Freeman's KIII - Reference

How to get back here

The `Help→Reference` option will open this PDF with the default desktop application.

The application

The application was made using Java, based off of work of Piazentin (2015) as an attempt to make the Freeman KIII set simulator more accessible and easier to configure. The source code for this tool can be found under the free software license at https://github.com/wqferr/ksets

The main window

The main application window consists of 3 text fields, of which only 2 are editable.
The first (non editable) text field refers to the name of the file which the model was last saved to, or where it was loaded from (or blank if it wasn't saved). The second one refers to the name of the dataset which will serve as input during the simulation. The last one defaults to `out.csv` and is the name of the file the model output will be saved to.
Upon setting both the input dataset file name and output `.csv` file name, pressing `Run` will run the simulation and automatically save the output to the file name specified.

Model file manipulation

Creating a model

As mentioned, this application was made for easier manipulation of Freeman KIII sets. Therefore, the most basic function is that of creating a new KIII model. This function can be found under `File→New`. The system will then prompt the user for the sizes of 3 different layers of KII sets, which will compose the KIII to be created. Layer 0, the one whose size is read first, is the input layer. Upon confirming the sizes, a new KIII model will be available for manipulation.

Saving a model

There are two ways to save a model to a file, namely `File→Save` and `File→Save as`. `Save as` will prompt the user for a file name. All information about the model will be saved in the `FILENAME.kset` file, where `FILENAME` is the file name provided by the user. If `Save as` was never used, `Save` will be functionally the same. Otherwise, instead of prompting for a file name, it will use the name of the last file the model was saved to.

Loading a model

The user may also load a model given a `.kset` file, using the `File→Open`. The option will prompt the user for a file name. If the user enters `FILENAME`, the application will try to load `FILENAME.kset`.

For the rest of this document, "the model in use" will refer to the most recently loaded or created model in the current session. It is said that "no model is in use" if none of these actions has been completed successfully in the current session.

Configuring the model

Before running the model, it would be logical to configure parameters and train the model with datasets.

Setting parameters

In order to set a parameter of the model, the user must select `Edit→Set parameter`. The system will open a pop up window for the user to type required information. The input for the first field is the parameter identifier, and the input for the second one are any additional arguments the setting requires. The available parameters are as follows:

- `layer_training` Boolean array which dictates the layers that will have their weights updated and those that will be left as is, such that

  $$\texttt{layer\_training[}i\texttt{]} \text{ is } \texttt{true} \implies \texttt{layer[}i\texttt{]} \text{ will be trained } \forall i \in \{0, 1, 2\}$$

  Additional arguments consist of three boolean values (either "`true`" or "`false`") separated by whitespace.
- `learning_rate` Learning rate parameter used during unsupervised training. Additional argument consists of a single real value.
- `detect_instability` Whether or not to look for instabilities mid-simulation. Additional argument consists of a single boolean value (either "`true`" or "`false`").
- `output_layer` The layer whose output will be recorded as the output of the whole model. Additional argument consists of a single integer in $\{0, 1, 2\}$, where 0 is the input layer.

All these parameters have default values and are not required to be set before running the model. A quick reference table for parameter information and usage can be found below:

Name	Identifier	Arguments	Default value
Layer training flags	`layer_training`	Three boolean values (either "`true`" or "`false`") separated by spaces	`false false false`
Learning rate	`learning_rate`	A real number	0.005
Detect instability	`detect_instability`	A boolean value (either "`true`" or "`false`")	`false`
Output layer	`output_layer`	An integer in $\{0, 1, 2\}$	2

Training

The user may also train the model by using a `.csv` file containing a dataset. This may be done either before or after setting the parameters, but doing so before setting the parameters may lead to unexpected results in output.
By selecting `Edit→Train`, the application will prompt the user for which layers will be trained (already set to the values previously attributed to `layer_training`, but can be changed), and the name of an input dataset. The input file must be a `.csv` file, with each row containing exactly as many numbers as the size of the input layer of the model (layer 0).
Upon confirming the settings, the model will be automatically trained.

Visualization

The user may also wish to view information about the model in use or a given dataset.

Model visualization

By selecting `View→Model`, the user will have access to information about the model in use. About the model itself, the function will display:

- The output layer
- The learning rate
- If it is set to detect instability

For every layer in the model, the system will display:

- If it is the output layer
- Its size
- If it is being trained
- A button to get additional information

The button will display the internal weights of the layer it is associated with.

Dataset visualization

By selecting `View→Dataset`, the system will prompt the user for a dataset file name. After confirmation, the system will display the dimensions of the dataset in a `rows×cols` format, as well as every row in the dataset.

The text interface

The application also comes with a built-in text interpreter for easier integration with other applications. It can be accessed by running it with the flag `-t` or `--text`.
Most of the previously mentioned functionalities are present in text mode, and can be used by inputting one command at a time to standard input, followed by a whitespace and the required arguments. For example:

```
new 3 2 2
```

Will create a new model with layer sizes 3, 2 and 2.
A list of text commands can be found below:

Description	Command	Arguments
Create new model	`new`	The sizes of the 3 layers
Save model to file	`save`	The file name to save the model to
Load model from file	`load`	The file name to load the model from
Train model	`train`	The input dataset file name
Process data	`run`	The input dataset to process
Set parameter[1]	`set_param`	The identifier of the parameter to set, along with its new value
Exit[2]	`exit`	None

[1] The parameter identifiers for `set_param` are the same as the ones presented previously in this document.
[2] The text interpreter can also be terminated by an end of file signal.

Error handling

When an operation cannot be completed, an error message will be shown. In graphical mode, it will appear as a pop up window. In text mode, it will be displayed as a simple message to `stdout`.
The possible errors are:

"No kset loaded"

Why: an operation requires a model, but there is no model in use.
Fix: create a new model or load an existing one.

"Must provide sizes for all 3 layers of the kset"

Why: trying to create a model, but at least one text field was left empty.
Fix: leave no text field empty.

"Invalid integer"

Why: text field required an integer, but a fractional value was found.
Fix: input an integer value.

"Size must be a positive integer"

Why: text field required a positive integer, but a negative value or 0 was found.
Fix: input a positive integer value.

"Not enough arguments"

Why: operation required more arguments to be executed.
Fix: input all required arguments.

"No such file or directory"

Why: no file with such name was found.
Fix: input a file name or path which exists.

"Must specify a parameter to set"

Why: tried to set a parameter, but parameter name was left blank.
Fix: input the parameter name into the text field.

"Invalid layer number"

Why: operation required a layer number and either a value not in $0, 1, 2$ was given.
Fix: input a valid layer number.

"Invalid learning rate"

Why: tried to set the learning rate of a given layer, but the value given was either not a number or a
Fix: input a positive real number.

"No parameter by name"

Why: tried to set a parameter which does not exist.
Fix: input correct parameter identifier.

"No such command"

Why: an invalid command was given in text mode.
Fix: input a valid command.