# PySAR Documentation

## Version 1.2

Zhang Yunjun, Heresh Fattahi

MAR 7, 2017

# Contents


**1 Welcome to PySAR!** 1

**2 _Sidebar** 3

**3 Attributes** 5

**4 Coordinate** 7

**5 File-Descriptions** 9

**6 Gamma-File-Decription** 11

**7 Google-Earth** 13

**8 Welcome to PySAR wiki!** 15

**9 Mask** 17

**10 pysarApp** 19

**11 UNAVCO-InSAR-Archive** 21

**12 InSAR Time Series Web Viewer: http://insarmaps.rsmas.miami.edu** 23

**13 Namespace Index** 25

13.1 Packages . . . . . . . . . . 25

**14 Hierarchical Index** 27

14.1 Class Hierarchy . . . . . . . . . . 27

**15 Class Index** **29**

15.1 Class List . . . . . . . . . . 29

**16 File Index** **31**

16.1 File List . . . . . . . . . . 31

**17 Namespace Documentation** **33**

17.1 delayTimeseries Namespace Reference . . . . . . . . . . 33

17.1.1 Function Documentation . . . . . . . . . . 33

17.1.1.1 nearest_valid() . . . . . . . . . . 33

17.1.1.2 write_to_h5() . . . . . . . . . . 33

17.2 dloadUtil Namespace Reference . . . . . . . . . . 34

17.2.1 Function Documentation . . . . . . . . . . 34

17.2.1.1 daterange() . . . . . . . . . . 34

17.2.1.2 download_atmosphereModel() . . . . . . . . . . 34

17.2.1.3 download_modis() . . . . . . . . . . 34

17.2.1.4 get_date() . . . . . . . . . . 34

17.2.1.5 pwv2zwd() . . . . . . . . . . 35

17.2.1.6 read_modis() . . . . . . . . . . 35

17.2.1.7 zwd2swd() . . . . . . . . . . 35

17.3 get_modis_v3 Namespace Reference . . . . . . . . . . 35

17.3.1 Function Documentation . . . . . . . . . . 35

17.3.1.1 main() . . . . . . . . . . 35

17.3.1.2 usage() . . . . . . . . . . 36

17.3.2 Variable Documentation . . . . . . . . . . 36

17.3.2.1 out . . . . . . . . . . 36

17.3.2.2 start_time_main . . . . . . . . . . 36

17.3.2.3 time_elapsed . . . . . . . . . . 36

17.4 plot_tropcor_phase_elevation Namespace Reference . . . . . . . . . . 36

17.4.1 Variable Documentation . . . . . . . . . . 37

17.4.1.1 atr . . . . . . . . . . 37

17.4.1.2 atr2 . . . . . 37
17.4.1.3 atr3 . . . . . 37
17.4.1.4 atr4 . . . . . 37
17.4.1.5 axes . . . . . 38
17.4.1.6 bbox_inches . . . . . 38
17.4.1.7 data . . . . . 38
17.4.1.8 data2 . . . . . 38
17.4.1.9 dataList . . . . . 38
17.4.1.10 dem . . . . . 38
17.4.1.11 dem_atr . . . . . 38
17.4.1.12 demFile . . . . . 38
17.4.1.13 dpi . . . . . 39
17.4.1.14 ecmwf . . . . . 39
17.4.1.15 ecmwfFile . . . . . 39
17.4.1.16 epoch . . . . . 39
17.4.1.17 fig . . . . . 39
17.4.1.18 figsize . . . . . 39
17.4.1.19 i . . . . . 39
17.4.1.20 mask . . . . . 39
17.4.1.21 maskFile . . . . . 40
17.4.1.22 ms . . . . . 40
17.4.1.23 msk_atr . . . . . 40
17.4.1.24 ncols . . . . . 40
17.4.1.25 ndx . . . . . 40
17.4.1.26 nrows . . . . . 40
17.4.1.27 sharex . . . . . 40
17.4.1.28 sharey . . . . . 40
17.4.1.29 timeseriesFile . . . . . 41
17.4.1.30 timeseriesFile2 . . . . . 41
17.4.1.31 tropHgt . . . . . 41

17.4.1.32 tropHgtFile . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
17.4.1.33 True . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
17.4.1.34 workDir . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
17.5 pysar Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . 42
17.5.1 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . 43
17.5.1.1 miami_path . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
17.6 pysar._datetime Namespace Reference . . . . . . . . . . . . . . . . . . . . . 43
17.6.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . 44
17.6.1.1 auto_adjust_xaxis_date() . . . . . . . . . . . . . . . . . . . . . . 44
17.6.1.2 date_index() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
17.6.1.3 date_list2tbase() . . . . . . . . . . . . . . . . . . . . . . . . . . 44
17.6.1.4 date_list2vector() . . . . . . . . . . . . . . . . . . . . . . . . . 44
17.6.1.5 igram_date_list() . . . . . . . . . . . . . . . . . . . . . . . . . . 45
17.6.1.6 read_date_list() . . . . . . . . . . . . . . . . . . . . . . . . . . 45
17.6.1.7 yymmdd() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
17.6.1.8 yymmdd2yyyymmdd() . . . . . . . . . . . . . . . . . . . . . . . . . 45
17.6.1.9 yyyymmdd() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
17.6.1.10 yyyymmdd2years() . . . . . . . . . . . . . . . . . . . . . . . . . . 45
17.7 pysar._gmt Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . 46
17.7.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . 46
17.7.1.1 write_gmt_simple() . . . . . . . . . . . . . . . . . . . . . . . . . 46
17.8 pysar._network Namespace Reference . . . . . . . . . . . . . . . . . . . . . . 46
17.8.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . 47
17.8.1.1 auto_adjust_yaxis() . . . . . . . . . . . . . . . . . . . . . . . . . 47
17.8.1.2 date12_list2index() . . . . . . . . . . . . . . . . . . . . . . . . . 47
17.8.1.3 get_date12_list() . . . . . . . . . . . . . . . . . . . . . . . . . . 47
17.8.1.4 igram_perp_baseline_list() . . . . . . . . . . . . . . . . . . . . . 48
17.8.1.5 pair_merge() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
17.8.1.6 pair_sort() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
17.8.1.7 plot_network() . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

17.8.1.8 plot_perp_baseline_hist() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
17.8.1.9 read_baseline_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
17.8.1.10 read_igram_pairs() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
17.8.1.11 read_pairs_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
17.8.1.12 select_pairs_all() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
17.8.1.13 select_pairs_delaunay() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
17.8.1.14 select_pairs_hierarchical() . . . . . . . . . . . . . . . . . . . . . . . . . . 50
17.8.1.15 select_pairs_mst() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
17.8.1.16 select_pairs_sequential() . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
17.8.1.17 select_pairs_star() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
17.8.1.18 threshold_perp_baseline() . . . . . . . . . . . . . . . . . . . . . . . . . . 51
17.8.1.19 threshold_temporal_baseline() . . . . . . . . . . . . . . . . . . . . . . . . 52
17.8.1.20 write_pairs_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
17.9 pysar._pysar_utilities Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . 52
17.9.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
17.9.1.1 Baseline_timeseries() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
17.9.1.2 Bh_Bv_timeseries() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
17.9.1.3 check_variable_name() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
17.9.1.4 date_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
17.9.1.5 dBh_dBv_timeseries() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
17.9.1.6 design_matrix() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
17.9.1.7 generate_curls() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
17.9.1.8 get_file_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
17.9.1.9 get_file_stack() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
17.9.1.10 get_triangles() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
17.9.1.11 glob2radar() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
17.9.1.12 hillshade() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
17.9.1.13 incidence_angle() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
17.9.1.14 make_triangle() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
17.9.1.15 nonzero_mask() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

17.9.1.16 print_progress() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
17.9.1.17 radar2glob() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
17.9.1.18 radar_or_geo() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
17.9.1.19 spatial_average() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
17.9.1.20 stacking() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
17.9.1.21 temporal_average() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
17.9.1.22 timeseries_inversion() . . . . . . . . . . . . . . . . . . . . . . . . . . 58
17.9.1.23 timeseries_inversion_FGLS() . . . . . . . . . . . . . . . . . . . . . . 58
17.9.1.24 timeseries_inversion_L1() . . . . . . . . . . . . . . . . . . . . . . . . 58
17.9.1.25 which() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
17.9.1.26 yymmdd() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
17.9.1.27 yymmdd2YYYYMMDD() . . . . . . . . . . . . . . . . . . . . . . . . . . 59
17.9.1.28 yyyymmdd() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
17.9.1.29 YYYYMMDD2years() . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
17.10 pysar._readfile Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . 59
17.10.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
17.10.1.1 check_variable_name() . . . . . . . . . . . . . . . . . . . . . . . . . . 60
17.10.1.2 merge_attribute() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
17.10.1.3 read() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
17.10.1.4 read_attribute() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
17.10.1.5 read_complex_float32() . . . . . . . . . . . . . . . . . . . . . . . . . 61
17.10.1.6 read_complex_int16() . . . . . . . . . . . . . . . . . . . . . . . . . . 61
17.10.1.7 read_dem() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
17.10.1.8 read_flag() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
17.10.1.9 read_float32() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
17.10.1.10 read_gamma_par() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
17.10.1.11 read_GPS_USGS() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
17.10.1.12 read_isce_xml() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
17.10.1.13 read_multiple() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
17.10.1.14 read_real_float32() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

17.10.1.15 read_real_int16() . . . . . 64
17.10.1.16 read_roipac_rsc() . . . . . 64
17.10.1.17 read_template() . . . . . 64
17.10.2 Variable Documentation . . . . . 64
17.10.2.1 multi_dataset_hdf5_file . . . . . 64
17.10.2.2 multi_group_hdf5_file . . . . . 64
17.10.2.3 single_dataset_hdf5_file . . . . . 65
17.11 pysar._remove_surface Namespace Reference . . . . . 65
17.11.1 Function Documentation . . . . . 65
17.11.1.1 remove_data_multiple_surface() . . . . . 65
17.11.1.2 remove_data_surface() . . . . . 65
17.11.1.3 remove_surface() . . . . . 65
17.12 pysar._writefile Namespace Reference . . . . . 66
17.12.1 Function Documentation . . . . . 66
17.12.1.1 write() . . . . . 66
17.12.1.2 write_complex64() . . . . . 66
17.12.1.3 write_complex_int16() . . . . . 67
17.12.1.4 write_dem() . . . . . 67
17.12.1.5 write_float32() . . . . . 67
17.12.1.6 write_real_float32() . . . . . 67
17.12.1.7 write_real_int16() . . . . . 67
17.13 pysar.add Namespace Reference . . . . . 68
17.13.1 Function Documentation . . . . . 68
17.13.1.1 add() . . . . . 68
17.13.1.2 main() . . . . . 68
17.13.1.3 usage() . . . . . 68
17.14 pysar.add_attribute Namespace Reference . . . . . 68
17.14.1 Function Documentation . . . . . 68
17.14.1.1 main() . . . . . 69
17.14.1.2 usage() . . . . . 69

17.15pysar.add_attributes_insarmaps Namespace Reference . . . . . 69
17.15.1 Function Documentation . . . . . 69
17.15.1.1 build_parser() . . . . . 69
17.15.1.2 main() . . . . . 69
17.15.1.3 parse_file_for_attributes() . . . . . 69
17.15.1.4 usage() . . . . . 70
17.16pysar.asc_desc Namespace Reference . . . . . 70
17.16.1 Function Documentation . . . . . 70
17.16.1.1 corners() . . . . . 70
17.16.1.2 find_row_column() . . . . . 70
17.16.1.3 get_lat_lon() . . . . . 70
17.16.1.4 main() . . . . . 71
17.16.1.5 nearest() . . . . . 71
17.16.1.6 nearest_neighbor() . . . . . 71
17.16.1.7 usage() . . . . . 71
17.17pysar.baseline_error Namespace Reference . . . . . 71
17.17.1 Function Documentation . . . . . 71
17.17.1.1 main() . . . . . 72
17.17.1.2 to_percent() . . . . . 72
17.17.1.3 usage() . . . . . 72
17.18pysar.baseline_trop Namespace Reference . . . . . 72
17.18.1 Function Documentation . . . . . 72
17.18.1.1 main() . . . . . 72
17.18.1.2 to_percent() . . . . . 72
17.18.1.3 usage() . . . . . 73
17.19pysar.convert2mat Namespace Reference . . . . . 73
17.19.1 Function Documentation . . . . . 73
17.19.1.1 main() . . . . . 73
17.19.1.2 usage() . . . . . 73
17.19.1.3 yyyymmdd2years() . . . . . 73

17.20 pysar.correct_dem Namespace Reference . . . . . 73
17.20.1 Function Documentation . . . . . 73
17.20.1.1 main() . . . . . 74
17.20.1.2 usage() . . . . . 74
17.21 pysar.correlation_with_dem Namespace Reference . . . . . 74
17.21.1 Function Documentation . . . . . 74
17.21.1.1 usage() . . . . . 74
17.21.2 Variable Documentation . . . . . 74
17.21.2.1 amp . . . . . 75
17.21.2.2 C1 . . . . . 75
17.21.2.3 data . . . . . 75
17.21.2.4 dem . . . . . 75
17.21.2.5 demRsc . . . . . 75
17.21.2.6 dset . . . . . 75
17.21.2.7 h5data . . . . . 75
17.21.2.8 ndx . . . . . 75
17.21.2.9 subx . . . . . 76
17.21.2.10 suby . . . . . 76
17.22 pysar.dem_error Namespace Reference . . . . . 76
17.22.1 Function Documentation . . . . . 76
17.22.1.1 main() . . . . . 76
17.22.1.2 usage() . . . . . 76
17.23 pysar.diff Namespace Reference . . . . . 76
17.23.1 Function Documentation . . . . . 76
17.23.1.1 diff() . . . . . 77
17.23.1.2 main() . . . . . 77
17.23.1.3 usage() . . . . . 77
17.24 pysar.drop_turbulence Namespace Reference . . . . . 77
17.24.1 Function Documentation . . . . . 77
17.24.1.1 circle_index() . . . . . 77

17.24.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
17.24.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.25pysar.filter_spatial Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.25.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.25.1.1 filter() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.25.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.25.1.3 multilook() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.25.1.4 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
17.26pysar.filter_temporal Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.26.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.26.1.1 get_data() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.26.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.26.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.27pysar.gamma_view Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.27.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
17.27.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.27.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.28pysar.generate_mask Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.28.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.28.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.28.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.29pysar.geocode Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
17.29.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
17.29.1.1 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
17.29.1.2 geocode_attribute() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
17.29.1.3 geocode_data_roipac() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
17.29.1.4 geocode_file_roipac() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
17.29.1.5 geomap4subset_radar_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
17.29.1.6 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
17.29.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82

17.29.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
17.30pysar.igram_closure Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
17.30.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
17.30.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
17.30.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.31pysar.igram_inversion Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.31.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.31.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.31.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.32pysar.image_math Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.32.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
17.32.1.1 add() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
17.32.1.2 diff() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
17.32.1.3 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
17.32.1.4 operation() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
17.32.1.5 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
17.33pysar.incidence_angle Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.33.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.33.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.33.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.34pysar.info Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.34.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.34.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
17.34.1.2 print_attributes() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
17.34.1.3 print_hdf5_structure() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
17.34.1.4 print_timseries_date_info() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
17.34.1.5 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
17.35pysar.insar_vs_gps Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
17.35.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
17.35.1.1 find_row_column() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88

17.35.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
17.35.1.3 nearest() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.1.4 readGPSfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.1.5 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.2.1 ax . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.2.2 C1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.2.3 Coh . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
17.35.2.4 coherence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.5 color . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.6 Cor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.7 dset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.8 fig . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.9 figName . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.10 fmt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.11 fontsize . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
17.35.2.12 GPS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.13 GPS1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.14 GPS2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.15 gps_comp_txt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.16 GPS_station . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.17 GPS_std . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.18 gpsLOS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.19 gpsLOS_ref . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
17.35.2.20 GPSx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.21 GPSx1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.22 GPSx2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.23 GPSy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.24 GPSy1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.25 GPSy2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92

17.35.2.26 h5coh . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.27 heading . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
17.35.2.28 idx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.29 IDX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.30 idxRef . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.31 IDXref . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.32 IDY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.33 IDYref . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.34 InSAR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.35 InSAR1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
17.35.2.36 InSAR2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.37 InSAR_GPS_Copmarison . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.38 insar_velocity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.39 insarData . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.40 kh5coh . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.41 Lat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.42 length . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
17.35.2.43 Lon . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.44 look_f . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.45 look_n . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.46 lt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.47 majorLocator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.48 maxV . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.49 minorLocator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.50 minV . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
17.35.2.51 ms . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
17.35.2.52 NoInSAR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
17.35.2.53 S . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
17.35.2.54 SAD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
17.35.2.55 Se . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96

17.35.2.56 Sg . . . . . 96
17.35.2.57 Sn . . . . . 96
17.35.2.58 Sr . . . . . 97
17.35.2.59 Stations . . . . . 97
17.35.2.60 stationsList . . . . . 97
17.35.2.61 Su . . . . . 97
17.35.2.62 theta . . . . . 97
17.35.2.63 unitVec . . . . . 97
17.35.2.64 Ve . . . . . 97
17.35.2.65 Vn . . . . . 98
17.35.2.66 Vu . . . . . 98
17.35.2.67 which . . . . . 98
17.35.2.68 width . . . . . 98
17.35.2.69 xerr . . . . . 98
17.35.2.70 xy . . . . . 98
17.35.2.71 xytext . . . . . 98
17.35.2.72 yerr . . . . . 98
17.36 pysar.insarmaps_query Namespace Reference . . . . . 99
17.36.1 Function Documentation . . . . . 99
17.36.1.1 build_parser() . . . . . 99
17.36.1.2 buildURL() . . . . . 99
17.36.1.3 main() . . . . . 99
17.37 pysar.l1 Namespace Reference . . . . . 99
17.37.1 Function Documentation . . . . . 100
17.37.1.1 l1() . . . . . 100
17.37.1.2 l1blas() . . . . . 100
17.37.1.3 l1mosek() . . . . . 100
17.37.1.4 l1mosek2() . . . . . 101
17.37.2 Variable Documentation . . . . . 101
17.37.2.1 __MOSEK . . . . . 101

17.37.2.2 task . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
17.37.2.3 x . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
17.38pysar.load_data Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
17.38.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
17.38.1.1 auto_path_miami() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
17.38.1.2 check_existed_hdf5_file() . . . . . . . . . . . . . . . . . . . . . . . . . . 102
17.38.1.3 check_file_size() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
17.38.1.4 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
17.38.1.5 copy_roipac_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
17.38.1.6 load_roipac2multi_group_h5() . . . . . . . . . . . . . . . . . . . . . . . 103
17.38.1.7 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
17.38.1.8 mode() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
17.38.1.9 roipac_nonzero_mask() . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
17.38.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
17.38.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
17.38.2.2 TEMPLATE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
17.39pysar.load_dem Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
17.39.1 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.1 amp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.2 dem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.3 demFile . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.4 demRsc . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.5 dset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.6 ext . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.7 group . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
17.39.1.8 h5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
17.39.1.9 outName . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
17.40pysar.lod Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
17.40.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
17.40.1.1 correct_lod_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106

17.40.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
17.40.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
17.41 pysar.look_angle Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.41.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.41.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.41.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.42 pysar.los2enu Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.42.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.42.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.42.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
17.43 pysar.mask Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
17.43.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
17.43.1.1 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
17.43.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
17.43.1.3 mask_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
17.43.1.4 mask_matrix() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
17.43.1.5 update_mask() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
17.43.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
17.43.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
17.44 pysar.match Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
17.44.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
17.44.1.1 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
17.44.1.2 corners() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
17.44.1.3 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
17.44.1.4 manual_offset_estimate() . . . . . . . . . . . . . . . . . . . . . . . . . 110
17.44.1.5 match_two_files() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
17.44.1.6 nearest() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
17.44.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
17.44.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
17.45 pysar.mean_spatial Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . 111

17.45.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
17.45.1.1 circle_index() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
17.45.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
17.45.1.3 Usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
17.46 pysar.modify_network Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
17.46.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
17.46.1.1 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
17.46.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
17.46.1.3 manual_select_pairs_to_remove() . . . . . . . . . . . . . . . . . . . . . . . . 113
17.46.1.4 modify_file_date12_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
17.46.1.5 nearest_neighbor() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
17.46.1.6 update_inps_with_template() . . . . . . . . . . . . . . . . . . . . . . . . . . 114
17.46.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
17.46.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
17.46.2.2 TEMPLATE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
17.47 pysar.multi_transect Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
17.47.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
17.47.1.1 check_st_in_box() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
17.47.1.2 check_st_in_box2() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
17.47.1.3 dist_point_from_line() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
17.47.1.4 dms2d() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
17.47.1.5 find_row_column() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
17.47.1.6 get_intersect() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
17.47.1.7 get_lat_lon() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
17.47.1.8 get_start_end_point() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
17.47.1.9 get_transect() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
17.47.1.10 gps_to_LOS() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
17.47.1.11 line() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
17.47.1.12 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
17.47.1.13 nanmean() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121

17.47.1.14 nanstd() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
17.47.1.15 nearest() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
17.47.1.16 onclick() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
17.47.1.17 point_on_line_with_distance_from_beginning() . . . . . . . . . . . . . . . . 121
17.47.1.18 point_with_distance_from_line() . . . . . . . . . . . . . . . . . . . . . . . 121
17.47.1.19 read_fault_coords() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.1.20 readGPSfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.1.21 redGPSfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.1.22 redGPSfile_cmm4() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.1.23 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.2.1 alpha . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
17.47.2.2 avgInSAR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.3 ax . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.4 axes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.5 axes2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.6 c . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.7 c_prof_edge . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.8 cf . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.9 check_result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
17.47.2.10 check_result2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.11 cid . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.12 color . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.13 D . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.14 dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.15 df0_km . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.16 dg . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
17.47.2.17 DistGPS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125
17.47.2.18 dp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125
17.47.2.19 dset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125

17.47.2.20 dx . . . 125
17.47.2.21 DX . . . 125
17.47.2.22 dy . . . 125
17.47.2.23 DY . . . 125
17.47.2.24 Fault_lat . . . 125
17.47.2.25 Fault_lon . . . 126
17.47.2.26 FaultCoords . . . 126
17.47.2.27 FaultLine . . . 126
17.47.2.28 fig . . . 126
17.47.2.29 fig2 . . . 126
17.47.2.30 figName . . . 126
17.47.2.31 fileExtension . . . 126
17.47.2.32 fileName . . . 127
17.47.2.33 fontsize . . . 127
17.47.2.34 GPS . . . 127
17.47.2.35 GPS_in_bound . . . 127
17.47.2.36 GPS_in_bound_st . . . 127
17.47.2.37 GPS_lat . . . 127
17.47.2.38 GPS_lon . . . 127
17.47.2.39 GPS_station . . . 127
17.47.2.40 gpsFile . . . 128
17.47.2.41 gpsLOS . . . 128
17.47.2.42 gpsLOS_ref . . . 128
17.47.2.43 GPSx . . . 128
17.47.2.44 GPSxx . . . 128
17.47.2.45 GPSy . . . 128
17.47.2.46 GPSyy . . . 128
17.47.2.47 gx . . . 128
17.47.2.48 gy . . . 129
17.47.2.49 h5file_theta . . . 129

17.47.2.50 hbound . . . 129
17.47.2.51 heading . . . 129
17.47.2.52 idx . . . 129
17.47.2.53 IDX . . . 129
17.47.2.54 idxRef . . . 129
17.47.2.55 IDXref . . . 129
17.47.2.56 IDY . . . 130
17.47.2.57 IDYref . . . 130
17.47.2.58 insarData . . . 130
17.47.2.59 lat . . . 130
17.47.2.60 Lat . . . 130
17.47.2.61 Lat0 . . . 130
17.47.2.62 Lat1 . . . 130
17.47.2.63 lat_all . . . 130
17.47.2.64 lat_step . . . 131
17.47.2.65 lat_transect . . . 131
17.47.2.66 lbound . . . 131
17.47.2.67 Length . . . 131
17.47.2.68 length . . . 131
17.47.2.69 linewidth . . . 131
17.47.2.70 lon . . . 131
17.47.2.71 Lon . . . 132
17.47.2.72 lon_all . . . 132
17.47.2.73 lon_step . . . 132
17.47.2.74 lon_transect . . . 132
17.47.2.75 m . . . 132
17.47.2.76 m1 . . . 132
17.47.2.77 m_prof_edge . . . 132
17.47.2.78 matFile . . . 132
17.47.2.79 mf . . . 133

17.47.2.80 mfc . . . . . 133
17.47.2.81 ms . . . . . 133
17.47.2.82 NoInSAR . . . . . 133
17.47.2.83 nrows . . . . . 133
17.47.2.84 Num_profiles . . . . . 133
17.47.2.85 Se . . . . . 133
17.47.2.86 Sn . . . . . 134
17.47.2.87 Stations . . . . . 134
17.47.2.88 stationsList . . . . . 134
17.47.2.89 stdInSAR . . . . . 134
17.47.2.90 theta . . . . . 134
17.47.2.91 transect . . . . . 134
17.47.2.92 transect_lat . . . . . 134
17.47.2.93 transect_lon . . . . . 134
17.47.2.94 unitVec . . . . . 135
17.47.2.95 Ve . . . . . 135
17.47.2.96 Vn . . . . . 135
17.47.2.97 Width . . . . . 135
17.47.2.98 x . . . . . 135
17.47.2.99 x0 . . . . . 135
17.47.2.100 X0 . . . . . 135
17.47.2.101 x1 . . . . . 135
17.47.2.102 X1 . . . . . 136
17.47.2.103 xc . . . . . 136
17.47.2.104 Xf0 . . . . . 136
17.47.2.105 Xf1 . . . . . 136
17.47.2.106 xlim . . . . . 136
17.47.2.107 XX0 . . . . . 136
17.47.2.108 y . . . . . 136
17.47.2.109 y0 . . . . . 136

17.47.2.110 Y0 . . . 137
17.47.2.111 y1 . . . 137
17.47.2.112 Y1 . . . 137
17.47.2.113 yc . . . 137
17.47.2.114 Yf0 . . . 137
17.47.2.115 Yf1 . . . 137
17.47.2.116 ydim . . . 137
17.47.2.117 YY0 . . . 137
17.47.2.118 z . . . 138
17.48 pysar.multilook Namespace Reference . . . 138
17.48.1 Function Documentation . . . 138
17.48.1.1 cmdLineParse() . . . 138
17.48.1.2 main() . . . 138
17.48.1.3 multilook_attribute() . . . 138
17.48.1.4 multilook_file() . . . 139
17.48.1.5 multilook_matrix() . . . 139
17.48.2 Variable Documentation . . . 139
17.48.2.1 EXAMPLE . . . 139
17.49 pysar.plot_atmDrop Namespace Reference . . . 139
17.49.1 Variable Documentation . . . 140
17.49.1.1 alpha . . . 140
17.49.1.2 ax1 . . . 140
17.49.1.3 ax2 . . . 140
17.49.1.4 bbox_inches . . . 140
17.49.1.5 c . . . 141
17.49.1.6 cbar . . . 141
17.49.1.7 dateList . . . 141
17.49.1.8 dateList6 . . . 141
17.49.1.9 dates . . . 141
17.49.1.10 datevector . . . 141

17.49.1.11 fig . . . 141
17.49.1.12 fl . . . 142
17.49.1.13 fontsize . . . 142
17.49.1.14 idxMean . . . 142
17.49.1.15 idxPix . . . 142
17.49.1.16 line_s . . . 142
17.49.1.17 lineNum . . . 142
17.49.1.18 lines . . . 142
17.49.1.19 meanList . . . 143
17.49.1.20 numProject . . . 143
17.49.1.21 offset . . . 143
17.49.1.22 pixList . . . 143
17.49.1.23 projectDir . . . 143
17.49.1.24 projectList . . . 143
17.49.1.25 s . . . 143
17.49.1.26 sc1 . . . 144
17.49.1.27 sc2 . . . 144
17.49.1.28 transparent . . . 144
17.49.1.29 vmax . . . 144
17.49.1.30 vmin . . . 144
17.50 pysar.plot_network Namespace Reference . . . 144
17.50.1 Function Documentation . . . 145
17.50.1.1 cmdLineParse() . . . 145
17.50.1.2 main() . . . 145
17.50.2 Variable Documentation . . . 145
17.50.2.1 BL_LIST . . . 145
17.50.2.2 DATE12_LIST . . . 145
17.50.2.3 EXAMPLE . . . 146
17.51 pysar.pysar2insarmaps Namespace Reference . . . 146
17.51.1 Function Documentation . . . 146

17.51.1.1 build_parser() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 146
17.51.1.2 get_H5_filename() . . . . . . . . . . . . . . . . . . . . . . . . . . 146
17.51.1.3 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 146
17.51.1.4 project_name_from_path() . . . . . . . . . . . . . . . . . . . . . . 147
17.51.1.5 rev_sorted_ls() . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
17.51.1.6 sorted_ls() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
17.52pysar.pysarApp Namespace Reference . . . . . . . . . . . . . . . . . . . . . 147
17.52.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . 147
17.52.1.1 check_geocode_file() . . . . . . . . . . . . . . . . . . . . . . . . 148
17.52.1.2 check_isfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . 148
17.52.1.3 check_subset_file() . . . . . . . . . . . . . . . . . . . . . . . . . 148
17.52.1.4 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . 148
17.52.1.5 create_subset_dataset() . . . . . . . . . . . . . . . . . . . . . . . 148
17.52.1.6 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
17.52.1.7 subset_dataset() . . . . . . . . . . . . . . . . . . . . . . . . . . 149
17.52.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . 149
17.52.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
17.52.2.2 LOGO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
17.52.2.3 TEMPLATE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
17.52.2.4 UM_FILE_STRUCT . . . . . . . . . . . . . . . . . . . . . . . . . . 150
17.53pysar.pysarApp_cmd Namespace Reference . . . . . . . . . . . . . . . . . . . 150
17.53.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . 151
17.53.1.1 check_geocode_file() . . . . . . . . . . . . . . . . . . . . . . . . 151
17.53.1.2 check_isfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
17.53.1.3 check_subset_file() . . . . . . . . . . . . . . . . . . . . . . . . . 151
17.53.1.4 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
17.53.1.5 create_subset_dataset() . . . . . . . . . . . . . . . . . . . . . . . 152
17.53.1.6 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
17.53.1.7 subset_dataset() . . . . . . . . . . . . . . . . . . . . . . . . . . 152
17.53.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . 152

17.53.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
17.53.2.2 LOGO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
17.53.2.3 TEMPLATE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
17.53.2.4 UM_FILE_STRUCT . . . . . . . . . . . . . . . . . . . . . . . . . . 154
17.54pysar.pysarApp_orig Namespace Reference . . . . . . . . . . . . . . . . . . . . . . 154
17.54.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
17.54.1.1 check_geocode() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
17.54.1.2 check_mask() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
17.54.1.3 check_subset() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
17.54.1.4 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
17.54.1.5 find_filename() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
17.54.1.6 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
17.54.1.7 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
17.55pysar.quality_map Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . 156
17.55.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
17.55.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
17.55.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
17.56pysar.reconstruct_igrams Namespace Reference . . . . . . . . . . . . . . . . . . . . 156
17.56.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
17.56.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
17.56.1.2 reconstruct_igrams_from_timeseries() . . . . . . . . . . . . . . . . . . 157
17.56.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
17.57pysar.reference_epoch Namespace Reference . . . . . . . . . . . . . . . . . . . . . . 157
17.57.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
17.57.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
17.57.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
17.57.1.3 yymmdd2yyyymmdd() . . . . . . . . . . . . . . . . . . . . . . . . . . 158
17.58pysar.remove_dates Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . 158
17.58.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
17.58.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158

17.58.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
17.59pysar.remove_plane Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
17.59.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
17.59.1.1 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.59.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.59.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.59.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.60pysar.rewrap Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.60.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.60.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
17.60.1.2 rewrap() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.60.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.61pysar.save_gmt Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.61.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.61.1.1 get_geo_lat_lon() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.61.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.61.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
17.62pysar.save_kml Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
17.62.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
17.62.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
17.62.1.2 rewrap() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
17.62.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
17.63pysar.save_unavco Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
17.63.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
17.63.1.1 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
17.63.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
17.63.1.3 metadata_pysar2unavco() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
17.63.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
17.63.2.1 CPX_ZERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
17.63.2.2 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162

17.63.2.3 FLOAT_ZERO . . . . . 163
17.63.2.4 INT_ZERO . . . . . 163
17.64 pysar.save_unw Namespace Reference . . . . . 163
17.64.1 Function Documentation . . . . . 163
17.64.1.1 main() . . . . . 163
17.64.1.2 usage() . . . . . 163
17.65 pysar.seed_data Namespace Reference . . . . . 163
17.65.1 Function Documentation . . . . . 164
17.65.1.1 cmdLineParse() . . . . . 164
17.65.1.2 main() . . . . . 164
17.65.1.3 manual_select_reference_yx() . . . . . 164
17.65.1.4 nearest() . . . . . 164
17.65.1.5 print_warning() . . . . . 164
17.65.1.6 random_select_reference_yx() . . . . . 165
17.65.1.7 read_seed_reference2inps() . . . . . 165
17.65.1.8 read_seed_template2inps() . . . . . 165
17.65.1.9 seed_attributes() . . . . . 165
17.65.1.10 seed_file_inps() . . . . . 165
17.65.1.11 seed_file_reference_value() . . . . . 166
17.65.1.12 select_max_coherence_yx() . . . . . 166
17.65.1.13 usage() . . . . . 166
17.66 pysar.simulation Namespace Reference . . . . . 166
17.66.1 Function Documentation . . . . . 166
17.66.1.1 main() . . . . . 166
17.66.1.2 usage() . . . . . 166
17.67 pysar.spatial_average Namespace Reference . . . . . 167
17.67.1 Function Documentation . . . . . 167
17.67.1.1 cmdLineParse() . . . . . 167
17.67.1.2 main() . . . . . 167
17.67.2 Variable Documentation . . . . . 167

17.67.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
17.68 pysar.subset Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
17.68.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
17.68.1.1 box_geo2pixel() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
17.68.1.2 box_pixel2geo() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
17.68.1.3 check_box_within_data_coverage() . . . . . . . . . . . . . . . . . . . . . . 169
17.68.1.4 cmdLineParse() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
17.68.1.5 coord_geo2radar() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
17.68.1.6 coord_radar2geo() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
17.68.1.7 get_box_overlap_index() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
17.68.1.8 get_coverage_box() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170
17.68.1.9 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170
17.68.1.10 read_subset_template2box() . . . . . . . . . . . . . . . . . . . . . . . . . 170
17.68.1.11 subset_attribute() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170
17.68.1.12 subset_box2inps() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 171
17.68.1.13 subset_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 171
17.68.1.14 subset_input_dict2box() . . . . . . . . . . . . . . . . . . . . . . . . . . . 171
17.68.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
17.68.2.1 EXAMPLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
17.69 pysar.sum_epochs Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
17.69.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
17.69.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
17.69.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
17.70 pysar.temporal_average Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . 173
17.70.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
17.70.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
17.70.1.2 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
17.71 pysar.temporal_coherence Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . 173
17.71.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
17.71.1.1 date_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173

