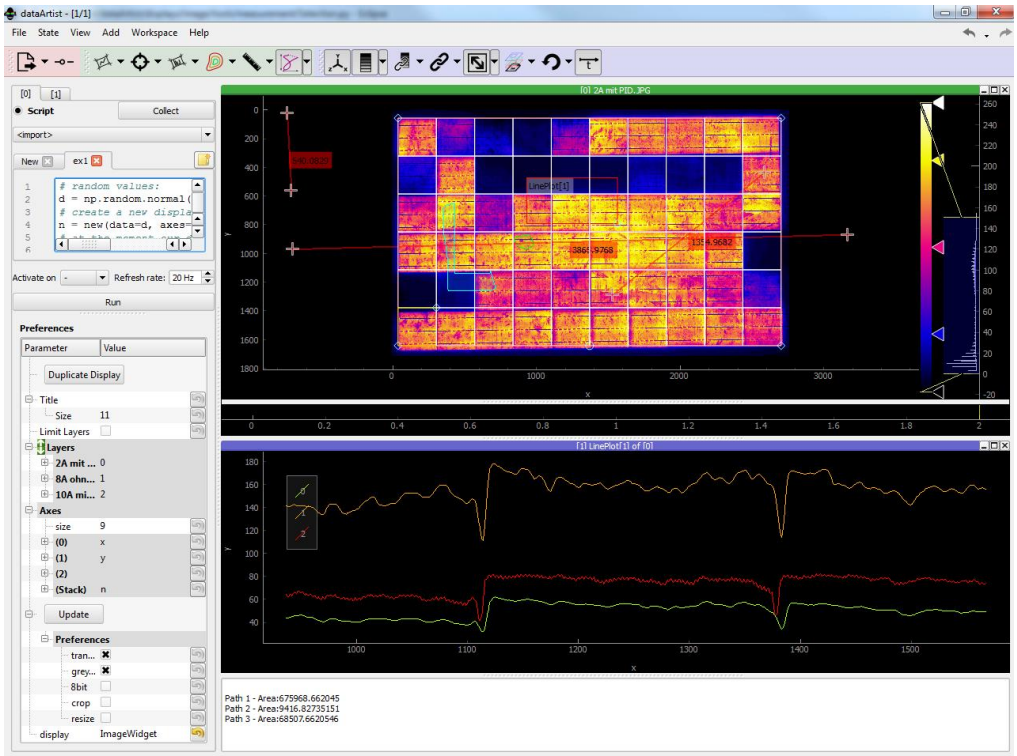
dataArtist – User manual

…scientific data processing made easy.



1 Contents

2 About

dataArtist is a graphical program for interactive data analysis and processing. It is currently specialized image processing tasks in combination with electroluminescence imaging of photovoltaic devices.

dataArtist is written in Python (2.7) and is released under open source. You are allowed to use and redistribute the code following the limitations of the GNU public license v3 (http://www.gnu.org/licenses/gpl-3.0.en.html). A brief summary of it can be found at https://tldrlegal.com/license/gnu-general-public-license-v3-(gpl-3).

dataArtist is written to be platform independent. It is known to run under Windows 7-10 and Ubuntu Linux 14.10 (soon).

3 Features

- Importing local files (as well as files from a browser) through drag and drop, see (6). Currently supported file types are:
  - Images: TIF, BMP, PNG, JPG, CSV, TXT, MAT
  - 2D Graphs: CSV, TXT
  - Numpy arrays, saved in *.npy
  - Text and tables from e.g. Excel, Word, Notepad
- Workspaces: Multiple windows of dataArtist are grouped as workspaces in one single window, see (8.3).
- Sessions: Multiple states of a dataArtist session can be saved to file. It is possible to change between saved states while dataArtist is running, see (8.1).
- Data representation in docks: Multiple data sets can be displayed in tiled or tabbed windows, see (8.4).
- Interactive built-in console: Python commands to automate dataArtist or manipulate data sets can be created and interactively executed in a built-in console, see (8.10.1).
- Layers: Multiple data sets of the same kind can be stacked as layers within a display dock.
- Tools: Commonly used processing algorithms can be accessed through tools specific for the currently active display dock, see (8.9).
- Modding and extending dataArtist: dataArtist can be easily configured for special tasks, through adding own tools, file readers and display widgets. Python is a fast and easy to learn programming language. Adding own tools can be done with only a few lines of code, see (11).
- Interaction with other programs: dataArtist can be used to as output of other programs. The communication with other programs is done through the file server RabbitMQ, see (9.4)

4 Installation

There are three different ways to install dataArtist:

4.1 Portable version

A precompiled version which has no external dependencies can be downloaded from GitHub (https://github.com/radjkarl/dataArtist - releases). There is no extra installation needed.

