

Python commands and attributes in BrainVoyager 22

UNOFFICIAL DOCUMENT

```
# get function names for BrainVoyager object
help(brainvoyager)

# get function names for BrainVoyager document
import PythonQt
help(PythonQt.private.DocPyBV)
help(PythonQt.private.MeshScenePyBV)

# also, one can use:
doc.methods() → tuple
```

Latest update: 25-11-21

Until BV 22 - same API as for JavaScript Python commands from BV 22

BrainVoyager	brainvoyager
BrowseDirectory()	choose_directory
TimeOutMessageBox(msg, dur)	show_timeout_message_box
ActiveDocument	active_document
ActiveDocument	adoc
AnonymizeDicomFilesInDirectory(path, new patient name)	anonymize_dicoms
BuildNumber	build_number
	cdir
BrowseFile()	choose_file
CreateDocumentAMR()	X.create_amr(scanner_file_type, first_file, n_slices, big_endian, slice_rows, slice_cols, byte_per_pixel)
CreateProjectDMR()	X.create_dmr(scanner_file_type, first_file, n_volumes, skip_n_volumes, first_volume_amr, n_slices, fmr_dwi_filename, big_endian, slice_rows, slice_cols, byte_per_pixel, target_folder)
CreateProjectFMR()	X.create_fmr(scanner_file_type, first_file, n_volumes, skip_n_volumes, first_volume_amr, n_slices, fmr_stc_filename, big_endian, slice_rows, slice_cols, byte_per_pixel, target_folder) X.create_fmr_dicom(file_of_series, fmr_stc_filename, target_folder, protocol_file) X.create_fmr_dicom_nifti_bids(file_of_series, subj_id, ses_id, run_id, task_name, project_folder, protocol_file)

CreateDocumentFMRslicesTimeLooping()	X.create_fmr_slices_time_looping(scanner_file_type, first_file, n_volumes, skip_n_volumes, first_volume_amr, n_slices, fmr_stc_filename, big_endian, slice_rows, slice_cols, byte_per_pixel, target_folder)
CreateProjectMosaicDMR()	X.create_mosaic_dmr(first_file, n_volumes, skip_n_volumes, first_volume_amr, n_slices, fmr_dwi_filename, big_endian, mosaic_rows, mosaic_cols, slice_rows, slice_cols, byte_per_pixel, target_folder)
CreateProjectMosaicFMR()	X.create_mosaic_fmr(first_file, n_volumes, skip_n_volumes, first_volume_amr, n_slices, fmr_stc_filename, big_endian, mosaic_rows, mosaic_cols, slice_rows, slice_cols, byte_per_pixel, target_folder)
CreateProjectVMR()	X.create_vmr(scanner_file_type, first_file, n_slices, big_endian, slice_rows, slice_cols, byte_per_pixel) X.create_vmr_dicom(file_of_series) X.create_vmr_dicom_nifti_bids(file_of_series, subj_id, ses_id, project_folder)
CurrentDirectory	current_directory
PathToSampleData	data_path
	X.describe_method(methodName)
	docs (DocListPyBV)
	documents (DocListPyBV)
	X.dynamicPropertyNames()
	X.exit()
	X.findChild(type, name)
	X.findChildren(type, name) -> tuple
	X.get_vtcs_of_mdm(mdm_file) -> tuple
	X.hide_log_pane()
	X.isWidgetType()
	X.isWindowType()
	is_64_bits
	X.killTimer(id)
	X.methods() -> tuple
LinkStimulationProtocol()	link_protocol
	X.moveToThread(thread)
MoveWindow()	X.move_window(new_x, new_y)
	notebook_path
	objectName
	objectNameChanged
	X.open(file_path, close_current_doc, remove_current_doc)
	open
OpenDocument()	X.open_document(file_path)
PrintToLog()	X.print_to_log(text)
	projects
	X.property(name) -> object
	X.removeEventFilter(obj)

