



# Introduction to ufit

– a convenient scattering data  
evaluation tool

G. Brandl, P. Cermak  
Forschungszentrum Jülich

# What is ufit?

- Started as a private collection of data readers for evaluation scripts
- Evolved to a fitting tool with GUI
- Often repeated during beamtimes:
  - read in data from 1-n files
  - apply a model (Gauß or more) and fit
- Lots of solutions from different institutes, often idiosyncratic
- ufit tries to be more universal, and cover more than “just” quick view during experiments

# Basic features

- Reads and fits mostly 1-dimensional data
- Complex models, fitting multiple datasets to one model
- Written in Python (2.7, recently ported to 3.5+ compatibility)
- Uses standard parts of scientific stack:
  - numpy, scipy
  - matplotlib
  - Qt 4 (porting to Qt 5 planned)
  - some specialized libraries for fitting
- Two interfaces: Scripting and GUI

# Two interfaces

- GUIs are nicely discoverable and easy for quick look at data
  - but horribly tedious and error-prone for reproducible treatment
- Scripting is the most reproducible way of data evaluation
  - this is especially nice with notebooks: shown in session tomorrow
- ufit provides both interfaces, with shared functionality where useful

# Sample scripting usage

```
# import all ufit API plus pylab
from ufit.lab import *

# set a template, so that data can be referenced by number only
set_datatemplate('path/to/data%04d.dat')
# read one dataset from a file, with given X and Y columns
data = read_data(104, 'A3', 'CNTS')

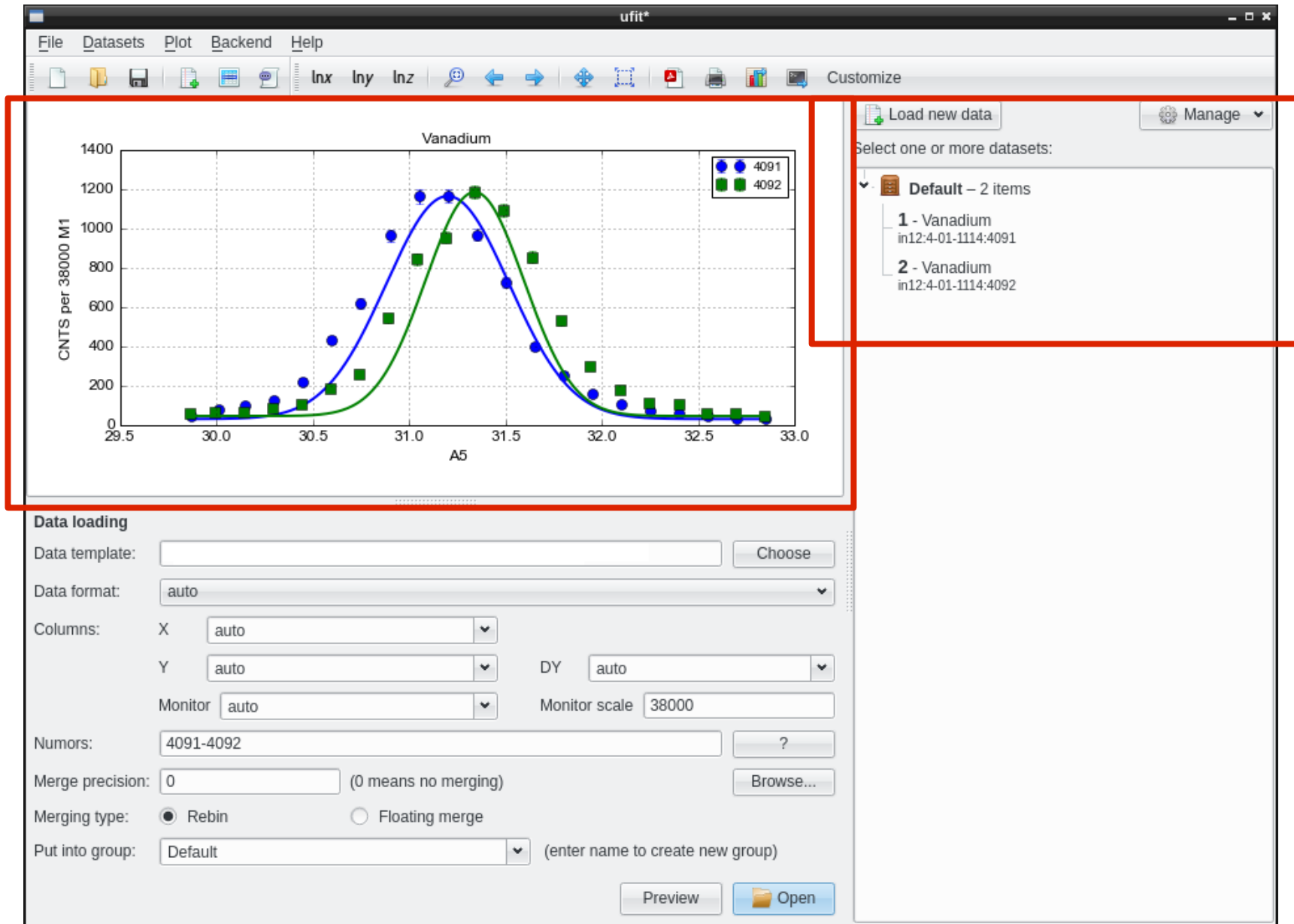
# create a model of a simple Gaussian peak with given initial guess
model = Background() + Gauss('peak', pos=97.5, ampl=100, fwhm=0.5)

# fit the model, then print and plot the result
result = model.fit(data)
result.printout()
result.plot()
show()
```