The portable versions are platform dependent. At the moment only a version to be known to run under Windows 7-10 is available.

dataArtist can be started by double clicking on dataArtist/dataArtist.exe

4.2 Installation into existing Python installation using pip

dataArtist can be automatically downloaded/installed/updated through pip (pip installs Python).

Preparation

You need to have Python 2.7. installed (https://www.python.org/download/releases/2.7/). Both 'python' and 'pip' should be commands recognized by your systems shell (Windows: cmd.exe). If these commands are not known see https://pip.pypa.io/en/stable/ and https://docs.python.org/2/faq/windows.html for further instructions.

dataArtist and most of its dependencies are available through PyPi (Python package index), therefore the installation can be done through typing the following command in your system console:

```
pip install dataArtist
```

A few of dataArtist's dependencies (including **numpy** and **scipy**) need to be compiled on your computer. In the Windows installation of Python 2.7 a quite common problem is, that the file 'vsvarsall.bat' is not recognized by the system. If you identify this error, you can either solve the problem (see http://stackoverflow.com/posts/27210430/revisions) or install the pre-compiled versions of all packages that show this error from this site: http://www.lfd.uci.edu/~gohlke/pythonlibs/.

The following packages are not (jet) available through PyPI and have to be installed additionally:

OpenCV 2.4

Windows: http://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv

Linux: available in many package repositories under the name 'python-opencv'.

Ubuntu/Debian: open terminal and type: 'sudo apt-get install python-opencv')

PtQt4

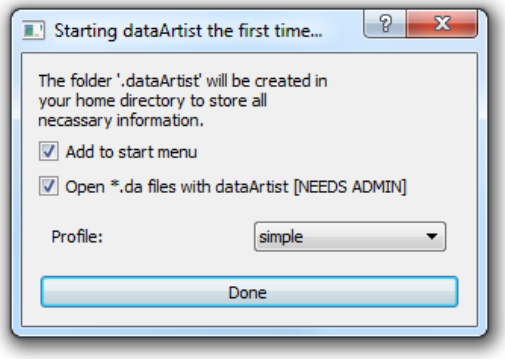
https://www.riverbankcomputing.com/software/pyqt/download

4.3 Installation into new and isolated Python installation using virtualenv

In case you don't want packaged to be installed to interact with your existing Python installation you can use virtualenv to create an isolated Python environment. See http://docs.python-guide.org/en/latest/dev/virtualenvs/ for more information.

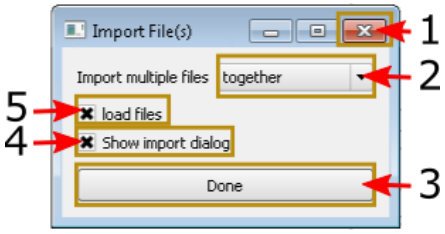
5 First start

dataArtist creates the (hidden) directory '.dataArtist' in your user/home folder. If this one doesn't exist e.g. at the first start of the program, the following window opens and asks whether and how to embed dataArtist into the operation system:



The folder '.dataArtist' contains all configuration variables that are not session dependent and the last auto-saved session. To show the first start window and to restore the initial state of the program simply delete that folder.

6 Importing data



The usual way for importing data is through drag and drop one or multiple selected files or folders into dataArtist. Alternatively the file import dialog can be opened using the menu bar: File→Import data. If not deactivated (in the preferences, 9.3) the shown window pops up. The selected files are only imported, if the done button [3] is pressed. In import can be cancelled by clicking on the close button [1]. Three options are available when importing multiple files at the same time [2]:

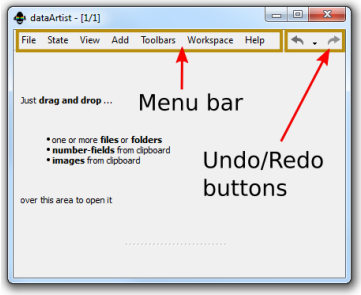
- Together
  - All files are stacked in one display. For this the files need to be of the same type and shape (for images: same resolution).
- Separated
  - Open a new display for every single file.
- In the current display
  - Add all files as new layers in the current display.
- In import display
  - Fix the current display as the general import display.

Importing many or huge files can take time or slow down the computer. Do avoid this or to change the preferences for importing files in prior it might be useful to not automatically load the files [5]. The imported files can be loaded later through clicking on the 'update' button in the display preferences (8.10.2). Some tools (8.9) also work on unloaded files. Loading all files is than not necessary. The shown import dialog will pop up every time when new files are dropped into dataArtist. To avoid this and the load the files with the last preferences, uncheck [4].