17.71.1.2 design_matrix() . . . . . 174
17.71.1.3 main() . . . . . 174
17.71.1.4 usage() . . . . . 174
17.72pysar.temporal_derivative Namespace Reference . . . . . 174
17.72.1 Function Documentation . . . . . 174
17.72.1.1 main() . . . . . 174
17.72.1.2 usage() . . . . . 174
17.73pysar.timeseries2velocity Namespace Reference . . . . . 175
17.73.1 Function Documentation . . . . . 175
17.73.1.1 cmdLineParse() . . . . . 175
17.73.1.2 main() . . . . . 175
17.73.1.3 update_inps_from_template() . . . . . 175
17.73.1.4 yyyymmdd2years() . . . . . 175
17.73.2 Variable Documentation . . . . . 176
17.73.2.1 DROP_DATE_TXT . . . . . 176
17.73.2.2 EXAMPLE . . . . . 176
17.73.2.3 TEMPLATE . . . . . 176
17.74pysar.transect Namespace Reference . . . . . 177
17.74.1 Function Documentation . . . . . 177
17.74.1.1 cmdLineParse() . . . . . 177
17.74.1.2 get_scale_from_disp_unit() . . . . . 177
17.74.1.3 main() . . . . . 177
17.74.1.4 manual_select_start_end_point() . . . . . 178
17.74.1.5 read_lonlat_file() . . . . . 178
17.74.1.6 transect_lalo() . . . . . 178
17.74.1.7 transect_list() . . . . . 178
17.74.1.8 transect_yx() . . . . . 179
17.74.2 Variable Documentation . . . . . 179
17.74.2.1 EXAMPLE . . . . . 179
17.75pysar.transect_legacy Namespace Reference . . . . . 179

17.75.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182
17.75.1.1 check_st_in_box() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182
17.75.1.2 check_st_in_box2() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183
17.75.1.3 dist_point_from_line() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183
17.75.1.4 dms2d() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183
17.75.1.5 find_row_column() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183
17.75.1.6 get_intersect() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
17.75.1.7 get_lat_lon() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
17.75.1.8 get_transect() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
17.75.1.9 gps_to_LOS() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
17.75.1.10 line() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
17.75.1.11 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
17.75.1.12 nanmean() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
17.75.1.13 nanstd() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
17.75.1.14 nearest() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
17.75.1.15 onclick() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
17.75.1.16 readGPSfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
17.75.1.17 redGPSfile() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.1.18 redGPSfile_cmm4() . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.1.19 Usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.2.1 alpha . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.2.2 avgInSAR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.2.3 ax . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.2.4 axes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
17.75.2.5 axes2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
17.75.2.6 c . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
17.75.2.7 c_prof_edge . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
17.75.2.8 cf . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
17.75.2.9 check_result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187

17.75.2.10 check_result2 . . . . . 187
17.75.2.11 cid . . . . . 187
17.75.2.12 color . . . . . 188
17.75.2.13 D . . . . . 188
17.75.2.14 dataset . . . . . 188
17.75.2.15 df0 . . . . . 188
17.75.2.16 df0_km . . . . . 188
17.75.2.17 df1 . . . . . 188
17.75.2.18 dg . . . . . 188
17.75.2.19 DistGPS . . . . . 189
17.75.2.20 dset . . . . . 189
17.75.2.21 dx . . . . . 189
17.75.2.22 DX . . . . . 189
17.75.2.23 dy . . . . . 189
17.75.2.24 DY . . . . . 189
17.75.2.25 earth_radius . . . . . 189
17.75.2.26 fault_loc . . . . . 189
17.75.2.27 FaultLine . . . . . 190
17.75.2.28 fig . . . . . 190
17.75.2.29 fig2 . . . . . 190
17.75.2.30 figName . . . . . 190
17.75.2.31 fileExtension . . . . . 190
17.75.2.32 fileName . . . . . 190
17.75.2.33 fontsize . . . . . 190
17.75.2.34 GPS . . . . . 191
17.75.2.35 GPS_in_bound . . . . . 191
17.75.2.36 GPS_in_bound_st . . . . . 191
17.75.2.37 GPS_lat . . . . . 191
17.75.2.38 GPS_lon . . . . . 191
17.75.2.39 GPS_station . . . . . 191

17.75.2.40 gpsFile . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 191
17.75.2.41 gpsLOS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 191
17.75.2.42 gpsLOS_ref . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.43 GPSx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.44 GPSxx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.45 GPSy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.46 GPSyy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.47 gx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.48 gy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.49 h5file_theta . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
17.75.2.50 hbound . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.51 heading . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.52 idx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.53 IDX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.54 idxRef . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.55 IDXref . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.56 IDY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.57 IDYref . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
17.75.2.58 Info_aboutFault . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.59 insarData . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.60 Lat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.61 lat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.62 lat_all . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.63 lat_step . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.64 lat_transect . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
17.75.2.65 lbound . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
17.75.2.66 length . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
17.75.2.67 Length . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
17.75.2.68 linewidth . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
17.75.2.69 Lon . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195

17.75.2.70 lon . . . . . 195
17.75.2.71 lon_all . . . . . 195
17.75.2.72 lon_step . . . . . 196
17.75.2.73 lon_transect . . . . . 196
17.75.2.74 m . . . . . 196
17.75.2.75 m1 . . . . . 196
17.75.2.76 m_prof_edge . . . . . 196
17.75.2.77 matFile . . . . . 196
17.75.2.78 mf . . . . . 196
17.75.2.79 mfc . . . . . 196
17.75.2.80 mp . . . . . 197
17.75.2.81 ms . . . . . 197
17.75.2.82 NoInSAR . . . . . 197
17.75.2.83 nrows . . . . . 197
17.75.2.84 Se . . . . . 197
17.75.2.85 Sn . . . . . 197
17.75.2.86 Stations . . . . . 197
17.75.2.87 stationsList . . . . . 198
17.75.2.88 stdInSAR . . . . . 198
17.75.2.89 theta . . . . . 198
17.75.2.90 transect . . . . . 198
17.75.2.91 transect_lat . . . . . 198
17.75.2.92 transect_lon . . . . . 198
17.75.2.93 unitVec . . . . . 198
17.75.2.94 Ve . . . . . 198
17.75.2.95 Vn . . . . . 199
17.75.2.96 Width . . . . . 199
17.75.2.97 x . . . . . 199
17.75.2.98 x0 . . . . . 199
17.75.2.99 X0 . . . . . 199

17.75.2.100 x1 . . . 199
17.75.2.101 X1 . . . 199
17.75.2.102 xc . . . 199
17.75.2.103 XX0 . . . 200
17.75.2.104 y . . . 200
17.75.2.105 y0 . . . 200
17.75.2.106 Y0 . . . 200
17.75.2.107 y1 . . . 200
17.75.2.108 Y1 . . . 200
17.75.2.109 yc . . . 200
17.75.2.110 ydim . . . 200
17.75.2.111 YY0 . . . 201
17.75.2.112 z . . . 201
17.76 pysar.tropcor_phase_elevation Namespace Reference . . . 201
17.76.1 Function Documentation . . . 201
17.76.1.1 main() . . . 201
17.76.1.2 usage() . . . 201
17.77 pysar.tropcor_pyaps Namespace Reference . . . 201
17.77.1 Function Documentation . . . 202
17.77.1.1 closest_weather_product_time() . . . 202
17.77.1.2 cmdLineParse() . . . 202
17.77.1.3 get_delay() . . . 202
17.77.1.4 main() . . . 202
17.77.2 Variable Documentation . . . 202
17.77.2.1 EXAMPLE . . . 203
17.77.2.2 REFERENCE . . . 203
17.77.2.3 TEMPLATE . . . 203
17.78 pysar.tsview_mli Namespace Reference . . . 203
17.78.1 Function Documentation . . . 204
17.78.1.1 main() . . . 204

17.78.1.2 transect_lalo() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 204
17.78.1.3 transect_list() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 204
17.78.1.4 transect_yx() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 204
17.78.1.5 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
17.79pysar.tsviewer Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . 205
17.79.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
17.79.1.1 check_yx() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
17.79.1.2 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
17.79.1.3 read_dis_lalo() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
17.79.1.4 read_dis_xy() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
17.79.1.5 update_lim() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
17.79.1.6 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
17.80pysar.unavco2insarmaps Namespace Reference . . . . . . . . . . . . . . . . . . . . 206
17.80.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
17.80.1.1 build_parser() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
17.80.1.2 convert_data() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
17.80.1.3 get_date() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
17.80.1.4 get_decimal_date() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
17.80.1.5 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
17.80.1.6 make_json_file() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
17.80.2 Variable Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
17.80.2.1 dbHost . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
17.80.2.2 dbPassword . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
17.80.2.3 dbUsername . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
17.81pysar.unwrap_error Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . 208
17.81.1 Function Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
17.81.1.1 main() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 209
17.81.1.2 phase_bonding() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 209
17.81.1.3 usage() . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 209
17.82pysar.view Namespace Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . 209

17.82.1 Function Documentation . . . 210
17.82.1.1 add_inner_title() . . . 210
17.82.1.2 auto_figure_title() . . . 210
17.82.1.3 auto_flip_direction() . . . 210
17.82.1.4 auto_row_col_num() . . . 211
17.82.1.5 check_colormap_input() . . . 211
17.82.1.6 check_multilook_input() . . . 211
17.82.1.7 cmdLineParse() . . . 211
17.82.1.8 get_epoch_full_list_from_input() . . . 211
17.82.1.9 main() . . . 212
17.82.1.10 plot_dem_lalo() . . . 212
17.82.1.11 plot_dem_yx() . . . 212
17.82.1.12 plot_matrix() . . . 213
17.82.1.13 round_to_1() . . . 213
17.82.1.14 scale_data2disp_unit() . . . 213
17.82.1.15 update_matrix_with_plot_inps() . . . 214
17.82.1.16 update_plot_inps_with_display_setting_file() . . . 214
17.82.1.17 update_plot_inps_with_meta_dict() . . . 214
17.82.2 Variable Documentation . . . 214
17.82.2.1 EXAMPLE . . . 214
17.82.2.2 PLOT_TEMPLATE . . . 215
17.83 pysar.view_legacy Namespace Reference . . . 215
17.83.1 Function Documentation . . . 215
17.83.1.1 add_inner_title() . . . 215
17.83.1.2 auto_flip_check() . . . 216
17.83.1.3 main() . . . 216
17.83.1.4 orbit_direction() . . . 216
17.83.1.5 plot_dem_lalo() . . . 216
17.83.1.6 plot_dem_yx() . . . 216
17.83.1.7 rewrap() . . . 217
17.83.1.8 unit_and_scale() . . . 217
17.83.1.9 unit_type() . . . 217
17.83.1.10 usage() . . . 217
17.84 troposphere_uncertainty Namespace Reference . . . 217
17.84.1 Function Documentation . . . 217
17.84.1.1 cmdLineParse() . . . 218
17.84.1.2 createParser() . . . 218
17.84.1.3 download() . . . 218
17.84.1.4 estimate_seasonal() . . . 218
17.84.1.5 main() . . . 218
17.84.1.6 statistics() . . . 218
17.84.1.7 velocity_uncertainty() . . . 218
17.84.1.8 velocity_uncertainty_vs_distance() . . . 218

**18 Class Documentation** 219

18.1 Basemap2 Class Reference . . . 219

18.1.1 Detailed Description . . . 220

18.1.2 Member Function Documentation . . . 220

18.1.2.1 drawscale() . . . 220

18.2 BasicHTTP Class Reference . . . 221

18.2.1 Detailed Description . . . 221

18.2.2 Member Function Documentation . . . 221

18.2.2.1 get() . . . 221

18.3 InsarDatabaseController Class Reference . . . 222

18.3.1 Detailed Description . . . 223

18.3.2 Constructor & Destructor Documentation . . . 223

18.3.2.1 __init__() . . . 223

18.3.3 Member Function Documentation . . . 223

18.3.3.1 add_attribute() . . . 223

18.3.3.2 attribute_exists_for_dataset() . . . 223

18.3.3.3 close() . . . 223

18.3.3.4 connect() . . . 224

18.3.3.5 get_dataset_id() . . . 224

18.3.3.6 get_dataset_names() . . . 224

18.3.3.7 index_table_on() . . . 224

18.3.3.8 table_exists() . . . 224

18.3.4 Member Data Documentation . . . 224

18.3.4.1 con . . . 224

18.3.4.2 cursor . . . 225

18.3.4.3 db . . . 225

18.3.4.4 host . . . 225

18.3.4.5 password . . . 225

18.3.4.6 username . . . 225

18.4 timeseries Class Reference . . . 226

18.4.1 Detailed Description . . . . . 228
18.4.2 Constructor & Destructor Documentation . . . . . 228
18.4.2.1 __init__() . . . . . 228
18.4.3 Member Function Documentation . . . . . 228
18.4.3.1 close() . . . . . 228
18.4.3.2 distance() . . . . . 228
18.4.3.3 estimate_seasonal() . . . . . 229
18.4.3.4 load() . . . . . 229
18.4.3.5 open() . . . . . 229
18.4.3.6 sample() . . . . . 229
18.4.3.7 statistics() . . . . . 229
18.4.3.8 std_timeseries() . . . . . 229
18.4.3.9 std_velocity() . . . . . 230
18.4.3.10 uncertainty_vs_distance() . . . . . 230
18.4.4 Member Data Documentation . . . . . 230
18.4.4.1 cols . . . . . 230
18.4.4.2 Data . . . . . 230
18.4.4.3 dateList . . . . . 230
18.4.4.4 dist . . . . . 230
18.4.4.5 file . . . . . 230
18.4.4.6 h5 . . . . . 231
18.4.4.7 idx . . . . . 231
18.4.4.8 lat . . . . . 231
18.4.4.9 lat_first . . . . . 231
18.4.4.10 lat_step . . . . . 231
18.4.4.11 lon . . . . . 231
18.4.4.12 lon_first . . . . . 231
18.4.4.13 lon_step . . . . . 231
18.4.4.14 numDates . . . . . 232
18.4.4.15 numPixels . . . . . 232
18.4.4.16 relative_std . . . . . 232
18.4.4.17 relative_std_velocity . . . . . 232

**19 File Documentation 233**

19.1 /Users/jeromezhang/Documents/development/python/PySAR.wiki/_Sidebar.md File Reference . . . 233

19.2 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Attributes.md File Reference . . 233

19.3 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Coordinate.md File Reference . 233

19.4 /Users/jeromezhang/Documents/development/python/PySAR.wiki/File-Descriptions.md File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 233

19.5 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Gamma-File-Decription.md File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 233

19.6 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Google-Earth.md File Reference 233

19.7 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Home.md File Reference . . . . 233

19.8 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Mask.md File Reference . . . . 233

19.9 /Users/jeromezhang/Documents/development/python/PySAR.wiki/pysarApp.md File Reference . . 234

19.10/Users/jeromezhang/Documents/development/python/PySAR.wiki/UNAVCO-InSAR-Archive.md File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 234

19.11/Users/jeromezhang/Documents/development/python/PySAR.wiki/Web-Viewer.md File Reference . 234

19.12/Users/jeromezhang/Documents/development/python/PySAR/pysar/__init__.py File Reference . . . 234

19.13/Users/jeromezhang/Documents/development/python/PySAR/pysar/_datetime.py File Reference . . 234

19.14/Users/jeromezhang/Documents/development/python/PySAR/pysar/_gmt.py File Reference . . . . 235

19.15/Users/jeromezhang/Documents/development/python/PySAR/pysar/_network.py File Reference . . 235

19.16/Users/jeromezhang/Documents/development/python/PySAR/pysar/_pysar_utilities.py File Reference 235

19.17/Users/jeromezhang/Documents/development/python/PySAR/pysar/_readfile.py File Reference . . 236

19.18/Users/jeromezhang/Documents/development/python/PySAR/pysar/_remove_surface.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 237

19.19/Users/jeromezhang/Documents/development/python/PySAR/pysar/_writefile.py File Reference . . 237

19.20/Users/jeromezhang/Documents/development/python/PySAR/pysar/add.py File Reference . . . . . 238

19.21/Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attribute.py File Reference 238

19.22/Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attributes_insarmaps.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 238

19.23/Users/jeromezhang/Documents/development/python/PySAR/pysar/asc_desc.py File Reference . . 239

19.24/Users/jeromezhang/Documents/development/python/PySAR/pysar/baseline_error.py File Reference 239

19.25/Users/jeromezhang/Documents/development/python/PySAR/pysar/baseline_trop.py File Reference 239

19.26/Users/jeromezhang/Documents/development/python/PySAR/pysar/convert2mat.py File Reference 240

19.27/Users/jeromezhang/Documents/development/python/PySAR/pysar/correct_dem.py File Reference 240

19.28/Users/jeromezhang/Documents/development/python/PySAR/pysar/correlation_with_dem.py File Reference . . . 240

19.29/Users/jeromezhang/Documents/development/python/PySAR/pysar/dem_error.py File Reference . 241

19.30/Users/jeromezhang/Documents/development/python/PySAR/pysar/diff.py File Reference . . . . . 241

19.31/Users/jeromezhang/Documents/development/python/PySAR/pysar/drop_turbulence.py File Reference . . . 241

19.32/Users/jeromezhang/Documents/development/python/PySAR/pysar/filter_spatial.py File Reference 242

19.33/Users/jeromezhang/Documents/development/python/PySAR/pysar/filter_temporal.py File Reference 242

19.34/Users/jeromezhang/Documents/development/python/PySAR/pysar/gamma_view.py File Reference 242

19.35/Users/jeromezhang/Documents/development/python/PySAR/pysar/generate_mask.py File Reference 242

19.36/Users/jeromezhang/Documents/development/python/PySAR/pysar/geocode.py File Reference . . 243

19.37/Users/jeromezhang/Documents/development/python/PySAR/pysar/igram_closure.py File Reference 243

19.38/Users/jeromezhang/Documents/development/python/PySAR/pysar/igram_inversion.py File Reference . . . 243

19.39/Users/jeromezhang/Documents/development/python/PySAR/pysar/image_math.py File Reference 244

19.40/Users/jeromezhang/Documents/development/python/PySAR/pysar/incidence_angle.py File Reference . . . 244

19.41/Users/jeromezhang/Documents/development/python/PySAR/pysar/info.py File Reference . . . . . 244

19.42/Users/jeromezhang/Documents/development/python/PySAR/pysar/insar_vs_gps.py File Reference 245

19.43/Users/jeromezhang/Documents/development/python/PySAR/pysar/insarmaps_query.py File Reference . . . 247

19.44/Users/jeromezhang/Documents/development/python/PySAR/pysar/l1.py File Reference . . . . . . 247

19.45/Users/jeromezhang/Documents/development/python/PySAR/pysar/load_data.py File Reference . . 247

19.46/Users/jeromezhang/Documents/development/python/PySAR/pysar/load_dem.py File Reference . . 248

19.47/Users/jeromezhang/Documents/development/python/PySAR/pysar/lod.py File Reference . . . . . 248

19.48/Users/jeromezhang/Documents/development/python/PySAR/pysar/look_angle.py File Reference . 249

19.49/Users/jeromezhang/Documents/development/python/PySAR/pysar/los2enu.py File Reference . . . 249

19.50/Users/jeromezhang/Documents/development/python/PySAR/pysar/mask.py File Reference . . . . 249

19.51/Users/jeromezhang/Documents/development/python/PySAR/pysar/match.py File Reference . . . . 250

19.52/Users/jeromezhang/Documents/development/python/PySAR/pysar/mean_spatial.py File Reference 250

19.53/Users/jeromezhang/Documents/development/python/PySAR/pysar/modify_network.py File Reference . . . 250

19.54/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/delayTimeseries.py File Reference . . . 251
19.55/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/dloadUtil.py File Reference 251
19.56/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/get_modis_v3.py File Reference . . . 252
19.57/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/troposphere_uncertainty.py File Reference . . . 252
19.58/Users/jeromezhang/Documents/development/python/PySAR/pysar/multi_transect.py File Reference 252
19.59/Users/jeromezhang/Documents/development/python/PySAR/pysar/multilook.py File Reference . . 255
19.60/Users/jeromezhang/Documents/development/python/PySAR/pysar/plot/plot_tropcor_phase_↩elevation.py File Reference . . . 256
19.61/Users/jeromezhang/Documents/development/python/PySAR/pysar/plot_atmDrop.py File Reference 257
19.62/Users/jeromezhang/Documents/development/python/PySAR/pysar/plot_network.py File Reference 257
19.63/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysar2insarmaps.py File Reference . . . 258
19.64/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp.py File Reference . . 258
19.65/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp_cmd.py File Reference 259
19.66/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp_orig.py File Reference 259
19.67/Users/jeromezhang/Documents/development/python/PySAR/pysar/quality_map.py File Reference 260
19.68/Users/jeromezhang/Documents/development/python/PySAR/pysar/reconstruct_igrams.py File Reference . . . 260
19.69/Users/jeromezhang/Documents/development/python/PySAR/pysar/reference_epoch.py File Reference . . . 261
19.70/Users/jeromezhang/Documents/development/python/PySAR/pysar/remove_dates.py File Reference 261
19.71/Users/jeromezhang/Documents/development/python/PySAR/pysar/remove_plane.py File Reference 261
19.72/Users/jeromezhang/Documents/development/python/PySAR/pysar/rewrap.py File Reference . . . 262
19.73/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_gmt.py File Reference . . 262
19.74/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_kml.py File Reference . . 262
19.75/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_unavco.py File Reference 263
19.76/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_unw.py File Reference . . 263
19.77/Users/jeromezhang/Documents/development/python/PySAR/pysar/seed_data.py File Reference . 263
19.78/Users/jeromezhang/Documents/development/python/PySAR/pysar/simulation.py File Reference . 264
19.79/Users/jeromezhang/Documents/development/python/PySAR/pysar/spatial_average.py File Reference . . . 264

19.80/Users/jeromezhang/Documents/development/python/PySAR/pysar/subset.py File Reference . . . 265
19.81/Users/jeromezhang/Documents/development/python/PySAR/pysar/sum_epochs.py File Reference 265
19.82/Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_average.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 266
19.83/Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_coherence.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 266
19.84/Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_derivative.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 266
19.85/Users/jeromezhang/Documents/development/python/PySAR/pysar/timeseries2velocity.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 267
19.86/Users/jeromezhang/Documents/development/python/PySAR/pysar/transect.py File Reference . . . 267
19.87/Users/jeromezhang/Documents/development/python/PySAR/pysar/transect_legacy.py File Reference 268
19.88/Users/jeromezhang/Documents/development/python/PySAR/pysar/tropcor_phase_elevation.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 270
19.89/Users/jeromezhang/Documents/development/python/PySAR/pysar/tropcor_pyaps.py File Reference 271
19.90/Users/jeromezhang/Documents/development/python/PySAR/pysar/tsview_mli.py File Reference . 271
19.91/Users/jeromezhang/Documents/development/python/PySAR/pysar/tsviewer.py File Reference . . . 272
19.92/Users/jeromezhang/Documents/development/python/PySAR/pysar/unavco2insarmaps.py File Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 272
19.93/Users/jeromezhang/Documents/development/python/PySAR/pysar/unwrap_error.py File Reference 272
19.94/Users/jeromezhang/Documents/development/python/PySAR/pysar/view.py File Reference . . . . . 273
19.95/Users/jeromezhang/Documents/development/python/PySAR/pysar/view_legacy.py File Reference 274
19.96/Users/jeromezhang/Documents/development/python/PySAR/README.md File Reference . . . . . 274

# Chapter 1

# Welcome to PySAR!

PySAR is an InSAR (Interferometric Synthetic Aperture Radar) time series package to produce three dimensional (space and time) ground displacement from InSAR data. To use the package add the path to PySAR directory to your $PYTHONPATH and add PySAR/pysar to your $path

Depending on your shell you may use commands such as the following examples to setup pysar:

Using bash: export PYTHONPATH=/nethome/hfattahi/development/PySAR:${PYTHONPATH} export PATH="/nethome/hfattahi/development/PySAR/pysar:$PATH" export TSSARDIR=/nethome/timeseries/

Using csh: setenv PYTHONPATH "/nethome/hfattahi/development/PySAR" set path = (/nethome/hfattahi/development/PySAR/pysar $path) setenv TSSARDIR "/nethome/timeseries/"

Run pysarApp.py to see the examples of processing options.

The current version of PySAR is compatible with roi_pac outputs. pysar reads unwrapped interefrograms (at the same coordinate system: radar or geo) and the baseline files for each interefrogram. You need to give the path to where the interferograms are and pysar takes care of the rest!

Run pysarApp.py to see examples of processing options.

How to run pysar:

When you have a stack of interferograms processed with roi_pac, make a pysar processing file (a text file) in your shell using for example vi or any other text editor:

eg: vi YourProjectName.template

and include the following pysar processing options in your template:

pysar.inputdata=/scratch/hfattahi/PROCESS/SanAndreasT356EnvD/DONE/IFG∗/filt∗0∗c10.unw pysar.CorFiles = /scratch/hfattahi/PROCESS/SanAndreasT356EnvD/DONE/IFG∗/filt∗0∗.cor pysar.wrapped = /scratch/hfattahi/PROCESS/SanAndreasT356EnvD/DONE/IFG∗/filt∗0∗.int pysar.geomap = /scratch/hfattahi/PROCESS/SanAndreasT356EnvD/GEO/geomap_12/geomap_8rlks.trans pysar.dem = /scratch/hfattahi/PROCESS/SanAndreasT356EnvD/DONE/IFG_20050102_20070809/radar_8lks.hgt pysar.topo_error = yes # [no] pysar.orbit_error = yes # [np] pysar.orbit_error.method = plane #['quadratic', 'plane', 'quardatic_range', 'quadratic_azimiuth', 'plane_range', 'plane_azimuth','baselineCor','BaseTropCor'] pysar.mask=yes pysar.mask.threshold = 0.7

Save your template file and run pysar as: pysarApp.py YourProjectName.template

pysar reads the unwrapped interferograms, refernces all of them to the same coherent pixel (a seed point point), calculates the phase closure and estimates the unwrapping errors (if it has been asked for), inverts the interferograms, calculates a parameter called "temporal_coherence" which can be used to evaluate the quality of inversion, removes ramps or surface from time-series epochs, corrects dem errors, corrects local oscilator drift (for Envisat only), corrects stratified tropospheric delay (using pyaps and using phase-elevation approach), ... and finally estimates the velocity.

use view.py to view any pysar output. use tsviewer.py to plot the time-series for each point (relative to the refernce point and epoch!).

You may need to install some more packages including, pyaps, pykml, GDAL to get full advantage of PySAR. Basic time-series analysis does not need these packages though. However you need python with numpy, scipy, h5py and matplotlib installed.

pykml installation:

website:http://pythonhosted.org/pykml/

wget `https://pypi.python.org/packages/source/p/pykml/pykml-0.1.0.tar.gz` tar -xvf pykml-0.1.0.tar.gz cd pykml-0.1.0 easy_install pykml

GDAL installation: %%%%%%%%%%% wget `ftp://ftp.remotesensing.org/gdal/gdal-1.9.1.tar.gz`

./configure –with-python –prefix=/nethome/hfattahi/development/utilities/gdal-1.9.1 make make install

%%%%%%%%%%% setenv GDALHOME /nethome/hfattahi/development/utilities/gdal-1.9.1 set path= ( $path $GDALHOME/bin ) setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${GDALHOME}/lib

PySAR uses cvxopt-1.1.6 for L1 norm minimization See `http://cvxopt.org` to download and installation

This package is used if user choose to use L1 norm minimization for inversion of interferograms or to estimate the velocity field.

link to download: `https://github.com/cvxopt/cvxopt/archive/1.1.6.tar.gz`

To install: Untar the package cd cvxopt-1.1.6 python setup.py install

# Chapter 2

# _Sidebar

**Wiki**

- `Home`
- `pysarApp`
- `File Description`
- `Attributes`
- `Coordinate`

**Output**

- `Google Earth`
- `UNAVCO`
- `Web Viewer`

# Chapter 3

# Attributes

PySAR mainly use attribute name from ROI_PAC, with some additional attributes generated by PySAR itself.

### `ROI_PAC attribute` used in PySAR:

If using ROI_PAC as InSAR processor, both "baseline parameter RSC" file (i.e. *100416-100901_baseline.rsc*) and basic metadata file (i.e. *filt_100416-100901-sim_HDR_4rlks_c10.unw.rsc*) will be imported into PySAR.

- FILE_LENGTH = number of rows
- WIDTH = number of columns
- X/Y_STEP = Ground resolution in degree in Longitude/latitude direction, for geocoded product
- X/Y_FIRST = Longitude/latitude in degree of the first pixel - Upper left corner, for geocoded product
- WAVELENGTH = Radar wavelength (m)
- RANGE_PIXEL_SIZE = Slant range pixel size (search for pixel_ratio to convert to ground size, in m), used in dem_error, incidence_angle, multilook, transect.
- AZIMUTH_PIXEL_SIZE = Azimuth pixel size at orbital altitude (multiply by Re/(Re+h) for ground size (m), where Re is the local earth radius), used in baseline_error/trop and multilook.
- EARTH_RADIUS = Best fitting spheroid radius (m), used in dem_error, incidence_angle, convert2mat
- CENTER_LINE_UTC = Time at middle of interferogram (seconds)
- HEIGHT = Height of satellite (m), used in dem_error, incidence_angle, convert2mat
- STARTING_RANGE = Distance from satellite to first ground pixel (m), used in incidence_angle calculation
- LOOK_REF1/2 = Look angle at corner 1/2 (degree), not accurate
- LAT/LON_REF1/2/3/4 = Latitude/longitude at corner 1/2/3/4 (degree), used in save_unavco, not accurate
- DATE12 = (date1)-(date2), master - slave date of interferogram in 6 digit number
- DATE = Date of master scene in 6 digit number
- PLATFORM = satellite/sensor name, used in Local Oscillator Drift correction for Envisat
- PRF = Pulse repetition frequency (Hz), used in save_unavco
- ANTENNA_SIDE = -1 for right looking radar

- HEADING = Spacecraft heading at peg point (degree), used in asc_desc, los2enu
- ORBIT_DIRECTION = ascending, or descending
- P_BASELINE_TOP_HDR = Perpendicular baseline at top of interferogram (m), used in _network, _pysar_↩ utilities
- P_BASELINE_BOTTOM_HDR = Perpendicular baseline at bottom of interferogram (m), used in _network, _pysar_utilities
- H_BASELINE_RATE_HDR = Rate of change of horizontal baseline as a function of line number (linear term), used in _pysar_utilities
- H_BASELINE_TOP_HDR = Horizontal baseline separation at the top of the interferogram calculated from orbital parameters, used in _pysar_utilities
- V_BASELINE_RATE_HDR = Linear term for vertical baseline change, used in _pysar_utilities
- V_BASELINE_TOP_HDR = Vertical baseline separation at top of the interferogram, used in _pysar_utilities

#### PySAR attribute:

- FILE_TYPE = file type, velocity, timeseries, interferograms, etc.; for non-HDF5 file, it's the file extension name.
- PROCESSOR = InSAR processor, i.e. isce, roipac, gamma
- P_BASELINE_TIMESERIES = timeseries of perpendicular baseline
- UNIT = data unit, i.e. m, m/yr, radian, and 1 for file without unit, such as coherence
- ref_x/y/lat/lon = column/row/latitude/longitude of reference point
- ref_date = reference date
- subest_x0/y0/x1/y1 = start/end column/row number of subset in the original coverage
- date1 = start time of dataset
- date2 = end time of dataset

#### Reference

Pritchard et al., (2014), Open-source software for geodetic imaging: ROI_PAC for InSAR and pixel trakcing, pp 44-48. PDF

## Chapter 4

# Coordinate

There are two coordination systems in PySAR: **radar coordinate** and **geo coordinate**. Geo coordinate is defined in WGS84 coordination for horizontal direction, and determined by the following `ROI_PAC attributes` in latitude and longitude. The following shows examples from *AlosAT422F650/geo_velocity.h5*:

```
X_FIRST    131.02409876
Y_FIRST    33.63756779
X_STEP     0.00033333
Y_STEP     -0.00033333
X_UNIT     degrees
Y_UNIT     degrees
```

X/Y_FIRST are the longitude/latitude value of the first (upper left corner) pixel's upper left corner, as shown below:

# Chapter 5

# File-Descriptions

PySAR use HDF5 file internally. It loads ROI_PAC file into .h5 file in the beginning and has the capability to output to UNAVCO hdf5 file, .grd file, ROI_PAC file and Google Earth KMZ file.

## HDF5 File Types

There are 3 types of HDF5 file structures used in PySAR:

- multi_group (**Ngroup-1dset-1atr**) = multiple groups with one dataset and one attribute dict per group i.e. interferograms, coherence, wrapped, snaphu_connect_component
- multi_dataset (**1group-Ndset-1atr**) = one group with multiple dataset and one attribute dict per group i.e. timeseries
- single_dataset (**1group-1dset-1atr**) = one group with one dataset and one attribute dict per group i.e. velocity, dem, rmse, temporal_coherence, mask

## Default File Names

### multi_group

- coherence.h5 = spatial coherence files loaded from ROI_PAC, generated in load_data step
- snaphuConnectComponent.h5 = multi_group type, mask of connect component files from SNAPHU phase unwrapping, loaded from ROI_PAC, generated in load_data step
- wrapIfgram.h5 = wrapped interferograms loaded from ROI_PAC, generated in load_data step
- unwrapIfgram.h5 = unwrapped interferograms loaded from ROI_PAC, generated in load_data step

### multi_dataset

- timeseries.h5 = multi_dataset type, time series displacement, generated in network inversion step

#### single_dataset

- average_spatial_coherence.h5 = temporal mean of all spatial coherence, generated from coherence.h5 in data loading step
- Mask.h5 = mask of non-zero amplitude pixels, generated from .unw file list in data loading step
- velocity.h5 = single_dataset type, Line-Of-Sight (LOS) velocity, generated in time series inversion step

#### ROI_PAC files

- geomap_*rlks.trans = ROI_PAC file, with inverse mapping transformation from radar to geo coordinates, check more `ROI_PAC File Descriptions`, copied in load_data step
- radar_*rlks.hgt = ROI_PAC DEM file in radar coordinate, check more `ROI_PAC File Descriptions`, copied in load_data step

### Prefixes

- geo_* = transformed from radar coord to geo coord using geocode.py
- Modified_* = network modification using modify_network.py
- subset_* = subset/crop in space using subset.py
- Seeded_* = referencing/seeding in space using seed_data.py

### Suffixes

- *_demCor = DEM error correction in time series domain
- *_ex = date(s) have been dropped
- *_ECMWF/MERRA/NARR = tropospheric correction using PyAPS, name is the weather re-analysis data used to estimate the tropospheric phase delay
- *_plane/quadratic/... = phase ramp removal
- *_refDate = referencing in time

# Chapter 6

# Gamma-File-Decription

**Basically, in GAMMA, we can name the file in any "nickname" if we want. But, there are also some common habits to name different type of files to make non-GAMMA guys readable, which is very similar like other softwares but not absolutely same.Here will introduce some common names of GAMMA-based files from SLC step to Unwrapping step.**

*ps: GAMMA software has several modules: MSP, ISP, DIFF&GEO, IPTA. MSP for focusing, ISP for interferometry, DIFF&GEO for DInSAR and gecoding, IPTA mainly for TS-InSAR (conventional PS and SBAS).*

∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗MSP∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗ (skipped here) ∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗↩ ISP∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗ ∗.slc (same thing as roi_pac) ∗.slc.par (parameters' file about orbit, width, length, time, ... But parameters in ∗.par file is far less than ∗.rsc file) ∗.mli (magnitude image of SLC after doing multilook) ∗.mli.par (same thing like ∗.slc.par, but width and length are changed due to multi-looking) ∗.rslc (co-registrated SLC, for TS-InSAR, usually coregistrated to one master image) ∗.rslc.par (parameter file of ∗.rslc, absolutely same as ∗.slc.par) ∗.rmli (co-registrated magnitude images of multi-looked SLC) ∗.rmli.par (parameter file of ...)