# Supported data formats

- Simple multicolumn data (space/comma separated)
- MLZ NICOS (old/new)
- MLZ TRISP
- LLB binary 3-axis data
- ILL format 3-axis data
- ANSTO TAIPAN
  
- Add new formats: needs
  - reader
  - guesser

# GUI: Main window



# Basic GUI features

- Save/load sessions (everything you see)
- Group data together into a tree
- Customize, save, print plots
- IPython console: customize plots even further!
- Export raw data, fits, fit parameters as ASCII
- Save data plus model as Python for further use with scripting interface



# GUI: Data loading

The screenshot displays the ufit software interface. The main window shows a plot titled "Vanadium" with the y-axis labeled "CNTS per 38000 M1" and the x-axis labeled "A5". Two data series are plotted: 4091 (blue circles) and 4092 (green squares). Both series show a peak around A5 = 31.2. The plot includes a legend and a grid.

The "Data loading" panel, highlighted with a red border, contains the following settings:

- Data template: [ ] Choose
- Data format: auto
- Columns: X auto, Y auto, DY auto
- Monitor: auto, Monitor scale: 38000
- Numors: 4091-4092 ?
- Merge precision: 0 (0 means no merging) Browse...
- Merging type:  Rebin  Floating merge
- Put into group: Default (enter name to create new group)

Buttons at the bottom of the panel include "Preview" and "Open".

# Data loading

- Main dataset columns: X, Y,  $\Delta Y$ , Normalization
- Relevant columns are normally autodetected
- All other columns, and scalar metadata, are also loaded and can be used later
- Mini-language for loading multiple files (“numors”), e.g. “4000>4010, 4011+4013, 4014-4017”
- Configurable merging of similar scans
- Same in scripting interface!

# GUI: Data operations

The screenshot displays the ufit software interface. The main window shows a plot of Vanadium data with a fitted curve. The plot title is "Vanadium" and the parameters are "sc a5 31.36 da5 0.15 np 21". The y-axis is labeled "CNTS per 38000 M1" and the x-axis is labeled "A5". The plot shows a peak centered around 31.36. The legend indicates the data is from "4091".

The "Data operations" panel is highlighted with a red border and contains the following controls:

- Data operations** (Modeling | Fitting)
- Set title: Vanadium (Set)
- Set short name: 4091 (Set)
- Clone to new: Clone (Change x/y values by formula: Do it)
- Merge/Rebin (destructive): 0.001 (Float merge) (Rebin)
- Add a constant: 0 (Add)
- Scale Y with constant: 1 (Scale)
- Shift X values: 0 (Shift)
- Scale X with constant: 1 (Scale)
- New monitor scale: 38000 (Change)
- Subtract dataset: Subtract...
- Mask out points: Start (Reset)
- Fit limits: [ ] to [ ] (Set)

# GUI: Model creation

The screenshot displays the ufit software interface. The main window shows a plot of Vanadium data with a fitted model. The plot title is "Vanadium" and the y-axis is labeled "CNTS per 38000 M1". The x-axis is labeled "A5". The plot shows a peak centered around 31.36. The model definition panel is highlighted with a red box and contains the following information:

**Data operations** | **Modeling** | **Fitting**

**Model definition:** Gauss peaks only mode

Background() + Gauss('peak')

**Add predefined model:**

- GaussInt
- Gauss
- LorentzInt
- Lorentz
- Voigt
- PseudoVoigt
- DHO
- Background
- SlopingBackground

(Model info)

Add Add custom model...