7 Program layout

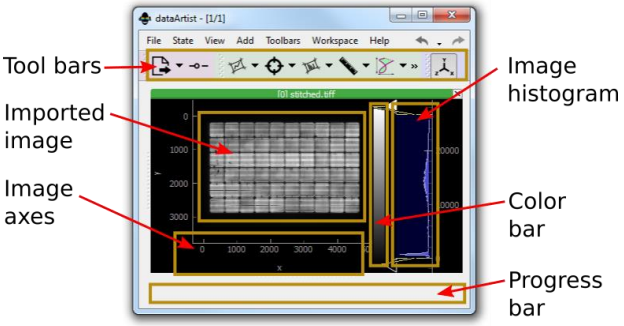
After starting dataArtist you see a practically empty window. Now you can …

- continue Importing files (6)
- adding an empty display (8.4),
- adding a notepad or table (8.5, 8.6),
- opening an existant session (8.1)



As soon a a file is imported into dataArtist you see …

- several tool bars at the top of the window (8.9)
- depending on the number of separatly shown data sets one or multiple display docks (8.4)
- your imported data, together with axes, image hisogram and colorbar (only for images)



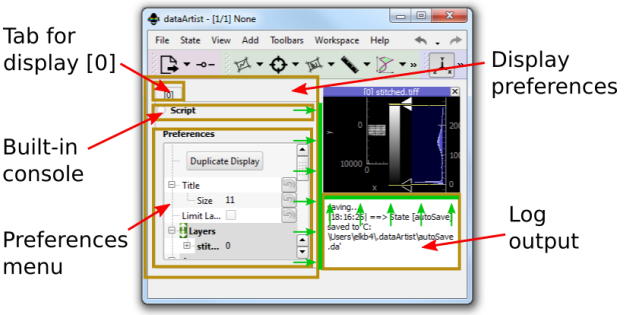
Two sliders hide the..

- display preferences (8.10) and
- built-in console (8.10.1)

one the left side

- logged output (8.12)

on the bottom of the window. To show them, click and drag the corresponding slider, highlighted in green in the image.



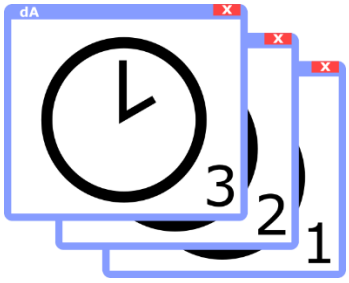
8 Program components

This section describes the specific components dataArtist is built on.

8.1 Sessions

All imported and displayed input data, visual and layer settings as well as tool parameters built up a dataArtist session which can be saved and restored using the Menu bar File→Save Session / Open Session or using the shortcuts CTRL+S or CTRL+O. The file format for dataArtist sessions is *.da. A saved session can also be opened through drag and drop the file into an opened dataArtist window or directly through double clicking on it. For the latter option "open *.da files with dataArtist" needs to be selected in the first start window (5).

8.2 States



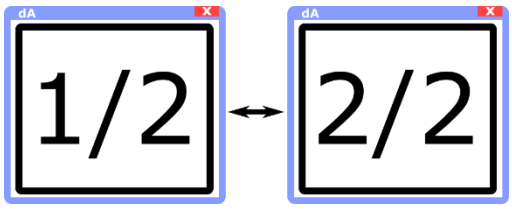
A dataArtist session can contain multiple states (resp. version, revision) which display the currently visible data and settings. The maximum number of sates to be saved in a *.da file can be chosen in the preferences menu as follows:

Menu bar→File→Preferences→Session

By default each state is named with its absolute state number. In individual name can be defined through the Menu bar: State->Rename

To fit maximum number of states new saved states override older ones. When a dataArtist session is opened the latest state is loaded. To change the currently displayed state, use the
Menu bar: State→Set

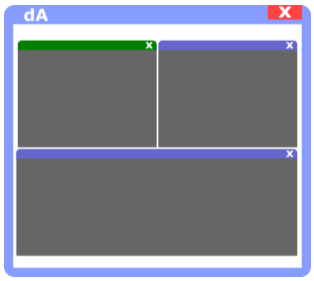
8.3 Workspaces



Especially when many different input files are shown dataArtist can look messy. Instead of opening dataArtist another time one can add another workspace. A Workspace can be compared with the desktop of your OS. If there are too many windows you go to an empty (e.g. virtual) desktop or you change between desktops specified for specific tasks. To change between dataArtist workspaces use the menu bar: Workspaces→Next/Previous or use the shortcuts Ctrl+PgUp or Ctrl+PgDown.