∗.off (offset file of co-registration, include fitted polynomial parameters, length, width, ...) ∗.offs (COMPLEX file, offset value in each chosen points, real and imaginary parts for Range and Azimuth offset) ∗.snr (std of co-registration in each point, which will be used to mask some points based on a threshold) ∗.offset ( text file of ∗.offs)

∗.coffs (COMPLEX file, culled offset of ∗.offs) ∗.coffsets (text type of ∗.coffs)

∗.base (baseline file) ∗.base.perp (perpendicular baseline file)

∗.cc (coherence map) ∗.int (original interferometry file, include every signal, flatten phase, DEM, Def, APS,...) ∗.flt ( "flatten" interferogram, after removing flatten signals from ∗.int) ∗.smcc (coherence map based on filtered interferogram) ∗.sm_flt (filtered ∗.flt interferogram)

∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗DIFF & GEOCODE∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗ ∗.diff (interferogram that has removed flatten signals and topography signals) ∗.flag (masked file based on coherence map, 0 and 1, only used for Branch-cut unwrapping )

∗.mask.ras (masked file for MCF unwrapping, also masked based on coherence) ∗.unw (unwrapped interferogram, usually unwrapped from ∗.diff , data type order is different from that of ROI_PAC's .unw file)

The same thing as ISP, all files based on filtering will include "sm", e.g., ∗sm.diff, ∗.sm.unw, but the final part of suffix will not change.

∗.htg (digital elevation model in radar coordinates) ∗.dem (..... in UTM coordinates) ∗.dem.par (parameters of ∗.dem file, which is in UTM coordinates, same as ∗.dem.rsc in ROI_PAC)

∗.utm_to_rdc (lookup table: from utm to radar coordinates)

## Chapter 7

# Google-Earth

asdfa

# Chapter 8

# Welcome to PySAR wiki!

Github Page: https://yunjunz.github.io/PySAR/ Google Group: https://groups.google.com/forum/#!forum/py-sar

Simple Tutorial: PDF

## Chapter 9

# Mask

Mask file is used in PySAR for DEM error estimation, phase ramp estimation, velocity inversion, etc. It use Mask.h5 file by default, or Modified_Mask.h5 if existed, or it can be specified in template option 'pysar.mask.file', the priority is:

script input $>$ template option $>$ Modified_Mask.h5 $>$ Mask.h5

# Chapter 10

# pysarApp

To run the default processing chain:

```
cd SanAndreasT356EnvD/PYSAR/
pysarApp.py SanAndreasT356EnvD.template
```

SanAndreasT356EnvD.template is a text file with option names and values. An example is shown below:

## Template

```
# Input Data (not needed for Miami user)
pysar.unwrapFiles    = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*.unw
pysar.corFiles       = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.cor
pysar.wrapFiles      = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.int     #optional
pysar.geomap         = /SanAndreasT356EnvD/PROCESS/GEO/*050102-070809*/geomap*.trans
pysar.dem.radarCoord = /SanAndreasT356EnvD/PROCESS/DONE/*050102-070809*/radar*.hgt
pysar.dem.geoCoord   = /SanAndreasT356EnvD/DEM/srtm1_30m.dem                   #optional

pysar.network.reference       = date12.list         #optional
pysar.network.coherenceBase   = yes                 #optional, auto for yes

pysar.subset.yx          = 1800:2000,700:800        #optional, auto/no/off for whole area
pysar.subset.lalo        = 31.5:32.5,130.5:131.0    #optional, auto/no/off for whole area

pysar.reference.yx       = 257 , 151                #optional, auto for max coherence selection
pysar.reference.lalo     = 31.8, 130.8              #optional, auto for max coherence selection
pysar.reference.date     = 20090120                 #optional, auto for the first date

pysar.troposphericDelay.method        = pyaps   #[height_correlation], auto for no tropospheric correction
pysar.troposphericDelay.polyOrder     = 1       #for height_correlation method
pysar.troposphericDelay.weatherModel  = ECMWF   #[ERA, MERRA, NARR], for pyaps method

pysar.topoError = yes               #[no], auto for yes
pysar.deramp    = plane             #[plane, quadratic, baseline_cor, base_trop_cor], auto for no
pysar.geocode   = yes               #[no], auto for yes
```

## pysarApp processing chain:

### 1. Data Loading and Preparation

- Load Data
- Subset (optional)
- Modify Network (optional)
- Reference in space
- Unwrapping Error Correction (optional)

#### 2. SBAS Network Inversion

- Time series Inversion
- Calculate Temporal Coherence
- Update Mask based on Temporal Coherence
- Calculate Incident Angle

#### 3. Phase Error Corrections

- Local Oscillator Drift (LOD) Correction (for Envisat)
- Tropospheric Delay Correction BaseTropCor Height-Correlation PyAPS: ECMWF, MERRA, NARR
- Topographic Residual (DEM error) Correction
- Deramp/Ramp Removal plane, quadratic, quadratic_range, quadratic_azimuth, plane_range, plane_azimuth; baselineCor BaseTropCor

#### 4. Linear Velocity Inversion

- Velocity Inversion

#### 5. Post-processing

- Geocode
- Mask

#### 6. Output

- Google Earth
- UNAVCO InSAR Archive

# Chapter 11

# UNAVCO-InSAR-Archive

Use the following commands to convert PySAR product into UNAVCO InSAR Archive format. All files should be geocoded in the same coordinations and resolution.

```
add_attribute.py timeseries.h5 add_attribute.txt
save_unavco.py timeseries.h5 -i incidence_angle.h5 -d dem.h5 -c temporal_coherence.h5 -m mask.h5
```

#### add_attribute.txt

Create an text file (i.e. *add_attribute.txt*) with the following attributes and manual modify them for your dataset.

```
##### UNAVCO Required Metadata
mission                = ALOS                 # ERS,ENV,S1,RS1,RS2,CSK,TSX,JERS,ALOS,ALOS2
beam_mode              = FB                   # S2,FB08,IW
beam_swath             = 70km
relative_orbit         = 422
#first_date            =                      # grab by script
#last_date             =                      # grab by script
#scene_footprint       =                      # grab by script
processing_type        = LOS_TIMESERIES
processing_software    = ROI_PAC
#history               =                      # grab by script

##### UNAVCO Recommended Metadata
frame                      = 650              # first frame number
#flight_direction          =                  # grab by script
#look_direction            =                  # grab by script
#polarization              =
#prf                       =                  # grab by script
#wavelength                =                  # grab by script
atmos_correct_method       = ERA-Interim
processing_dem             = GSI_DEHM_10m
unwrap_method              = SNAPHU
post_processing_method     = PySAR
#master_platform           =            #For INTERFEROGRAM products
#master_absolute_orbit     =            #For INTERFEROGRAM products
#master_doppler            =            #For INTERFEROGRAM products
#slave_platform            =            #For INTERFEROGRAM products
#slave_absolute_orbit      =            #For INTERFEROGRAM products
#slave_doppler             =            #For INTERFEROGRAM products
#percent_unwrapped         =
#average_coherence         =
#max_coherence             =
#percent_atmos_corrected =
#baseline_perp             =

##### INSARMAPS Metadata
reference    = 'Yunjun, Z., Amelung F., Aoki Y., (2016). Poster: A time series InSAR survey of volcanic deform
referencePdf = 'https://yunjunzhang.files.wordpress.com/2015/01/yunjun_2016_agu.pdf'
unavcoUrl    = ''
```

#### Reference


Baker, S., (2015), Product Format Specification of UNAVCO InSAR Product Archive DOC

## Chapter 12

# InSAR Time Series Web Viewer: http://insarmaps.rsmas.miami.edu

# Chapter 13

# Namespace Index

## 13.1 Packages

Here are the packages with brief descriptions (if available):


delayTimeseries . . . 33
dloadUtil . . . 34
get_modis_v3 . . . 35
plot_tropcor_phase_elevation . . . 36
pysar . . . 42
pysar._datetime . . . 43
pysar._gmt . . . 46
pysar._network . . . 46
pysar._pysar_utilities . . . 52
pysar._readfile . . . 59
pysar._remove_surface . . . 65
pysar._writefile . . . 66
pysar.add . . . 68
pysar.add_attribute . . . 68
pysar.add_attributes_insarmaps . . . 69
pysar.asc_desc . . . 70
pysar.baseline_error . . . 71
pysar.baseline_trop . . . 72
pysar.convert2mat . . . 73
pysar.correct_dem . . . 73
pysar.correlation_with_dem . . . 74
pysar.dem_error . . . 76
pysar.diff . . . 76
pysar.drop_turbulence . . . 77
pysar.filter_spatial . . . 78
pysar.filter_temporal . . . 79
pysar.gamma_view . . . 79
pysar.generate_mask . . . 80
pysar.geocode . . . 80
pysar.igram_closure . . . 82
pysar.igram_inversion . . . 83
pysar.image_math . . . 83
pysar.incidence_angle . . . 85
pysar.info . . . 85
pysar.insar_vs_gps . . . 86

pysar.insarmaps_query . . . 99
pysar.l1 . . . 99
pysar.load_data . . . 101
pysar.load_dem . . . 104
pysar.lod . . . 106
pysar.look_angle . . . 107
pysar.los2enu . . . 107
pysar.mask . . . 108
pysar.match . . . 109
pysar.mean_spatial . . . 111
pysar.modify_network . . . 112
pysar.multi_transect . . . 115
pysar.multilook . . . 138
pysar.plot_atmDrop . . . 139
pysar.plot_network . . . 144
pysar.pysar2insarmaps . . . 146
pysar.pysarApp . . . 147
pysar.pysarApp_cmd . . . 150
pysar.pysarApp_orig . . . 154
pysar.quality_map . . . 156
pysar.reconstruct_igrams . . . 156
pysar.reference_epoch . . . 157
pysar.remove_dates . . . 158
pysar.remove_plane . . . 158
pysar.rewrap . . . 159
pysar.save_gmt . . . 160
pysar.save_kml . . . 161
pysar.save_unavco . . . 161
pysar.save_unw . . . 163
pysar.seed_data . . . 163
pysar.simulation . . . 166
pysar.spatial_average . . . 167
pysar.subset . . . 168
pysar.sum_epochs . . . 172
pysar.temporal_average . . . 173
pysar.temporal_coherence . . . 173
pysar.temporal_derivative . . . 174
pysar.timeseries2velocity . . . 175
pysar.transect . . . 177
pysar.transect_legacy . . . 179
pysar.tropcor_phase_elevation . . . 201
pysar.tropcor_pyaps . . . 201
pysar.tsview_mli . . . 203
pysar.tsviewer . . . 205
pysar.unavco2insarmaps . . . 206
pysar.unwrap_error . . . 208
pysar.view . . . 209
pysar.view_legacy . . . 215
troposphere_uncertainty . . . 217

# Chapter 14

# Hierarchical Index

## 14.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:


- BasicHTTP . . . . . 221
- InsarDatabaseController . . . . . 222
- object
  - timeseries . . . . . 226
- Basemap
  - Basemap2 . . . . . 219

# Chapter 15

# Class Index

## 15.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:


Basemap2
Class ################################################ . . . . . . . . . . . . . . . 219
BasicHTTP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 221
InsarDatabaseController . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 222
timeseries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 226

# Chapter 16

# File Index

## 16.1 File List

Here is a list of all files with brief descriptions:


/Users/jeromezhang/Documents/development/python/PySAR/pysar/__init__.py . . . . . . . . . . . . . 234
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_datetime.py . . . . . . . . . . . . 234
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_gmt.py . . . . . . . . . . . . . . 235
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_network.py . . . . . . . . . . . . 235
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_pysar_utilities.py . . . . . . . . . . 235
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_readfile.py . . . . . . . . . . . . 236
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_remove_surface.py . . . . . . . . . 237
/Users/jeromezhang/Documents/development/python/PySAR/pysar/_writefile.py . . . . . . . . . . . . 237
/Users/jeromezhang/Documents/development/python/PySAR/pysar/add.py . . . . . . . . . . . . . . . 238
/Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attribute.py . . . . . . . . . . . 238
/Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attributes_insarmaps.py . . . . 238
/Users/jeromezhang/Documents/development/python/PySAR/pysar/asc_desc.py . . . . . . . . . . . . 239
/Users/jeromezhang/Documents/development/python/PySAR/pysar/baseline_error.py . . . . . . . . . . 239
/Users/jeromezhang/Documents/development/python/PySAR/pysar/baseline_trop.py . . . . . . . . . . . 239
/Users/jeromezhang/Documents/development/python/PySAR/pysar/convert2mat.py . . . . . . . . . . . 240
/Users/jeromezhang/Documents/development/python/PySAR/pysar/correct_dem.py . . . . . . . . . . . 240
/Users/jeromezhang/Documents/development/python/PySAR/pysar/correlation_with_dem.py . . . . . . 240
/Users/jeromezhang/Documents/development/python/PySAR/pysar/dem_error.py . . . . . . . . . . . . 241
/Users/jeromezhang/Documents/development/python/PySAR/pysar/diff.py . . . . . . . . . . . . . . . 241
/Users/jeromezhang/Documents/development/python/PySAR/pysar/drop_turbulence.py . . . . . . . . . 241
/Users/jeromezhang/Documents/development/python/PySAR/pysar/filter_spatial.py . . . . . . . . . . . 242
/Users/jeromezhang/Documents/development/python/PySAR/pysar/filter_temporal.py . . . . . . . . . . 242
/Users/jeromezhang/Documents/development/python/PySAR/pysar/gamma_view.py . . . . . . . . . . . 242
/Users/jeromezhang/Documents/development/python/PySAR/pysar/generate_mask.py . . . . . . . . . . 242
/Users/jeromezhang/Documents/development/python/PySAR/pysar/geocode.py . . . . . . . . . . . . . 243
/Users/jeromezhang/Documents/development/python/PySAR/pysar/igram_closure.py . . . . . . . . . . 243
/Users/jeromezhang/Documents/development/python/PySAR/pysar/igram_inversion.py . . . . . . . . . 243
/Users/jeromezhang/Documents/development/python/PySAR/pysar/image_math.py . . . . . . . . . . . 244
/Users/jeromezhang/Documents/development/python/PySAR/pysar/incidence_angle.py . . . . . . . . . 244
/Users/jeromezhang/Documents/development/python/PySAR/pysar/info.py . . . . . . . . . . . . . . . 244
/Users/jeromezhang/Documents/development/python/PySAR/pysar/insar_vs_gps.py . . . . . . . . . . . 245
/Users/jeromezhang/Documents/development/python/PySAR/pysar/insarmaps_query.py . . . . . . . . . 247
/Users/jeromezhang/Documents/development/python/PySAR/pysar/l1.py . . . . . . . . . . . . . . . . 247
/Users/jeromezhang/Documents/development/python/PySAR/pysar/load_data.py . . . . . . . . . . . . 247
/Users/jeromezhang/Documents/development/python/PySAR/pysar/load_dem.py . . . . . . . . . . . . 248

/Users/jeromezhang/Documents/development/python/PySAR/pysar/lod.py . . . . . . . . . . . . . . . . 248
/Users/jeromezhang/Documents/development/python/PySAR/pysar/look_angle.py . . . . . . . . . . . . 249
/Users/jeromezhang/Documents/development/python/PySAR/pysar/los2enu.py . . . . . . . . . . . . . 249
/Users/jeromezhang/Documents/development/python/PySAR/pysar/mask.py . . . . . . . . . . . . . . . 249
/Users/jeromezhang/Documents/development/python/PySAR/pysar/match.py . . . . . . . . . . . . . . 250
/Users/jeromezhang/Documents/development/python/PySAR/pysar/mean_spatial.py . . . . . . . . . . . 250
/Users/jeromezhang/Documents/development/python/PySAR/pysar/modify_network.py . . . . . . . . . 250
/Users/jeromezhang/Documents/development/python/PySAR/pysar/multi_transect.py . . . . . . . . . . 252
/Users/jeromezhang/Documents/development/python/PySAR/pysar/multilook.py . . . . . . . . . . . . . 255
/Users/jeromezhang/Documents/development/python/PySAR/pysar/plot_atmDrop.py . . . . . . . . . . . 257
/Users/jeromezhang/Documents/development/python/PySAR/pysar/plot_network.py . . . . . . . . . . . 257
/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysar2insarmaps.py . . . . . . . . . 258
/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp.py . . . . . . . . . . . . . 258
/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp_cmd.py . . . . . . . . . . 259
/Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp_orig.py . . . . . . . . . . 259
/Users/jeromezhang/Documents/development/python/PySAR/pysar/quality_map.py . . . . . . . . . . . 260
/Users/jeromezhang/Documents/development/python/PySAR/pysar/reconstruct_igrams.py . . . . . . . 260
/Users/jeromezhang/Documents/development/python/PySAR/pysar/reference_epoch.py . . . . . . . . . 261
/Users/jeromezhang/Documents/development/python/PySAR/pysar/remove_dates.py . . . . . . . . . . 261
/Users/jeromezhang/Documents/development/python/PySAR/pysar/remove_plane.py . . . . . . . . . . 261
/Users/jeromezhang/Documents/development/python/PySAR/pysar/rewrap.py . . . . . . . . . . . . . . 262
/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_gmt.py . . . . . . . . . . . . . 262
/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_kml.py . . . . . . . . . . . . . 262
/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_unavco.py . . . . . . . . . . . 263
/Users/jeromezhang/Documents/development/python/PySAR/pysar/save_unw.py . . . . . . . . . . . . 263
/Users/jeromezhang/Documents/development/python/PySAR/pysar/seed_data.py . . . . . . . . . . . . 263
/Users/jeromezhang/Documents/development/python/PySAR/pysar/simulation.py . . . . . . . . . . . . 264
/Users/jeromezhang/Documents/development/python/PySAR/pysar/spatial_average.py . . . . . . . . . 264
/Users/jeromezhang/Documents/development/python/PySAR/pysar/subset.py . . . . . . . . . . . . . . 265
/Users/jeromezhang/Documents/development/python/PySAR/pysar/sum_epochs.py . . . . . . . . . . . 265
/Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_average.py . . . . . . . . 266
/Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_coherence.py . . . . . . . 266
/Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_derivative.py . . . . . . . 266
/Users/jeromezhang/Documents/development/python/PySAR/pysar/timeseries2velocity.py . . . . . . . 267
/Users/jeromezhang/Documents/development/python/PySAR/pysar/transect.py . . . . . . . . . . . . . 267
/Users/jeromezhang/Documents/development/python/PySAR/pysar/transect_legacy.py . . . . . . . . . 268
/Users/jeromezhang/Documents/development/python/PySAR/pysar/tropcor_phase_elevation.py . . . . . 270
/Users/jeromezhang/Documents/development/python/PySAR/pysar/tropcor_pyaps.py . . . . . . . . . . 271
/Users/jeromezhang/Documents/development/python/PySAR/pysar/tsview_mli.py . . . . . . . . . . . . 271
/Users/jeromezhang/Documents/development/python/PySAR/pysar/tsviewer.py . . . . . . . . . . . . . 272
/Users/jeromezhang/Documents/development/python/PySAR/pysar/unavco2insarmaps.py . . . . . . . . 272
/Users/jeromezhang/Documents/development/python/PySAR/pysar/unwrap_error.py . . . . . . . . . . . 272
/Users/jeromezhang/Documents/development/python/PySAR/pysar/view.py . . . . . . . . . . . . . . . 273
/Users/jeromezhang/Documents/development/python/PySAR/pysar/view_legacy.py . . . . . . . . . . . 274
/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/delayTimeseries.py . . . . . 251
/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/dloadUtil.py . . . . . . . . . 251
/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/get_modis_v3.py . . . . . . . 252
/Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/troposphere_uncertainty.py . 252
/Users/jeromezhang/Documents/development/python/PySAR/pysar/plot/plot_tropcor_phase_elevation.py 256

# Chapter 17

# Namespace Documentation

## 17.1 delayTimeseries Namespace Reference

### Classes

- class timeseries

### Functions

- def write_to_h5 (dataset, outName, groupName, h5withAttributes)
- def nearest_valid (xr, yr, data_flat, rows, cols)

### 17.1.1 Function Documentation

#### 17.1.1.1 nearest_valid()

```
def delayTimeseries.nearest_valid (
              xr,
              yr,
              data_flat,
              rows,
              cols )
```

#### 17.1.1.2 write_to_h5()

```
def delayTimeseries.write_to_h5 (
              dataset,
              outName,
              groupName,
              h5withAttributes )
```

## 17.2 dloadUtil Namespace Reference

### Functions

- def download_modis (inps)
- def download_atmosphereModel (inps)
- def daterange (start_date, end_date)
- def get_date (f)
- def pwv2zwd (pwv)
- def zwd2swd (zwd, theta)
- def read_modis (file)

### 17.2.1 Function Documentation

#### 17.2.1.1 daterange()

```
def dloadUtil.daterange (
              start_date,
              end_date )
```

#### 17.2.1.2 download_atmosphereModel()

```
def dloadUtil.download_atmosphereModel (
              inps )
```

#### 17.2.1.3 download_modis()

```
def dloadUtil.download_modis (
              inps )
```

#### 17.2.1.4 get_date()

```
def dloadUtil.get_date (
              f )
```

#### 17.2.1.5 pwv2zwd()

```
def dloadUtil.pwv2zwd (
              pwv )
```

#### 17.2.1.6 read_modis()

```
def dloadUtil.read_modis (
              file )
```

#### 17.2.1.7 zwd2swd()

```
def dloadUtil.zwd2swd (
              zwd,
              theta )
```

## 17.3 get_modis_v3 Namespace Reference

### Functions

- def usage ()
- def main ()

### Variables

- out = sys.stdout
- start_time_main = time.time()
- time_elapsed = time.time() - start_time_main

### 17.3.1 Function Documentation

#### 17.3.1.1 main()

```
def get_modis_v3.main ( )
```

#### 17.3.1.2 usage()

```
def get_modis_v3.usage ( )
```

### 17.3.2 Variable Documentation

#### 17.3.2.1 out

```
out = sys.stdout
```

#### 17.3.2.2 start_time_main

```
start_time_main = time.time()
```

#### 17.3.2.3 time_elapsed

```
time_elapsed = time.time() - start_time_main
```

## 17.4 plot_tropcor_phase_elevation Namespace Reference

### Variables

- string workDir = '/scratch/projects/insarlab/yzhang1/KyushuT80F245_246JersD/TSSAR'
- string demFile = 'radar_4rlks.hgt'
- string timeseriesFile = 'timeseries_demCor.h5'
- string timeseriesFile2 = 'timeseries_demCor_tropHgt.h5'
- string maskFile = 'Mask_tempCoh_dis.h5'
- string tropHgtFile = 'tropHgt.h5'
- string ecmwfFile = 'ECMWF.h5'
- string epoch = '19980926'
- dem
- dem_atr
- data
- atr
- data2
- atr2
- tropHgt
- atr3
- ecmwf
- atr4
- mask

- msk_atr
- ndx = np.nan
- list dataList = [data,data2,-tropHgt,-ecmwf]
- fig
- axes
- nrows
- ncols
- sharex
- True
- sharey
- figsize
- int i = 0
- ms
- bbox_inches
- dpi

### 17.4.1 Variable Documentation

#### 17.4.1.1 atr

```
atr
```

#### 17.4.1.2 atr2

```
atr2
```

#### 17.4.1.3 atr3

```
atr3
```

#### 17.4.1.4 atr4

```
atr4
```

#### 17.4.1.5 axes

```
axes
```

#### 17.4.1.6 bbox_inches

```
bbox_inches
```

#### 17.4.1.7 data

```
data
```

#### 17.4.1.8 data2

```
data2
```

#### 17.4.1.9 dataList

```
list dataList = [data,data2,-tropHgt,-ecmwf]
```

#### 17.4.1.10 dem

```
dem
```

#### 17.4.1.11 dem_atr

```
dem_atr
```

#### 17.4.1.12 demFile

```
string demFile = 'radar_4rlks.hgt'
```

#### 17.4.1.13 dpi

```
dpi
```

#### 17.4.1.14 ecmwf

```
ecmwf
```

#### 17.4.1.15 ecmwfFile

```
string ecmwfFile = 'ECMWF.h5'
```

#### 17.4.1.16 epoch

```
string epoch = '19980926'
```

#### 17.4.1.17 fig

```
fig
```

#### 17.4.1.18 figsize

```
figsize
```

#### 17.4.1.19 i

```
int i = 0
```

#### 17.4.1.20 mask

```
mask
```

#### 17.4.1.21 maskFile

```
string maskFile = 'Mask_tempCoh_dis.h5'
```

#### 17.4.1.22 ms

```
ms
```

#### 17.4.1.23 msk_atr

```
msk_atr
```

#### 17.4.1.24 ncols

```
ncols
```

#### 17.4.1.25 ndx

```
ndx = np.nan
```

#### 17.4.1.26 nrows

```
nrows
```

#### 17.4.1.27 sharex

```
sharex
```

#### 17.4.1.28 sharey

```
sharey
```

#### 17.4.1.29 timeseriesFile

```
string timeseriesFile = 'timeseries_demCor.h5'
```

#### 17.4.1.30 timeseriesFile2

```
string timeseriesFile2 = 'timeseries_demCor_tropHgt.h5'
```

#### 17.4.1.31 tropHgt

```
tropHgt
```

#### 17.4.1.32 tropHgtFile

```
string tropHgtFile = 'tropHgt.h5'
```

#### 17.4.1.33 True

```
True
```

#### 17.4.1.34 workDir

```
string workDir = '/scratch/projects/insarlab/yzhang1/KyushuT80F245_246JersD/TSSAR'
```

## 17.5 pysar Namespace Reference

### Namespaces

- _datetime
- _gmt
- _network
- _pysar_utilities
- _readfile
- _remove_surface
- _writefile
- add
- add_attribute
- add_attributes_insarmaps
- asc_desc
- baseline_error
- baseline_trop
- convert2mat
- correct_dem
- correlation_with_dem
- dem_error
- diff
- drop_turbulence
- filter_spatial
- filter_temporal
- gamma_view
- generate_mask
- geocode
- igram_closure
- igram_inversion
- image_math
- incidence_angle
- info
- insar_vs_gps
- insarmaps_query
- l1
- load_data
- load_dem
- lod
- look_angle
- los2enu
- mask
- match
- mean_spatial
- modify_network
- multi_transect
- multilook
- plot_atmDrop
- plot_network
- pysar2insarmaps
- pysarApp
- pysarApp_cmd
- pysarApp_orig
- quality_map

- reconstruct_igrams
- reference_epoch
- remove_dates
- remove_plane
- rewrap
- save_gmt
- save_kml
- save_unavco
- save_unw
- seed_data
- simulation
- spatial_average
- subset
- sum_epochs
- temporal_average
- temporal_coherence
- temporal_derivative
- timeseries2velocity
- transect
- transect_legacy
- tropcor_phase_elevation
- tropcor_pyaps
- tsview_mli
- tsviewer
- unavco2insarmaps
- unwrap_error
- view
- view_legacy

### Variables

- bool miami_path = True

## 17.5.1 Variable Documentation

### 17.5.1.1 miami_path

```
bool miami_path = True
```

# 17.6 pysar._datetime Namespace Reference

### Functions

- def yyyymmdd2years (dates)
- def yymmdd2yyyymmdd (date)
- def yyyymmdd (dates)
- def yymmdd (dates)
- def igram_date_list (igramFile)
- def read_date_list (date_list_file)
- def date_index (dateList)
- def date_list2tbase (dateList)
- def date_list2vector (dateList)
- def auto_adjust_xaxis_date (ax, datevector, fontSize=12)

### 17.6.1 Function Documentation

#### 17.6.1.1 auto_adjust_xaxis_date()

```
def pysar._datetime.auto_adjust_xaxis_date (
              ax,
              datevector,
              fontSize = 12 )
```

```
Adjust X axis
Input:
    ax : matplotlib figure axes object
    datevector : list of float, date in years
                 i.e. [2007.013698630137, 2007.521917808219, 2007.6463470319634]
Output:
    ax : matplotlib figure axes object
```

#### 17.6.1.2 date_index()

```
def pysar._datetime.date_index (
              dateList )
```

#### 17.6.1.3 date_list2tbase()

```
def pysar._datetime.date_list2tbase (
              dateList )
```

```
Get temporal Baseline in days with respect to the 1st date
```

#### 17.6.1.4 date_list2vector()

```
def pysar._datetime.date_list2vector (
              dateList )
```

```
Get time in datetime format: datetime.datetime(2006, 5, 26, 0, 0)
```

#### 17.6.1.5 igram_date_list()

```
def pysar._datetime.igram_date_list (
              igramFile )
```

```
Read Date List from Interferogram file
    for timeseries file, use h5file['timeseries'].keys() directly
```

#### 17.6.1.6 read_date_list()

```
def pysar._datetime.read_date_list (
              date_list_file )
```

```
Read Date List from txt file
```

#### 17.6.1.7 yymmdd()

```
def pysar._datetime.yymmdd (
              dates )
```

#### 17.6.1.8 yymmdd2yyyymmdd()

```
def pysar._datetime.yymmdd2yyyymmdd (
              date )
```

#### 17.6.1.9 yyyymmdd()

```
def pysar._datetime.yyyymmdd (
              dates )
```

#### 17.6.1.10 yyyymmdd2years()

```
def pysar._datetime.yyyymmdd2years (
              dates )
```

## 17.7 pysar._gmt Namespace Reference

### Functions

- def write_gmt_simple (lons, lats, z, fname, title='default', name='z', scale=1.0, offset=0, units='meters')

### 17.7.1 Function Documentation

#### 17.7.1.1 write_gmt_simple()

```
def pysar._gmt.write_gmt_simple (
              lons,
              lats,
              z,
              fname,
              title = 'default',
              name = 'z',
              scale = 1.0,
              offset = 0,
              units = 'meters' )
```

```
Writes a simple GMT grd file with one array.

.. Args:

    * lons      -> 1D Array of lon values
    * lats      -> 1D Array of lat values
    * z         -> 2D slice to be saved
    * fname     -> Output file name

.. Kwargs:

    * title     -> Title for the grd file
    * name      -> Name of the field in the grd file
    * scale     -> Scale value in the grd file
    * offset    -> Offset value in the grd file

.. Returns:

    * None
```

## 17.8 pysar._network Namespace Reference

### Functions

- def read_pairs_list (date12ListFile, dateList=[ ])
- def write_pairs_list (pairs, dateList, outName)
- def read_igram_pairs (igramFile)
- def read_baseline_file (baselineFile, exDateList=[ ])
- def date12_list2index (date12_list, date_list=[ ])
- def get_date12_list (File)
- def igram_perp_baseline_list (File)

- def threshold_perp_baseline (igramIdxList, perpBaseList, perpBaseMax=800, perpBaseMin=0)
- def threshold_temporal_baseline (igramIdxList, tempBaseList, tempBaseMax=365, seasonal=1, tempBase↩ Min=0)
- def pair_sort (pairs)
- def pair_merge (pairs1, pairs2)
- def select_pairs_all (dateList)
- def select_pairs_delaunay (tempBaseList, perpBaseList, normalize=1)
- def select_pairs_sequential (dateList, num_incr=2)
- def select_pairs_hierarchical (tempBaseList, perpBaseList, tempPerpList)
- def select_pairs_mst (tempBaseList, perpBaseList, normalize=1)
- def select_pairs_star (dateList, m_date)
- def plot_network (ax, pairs_idx, date8List, bperpList, plot_dict={})
- def plot_perp_baseline_hist (ax, date8List, bperpList, plot_dict={})
- def auto_adjust_yaxis (ax, dataList, fontSize=12)

## 17.8.1 Function Documentation

### 17.8.1.1 auto_adjust_yaxis()

```
def pysar._network.auto_adjust_yaxis (
              ax,
              dataList,
              fontSize = 12 )
```

```
Adjust Y axis
Input:
    ax - matplot figure axes object
    dataList : list of float, value in y axis
```

### 17.8.1.2 date12_list2index()

```
def pysar._network.date12_list2index (
              date12_list,
              date_list = [] )
```

```
Convert list of date12 string into list of index
```

### 17.8.1.3 get_date12_list()

```
def pysar._network.get_date12_list (
              File )
```

```
Read Date12 info from input file: Pairs.list or multi-group hdf5 file
Example:
    date12List = get_date12_list('unwrapIfgram.h5')
    date12List = get_date12_list('Pairs.list')
```

#### 17.8.1.4 igram_perp_baseline_list()

```
def pysar._network.igram_perp_baseline_list (
             File )
```

```
Get perpendicular baseline list from input multi_group hdf5 file
```

#### 17.8.1.5 pair_merge()

```
def pysar._network.pair_merge (
             pairs1,
             pairs2 )
```

#### 17.8.1.6 pair_sort()

```
def pysar._network.pair_sort (
             pairs )
```

#### 17.8.1.7 plot_network()

```
def pysar._network.plot_network (
             ax,
             pairs_idx,
             date8List,
             bperpList,
             plot_dict = {} )
```

```
Plot Temporal-Perp baseline Network
Inputs
    ax : matplotlib axes object
    pairs_idx : list of list of 2 int, pairs index, len = number of interferograms
    date8List : list of 8-digit string, date, len=number of acquisition
    bperpList : list of float, perp baseline, len=number of acquisition
    plot_dict : dictionary with the following items:
                fontsize
                linewidth
                markercolor
                markersize

                coherence_list : list of float, coherence value of each interferogram, len = number of ifgrams
                disp_min/max :  float, min/max range of the color display based on coherence_list
                colormap : string, colormap name
Output
    ax : matplotlib axes object
```

#### 17.8.1.8 plot_perp_baseline_hist()

```
def pysar._network.plot_perp_baseline_hist (
              ax,
              date8List,
              bperpList,
              plot_dict = {} )
```

```
Plot Perpendicular Spatial Baseline History
Inputs
    ax : matplotlib axes object
    date8List : list of 8-digit string, date
    bperpList : list of float, perp baseline
    plot_dict : dictionary with the following items:
                fontsize
                linewidth
                markercolor
                markersize
Output:
    ax : matplotlib axes object
```

#### 17.8.1.9 read_baseline_file()

```
def pysar._network.read_baseline_file (
              baselineFile,
              exDateList = [] )
```

```
Read bl_list.txt without dates listed in exDateList
Examples:
    date8List, perpBaseList, dopList, prfList, slcDirList = read_baseline_file(baselineFile)
    date8List, perpBaseList, dopList, prfList, slcDirList = read_baseline_file(baselineFile,['080520','100726'
    date8List, perpBaseList = read_baseline_file(baselineFile)[0:2]
```

#### 17.8.1.10 read_igram_pairs()

```
def pysar._network.read_igram_pairs (
              igramFile )
```

```
Read pairs index from hdf5 file
```

#### 17.8.1.11 read_pairs_list()

```
def pysar._network.read_pairs_list (
              date12ListFile,
              dateList = [] )
```

```
Read Pairs List file like below:
070311-070426
070311-070611
...
```

#### 17.8.1.12 select_pairs_all()

```
def pysar._network.select_pairs_all (
              dateList )
```

```
Select All Possible Pairs/Interferograms
Reference:
    Berardino, P., G. Fornaro, R. Lanari, and E. Sansosti (2002), A new algorithm for surface deformation moni
    based on small baseline differential SAR interferograms, IEEE TGRS, 40(11), 2375-2383.
```

