Using TD

This gives an overview on how to run the td tool from the command line.

Requirements

TD is written in python . In order to run the tool on your system, python must be installed. You can find out which version of python is installed (or is the default) on your system with `python --version`.

In addition, td.py makes use of the below libraries:

- click
- pathlib
- cairo
- json
- math
- re
- sys

These packages need to be installed with `pip install [library]`.

Running TD

Generally, it is assumed that only python is installed on the host system. TD can then be executed from the command line with: `python td.py [FLAGS] [INPUT]` where *INPUT* is the json-formatted input file (see Input for details). The optional *FLAGS* can be used to further specify the visual output.

In cases where both python and python . are installed on the host system, it may be necessary to be specific about the python version to be used, this can be achieved using the `python3 td.py [FLAGS] [INPUT]` command.

The available option flags can be shown with `python3 td.py --help`. In the current version of TD, they include:

Option	Alternative	Description
--output TEXT	-o	Output file name. Default: INPUT.svg
--fontsize INTEGER	-s	Output font size (default)
--font TEXT	-f	Output font type (default: Arial)
--padding FLOAT	-p	Y-axis padding factor (default)
--condensed	-c	Show condensed daygrid
--timescale	-t	Show time scale
--graph	-g	Show dose graph
--ellipsis	-e	Reduce symbols in condensed output
--footnotes	-n	Show footnotes
--all	-A	All options, equivalent to -ctge
--debug	-d	Debug output
--help	-h	Show this message and exit.

Output file

The default output file name is the input file name (e.g., "test.json") with the .svg extension (i.e., "test.svg"). This can be overridden with the *--output* or *-o* option.

Font size and family

The default font is Arial point. Both font and size can be overridden using the *--font* (*-f*) and *--fontsize* (*-s*) options. The available font families depend on the fonts installed on the target system.

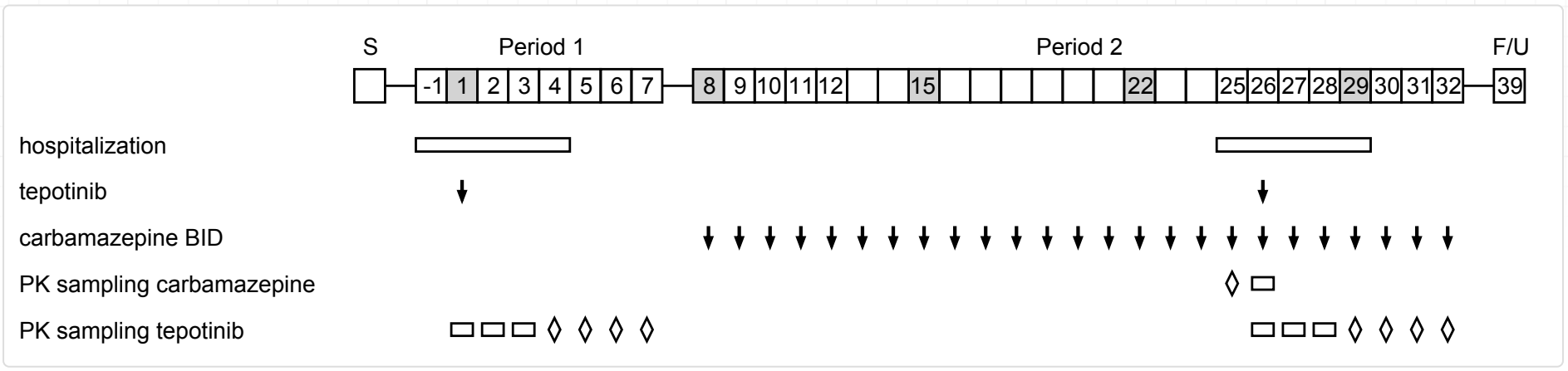
In addition, the generated svg file can be scaled in the target application (e.g., MS PowerPoint), and any element can be reformatted separately.