RenameDicomFilesInDirectory()	X.rename_dicoms(folder)
	X.resize_window(new_width, new_height)
ShowLogTab()	sampledata_path
MessageBox()	show_log_pane
TimeOutMessageBox	X.show_message_box(message)
	X.show_timeout_message_box(message, duration) -> bool
	X.startTimer(interval, timerType) -> int
VersionMajor	X.tr(text, ambig, n) -> str
VersionMinor	version_major
	version_minor
	version_patch
	voi_file_name
	wnd_h
	wnd_visible
	wnd_w
	wnd_x
	wnd_y

Until BV 22 - same API as for JavaScript

AddCondition()
AddContrast()
AddInterval()

AddMesh()
AddPredictor()
AddROIContrast()

AddStudyAndDesignMatrix()
AdjustMeanIntensity()
ApplyHemodynamicResponseFunctionToPredictor()
AutoACPCAndTALTransformation()
AutoTransformToIsoVoxel()

AutoTransformToSAG()

ClearContrasts()
ClearMultiStudyGLMDefinition()
ClearStimulationProtocol()
ClearDesignMatrix()
Close()
ComputeMultiStudyGLM()
ComputeRFXGLM()
ComputeSingleStudyGLM()
ComputeSingleStudyGLMForVOI()

CoregisterFMRTtoVMR()

CoregisterFMRTtoVMRUsingBBR()

CorrectIntensityInhomogeneities()

CorrectMotion()
CorrectMotionEx()

Python commands from BV 22

<doc.functionname(arguments) -> returnvalue>

X.add_condition(condition_name)
X.add_contrast(contrast_name)
X.add_interval(condition_name, interval_from, interval_to)

X.add_mesh(mesh_file) -> bool
X.add_predictor(predictor_name)
X.add_roi_contrast(contrast_by_name, contrast_string)
X.add_run_designmatrix(sdm_file) -> bool

X.add_study_and_dm(func_file, dm_file)
X.adjust_mean_intensity() -> bool
X.apply_hrf_to_predictor(predictor_name)

X.auto_acpc_tal_transformation() -> bool
X.auto_transform_to_isovoxel(interpolation, vmr_iso_filename) -> bool
transform_to_std_isovoxel(int interpolation_method, str out_vmr_iso_filename) -> bool
transform_to_iso_voxel(float target_res, int framing_cube_dim, int interpolation_method, str out_vmr_iso_filename) -> bool

X.auto_transform_to_sag(vmr_sag_filename) -> bool
transform_to_std_sag(str out_vmr_sag_filename) -> bool

X.clear_contrasts()
X.clear_multistudy_glm_definition()
X.clear_protocol()
X.clear_run_designmatrix()

X.close()
X.compute_multistudy_glm()
X.compute_rfx_glm()
X.compute_run_glm()

X.compute_run_glm_for_voi(voi_idx, normalize_tc, serial_correlation_ar_model)

X.coregister_fmr_to_vmr(fmr_file, use_attached_amr) -> bool

X.coregister_fmr_to_vmr_using_bbr(fmr_file) -> bool

X.correct_intensity_inhomogeneities() -> bool

X.correct_intensity_inhomogeneities_ext(include_brain_extraction, n_cycles, tissue_range_thresh, intensity_thresh, fit_polynom_order) -> bool

X.correct_motion() -> bool

X.correct_motion_ext(target_vol_idx, interpolation_method, full_data, max_iterations,

