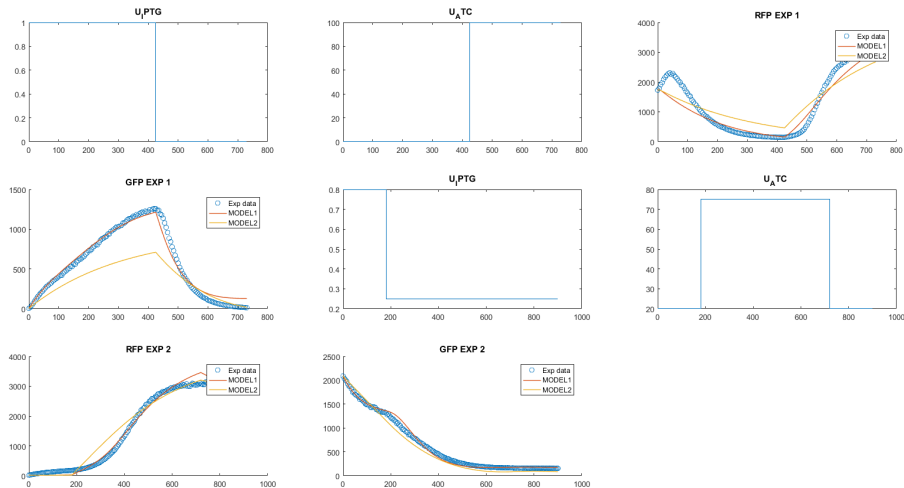
We generate a family of models with SBL and fit with AMIGO+scatter search. The convergence curves are given by scatter search. Time courses for each modeled observable are also plotted along with the pseudo experimental data. There is substantial progress made by the parameter estimation process.

```
MODELS=SBL_gen_model_family(sbl_config);  
SBL_plotFamilyFit(MODELS);  
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
```

```
22-Aug-2019 11:10:04 | loop iter: 1, generating SBL data  
22-Aug-2019 11:10:04 | file: experimental_data_loop_1.csv was  
    successfully imported | 2 experiments were selected  
22-Aug-2019 11:10:04 | loop iter: 1, running SBL  
running sparsity case: 1/2  
running SBL on state: 1/2
```

SBL iter: 1/10 took 0.934112 sec
 SBL iter: 2/10 took 1.77113 sec
 SBL iter: 3/10 took 1.85594 sec
 SBL iter: 4/10 took 2.50706 sec
 SBL iter: 5/10 took 1.23525 sec
 SBL iter: 6/10 took 2.06935 sec
 SBL iter: 7/10 took 2.14355 sec
 SBL iter: 8/10 took 2.16152 sec
 SBL iter: 9/10 took 1.75861 sec
 SBL iter: 10/10 took 1.13976 sec
 runnging SBL on state: 2/2
 SBL iter: 1/10 took 1.58408 sec
 SBL iter: 2/10 took 0.518963 sec
 SBL iter: 3/10 took 0.491092 sec
 SBL iter: 4/10 took 0.456028 sec
 SBL iter: 5/10 took 0.449554 sec
 SBL iter: 6/10 took 0.479244 sec
 SBL iter: 7/10 took 0.492646 sec
 SBL iter: 8/10 took 0.501306 sec
 SBL iter: 9/10 took 0.470418 sec
 SBL iter: 10/10 took 0.476625 sec
 Elapsed time is 0.501144 seconds.
 state: x_1 zero_th: 9.49847e-05 dict_num: 4 (0.249532%)
 state: x_2 zero_th: 9.49847e-05 dict_num: 6 (0.747198%)
 ODE simulation OK
 ...





Execute OED for model discrimination

Optimal experimental design for model discrimination seeks to find the experiment that maximizes the predicted different between the models.

```
modelsAfterOED=OED4SBLdiscrimination(MODELS,sbl_config);
SBL_plotDiscriminationResult(modelsAfterOED);
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
```

AMIGO2, Copyright @CSIC

AMIGO2_R2017a [March 2017]

*Date: 22-Aug-2019

----->Pre processing....this may take a few seconds.