Padding

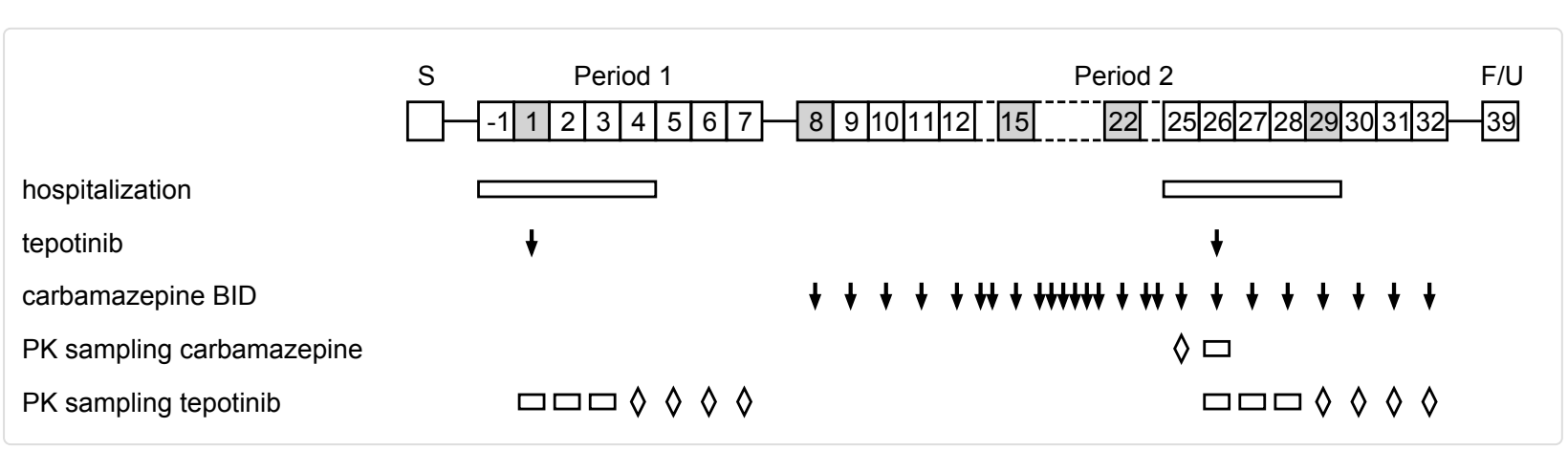
The parameter *--padding* (*-p*) increases or decreases the vertical space between period elements. The default of should work in most cases.

Condensed

This option can be used to compress the output horizontally. Days that have no daylabel (see "period formatting") assigned will be rendered narrower. As an example, the following figure was rendered normally (using `python td.py test.json`):



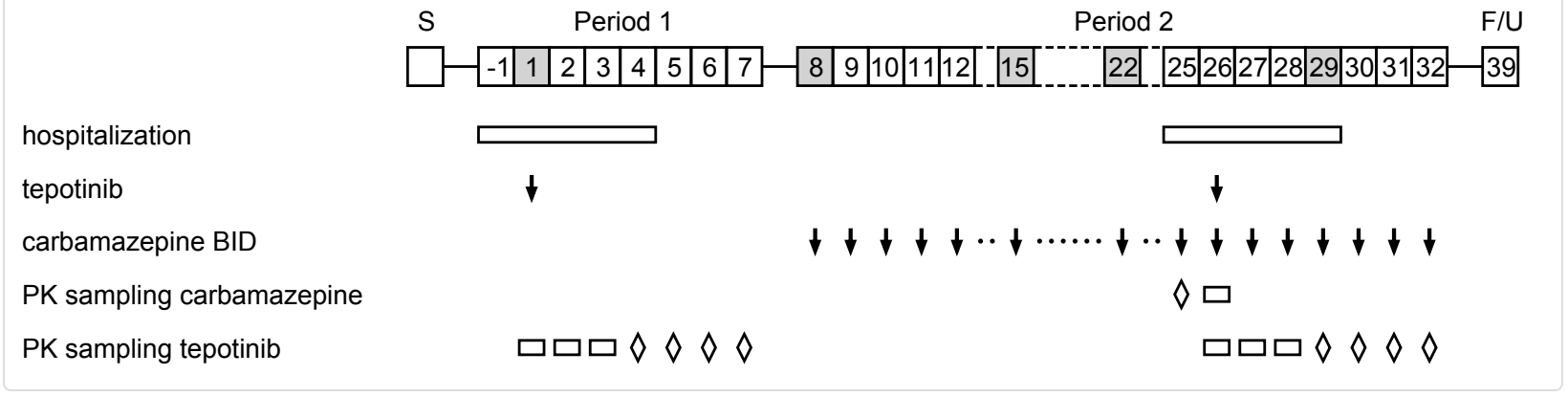
The below version was rendered with the *--condensed* option (`python td.py --condensed test.json`):