#### 17.8.1.13 select_pairs_delaunay()

```
def pysar._network.select_pairs_delaunay (
              tempBaseList,
              perpBaseList,
              normalize = 1 )
```

```
Select Pairs using Delaunay Triangulation based on temporal/perpendicular baselines
Usage:
    tempBaseList : list of temporal baseline
    perpBaseList : list of perpendicular spatial baseline
    normalize    : normalize temporal baseline to perpendicular baseline
                   1 - enable  normalization, default
                   0 - disable normalization
Key points
    1. Define a ratio between perpendicular and temporal baseline axis units (Pepe and Lanari, 2006, TGRS).
    2. Pairs with too large perpendicular / temporal baseline or Doppler centroid difference should be removed
       after this, using a threshold, to avoid strong decorrelations (Zebker and Villasenor, 1992, TGRS).
Reference:
    Pepe, A., and R. Lanari (2006), On the extension of the minimum cost flow algorithm for phase unwrapping
    of multitemporal differential SAR interferograms, IEEE TGRS, 44(9), 2374-2383.
    Zebker, H. A., and J. Villasenor (1992), Decorrelation in interferometric radar echoes, IEEE TGRS, 30(5),
```

#### 17.8.1.14 select_pairs_hierarchical()

```
def pysar._network.select_pairs_hierarchical (
              tempBaseList,
              perpBaseList,
              tempPerpList )
```

```
Select Pairs in a hierarchical way using list of temporal and perpendicular baseline thresholds
    For each temporal/perpendicular combination, select all possible pairs; and then merge all combination res
    together for the final output (Zhao, 2015).
Examples:
    pairs = select_pairs_hierarchical(tempBaseList,perpBaseList,[[32, 800], [48, 600], [64, 200]])
Reference:
    Zhao, W., (2015), Small deformation detected from InSAR time-series and their applications in geophysics,
    dissertation, Univ. of Miami, Section 6.3.
```

#### 17.8.1.15 select_pairs_mst()

```
def pysar._network.select_pairs_mst (
              tempBaseList,
              perpBaseList,
              normalize = 1 )
```

```
Select Pairs using Minimum Spanning Tree technique
    Connection Cost is calculated using the baseline distance in perp and scaled temporal baseline (Pepe and L
    2006, TGRS) plane.
References:
    Pepe, A., and R. Lanari (2006), On the extension of the minimum cost flow algorithm for phase unwrapping
    of multitemporal differential SAR interferograms, IEEE TGRS, 44(9), 2374-2383.
    Perissin D., Wang T. (2012), Repeat-pass SAR interferometry with partially coherent targets. IEEE TGRS. 27
```

#### 17.8.1.16 select_pairs_sequential()

```
def pysar._network.select_pairs_sequential (
              dateList,
              num_incr = 2 )
```

```
Select Pairs in a Sequential way:
    For each acquisition, find its num_incr nearest acquisitions in the past time.
Reference:
    Fattahi, H., and F. Amelung (2013), DEM Error Correction in InSAR Time Series, IEEE TGRS, 51(7), 4249-4259
```

#### 17.8.1.17 select_pairs_star()

```
def pysar._network.select_pairs_star (
              dateList,
              m_date )
```

```
Select Star-like network/interferograms/pairs, it's a single master network, similar to PS approach.
Usage:
    m_date : master date, choose it based on the following cretiria:
             1) near the center in temporal and spatial baseline
             2) prefer winter season than summer season for less temporal decorrelation
Reference:
    Ferretti, A., C. Prati, and F. Rocca (2001), Permanent scatterers in SAR interferometry, IEEE TGRS, 39(1),
```

#### 17.8.1.18 threshold_perp_baseline()

```
def pysar._network.threshold_perp_baseline (
              igramIdxList,
              perpBaseList,
              perpBaseMax = 800,
              perpBaseMin = 0 )
```

```
Remove pairs/interoferogram out of [perpBaseMin, perpBaseMax]
Example:
    pairs = threshold_perp_baseline(pairs,perpBaseList,500)
```

#### 17.8.1.19 threshold_temporal_baseline()

```
def pysar._network.threshold_temporal_baseline (
              igramIdxList,
              tempBaseList,
              tempBaseMax = 365,
              seasonal = 1,
              tempBaseMin = 0 )
```

```
Remove pairs/interferograms out of min/max/seasonal temporal baseline limits
Usage:
    seasonal : keep interferograms with seasonal temporal baseline
               1 - keep them, by default
               0 - do not keep them
Example:
    pairs = threshold_temporal_baseline(pairs,tempBaseList,80)
    pairs = threshold_temporal_baseline(pairs,tempBaseList,80,0)  # disable seasonal checking
```

#### 17.8.1.20 write_pairs_list()

```
def pysar._network.write_pairs_list (
              pairs,
              dateList,
              outName )
```

## 17.9 pysar._pysar_utilities Namespace Reference

### Functions

- def incidence_angle (atr, dimension=2)
- def which (program)
- def get_file_stack (File, maskFile=None)
- def nonzero_mask (File, outFile='Mask.h5')
- def spatial_average (File, mask=None, box=None, saveList=False)
- def temporal_average (File, outFile=None)
- def get_file_list (fileList)
- def print_progress (iteration, total, prefix='calculating:', suffix='complete', decimals=1, barLength=50)
- def glob2radar (lat, lon, geomapFile='geomap ∗.trans', rdrFile=None)
- def radar2glob (az, rg, geomapFile='geomap ∗.trans', rdrFile=None)
- def radar_or_geo (File)
- def check_variable_name (path)
- def hillshade (data, scale)
- def date_list (h5file)
- def YYYYMMDD2years (d)
- def design_matrix (h5file)
- def timeseries_inversion (igramsFile, timeseriesFile)
- def timeseries_inversion_FGLS (h5flat, h5timeseries)
- def timeseries_inversion_L1 (h5flat, h5timeseries)
- def Baseline_timeseries (igramsFile)
- def dBh_dBv_timeseries (igramsFile)

- def Bh_Bv_timeseries (igramsFile)
- def stacking (File)
- def yymmdd2YYYYMMDD (date)
- def yyyymmdd (dates)
- def yymmdd (dates)
- def make_triangle (dates12, igram1, igram2, igram3)
- def get_triangles (h5file)
- def generate_curls (curlfile, h5file, Triangles, curls)

## 17.9.1 Function Documentation

#### 17.9.1.1 Baseline_timeseries()

```
def pysar._pysar_utilities.Baseline_timeseries (
              igramsFile )
```

#### 17.9.1.2 Bh_Bv_timeseries()

```
def pysar._pysar_utilities.Bh_Bv_timeseries (
              igramsFile )
```

#### 17.9.1.3 check_variable_name()

```
def pysar._pysar_utilities.check_variable_name (
              path )
```

#### 17.9.1.4 date_list()

```
def pysar._pysar_utilities.date_list (
              h5file )
```

#### 17.9.1.5 dBh_dBv_timeseries()

```
def pysar._pysar_utilities.dBh_dBv_timeseries (
              igramsFile )
```

#### 17.9.1.6 design_matrix()

```
def pysar._pysar_utilities.design_matrix (
              h5file )
```

```
Make the design matrix for the inversion.
```

#### 17.9.1.7 generate_curls()

```
def pysar._pysar_utilities.generate_curls (
              curlfile,
              h5file,
              Triangles,
              curls )
```

#### 17.9.1.8 get_file_list()

```
def pysar._pysar_utilities.get_file_list (
              fileList )
```

```
Get all existed files matching the input list of file pattern
Example:
fileList = get_file_list(['*velocity*.h5','timeseries*.h5'])
```

#### 17.9.1.9 get_file_stack()

```
def pysar._pysar_utilities.get_file_stack (
              File,
              maskFile = None )
```

```
Get stack file of input File and return the stack 2D matrix
Input:   File/maskFile - string
Output:  stack - 2D np.array matrix
```

#### 17.9.1.10 get_triangles()

```
def pysar._pysar_utilities.get_triangles (
              h5file )
```

#### 17.9.1.11 glob2radar()

def pysar._pysar_utilities.glob2radar (
*lat,*
*lon,*
*geomapFile = 'geomap*.trans',*
*rdrFile = None* )

```
Convert geo coordinates into radar coordinates.
Inputs:
    lat/lon    - np.array, float, latitude/longitude
    geomapFile - string, trans/look up file
    rdrFile    - string, file in radar coord, optional but recommended.
Output:
    az/rg     - np.array, float, range/azimuth pixel number
    az/rg_res - float, residul/uncertainty of coordinate conversion
```

#### 17.9.1.12 hillshade()

def pysar._pysar_utilities.hillshade (
*data,*
*scale* )

```
from scott baker, ptisk library
```

#### 17.9.1.13 incidence_angle()

def pysar._pysar_utilities.incidence_angle (
*atr,*
*dimension = 2* )

```
Calculate 2D matrix of incidence angle from ROI_PAC attributes, very accurate.
Input:
    dictionary - ROI_PAC attributes including the following items:
                 STARTING_RANGE
                 RANGE_PIXEL_SIZE
                 EARTH_RADIUS
                 HEIGHT
                 FILE_LENGTH
                 WIDTH
    dimension - int,
                2 for 2d matrix
                1 for 1d array
                0 for one center value
Output: 2D np.array - incidence angle in degree for each pixel
```

#### 17.9.1.14 make_triangle()

```
def pysar._pysar_utilities.make_triangle (
              dates12,
              igram1,
              igram2,
              igram3 )
```

#### 17.9.1.15 nonzero_mask()

```
def pysar._pysar_utilities.nonzero_mask (
              File,
              outFile = 'Mask.h5' )
```

```
Generate mask file for non-zero value of input multi-group hdf5 file
```

#### 17.9.1.16 print_progress()

```
def pysar._pysar_utilities.print_progress (
              iteration,
              total,
              prefix = 'calculating:',
              suffix = 'complete',
              decimals = 1,
              barLength = 50 )
```

```
Print iterations progress - Greenstick from Stack Overflow
Call in a loop to create terminal progress bar
@params:
    iteration   - Required  : current iteration (Int)
    total       - Required  : total iterations (Int)
    prefix      - Optional  : prefix string (Str)
    suffix      - Optional  : suffix string (Str)
    decimals    - Optional  : number of decimals in percent complete (Int)
    barLength   - Optional  : character length of bar (Int)

Reference: http://stackoverflow.com/questions/3173320/text-progress-bar-in-the-console
```

#### 17.9.1.17 radar2glob()

```
def pysar._pysar_utilities.radar2glob (
              az,
              rg,
              geomapFile = 'geomap*.trans',
              rdrFile = None )
```

```
Convert radar coordinates into geo coordinates
Inputs:
    rg/az      - np.array, int, range/azimuth pixel number
    geomapFile - string, trans/look up file
    rdrFile    - string, file in radar coord, optional but recommended.
Output:
    lon/lat    - np.array, float, longitude/latitude of input point (rg,az)
    latlon_res - float, residul/uncertainty of coordinate conversion
```

#### 17.9.1.18 radar_or_geo()

```
def pysar._pysar_utilities.radar_or_geo (
             File )
```

```
Check File is in Radar or Geo coordinate
```

#### 17.9.1.19 spatial_average()

```
def pysar._pysar_utilities.spatial_average (
             File,
             mask = None,
             box = None,
             saveList = False )
```

```
Calculate  Spatial Average.
    Only non-nan pixel is considered.
Input:
    File : string, path of input file
    mask : 2D np.array, mask file
    box  : 4-tuple defining the left, upper, right, and lower pixel coordinate
    saveList: bool, save (list of) mean value into text file
Output:
    meanList : list for float, average value in space for each epoch of input file
Example:
    meanList = spatial_average('coherence.h5')
    meanList = spatial_average('coherence.h5', mask, saveList=True)
    refList = spatial_average('unwrapIfgram.h5', box=(100,200,101,201))
```

#### 17.9.1.20 stacking()

```
def pysar._pysar_utilities.stacking (
             File )
```

```
Stack multi-temporal dataset into one
   equivalent to temporal sum
```

#### 17.9.1.21 temporal_average()

```
def pysar._pysar_utilities.temporal_average (
             File,
             outFile = None )
```

```
Calculate temporal average.
```

#### 17.9.1.22 timeseries_inversion()

def pysar._pysar_utilities.timeseries_inversion (
*igramsFile,*
*timeseriesFile* )

```
Implementation of the SBAS algorithm.
modified from sbas.py written by scott baker, 2012

Usage:
timeseries_inversion(h5flat,h5timeseries)
  h5flat: hdf5 file with the interferograms
  h5timeseries: hdf5 file with the output from the inversion
```

#### 17.9.1.23 timeseries_inversion_FGLS()

def pysar._pysar_utilities.timeseries_inversion_FGLS (
*h5flat,*
*h5timeseries* )

```
Implementation of the SBAS algorithm.

Usage:
timeseries_inversion(h5flat,h5timeseries)
  h5flat: hdf5 file with the interferograms
  h5timeseries: hdf5 file with the output from the inversion
##################################################
```

#### 17.9.1.24 timeseries_inversion_L1()

def pysar._pysar_utilities.timeseries_inversion_L1 (
*h5flat,*
*h5timeseries* )

#### 17.9.1.25 which()

def pysar._pysar_utilities.which (
*program* )

```
Test if executable exists
```

#### 17.9.1.26 yymmdd()

```
def pysar._pysar_utilities.yymmdd (
              dates )
```

#### 17.9.1.27 yymmdd2YYYYMMDD()

```
def pysar._pysar_utilities.yymmdd2YYYYMMDD (
              date )
```

#### 17.9.1.28 yyyymmdd()

```
def pysar._pysar_utilities.yyyymmdd (
              dates )
```

#### 17.9.1.29 YYYYMMDD2years()

```
def pysar._pysar_utilities.YYYYMMDD2years (
              d )
```

## 17.10 pysar._readfile Namespace Reference

### Functions

- def read (File, box=(), epoch='')
- def read_attribute (File, epoch='')
- def check_variable_name (path)
- def read_template (File, delimiter='=')
- def read_roipac_rsc (File)
- def read_gamma_par (File)
- def read_isce_xml (File)
- def merge_attribute (atr1, atr2)
- def read_float32 (File, box=None)
- def read_complex_float32 (File, real_imag=False)
- def read_real_float32 (File)
- def read_complex_int16 (File, box=None, real_imag=False)
- def read_dem (File)
- def read_real_int16 (File)
- def read_flag (File)
- def read_GPS_USGS (File)
- def read_multiple (File, box='')

### Variables

- list multi_group_hdf5_file = ['interferograms','coherence','wrapped','snaphu_connect_component']
- list multi_dataset_hdf5_file = ['timeseries']
- list single_dataset_hdf5_file = ['dem','mask','rmse','temporal_coherence', 'velocity']

## 17.10.1 Function Documentation

#### 17.10.1.1 check_variable_name()

```
def pysar._readfile.check_variable_name (
              path )
```

#### 17.10.1.2 merge_attribute()

```
def pysar._readfile.merge_attribute (
              atr1,
              atr2 )
```

#### 17.10.1.3 read()

```
def pysar._readfile.read (
              File,
              box = (),
              epoch = '' )
```

```
Read one dataset and its attributes from input file.

Read one dataset, i.e. interferogram, coherence, velocity, dem ...
return 0 if failed.

Inputs:
    File  : str, path of file to read
            PySAR   file: interferograms, timeseries, velocity, etc.
            ROI_PAC file: .unw .cor .hgt .dem .trans
            Gamma   file: .mli .slc
            Image   file: .jpeg .jpg .png .ras .bmp
    box   : 4-tuple of int, area to read, defined in (x0, y0, x1, y1) in pixel coordinate
    epoch : string, epoch to read, for multi-dataset files
            for .trans file:
            '' - return both dataset
            rg, range   - for geomap_*.trans file
            az, azimuth - for geomap_*.trans file

Outputs:
    data : 2-D matrix in numpy.array format, return None if failed
    atr  : dictionary, attributes of data, return None if failed

Examples:
    data, atr = read('velocity.h5')
    data, atr = read('100120-110214.unw', (100,1100, 500, 2500))
    data, atr = read('timeseries.h5', (), '20101120')
    data, atr = read('timeseries.h5', (100,1100, 500, 2500), '20101120')
    az,   atr = read('geomap*.trans', (), 'azimuth')
    rg,az,atr = read('geomap*.trans')
```

#### 17.10.1.4 read_attribute()

```
def pysar._readfile.read_attribute (
              File,
              epoch = '' )
```

```
Read attributes of input file into a dictionary
Input  : string, file name and epoch (optional)
Output : dictionary, attributes dictionary
```

#### 17.10.1.5 read_complex_float32()

```
def pysar._readfile.read_complex_float32 (
              File,
              real_imag = False )
```

```
Read complex float 32 data matrix, i.e. roi_pac int or slc data.
old name: read_complex64()

ROI_PAC file: .slc, .int, .amp

Data is sotred as:
real, imaginary, real, imaginary, ...
real, imaginary, real, imaginary, ...
...

Usage:
    File : input file name
    real_imag : flag for output format,
                0 for amplitude and phase [by default],
                non-0 : for real and imagery

Example:
    amp, phase, atr = read_complex_float32('geo_070603-070721_0048_00018.int')
    data, atr       = read_complex_float32('150707.slc', 1)
```

#### 17.10.1.6 read_complex_int16()

```
def pysar._readfile.read_complex_int16 (
              File,
              box = None,
              real_imag = False )
```

```
Read complex int 16 data matrix, i.e. GAMMA SCOMPLEX file (.slc)

Gamma file: .slc

Inputs:
   file: complex data matrix (cpx_int16)
   box: 4-tuple defining the left, upper, right, and lower pixel coordinate.
Example:
   data,rsc = read_complex_int16('100102.slc')
   data,rsc = read_complex_int16('100102.slc',(100,1200,500,1500))
```

#### 17.10.1.7 read_dem()

```
def pysar._readfile.read_dem (
             File )
```

```
Read real int 16 data matrix, i.e. ROI_PAC .dem file.
Input:  roi_pac format dem file
Usage:  dem, atr = read_real_int16('gsi10m_30m.dem')
```

#### 17.10.1.8 read_flag()

```
def pysar._readfile.read_flag (
             File )
```

```
Read binary file with flags, 1-byte values with flags set in bits
For ROI_PAC .flg, *_snap_connect.byt file.
```

#### 17.10.1.9 read_float32()

```
def pysar._readfile.read_float32 (
             File,
             box = None )
```

```
Reads roi_pac data (RMG format, interleaved line by line)
should rename it to read_rmg_float32()

ROI_PAC file: .unw, .cor, .hgt, .trans, .msk

RMG format (named after JPL radar pionner Richard M. Goldstein): made
up of real*4 numbers in two arrays side-by-side. The two arrays often
show the magnitude of the radar image and the phase, although not always
(sometimes the phase is the correlation). The length and width of each
array are given as lines in the metadata (.rsc) file. Thus the total
width width of the binary file is (2*width) and length is (length), data
are stored as:
magnitude, magnitude, magnitude, ...,phase, phase, phase, ...
magnitude, magnitude, magnitude, ...,phase, phase, phase, ...
......

   box  : 4-tuple defining the left, upper, right, and lower pixel coordinate.
Example:
   a,p,r = read_float32('100102-100403.unw')
   a,p,r = read_float32('100102-100403.unw',(100,1200,500,1500))
```

#### 17.10.1.10 read_gamma_par()

```
def pysar._readfile.read_gamma_par (
              File )
```

```
Read GAMMA .par file into a python dictionary structure.
```

#### 17.10.1.11 read_GPS_USGS()

```
def pysar._readfile.read_GPS_USGS (
              File )
```

#### 17.10.1.12 read_isce_xml()

```
def pysar._readfile.read_isce_xml (
              File )
```

```
Read ISCE .xml file input a python dictionary structure.
```

#### 17.10.1.13 read_multiple()

```
def pysar._readfile.read_multiple (
              File,
              box = '' )
```

```
Read multi-temporal 2D datasets into a 3-D data stack
Inputs:
    File  : input file, interferograms,coherence, timeseries, ...
    box   : 4-tuple defining the left, upper, right, and lower pixel coordinate [optional]
Examples:
    stack = stacking('timeseries.h5',(100,1200,500,1500))
```

#### 17.10.1.14 read_real_float32()

```
def pysar._readfile.read_real_float32 (
              File )
```

```
Read real float 32 data matrix, i.e. GAMMA .mli file
Usage:  data, atr = read_real_float32('20070603.mli')
```

#### 17.10.1.15 read_real_int16()

```
def pysar._readfile.read_real_int16 (
              File )
```

```
Same as read_dem() above
```

#### 17.10.1.16 read_roipac_rsc()

```
def pysar._readfile.read_roipac_rsc (
              File )
```

```
Read ROI_PAC .rsc file into a python dictionary structure.
```

#### 17.10.1.17 read_template()

```
def pysar._readfile.read_template (
              File,
              delimiter = '=' )
```

```
Reads the template file into a python dictionary structure.
Input : string, full path to the template file
Output: dictionary, pysar template content
Example:
    tmpl = read_template(KyushuT424F610_640AlosA.template)
    tmpl = read_template(R1_54014_ST5_L0_F898.000.pi, ':')
```

### 17.10.2 Variable Documentation

#### 17.10.2.1 multi_dataset_hdf5_file

```
list multi_dataset_hdf5_file = ['timeseries']
```

#### 17.10.2.2 multi_group_hdf5_file

```
list multi_group_hdf5_file = ['interferograms','coherence','wrapped','snaphu_connect_component']
```

#### 17.10.2.3 single_dataset_hdf5_file

```
list single_dataset_hdf5_file = ['dem','mask','rmse','temporal_coherence', 'velocity']
```

## 17.11 pysar._remove_surface Namespace Reference

### Functions

- def remove_data_surface (data, mask, surf_type='plane')
- def remove_data_multiple_surface (data, mask, surf_type, ysub)
- def remove_surface (File, surf_type, maskFile=None, outFile=None, ysub=None)

### 17.11.1 Function Documentation

#### 17.11.1.1 remove_data_multiple_surface()

```
def pysar._remove_surface.remove_data_multiple_surface (
              data,
              mask,
              surf_type,
              ysub )
```

#### 17.11.1.2 remove_data_surface()

```
def pysar._remove_surface.remove_data_surface (
              data,
              mask,
              surf_type = 'plane' )
```

```
Remove surface from input data matrix based on pixel marked by mask
```

#### 17.11.1.3 remove_surface()

```
def pysar._remove_surface.remove_surface (
              File,
              surf_type,
              maskFile = None,
              outFile = None,
              ysub = None )
```

## 17.12 pysar._writefile Namespace Reference

### Functions

- def write (args)
- def write_float32 (args)
- def write_complex64 (data, outname)
- def write_real_int16 (data, outname)
- def write_dem (data, outname)
- def write_real_float32 (data, outname)
- def write_complex_int16 (data, outname)

### 17.12.1 Function Documentation

#### 17.12.1.1 write()

```
def pysar._writefile.write (
              args )
```

```
Write one dataset, i.e. interferogram, coherence, velocity, dem ...
    Return 0 if failed.

Usage:
    write(data,atr,outname)
    write(rg,az,atr,outname)

Inputs:
    data : 2D data matrix
    atr  : attribute object
    outname : output file name

Output:
    output file name

Examples:
    write(data,atr,'velocity.h5')
    write(data,atr,'temporal_coherence.h5')
    write(data,atr,'100120-110214.unw')
    write(data,atr,'strm1.dem')
    write(data,atr,'100120.mli')
    write(rg,az,atr,'geomap_4lks.trans')
```

#### 17.12.1.2 write_complex64()

```
def pysar._writefile.write_complex64 (
              data,
              outname )
```

```
Writes roi_pac .int data
```

#### 17.12.1.3 write_complex_int16()

```
def pysar._writefile.write_complex_int16 (
              data,
              outname )
```

```
Write gamma scomplex data, i.e. .slc file.
    data is complex 2-D matrix
    real, imagery, real, ...
```

#### 17.12.1.4 write_dem()

```
def pysar._writefile.write_dem (
              data,
              outname )
```

#### 17.12.1.5 write_float32()

```
def pysar._writefile.write_float32 (
              args )
```

```
Write ROI_PAC rmg format with float32 precision
Format of the binary file is same as roi_pac unw, cor, or hgt data.
      should rename to write_rmg_float32()

Exmaple:
        write_float32(phase, outname)
        write_float32(amp, phase, outname)
```

#### 17.12.1.6 write_real_float32()

```
def pysar._writefile.write_real_float32 (
              data,
              outname )
```

```
write gamma float data, i.e. .mli file.
```

#### 17.12.1.7 write_real_int16()

```
def pysar._writefile.write_real_int16 (
              data,
              outname )
```

# 17.13 pysar.add Namespace Reference

## Functions

- def add (data1, data2)
- def usage ()
- def main (argv)

### 17.13.1 Function Documentation

#### 17.13.1.1 add()

```
def pysar.add.add (
              data1,
              data2 )
```

#### 17.13.1.2 main()

```
def pysar.add.main (
              argv )
```

#### 17.13.1.3 usage()

```
def pysar.add.usage ( )
```

# 17.14 pysar.add_attribute Namespace Reference

## Functions

- def usage ()
- def main (argv)

### 17.14.1 Function Documentation

#### 17.14.1.1 main()

```
def pysar.add_attribute.main (
              argv )
```

#### 17.14.1.2 usage()

```
def pysar.add_attribute.usage ( )
```

## 17.15 pysar.add_attributes_insarmaps Namespace Reference

### Classes

- class InsarDatabaseController

### Functions

- def usage ()
- def parse_file_for_attributes (file)
- def build_parser ()
- def main (argv)

### 17.15.1 Function Documentation

#### 17.15.1.1 build_parser()

```
def pysar.add_attributes_insarmaps.build_parser ( )
```

#### 17.15.1.2 main()

```
def pysar.add_attributes_insarmaps.main (
              argv )
```

#### 17.15.1.3 parse_file_for_attributes()

```
def pysar.add_attributes_insarmaps.parse_file_for_attributes (
              file )
```

#### 17.15.1.4 usage()

```
def pysar.add_attributes_insarmaps.usage ( )
```

## 17.16 pysar.asc_desc Namespace Reference

### Functions

- def usage ()
- def corners (h5V1)
- def nearest_neighbor (x, y, tbase, pbase)
- def nearest (x, X)
- def find_row_column (Lon, Lat, h5file)
- def get_lat_lon (h5file)
- def main (argv)

### 17.16.1 Function Documentation

#### 17.16.1.1 corners()

```
def pysar.asc_desc.corners (
              h5V1 )
```

#### 17.16.1.2 find_row_column()

```
def pysar.asc_desc.find_row_column (
              Lon,
              Lat,
              h5file )
```

#### 17.16.1.3 get_lat_lon()

```
def pysar.asc_desc.get_lat_lon (
              h5file )
```

#### 17.16.1.4 main()

```
def pysar.asc_desc.main (
              argv )
```

#### 17.16.1.5 nearest()

```
def pysar.asc_desc.nearest (
              x,
              X )
```

find nearest neighbour

#### 17.16.1.6 nearest_neighbor()

```
def pysar.asc_desc.nearest_neighbor (
              x,
              y,
              tbase,
              pbase )
```

find nearest neighbour

#### 17.16.1.7 usage()

```
def pysar.asc_desc.usage ( )
```

## 17.17 pysar.baseline_error Namespace Reference

### Functions

- def to_percent (y, position)
- def usage ()
- def main (argv)

### 17.17.1 Function Documentation

#### 17.17.1.1 main()

```
def pysar.baseline_error.main (
              argv )
```

#### 17.17.1.2 to_percent()

```
def pysar.baseline_error.to_percent (
              y,
              position )
```

#### 17.17.1.3 usage()

```
def pysar.baseline_error.usage ( )
```

## 17.18 pysar.baseline_trop Namespace Reference

### Functions

- def to_percent (y, position)
- def usage ()
- def main (argv)

### 17.18.1 Function Documentation

#### 17.18.1.1 main()

```
def pysar.baseline_trop.main (
              argv )
```

#### 17.18.1.2 to_percent()

```
def pysar.baseline_trop.to_percent (
              y,
              position )
```

#### 17.18.1.3 usage()

```
def pysar.baseline_trop.usage ( )
```

## 17.19 pysar.convert2mat Namespace Reference

### Functions

- def usage ()
- def yyyymmdd2years (date)
- def main (argv)

### 17.19.1 Function Documentation

#### 17.19.1.1 main()

```
def pysar.convert2mat.main (
              argv )
```

#### 17.19.1.2 usage()

```
def pysar.convert2mat.usage ( )
```

#### 17.19.1.3 yyyymmdd2years()

```
def pysar.convert2mat.yyyymmdd2years (
              date )
```

## 17.20 pysar.correct_dem Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.20.1 Function Documentation

#### 17.20.1.1 main()

```
def pysar.correct_dem.main (
              argv )
```

#### 17.20.1.2 usage()

```
def pysar.correct_dem.usage ( )
```

## 17.21 pysar.correlation_with_dem Namespace Reference

### Functions

- def usage ()

### Variables

- amp
- dem = dem[int(suby[0]):int(suby[1]),int(subx[0]):int(subx[1])]
- demRsc
- h5data = h5py.File(File)
- dset = h5data['velocity'].get('velocity')
- data = dset[0:dset.shape[0],0:dset.shape[1]]
- suby = sys.argv[3].split(':')
- subx = sys.argv[4].split(':')
- ndx = ∼np.isnan(data)
- C1 = np.zeros([2,len(dem[ndx])])

### 17.21.1 Function Documentation

#### 17.21.1.1 usage()

```
def pysar.correlation_with_dem.usage ( )
```

### 17.21.2 Variable Documentation

#### 17.21.2.1 amp

```
amp
```

#### 17.21.2.2 C1

```
C1 = np.zeros([2,len(dem[ndx])])
```

#### 17.21.2.3 data

```
data = dset[0:dset.shape[0],0:dset.shape[1]]
```

#### 17.21.2.4 dem

```
dem = dem[int(suby[0]):int(suby[1]),int(subx[0]):int(subx[1])]
```

#### 17.21.2.5 demRsc

```
demRsc
```

#### 17.21.2.6 dset

```
dset = h5data['velocity'].get('velocity')
```

#### 17.21.2.7 h5data

```
h5data = h5py.File(File)
```

#### 17.21.2.8 ndx

```
ndx = ~np.isnan(data)
```

#### 17.21.2.9 subx

```
subx = sys.argv[4].split(':')
```

#### 17.21.2.10 suby

```
suby = sys.argv[3].split(':')
```

## 17.22 pysar.dem_error Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.22.1 Function Documentation

#### 17.22.1.1 main()

```
def pysar.dem_error.main (
              argv )
```

#### 17.22.1.2 usage()

```
def pysar.dem_error.usage ( )
```

## 17.23 pysar.diff Namespace Reference

### Functions

- def diff (data1, data2)
- def usage ()
- def main (argv)

### 17.23.1 Function Documentation

#### 17.23.1.1 diff()

```
def pysar.diff.diff (
              data1,
              data2 )
```

#### 17.23.1.2 main()

```
def pysar.diff.main (
              argv )
```

#### 17.23.1.3 usage()

```
def pysar.diff.usage ( )
```

## 17.24 pysar.drop_turbulence Namespace Reference

### Functions

- def circle_index (atr, circle_par)
- def usage ()

  *Usage ####################################.*
- def main (argv)

  *Main Function ################################.*

### 17.24.1 Function Documentation

#### 17.24.1.1 circle_index()

```
def pysar.drop_turbulence.circle_index (
              atr,
              circle_par )
```

#### 17.24.1.2 main()

```
def pysar.drop_turbulence.main (
              argv )
```

Main Function ################################.

#### 17.24.1.3 usage()

```
def pysar.drop_turbulence.usage ( )
```

Usage ####################################.

## 17.25 pysar.filter_spatial Namespace Reference

### Functions

- def usage ()
- def filter (data, filtType, par)
- def multilook (ifg, lksy, lksx)
- def main (argv)

### 17.25.1 Function Documentation

#### 17.25.1.1 filter()

```
def pysar.filter_spatial.filter (
              data,
              filtType,
              par )
```

#### 17.25.1.2 main()

```
def pysar.filter_spatial.main (
              argv )
```

#### 17.25.1.3 multilook()

```
def pysar.filter_spatial.multilook (
              ifg,
              lksy,
              lksx )
```

#### 17.25.1.4 usage()

```
def pysar.filter_spatial.usage ( )
```

## 17.26 pysar.filter_temporal Namespace Reference

### Functions

- def get_data (h5timeseries)
- def usage ()
- def main (argv)

### 17.26.1 Function Documentation

#### 17.26.1.1 get_data()

```
def pysar.filter_temporal.get_data (
              h5timeseries )
```

#### 17.26.1.2 main()

```
def pysar.filter_temporal.main (
              argv )
```

#### 17.26.1.3 usage()

```
def pysar.filter_temporal.usage ( )
```

## 17.27 pysar.gamma_view Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.27.1 Function Documentation

#### 17.27.1.1 main()

```
def pysar.gamma_view.main (
              argv )
```

#### 17.27.1.2 usage()

```
def pysar.gamma_view.usage ( )
```

## 17.28 pysar.generate_mask Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.28.1 Function Documentation

#### 17.28.1.1 main()

```
def pysar.generate_mask.main (
              argv )
```

#### 17.28.1.2 usage()

```
def pysar.generate_mask.usage ( )
```

## 17.29 pysar.geocode Namespace Reference

### Functions

- def geomap4subset_radar_file (radar_atr, geomap_file)
- def geocode_data_roipac (data, geomapFile, outname)

  *Geocode one data ########################.*
- def geocode_attribute (atr_rdr, atr_geo)
- def geocode_file_roipac (infile, geomap_file, outfile=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 17.29.1 Function Documentation

### 17.29.1.1 cmdLineParse()

```
def pysar.geocode.cmdLineParse ( )
```

### 17.29.1.2 geocode_attribute()

```
def pysar.geocode.geocode_attribute (
              atr_rdr,
              atr_geo )
```

### 17.29.1.3 geocode_data_roipac()

```
def pysar.geocode.geocode_data_roipac (
              data,
              geomapFile,
              outname )
```

Geocode one data ########################.

### 17.29.1.4 geocode_file_roipac()

```
def pysar.geocode.geocode_file_roipac (
              infile,
              geomap_file,
              outfile = None )
```

```
Geocode one file
```

#### 17.29.1.5 geomap4subset_radar_file()

```
def pysar.geocode.geomap4subset_radar_file (
              radar_atr,
              geomap_file )
```

Add offset value to geomap file if input radar file has been subsetted.

#### 17.29.1.6 main()

```
def pysar.geocode.main (
              argv )
```

### 17.29.2 Variable Documentation

#### 17.29.2.1 EXAMPLE

string EXAMPLE

**Initial value:**

```
= '''example:
  geocode.py  geomap_8rlks.trans  velocity.py
  geocode.py  geomap_8rlks.trans  *velocity*h5
  geocode.py  geomap_8rlks.trans  timeseries_ECMWF_demCor.h5 velocity_ex.h5
'''
```

## 17.30 pysar.igram_closure Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.30.1 Function Documentation

#### 17.30.1.1 main()

```
def pysar.igram_closure.main (
              argv )
```

#### 17.30.1.2 usage()

```
def pysar.igram_closure.usage ( )
```

## 17.31 pysar.igram_inversion Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.31.1 Function Documentation

#### 17.31.1.1 main()

```
def pysar.igram_inversion.main (
              argv )
```

#### 17.31.1.2 usage()

```
def pysar.igram_inversion.usage ( )
```

## 17.32 pysar.image_math Namespace Reference

### Functions

- def operation (data, operator, operand)
  *Sub Functions ######################## Operation ####################.*
- def add (data1, data2)
  *Image Add ####################.*
- def diff (data1, data2)
  *Image Diff ####################.*
- def usage ()
  *Usage #####################.*
- def main (argv)
  *Main Functions #######################.*

### 17.32.1 Function Documentation

#### 17.32.1.1 add()

```
def pysar.image_math.add (
              data1,
              data2 )
```

Image Add ####################.

#### 17.32.1.2 diff()

```
def pysar.image_math.diff (
              data1,
              data2 )
```

Image Diff ####################.