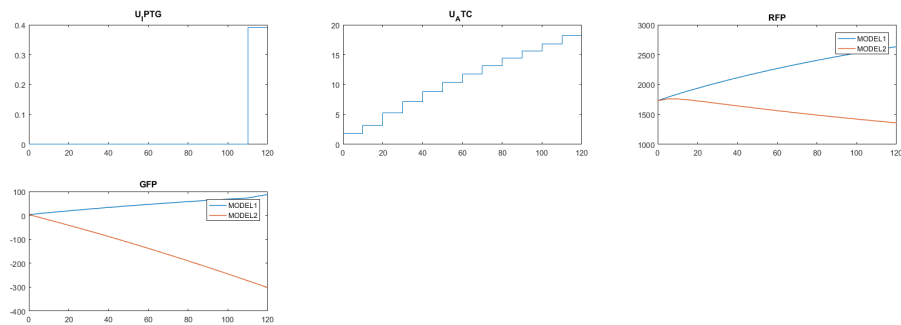
----->Checking inputs....

```
n_exp
exp_type
n_obs
obs_names
obs
exp_y0
t_f
n_s
t_s
u_type
u_interp
n_steps
t_con
```

```

u_min
u_max
u
exp_data
...

```



Generate new pseudo experimental data

```

data_file2=fullfile(pwd, 'Data', 'experimental_data_2.csv');
gen_pseudo_data(modelsAfterOED{1}
{1}.exps,noise_pseudo_data,data_file2,'modell1');
sbl_config.data_dir_name = 'Data';
sbl_config.data_file_name = 'experimental_data_2.csv';

```

```

*****
      AMIGO2, Copyright @CSIC
      AMIGO2_R2017a [March 2017]
*****

```

*Date: 22-Aug-2019

----->Pre processing....this may take a few seconds.

----->Checking inputs....

```

n_exp
exp_type
n_obs
obs_names
obs
t_f
n_s
t_s

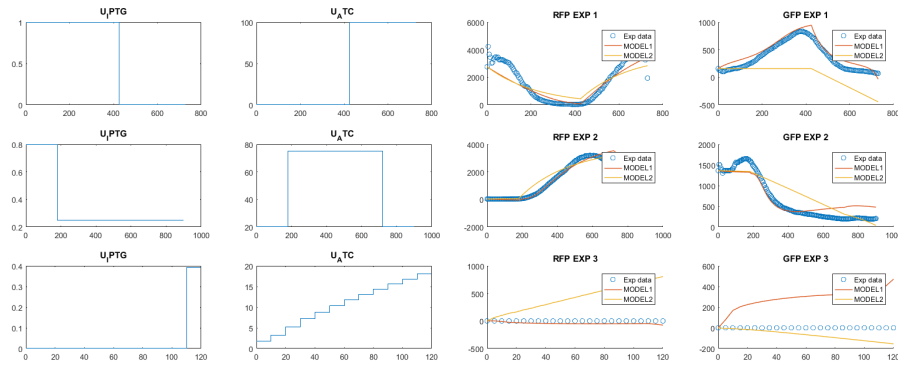
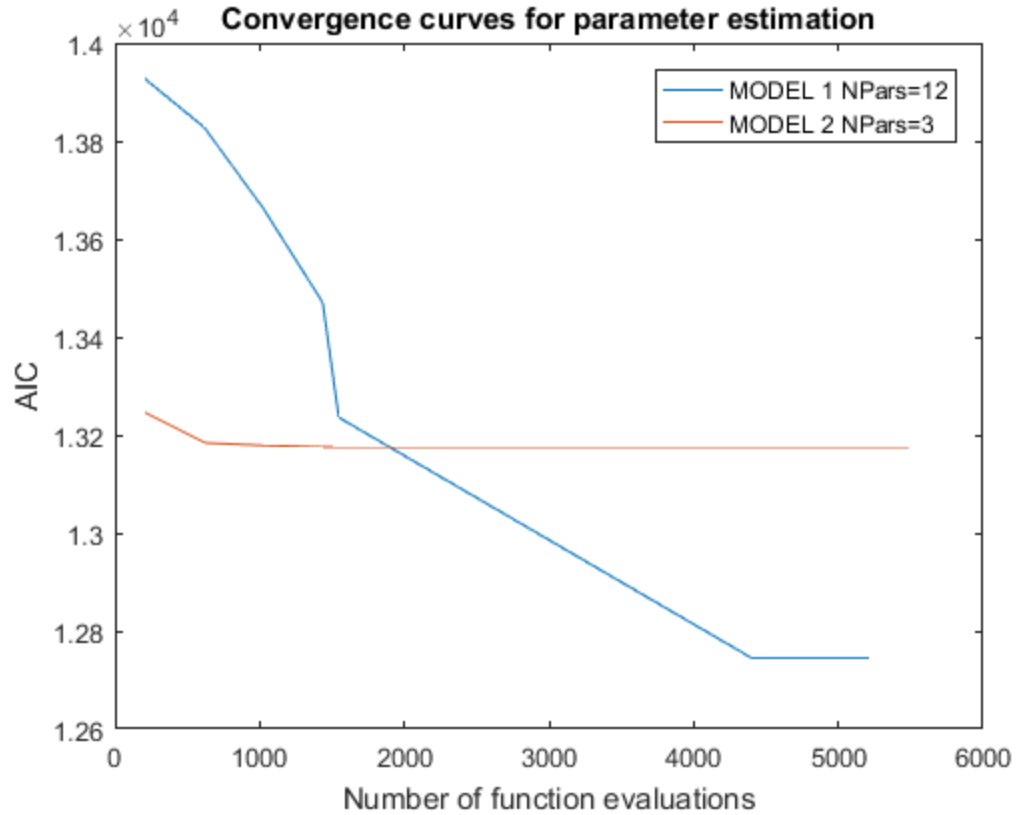
```

```
u_interp
t_con
n_steps
u
data_type
noise_type
exp_data
error_data
exp_y0
...
```