Workspaces can be added or removed using the menu bar: Workspace→Add/Remove or using the shortcuts: Ctrl+W or Ctrl+Q. To move a display to another workspace, use the menu bar: Docks→Move current dock to other workspace.

8.4 Display docks



Every data set loaded into dataArtist opens a new display dock. These docks are similar to windows in your OS. Multiple docks are tiled (see Wikipedia) or tabbed on top of each other. The docks position can be changed through clicking on a docks decorator (top label) and drag and drop it to its destination.

Every dock can only represent data of the same file type.

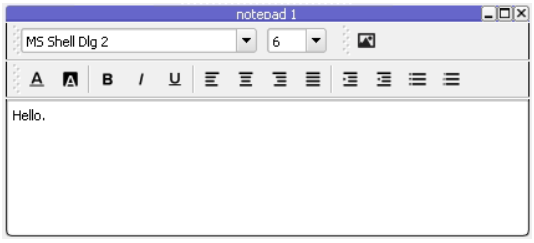
Like windows, docks can be closed, minimized and maximised using the dock decorators on the top right of each dock. To show a dock in full screen double click on its label on its top. Press ESC to exit full screen. Right click on the dock label to rename the current display dock.

The label colour of a display dock indicates whether it is selected or not. The currently selected display dock also changes the visible display tools (8.9) on the top and the display preferences (8.10) on the left of the window. A new and empty dock can be added through the menu bar: Add→Display.

Right clicking on the dock label gives the options:

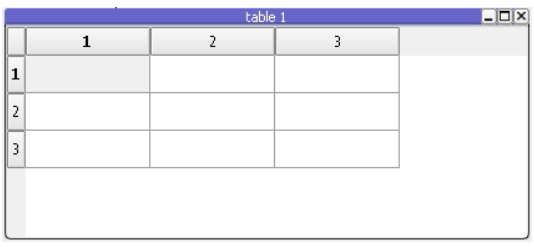
- Pop out: show the dock in an extra window in front of other windows
- Full screen: show the dock in full screen (Press Esc to go back)
- Set name: rename the dock

8.5 Text editor



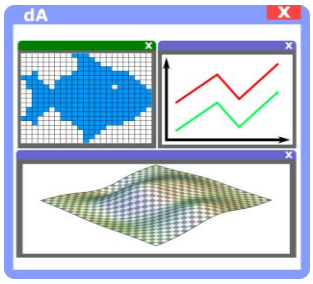
A simple text editor for making notes can be added through the menu bar: Add→Notepad. If marked text is imported into dataArtist via drag and drop a new text editor containing the marked text is added as well. A context menu for e.g. sowing/hiding the format tools or saving the text can opened through right click.

8.6 Table editor



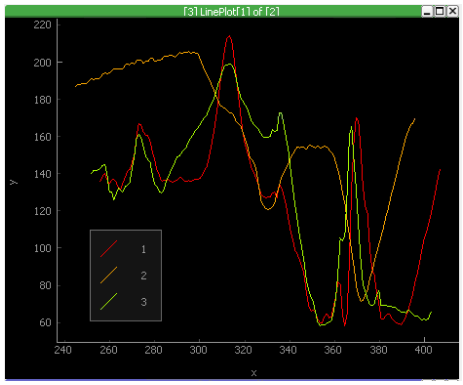
Structured data can be created/edited with the table editor through the menu bar: Add→Table. Importing marked a table from e.g. Excel will also open a new table editor containing the marked table. The table editor support normal mark-, copy-, paste features, known from common table calculation programs. A context menu for e.g. inserting new rows/columns can be accessed through right click.

8.7 Display figures



Depending on the number of dimensions of the imported data different figures can be chosen to visualise the data. At the moment 2d plots, grey and coloured images/videos can be shown. Each figure type has its own set of toolbars (8.9). The figure type can be chosen in the last parameter “figure” in the display preferences (8.10).

8.7.1 Plot



The interactive plot figure can be used to plot 2d graphs of different colours, thickness and effects. Drag gestures with clicked left mouse button can be used to change the position in x or y dimension. Through pressing on the small “A” button on the bottom left all plotted graphs can be fitted. The mouse wheel can be used to change the zoom level. The figure-specific context menu can be opened through clicking the right mouse button over the figure. This menu allows to change the viewed position, zoom, log plotting and allows to toggle the visibility of a screen grid and a down sampling of the displayed data in order to speed up the processing time.