#### 17.32.1.3 main()

```
def pysar.image_math.main (
              argv )
```

Main Functions ########################.

#### 17.32.1.4 operation()

```
def pysar.image_math.operation (
              data,
              operator,
              operand )
```

Sub Functions ########################## Operation ####################.

#### 17.32.1.5 usage()

```
def pysar.image_math.usage ( )
```

Usage ######################.

## 17.33 pysar.incidence_angle Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.33.1 Function Documentation

#### 17.33.1.1 main()

```
def pysar.incidence_angle.main (
              argv )
```

#### 17.33.1.2 usage()

```
def pysar.incidence_angle.usage ( )
```

## 17.34 pysar.info Namespace Reference

### Functions

- def print_attributes (atr, sorting=True)
- def print_hdf5_structure (File)

  *By andrewcollette at* https://github.com/h5py/h5py/issues/406.
- def print_timseries_date_info (dateList)
- def usage ()
- def main (argv)

### 17.34.1 Function Documentation

#### 17.34.1.1 main()

```
def pysar.info.main (
              argv )
```

#### 17.34.1.2 print_attributes()

```
def pysar.info.print_attributes (
              atr,
              sorting = True )
```

#### 17.34.1.3 print_hdf5_structure()

```
def pysar.info.print_hdf5_structure (
              File )
```

By andrewcollette at https://github.com/h5py/h5py/issues/406.

#### 17.34.1.4 print_timseries_date_info()

```
def pysar.info.print_timseries_date_info (
              dateList )
```

#### 17.34.1.5 usage()

```
def pysar.info.usage ( )
```

## 17.35 pysar.insar_vs_gps Namespace Reference

### Functions

- def readGPSfile (gpsFile, gps_source)
- def nearest (x, tbase, xstep)
- def find_row_column (Lon, Lat, lon, lat, lon_step, lat_step)
- def usage ()
- def main (argv)

## Variables

- Stations

  *finding the raw an column of the reference gps station and referencing insar data to this pixel*
- Lat
- Lon
- Ve
- Se
- Vn
- Sn
- Vu
- Su
- idxRef = Stations.index(refStation)
- IDYref
- IDXref
- insarData = insarData - insarData[IDYref][IDXref]

  *Stations, gpsData = redGPSfile(gpsFile) idxRef=Stations.index(refStation) Lat,Lon,Vn,Ve,Sn,Se,Corr,Vu,Su = gpsData[idxRef,:] IDYref,IDXref=find_row_column(Lon,Lat,lon,lat,lon_step,lat_step)*
- stationsList = Stations
- look_n = float(h5file['velocity'].attrs['LOOK_REF1'])
- look_f = float(h5file['velocity'].attrs['LOOK_REF2'])
- tuple theta = (look_n+look_f)/2.
- heading = float(h5file['velocity'].attrs['HEADING'])
- list unitVec = [np.cos(heading)∗np.sin(theta),-np.sin(theta)∗np.sin(heading),-np.cos(theta)]
- string gps_comp_txt = ' projecting three gps components to LOS'
- list gpsLOS_ref = unitVec[0]∗Ve[idxRef]+unitVec[1]∗Vn[idxRef]+unitVec[2]∗Vu[idxRef]
- tuple Sr = ((unitVec[0]∗∗2)∗Se[idxRef]∗∗2+(unitVec[1]∗∗2)∗Sn[idxRef]∗∗2+(unitVec[2]∗∗2)∗Su[idxRef]∗∗2)∗∗0.5
- h5coh = h5py.File(coherenceFile)
- kh5coh = h5coh.keys()
- dset = h5coh[kh5coh[0]].get(kh5coh[0])
- Coh = dset[0:dset.shape[0],0:dset.shape[1]]
- list InSAR = [ ]
- list GPS = [ ]
- list InSAR1 = [ ]
- list GPS1 = [ ]
- list InSAR2 = [ ]
- list GPS2 = [ ]
- list coherence = [ ]
- list GPSx = [ ]
- list GPSy = [ ]
- list GPSx1 = [ ]
- list GPSy1 = [ ]
- list GPSx2 = [ ]
- list GPSy2 = [ ]
- list GPS_station = [ ]
- list GPS_std = [ ]
- idx = Stations.index(st)
- list gpsLOS = unitVec[0]∗Ve[idx]+unitVec[1]∗Vn[idx]+unitVec[2]∗Vu[idx]
- tuple Sg = ((unitVec[0]∗∗2)∗Se[idx]∗∗2+(unitVec[1]∗∗2)∗Sn[idx]∗∗2+(unitVec[2]∗∗2)∗Su[idx]∗∗2)∗∗0.5
- tuple S = (Sg∗∗2+Sr∗∗2)∗∗0.5
- IDY
- IDX
- insar_velocity = -insarData[IDY][IDX]
- string InSAR_GPS_Copmarison = 'yes'

- string NoInSAR = 'yes'
- lt = len(InSAR)
- SAD = np.sum(np.abs(InSAR-GPS),0)/lt
- C1 = np.zeros([2,len(InSAR)])
- Cor = np.corrcoef(C1)[0][1]
- minV = np.min([InSAR,GPS])
- maxV = np.max([InSAR,GPS])
- fig = plt.figure()
- ax = fig.add_subplot(111)
- yerr
- xerr
- fmt
- ms
- fontsize
- xy
- xytext
- color
- majorLocator = MultipleLocator(5)
- minorLocator = MultipleLocator(1)
- which
- length
- width
- string figName = 'InSARvsGPS_errorbar.png'

## 17.35.1 Function Documentation

### 17.35.1.1 find_row_column()

```
def pysar.insar_vs_gps.find_row_column (
              Lon,
              Lat,
              lon,
              lat,
              lon_step,
              lat_step )
```

### 17.35.1.2 main()

```
def pysar.insar_vs_gps.main (
              argv )
```

#### 17.35.1.3 nearest()

```
def pysar.insar_vs_gps.nearest (
              x,
              tbase,
              xstep )
```

#### 17.35.1.4 readGPSfile()

```
def pysar.insar_vs_gps.readGPSfile (
              gpsFile,
              gps_source )
```

#### 17.35.1.5 usage()

```
def pysar.insar_vs_gps.usage ( )
```

### 17.35.2 Variable Documentation

#### 17.35.2.1 ax

```
ax = fig.add_subplot(111)
```

#### 17.35.2.2 C1

```
C1 = np.zeros([2,len(InSAR)])
```

#### 17.35.2.3 Coh

```
Coh = dset[0:dset.shape[0],0:dset.shape[1]]
```

#### 17.35.2.4 coherence

```
list coherence = []
```

#### 17.35.2.5 color

```
color
```

#### 17.35.2.6 Cor

```
Cor = np.corrcoef(C1)[0][1]
```

#### 17.35.2.7 dset

```
dset = h5coh[kh5coh[0]].get(kh5coh[0])
```

#### 17.35.2.8 fig

```
fig = plt.figure()
```

#### 17.35.2.9 figName

```
string figName = 'InSARvsGPS_errorbar.png'
```

#### 17.35.2.10 fmt

```
fmt
```

#### 17.35.2.11 fontsize

```
fontsize
```

#### 17.35.2.12 GPS

```
GPS = [ ]
```

#### 17.35.2.13 GPS1

```
GPS1 = [ ]
```

#### 17.35.2.14 GPS2

```
GPS2 = [ ]
```

#### 17.35.2.15 gps_comp_txt

```
string gps_comp_txt = ' projecting three gps components to LOS'
```

#### 17.35.2.16 GPS_station

```
list GPS_station = [ ]
```

#### 17.35.2.17 GPS_std

```
GPS_std = [ ]
```

#### 17.35.2.18 gpsLOS

```
float gpsLOS = unitVec[0]*Ve[idx]+unitVec[1]*Vn[idx]+unitVec[2]*Vu[idx]
```

#### 17.35.2.19 gpsLOS_ref

```
list gpsLOS_ref = unitVec[0]*Ve[idxRef]+unitVec[1]*Vn[idxRef]+unitVec[2]*Vu[idxRef]
```

#### 17.35.2.20 GPSx

```
list GPSx = []
```

#### 17.35.2.21 GPSx1

```
list GPSx1 = []
```

#### 17.35.2.22 GPSx2

```
list GPSx2 = []
```

#### 17.35.2.23 GPSy

```
list GPSy = []
```

#### 17.35.2.24 GPSy1

```
list GPSy1 = []
```

#### 17.35.2.25 GPSy2

```
list GPSy2 = []
```

#### 17.35.2.26 h5coh

```
h5coh = h5py.File(coherenceFile)
```

#### 17.35.2.27 heading

```
float heading = float(h5file['velocity'].attrs['HEADING'])
```

#### 17.35.2.28 idx

```
idx = Stations.index(st)
```

#### 17.35.2.29 IDX

```
IDX
```

#### 17.35.2.30 idxRef

```
idxRef = Stations.index(refStation)
```

#### 17.35.2.31 IDXref

```
IDXref
```

#### 17.35.2.32 IDY

```
IDY
```

#### 17.35.2.33 IDYref

```
IDYref
```

#### 17.35.2.34 InSAR

```
InSAR = []
```

#### 17.35.2.35 InSAR1

```
InSAR1 = []
```

#### 17.35.2.36 InSAR2

```
InSAR2 = [ ]
```

#### 17.35.2.37 InSAR_GPS_Copmarison

```
string InSAR_GPS_Copmarison = 'yes'
```

#### 17.35.2.38 insar_velocity

```
insar_velocity = -insarData[IDY][IDX]
```

#### 17.35.2.39 insarData

```
insarData = insarData - insarData[IDYref][IDXref]
```

Stations, gpsData = redGPSfile(gpsFile) idxRef=Stations.index(refStation) Lat,Lon,Vn,Ve,Sn,Se,Corr,Vu,Su = gps↵ Data[idxRef,:] IDYref,IDXref=find_row_column(Lon,Lat,lon,lat,lon_step,lat_step)

#### 17.35.2.40 kh5coh

```
kh5coh = h5coh.keys()
```

#### 17.35.2.41 Lat

```
Lat
```

#### 17.35.2.42 length

```
length
```

#### 17.35.2.43 Lon

```
Lon
```

#### 17.35.2.44 look_f

```
look_f = float(h5file['velocity'].attrs['LOOK_REF2'])
```

#### 17.35.2.45 look_n

```
look_n = float(h5file['velocity'].attrs['LOOK_REF1'])
```

#### 17.35.2.46 lt

```
lt = len(InSAR)
```

#### 17.35.2.47 majorLocator

```
majorLocator = MultipleLocator(5)
```

#### 17.35.2.48 maxV

```
maxV = np.max([InSAR,GPS])
```

#### 17.35.2.49 minorLocator

```
minorLocator = MultipleLocator(1)
```

#### 17.35.2.50 minV

```
minV = np.min([InSAR,GPS])
```

#### 17.35.2.51 ms

```
ms
```

#### 17.35.2.52 NoInSAR

```
string NoInSAR = 'yes'
```

#### 17.35.2.53 S

```
tuple S = (Sg**2+Sr**2)**0.5
```

#### 17.35.2.54 SAD

```
SAD = np.sum(np.abs(InSAR-GPS),0)/lt
```

#### 17.35.2.55 Se

```
Se
```

#### 17.35.2.56 Sg

```
tuple Sg = ((unitVec[0]**2)*Se[idx]**2+(unitVec[1]**2)*Sn[idx]**2+(unitVec[2]**2)*Su[idx]**2)**0.↩
5
```

#### 17.35.2.57 Sn

```
Sn
```

#### 17.35.2.58 Sr

```
tuple Sr = ((unitVec[0]**2)*Se[idxRef]**2+(unitVec[1]**2)*Sn[idxRef]**2+(unitVec[2]**2)*Su[idx←
Ref]**2)**0.5
```

#### 17.35.2.59 Stations

```
Stations
```

finding the raw an column of the reference gps station and referencing insar data to this pixel

#### 17.35.2.60 stationsList

```
stationsList = Stations
```

#### 17.35.2.61 Su

```
Su
```

#### 17.35.2.62 theta

```
tuple theta = (look_n+look_f)/2.
```

#### 17.35.2.63 unitVec

```
list unitVec = [np.cos(heading)*np.sin(theta),-np.sin(theta)*np.sin(heading),-np.cos(theta)]
```

#### 17.35.2.64 Ve

```
Ve
```

#### 17.35.2.65 Vn

Vn

#### 17.35.2.66 Vu

Vu

#### 17.35.2.67 which

which

#### 17.35.2.68 width

width

#### 17.35.2.69 xerr

xerr

#### 17.35.2.70 xy

xy

#### 17.35.2.71 xytext

xytext

#### 17.35.2.72 yerr

yerr

## 17.36 pysar.insarmaps_query Namespace Reference

### Classes

- class BasicHTTP

### Functions

- def buildURL (args)
- def build_parser ()
- def main ()

### 17.36.1 Function Documentation

#### 17.36.1.1 build_parser()

```
def pysar.insarmaps_query.build_parser ( )
```

#### 17.36.1.2 buildURL()

```
def pysar.insarmaps_query.buildURL (
              args )
```

#### 17.36.1.3 main()

```
def pysar.insarmaps_query.main ( )
```

## 17.37 pysar.l1 Namespace Reference

### Functions

- def l1mosek (P, q)
- def l1mosek2 (P, q)
- def l1 (P, q)
- def l1blas (P, q)

### Variables

- bool __MOSEK = True
- task = env.Task(0,0)
- x = zeros(n, float)

## 17.37.1 Function Documentation

#### 17.37.1.1 l1()

```
def pysar.l1.l1 (
              P,
              q )
```

```
Returns the solution u of the ell-1 approximation problem

    (primal) minimize ||P*u - q||_1

    (dual)   maximize    q'*w
             subject to  P'*w = 0
                         ||w||_infty <= 1.
```

#### 17.37.1.2 l1blas()

```
def pysar.l1.l1blas (
              P,
              q )
```

```
Returns the solution u of the ell-1 approximation problem

    (primal) minimize ||P*u - q||_1

    (dual)   maximize    q'*w
             subject to  P'*w = 0
                         ||w||_infty <= 1.
```

#### 17.37.1.3 l1mosek()

```
def pysar.l1.l1mosek (
              P,
              q )
```

```
minimize    e'*v

subject to  P*u - v <=  q
           -P*u - v <= -q
```

#### 17.37.1.4 l1mosek2()

```
def pysar.l1.l1mosek2 (
              P,
              q )
```

```
minimize    e'*s + e'*t

subject to  P*u - q = s - t
            s, t >= 0
```

### 17.37.2 Variable Documentation

#### 17.37.2.1 __MOSEK

```
__MOSEK = True  [private]
```

#### 17.37.2.2 task

```
task = env.Task(0,0)
```

#### 17.37.2.3 x

```
x = zeros(n, float)
```

## 17.38 pysar.load_data Namespace Reference

### Functions

- def auto_path_miami (inps, template_dict={})
  *Sub Functions ####################################.*
- def mode (thelist)
  *Find Mode (most common) item in the list #############.*
- def check_file_size (fileList, mode_width=None, mode_length=None)
- def check_existed_hdf5_file (roipacFileList, hdf5File)
- def load_roipac2multi_group_h5 (fileType, fileList, hdf5File='unwrapIfgram.h5', pysar_meta_dict=None)
- def roipac_nonzero_mask (unwFileList, maskFile='Mask.h5')
- def copy_roipac_file (targetFile, destDir)
- def cmdLineParse ()
- def main (argv)
  *Main Function ################################.*

## Variables

- string EXAMPLE

  *Usage ###############################.*
- string TEMPLATE

### 17.38.1 Function Documentation

#### 17.38.1.1 auto_path_miami()

```
def pysar.load_data.auto_path_miami (
              inps,
              template_dict = {} )
```

Sub Functions ####################################.

```
Auto File Path Setting for Geodesy Lab - University of Miami
```

#### 17.38.1.2 check_existed_hdf5_file()

```
def pysar.load_data.check_existed_hdf5_file (
              roipacFileList,
              hdf5File )
```

```
Check file list with existed hdf5 file
```

#### 17.38.1.3 check_file_size()

```
def pysar.load_data.check_file_size (
              fileList,
              mode_width = None,
              mode_length = None )
```

```
Update file list and drop those not in the same size with majority.
```

#### 17.38.1.4 cmdLineParse()

```
def pysar.load_data.cmdLineParse ( )
```

#### 17.38.1.5 copy_roipac_file()

```
def pysar.load_data.copy_roipac_file (
              targetFile,
              destDir )
```

```
Copy ROI_PAC file and its .rsc file to destination directory.
```

#### 17.38.1.6 load_roipac2multi_group_h5()

```
def pysar.load_data.load_roipac2multi_group_h5 (
              fileType,
              fileList,
              hdf5File = 'unwrapIfgram.h5',
              pysar_meta_dict = None )
```

```
Load multiple ROI_PAC product into (Multi-group, one dataset and one attribute dict per group) HDF5 file.
Inputs:
    fileType : string, i.e. interferograms, coherence, snaphu_connect_component, etc.
    fileList : list of path, ROI_PAC .unw/.cor/.int/.byt file
    hdf5File : string, file name/path of the multi-group hdf5 PySAR file
    pysar_meta_dict : dict, extra attribute dictionary
Outputs:
    hdf5File
```

#### 17.38.1.7 main()

```
def pysar.load_data.main (
              argv )
```

Main Function ##################################.

#### 17.38.1.8 mode()

```
def pysar.load_data.mode (
              thelist )
```

Find Mode (most common) item in the list #############.

#### 17.38.1.9 roipac_nonzero_mask()

```
def pysar.load_data.roipac_nonzero_mask (
              unwFileList,
              maskFile = 'Mask.h5' )
```

Generate mask for non-zero amplitude pixel of ROI_PAC .unw file list.

### 17.38.2 Variable Documentation

#### 17.38.2.1 EXAMPLE

string EXAMPLE

**Initial value:**

```
= '''example:
  load_data_roipac.py  $TE/SanAndreasT356EnvD.template
  load_data_roipac.py  $TE/SanAndreasT356EnvD.template  --dir $SC/SanAndreasT356EnvD/TIMESERIES
'''
```

Usage ################################.

#### 17.38.2.2 TEMPLATE

string TEMPLATE

**Initial value:**

```
= '''template:
  pysar.unwrapFiles    = $SC/SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*.unw
  pysar.corFiles       = $SC/SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.cor
  pysar.wrapFiles      = $SC/SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.int
     #optional
  pysar.geomap         = $SC/SanAndreasT356EnvD/PROCESS/GEO/*050102-070809*/geomap*.trans
  pysar.dem.radarCoord = $SC/SanAndreasT356EnvD/PROCESS/DONE/*050102-070809*/radar*.hgt
  pysar.dem.geoCoord   = $SC/SanAndreasT356EnvD/DEM/srtm1_30m.dem
     #optional
'''
```

## 17.39 pysar.load_dem Namespace Reference

### Variables

- demFile = sys.argv[1]
- ext = os.path.splitext(demFile)[1]
- amp
- dem
- demRsc
- outName
- h5 = h5py.File(outName,'w')
- group = h5.create_group('dem')
- dset = group.create_dataset('dem', data=dem, compression='gzip')

### 17.39.1 Variable Documentation

#### 17.39.1.1 amp

```
amp
```

#### 17.39.1.2 dem

```
dem
```

#### 17.39.1.3 demFile

```
demFile = sys.argv[1]
```

#### 17.39.1.4 demRsc

```
demRsc
```

#### 17.39.1.5 dset

```
dset = group.create_dataset('dem', data=dem, compression='gzip')
```

#### 17.39.1.6 ext

```
ext = os.path.splitext(demFile)[1]
```

#### 17.39.1.7 group

```
group = h5.create_group('dem')
```

#### 17.39.1.8 h5

```
h5 = h5py.File(outName,'w')
```

#### 17.39.1.9 outName

```
outName
```

## 17.40 pysar.lod Namespace Reference

### Functions

- def correct_lod_file (File, outFile=None)
- def usage ()
- def main (argv)

### 17.40.1 Function Documentation

#### 17.40.1.1 correct_lod_file()

```
def pysar.lod.correct_lod_file (
              File,
              outFile = None )
```

#### 17.40.1.2 main()

```
def pysar.lod.main (
              argv )
```

#### 17.40.1.3 usage()

```
def pysar.lod.usage ( )
```

## 17.41 pysar.look_angle Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.41.1 Function Documentation

#### 17.41.1.1 main()

```
def pysar.look_angle.main (
              argv )
```

#### 17.41.1.2 usage()

```
def pysar.look_angle.usage ( )
```

## 17.42 pysar.los2enu Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.42.1 Function Documentation

#### 17.42.1.1 main()

```
def pysar.los2enu.main (
              argv )
```

#### 17.42.1.2 usage()

```
def pysar.los2enu.usage ( )
```

## 17.43 pysar.mask Namespace Reference

### Functions

- def mask_matrix (data_mat, mask_mat)
- def update_mask (mask, inps_dict=None)
- def mask_file (File, maskFile, outFile=None, inps_dict=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

### 17.43.1 Function Documentation

#### 17.43.1.1 cmdLineParse()

```
def pysar.mask.cmdLineParse ( )
```

#### 17.43.1.2 main()

```
def pysar.mask.main (
              argv )
```

#### 17.43.1.3 mask_file()

```
def pysar.mask.mask_file (
              File,
              maskFile,
              outFile = None,
              inps_dict = None )
```

```
Mask input File with maskFile
Inputs:
    File/maskFile - string,
    inps_dict - dictionary including the following options:
                subset_x/y - list of 2 ints, subset in x/y direction
                thr - float, threshold/minValue to generate mask
Output:
    outFile - string
```

#### 17.43.1.4 mask_matrix()

```
def pysar.mask.mask_matrix (
              data_mat,
              mask_mat )
```

mask a 2D matrxi data with mask

#### 17.43.1.5 update_mask()

```
def pysar.mask.update_mask (
              mask,
              inps_dict = None )
```

Update mask matrix from input options: subset_x/y and threshold

### 17.43.2 Variable Documentation

#### 17.43.2.1 EXAMPLE

string EXAMPLE

**Initial value:**

```
= '''example:
  mask.py  velocity.h5     -m Mask.h5
  mask.py  timeseries.h5   -m temporal_coherence.h5  -t 0.7
  mask.py  unwrapIfgram.h5 -m 100102_101120.cor      -t 0.9  -y  200 300  -x 300 400
  mask.py  timeseries*.h5 velocity*.h5  -m temporal_coherence.h5  -t 0.7
'''
```

## 17.44 pysar.match Namespace Reference

### Functions

- def corners (atr)
- def nearest (x, X)
- def manual_offset_estimate (matrix1, matrix2)
- def match_two_files (File1, File2, outName=None, manual_match=False, disp_fig=False)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 17.44.1 Function Documentation

### 17.44.1.1 cmdLineParse()

```
def pysar.match.cmdLineParse ( )
```

### 17.44.1.2 corners()

```
def pysar.match.corners (
              atr )
```

```
Get corners coordinate.
```

### 17.44.1.3 main()

```
def pysar.match.main (
              argv )
```

### 17.44.1.4 manual_offset_estimate()

```
def pysar.match.manual_offset_estimate (
              matrix1,
              matrix2 )
```

```
Manually estimate offset between two data matrix.
By manually selecting a line from each of them, and estimate the difference.
It usually used when 2 input data matrix have no area in common.
```

#### 17.44.1.5 match_two_files()

```
def pysar.match.match_two_files (
              File1,
              File2,
              outName = None,
              manual_match = False,
              disp_fig = False )
```

```
Match two geocoded files by estimating their offset.
Better for two files with common area overlaping.
```

#### 17.44.1.6 nearest()

```
def pysar.match.nearest (
              x,
              X )
```

```
find nearest neighbour
```

### 17.44.2 Variable Documentation

#### 17.44.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  match.py  vel_AlosAT42*.h5
  match.py  vel_AlosAT42*.h5  -o vel_AlosA.h5
  match.py  vel_AlosAT422.h5  vel_AlosAT423.h5  vel_AlosAT424.h5  vel_AlosAT425.h5
  match.py  vel_AlosAT422.h5  vel_AlosAT423.h5
  match.py  vel_AlosAT422.h5  vel_AlosAT423.h5  --manual
'''
```

## 17.45 pysar.mean_spatial Namespace Reference

### Functions

- def circle_index (atr, circle_par)
- def Usage ()

  *Usage ####################################.*
- def main (argv)

  *Main Function ################################.*

### 17.45.1 Function Documentation

#### 17.45.1.1 circle_index()

```
def pysar.mean_spatial.circle_index (
              atr,
              circle_par )
```

#### 17.45.1.2 main()

```
def pysar.mean_spatial.main (
              argv )
```

Main Function #################################.

#### 17.45.1.3 Usage()

```
def pysar.mean_spatial.Usage ( )
```

Usage #####################################.

## 17.46 pysar.modify_network Namespace Reference

### Functions

- def nearest_neighbor (x, y, x_array, y_array)

  *Sub Function #############################.*
- def manual_select_pairs_to_remove (File)
- def update_inps_with_template (inps, template_file)
- def modify_file_date12_list (File, date12_to_rmv, outFile=None)
- def cmdLineParse ()
- def main (argv)

  *Main Function ##############################.*

### Variables

- string EXAMPLE

  *Usage ################################.*
- string TEMPLATE

### 17.46.1 Function Documentation

#### 17.46.1.1 cmdLineParse()

```
def pysar.modify_network.cmdLineParse ( )
```

#### 17.46.1.2 main()

```
def pysar.modify_network.main (
              argv )
```

Main Function ###############################.

#### 17.46.1.3 manual_select_pairs_to_remove()

```
def pysar.modify_network.manual_select_pairs_to_remove (
              File )
```

```
Manually select interferograms to remove
```

#### 17.46.1.4 modify_file_date12_list()

```
def pysar.modify_network.modify_file_date12_list (
              File,
              date12_to_rmv,
              outFile = None )
```

```
Update multiple group hdf5 file using date12 to remove/keep
```

#### 17.46.1.5 nearest_neighbor()

```
def pysar.modify_network.nearest_neighbor (
              x,
              y,
              x_array,
              y_array )
```

Sub Function ##############################.

```
find nearest neighbour
Input:
    x/y       : float
    x/y_array : numpy.array, temporal/perpendicular spatial baseline
Output:
    idx : int, index of min distance - nearest neighbour
```

#### 17.46.1.6 update_inps_with_template()

```
def pysar.modify_network.update_inps_with_template (
              inps,
              template_file )
```

### 17.46.2 Variable Documentation

#### 17.46.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  modify_network.py unwrapIfgram.h5 coherence.h5 --template KyushuT422F650AlosA.template
  modify_network.py unwrapIfgram.h5 coherence.h5 -t 365 -b 200
  modify_network.py unwrapIfgram.h5 coherence.h5 --coherence-base coherence.h5 --mask Mask.h5
     --min-coherence 0.7
  modify_network.py unwrapIfgram.h5 -r Modified_coherence.h5
  modify_network.py unwrapIfgram.h5 --drop-date 20080520 20090816
  modify_network.py unwrapIfgram.h5 --drop-ifg-index 3:9 11 23
  modify_network.py unwrapIfgram.h5 --manual
'''
```

Usage #################################.

#### 17.46.2.2 TEMPLATE

```
string TEMPLATE
```

**Initial value:**

```
= '''
pysar.network.dropIfgramIndex = 7:9 15 25 26       #start from 1
pysar.network.dropDate        = 20080520 20090816
pysar.network.maxTempBaseline = 720
pysar.network.maxPerpBaseline = 2000
pysar.network.reference       = Modified_unwrapIfgram.h5
pysar.network.reference       = Paris.list
pysar.network.coherenceBase   = yes     #search and use input coherence file, set to no or comment the line
       to disable
'''
```

## 17.47 pysar.multi_transect Namespace Reference

### Functions

- def usage ()
- def dms2d (Coord)
- def gps_to_LOS (Ve, Vn, theta, heading)
- def check_st_in_box (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def check_st_in_box2 (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def line (x0, y0, x1, y1)
- def dist_point_from_line (m, c, x, y, dx, dy)
- def get_intersect (m, c, x, y)
- def readGPSfile (gpsFile, gps_source)
- def redGPSfile (gpsFile)
- def redGPSfile_cmm4 (gpsFile)
- def nearest (x, tbase, xstep)
- def find_row_column (Lon, Lat, lon, lat, lon_step, lat_step)
- def get_lat_lon (h5file)
- def nanmean (data, args)
- def nanstd (data, args)
- def get_transect (z, x0, y0, x1, y1)
- def get_start_end_point (Xf0, Yf0, Xf1, Yf1, L, dx, dy)
- def point_with_distance_from_line (Xf0, Yf0, Xf1, Yf1, L)
- def point_on_line_with_distance_from_beginning (Xf0, Yf0, Xf1, Yf1, L)
- def read_fault_coords (Fault_coord_file, Dp)
- def main (argv)
- def onclick (event)

### Variables

- lat
- lon
- lat_step
- lon_step
- lat_all
- lon_all
- Fault_lon
- Fault_lat
- int Num_profiles = len(Fault_lon)-1
- list FaultCoords = [Fault_lat[Np],Fault_lon[Np],Fault_lat[Np+1],Fault_lon[Np+1]]
- list Lat0 = FaultCoords[1]
- list Lat1 = FaultCoords[3]
- Length
- Width
- Yf0
- Xf0
- Yf1
- Xf1
- y0 = yc[1]
- x0 = xc[1]
- y1
- x1
- fig = plt.figure()
- ax = fig.add_subplot(111)
- list xc = [ ]
- list yc = [ ]
- cid = fig.canvas.mpl_connect('button_press_event', onclick)
- length = int(np.hypot(x1-x0, y1-y0))

  *try: mf=float(Yf1-Yf0)/float((Xf1-Xf0)) # slope of the fault line cf=float(Yf0-mf∗Xf0) # intercept of the fault line df0=dist←↩ _point_from_line(mf,cf,x0,y0,1,1) #distance of the profile start point from the Fault line df1=dist_point_from_←↩ line(mf,cf,x1,y1,1,1) #distance of the profile end point from the Fault line*
- x
- y
- zi = z[y.astype(np.int), x.astype(np.int)]
- lat_transect = lat_all[y.astype(np.int), x.astype(np.int)]
- lon_transect = lon_all[y.astype(np.int), x.astype(np.int)]
- float dx = float(h5file[k[0]].attrs['X_STEP'])∗6375000.0∗np.pi/180.0
- float dy = float(h5file[k[0]].attrs['Y_STEP'])∗6375000.0∗np.pi/180.0
- tuple DX = (x-x0)∗dx
- tuple DY = (y-y0)∗dy
- D = np.hypot(DX, DY)
- mf
- cf
- def df0_km = dist_point_from_line(mf,cf,x0,y0,dx,dy)
- transect = np.zeros([len(D),ntrans])
- list XX0 = [ ]
- list YY0 = [ ]
- m = float(y1-y0)/float((x1-x0))
- c = float(y0-m∗x0)
- float m1 = -1.0/m
- float dp = 1.0
- float X0 = i∗dp/np.sqrt(1+m1∗∗2)+x0
- float Y0 = m1∗(X0-x0)+y0

- float X1 = i∗dp/np.sqrt(1+m1∗∗2)+x1
- float Y1 = m1∗(X1-x1)+y1
- transect_lat = np.zeros([len(D),ntrans])
- transect_lon = np.zeros([len(D),ntrans])
- m_prof_edge
- c_prof_edge
- string gpsFile = 'Nogps'
- insarData = z
- fileName
- fileExtension
- Stations
- Lat
- Lon
- Ve
- Se
- Vn
- Sn
- idxRef = Stations.index(refStation)
- IDYref
- IDXref
- stationsList = Stations
- h5file_theta = h5py.File(incidence_file,'r')
- dset = h5file_theta['mask'].get('mask')
- theta = dset[0:dset.shape[0],0:dset.shape[1]]
- float heading = 193.0∗np.pi/180.0
- list unitVec = [np.cos(heading)∗np.sin(theta),-np.sin(theta)∗np.sin(heading),0]
- def gpsLOS_ref = gps_to_LOS(Ve[idxRef],Vn[idxRef],theta[IDYref,IDXref],heading)
- list GPS = [ ]
- list GPS_station = [ ]
- list GPSx = [ ]
- list GPSy = [ ]
- list GPS_lat = [ ]
- list GPS_lon = [ ]
- idx = Stations.index(st)
- IDY
- IDX
- def gpsLOS = gps_to_LOS(Ve[idx],Vn[idx],theta[IDY,IDX],heading)
- string NoInSAR = 'yes'
- list DistGPS = [ ]
- list GPS_in_bound = [ ]
- list GPS_in_bound_st = [ ]
- list GPSxx = [ ]
- list GPSyy = [ ]
- list gx = GPSx[i]
- list gy = GPSy[i]
- string check_result = 'True'
- def check_result2 = check_st_in_box2(gx,gy,x0,y0,x1,y1,X0,Y0,X1,Y1)
- def dg = dist_point_from_line(m,c,gx,gy,1,1)
- axes
- nrows
- ms

  *ax.fill_between(D/1000.0, (avgInSAR-stdInSAR)∗1000, (avgInSAR+stdInSAR)∗1000,where=(avgInSAR+stdInS←AR)∗1000>=(avgInSAR-stdInSAR)∗1000,alpha=1, facecolor='Red')*
- avgInSAR = np.array(nanmean(transect,axis=1))

- stdInSAR = np.array(nanstd(transect,axis=1))
- fig2
- axes2
- string FaultLine = 'None'
- string figName = 'transect_area_'+str(Np)+'.png'

  *Temporary To plot DEM try: majorLocator = MultipleLocator(5) ax.yaxis.set_major_locator(majorLocator) minor↩ Locator = MultipleLocator(1) ax.yaxis.set_minor_locator(minorLocator)*
- mfc
- linewidth
- string matFile = 'transect'+str(Np)+'.mat'
- dictionary dataset = {}
- color

  *ax.plot(D/1000.0, avgInSAR∗1000, 'r-')*
- alpha
- fontsize
- int lbound = np.nanmin(transect)∗1000

  *lower and higher bounds for diplaying the profile*
- int hbound = np.nanmax(transect)∗1000
- string ylim = 'no'
- string xlim = 'no'

### 17.47.1 Function Documentation

#### 17.47.1.1 check_st_in_box()

```
def pysar.multi_transect.check_st_in_box (
              x,
              y,
              x0,
              y0,
              x1,
              y1,
              X0,
              Y0,
              X1,
              Y1 )
```

#### 17.47.1.2 check_st_in_box2()

```
def pysar.multi_transect.check_st_in_box2 (
              x,
              y,
              x0,
              y0,
              x1,
              y1,
              X0,
              Y0,
              X1,
              Y1 )
```

#### 17.47.1.3 dist_point_from_line()

```
def pysar.multi_transect.dist_point_from_line (
              m,
              c,
              x,
              y,
              dx,
              dy )
```

#### 17.47.1.4 dms2d()

```
def pysar.multi_transect.dms2d (
              Coord )
```

#### 17.47.1.5 find_row_column()

```
def pysar.multi_transect.find_row_column (
              Lon,
              Lat,
              lon,
              lat,
              lon_step,
              lat_step )
```

#### 17.47.1.6 get_intersect()

```
def pysar.multi_transect.get_intersect (
              m,
              c,
              x,
              y )
```

#### 17.47.1.7 get_lat_lon()

```
def pysar.multi_transect.get_lat_lon (
              h5file )
```

#### 17.47.1.8 get_start_end_point()

```
def pysar.multi_transect.get_start_end_point (
              Xf0,
              Yf0,
              Xf1,
              Yf1,
              L,
              dx,
              dy )
```