Second iteration: generate and fit a family of models

```
sbl_config.exp_idx=[1 2 8];
% %% Generate and fit a new family of models
MODELS=SBL_gen_model_family(sbl_config);
SBL_plotFamilyFit(MODELS);
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);

22-Aug-2019 11:13:48 | loop iter: 1, generating SBL data
22-Aug-2019 11:13:48 | file: experimental_data_2.csv was successfully
imported | 3 experiments was selected
22-Aug-2019 11:13:48 | loop iter: 1, running SBL
runnging sparsity case: 1/2
runnging SBL on state: 1/2
SBL iter: 1/10 took 1.27991 sec
SBL iter: 2/10 took 1.0065 sec
SBL iter: 3/10 took 1.0357 sec
SBL iter: 4/10 took 0.999213 sec
SBL iter: 5/10 took 0.974128 sec
SBL iter: 6/10 took 0.999237 sec
SBL iter: 7/10 took 1.01408 sec
SBL iter: 8/10 took 0.96876 sec
SBL iter: 9/10 took 0.98798 sec
SBL iter: 10/10 took 0.943812 sec
runnging SBL on state: 2/2
SBL iter: 1/10 took 0.539054 sec
SBL iter: 2/10 took 0.553278 sec
SBL iter: 3/10 took 0.506341 sec
SBL iter: 4/10 took 0.538027 sec
SBL iter: 5/10 took 0.511941 sec
SBL iter: 6/10 took 0.54133 sec
SBL iter: 7/10 took 0.52264 sec
SBL iter: 8/10 took 0.528293 sec
SBL iter: 9/10 took 0.547305 sec
SBL iter: 10/10 took 0.530589 sec
Elapsed time is 0.557842 seconds.
state: x_1 zero_th: 9.49847e-05 dict_num: 6 (0.374298%)
state: x_2 zero_th: 9.49847e-05 dict_num: 6 (0.747198%)
ODE simulation OK
...
```



Second iteration of OED for model discrimination

Optimal experimental design for model discrimination seeks to find the experiment that maximizes the predicted different between the models.

```
modelsAfterOED=OED4SBLdiscrimination(MODELS,sbl_config);
```

```
SBL_plotDiscriminationResult(modelsAfterOED);
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
```

```
*****
```

```
    AMIGO2, Copyright @CSIC
```

```
    AMIGO2_R2017a [March 2017]
```

```
*****
```

```
*Date: 22-Aug-2019
```

```
----->Pre processing....this may take a few seconds.
```

```
----->Checking inputs....
```

```
n_exp
```

```
exp_type
```

```
n_obs
```

```
obs_names
```

```
obs
```

```
exp_y0
```

```
t_f
```

```
n_s
```

```
t_s
```

```
u_type
```

```
u_interp
```

```
n_steps
```

```
t_con
```

```
u_min
```

```
u_max
```

```
u
```

```
exp_data
```

```
...
```

