Current status of `DirectILLReduction`

- Basically usable for simple data reduction scripts.
  - Vanadium normalisation, empty container subtraction, but no Cadmium ATM.
  - Problems with the TOF axis solved.
  - Detector diagnostics and reporting.
  - Detector grouping.
- Parts of functionality unit tested.

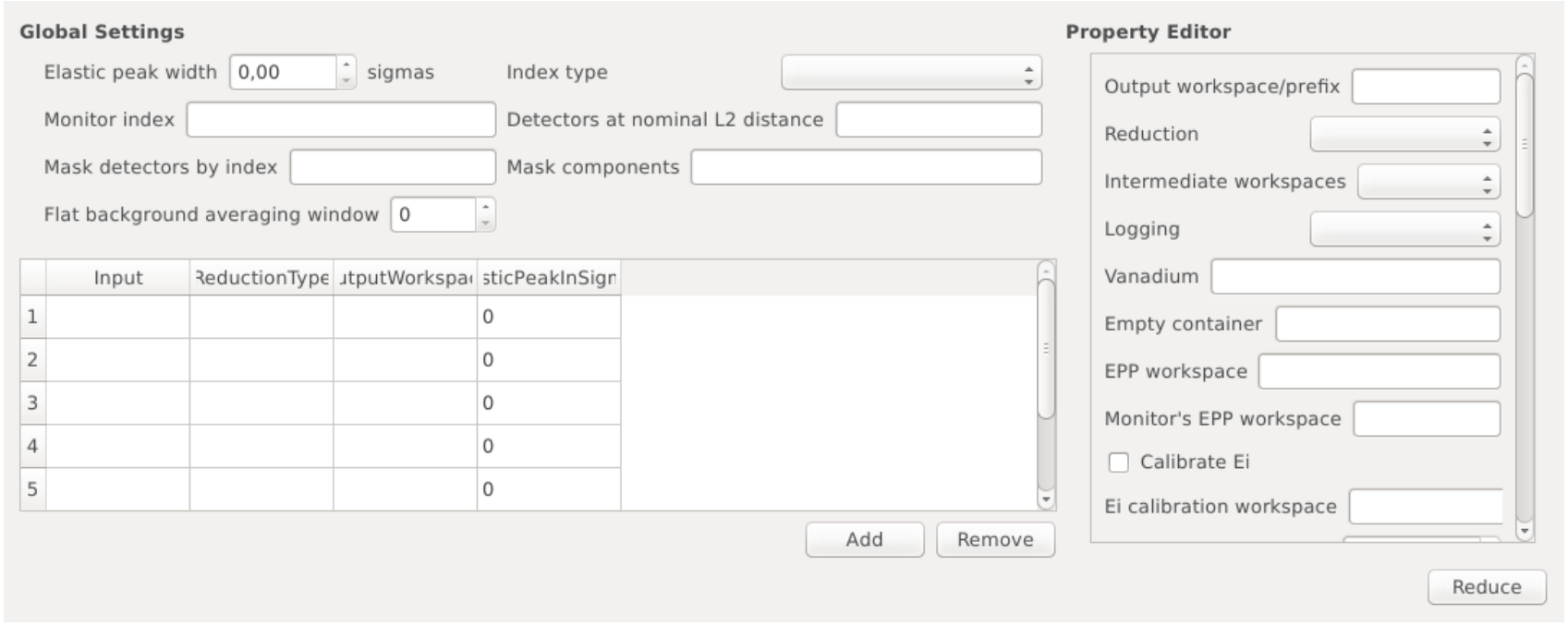
Self-shielding corrections

- Plan: use the Paalman-Pings correction algorithms.
- A prototype exists.
- Issues with `ApplyPaalmanPingsCorrection`.
  - Unconditional use of `RebinToWorkspace`, does not work with workspaces having varying bins.
  - Expects input workspace names in certain format.
- Normal empty container subtraction still possible.

Other functionality to be implemented

- Absolute normalisation.
  - Waiting for self-shielding corrections as they have common input properties.
- Saving output to disk.
- Plotting.
- Detector efficiency.
- Vanadium Debye-Waller correction.
- Small issues, like `Transpose` output, automatic rebinning in q...
- Memory usage optimisation (IN5).
- Documentation.
- Testing and bug fixing.

GUI



- PyQt based prototype.
- No Python bindings for `DataProcessor*` yet.
- How to handle large number of input properties?
- Reporting, property adjustment (detector diagnostics, self-shielding)?
- Plotting?

Breaking `DirectILLReduction` into smaller parts?

- Inspired by `IndirectILLReduction`.
- GUI workflow could be broken into logical steps.
  - Initial inputs.
  - Detector diagnostics.
  - Self-shielding and empty container subtraction.
  - Initial data analysis.
- Might cause confusion when used in scripts.
  - Possibility to wrong ordering or forgetting algorithm calls.
  - A single master algorithm to be used in scripts?

Tentative plans for the future

- Fix most urgent issues in `DirectILLReduction`, open a pull request.
- Start serious work on GUI.
- Implement missing functionality.