8.7.2 Image/Video

The image/video figure is built on top of the plot figure and extends its features. On the right side a colour bar connects the displayed colour to an intensity value. The menu for changing the colour scheme can be opened through clicking with the right mouse button on the colour bar. The position of the individual colours in the colour bar can be changed through moving the colour arrows on the right side of the colour bar up and down. The colour dialog for changing the individual value of the colour arrow can be opened through double clicking on the arrow. New colours can be added through clicking with the left mouse button on empty areas in the colour bar and colours can be removed through right clicking on existing colour arrows. On the right of the colour bar a histogram of the distribution of the individual image intensities is shown. A region shows the displayed intensity range. Through moving its top or button border or the region itself the displayed range can be changed.

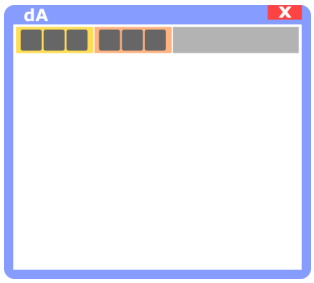
If multiple images are loaded in a figure a “time”-axis is shown on the bottom of the figure. Moving the marker line in the time-axis changes between each individual image. The following keyboard buttons can also be used to change between the individual images:

Key	Action
Space	Start/Stop switching between all images
Left arrow	Show last image
Right arrow	Show next image

8.8 Layers

dataArtist allows import display and processing of multiple datasets within a single display as long as the data shape is compatible (e.g. same resolution for images). Each data set (image or graph) acts as a layer within a stack. Layer names and position within the stack can be changed in the display preferences (8.10)

8.9 Toolbars



Each display figure has its own set of toolbars. A toolbar contains tools to change the visibility or visual representation of items in the figure or to modify the displayed data. Most of the available toolbars are hidden be default. The visibility of a toolbar can be changed using the menu bar: View→Toolbars. Displayed toolbars also can be removed through right clicking on a toolbar. Many of the tool buttons in a toolbar have preferences accessible through clicking on the down arrow on the right of the button.

8.9.1 Plot

The following table includes commonly used tools of the plot figure sorted in tool bars.

Bar	Icon	Name	Description
View		Axes	Show/Hide the x and y axes.
		Colour	Change the colour of each layer.
		Error bars	Add values from another layer to a given layer as error bar.
		Legend	Show/hide legend containing the layer names. Also: change number of columns in legend
		Line style	Change layer line style
		Line thickness	Change line thickness.

		Link view	Link the visible range of x and y axis to the one of another display.
		Symbol colour	Change the colour of a layer symbol.
		Symbol shape	Change the symbol shape of a layer.
General		Fit to function	Add a functional fit of all layers as new layer. This tool is in development.
		Export to table	Add new table editor showing all values of the figure.
Processing		Average	Add a layer containing an average of all existing layers.
		Average range	Average all layers in x direction. Optional average only a given range.
		Moving average	Add a new layer containing a moving average (smoothing) for each existing layer.
		Normalize	Scale the y values of all layers between 0 and 1
		Stitch plots	Add values from another layer to a given layer
		Remove outliers	Remove outlying values through limiting the maximum ascent.

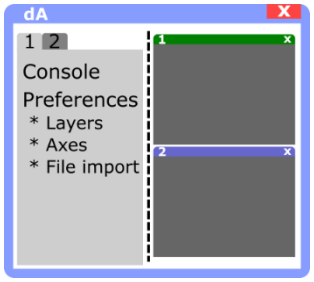
8.9.2 Image/Video

The following table includes commonly used tools of the image figure sorted in tool bars.

	Icon	Name	Description
General		Export image(s)	Export the shown image (rendered) of the display values to file.
		Sequester	If new displays are created from existing one the shown data only link to its origin data. Click this tool to make the data independent from its origin.
Measurement		Region average	Create a new display containing the average a region of interest (RIO) in the current display. Values can be averaged in x and/or y dimension. If no dimension is average the new display contains the sniped of its origin.
		Crosshair	View intensity value of an image. Multiple values can be shown at the same time.
		Region histogram	Show the image intensity histogram of an ROI.
		Ruler	Measure lengths in the image of scale the x or y dimension.
		Selection	Select an area in this image to highlighting or measuring area or position. Different shapes (e.g. ellipse, freehand) are available.
		Signal-to-noise ratio	Calculate the averaged or spatially resolved signal-to-noise ratio (SNR) using the method defined in ### or the IEC TS 60904-13.
View		Axes	Show/hide the x and y axis
		Colour bar	Show/hide the image colour bar and histogram. Also: enable histogram print view.

