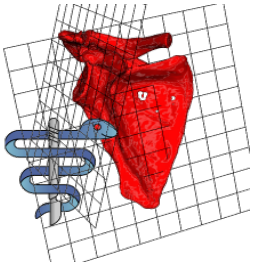
glenoidplanefitting Documentation

Asta Olafsdottir

Jan 25, 2023

CONTENTS



Authors: Asta Olafsdottir, Stephen Thompson

glenoidplanefitting provides tools for measuring the Glenoid version, useful when planning reconstructive shoulder surgery. glenoidplanefitting is part of the scikit-surgery software project, developed at the Wellcome EPSRC Centre for Interventional and Surgical Sciences, part of University College London (UCL), in collaboration with surgeons at the Royal National Orthopaedic Hospital

glenoidplanefitting is tested on Python 3.x, but may work with other versions of Python

glenoidplanefitting currently requires the user to manually identify relevant anatomical landmarks on the scapula and glenoid dish. These points can then be passed to glenoidplanefitting to calculate the Glenoid version using multiple methods. Example usage:

```
python glenoidplanefitting.py --fried_points landmark_friedman.fcsv --output friedman.
↪vtp --visualise glenoid.vtp
```

glenoidplanefitting is a work in progress with plans to automate the point picking to create software to enable a large scale comparison of different methods of Glenoid version methods on a clinical data set. If you are interested in contributing to this work please get in touch or follow the guides below.

CHAPTER

ONE

DEVELOPING

1.1 Cloning

You can clone the repository using the following command:

```
git clone https://github.com/SciKit-Surgery/glenoidplanefitting
```

1.2 Running tests

We use tox to run tests and run static code analysis using Lint.

```
pip install tox
python -m tox
```

CHAPTER

TWO

INSTALLING

You can pip install directly from the repository as follows:

```
pip install git+https://github.com/SciKit-Surgery/glenoidplanefitting
```

2.1 Contributing

Please see the contributing guidelines.

2.2 Useful links

- Source code repository
- Documentation

CHAPTER

THREE

LICENSING AND COPYRIGHT

CHAPTER

FOUR

ACKNOWLEDGEMENTS

Supported by Wellcome and EPSRC.

4.1 Dependency Graph

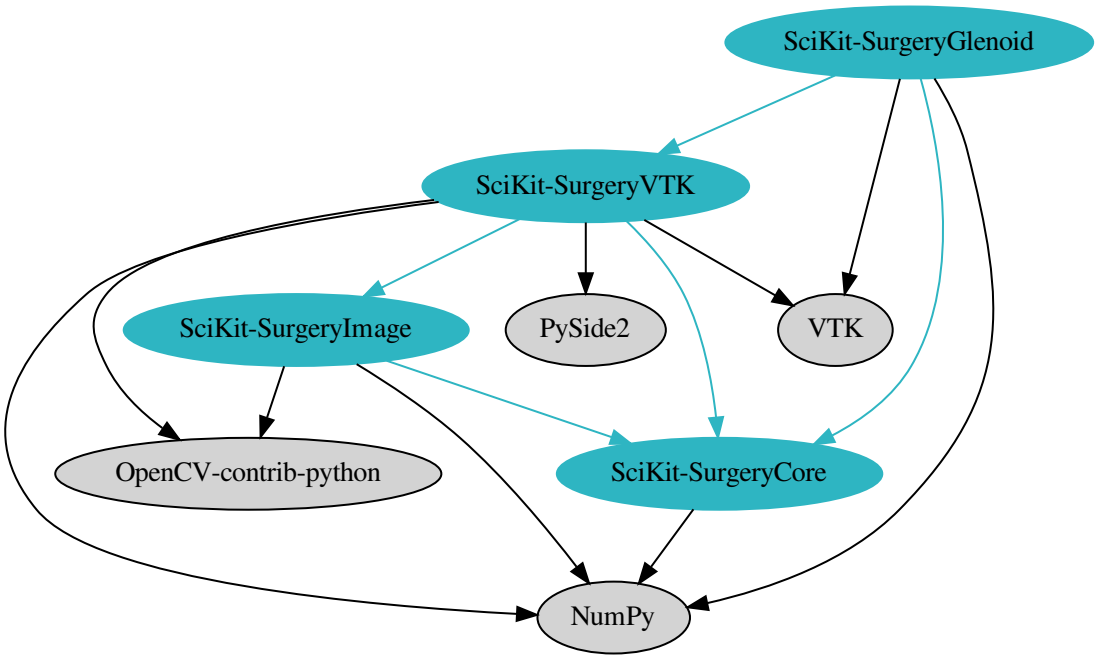


Fig. 1: glenoidplanefitting's Dependencies

4.2 latest

4.2.1 glenoidplanefitting package

Subpackages

glenoidplanefitting.algorithms package

Submodules

glenoidplanefitting.algorithms.colour_palette module

A colour palette based on Bang's colour pallette Wong, B. Points of view: Color blindness. Nat Methods 8, 441 (2011). https://doi.org/10.1038/nmeth.1618

glenoidplanefitting.algorithms.colour_palette.**bang_list**()