CorrectMotionTargetVolumeInOtherRun()	create_movie, extended_log_file) -> bool X.correct_motion_to_run(target_fmr_file, target_vol_idx) -> bool
CorrectMotionTargetVolumeInOtherRunEx()	X.correct_motion_to_run_ext(target_fmr_file, target_vol_idx, interpolation_method, full_data, max_iterations, create_movie, extended_log_file) -> bool
CorrectSliceTiming()	X.correct_motion_to_vol(target_vol_idx) -> bool X.correct_slicetiming(slice_order_scheme, interpolation_method, multiband_factor) -> bool
CorrectSliceTimingWithSliceOrder()	X.correct_slicetiming_using_sliceorder(slice_list, interpolation_method, multiband_factor) -> bool
CorrectSliceTimingUsingTimeTable()	X.correct_slicetiming_using_timingtable(interpolation_method) -> bool
GetMeshScene()	X.create_mesh_scene() -> PythonQt.private.MeshScenePyBV
CreateMTCFromVTC()	X.create_mtc_from_vtc(sample_along_normal_from, sample_along_normal_to, output_mtc_file) -> bool X.create_run_designmatrix_from_protocol(skip_condition, apply_hrf) -> bool
CreateVDWinACPCSpace()	X.create_vdw_in_acpc_space(dmr_file, coreg_ia_trf_file, coreg_fa_trf_file, acpc_trf_file, vdw_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool X.create_vdw_in_mni_space(dmr_file, coreg_ia_trf_file, coreg_fa_trf_file, mni_trf_file, vdw_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool
CreateVDWInVMRSpace()	X.create_vdw_in_native_space(dmr_file, coreg_ia_trf_file, coreg_fa_trf_file, vdw_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool
CreateVDWInTALSpace()	X.create_vdw_in_tal_space(dmr_file, coreg_ia_trf_file, coreg_fa_trf_file, acpc_trf_file, tal_file, vdw_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool
CreateVTCInACPCSpace()	X.create_vtc_in_acpc_space(fmr_file, coreg_ia_trf_file, coreg_fa_trf_file, acpc_trf_file, vtc_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool
CreateVTCInMNISpace()	X.create_vtc_in_mni_space(fmr_file, coreg_ia_trf_file, coreg_fa_trf_file, mni_trf_file, vtc_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool

CreateVTCInVMRSpace()	bool X.create_vtc_in_native_space(fmr_file, coreg_ia_trf_file, coreg_fa_trf_file, vtc_file, res_to_anat, interpolation_method, bounding_box_intensity_threshold, data_type) -> bool
CreateVTCInTALSpace()	bool X.create_vtc_in_tal_space(fmr_file, coreg_ia_trf_file, coreg_fa_trf_file, acpc_trf_file, tal_file, vtc_file, data_type, interpolation_method, bounding_box_intensity_threshold, res_to_anat) -> bool
CurrentMesh	current_mesh = MeshPyBV X.describe_method(methodName) -> str X.deselect_voi(voi_idx) dim_x dim_y dim_z file_name
TemporalHighPassFilter()	X.filter_temporal_highpass_fft(highpass, highpass_unit) -> bool
TemporalHighPassFilterGLMDCT()	X.filter_temporal_highpass_glm_dct(n_basis_functions) -> bool
TemporalHighPassFilterGLMFourier()	X.filter_temporal_highpass_glm_fourier(n_cycles) -> bool
findChild()	X.findChild(type, name) -> PythonQt.private.QObject X.findChildren(type, name) -> tuple
FirstConfoundPredictorOfSDM	first_confound_predictor_sdm
GapThickness	gap_thickness
GetNameOfROIContrast	X.get_beta_name_of_roi_glm(beta_idx) -> str
?	
GetBetaValueOfROIGLM()	X.get_beta_value_of_roi_glm(beta_idx) -> float X.get_fmr_data_as_float64_list() -> tuple
GetNameOfROIContrast()	X.get_name_of_roi_contrast(contrast_idx) -> str
GetNameOfVOI()	X.get_name_of_voi(voi_idx) -> str
GetPValueOfROIContrast	X.get_p_value_of_roi_contrast(contrast_idx) -> float
GetTValueOfROIContrast	X.get_t_value_of_roi_contrast(contrast_idx) -> float X.get_volume_data_as_byte_buffer() -> PythonQt.QtCore.QByteArray X.get_volume_data_as_float64_list() -> tuple X.get_volume_data_as_int_list() -> tuple
GetVoxelIntensity(x, y, z)	X.get_voxel_intensity(x, y, z) -> int
HideAllVOIs()	has_slice_time_table X.hide_all_vois() X.hide_maps_dialog()
InterSliceTime	inter_slice_time
LinkAMR()	X.link_amr(amr_file) -> bool