		Colour←→Grey	Indicated whether an image is colour or grey and transform between both.
		Link colour bar	Link the colour bar in visible intensity range of this display to another one.
		Link view	Link the visible area of this display to another one.
		Lock aspect ratio	Lock/Unlock the aspect ratio of the image.
		Overlays	Manage single-coloured overlays. These overlays indicate e.g. changes and can be added be other tools.
		Rotate	Rotate the image 90 DEG or transpose x and y axis.
		Time axis	Show/hide the time axis.

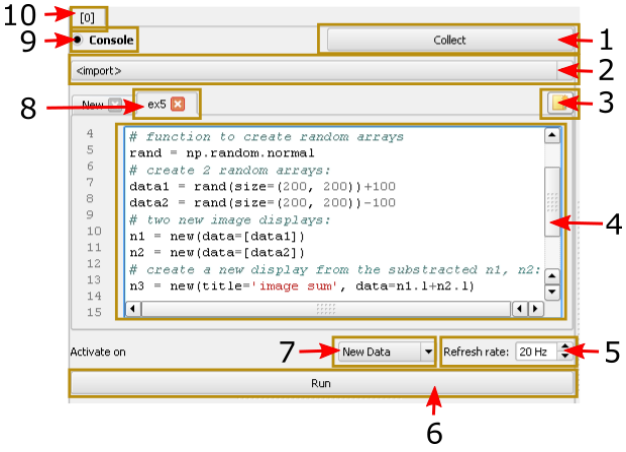
8.10 Display console and preferences



The display preferences of each display are hidden by default but can be shown if the slider on the left edge if moved towards the middle of the window. The preferences of all displays are represented as tabs. The tab of the current display is made visible. The display preferences contains…

- a built-in Python console for process automation and direct access to the contained data
- preferences to modify axes names and scaling
- preferences and information to all individual data layers of the display

8.10.1 Built-in console

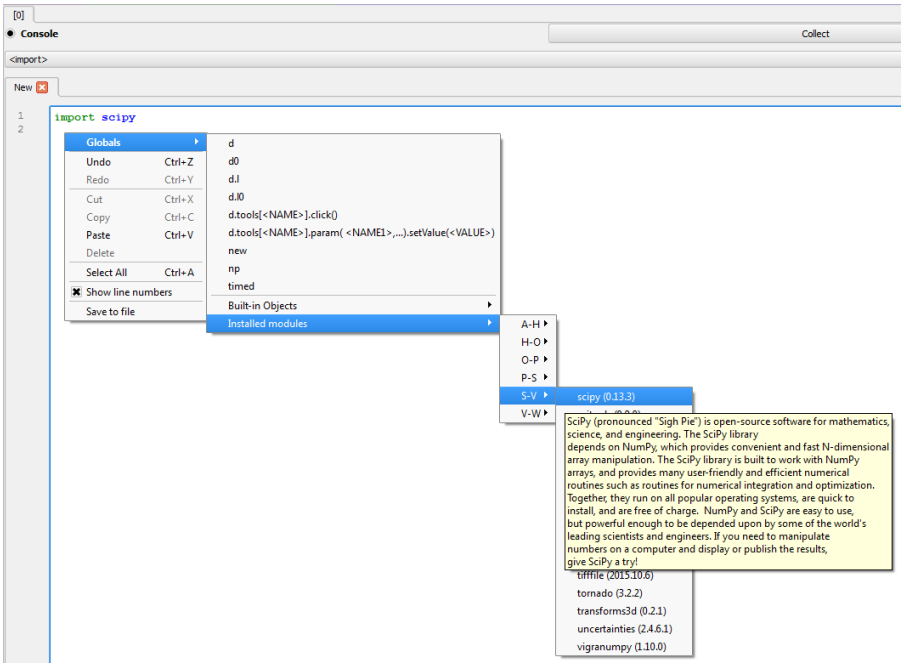


Individual routines and process automation can be set up though a built-in Python console, which is shown after checking [9]. The created scripts [4] can be triggered manually or after changed or added input data [7]. While running a script the current display figure is updated with a fixed refresh rate [5]. Extensive calculations can be accelerated with decreased refresh rate. To access the tools of the current display click on 'Collect' [1] and then on the specific tool or one of its parameters. The tool is than added to the opened console. Press 'Collect' again to deactivate the collection procedure. A range to examples can be opened through [2]. Scripts, saved to file, can be accessed the same way. When pressing 'Run' [6] the currently opened script is executed. New scripts can be opened through clicking on [3]. If a script is running a red indicator is shown on the top right of the current display:  .