Returns the bang palette as a list

glenoidplanefitting.algorithms.friedman module

This is an implentation of Friedman's method, see

Friedman RJ, Hawthorne KB and Genez BM. The use of computerized tomography in the measurement of glenoid version. J Bone and Joint Surg Am 1992; 74: 1032–7.

glenoidplanefitting.algorithms.friedman.**create_friedman_line**(*point0*, *point1*)

Determines the second point needed to form the Friedman line

Parameters

- **point0** – First point on glenoid line, anatomically defined as a point on the anterior margin of glenoid
- **point1** – Second point on glenoid line anatomically defined as a point on the posterior margin of glenoid

Raises

Value Error if the z values of point0 and point1 are not equal

Returns

The midpoint of the glenoid line, which is the second point of the Friedman line

glenoidplanefitting.algorithms.friedman.**friedman_version**(*glenoid0*, *friedman1*, *friedman0*)

Determines the glenoid version using the Friedman line

Parameters

- **glenoid0** – First point on glenoid line, anatomically defined as a point on the anterior margin of the glenoid
- **friedman1** – Second point on the Friedman line, anatomically defined as the midpoint of the glenoid fossa
- **friedman0** – First point on the Friedman line, anatomically defined as the medial tip of the scapula

Returns
The glenoid version (positive value indicates retroversion)

glenoidplanefitting.algorithms.models module

Functions to create vtk models for visualisation of results

glenoidplanefitting.algorithms.models.**make_friedman_model**(*point1*, *point2*)

Makes a vtk line source from two set points

Parameters

- **point1** – one end of the line
- **point2** – other end of the line

Returns
The line

glenoidplanefitting.algorithms.models.**make_plane_model**(*plane_centre*, *normal_vector*, *resolution=10*, *plane_size=200.0*)

Makes a vtk plane source, with centre and normal vector

Parameters

- **plane_centre** – a point on the plane
- **normal_vector** – the plane normal vector

Returns
The plane

glenoidplanefitting.algorithms.models.**make_sphere_model**(*point*, *radius=5.0*)

Make a sphere source which we can use to represent a landmark point

Parameters
point – the point

:returns the vtkPointSource

glenoidplanefitting.algorithms.models.**make_vault_model**(*point1*, *point2*)

Makes a vtk line source from two set points

Parameters

- **point1** – one end of the line
- **point2** – other end of the line

Returns
The line

glenoidplanefitting.algorithms.plane_fitting module

This is an implementation of a two plane method, see

A. Ganapathi, J. McCarron, Chen, J. Iannotti. Predicting normal glenoid version from the pathologic scapula: a comparison of 4 methods in 2- and 3-dimensional models J Shoulder Elbow Surg (2011) 20, 234-244

glenoidplanefitting.algorithms.plane_fitting.**fit_plane_to_points_glenoid**(*points2*, *return_meta2=False*)

Fit a plane to a set of manually selected points on the glenoid face

Parameters

- **points1** – np.ndarray, size n by 3 array of the following points, one superior on the glenoid face, two inferior on the glenoid face left and right side
- **return_meta** – If true, also returns the center and normal used to generate the plane

Returns
the fitted plane of the glenoid face

glenoidplanefitting.algorithms.plane_fitting.**fit_plane_to_points_scapula**(*points1*, *return_meta1=False*)

Fit a plane to a set of manually selected points on the scapula

Parameters

- **points1** – np.ndarray, size n by 3 array of the following points, inferior tip of scapula, medial border of scapula, and center of glenoid fossa.
- **return_meta** – If true, also returns the center and normal used to generate the plane

Returns
the fitted plane through the scapula

glenoidplanefitting.algorithms.plane_fitting.**fit_plane_transverse**(*points1*, *points3*, *return_meta3=False*)

Fit a transverse plane perpendicular to the scapular plane and passing through the scapular axis.

Parameters

- **points1** – np.ndarray, size n by 3 array of the following points, inferior tip of scapula medial border of scapula, and center of glenoid fossa.
- **points3** – np.ndarray, size n by 3 of the following points, center of glenoid fossa, and medial border
- **return_meta** – If true, also returns the center and normal used to generate the plane

Returns
the fitted transverse plane

glenoidplanefitting.algorithms.plane_fitting.**planes_version**(*normal_plane1*, *normal_plane2*)

Determines the glenoid version using the two planes method.

Parameters

- **normal_plane1** – The normal vector of the scapula plane.
- **normal_plane2** – The normal vector of the glenoid plane.

