INSIGHT Cheatsheet

Manuel Reus, Lennart Reb

This cheat sheet included the most used commands for INSIGHT.
Most functions have attributes that you can specify if you wish. Arguments you must provide are given in the tables below in many cases.
Find out more with 'Ctrl+B' or in the documentation.
INSIGHT is assumed to be imported as `ins`; the giwaxs_sim moduel as `ins.gs`.

ins.SingleImage(ReshapeParams, raw_image_path)	load one GIXS image
plot_raw_Image()	plots the raw image
rot90(), fliplr(), flipud(), transpose()	re-orient raw image
calculate_geometry()	reshapes image
calculate_corrected_intensity()	corrects image intensity
plot_reshaped_image()	plot the reshaped image
plot_reshaped_chiq()	plot reshaped image (chi vs q representation)
delete_hot_pixels()	delete hot pixels, identify outlier intensities and delete them
create_cut_waxs(cut_name, q_min, q_max, chi_min, chi_max)	make a WAXS cut and specify the cut_name and cut limits
create_cut_saxs(cut_name, q_min, q_max, chi_min, chi_max)	make a SAXS cut with cut_name and cut limits
list_cuts()	list all cuts
plot_local_bg_annuli(cut_name)	plot the 2 annuli from local background subtraction of the cut cut_name
optimize_sdd(cut_name, q_ref)	optimize the SDD by providing a reference q-value for one Bragg reflex position and a cut that contains this Bragg peak
save()	saves maps from GeometryCalculations
apply_gap_mask(gap_array), apply_pixel_mask(pixel_array), apply_flatfield(flatfield_array)	apply loaded masks and flatfield, load with staticfunction load_aux_file()
plot_gap_mask(), plot_pixel_mask(), plot_flatfield()	plot masks and flatfield
denoise()	denoise the image with Gaussian or median filter (be careful!)
save_cut()	save to file

ins.Params()	load Params
print_params()	print out the current parameter set to the console
get_dict()	returns a dictionary with the current parameter set
set_*(val)	set the specified parameter to val, * means one of the essential parameter names, e.g. *='sdd' or 'wl' or 'db_x'
get_essential_params_keys()	get a list of all possible parameters by
get_single_dict(n)	for batch processing, returns a dictionary containing the parameters for the nth image
fill()	fill up the parameters (for batch processing)
plot_cut_outline_waxs(), plot_cut_outline_saxs()	plot the cut outline into the reshaped image

ins.CUTWAXS(), ins.CUTSAXS()	create cut with SingleImage.create_cut_*() call the cut with SingleImage.CUT
bin_cut(n)	bin the cut to n bins
plot_raw(along), plot_grained()	plot raw/grained data (might take a long time), specify if you want to plot along 'q' or 'chi'
plot_binned(along), plot_bg_corrected(along)	plot binned/bg_corrected data, specify if you want to plot along 'q' or 'chi'
optimize_tube_cut_q_limits(sigmas)	optimize the cut limits for your tube cut, specify width in sigmas
subtract_local_background(sigmas, annuli_width, n)	subtract local background from your cut by making 2 additional cuts (inner and outer cut), spedify distance in sigmas and annuli_width in sigmas and bin number n

`ins.GeometryCalculations()`	reshaping the image, results from reshaping are stored in ins.SingleImage.Geo
`plot_geometry_3d(x, y, z, sdd, im)`	plot a 3d image to check detector geometry, specify x, y, z coordinates, sdd, and image array)
`plot_geo_map(map_key)`	plot map specified by map_key
`plot_geo_map_all()`	plot all calculated maps

`ins.IntensityCorrections()`	corrects intensities, results are stored in ins.SingleImage.IntensityCorrections
`plot_corr_map(map_key)`	plot map specified by map_key
`plot_corr_map_all()`	plot all maps

`ins.NexusConverter(input_path, output_path)`	use this class to convert nxs to cbf, specify the path to one nxs file and and output path
`conversion2cbf()`	save all images as cbf

`ins.gs` `//giwaxs_sim module`	giwaxs_sim module for GIWAXS indexing and simulation
`ins.gs.cubic.SG221(a)`	eg, calling cubic crystal structure (space group 221)
`ins.gs.SimParams()`	load parameters
`ins.gs.calc_q_data(simparams)`	calc q-data (can be plotted in `plot_reshaped_image()`)
`ins.gs.simulate_giwaxs(simparams)`	simulate GIWAXS data in q-space

`ins.staticfunctions`	auxiliary functions with general purpose (available directly in ins)
`switch_energy_wavelength()`	switch from Angstrom to keV (or other way)
`q_to_2theta(), twotheta_to_q()`	inv Angstrom to °2theta (or other way)
`q_to_d(), d_to_q()`	inv Angstrom to Angstrom (or other way)
`twotheta_to_d()`	°2theta to Angstrom
`calc_spec_beam_pos()`	calculate the specular beam position
`correct_sdd()`	calculate new SDD
`smooth()`	smooth data
`sum_images()`	sums up images
`calculate_detectormask()`	calculate detector gap mask
`scherrer_domain_size()`	calculate lower limit crystal domain size
`load_aux_file()`	load tif file from fiel for masks and flatfield
`lmfit_simple_fit()`	sets up easy fitting routine, takes lmfit models as input
`criticalAngle()`	calculates critical angle from SLD and wavelength
`import_tubeCut(), import_cakeCut()`	import cuts (eg for 2D plotting)
`make_CutArray()`	imports multiple cake or tube cuts from folder and returns values for easy 2D plotting (see demo_2Dplot.py)