LinkStimulationProtocol()	X.link_protocol(protocol_file) -> bool
LinkVTC()	X.link_vtc(vtc_file) -> bool
LoadGLM()	X.load_glm(glm_file) -> bool
LoadVolumeMaps()	X.load_maps(vmp_file) -> bool
LoadMesh()	X.load_mesh(mesh_file) -> bool
LoadMultiStudyGLMDefinitionFile()	X.load_multistudy_glm_definition_file(mdm_file) -> bool
LoadSingleStudyGLMDesignMatrix()	X.load_run_designmatrix(sdm_file) -> bool
LoadVOIFile()	X.load_vois(voi_file) -> bool X.map_name(idx) -> str mesh_scene = MeshScenePyBV X.methods() -> tuple n_maps
NrOfPredictorsInSingleStudyDM	n_predictors_sdm n_roi_contrasts n_skipped_volumes n_slices
NrOfTimePointsInSingleStudyDM	n_timepoints_sdm n_vois n_volumes
NormalizeToMNIspace()	X.normalize_to_mni_colin() -> bool X.normalize_to_mni_space() -> bool X.normalize_to_mni_tal() -> bool
<i>new in BV 22 (only in Python)</i>	segment_cortex() path path_file_name
PixelSizeOfSliceDimX	pixelsize_x
PixelSizeOfSliceDimY	pixelsize_y
PrepareROIContrasts()	X.prepare_roi_contrasts(n_predictors) prepare_roi_contrasts
FileNameOfPreprocessdFMR	preprocessed_fmr_name preprocessed_vtc_name
StimulationProtocolBackgroundColorB	protocol_background_color_b
StimulationProtocolBackgroundColorG	protocol_background_color_g
StimulationProtocolBackgroundColorR	protocol_background_color_r
StimulationProtocolExperimentName	protocol_experiment_name
StimulationProtocolFile	protocol_file
StimulationProtocolResolution	protocol_resolution
StimulationProtocolTextColorB	protocol_text_color_b
StimulationProtocolTextColorG	protocol_text_color_g
StimulationProtocolTextColorR	protocol_text_color_r
StimulationProtocolTimeCourseColorB	protocol_timecourse_color_b
StimulationProtocolTimeCourseColorG	protocol_timecourse_color_g
StimulationProtocolTimeCourseColorR	protocol_timecourse_color_r
StimulationProtocolTimeCourseThickne ss	protocol_timecourse_pen_width
PSCTransformStudies()	psc_transform_studies

LinearTrendRemoval()	X.remove() X.remove_linear_trend() -> bool
SaveAs()	X.save() -> bool X.save_as(FileName) -> bool X.save_fmr_stc_from_mem(FMRFileName) -> bool
SaveGLM()	X.save_glm(glm_file)
SaveMultiStudyGLMDefinitionFile()	X.save_maps(vmp_file) X.save_multistudy_glm_definition_file(mdm_file)
SaveStimulationProtocol()	X.save_protocol(protocol_filename)
SaveSingleStudyGLMDesignMatrix()	X.save_run_designmatrix(sdm_file)
SaveSnapshotOfSurfaceWindow()	X.save_screenshot_of_mesh_viewer(screenshot_file) -> bool
SaveVTC()	X.save_vtc(vtc_file) -> bool
ScalePredictorValues()	X.scale_predictor(predictor_name, new_max_value, scale_only_if_vals_positive) sdm_contains_constant_predictor
SelectVOI()	X.select_voi(voi_idx)
SeparationOfStudyPredictors	separate_study_predictors
SeparationOfSubjectPredictors	separate_subject_predictors
SetConditionColor()	X.setObjectName(name) X.set_condition_color(condition_name, color_r, color_g, color_b)
CorrectForSerialCorrelations	serial_correlation_correction_level set_condition_color
SetContrastString()	X.set_contrast_string(contrast_string)
SetContrastValue()	X.set_contrast_value(predictor_name, value)
SetContrastValueAtIndex()	X.set_contrast_value_at_index(predictor_index, value)
SetCurrentContrast()	X.set_current_contrast(contrast_name)
SetCurrentContrastAtIndex()	X.set_current_contrast_at_index(contrast_index)
SetPredictorValues()	X.set_predictor_values(predictor_name, interval_start, interval_end, value)
SetPredictorValuesFromCondition()	X.set_predictor_values_from_condition(predictor_name, protocol_condition_name, predictor_value)
SetVoxelIntensity(x, y, z, intensity)	X.set_voxel_intensity(x, y, z, value)
ShowGLM()	X.show_glm()
ShowVolumeMap()	X.show_map(idx) X.show_maps_dialog()
ShowSelectedVOIs()	X.show_selected_vois() slice_thickness
SpatialGaussianSmoothing()	X.smooth_spatial(gauss_fwhm, fwhm_unit) -> bool
TemporalGaussianSmoothing()	X.smooth_temporal(gauss_fwhm, fwhm_unit) -> bool
	X.snapshot() (for documents)
	X.startTimer(interval, timerType) -> int
	X.thread()
	X.timerEvent(event)
	timeres_verified
	X.tr(text, ambig, n) -> str