#### 17.47.1.9 get_transect()

```
def pysar.multi_transect.get_transect (
              z,
              x0,
              y0,
              x1,
              y1 )
```

#### 17.47.1.10 gps_to_LOS()

```
def pysar.multi_transect.gps_to_LOS (
              Ve,
              Vn,
              theta,
              heading )
```

#### 17.47.1.11 line()

```
def pysar.multi_transect.line (
              x0,
              y0,
              x1,
              y1 )
```

#### 17.47.1.12 main()

```
def pysar.multi_transect.main (
              argv )
```

#### 17.47.1.13 nanmean()

```
def pysar.multi_transect.nanmean (
              data,
              args )
```

#### 17.47.1.14 nanstd()

```
def pysar.multi_transect.nanstd (
              data,
              args )
```

#### 17.47.1.15 nearest()

```
def pysar.multi_transect.nearest (
              x,
              tbase,
              xstep )
```

#### 17.47.1.16 onclick()

```
def pysar.multi_transect.onclick (
              event )
```

#### 17.47.1.17 point_on_line_with_distance_from_beginning()

```
def pysar.multi_transect.point_on_line_with_distance_from_beginning (
              Xf0,
              Yf0,
              Xf1,
              Yf1,
              L )
```

#### 17.47.1.18 point_with_distance_from_line()

```
def pysar.multi_transect.point_with_distance_from_line (
              Xf0,
              Yf0,
              Xf1,
              Yf1,
              L )
```

#### 17.47.1.19 read_fault_coords()

```
def pysar.multi_transect.read_fault_coords (
              Fault_coord_file,
              Dp )
```

#### 17.47.1.20 readGPSfile()

```
def pysar.multi_transect.readGPSfile (
              gpsFile,
              gps_source )
```

#### 17.47.1.21 redGPSfile()

```
def pysar.multi_transect.redGPSfile (
              gpsFile )
```

#### 17.47.1.22 redGPSfile_cmm4()

```
def pysar.multi_transect.redGPSfile_cmm4 (
              gpsFile )
```

#### 17.47.1.23 usage()

```
def pysar.multi_transect.usage ( )
```

### 17.47.2 Variable Documentation

#### 17.47.2.1 alpha

```
alpha
```

#### 17.47.2.2 avgInSAR

```
avgInSAR = np.array(nanmean(transect,axis=1))
```

#### 17.47.2.3 ax

```
ax = fig.add_subplot(111)
```

#### 17.47.2.4 axes

```
axes
```

#### 17.47.2.5 axes2

```
axes2
```

#### 17.47.2.6 c

```
c = float(y0-m*x0)
```

#### 17.47.2.7 c_prof_edge

```
c_prof_edge
```

#### 17.47.2.8 cf

```
cf
```

#### 17.47.2.9 check_result

```
def check_result = 'True'
```

#### 17.47.2.10 check_result2

```
def check_result2 = check_st_in_box2(gx,gy,x0,y0,x1,y1,X0,Y0,X1,Y1)
```

#### 17.47.2.11 cid

```
cid = fig.canvas.mpl_connect('button_press_event', onclick)
```

#### 17.47.2.12 color

```
color
```

ax.plot(D/1000.0, avgInSAR*1000, 'r-')

To plot the Fault location on the profile try:

#### 17.47.2.13 D

```
D = np.hypot(DX, DY)
```

#### 17.47.2.14 dataset

```
dictionary dataset = {}
```

#### 17.47.2.15 df0_km

```
def df0_km = dist_point_from_line(mf,cf,x0,y0,dx,dy)
```

#### 17.47.2.16 dg

```
def dg = dist_point_from_line(m,c,gx,gy,1,1)
```

#### 17.47.2.17 DistGPS

```
DistGPS = [ ]
```

#### 17.47.2.18 dp

```
float dp = 1.0
```

#### 17.47.2.19 dset

```
dset = h5file_theta['mask'].get('mask')
```

#### 17.47.2.20 dx

```
dx = float(h5file[k[0]].attrs['X_STEP'])*6375000.0*np.pi/180.0
```

#### 17.47.2.21 DX

```
tuple DX = (x-x0)*dx
```

#### 17.47.2.22 dy

```
dy = float(h5file[k[0]].attrs['Y_STEP'])*6375000.0*np.pi/180.0
```

#### 17.47.2.23 DY

```
tuple DY = (y-y0)*dy
```

#### 17.47.2.24 Fault_lat

```
Fault_lat
```

#### 17.47.2.25 Fault_lon

```
Fault_lon
```

#### 17.47.2.26 FaultCoords

```
list FaultCoords = [Fault_lat[Np],Fault_lon[Np],Fault_lat[Np+1],Fault_lon[Np+1]]
```

#### 17.47.2.27 FaultLine

```
string FaultLine = 'None'
```

#### 17.47.2.28 fig

```
fig = plt.figure()
```

#### 17.47.2.29 fig2

```
fig2
```

#### 17.47.2.30 figName

```
string figName = 'transect_area_'+str(Np)+'.png'
```

Temporary To plot DEM try: majorLocator = MultipleLocator(5) ax.yaxis.set_major_locator(majorLocator) minorLocator = MultipleLocator(1) ax.yaxis.set_minor_locator(minorLocator)

#### 17.47.2.31 fileExtension

```
fileExtension
```

#### 17.47.2.32 fileName

```
fileName
```

#### 17.47.2.33 fontsize

```
fontsize
```

#### 17.47.2.34 GPS

```
list GPS = []
```

#### 17.47.2.35 GPS_in_bound

```
int GPS_in_bound = []
```

#### 17.47.2.36 GPS_in_bound_st

```
list GPS_in_bound_st = []
```

#### 17.47.2.37 GPS_lat

```
list GPS_lat = []
```

#### 17.47.2.38 GPS_lon

```
list GPS_lon = []
```

#### 17.47.2.39 GPS_station

```
list GPS_station = []
```

#### 17.47.2.40 gpsFile

```
string gpsFile = 'Nogps'
```

#### 17.47.2.41 gpsLOS

```
def gpsLOS = gps_to_LOS(Ve[idx],Vn[idx],theta[IDY,IDX],heading)
```

#### 17.47.2.42 gpsLOS_ref

```
def gpsLOS_ref = gps_to_LOS(Ve[idxRef],Vn[idxRef],theta[IDYref,IDXref],heading)
```

#### 17.47.2.43 GPSx

```
list GPSx = []
```

#### 17.47.2.44 GPSxx

```
list GPSxx = []
```

#### 17.47.2.45 GPSy

```
list GPSy = []
```

#### 17.47.2.46 GPSyy

```
list GPSyy = []
```

#### 17.47.2.47 gx

```
list gx = GPSx[i]
```

#### 17.47.2.48 gy

```
list gy = GPSy[i]
```

#### 17.47.2.49 h5file_theta

```
h5file_theta = h5py.File(incidence_file,'r')
```

#### 17.47.2.50 hbound

```
int hbound = np.nanmax(transect)*1000
```

#### 17.47.2.51 heading

```
float heading = 193.0*np.pi/180.0
```

#### 17.47.2.52 idx

```
idx = Stations.index(st)
```

#### 17.47.2.53 IDX

```
IDX
```

#### 17.47.2.54 idxRef

```
idxRef = Stations.index(refStation)
```

#### 17.47.2.55 IDXref

```
IDXref
```

#### 17.47.2.56 IDY

```
IDY
```

#### 17.47.2.57 IDYref

```
IDYref
```

#### 17.47.2.58 insarData

```
insarData = z
```

#### 17.47.2.59 lat

```
lat
```

#### 17.47.2.60 Lat

```
Lat
```

#### 17.47.2.61 Lat0

```
list Lat0 = FaultCoords[1]
```

#### 17.47.2.62 Lat1

```
list Lat1 = FaultCoords[3]
```

#### 17.47.2.63 lat_all

```
lat_all
```

#### 17.47.2.64 lat_step

```
lat_step
```

#### 17.47.2.65 lat_transect

```
def lat_transect = lat_all[y.astype(np.int), x.astype(np.int)]
```

#### 17.47.2.66 lbound

```
int lbound = np.nanmin(transect)*1000
```

lower and higher bounds for diplaying the profile

#### 17.47.2.67 Length

```
Length
```

#### 17.47.2.68 length

```
length = int(np.hypot(x1-x0, y1-y0))
```

try: mf=float(Yf1-Yf0)/float((Xf1-Xf0)) # slope of the fault line cf=float(Yf0-mf*Xf0) # intercept of the fault line df0=dist_point_from_line(mf,cf,x0,y0,1,1) #distance of the profile start point from the Fault line df1=dist_point_↩ from_line(mf,cf,x1,y1,1,1) #distance of the profile end point from the Fault line

#### 17.47.2.69 linewidth

```
linewidth
```

#### 17.47.2.70 lon

```
lon
```

#### 17.47.2.71 Lon

```
Lon
```

#### 17.47.2.72 lon_all

```
lon_all
```

#### 17.47.2.73 lon_step

```
lon_step
```

#### 17.47.2.74 lon_transect

```
def lon_transect = lon_all[y.astype(np.int), x.astype(np.int)]
```

#### 17.47.2.75 m

```
m = float(y1-y0)/float((x1-x0))
```

#### 17.47.2.76 m1

```
float m1 = -1.0/m
```

#### 17.47.2.77 m_prof_edge

```
m_prof_edge
```

#### 17.47.2.78 matFile

```
string matFile = 'transect'+str(Np)+'.mat'
```

#### 17.47.2.79 mf

```
mf
```

#### 17.47.2.80 mfc

```
mfc
```

#### 17.47.2.81 ms

```
ms
```

ax.fill_between(D/1000.0, (avgInSAR-stdInSAR)∗1000, (avgInSAR+stdInSAR)∗1000,where=(avgInSAR+stdInS↩
AR)∗1000>=(avgInSAR-stdInSAR)∗1000,alpha=1, facecolor='Red')

#### 17.47.2.82 NoInSAR

```
string NoInSAR = 'yes'
```

#### 17.47.2.83 nrows

```
nrows
```

#### 17.47.2.84 Num_profiles

```
int Num_profiles = len(Fault_lon)-1
```

#### 17.47.2.85 Se

```
Se
```

#### 17.47.2.86 Sn

```
Sn
```

#### 17.47.2.87 Stations

```
Stations
```

#### 17.47.2.88 stationsList

```
stationsList = Stations
```

#### 17.47.2.89 stdInSAR

```
stdInSAR = np.array(nanstd(transect,axis=1))
```

#### 17.47.2.90 theta

```
float theta = dset[0:dset.shape[0],0:dset.shape[1]]
```

#### 17.47.2.91 transect

```
int transect = np.zeros([len(D),ntrans])
```

#### 17.47.2.92 transect_lat

```
transect_lat = np.zeros([len(D),ntrans])
```

#### 17.47.2.93 transect_lon

```
transect_lon = np.zeros([len(D),ntrans])
```

#### 17.47.2.94 unitVec

```
list unitVec = [np.cos(heading)*np.sin(theta),-np.sin(theta)*np.sin(heading),0]
```

#### 17.47.2.95 Ve

```
Ve
```

#### 17.47.2.96 Vn

```
Vn
```

#### 17.47.2.97 Width

```
Width
```

#### 17.47.2.98 x

```
x
```

#### 17.47.2.99 x0

```
list x0 = xc[1]
```

#### 17.47.2.100 X0

```
float X0 = i*dp/np.sqrt(1+m1**2)+x0
```

#### 17.47.2.101 x1

```
x1
```

#### 17.47.2.102 X1

```
float X1 = i*dp/np.sqrt(1+m1**2)+x1
```

#### 17.47.2.103 xc

```
list xc = [ ]
```

#### 17.47.2.104 Xf0

```
Xf0
```

#### 17.47.2.105 Xf1

```
Xf1
```

#### 17.47.2.106 xlim

```
string xlim = 'no'
```

#### 17.47.2.107 XX0

```
list XX0 = [ ]
```

#### 17.47.2.108 y

```
y
```

#### 17.47.2.109 y0

```
list y0 = yc[1]
```

#### 17.47.2.110 Y0

```
float Y0 = m1*(X0-x0)+y0
```

#### 17.47.2.111 y1

```
y1
```

#### 17.47.2.112 Y1

```
float Y1 = m1*(X1-x1)+y1
```

#### 17.47.2.113 yc

```
list yc = [ ]
```

#### 17.47.2.114 Yf0

```
Yf0
```

#### 17.47.2.115 Yf1

```
Yf1
```

#### 17.47.2.116 ylim

```
string ylim = 'no'
```

#### 17.47.2.117 YY0

```
list YY0 = [ ]
```

#### 17.47.2.118 zi

```
def zi = z[y.astype(np.int), x.astype(np.int)]
```

## 17.48 pysar.multilook Namespace Reference

### Functions

- def multilook_matrix (matrix, lks_y, lks_x)
  *Sub Functions ##############################################.*
- def multilook_attribute (atr_dict, lks_y, lks_x)
- def multilook_file (infile, lks_y, lks_x, outfile=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

### 17.48.1 Function Documentation

#### 17.48.1.1 cmdLineParse()

```
def pysar.multilook.cmdLineParse ( )
```

#### 17.48.1.2 main()

```
def pysar.multilook.main (
              argv )
```

#### 17.48.1.3 multilook_attribute()

```
def pysar.multilook.multilook_attribute (
              atr_dict,
              lks_y,
              lks_x )
```

#### 17.48.1.4 multilook_file()

```
def pysar.multilook.multilook_file (
              infile,
              lks_y,
              lks_x,
              outfile = None )
```

#### 17.48.1.5 multilook_matrix()

```
def pysar.multilook.multilook_matrix (
              matrix,
              lks_y,
              lks_x )
```

Sub Functions ################################################.

### 17.48.2 Variable Documentation

#### 17.48.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''
example:
  multilook.py  velocity.h5  15 15
  multilook.py  srtm30m.dem  10 10  -o srtm30m_300m.dem
'''
```

## 17.49 pysar.plot_atmDrop Namespace Reference

### Variables

- list projectList = ['AlosAT422','AlosAT423','AlosDT72','AlosDT73']
- string projectDir = '/Users/jeromezhang/Documents/insarlab/Kyushu/Volcanoes/Kuju'
- numProject = len(projectList)
- fig = plt.figure(figsize=(12,12))
- ax1 = fig.add_subplot(211)
- ax2 = fig.add_subplot(212)
- offset = range(1,numProject+1)
- fl = open(projectDir+'/'+projectList[i]+'/spatialMean_sum_Seeded_ts.txt','r')

  *Read txt file.*
- list lines = [ ]
- int lineNum = 0

- list dateList6 = [ ]
- list meanList = [ ]
- list pixList = [ ]
- line_s = line.split()
- dateList = ptime.yyyymmdd(dateList6)
- dates = np.array(dates)
- datevector
- idxMean = max(enumerate(meanList),key=lambda x: x[1])[0]
- list idxPix = pixList $<$ 0.7
- sc1 = ax1.scatter(dates, np.tile(offset[i],lineNum), c=meanList, s=22$**$2, alpha=0.3, vmin=0.0, vmax=1.0)

  *Plot.*
- c
- s
- alpha
- vmin
- vmax
- sc2 = ax2.scatter(dates, np.tile(offset[i],lineNum), c=pixList, s=22$**$2, alpha=0.3, vmin=0.0, vmax=1.0)
- fontsize
- cbar = fig.colorbar(sc2)
- bbox_inches
- transparent

### 17.49.1 Variable Documentation

#### 17.49.1.1 alpha

```
alpha
```

#### 17.49.1.2 ax1

```
ax1 = fig.add_subplot(211)
```

#### 17.49.1.3 ax2

```
ax2 = fig.add_subplot(212)
```

#### 17.49.1.4 bbox_inches

```
bbox_inches
```

#### 17.49.1.5 c

```
c
```

#### 17.49.1.6 cbar

```
cbar = fig.colorbar(sc2)
```

#### 17.49.1.7 dateList

```
dateList = ptime.yyyymmdd(dateList6)
```

#### 17.49.1.8 dateList6

```
list dateList6 = []
```

#### 17.49.1.9 dates

```
dates = np.array(dates)
```

#### 17.49.1.10 datevector

```
datevector
```

#### 17.49.1.11 fig

```
fig = plt.figure(figsize=(12,12))
```

#### 17.49.1.12 fl

```
fl = open(projectDir+'/'+projectList[i]+'/spatialMean_sum_Seeded_ts.txt','r')
```

Read txt file.

#### 17.49.1.13 fontsize

```
fontsize
```

#### 17.49.1.14 idxMean

```
idxMean = max(enumerate(meanList),key=lambda x:  x[1])[0]
```

#### 17.49.1.15 idxPix

```
list idxPix = pixList < 0.7
```

#### 17.49.1.16 line_s

```
line_s = line.split()
```

#### 17.49.1.17 lineNum

```
int lineNum = 0
```

#### 17.49.1.18 lines

```
lines = []
```

#### 17.49.1.19 meanList

```
meanList = []
```

#### 17.49.1.20 numProject

```
numProject = len(projectList)
```

#### 17.49.1.21 offset

```
offset = range(1,numProject+1)
```

#### 17.49.1.22 pixList

```
pixList = []
```

#### 17.49.1.23 projectDir

```
string projectDir = '/Users/jeromezhang/Documents/insarlab/Kyushu/Volcanoes/Kuju'
```

#### 17.49.1.24 projectList

```
list projectList = ['AlosAT422','AlosAT423','AlosDT72','AlosDT73']
```

#### 17.49.1.25 s

```
s
```

#### 17.49.1.26 sc1

```
sc1 = ax1.scatter(dates, np.tile(offset[i],lineNum), c=meanList, s=22**2, alpha=0.3, vmin=0.0,
vmax=1.0)
```

Plot.

#### 17.49.1.27 sc2

```
sc2 = ax2.scatter(dates, np.tile(offset[i],lineNum), c=pixList, s=22**2, alpha=0.3, vmin=0.0,
vmax=1.0)
```

#### 17.49.1.28 transparent

```
transparent
```

#### 17.49.1.29 vmax

```
vmax
```

#### 17.49.1.30 vmin

```
vmin
```

## 17.50 pysar.plot_network Namespace Reference

### Functions

- def cmdLineParse ()
- def main (argv)

    *Main Function ##############################.*

### Variables

- string BL_LIST
- string DATE12_LIST
- string EXAMPLE

## 17.50.1 Function Documentation

### 17.50.1.1 cmdLineParse()

```
def pysar.plot_network.cmdLineParse ( )
```

### 17.50.1.2 main()

```
def pysar.plot_network.main (
              argv )
```

Main Function ################################.

## 17.50.2 Variable Documentation

### 17.50.2.1 BL_LIST

```
string BL_LIST
```

**Initial value:**

```
= '''
070106     0.0   0.03  0.0000000  0.00000000000 2155.2 /scratch/SLC/070106/
070709  2631.9   0.07  0.0000000  0.00000000000 2155.2 /scratch/SLC/070709/
070824  2787.3   0.07  0.0000000  0.00000000000 2155.2 /scratch/SLC/070824/
'''
```

### 17.50.2.2 DATE12_LIST

```
string DATE12_LIST
```

**Initial value:**

```
= '''
070709-100901
070709-101017
070824-071009
'''
```

#### 17.50.2.3 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  plot_network.py unwrapIfgram.h5
  plot_network.py unwrapIfgram.h5 --coherence coherence_spatialAverage.list
  plot_network.py unwrapIfgram.h5 --coherence coherence.h5
  plot_network.py Modified_coherence.h5 --save
  plot_network.py Modified_coherence.h5 --nodisplay
  plot_network.py Pairs.list               -b bl_list.txt
  plot_network.py unwrapIfgram_date12.list -b bl_list.txt
'''
```

## 17.51 pysar.pysar2insarmaps Namespace Reference

### Functions

- def project_name_from_path (path)
- def sorted_ls (path)
- def rev_sorted_ls (path)
- def get_H5_filename (path)
- def build_parser ()
- def main ()

### 17.51.1 Function Documentation

#### 17.51.1.1 build_parser()

```
def pysar.pysar2insarmaps.build_parser ( )
```

#### 17.51.1.2 get_H5_filename()

```
def pysar.pysar2insarmaps.get_H5_filename (
              path )
```

#### 17.51.1.3 main()

```
def pysar.pysar2insarmaps.main ( )
```

#### 17.51.1.4 project_name_from_path()

```
def pysar.pysar2insarmaps.project_name_from_path (
              path )
```

#### 17.51.1.5 rev_sorted_ls()

```
def pysar.pysar2insarmaps.rev_sorted_ls (
              path )
```

#### 17.51.1.6 sorted_ls()

```
def pysar.pysar2insarmaps.sorted_ls (
              path )
```

## 17.52 pysar.pysarApp Namespace Reference

### Functions

- def check_isfile (File)
- def check_subset_file (File, inps_dict, outFile=None, overwrite=False)
- def check_geocode_file (geomapFile, File, outFile=None)
- def subset_dataset (inps, geo_box4geo, pix_box4rdr)
- def create_subset_dataset (inps, pix_box=None, geo_box=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string LOGO
- string TEMPLATE
- string EXAMPLE
- string UM_FILE_STRUCT

### 17.52.1 Function Documentation

#### 17.52.1.1 check_geocode_file()

```
def pysar.pysarApp.check_geocode_file (
              geomapFile,
              File,
              outFile = None )
```

Geocode input file or use existed geocoded file.

#### 17.52.1.2 check_isfile()

```
def pysar.pysarApp.check_isfile (
              File )
```

Check if input file exists and readable.

#### 17.52.1.3 check_subset_file()

```
def pysar.pysarApp.check_subset_file (
              File,
              inps_dict,
              outFile = None,
              overwrite = False )
```

Subset input file or use existed subseted file.

#### 17.52.1.4 cmdLineParse()

```
def pysar.pysarApp.cmdLineParse ( )
```

#### 17.52.1.5 create_subset_dataset()

```
def pysar.pysarApp.create_subset_dataset (
              inps,
              pix_box = None,
              geo_box = None )
```

```
Create/prepare subset of datasets in different folder for time series analysis.
For dataset (unwrapped interferograms) in radar coord, only support subset in row/col or y/x
For dataset (unwrapped interferograms) in geo coord, lalo has higher priority than yx, if both are specified.
```

#### 17.52.1.6 main()

```
def pysar.pysarApp.main (
            argv )
```

#### 17.52.1.7 subset_dataset()

```
def pysar.pysarApp.subset_dataset (
            inps,
            geo_box4geo,
            pix_box4rdr )
```

```
Subset all file within dataset
with geo_box4geo for all geocoded file and pix_box4rdr for all files in radar coord.
Inputs:
    inps - Namespace with all files that needs to be subseted and work_dir
    geo_box4geo - tuple of 4 float, subset range in lat/lon for files in geo coord
    pix_box4rdr - tuple of 4 int, subset range in y/x for files in radar coord
Output:
    inps - Namespace, update file name/path info
```

### 17.52.2 Variable Documentation

#### 17.52.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  pysarApp.py  SanAndreasT356EnvD.template
  pysarApp.py  SanAndreasT356EnvD.template  --dir ~/insarlab/SanAndreasT356EnvD/TIMESERIES
'''
```

#### 17.52.2.2 LOGO

```
string LOGO
```

**Initial value:**

```
= '''
__________________________________________________
       ____              __     __     ____
       /    )          /    )   / |    /    )
------/____/-----------\-------/__|---/___ /------
     /         /   /    \     /   |  /    |
____/_________(___/_(____/___/____|_/_____|_______
                 /
             (_ /

 A Python package for InSAR time series analysis.
               PySAR v1.2, Jan 2017
 Geodesy Lab, University of Miami, Maimi FL, USA
__________________________________________________
'''
```

#### 17.52.2.3 TEMPLATE

```
string TEMPLATE
```

**Initial value:**

```
= '''template:
# Input Data (not needed for Miami user)
pysar.unwrapFiles    = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*.unw
pysar.corFiles       = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.cor
pysar.wrapFiles      = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.int     #optional
pysar.geomap         = /SanAndreasT356EnvD/PROCESS/GEO/*050102-070809*/geomap*.trans
pysar.dem.radarCoord = /SanAndreasT356EnvD/PROCESS/DONE/*050102-070809*/radar*.hgt
pysar.dem.geoCoord   = /SanAndreasT356EnvD/DEM/srtm1_30m.dem                   #optional

pysar.network.reference       = date12.list         #optional
pysar.network.coherenceBase   = yes                 #optional, auto for yes

pysar.subset.yx          = 1800:2000,700:800        #optional, auto/no/off for whole area
pysar.subset.lalo        = 31.5:32.5,130.5:131.0    #optional, auto/no/off for whole area

pysar.reference.yx       = 257 , 151                #optional, auto for max coherence selection
pysar.reference.lalo     = 31.8, 130.8              #optional, auto for max coherence selection
pysar.reference.date     = 20090120                 #optional, auto for the first date

pysar.troposphericDelay.method        = pyaps   #[height_correlation], auto for no tropospheric correction
pysar.troposphericDelay.polyOrder     = 1       #for height_correlation method
pysar.troposphericDelay.weatherModel  = ECMWF   #[ERA, MERRA, NARR], for pyaps method

pysar.topoError = yes               #[no], auto for yes
pysar.deramp    = plane             #[plane, quadratic, baseline_cor, base_trop_cor], auto for no
pysar.geocode   = yes               #[no], auto for yes
'''
```

#### 17.52.2.4 UM_FILE_STRUCT

```
string UM_FILE_STRUCT
```

**Initial value:**

```
= '''
    scratch/                 # $SCRATCHDIR defined in environmental variable
        SanAndreasT356EnvD/  # my_projectName, same as the basename of template file
            DEM/             # DEM file(s) (for topographic phase and geocode)
            DOWNLOAD/        # (optional) Data downloaded from agencies
            PROCESS/         # Interferograms processed by ROI_PAC, Gamma, ISCE, ...
            RAW/             # (optional) Raw SAR data untared from DOWNLOAD directory
            SLC/             # (optional) SLC SAR data after focusing from RAW directory
            TIMESERIES/           # PySAR work directory for time series analysis
'''
```

## 17.53 pysar.pysarApp_cmd Namespace Reference

### Functions

- def check_isfile (File)
- def check_subset_file (File, inps_dict, outFile=None, overwrite=False)
- def check_geocode_file (geomapFile, File, outFile=None)
- def subset_dataset (inps, geo_box4geo, pix_box4rdr)
- def create_subset_dataset (inps, pix_box=None, geo_box=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string LOGO
- string TEMPLATE
- string EXAMPLE
- string UM_FILE_STRUCT

## 17.53.1 Function Documentation

### 17.53.1.1 check_geocode_file()

```
def pysar.pysarApp_cmd.check_geocode_file (
              geomapFile,
              File,
              outFile = None )
```

```
Geocode input file or use existed geocoded file.
```

### 17.53.1.2 check_isfile()

```
def pysar.pysarApp_cmd.check_isfile (
              File )
```

```
Check if input file exists and readable.
```

### 17.53.1.3 check_subset_file()

```
def pysar.pysarApp_cmd.check_subset_file (
              File,
              inps_dict,
              outFile = None,
              overwrite = False )
```

```
Subset input file or use existed subseted file.
```

#### 17.53.1.4 cmdLineParse()

```
def pysar.pysarApp_cmd.cmdLineParse ( )
```

#### 17.53.1.5 create_subset_dataset()

```
def pysar.pysarApp_cmd.create_subset_dataset (
              inps,
              pix_box = None,
              geo_box = None )
```

```
Create/prepare subset of datasets in different folder for time series analysis.
For dataset (unwrapped interferograms) in radar coord, only support subset in row/col or y/x
For dataset (unwrapped interferograms) in geo coord, lalo has higher priority than yx, if both are specified.
```

#### 17.53.1.6 main()

```
def pysar.pysarApp_cmd.main (
              argv )
```

#### 17.53.1.7 subset_dataset()

```
def pysar.pysarApp_cmd.subset_dataset (
              inps,
              geo_box4geo,
              pix_box4rdr )
```

```
Subset all file within dataset
with geo_box4geo for all geocoded file and pix_box4rdr for all files in radar coord.
Inputs:
    inps - Namespace with all files that needs to be subseted and work_dir
    geo_box4geo - tuple of 4 float, subset range in lat/lon for files in geo coord
    pix_box4rdr - tuple of 4 int, subset range in y/x for files in radar coord
Output:
    inps - Namespace, update file name/path info
```

### 17.53.2 Variable Documentation

#### 17.53.2.1 EXAMPLE

string EXAMPLE

**Initial value:**

```
= '''example:
  pysarApp.py  SanAndreasT356EnvD.template
  pysarApp.py  SanAndreasT356EnvD.template  --dir ~/insarlab/SanAndreasT356EnvD/TIMESERIES
'''
```

#### 17.53.2.2 LOGO

string LOGO

**Initial value:**

```
= '''
_________________________________________________
       ____             __     __     ____
       /    )         /    )   / |    /    )
------/____/----------\-------/__|---/___ /------
     /        /   /    \     /   |  /    |
____/________(___/_(____/___/____|_/_____|_______
                /
            (_ /

 A Python package for InSAR time series analysis.
               PySAR v1.2, Jan 2017
 Geodesy Lab, University of Miami, Maimi FL, USA
_________________________________________________
'''
```

#### 17.53.2.3 TEMPLATE

string TEMPLATE

**Initial value:**

```
= '''template:
# Input Data (not needed for Miami user)
pysar.unwrapFiles    = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*.unw
pysar.corFiles       = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.cor
pysar.wrapFiles      = /SanAndreasT356EnvD/PROCESS/DONE/IFG*/filt*rlks.int     #optional
pysar.geomap         = /SanAndreasT356EnvD/PROCESS/GEO/*050102-070809*/geomap*.trans
pysar.dem.radarCoord = /SanAndreasT356EnvD/PROCESS/DONE/*050102-070809*/radar*.hgt
pysar.dem.geoCoord   = /SanAndreasT356EnvD/DEM/srtm1_30m.dem                    #optional

pysar.subset.yx          = 1800:2000,700:800        #optional, auto/no for whole area
pysar.subset.lalo        = 31.5:32.5,130.5:131.0    #optional, auto/no for whole area

pysar.network.reference       = date12.list         #optional
pysar.network.coherenceBase   = yes                 #optional, auto for yes

pysar.reference.yx       = 257 , 151                #optional, auto for max coherence selection
pysar.reference.lalo     = 31.8, 130.8              #optional, auto for max coherence selection
pysar.reference.date     = 20090120                 #optional, auto for the first date

pysar.troposphericDelay.method        = pyaps   #[height_correlation]
pysar.troposphericDelay.polyOrder     = 1       #for height_correlation method
pysar.troposphericDelay.weatherModel  = ECMWF   #[ERA, MERRA, NARR], for pyaps method

pysar.topoError = yes               #[no], auto for yes
pysar.deramp    = plane             #[plane, quadratic, baseline_cor, base_trop_cor], auto for no
pysar.geocode   = yes               #[no], auto for yes
'''
```

#### 17.53.2.4 UM_FILE_STRUCT

```
string UM_FILE_STRUCT
```

**Initial value:**

```
= '''
    scratch/                 # $SCRATCHDIR defined in environmental variable
        SanAndreasT356EnvD/  # my_projectName, same as the basename of template file
            DEM/             # DEM file(s) (for topographic phase and geocode)
            DOWNLOAD/        # (optional) Data downloaded from agencies
            PROCESS/         # Interferograms processed by ROI_PAC, Gamma, ISCE, ...
            RAW/             # (optional) Raw SAR data untared from DOWNLOAD directory
            SLC/             # (optional) SLC SAR data after focusing from RAW directory
            TIMESERIES/          # PySAR work directory for time series analysis
'''
```

## 17.54 pysar.pysarApp_orig Namespace Reference

### Functions

- def find_filename (template, option, workDir='.')

  *Sub Functions ########################################## find accurate file name for template input ##############.*
- def check_subset (inName, subset, option='yx', workDir='.')

  *update the subset of input file ########################*
- def check_geocode (inName, geomapFile, workDir='.')

  *update geocoding of input file #########################*
- def check_mask (inName, maskFile, workDir='.')

  *update masking of input file ############################*
- def usage ()

  *Usage Function ###########################.*
- def cmdLineParse ()
- def main (argv)

  *Main Function ##########################################.*

### 17.54.1 Function Documentation

#### 17.54.1.1 check_geocode()

```
def pysar.pysarApp_orig.check_geocode (
            inName,
            geomapFile,
            workDir = '.' )
```

update geocoding of input file #########################

#### 17.54.1.2 check_mask()

```
def pysar.pysarApp_orig.check_mask (
              inName,
              maskFile,
              workDir = '.' )
```

update masking of input file #############################

#### 17.54.1.3 check_subset()

```
def pysar.pysarApp_orig.check_subset (
              inName,
              subset,
              option = 'yx',
              workDir = '.' )
```

update the subset of input file ##########################

#### 17.54.1.4 cmdLineParse()

```
def pysar.pysarApp_orig.cmdLineParse ( )
```

#### 17.54.1.5 find_filename()

```
def pysar.pysarApp_orig.find_filename (
              template,
              option,
              workDir = '.' )
```

Sub Functions ########################################## find accurate file name for template input ##############.

#### 17.54.1.6 main()

```
def pysar.pysarApp_orig.main (
              argv )
```

Main Function #########################################.

#### 17.54.1.7 usage()

```
def pysar.pysarApp_orig.usage ( )
```

Usage Function ############################.

## 17.55 pysar.quality_map Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.55.1 Function Documentation

#### 17.55.1.1 main()

```
def pysar.quality_map.main (
              argv )
```

#### 17.55.1.2 usage()

```
def pysar.quality_map.usage ( )
```

## 17.56 pysar.reconstruct_igrams Namespace Reference

### Functions

- def reconstruct_igrams_from_timeseries (h5timeseries, h5igrams)
- def usage ()
- def main (argv)

### 17.56.1 Function Documentation

#### 17.56.1.1 main()

```
def pysar.reconstruct_igrams.main (
              argv )
```

#### 17.56.1.2 reconstruct_igrams_from_timeseries()

```
def pysar.reconstruct_igrams.reconstruct_igrams_from_timeseries (
              h5timeseries,
              h5igrams )
```

#### 17.56.1.3 usage()

```
def pysar.reconstruct_igrams.usage ( )
```

## 17.57 pysar.reference_epoch Namespace Reference

### Functions

- def yymmdd2yyyymmdd (date)
- def usage ()
- def main (argv)

### 17.57.1 Function Documentation

#### 17.57.1.1 main()

```
def pysar.reference_epoch.main (
              argv )
```

#### 17.57.1.2 usage()

```
def pysar.reference_epoch.usage ( )
```

#### 17.57.1.3 yymmdd2yyyymmdd()

```
def pysar.reference_epoch.yymmdd2yyyymmdd (
              date )
```

## 17.58 pysar.remove_dates Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.58.1 Function Documentation

#### 17.58.1.1 main()

```
def pysar.remove_dates.main (
              argv )
```

#### 17.58.1.2 usage()

```
def pysar.remove_dates.usage ( )
```

## 17.59 pysar.remove_plane Namespace Reference

### Functions

- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

### 17.59.1 Function Documentation

#### 17.59.1.1 cmdLineParse()

```
def pysar.remove_plane.cmdLineParse ( )
```

#### 17.59.1.2 main()

```
def pysar.remove_plane.main (
              argv )
```

### 17.59.2 Variable Documentation

#### 17.59.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  remove_plane.py  timeseries.h5      -m Mask.h5
  remove_plane.py  timeseries.h5      -m Mask.h5          -s quadratic
  remove_plane.py  090214_101120.unw  -m Mask_tempCoh.h5 -s quadratic  -y 0,2400,2000,6843
'''
```

## 17.60 pysar.rewrap Namespace Reference

### Functions

- def usage ()
- def rewrap (unw)
- def main (argv)

### 17.60.1 Function Documentation

#### 17.60.1.1 main()

```
def pysar.rewrap.main (
              argv )
```

#### 17.60.1.2 rewrap()

```
def pysar.rewrap.rewrap (
              unw )
```

#### 17.60.1.3 usage()

```
def pysar.rewrap.usage ( )
```

## 17.61 pysar.save_gmt Namespace Reference

### Functions

- def get_geo_lat_lon (atr)
- def usage ()
- def main (argv)

    *Main Function ####################################.*

### 17.61.1 Function Documentation

#### 17.61.1.1 get_geo_lat_lon()

```
def pysar.save_gmt.get_geo_lat_lon (
              atr )
```

#### 17.61.1.2 main()

```
def pysar.save_gmt.main (
              argv )
```

Main Function ####################################.