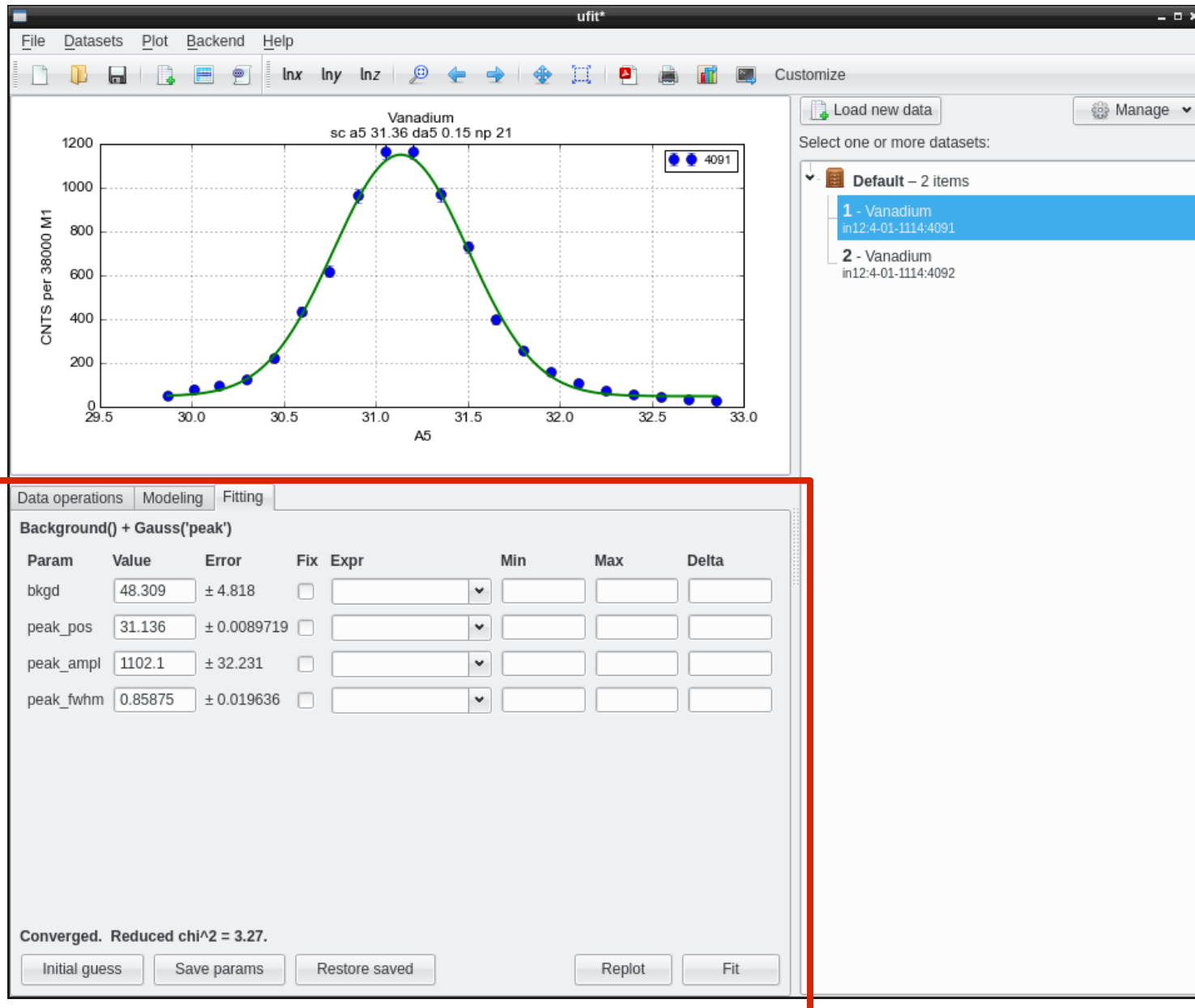
Click "Check" to check model definition.

Reset Check Apply

# Models

- Defined in Python with classes representing functions with arguments as parameters  
`Gauss(...)`, `DH0(...)`, ...
- Functions have a name that distinguishes same functions in the same model: `Gauss('p1')` + `Gauss('p2')`
- Functions can be added, multiplied, etc.
- Completely custom models can also be created referencing a Python lambda function
- Same in scripting interface!