Ellipsis

In addition to condensed output, daily recurring procedure symbols that visually clutter the output can be reduced using the *--ellipsis* or *-e* option. This is only done for days that have no daylabel.

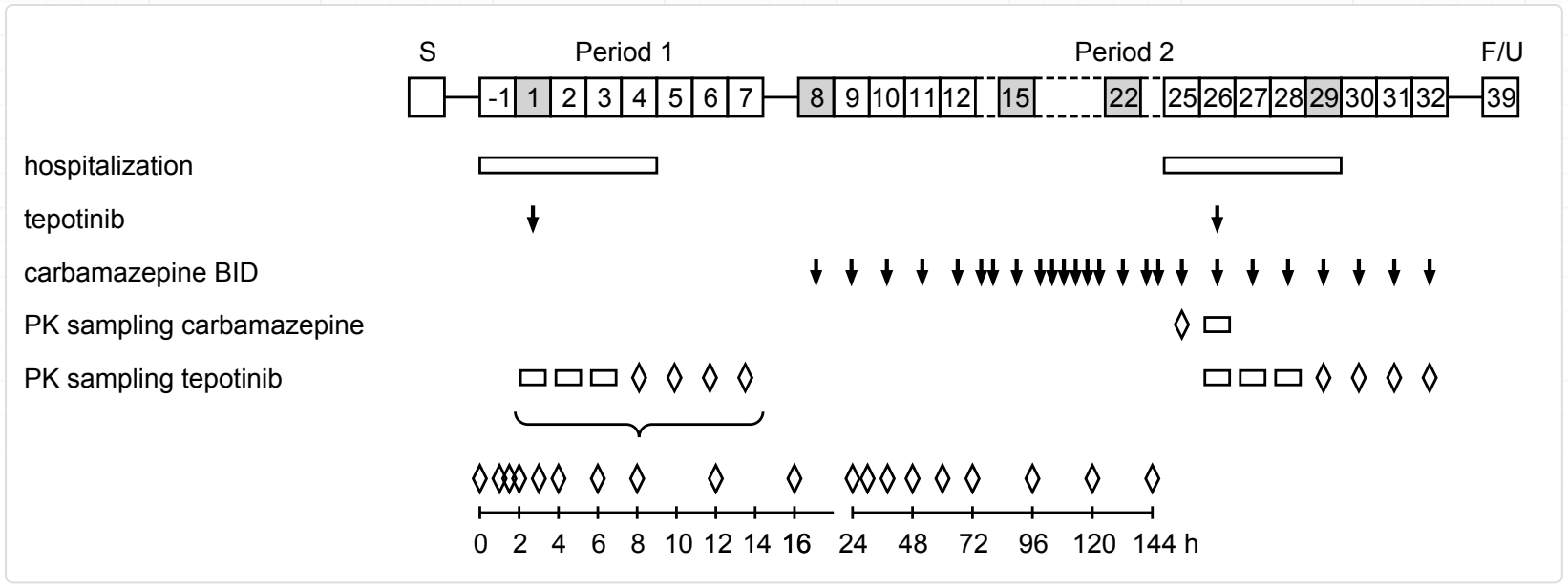
The below version was rendered using `python td.py -ce test.json` for a combination of condensed and ellipsis output:



Timescale

Procedures that have exact time information included in the input file, e.g., PK samplings (see "exact procedure times"), can be displayed with an inset figure underneath that shows the timescale detail.

The following output was generated using the `python td.py --condensed --timescale test.json` command:



Note that that there must be the `"timescale": "show"` line included in the json input file for this procedure (see "exact procedure times").

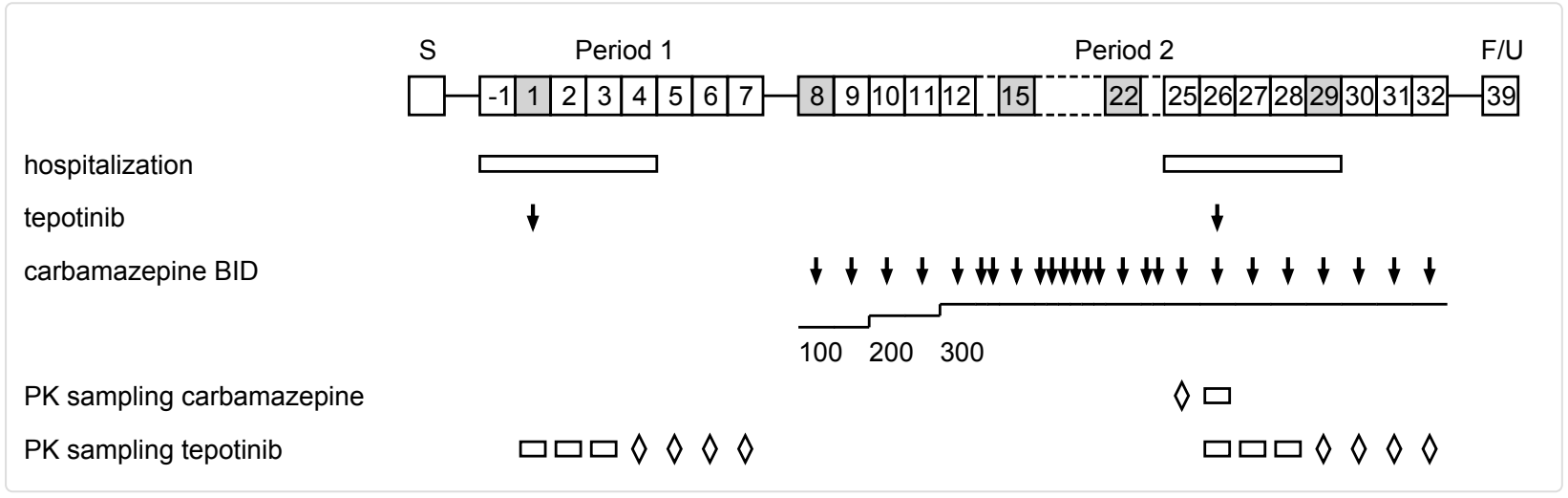
For visual clarity, display of a timescale should be limited to the last element (the one at the bottom) of a trial design visualization.

Dose graph

In cases where intraindividual dose escalation occurs, e.g., to phase a drug in or out, a dose graph can be shown underneath the administration symbols to indicate dose over time.

A prerequisite is that the respective dosing information is included per day in the input file (see Input).

The following output was created using the `python td.py --condensed --graph test.json` command to include a dose graph for carbamazepine:



Footnotes

If footnotes have been defined in the input file (see Input), they can be rendered in the output using the "--footnotes" (or "-n") option.