TransformToIsoVoxel()	TR X.transform_to_isovoxel(interpolation, vmr_iso_filename, target_res, framing_cube_dim) -> bool transform_to_iso_voxel(float target_res, int framing_cube_dim, int interpolation_method, strout_vmr_iso_filename) → bool
UpdateSurfaceWindow()	X.update_mesh_viewer() X.update_view(process_events) X.vmp_dims() -> tuple
VoxelResolutionVerified	voxelsize_verified
PixelSizeOfSliceDimX	voxelsize_x
PixelSizeOfSliceDimY	voxelsize_y
SliceThickness	voxelsize_z
TargetVTCBoundingBoxXStart	vtc_creation_bounding_box_from_x
TargetVTCBoundingBoxYStart	vtc_creation_bounding_box_from_y
TargetVTCBoundingBoxZStart	vtc_creation_bounding_box_from_z
TargetVTCBoundingBoxXEnd	vtc_creation_bounding_box_to_x
TargetVTCBoundingBoxYEnd	vtc_creation_bounding_box_to_y
TargetVTCBoundingBoxZEnd	vtc_creation_bounding_box_to_z
ExtendedTALSpaceForVTCCreation	vtc_creation_extended_tal_space vtc_creation_use_bounding_box vtc_file
ZTransformStudies	z_transform_studies
ZTransformStudiesBaselineOnly	z_transform_studies_baseline_only

Until BV 22 - general BV API

From BV 22 - MeshScenePyBV

<doc.functionname(arguments) -> returnvalue>

AddCurvatureFileForGroupCBA(CurvatureFile)	X.add_curvature_file_for_group_cba(curvature_smp_file) -> bool
AddMesh()	X.add_mesh(mesh_file) -> bool
ClearGroupCBACurvatureFiles()	X.clear_group_cba_curvature_files() count
CalculateCurvatureCBA()	X.create_average_curvature_group_map() -> bool X.create_average_folded_group_mesh() -> bool
CreateSphereFromFoldedMesh()	X.create_sphere_mesh() -> str
CurrentMesh	current_mesh = MeshPyBV
	X.findChild(type, name)
	X.findChildren(type, name)
	X.isWidgetType() -> bool
	X.isWindowType() -> bool
	X.killTimer(id)
LoadMesh()	X.load_mesh(mesh_file) -> bool
MapSphereMeshFromStandardSphere()	X.map_sphere_from_standard_sphere() -> str