#### 17.61.1.3 usage()

```
def pysar.save_gmt.usage ( )
```

## 17.62 pysar.save_kml Namespace Reference

### Functions

- def rewrap (unw)
- def usage ()
- def main (argv)

### 17.62.1 Function Documentation

#### 17.62.1.1 main()

```
def pysar.save_kml.main (
              argv )
```

#### 17.62.1.2 rewrap()

```
def pysar.save_kml.rewrap (
              unw )
```

#### 17.62.1.3 usage()

```
def pysar.save_kml.usage ( )
```

## 17.63 pysar.save_unavco Namespace Reference

### Functions

- def metadata_pysar2unavco (pysar_meta_dict, dateList)
- def cmdLineParse ()
- def main (argv)

### Variables

- INT_ZERO = np.int16(0)
- FLOAT_ZERO = np.float32(0.0)
- CPX_ZERO = np.complex64(0.0)
- string EXAMPLE

### 17.63.1 Function Documentation

#### 17.63.1.1 cmdLineParse()

```
def pysar.save_unavco.cmdLineParse ( )
```

#### 17.63.1.2 main()

```
def pysar.save_unavco.main (
              argv )
```

#### 17.63.1.3 metadata_pysar2unavco()

```
def pysar.save_unavco.metadata_pysar2unavco (
              pysar_meta_dict,
              dateList )
```

### 17.63.2 Variable Documentation

#### 17.63.2.1 CPX_ZERO

```
CPX_ZERO = np.complex64(0.0)
```

#### 17.63.2.2 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  save_unavco.py timeseries.h5 -i incidence_angle -d dem.h5 -c temporal_coherence.h5 -m mask.h5
'''
```

#### 17.63.2.3 FLOAT_ZERO

```
FLOAT_ZERO = np.float32(0.0)
```

#### 17.63.2.4 INT_ZERO

```
INT_ZERO = np.int16(0)
```

## 17.64 pysar.save_unw Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.64.1 Function Documentation

#### 17.64.1.1 main()

```
def pysar.save_unw.main (
              argv )
```

#### 17.64.1.2 usage()

```
def pysar.save_unw.usage ( )
```

## 17.65 pysar.seed_data Namespace Reference

### Functions

- def nearest (x, tbase, xstep)

  *Sub Functions ################################################.*
- def seed_file_reference_value (File, outName, refList, ref_y='', ref_x='')
- def seed_file_inps (File, inps=None, outFile=None)
- def seed_attributes (atr_in, x, y)
- def random_select_reference_yx (data_mat)
- def manual_select_reference_yx (stack, inps)
- def select_max_coherence_yx (corFile, mask=None)
- def print_warning (next_method)
- def read_seed_template2inps (template_file, inps=None)
- def read_seed_reference2inps (reference_file, inps=None)
- def usage ()

  *Usage ################################################.*
- def cmdLineParse ()
- def main (argv)

  *Main Function ##########################################.*

### 17.65.1 Function Documentation

#### 17.65.1.1 cmdLineParse()

```
def pysar.seed_data.cmdLineParse ( )
```

#### 17.65.1.2 main()

```
def pysar.seed_data.main (
              argv )
```

Main Function #########################################.

#### 17.65.1.3 manual_select_reference_yx()

```
def pysar.seed_data.manual_select_reference_yx (
              stack,
              inps )
```

```
Input:
    data4display : 2D np.array, stack of input file
    inps    : namespace, with key 'ref_x' and 'ref_y', which will be updated
```

#### 17.65.1.4 nearest()

```
def pysar.seed_data.nearest (
              x,
              tbase,
              xstep )
```

Sub Functions ##############################################.

#### 17.65.1.5 print_warning()

```
def pysar.seed_data.print_warning (
              next_method )
```

#### 17.65.1.6 random_select_reference_yx()

def pysar.seed_data.random_select_reference_yx (
*data_mat* )

#### 17.65.1.7 read_seed_reference2inps()

def pysar.seed_data.read_seed_reference2inps (
*reference_file,*
*inps = None* )

```
Read seed/reference info from reference file and update input namespace
```

#### 17.65.1.8 read_seed_template2inps()

def pysar.seed_data.read_seed_template2inps (
*template_file,*
*inps = None* )

```
Read seed/reference info from template file and update input namespace
```

#### 17.65.1.9 seed_attributes()

def pysar.seed_data.seed_attributes (
*atr_in,*
*x,*
*y* )

#### 17.65.1.10 seed_file_inps()

def pysar.seed_data.seed_file_inps (
*File,*
*inps = None,*
*outFile = None* )

```
Seed input file with option from input namespace
Return output file name if succeed; otherwise, return None
```

#### 17.65.1.11 seed_file_reference_value()

```
def pysar.seed_data.seed_file_reference_value (
              File,
              outName,
              refList,
              ref_y = '',
              ref_x = '' )
```

#### 17.65.1.12 select_max_coherence_yx()

```
def pysar.seed_data.select_max_coherence_yx (
              corFile,
              mask = None )
```

#### 17.65.1.13 usage()

```
def pysar.seed_data.usage ( )
```

Usage ################################################.

## 17.66 pysar.simulation Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.66.1 Function Documentation

#### 17.66.1.1 main()

```
def pysar.simulation.main (
              argv )
```

#### 17.66.1.2 usage()

```
def pysar.simulation.usage ( )
```

## 17.67 pysar.spatial_average Namespace Reference

### Functions

- def cmdLineParse ()
- def main (argv)

    *Main Function ################################.*

### Variables

- string EXAMPLE

    *Usage ####################################.*

### 17.67.1 Function Documentation

#### 17.67.1.1 cmdLineParse()

```
def pysar.spatial_average.cmdLineParse ( )
```

#### 17.67.1.2 main()

```
def pysar.spatial_average.main (
              argv )
```

Main Function ################################.

### 17.67.2 Variable Documentation

#### 17.67.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  spatial_average.py coherence.h5
  spatial_average.py unwrapIfgram.h5 -m Mask.h5
  spatial_average.py sum_timeseries_ECMWF_demCor.h5 -m Mask_tempCoh.h5
'''
```

Usage ####################################.

## 17.68 pysar.subset Namespace Reference

### Functions

- def coord_geo2radar (geoCoord, atr, coordType)

  *Example: 300 = coord_geo2radar(32.104990, atr,'lat') [1000,1500] = coord_geo2radar([130.5,131.4],atr,'lon')*
- def coord_radar2geo (radarCoord, atr, coordType)

  *Inputs: radarCoord : coordinate (list) in row/col in int atr : dictionary of file attributes coordType : coordinate type: row, col, y, x.*
- def check_box_within_data_coverage (pixel_box, atr_dict)
- def subset_attribute (atr_dict, subset_box)
- def get_coverage_box (atr)
- def read_subset_template2box (templateFile)
- def subset_box2inps (inps, pix_box, geo_box)
- def get_box_overlap_index (box1, box2)
- def subset_input_dict2box (subset_dict, meta_dict)
- def box_pixel2geo (pixel_box, meta_dict)
- def box_geo2pixel (geo_box, meta_dict)
- def subset_file (File, subset_dict, outFile=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

### 17.68.1 Function Documentation

#### 17.68.1.1 box_geo2pixel()

```
def pysar.subset.box_geo2pixel (
              geo_box,
              meta_dict )
```

```
Convert geo_box to pixel_box
```

#### 17.68.1.2 box_pixel2geo()

```
def pysar.subset.box_pixel2geo (
              pixel_box,
              meta_dict )
```

```
Convert pixel_box to geo_box
```

#### 17.68.1.3 check_box_within_data_coverage()

```
def pysar.subset.check_box_within_data_coverage (
              pixel_box,
              atr_dict )
```

```
Check the subset box's conflict with data coverage
Inputs:
    pixel_box : 4-tuple of int, indicating y/x coordinates of subset
    atr       : dictionary of file attributes
```

#### 17.68.1.4 cmdLineParse()

```
def pysar.subset.cmdLineParse ( )
```

#### 17.68.1.5 coord_geo2radar()

```
def pysar.subset.coord_geo2radar (
              geoCoord,
              atr,
              coordType )
```

Example: 300 = coord_geo2radar(32.104990, atr,'lat') [1000,1500] = coord_geo2radar([130.5,131.4],atr,'lon')

#### 17.68.1.6 coord_radar2geo()

```
def pysar.subset.coord_radar2geo (
              radarCoord,
              atr,
              coordType )
```

Inputs: radarCoord : coordinate (list) in row/col in int atr : dictionary of file attributes coordType : coordinate type: row, col, y, x.

Example: 32.104990 = coord_radar2geo(300, atr,'y') [130.5,131.4] = coord_radar2geo([1000,1500],atr,'x')

#### 17.68.1.7 get_box_overlap_index()

```
def pysar.subset.get_box_overlap_index (
              box1,
              box2 )
```

```
Get index box overlap area of two input boxes

Inputs:
    box1/2 : 4-tuple of int, indicating coverage of box1/2
             defining in (x0, y0, x1, y1)
Outputs:
    overlap_idx_box1/2 : 4-tuple of int, indicating index of overlap area in box1/2
                         defining in (idx_x0, idx_y0, idx_x1, idx_y1)
```

#### 17.68.1.8 get_coverage_box()

```
def pysar.subset.get_coverage_box (
              atr )
```

```
Get Coverage Box of data in geo and pixel coordinates
Inputs: atr - dict, meta data dictionary
Outputs:
    pix_box : 4-tuple of int, defining in (UL_X, UL_Y, LR_X, LR_Y)
    geo_box : 4-tuple of float in lat/lon
```

#### 17.68.1.9 main()

```
def pysar.subset.main (
              argv )
```

#### 17.68.1.10 read_subset_template2box()

```
def pysar.subset.read_subset_template2box (
              templateFile )
```

```
Read pysar.subset.lalo/yx option from template file into box type
Return None if not specified.
```

#### 17.68.1.11 subset_attribute()

```
def pysar.subset.subset_attribute (
              atr_dict,
              subset_box )
```

```
Update attributes dictionary due to subset
Inputs:
    atr_dict   : dict, data attributes to update
    subset_box : 4-tuple of int, subset box defined in (x0, y0, x1, y1)
Outputs:
    atr      : dict, updated data attributes
```

#### 17.68.1.12 subset_box2inps()

```
def pysar.subset.subset_box2inps (
              inps,
              pix_box,
              geo_box )
```

```
Update inps.subset_y/x/lat/lon from pixel_box and geo_box
```

#### 17.68.1.13 subset_file()

```
def pysar.subset.subset_file (
              File,
              subset_dict,
              outFile = None )
```

```
Subset file with
Inputs:
    File        : str, path/name of file
    outFile     : str, path/name of output file
    subset_dict : dict, subsut parameter, including the following items:
                  subset_x   : list of 2 int,   subset in x direction,   default=None
                  subset_y   : list of 2 int,   subset in y direction,   default=None
                  subset_lat : list of 2 float, subset in lat direction, default=None
                  subset_lon : list of 2 float, subset in lon direction, default=None
                  fill_value : float, optional. filled value for area outside of data coverage. default=None
                               None/not-existed to subset within data coverage only.
Outputs:
    outFile :  str, path/name of output file
```

#### 17.68.1.14 subset_input_dict2box()

```
def pysar.subset.subset_input_dict2box (
              subset_dict,
              meta_dict )
```

```
Convert subset inputs dict into bbox in radar and/or geo bounding box
Inputs:
    subset_dict : dict, including the following 4 objects:
                  subset_x   : list of 2 int,   subset in x direction,   default=None
                  subset_y   : list of 2 int,   subset in y direction,   default=None
                  subset_lat : list of 2 float, subset in lat direction, default=None
                  subset_lon : list of 2 float, subset in lon direction, default=None
    meta_dict   : dict, including the following items:
                  'WIDTH'      : int
                  'FILE_LENGTH': int
                  'X_FIRST'    : float, optional
                  'Y_FIRST'    : float, optional
                  'X_STEP'     : float, optional
                  'Y_STEP'     : float, optional
Outputs:
    # box defined by 4-tuple of number, defining (left, upper, right, lower) coordinate,
    #                                            (UL_X, UL_Y,  LR_X,  LR_Y )
    pixel_box   : 4-tuple of int, in pixel unit - 1
    geo_box     : 4-tuple of float, in  lat/lon unit - degree
                  None if file is in radar coordinate.
example:
    subset_dict = {'subset_x': None, 'subset_y': None, 'subset_lat': [30.5, 31.0], 'subset_lon': [130.0, 131.0
    subset_dict = {'subset_x': [100, 1100], 'subset_y': [2050, 2550], 'subset_lat': None, 'subset_lon': None}
    pixel_box          = subset_input_dict2box(subset_dict, pysar_meta_dict)[0]
    pixel_box, geo_box = subset_input_dict2box(subset_dict, pysar_meta_dict)
```

### 17.68.2 Variable Documentation

#### 17.68.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  subset.py unwrapIfgram.h5    -y    400  1500   -x    200   600
  subset.py geo_velocity.h5    -l    30.5 30.8   -L    130.3 130.9
  subset.py geo_timeseries.h5  --lat 30.5 30.8   --lon 130.3 130.9
  subset.py 030405_090801.unw  -t SinabungT495F50AlosA.template
  subset.py geo_incidence.h5   -r subset_geo_velocity.h
  subset.py *velocity*.h5 timeseries*.h5  -y 400:1500  -x 200:600
  subset.py geo_velocity.h5    -l 32.2:33.5  --outfill-nan
  subset.py Mask.h5            -x 500:3500   --outfill 0
  subset.py geomap_4rlks.trans --footprint
'''
```

## 17.69 pysar.sum_epochs Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.69.1 Function Documentation

#### 17.69.1.1 main()

```
def pysar.sum_epochs.main (
              argv )
```

#### 17.69.1.2 usage()

```
def pysar.sum_epochs.usage ( )
```

## 17.70 pysar.temporal_average Namespace Reference

### Functions

- def usage ()

    *Usage ####################################.*
- def main (argv)

    *Main Function ################################.*

### 17.70.1 Function Documentation

#### 17.70.1.1 main()

```
def pysar.temporal_average.main (
              argv )
```

Main Function ################################.

#### 17.70.1.2 usage()

```
def pysar.temporal_average.usage ( )
```

Usage ####################################.

## 17.71 pysar.temporal_coherence Namespace Reference

### Functions

- def date_list (h5file)
- def design_matrix (h5file)
- def usage ()
- def main (argv)

### 17.71.1 Function Documentation

#### 17.71.1.1 date_list()

```
def pysar.temporal_coherence.date_list (
              h5file )
```

#### 17.71.1.2 design_matrix()

```
def pysar.temporal_coherence.design_matrix (
              h5file )
```

```
Make the design matrix for the inversion.
```

#### 17.71.1.3 main()

```
def pysar.temporal_coherence.main (
              argv )
```

#### 17.71.1.4 usage()

```
def pysar.temporal_coherence.usage ( )
```

## 17.72 pysar.temporal_derivative Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.72.1 Function Documentation

#### 17.72.1.1 main()

```
def pysar.temporal_derivative.main (
              argv )
```

#### 17.72.1.2 usage()

```
def pysar.temporal_derivative.usage ( )
```

## 17.73 pysar.timeseries2velocity Namespace Reference

### Functions

- def yyyymmdd2years (date)
- def update_inps_from_template (inps, template_file)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE
- string TEMPLATE
- string DROP_DATE_TXT

### 17.73.1 Function Documentation

#### 17.73.1.1 cmdLineParse()

```
def pysar.timeseries2velocity.cmdLineParse ( )
```

#### 17.73.1.2 main()

```
def pysar.timeseries2velocity.main (
              argv )
```

#### 17.73.1.3 update_inps_from_template()

```
def pysar.timeseries2velocity.update_inps_from_template (
              inps,
              template_file )
```

```
Update inps.ex_date with input template file
```

#### 17.73.1.4 yyyymmdd2years()

```
def pysar.timeseries2velocity.yyyymmdd2years (
              date )
```

### 17.73.2 Variable Documentation

#### 17.73.2.1 DROP_DATE_TXT

```
string DROP_DATE_TXT
```

**Initial value:**

```
= '''drop_date.txt:
20040502
20060708
20090103
'''
```

#### 17.73.2.2 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  timeseries2velocity.py  timeSeries_ECMWF_demCor.h5
  timeseries2velocity.py  timeseries_ECMWF_demCor_plane.h5 -t KyushuT73F2980_2990AlosD.template
  timeseries2velocity.py  timeseries.h5 -m 20080201
  timeseries2velocity.py  timeseries.h5 -m 20080201 -M 20100508
  timeseries2velocity.py  timeseries.h5 -E 20040502,20060708,20090103
  timeseries2velocity.py  timeseries.h5 -E drop_date.txt
'''
```

#### 17.73.2.3 TEMPLATE

```
string TEMPLATE
```

**Initial value:**

```
= '''
pysar.network.dropDate = 20040502 20060708 20090103
pysar.network.dropDate = drop_date.txt
'''
```

## 17.74 pysar.transect Namespace Reference

### Functions

- def get_scale_from_disp_unit (disp_unit, data_unit)
- def read_lonlat_file (lonlat_file)
- def manual_select_start_end_point (File)
- def transect_yx (z, atr, start_yx, end_yx, interpolation='nearest')
- def transect_lalo (z, atr, start_lalo, end_lalo, interpolation='nearest')
- def transect_list (fileList, inps)
- def cmdLineParse ()
- def main (argv)

    *Main ####################################.*

### Variables

- string EXAMPLE

### 17.74.1 Function Documentation

#### 17.74.1.1 cmdLineParse()

```
def pysar.transect.cmdLineParse ( )
```

#### 17.74.1.2 get_scale_from_disp_unit()

```
def pysar.transect.get_scale_from_disp_unit (
              disp_unit,
              data_unit )
```

#### 17.74.1.3 main()

```
def pysar.transect.main (
              argv )
```

Main ####################################.

#### 17.74.1.4 manual_select_start_end_point()

```
def pysar.transect.manual_select_start_end_point (
              File )
```

```
Manual Select Start/End Point in display figure.
```

#### 17.74.1.5 read_lonlat_file()

```
def pysar.transect.read_lonlat_file (
              lonlat_file )
```

```
Read Start/End lat/lon from lonlat text file in gmt format.
Inputs:
    lonlat_file : text file in gmt lonlat point file
Outputs:
    start/end_lalo : list of 2 float
```

#### 17.74.1.6 transect_lalo()

```
def pysar.transect.transect_lalo (
              z,
              atr,
              start_lalo,
              end_lalo,
              interpolation = 'nearest' )
```

```
Extract 2D matrix (z) value along the line [start_lalo, end_lalo]
```

#### 17.74.1.7 transect_list()

```
def pysar.transect.transect_list (
              fileList,
              inps )
```

```
Get transection along input line from file list
Inputs:
    fileList : list of str, path of files to get transect
    inps     : Namespace including the following items:
               start/end_lalo
               start/end_yx
               interpolation
Outputs:
    transectList : list of N*2 matrix containing distance and its value
    atrList      : list of attribute dictionary, for each input file
```

#### 17.74.1.8 transect_yx()

```
def pysar.transect.transect_yx (
              z,
              atr,
              start_yx,
              end_yx,
              interpolation = 'nearest' )
```

```
Extract 2D matrix (z) value along the line [x0,y0;x1,y1]
Ref link: http://stackoverflow.com/questions/7878398/how-to-e
          xtract-an-arbitrary-line-of-values-from-a-numpy-array

Inputs:
    z         - (np.array)   2D data matrix
    atr       - (dictionary) 2D data matrix attribute dictionary
    start_yx - (list) y,x coordinate of start point
    end_yx   - (list) y,x coordinate of end   point
    interpolation - sampling/interpolation method, including:
            'nearest'  - nearest neighbour, by default
            'cubic'    - cubic interpolation
            'bilinear' - bilinear interpolation

Output:
    transect - N*2 matrix containing distance - 1st col - and its corresponding
               values - 2nd col - along the line, N is the number of points.

Example:
    transect = transect_yx(dem,demRsc,[10,15],[100,115])
```

### 17.74.2 Variable Documentation

#### 17.74.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  transect.py velocity.h5 -s 5290 5579 -e 12177 482
  transect.py velocity.h5 --start-lalo 30.125 129.988 --end-lalo 30.250 130.116
  transect.py velocity.h5 --line-file  transect_lonlat.xy -d gsi10m.dem
  transect.py AlosA*/velocity.h5 AlosD*/velocity.h5 --line-file  transect_lonlat.xy -d gsi10m.dem
'''
```

## 17.75 pysar.transect_legacy Namespace Reference

### Functions

- def dms2d (Coord)
- def gps_to_LOS (Ve, Vn, theta, heading)
- def check_st_in_box (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def check_st_in_box2 (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def line (x0, y0, x1, y1)

- def dist_point_from_line (m, c, x, y, dx, dy)
- def get_intersect (m, c, x, y)
- def readGPSfile (gpsFile, gps_source)
- def redGPSfile (gpsFile)
- def redGPSfile_cmm4 (gpsFile)
- def nearest (x, tbase, xstep)
- def find_row_column (Lon, Lat, lon, lat, lon_step, lat_step)
- def get_lat_lon (atr)
- def nanmean (data, args)
- def nanstd (data, args)
- def get_transect (z, x0, y0, x1, y1, interpolation='nearest')

  *Option: interpolation : sampling/interpolation method, including: 'nearest' - nearest neighbour, by default 'cubic' - cubic interpolation 'bilinear' - bilinear interpolation.*
- def Usage ()
- def main (argv)
- def onclick (event)

## Variables

- fig = plt.figure()
- ax = fig.add_subplot(111)
- list xc = [ ]
- list yc = [ ]
- cid = fig.canvas.mpl_connect('button_press_event', onclick)
- list x0 = xc[1]
- list y0 = yc[1]
- mf = float(Yf1-Yf0)/float((Xf1-Xf0))
- cf = float(Yf0-mf*Xf0)
- def df0 = dist_point_from_line(mf,cf,x0,y0,1,1)
- def df1 = dist_point_from_line(mf,cf,x1,y1,1,1)
- int mp = -1./mf
- x1 = int((df0+df1)/np.sqrt(1+mp**2)+x0)
- y1 = int(mp*(x1-x0)+y0)
- string Info_aboutFault = 'No'
- length = int(np.hypot(x1-x0, y1-y0))
- x
- y
- zi = z[y.astype(np.int), x.astype(np.int)]
- lat_transect = lat_all[y.astype(np.int), x.astype(np.int)]
- lon_transect = lon_all[y.astype(np.int), x.astype(np.int)]
- int earth_radius = 6371e3;
- float dx = float(atr['X_STEP'])*np.pi/180.0*earth_radius*np.sin(np.mean(lat)*np.pi/180)
- float dy = float(atr['Y_STEP'])*np.pi/180.0*earth_radius
- tuple DX = (x-x0)*dx
- tuple DY = (y-y0)*dy
- D = np.hypot(DX, DY)
- df0_km
- transect = np.zeros([len(D),ntrans])
- list XX0 = [ ]
- list YY0 = [ ]
- m = float(y1-y0)/float((x1-x0))
- c = float(y0-m*x0)
- float m1 = -1.0/m
- list X0 = i*dp/np.sqrt(1+m1**2)+x0

- float Y0 = m1∗(X0-x0)+y0
- X1 = i∗dp/np.sqrt(1+m1∗∗2)+x1
- float Y1 = m1∗(X1-x1)+y1
- transect_lat = np.zeros([len(D),ntrans])
- transect_lon = np.zeros([len(D),ntrans])
- m_prof_edge
- c_prof_edge
- gpsFile
- insarData = z
- fileName
- fileExtension
- Stations
- Lat
- Lon
- Ve
- Se
- Vn
- Sn
- idxRef = Stations.index(refStation)
- Length
- Width
- lat
- lon
- lat_step
- lon_step
- lat_all
- lon_all
- IDYref
- IDXref
- stationsList = Stations
- h5file_theta = h5py.File(incidence_file,'r')
- dset = h5file_theta['mask'].get('mask')
- theta = dset[0:dset.shape[0],0:dset.shape[1]]
- float heading = 193.0∗np.pi/180.0
- list unitVec = [np.cos(heading)∗np.sin(theta),-np.sin(theta)∗np.sin(heading),0]
- def gpsLOS_ref = gps_to_LOS(Ve[idxRef],Vn[idxRef],theta[IDYref,IDXref],heading)
- list GPS = [ ]
- list GPS_station = [ ]
- list GPSx = [ ]
- list GPSy = [ ]
- list GPS_lat = [ ]
- list GPS_lon = [ ]
- idx = Stations.index(st)
- IDY
- IDX
- def gpsLOS = gps_to_LOS(Ve[idx],Vn[idx],theta[IDY,IDX],heading)
- string NoInSAR = 'yes'
- list DistGPS = [ ]
- list GPS_in_bound = [ ]
- list GPS_in_bound_st = [ ]
- list GPSxx = [ ]
- list GPSyy = [ ]
- list gx = GPSx[i]
- list gy = GPSy[i]
- string check_result = 'True'

- def check_result2 = check_st_in_box2(gx,gy,x0,y0,x1,y1,X0,Y0,X1,Y1)
- def dg = dist_point_from_line(m,c,gx,gy,1,1)
- axes
- nrows
- ms

    *ax.fill_between(D/1000.0, (avgInSAR-stdInSAR)∗1000, (avgInSAR+stdInSAR)∗1000,where=(avgInSAR+stdInS←↩ AR)∗1000>=(avgInSAR-stdInSAR)∗1000,alpha=1, facecolor='Red')*
- avgInSAR = np.array(nanmean(transect,axis=1))
- stdInSAR = np.array(nanstd(transect,axis=1))
- fig2
- axes2
- string FaultLine = 'None'
- string figName = 'transect_area.png'

    *Temporary To plot DEM try: majorLocator = MultipleLocator(5) ax.yaxis.set_major_locator(majorLocator) minor←↩ Locator = MultipleLocator(1) ax.yaxis.set_minor_locator(minorLocator)*
- mfc
- linewidth
- string matFile = 'transect.mat'
- dictionary dataset = {}
- color

    *ax.plot(D/1000.0, avgInSAR∗1000, 'r-')*
- alpha
- fontsize
- int lbound = np.nanmin(transect)∗1000

    *lower and higher bounds for diplaying the profile*
- int hbound = np.nanmax(transect)∗1000
- string fault_loc = 'None'
- string ylim = 'no'

## 17.75.1 Function Documentation

### 17.75.1.1 check_st_in_box()

```
def pysar.transect_legacy.check_st_in_box (
            x,
            y,
            x0,
            y0,
            x1,
            y1,
            X0,
            Y0,
            X1,
            Y1 )
```

#### 17.75.1.2 check_st_in_box2()

```
def pysar.transect_legacy.check_st_in_box2 (
              x,
              y,
              x0,
              y0,
              x1,
              y1,
              X0,
              Y0,
              X1,
              Y1 )
```

#### 17.75.1.3 dist_point_from_line()

```
def pysar.transect_legacy.dist_point_from_line (
              m,
              c,
              x,
              y,
              dx,
              dy )
```

#### 17.75.1.4 dms2d()

```
def pysar.transect_legacy.dms2d (
              Coord )
```

#### 17.75.1.5 find_row_column()

```
def pysar.transect_legacy.find_row_column (
              Lon,
              Lat,
              lon,
              lat,
              lon_step,
              lat_step )
```

#### 17.75.1.6 get_intersect()

```
def pysar.transect_legacy.get_intersect (
              m,
              c,
              x,
              y )
```

#### 17.75.1.7 get_lat_lon()

```
def pysar.transect_legacy.get_lat_lon (
              atr )
```

#### 17.75.1.8 get_transect()

```
def pysar.transect_legacy.get_transect (
              z,
              x0,
              y0,
              x1,
              y1,
              interpolation = 'nearest' )
```

Option: interpolation : sampling/interpolation method, including: 'nearest' - nearest neighbour, by default 'cubic' - cubic interpolation 'bilinear' - bilinear interpolation.

#### 17.75.1.9 gps_to_LOS()

```
def pysar.transect_legacy.gps_to_LOS (
              Ve,
              Vn,
              theta,
              heading )
```

#### 17.75.1.10 line()

```
def pysar.transect_legacy.line (
              x0,
              y0,
              x1,
              y1 )
```

#### 17.75.1.11 main()

```
def pysar.transect_legacy.main (
              argv )
```

#### 17.75.1.12 nanmean()

```
def pysar.transect_legacy.nanmean (
              data,
              args )
```

#### 17.75.1.13 nanstd()

```
def pysar.transect_legacy.nanstd (
              data,
              args )
```

#### 17.75.1.14 nearest()

```
def pysar.transect_legacy.nearest (
              x,
              tbase,
              xstep )
```

#### 17.75.1.15 onclick()

```
def pysar.transect_legacy.onclick (
              event )
```

#### 17.75.1.16 readGPSfile()

```
def pysar.transect_legacy.readGPSfile (
              gpsFile,
              gps_source )
```

#### 17.75.1.17 redGPSfile()

```
def pysar.transect_legacy.redGPSfile (
              gpsFile )
```

#### 17.75.1.18 redGPSfile_cmm4()

```
def pysar.transect_legacy.redGPSfile_cmm4 (
              gpsFile )
```

#### 17.75.1.19 Usage()

```
def pysar.transect_legacy.Usage ( )
```

### 17.75.2 Variable Documentation

#### 17.75.2.1 alpha

```
alpha
```

#### 17.75.2.2 avgInSAR

```
avgInSAR = np.array(nanmean(transect,axis=1))
```

#### 17.75.2.3 ax

```
ax = fig.add_subplot(111)
```

#### 17.75.2.4 axes

```
axes
```

#### 17.75.2.5 axes2

```
axes2
```

#### 17.75.2.6 c

```
c = float(y0-m*x0)
```

#### 17.75.2.7 c_prof_edge

```
c_prof_edge
```

#### 17.75.2.8 cf

```
cf = float(Yf0-mf*Xf0)
```

#### 17.75.2.9 check_result

```
def check_result = 'True'
```

#### 17.75.2.10 check_result2

```
def check_result2 = check_st_in_box2(gx,gy,x0,y0,x1,y1,X0,Y0,X1,Y1)
```

#### 17.75.2.11 cid

```
cid = fig.canvas.mpl_connect('button_press_event', onclick)
```

#### 17.75.2.12 color

```
color
```

ax.plot(D/1000.0, avgInSAR∗1000, 'r-')

To plot the Fault location on the profile.

#### 17.75.2.13 D

```
D = np.hypot(DX, DY)
```

#### 17.75.2.14 dataset

```
dictionary dataset = {}
```

#### 17.75.2.15 df0

```
def df0 = dist_point_from_line(mf,cf,x0,y0,1,1)
```

#### 17.75.2.16 df0_km

```
df0_km
```

#### 17.75.2.17 df1

```
def df1 = dist_point_from_line(mf,cf,x1,y1,1,1)
```

#### 17.75.2.18 dg

```
def dg = dist_point_from_line(m,c,gx,gy,1,1)
```

#### 17.75.2.19 DistGPS

```
DistGPS = [ ]
```

#### 17.75.2.20 dset

```
dset = h5file_theta['mask'].get('mask')
```

#### 17.75.2.21 dx

```
dx = float(atr['X_STEP'])*np.pi/180.0*earth_radius*np.sin(np.mean(lat)*np.pi/180)
```

#### 17.75.2.22 DX

```
tuple DX = (x-x0)*dx
```

#### 17.75.2.23 dy

```
dy = float(atr['Y_STEP'])*np.pi/180.0*earth_radius
```

#### 17.75.2.24 DY

```
tuple DY = (y-y0)*dy
```

#### 17.75.2.25 earth_radius

```
int earth_radius = 6371e3;
```

#### 17.75.2.26 fault_loc

```
string fault_loc = 'None'
```

#### 17.75.2.27 FaultLine

```
string FaultLine = 'None'
```

#### 17.75.2.28 fig

```
fig = plt.figure()
```

#### 17.75.2.29 fig2

```
fig2
```

#### 17.75.2.30 figName

```
string figName = 'transect_area.png'
```

Temporary To plot DEM try: majorLocator = MultipleLocator(5) ax.yaxis.set_major_locator(majorLocator) minor↩
Locator = MultipleLocator(1) ax.yaxis.set_minor_locator(minorLocator)

#### 17.75.2.31 fileExtension

```
fileExtension
```

#### 17.75.2.32 fileName

```
fileName
```

#### 17.75.2.33 fontsize

```
fontsize
```

#### 17.75.2.34 GPS

```
list GPS = []
```

#### 17.75.2.35 GPS_in_bound

```
int GPS_in_bound = []
```

#### 17.75.2.36 GPS_in_bound_st

```
list GPS_in_bound_st = []
```

#### 17.75.2.37 GPS_lat

```
list GPS_lat = []
```

#### 17.75.2.38 GPS_lon

```
list GPS_lon = []
```

#### 17.75.2.39 GPS_station

```
list GPS_station = []
```

#### 17.75.2.40 gpsFile

```
gpsFile
```

#### 17.75.2.41 gpsLOS

```
def gpsLOS = gps_to_LOS(Ve[idx],Vn[idx],theta[IDY,IDX],heading)
```

#### 17.75.2.42 gpsLOS_ref

```
def gpsLOS_ref = gps_to_LOS(Ve[idxRef],Vn[idxRef],theta[IDYref,IDXref],heading)
```

#### 17.75.2.43 GPSx

```
list GPSx = [ ]
```

#### 17.75.2.44 GPSxx

```
list GPSxx = [ ]
```

#### 17.75.2.45 GPSy

```
list GPSy = [ ]
```

#### 17.75.2.46 GPSyy

```
list GPSyy = [ ]
```

#### 17.75.2.47 gx

```
list gx = GPSx[i]
```

#### 17.75.2.48 gy

```
list gy = GPSy[i]
```

#### 17.75.2.49 h5file_theta

```
h5file_theta = h5py.File(incidence_file,'r')
```

#### 17.75.2.50 hbound

```
int hbound = np.nanmax(transect)*1000
```

#### 17.75.2.51 heading

```
float heading = 193.0*np.pi/180.0
```

#### 17.75.2.52 idx

```
idx = Stations.index(st)
```

#### 17.75.2.53 IDX

```
IDX
```

#### 17.75.2.54 idxRef

```
idxRef = Stations.index(refStation)
```

#### 17.75.2.55 IDXref

```
IDXref
```

#### 17.75.2.56 IDY

```
IDY
```

#### 17.75.2.57 IDYref

```
IDYref
```

#### 17.75.2.58 Info_aboutFault

```
string Info_aboutFault = 'No'
```

#### 17.75.2.59 insarData

```
insarData = z
```

#### 17.75.2.60 Lat

```
Lat
```

#### 17.75.2.61 lat

```
lat
```

#### 17.75.2.62 lat_all

```
lat_all
```

#### 17.75.2.63 lat_step

```
lat_step
```

#### 17.75.2.64 lat_transect

```
def lat_transect = lat_all[y.astype(np.int), x.astype(np.int)]
```

#### 17.75.2.65 lbound