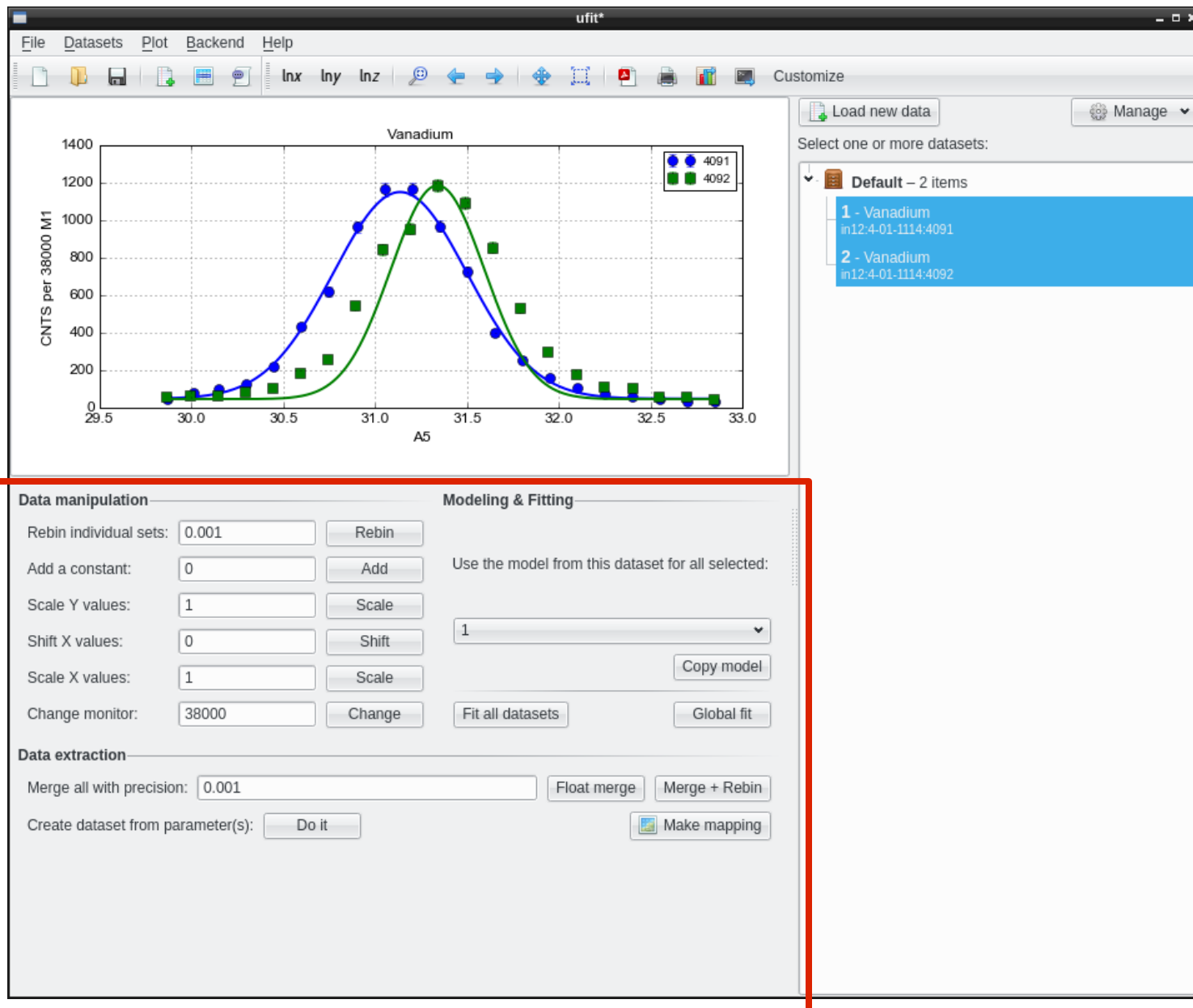
# GUI: Fitting



# Parameters

- Initial guess from model
- Many options for parameters:
  - variable
  - variable with limits
  - fixing it after fitting
  - fixed, with Python expression
  - fixed, depending on value from metadata
- Same, and more, in scripting interface!