RunCBA()	X.merge_meshes() -> str
RunRigidCBA()	X.run_cba() -> bool
	X.run_rigid_cba(target_pmp_file) -> bool
	X.save_snapshot_of_viewer(snapshot_file)
	-> bool
SetStandardSphereToFoldedMesh()	X.set_standard_sphere_to_folded_mesh(fold
	ed_mesh_file) -> str
SaveSnapshotOfSurfaceWindow()	X.snapshot()
	sphere_resolution_cba
mesh.UpdateAppearance()	X.startTimer(interval, timerType) -> int
mesh.MeshScene.UpdateSurfaceWindow()	X.update_viewer(process_events)
	viewpoint_pos_x
	viewpoint_pos_y
	viewpoint_pos_z
	viewpoint_rot_x
	viewpoint_rot_y
	viewpoint_rot_z
	shrink_wrap_morph(int n_cycles, float find_v
	mr_value) -> bool
	smooth_geometry_simple(int n_cycles, float s
	mooth_force) -> bool

MeshPyBV

```
int size() const;
MeshPyBV *load_mesh(QString mesh_file);
MeshPyBV *add_mesh(QString mesh_file);
QString merge_meshes();
MeshPyBV *create_sphere_mesh(int radius,
int resol_level=1, bool
use_head_colors=true); // not specific for cba
(see next), for general case, especially
subsequent head mesh reconstruction
MeshPyBV *reconstruct_mesh(bool
extended_nbrs=true); // new BV 22.0
MeshPyBV *reconstruct_volume(bool
extended_nbrs=true); // new BV 22.2, name
change, earlier name deprecated
```

Mesh morphing

```
bool shrink_wrap_morph(int n_cycles, double
find_vmr_value);
bool smooth_geometry_simple(int n_cycles,
double smooth_force);
bool smooth_geometry(int n_cycles, double
smooth_force);
bool inflate_geometry(int n_cycles, double
smooth_force, QString
area_reference_mesh="");
bool inflate_geometry_to_sphere(int
n_cycles);
//bool InflateGeometryToSphereExt(int
n_cycles, double ToSphereForceFrom,
double ToSphereForceTo, double
ToSphereCorrectionForceFrom, double
ToSphereCorrectionForceTo);
bool correct_inflated_sphere_distortions(int
n_cycles);
QString simplify_geometry(int
n_target_vertices);
```

SMP functions

```
bool load_maps(QString maps_file); // new
v20.0
void save_maps(QString maps_file); // new
v20.0
void show_map(int idx); // new v20.0
QString map_name(int idx); // new v20.0
void calculate_curvature();

void calculate_curvature_cba();
void create_map_from_volume_map(int
interpolation_method, bool
sample_only_nonzero_values); // new v20.6
```

```

void
create_map_from_volume_map_depth(int
interpolation_method, bool
sample_only_nonzero_values, bool
sample_max_value, double depth_start,
double depth_end, double stepsize); // new
v20.6
void smooth_current_map(int n_cycles);
void smooth_map(int idx, int n_cycles, bool
restrict_to_nonzero_values, bool
include_only_nonzero_values, bool
restrict_to_values_larger_abs_thresh); // new
v20.6
void smooth_map_lags(int idx, int n_cycles,
bool circular_lags, bool
restrict_to_nonzero_values, bool
include_only_nonzero_nbr_values, bool
restrict_to_values_larger_absthresh);
QString create_multiscale_curvature_map(int
smooth_level_1, int smooth_level_2, int
smooth_level_3, int smooth_level_4); // used
for CBA preparation
QString
create_spherical_coordinates_map_from_sm
p(QString on_sphere_smp_file); // used for
CBA preparation - returns created PMP file

```

MTC functions

FileNameOfPreprocessdMTC

```

file_name_preprocessd_mtc
bool create_mtc_from_vtc(double
sample_along_normal_from, double
sample_along_normal_to, QString
output_mtc_file);

```

LinkMTC()

```

bool link_mtc( QString mtc_file ); // , bool
LoadInMemory );

```

SaveMTC()

```

bool save_mtc( QString mtc_file );
bool smooth_mtc(int n_cycles);