Returns
The glenoid version (positive value indicates retroversion)

glenoidplanefitting.algorithms.vault module

This is an implentation of the vault method, see

Matsumura N et al. Computed tomography measurement of glenoid vault version as an alternative measuring method for glenoid version. Journal of Orthopaedic Surgery and Research 2014, 9:17

glenoidplanefitting.algorithms.vault.**create_vault_line**(*anterior_point*, *posterior_point*)

Determines the second point needed to form the Friedman line :param anterior_point: First point on glenoid line, anatomically defined

as a point on the anterior margin of glenoid

Parameters

posterior_point – Second point on glenoid line anatomically defined as a point on the posterior margin of glenoid

Returns

The midpoint of the glenoid line, or the second point of for the vault line

glenoidplanefitting.algorithms.vault.**vault_version**(*anterior_point*, *midpoint*, *vaultpoint*)

Determines the glenoid version using the glenoid vault as reference :param anterior_point: First point on glenoid line, anatomically defined

as a point on the anterior margin of the glenoid

Parameters

- **midpoint** – Second point on the vault line, anatomically defined as the midpoint of the glenoid fossa
- **vaultpoint** – First point on the vault line, anatomically defined as the tip of the glenoid vault

Returns

The glenoid version (positive value indicates retroversion)

Module contents

glenoidplanefitting.ui package

Submodules

glenoidplanefitting.ui.glenoidplanefitting_command_line module

Command line processing

glenoidplanefitting.ui.glenoidplanefitting_command_line.**main**(*args=None*)

Entry point for glenoidplanefitting application

glenoidplanefitting.ui.glenoidplanefitting_demo module

Main entry point function for the various plane fitting functions

glenoidplanefitting.ui.glenoidplanefitting_demo.**run_demo**(*model_file_name*, *planes=''*, *fried_points=''*, *vault_points=''*, *corr_fried=''*, *output=''*, *visualise=False*, *config_file=None*)

Parameters

- **planes** – File name pointing to file containing points for planes method. First three points are for scapula plane in the order of medial border, inferior tip, glenoid center. For left shoulder latter three points are in order of superior glenoid, left inferior glenoid, right inferior glenoid. For right shoulder latter three points are in order of superior glenoid, right inferior glenoid, left inferior glenoid.
- **fried_points** – File name pointing to a file containing points for the Friedman method. If left shoulder, points in order of, right glenoid tip, left glenoid tip, scapula tip. If right shoulder, points in order of left glenoid tip, right glenoid tip, scapula tip.
- **vault_points** – File name pointing to a file containing points for the vault method. If left shoulder, points in order of, right glenoid tip, left glenoid tip, vault tip. If right shoulder, points in order of, left glenoid tip, right glenoid tip, vault tip.
- **corr_fried** – File name pointing to a file containing points for the corrected Friedman method. Input file: If left shoulder, points in order of, right glenoid tip, left glenoid tip, vault tip. If right shoulder, points in order of, left glenoid tip, right glenoid tip, vault tip.
- **output** – Output filename, can be planes.vtp, friedman.vtp, or vault.vtp Choosing planes.vtp Writes the transverse plane into a file used as the new axial slice for picking the new landmark points for the 3D corrected Friedman method.
- **config_file** – We can pass a configuration file, currently focusing on visualisation parameters

Module contents

glenoidplanefitting

glenoidplanefitting.widgets package

Submodules

glenoidplanefitting.widgets.visualisation module

Widgets for show the results of plane fitting.

glenoidplanefitting.widgets.visualisation.**add_vtk_source**(*renderer*, *source*, *linewidth=1.0*, *opacity=1.0*, *wireframe=False*, *colour=None*)

simplifies adding a vtk geometry source to a renderer

Parameters

- **renderer** – a vtk renderer to add to

- **source** – a vtk geometry source

glenoidplanefitting.widgets.visualisation.**render_window_common**(*renderer*, *window_name*)

Creates and starts a render window and interactor.

Parameters

- **renderer** – A vtk renderer to add to the render window
- **window_name** – A name for the window

glenoidplanefitting.widgets.visualisation.**renderer_common**(*bone*, *background_colour=None*)

Initialises a vtk renderer and adds the bone model

glenoidplanefitting.widgets.visualisation.**vis_fried**(*bone*, *cross1*, *cross2*, *glenoid1*, *result*, *line_width=5*)

Visualise the lines resulting from the friedman method.

Parameters

- **cross2** (*cross1,*) – The end points of the line crossing the glenoid
- **result** (*glenoid1,*) – The end points of the line defining the glenoid version

glenoidplanefitting.widgets.visualisation.**vis_planes**(*bone*, *planes*, *points1=False*, *points2=False*, *resolution=1*, *plane_size=200.0*, *vary_plane_colour=False*, *point_size=5.0*)

Visualisation for plane fitting methods

Parameters

- **bone** – The model surface model
- **planes** – a list of planes, as returned by the plane fitting methods in algorithms.plane_fitting

glenoidplanefitting.widgets.visualisation.**vis_vault**(*bone*, *cross1*, *cross2*, *glenoid1*, *result*, *line_width=5*)

Visualise the lines resulting from the vault method.

Parameters

- **cross2** (*cross1,*) – The end points of the line crossing the glenoid
- **result** (*glenoid1,*) – The end points of the line defining the glenoid version

Module contents

Module contents

glenoidplanefitting

- modindex
- genindex
- search

PYTHON MODULE INDEX

a

g

u

w

INDEX

P

R

V