# GUI: Multi-dataset options





# Powerful multi-dataset options

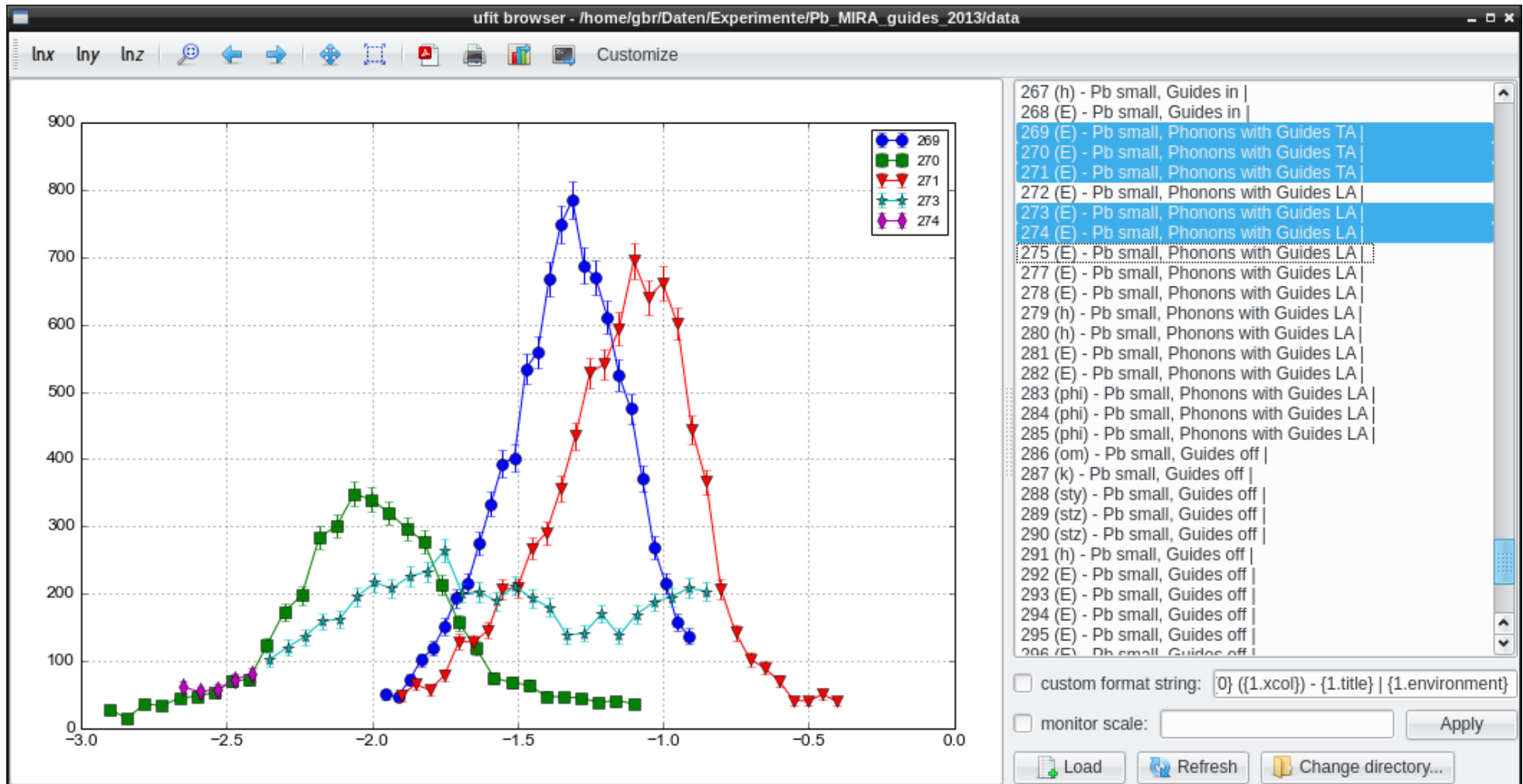
- Copy one model to all datasets
  - Fit all datasets at once
  - Create a 2-dimensional mapping
  - Create a dataset from parameters
- 
- Has to be repeated after data changes...

# GUI: Mappings

The screenshot displays the ufit software interface. The main window shows a 2D contour plot titled "mapping". The x-axis is labeled "QH" and ranges from 1.01 to 1.05. The y-axis is labeled "QK" and ranges from -0.040 to -0.015. A color bar on the right indicates values from 0 to 1800. The plot shows a central peak with concentric contours. Below the plot is a "Mapping control" panel, which is highlighted with a red border. This panel contains the following settings:

- Mapping name: mapping
- Remove masked out points
- Contour
- Draw dots
- Interpolation steps: 100
- X axis: QH
- Y axis: QK
- Y stepwidth scale: 1
- Z limits: -1e+300 to 1e+300
- Logarithmic Z scale
- 2D gauss fit
- pos\_x: 1.02764 pos\_y: -0.02780 theta: -0.13362 fwhm\_x: 0.01354 fwhm\_y: 0.00442 ampl: 1918.02117
- Apply button

# GUI: Data browser



# Data browser

- Quick selection (important during experiments!)
- See all data from a directory
- Select many datasets, load directly from there
- Custom display based on metadata within files (temperature? kf? monochromator scattering order?)

# Links

- Home:  
<http://wiki.mlz-garching.de/ufit:index>
- Documentation:  
<http://pythonhosted.org/ufit/>
- Tracker:  
<http://wiki.mlz-garching.de/ufit:support>

Thank you for your attention!