The Python 2.7 syntax is used. The following modules are can be accessed additionally to the Python standard library:

Command	Module / Origin
`np`	Numpy (various tools for numeric calculations including arrays).
`d`	Display of the current console.
`d0 … d[NUMBER]`	Display of a specific display.
`d.l`	Access to all data layers of a display. The data is usually represented as numpy array.
`d.l0 … d.l[NUMBER]`	Access to a specific data layer.
`d.tools[<NAME>].click()`	Access a specific tool and activate it.
`d.tools[<NAME>].param(<NAME1>,…).setValue(<VALUE>)`	Access to a specific parameter on a tool and set its value.

All locally available objects, as well as installed modules can be easily added to the console through the right click menu:



The following examples demonstrate the implementation of common image processing problems within the built-in console:

```
#subtract layer 2 from layer 1 in the
#current display:
d.l1 -= d.l2

#set layer 1 on display 4 to the sum of
#layer 3 and 4 in the current display:
d4.l1 = d.l3 + d.l4

#average all layers of the current display
d.l = d.l.mean(axis=0)

#print the average of an ROI of the size of
#10x10 pixels at position (100,200) of
```

```
#layer 3:
print d.l3[100:110,200:210].mean()

#import and execute a filter from the
#library 'scipy':
from scipy.ndimage.filters import median_filter
d.l = median_filter(d.l, 3)
```

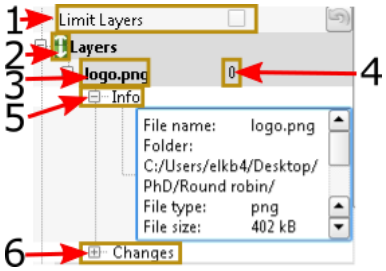
One dimensional plots e.g. created from a line cut can be manipulated correspondingly:

```
#add the y-component of two plots
d.l1 = d2.l0 + d4.l2

#print all x-values of all plots within
#the current display:
print d.l.x

#divide all plots in the current
#display by their average:
avg = d.l.y.mean()
d.l /= avg
```

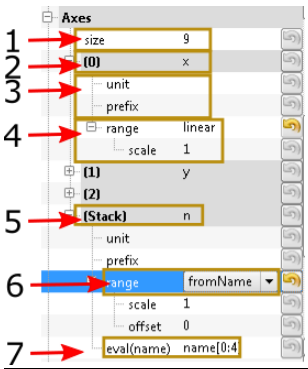
8.10.2 Layer preferences



Multiple data sets of the same shape/type (plots, images) can be stacked in one display. Each set is than represented as layer. The number of layers in a display can be limited [1]. If enabled new imported data sets cause deletion of the first layers in the stack (FIFO principle). Each layer is represented by a name [3] and a stack position [4]. Right clicking on the layer name [3] opens a context menu. The context menu features are shown in the table below. The position within the stack of the current selected layer can be changed through clicking on the up/down button [2]. The subordinate parameters 'Info' [5] and 'Changes' [6] contain respective information about the input file (e.g. file size) and changes done on this layer by a dataArtist tool.

context menu entry	Description
Rename	… selected layer
Remove	… selected layer
Copy	… selected layer to new/other display
Move	… selected layer to new/other display

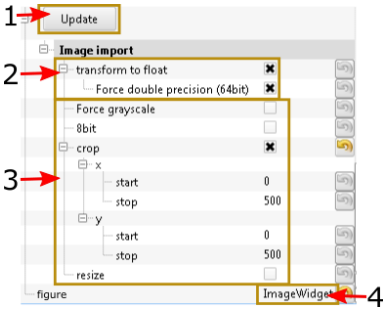
8.10.3 Axes preferences



All dimensions of the imported data set can be found and modified here. Exemplary for images have 3 dimensions: 'x' (width), 'y' (height) and a dimension ordering all image layers in the display stack 'n'. Axes names [2] and size [1] can be modified. Each axis can have a unit (e.g. 'm' for meters) and a SI-prefix (e.g. 'c' for 'centi…meter') [3]. For images the axis unit is in pixels. Therefore the image ends at its size (e.g. x=800, y=600). These values can be scaled with a factor [4]. The value range [6] of stack axis [5] additionally can be set to the following options:

Range	Description
Linear	Linear range defined by offset and scale
Individual	The stack value of each layer can be set individually, through direct change (see image marker [4] in 8.10.2
fromName	Stack values are read from the filename. Example: multiple images include their exposure time in their file name (e.g. 30 in 'img1_30sec.jpg'). To extract this value type 'name[5:6]]'. In background dataArtist takes this entry and uses the result gained from the Python function 'eval(filename)'.