```

LinearTrendRemoval()

```

bool remove_linear_trend_mtc();
bool filter_temporal_highpass_fft_mtc(int
cycles_in_timecourse);
bool smooth_temporal_mtc(double fwhm,
QString fwhm_unit);
void clear_run_designmatrix();
bool load_run_designmatrix(QString
sdm_file);
bool add_run_designmatrix(QString
sdm_file);
void save_run_designmatrix(QString
sdm_file);
void compute_run_glm();
void show_glm();

```

```

bool load_glm(QString glm_file);
void save_glm(QString glm_file);
void clear_multistudy_glm_definition();
void
add_study_dm_cortexmapping_glm(QString
func_file, QString dm_file, QString ssm_file);
void
save_multistudy_designmatrix_glm(QString
mdm_file);
bool
load_multistudy_designmatrix_glm(QString
mdm_file);
void compute_multistudy_glm();
void compute_rfx_glm();

```

CBA

```

MeshPyBV *create_sphere_mesh_cba();
QString
map_sphere_from_standard_sphere(); //
(QString MorphedSphereFileName); - we do
not use param, but created (morphed)
sphere mesh must be the only present
(loader) file when calling this fn
QString
set_standard_sphere_to_folded_mesh(QStri
ng folded_mesh_file);
void clear_group_cba_curvature_files();
bool
add_curvature_file_for_group_cba(QString
curvature_smp_file);
bool run_rigid_cba(QString target_pmp_file);
bool run_cba(); // default version - use
dynamic group averaging version; todo: add
pairwise (group) other alignment
bool
create_average_curvature_group_map();
bool create_average_folded_group_mesh();
void update_viewer(bool
process_events=true);
bool save_screenshot_of_viewer(QString
screenshot_file);

```

general/visualization functions

BV workflows

```

print(bv.projects_path)
bv.projects (property)
proj.name
proj = bv.project(0)
proj.subjects
info = proj.subject_info(subj)

```

```
# 'ppdict': helper to pretty-print dictionary (nbt.nbt.ppdict(info)
-> 'notebook tools' module)
```

```
wfs = proj.workflows
wfs.count
wf = wfs.item(i)
wf.reference_id
wf.name
proj.remove_all_workflows()
anatpp = proj.create_anatpp_workflow('vmr-pp')
type(anatpp)
proj.connect_sourcedata_to(anatpp)
wf_io = anatpp.inoutmap
anatnorm =
proj.create_anatnorm_workflow('vmr-norm')
anatnorm.reference_id
anatnorm.name
anatnorm.type_id
anatnorm.type_name
connect(workflow_from, workflow_to)
workflow.parameters
workflow.run()
nbt.ppdict(anatpp.inoutvars)
proj.remove_workflow(proj.workflow(4))
coreg = proj.create_coreg_workflow('coreg')
proj.connect_coreg(proj.workflow(3),
proj.workflow(1), coreg)
funcnorm =
proj.create_funcnorm_workflow('funcn')
proj.connect_funcnorm(proj.workflow(3),
proj.workflow(4), proj.workflow(2), funcnorm)
glm = proj.create_groupglm_workflow('group-
glm')
proj.connect_groupglm(proj.workflow(5),
proj.workflow(2), glm)

anatpp.include_subjects
```

```
Data access; requires:
import bva
```

```
import numpy as np
```

```
arr = bva.vmrarray(doc_vmr)
doc_vmr = bv.adoc
arrf = arr.astype(np.float32)
ret = bv.set_vmrarray(doc_vmr, arr_ui8)
doc_vmr.update_view()
uint8-array vmrarray(vmr-document)
set_vmrarray(vmr-document, uint8-array)
float32-array fmrarray_t(fmr-document, t)
set_fmrarray_t(fmr_document, t, float32-
array)
float32-array vmparray_i(vmp, i)
set_vmparray_i(vmp, i, float32-array)
(coords-x-float32-array, coords-y-float32-
array, coords-z-float32-array)
mesharrays(mesh-document)
set_mesharrays(coords-x-float32-array,
coords-y-float32-array, coords-z-float32-
array)
float32-array smparray_i(smp, i)
set_smparray_i(smp, i, float32-array)
```