8.10.4 File import preferences

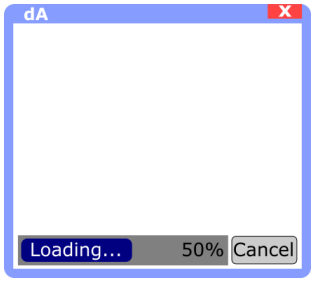


At the bottom of the display preferences the file import options can be found. Pressing the button 'Update' [1] causes reading all files represented by the display layers and updates the display figure. The import options can be different for the different file types. The image on the lefts shows exemplary preferences for the import of images.

The options marked by [2] control the transformation of the file data type to floating point precision (double). This option is usually preferable when the image is to be modified because images are originally stored as unsigned integers, rejecting negative values or values higher than e.g. 255. Other preferences organize the whether to load the image as greyscale, or to crop or rescale it. The latter two options are useful if only a small part or multiple images is of interest and loading all images would take too much memory.

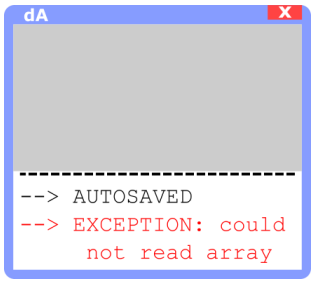
The last list parameter 'Figure' [4] allows to select a specific figure for the imported file. At the moment only one figure type if supported for each plots and images.

8.11 Progress bar



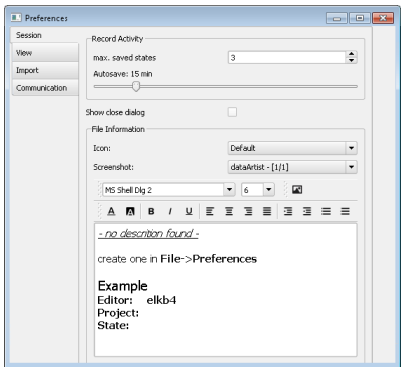
The progress of the importing data or tool algorithms is shown in the process bar at the bottom of the window. These processes can be aborted through clicking on the 'cancel' button. If the process bar is not visible the last printed output is shown for a limited time. Older prints can be viewed in the system log (8.12).

8.12 System log



The system log is hidden by default but can be shown through moving the horizontal slider at the bottom of the window upwards. The last 300 lines of system output are shown in the system log. Errors or exceptions are highlighted in red.

9 dataArtist preferences



dataArtist's preferences can be found in the menu bar: File→Preferences. There different sections can be chosen:

9.1 Session

Change the number of saves states (8.2) and the auto save frequency. The section "file information" is not fully implemented at the moment.

9.2 View

Choose a colour scheme, enable/disable antialiasing and the standard toolbar profile. By default background colour of display figures is black and label colour is white and antialiasing of switched off. Currently one two test toolbar profiles are enabled: 'simple' and 'electroluminescence'. The latter one shows more toolbars for image figures by default.

9.3 Import

Change all features described in Section 6.

9.4 Communication

9.4.1 Watch folder

Imports all newly created files in a chosen folder.

9.4.2 Using RabbitMQ

Some functions of dataArtist can be triggered by other programs, even from other computers. For this RabbitMQ, a lightweight message server is used. To enable inter-program communication RabbitMQ needs to be installed and a message server needs to be initialised, see https://www.rabbitmq.com/tutorials/tutorial-one-python.html.

To receive messages to field 'allow inter-program communication…' needs to be checked. dataArtist than listens to the following messages:

Message	Example	Description
addFile [FILE PATH]	`addFile c:/User/admin/img.png`	Import a new file given by its path.

showDisplay [DISPLAY NUMBER], [AREA in (x,y,width,height)]	`showDisplay 1 (20,40,600,400)` OR `showDisplay 1 False`	Show/hide a display in a frameless window. To show a display specify AREA
changeActiveDisplay [DISPLAY NUMBER]	`changeActiveDisplay 2`	Change the current/active display given by its number.
runScriptFromName [NAME]	`runScriptFromName script1`	Execute a console script of the current display given by its name.

10 Interactive tutorials

dataArtist embeds the option of running and creating interactive tutorials within a session. This option can be found in the menu bar: Help→Tutorial. The user manual for this feature can be found on the top right edge of the tutorial creation window (Help→Tutorial→Create/Edit).

11 Modifying dataArtist

dataArtist allows easy and fast specified for individual problems. Examples showing dataArtist starting under a different name with added individual display figure, file reader and figure tool can be found in the source code under /dataArtist/modding. Further reference can be found in the dataArtist API. Don't forget: dataArtists license forces all derived projects to be preleased under open source as well (if the code is to be distributed).

12 Known bugs and issues

dataArtist is still a young project. Bugs are common and the program might crash. Feature requests and bugs can be found and added in the GitHub project: https://github.com/radjkarl/dataArtist/issues.