```
int lbound = np.nanmin(transect)*1000
```

lower and higher bounds for diplaying the profile

#### 17.75.2.66 length

```
length = int(np.hypot(x1-x0, y1-y0))
```

#### 17.75.2.67 Length

```
Length
```

#### 17.75.2.68 linewidth

```
linewidth
```

#### 17.75.2.69 Lon

```
Lon
```

#### 17.75.2.70 lon

```
lon
```

#### 17.75.2.71 lon_all

```
lon_all
```

#### 17.75.2.72 lon_step

```
lon_step
```

#### 17.75.2.73 lon_transect

```
def lon_transect = lon_all[y.astype(np.int), x.astype(np.int)]
```

#### 17.75.2.74 m

```
m = float(y1-y0)/float((x1-x0))
```

#### 17.75.2.75 m1

```
float m1 = -1.0/m
```

#### 17.75.2.76 m_prof_edge

```
m_prof_edge
```

#### 17.75.2.77 matFile

```
string matFile = 'transect.mat'
```

#### 17.75.2.78 mf

```
mf = float(Yf1-Yf0)/float((Xf1-Xf0))
```

#### 17.75.2.79 mfc

```
mfc
```

#### 17.75.2.80 mp

```
int mp = -1./mf
```

#### 17.75.2.81 ms

```
ms
```

ax.fill_between(D/1000.0, (avgInSAR-stdInSAR)*1000, (avgInSAR+stdInSAR)*1000,where=(avgInSAR+stdInS↩AR)*1000>=(avgInSAR-stdInSAR)*1000,alpha=1, facecolor='Red')

#### 17.75.2.82 NoInSAR

```
string NoInSAR = 'yes'
```

#### 17.75.2.83 nrows

```
nrows
```

#### 17.75.2.84 Se

```
Se
```

#### 17.75.2.85 Sn

```
Sn
```

#### 17.75.2.86 Stations

```
Stations
```

#### 17.75.2.87 stationsList

```
stationsList = Stations
```

#### 17.75.2.88 stdInSAR

```
stdInSAR = np.array(nanstd(transect,axis=1))
```

#### 17.75.2.89 theta

```
float theta = dset[0:dset.shape[0],0:dset.shape[1]]
```

#### 17.75.2.90 transect

```
int transect = np.zeros([len(D),ntrans])
```

#### 17.75.2.91 transect_lat

```
transect_lat = np.zeros([len(D),ntrans])
```

#### 17.75.2.92 transect_lon

```
transect_lon = np.zeros([len(D),ntrans])
```

#### 17.75.2.93 unitVec

```
list unitVec = [np.cos(heading)*np.sin(theta),-np.sin(theta)*np.sin(heading),0]
```

#### 17.75.2.94 Ve

```
Ve
```

#### 17.75.2.95 Vn

```
Vn
```

#### 17.75.2.96 Width

```
Width
```

#### 17.75.2.97 x

```
x
```

#### 17.75.2.98 x0

```
list x0 = xc[1]
```

#### 17.75.2.99 X0

```
list X0 = i*dp/np.sqrt(1+m1**2)+x0
```

#### 17.75.2.100 x1

```
x1 = int((df0+df1)/np.sqrt(1+mp**2)+x0)
```

#### 17.75.2.101 X1

```
X1 = i*dp/np.sqrt(1+m1**2)+x1
```

#### 17.75.2.102 xc

```
list xc = []
```

#### 17.75.2.103 XX0

```
list XX0 = [ ]
```

#### 17.75.2.104 y

```
y
```

#### 17.75.2.105 y0

```
list y0 = yc[1]
```

#### 17.75.2.106 Y0

```
float Y0 = m1*(X0-x0)+y0
```

#### 17.75.2.107 y1

```
y1 = int(mp*(x1-x0)+y0)
```

#### 17.75.2.108 Y1

```
float Y1 = m1*(X1-x1)+y1
```

#### 17.75.2.109 yc

```
list yc = [ ]
```

#### 17.75.2.110 ylim

```
string ylim = 'no'
```

#### 17.75.2.111 YY0

```
list YY0 = [ ]
```

#### 17.75.2.112 zi

```
def zi = z[y.astype(np.int), x.astype(np.int)]
```

## 17.76 pysar.tropcor_phase_elevation Namespace Reference

### Functions

- def usage ()
- def main (argv)

### 17.76.1 Function Documentation

#### 17.76.1.1 main()

```
def pysar.tropcor_phase_elevation.main (
              argv )
```

#### 17.76.1.2 usage()

```
def pysar.tropcor_phase_elevation.usage ( )
```

## 17.77 pysar.tropcor_pyaps Namespace Reference

### Functions

- def closest_weather_product_time (sar_acquisition_time, grib_source='ECMWF')
- def get_delay (grib_file, atr, inps_dict)
- def cmdLineParse ()
- def main (argv)

**Variables**

- string EXAMPLE
- string REFERENCE
- string TEMPLATE

## 17.77.1 Function Documentation

### 17.77.1.1 closest_weather_product_time()

```
def pysar.tropcor_pyaps.closest_weather_product_time (
              sar_acquisition_time,
              grib_source = 'ECMWF' )
```

```
Find closest available time of weather product from SAR acquisition time
Inputs:
    sar_acquisition_time - string, SAR data acquisition time in seconds
    grib_source - string, Grib Source of weather reanalysis product
Output:
    grib_hr - string, time of closest available weather product
```

### 17.77.1.2 cmdLineParse()

```
def pysar.tropcor_pyaps.cmdLineParse ( )
```

### 17.77.1.3 get_delay()

```
def pysar.tropcor_pyaps.get_delay (
              grib_file,
              atr,
              inps_dict )
```

### 17.77.1.4 main()

```
def pysar.tropcor_pyaps.main (
              argv )
```

## 17.77.2 Variable Documentation

#### 17.77.2.1 EXAMPLE

string EXAMPLE

**Initial value:**

```
= '''example:
  tropcor_pyaps.py timeseries.h5 -d radar_8rlks.hgt
  tropcor_pyaps.py timeseries.h5 -d radar_8rlks.hgt -s NARR
  tropcor_pyaps.py timeseries.h5 -d radar_8rlks.hgt -s MERRA --delay dry -i 23
  tropcor_pyaps.py timeseries_LODcor.h5 -d radar_8rlks.hgt -s ECMWF
'''
```

#### 17.77.2.2 REFERENCE

string REFERENCE

**Initial value:**

```
= '''reference:
  Jolivet, R., R. Grandin, C. Lasserre, M.-P. Doin and G. Peltzer (2011), Systematic InSAR tropospheric
  phase delay corrections from global meteorological reanalysis data, Geophys. Res. Lett., 38, L17311,
  doi:10.1029/2011GL048757
'''
```

#### 17.77.2.3 TEMPLATE

string TEMPLATE

**Initial value:**

```
= '''
pysar.troposphericDelay.method        = pyaps    #['height-correlation']
pysar.troposphericDelay.weatherModel  = ECMWF    #['ERA', 'MERRA', 'NARR']
'''
```

## 17.78 pysar.tsview_mli Namespace Reference

### Functions

- def transect_yx (z, atr, start_yx, end_yx, interpolation='nearest')

  *Option: interpolation : sampling/interpolation method, including: 'nearest' - nearest neighbour, by default 'cubic' - cubic interpolation 'bilinear' - bilinear interpolation.*
- def transect_lalo (z, atr, start_lalo, end_lalo, interpolation='nearest')
- def transect_list (fileList, start_coord, end_coord, coord_type='radar', interpolation='nearest')
- def usage ()

  *Usage ####################################.*
- def main (argv)

  *Main ####################################.*

### 17.78.1 Function Documentation

#### 17.78.1.1 main()

```
def pysar.tsview_mli.main (
              argv )
```

Main ##################################.

#### 17.78.1.2 transect_lalo()

```
def pysar.tsview_mli.transect_lalo (
              z,
              atr,
              start_lalo,
              end_lalo,
              interpolation = 'nearest' )
```

#### 17.78.1.3 transect_list()

```
def pysar.tsview_mli.transect_list (
              fileList,
              start_coord,
              end_coord,
              coord_type = 'radar',
              interpolation = 'nearest' )
```

#### 17.78.1.4 transect_yx()

```
def pysar.tsview_mli.transect_yx (
              z,
              atr,
              start_yx,
              end_yx,
              interpolation = 'nearest' )
```

Option: interpolation : sampling/interpolation method, including: 'nearest' - nearest neighbour, by default 'cubic' - cubic interpolation 'bilinear' - bilinear interpolation.

Output: transect - N∗2 matrix containing distance and its corresponding values along the line.

#### 17.78.1.5 usage()

```
def pysar.tsview_mli.usage ( )
```

Usage ###################################.

## 17.79 pysar.tsviewer Namespace Reference

### Functions

- def check_yx (xsub, ysub, radius, ax, rectColor='black')
  *Sub Functions ####################################.*
- def read_dis_xy (xsub, ysub, dateList, h5file, unit='cm')
- def read_dis_lalo (lat, lon, dateList, timeseriesFile, radius=0, unit='cm')
- def update_lim (disp_min, disp_max, data_mean, data_std)
- def usage ()
  *Usage ###############################.*
- def main (argv)
  *Main Function #####################################.*

### 17.79.1 Function Documentation

#### 17.79.1.1 check_yx()

```
def pysar.tsviewer.check_yx (
              xsub,
              ysub,
              radius,
              ax,
              rectColor = 'black' )
```

Sub Functions ####################################.

#### 17.79.1.2 main()

```
def pysar.tsviewer.main (
              argv )
```

Main Function #####################################.

#### 17.79.1.3 read_dis_lalo()

```
def pysar.tsviewer.read_dis_lalo (
              lat,
              lon,
              dateList,
              timeseriesFile,
              radius = 0,
              unit = 'cm' )
```

#### 17.79.1.4 read_dis_xy()

```
def pysar.tsviewer.read_dis_xy (
              xsub,
              ysub,
              dateList,
              h5file,
              unit = 'cm' )
```

#### 17.79.1.5 update_lim()

```
def pysar.tsviewer.update_lim (
              disp_min,
              disp_max,
              data_mean,
              data_std )
```

#### 17.79.1.6 usage()

```
def pysar.tsviewer.usage ( )
```

Usage ################################.

## 17.80 pysar.unavco2insarmaps Namespace Reference

### Functions

- def get_date (date_string)
- def get_decimal_date (d)
- def convert_data (attributes, decimal_dates, timeseries_datasets, dataset_keys, json_path, folder_name, region_file_name)
- def make_json_file (chunk_num, points, dataset_keys, json_path, folder_name)
- def build_parser ()
- def main ()

## Variables

- string dbUsername = "INSERT"
- string dbPassword = "INSERT"
- string dbHost = "INSERT"

### 17.80.1 Function Documentation

#### 17.80.1.1 build_parser()

```
def pysar.unavco2insarmaps.build_parser ( )
```

#### 17.80.1.2 convert_data()

```
def pysar.unavco2insarmaps.convert_data (
              attributes,
              decimal_dates,
              timeseries_datasets,
              dataset_keys,
              json_path,
              folder_name,
              region_file_name )
```

#### 17.80.1.3 get_date()

```
def pysar.unavco2insarmaps.get_date (
              date_string )
```

#### 17.80.1.4 get_decimal_date()

```
def pysar.unavco2insarmaps.get_decimal_date (
              d )
```

#### 17.80.1.5 main()

```
def pysar.unavco2insarmaps.main ( )
```

#### 17.80.1.6 make_json_file()

```
def pysar.unavco2insarmaps.make_json_file (
              chunk_num,
              points,
              dataset_keys,
              json_path,
              folder_name )
```

### 17.80.2 Variable Documentation

#### 17.80.2.1 dbHost

```
string dbHost = "INSERT"
```

#### 17.80.2.2 dbPassword

```
string dbPassword = "INSERT"
```

#### 17.80.2.3 dbUsername

```
string dbUsername = "INSERT"
```

## 17.81 pysar.unwrap_error Namespace Reference

### Functions

- def phase_bonding (data, mask, x, y)
- def usage ()
- def main (argv)

### 17.81.1 Function Documentation

#### 17.81.1.1 main()

```
def pysar.unwrap_error.main (
              argv )
```

#### 17.81.1.2 phase_bonding()

```
def pysar.unwrap_error.phase_bonding (
              data,
              mask,
              x,
              y )
```

#### 17.81.1.3 usage()

```
def pysar.unwrap_error.usage ( )
```

## 17.82 pysar.view Namespace Reference

### Classes

- class Basemap2
  *Class ################################################.*

### Functions

- def add_inner_title (ax, title, loc, size=None, kwargs)
- def auto_flip_direction (atr_dict)
- def auto_figure_title (meta_dict, inps)
- def auto_row_col_num (subplot_num, data_shape, fig_size, fig_num=1)
- def check_colormap_input (atr_dict, colormap=None)
- def check_multilook_input (pixel_box, row_num, col_num)
- def get_epoch_full_list_from_input (all_epoch_list, epoch_input_list=[ ], epoch_num_input_list=[ ])
- def plot_dem_lalo (bmap, dem, box, inps_dict)
- def plot_dem_yx (ax, dem, inps_dict)
- def round_to_1 (x)
- def scale_data2disp_unit (matrix, atr_dict, disp_unit)
- def update_plot_inps_with_display_setting_file (inps, disp_set_file)
- def update_plot_inps_with_meta_dict (inps, meta_dict)
- def update_matrix_with_plot_inps (data, meta_dict, inps)
- def plot_matrix (ax, data, meta_dict, inps=None)
- def cmdLineParse (argv)
- def main (argv)
  *Main Function ########################################.*

### Variables

- string EXAMPLE
- string PLOT_TEMPLATE

## 17.82.1 Function Documentation

### 17.82.1.1 add_inner_title()

```
def pysar.view.add_inner_title (
              ax,
              title,
              loc,
              size = None,
              kwargs )
```

### 17.82.1.2 auto_figure_title()

```
def pysar.view.auto_figure_title (
              meta_dict,
              inps )
```

```
Get auto figure title from meta dict and input options
```

### 17.82.1.3 auto_flip_direction()

```
def pysar.view.auto_flip_direction (
              atr_dict )
```

```
Check flip left-right and up-down based on attribute dict, for radar-coded file only
```

#### 17.82.1.4 auto_row_col_num()

def pysar.view.auto_row_col_num (
*subplot_num,*
*data_shape,*
*fig_size,*
*fig_num = 1* )

```
Get optimal row and column number given figure size number of subplots

Inputs:
    subplot_num : int, total number of subplots
    data_shape  : list of 2 float, data size in pixel in row and column direction of each plot
    fig_size    : list of 2 float, figure window size in inches
    fig_num     : int, number of figure windows, optional, default = 1.

Outputs:
    row_num : number of subplots in row    direction per figure
    col_num : number of subplots in column direction per figure
```

#### 17.82.1.5 check_colormap_input()

def pysar.view.check_colormap_input (
*atr_dict,*
*colormap = None* )

#### 17.82.1.6 check_multilook_input()

def pysar.view.check_multilook_input (
*pixel_box,*
*row_num,*
*col_num* )

#### 17.82.1.7 cmdLineParse()

def pysar.view.cmdLineParse (
*argv* )

#### 17.82.1.8 get_epoch_full_list_from_input()

def pysar.view.get_epoch_full_list_from_input (
*all_epoch_list,*
*epoch_input_list = [],*
*epoch_num_input_list = []* )

```
Read/Get input epoch list from input epoch and epoch_num
```

#### 17.82.1.9 main()

```
def pysar.view.main (
              argv )
```

Main Function ##########################################.

#### 17.82.1.10 plot_dem_lalo()

```
def pysar.view.plot_dem_lalo (
              bmap,
              dem,
              box,
              inps_dict )
```

```
Plot DEM in geo-coordinate
Inputs:
    bmap  : basemap object
    dem   : dem data, 2D np.int16 matrix
    box   : geo bounding box, 4-tuple as (urcrnrlon,urcrnrlat,llcrnrlon,llcrnrlat)
    inps_dict : dict with the following 5 items:
                'disp_dem_shade'    : bool,  True/False
                'disp_dem_contour'  : bool,  True/False
                'dem_contour_step'  : float, 200.0
                'dem_contour_smooth': float, 3.0

Examples:
    dem_disp_dict = {'dem': 'gsi10m_30m.dem', 'disp_dem_shade': True, 'disp_dem_contour': True,\
                     'dem_contour_step': 200.0, 'dem_contour_smooth': 3.0}
    bmap = plot_dem_lalo(bmap,dem,geo_box,dem_inps_dict)
```

#### 17.82.1.11 plot_dem_yx()

```
def pysar.view.plot_dem_yx (
              ax,
              dem,
              inps_dict )
```

```
Plot DEM in radar coordinate
Inputs:
    ax         : matplotlib axes object
    dem        : dem data, 2D np.int16 matrix
    inps_dict : dict with the following 5 items:
                'disp_dem_shade'    : bool,  True/False
                'disp_dem_contour'  : bool,  True/False
                'dem_contour_step'  : float, 200.0
                'dem_contour_smooth': float, 3.0

Examples:
    dem_disp_dict = {'dem': 'gsi10m_30m.dem', 'disp_dem_shade': True, 'disp_dem_contour': True,\
                     'dem_contour_step': 200.0, 'dem_contour_smooth': 3.0}
    ax = plot_dem_yx(ax,dem,dem_disp_dict)
```

#### 17.82.1.12 plot_matrix()

def pysar.view.plot_matrix (
*ax,*
*data,*
*meta_dict,*
*inps = None* )

```
Plot 2D matrix

Inputs:
    ax   : matplot.pyplot axes object
    data : 2D np.array,
    meta_dict : dictionary, attributes of data
    inps : Namespace, optional, input options for display

Outputs:
    ax  : matplot.pyplot axes object

Example:
    import matplotlib.pyplot as plt
    import pysar._readfile as readfile
    import pysar.view as view

    data, atr = readfile.read('velocity.h5')
    fig = plt.figure()
    ax = fig.add_axes([0.1,0.1,0.8,0.8])
    ax = view.plot_matrix(ax, data, atr)
    plt.show()
```

#### 17.82.1.13 round_to_1()

def pysar.view.round_to_1 (
*x* )

```
Return the most significant digit of input number
```

#### 17.82.1.14 scale_data2disp_unit()

def pysar.view.scale_data2disp_unit (
*matrix,*
*atr_dict,*
*disp_unit* )

```
Scale data based on data unit and display unit
Inputs:
    matrix    : 2D np.array
    atr_dict  : dictionary, meta data
    disp_unit : str, display unit
Outputs:
    matrix    : 2D np.array, data after scaling
    disp_unit : str, display unit
Default data file units in PySAR are:  m, m/yr, radian, 1
```

#### 17.82.1.15 update_matrix_with_plot_inps()

```
def pysar.view.update_matrix_with_plot_inps (
              data,
              meta_dict,
              inps )
```

#### 17.82.1.16 update_plot_inps_with_display_setting_file()

```
def pysar.view.update_plot_inps_with_display_setting_file (
              inps,
              disp_set_file )
```

```
Update inps using values from display setting file
```

#### 17.82.1.17 update_plot_inps_with_meta_dict()

```
def pysar.view.update_plot_inps_with_meta_dict (
              inps,
              meta_dict )
```

### 17.82.2 Variable Documentation

#### 17.82.2.1 EXAMPLE

```
string EXAMPLE
```

**Initial value:**

```
= '''example:
  view.py SanAndreas.dem
  view.py velocity.h5 -u cm -m -2 -M 2 -c bwr --mask Mask_tempCoh.h5 -d SanAndreas.dem

  view.py timeseries.h5
  view.py unwrapIfgram.h5 070927-100217
  view.py Wrapped.h5    -n 5
  view.py geomap_4rlks.trans range

  # Display in subset:
  view.py velocity.h5 -x 100 600     -y 200 800
  view.py velocity.h5 -l 31.05 31.10 -L 130.05 130.10

  # Exclude Dates:
  view.py timeseries.h5 -ex drop_date.txt

  # Reference:
  view.py velocity.h5   --ref-yx   210 566
  view.py timeseries.h5 --ref-date 20101120

  # Save and Output:
  view.py velocity.h5 --save
  view.py velocity.h5 -o velocity.pdf
  view.py velocity.h5 --nodisplay
'''
```

#### 17.82.2.2 PLOT_TEMPLATE

```
string PLOT_TEMPLATE
```

**Initial value:**

```
= '''Plot Setting:
  plot.name          = 'Yunjun et al., 2016, AGU, Fig 4f'
  plot.type          = LOS_VELOCITY
  plot.startDate     =
  plot.endDate       =
  plot.displayUnit   = cm/yr
  plot.displayMin    = -2
  plot.displayMax    = 2
  plot.colormap      = jet
  plot.subset.lalo   = 33.05:33.15, 131.15:131.27
  plot.seed.lalo = 33.0651, 131.2076
'''
```

## 17.83 pysar.view_legacy Namespace Reference

### Functions

- def add_inner_title (ax, title, loc, size=None, kwargs)

  *Sub Function ########################################.*
- def rewrap (data, atr)
- def unit_and_scale (data_unit, display_unit)
- def unit_type (unit_in)
- def orbit_direction (atr)
- def auto_flip_check (atr_dict)
- def plot_dem_lalo (bmap, dem, geo_box, demShade='yes', demContour='no', contour_step=200.0, contour_sigma=3.0)

  *Examples: bmap = plot_dem_lalo(bmap,dem,geo_box,'no','yes')*
- def plot_dem_yx (ax, dem, demShade='yes', demContour='no', contour_step=200.0, contour_sigma=3.0)

  *Examples: ax = plot_dem_yx(ax,dem,'no','yes')*
- def usage ()

  *Usage ########################.*
- def main (argv)

  *Main Function ########################################.*

### 17.83.1 Function Documentation

#### 17.83.1.1 add_inner_title()

```
def pysar.view_legacy.add_inner_title (
                ax,
                title,
                loc,
                size = None,
                kwargs )
```

Sub Function ########################################.

#### 17.83.1.2 auto_flip_check()

```
def pysar.view_legacy.auto_flip_check (
              atr_dict )
```

#### 17.83.1.3 main()

```
def pysar.view_legacy.main (
              argv )
```

Main Function ########################################.

#### 17.83.1.4 orbit_direction()

```
def pysar.view_legacy.orbit_direction (
              atr )
```

#### 17.83.1.5 plot_dem_lalo()

```
def pysar.view_legacy.plot_dem_lalo (
              bmap,
              dem,
              geo_box,
              demShade = 'yes',
              demContour = 'no',
              contour_step = 200.0,
              contour_sigma = 3.0 )
```

Examples: bmap = plot_dem_lalo(bmap,dem,geo_box,'no','yes')

#### 17.83.1.6 plot_dem_yx()

```
def pysar.view_legacy.plot_dem_yx (
              ax,
              dem,
              demShade = 'yes',
              demContour = 'no',
              contour_step = 200.0,
              contour_sigma = 3.0 )
```

Examples: ax = plot_dem_yx(ax,dem,'no','yes')

#### 17.83.1.7 rewrap()

```
def pysar.view_legacy.rewrap (
            data,
            atr )
```

#### 17.83.1.8 unit_and_scale()

```
def pysar.view_legacy.unit_and_scale (
            data_unit,
            display_unit )
```

#### 17.83.1.9 unit_type()

```
def pysar.view_legacy.unit_type (
            unit_in )
```

#### 17.83.1.10 usage()

```
def pysar.view_legacy.usage ( )
```

Usage ######################.

## 17.84 troposphere_uncertainty Namespace Reference

### Functions

- def createParser ()
- def cmdLineParse (iargs=None)
- def velocity_uncertainty_vs_distance (inps)
- def statistics (inps)
- def estimate_seasonal (inps)
- def velocity_uncertainty (realtive_std_file, inps)
- def download (inps)
- def main (iargs=None)

### 17.84.1 Function Documentation

#### 17.84.1.1 cmdLineParse()

```
def troposphere_uncertainty.cmdLineParse (
              iargs = None )
```

#### 17.84.1.2 createParser()

```
def troposphere_uncertainty.createParser ( )
```

#### 17.84.1.3 download()

```
def troposphere_uncertainty.download (
              inps )
```

#### 17.84.1.4 estimate_seasonal()

```
def troposphere_uncertainty.estimate_seasonal (
              inps )
```

#### 17.84.1.5 main()

```
def troposphere_uncertainty.main (
              iargs = None )
```

#### 17.84.1.6 statistics()

```
def troposphere_uncertainty.statistics (
              inps )
```

#### 17.84.1.7 velocity_uncertainty()

```
def troposphere_uncertainty.velocity_uncertainty (
              realtive_std_file,
              inps )
```

#### 17.84.1.8 velocity_uncertainty_vs_distance()

```
def troposphere_uncertainty.velocity_uncertainty_vs_distance (
              inps )
```

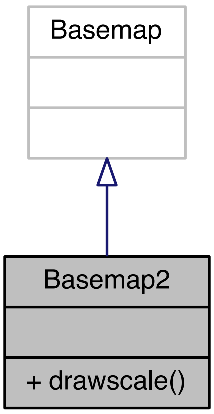
# Chapter 18

# Class Documentation

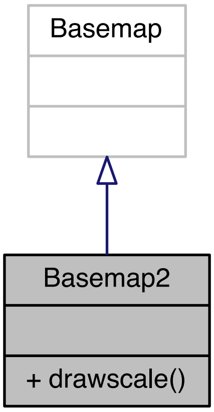
## 18.1 Basemap2 Class Reference

Class ################################################.

Inheritance diagram for Basemap2:



Collaboration diagram for Basemap2:




### Public Member Functions

- def drawscale (self, lat_c, lon_c, dist, font_size=12, yoffset=None)

## 18.1.1 Detailed Description

Class ###################################################.

## 18.1.2 Member Function Documentation

### 18.1.2.1 drawscale()

```
def drawscale (
              self,
              lat_c,
              lon_c,
              dist,
              font_size = 12,
              yoffset = None )
```

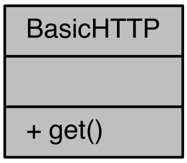
```
draw a simple map scale from x1,y to x2,y in map projection
coordinates, label it with actual distance in km
Inputs:
    lat_c/lon_c : float, longitude and latitude of scale bar center, in degree
    dist        : float, distance of scale bar, in m
    yoffset     : float, optional, scale bar length at two ends, in degree
Example:
    m.drawscale(33.06, 131.18, 2000)
ref_link: http://matplotlib.1069221.n5.nabble.com/basemap-scalebar-td14133.html
```

The documentation for this class was generated from the following file:

- /Users/jeromezhang/Documents/development/python/PySAR/pysar/view.py

## 18.2 BasicHTTP Class Reference

Collaboration diagram for BasicHTTP:




### Static Public Member Functions

- def get (url)

### 18.2.1 Detailed Description

### 18.2.2 Member Function Documentation

#### 18.2.2.1 get()

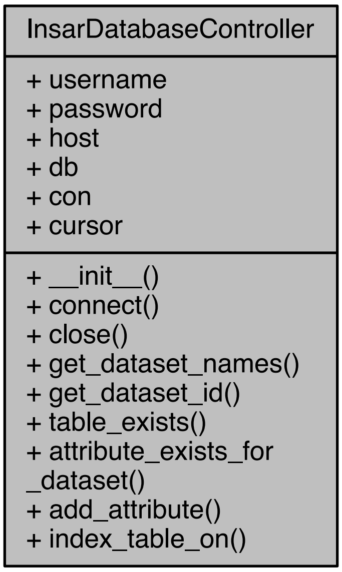
```
def get (
             url )  [static]
```

The documentation for this class was generated from the following file:

- /Users/jeromezhang/Documents/development/python/PySAR/pysar/insarmaps_query.py

## 18.3 InsarDatabaseController Class Reference

Collaboration diagram for InsarDatabaseController:




### Public Member Functions

- def __init__ (self, username, password, host, db)
- def connect (self)
- def close (self)
- def get_dataset_names (self)
- def get_dataset_id (self, dataset)
- def table_exists (self, table)
- def attribute_exists_for_dataset (self, dataset, attributekey)
- def add_attribute (self, dataset, attributekey, attributevalue)
- def index_table_on (self, table, on)

### Public Attributes

- username
- password
- host
- db
- con
- cursor

### 18.3.1 Detailed Description

### 18.3.2 Constructor & Destructor Documentation

#### 18.3.2.1 __init__()

```
def __init__ (
            self,
            username,
            password,
            host,
            db )
```

### 18.3.3 Member Function Documentation

#### 18.3.3.1 add_attribute()

```
def add_attribute (
            self,
            dataset,
            attributekey,
            attributevalue )
```

#### 18.3.3.2 attribute_exists_for_dataset()

```
def attribute_exists_for_dataset (
            self,
            dataset,
            attributekey )
```

#### 18.3.3.3 close()

```
def close (
            self )
```

#### 18.3.3.4 connect()

```
def connect (
              self )
```

#### 18.3.3.5 get_dataset_id()

```
def get_dataset_id (
              self,
              dataset )
```

#### 18.3.3.6 get_dataset_names()

```
def get_dataset_names (
              self )
```

#### 18.3.3.7 index_table_on()

```
def index_table_on (
              self,
              table,
              on )
```

#### 18.3.3.8 table_exists()

```
def table_exists (
              self,
              table )
```

### 18.3.4 Member Data Documentation

#### 18.3.4.1 con

```
con
```

#### 18.3.4.2 cursor

`cursor`

#### 18.3.4.3 db

`db`

#### 18.3.4.4 host

`host`

#### 18.3.4.5 password

`password`

#### 18.3.4.6 username

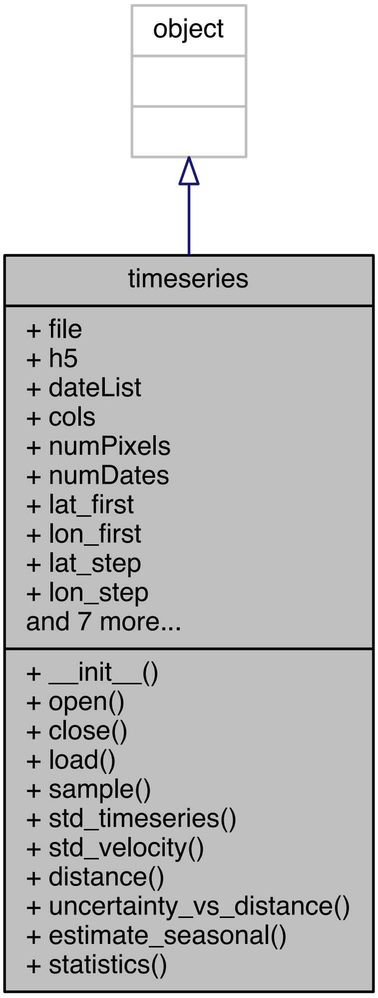
`username`

The documentation for this class was generated from the following file:

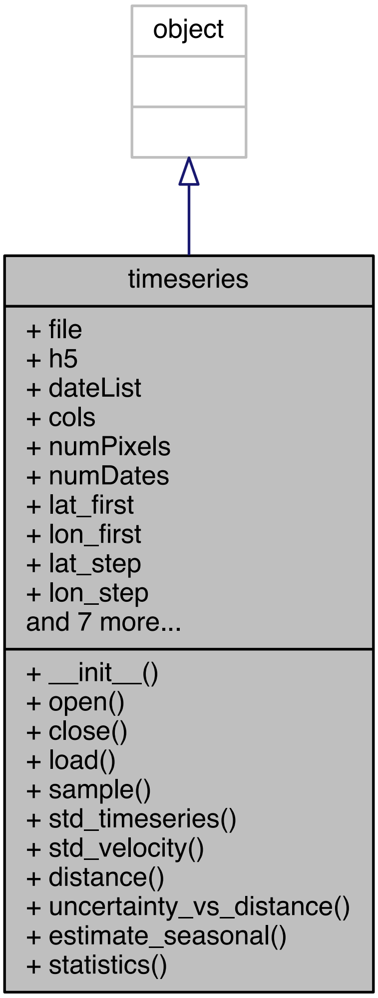
- /Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attributes_insarmaps.py

## 18.4 timeseries Class Reference

Inheritance diagram for timeseries:



Collaboration diagram for timeseries:




## Public Member Functions

- def __init__ (self, file=None)
- def open (self)
- def close (self)
- def load (self)
- def sample (self, numSamples=500, mask=None)
- def std_timeseries (self, ref)
- def std_velocity (self, sar_dates)
- def distance (self, i)
- def uncertainty_vs_distance (self, sar_dates)
- def estimate_seasonal (self, inps)
- def statistics (self, inps)

## Public Attributes

- file
- h5
- dateList
- cols
- numPixels
- numDates
- lat_first
- lon_first
- lat_step
- lon_step
- lat
- lon
- Data
- idx
- relative_std
- relative_std_velocity
- dist

### 18.4.1 Detailed Description

### 18.4.2 Constructor & Destructor Documentation

#### 18.4.2.1 __init__()

```
def __init__ (
                self,
                file = None )
```

### 18.4.3 Member Function Documentation

#### 18.4.3.1 close()

```
def close (
                self )
```

#### 18.4.3.2 distance()

```
def distance (
                self,
                i )
```

#### 18.4.3.3 estimate_seasonal()

```
def estimate_seasonal (
              self,
              inps )
```

#### 18.4.3.4 load()

```
def load (
              self )
```

#### 18.4.3.5 open()

```
def open (
              self )
```

#### 18.4.3.6 sample()

```
def sample (
              self,
              numSamples = 500,
              mask = None )
```

#### 18.4.3.7 statistics()

```
def statistics (
              self,
              inps )
```

#### 18.4.3.8 std_timeseries()

```
def std_timeseries (
              self,
              ref )
```

#### 18.4.3.9 std_velocity()

```
def std_velocity (
              self,
              sar_dates )
```

#### 18.4.3.10 uncertainty_vs_distance()

```
def uncertainty_vs_distance (
              self,
              sar_dates )
```

### 18.4.4 Member Data Documentation

#### 18.4.4.1 cols

```
cols
```

#### 18.4.4.2 Data

```
Data
```

#### 18.4.4.3 dateList

```
dateList
```

#### 18.4.4.4 dist

```
dist
```

#### 18.4.4.5 file

```
file
```

#### 18.4.4.6 h5

```
h5
```

#### 18.4.4.7 idx

```
idx
```

#### 18.4.4.8 lat

```
lat
```

#### 18.4.4.9 lat_first

```
lat_first
```

#### 18.4.4.10 lat_step

```
lat_step
```

#### 18.4.4.11 lon

```
lon
```

#### 18.4.4.12 lon_first

```
lon_first
```

#### 18.4.4.13 lon_step

```
lon_step
```

#### 18.4.4.14 numDates

```
numDates
```

#### 18.4.4.15 numPixels

```
numPixels
```

#### 18.4.4.16 relative_std

```
relative_std
```

#### 18.4.4.17 relative_std_velocity

```
relative_std_velocity
```

The documentation for this class was generated from the following file:

- /Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/delayTimeseries.py

# Chapter 19

# File Documentation

## 19.1 /Users/jeromezhang/Documents/development/python/PySAR.wiki/_Sidebar.md File Reference

## 19.2 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Attributes.md File Reference

## 19.3 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Coordinate.md File Reference

## 19.4 /Users/jeromezhang/Documents/development/python/PySAR.wiki/File-Descriptions.md File Reference

## 19.5 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Gamma-File-↩ Decription.md File Reference

## 19.6 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Google-Earth.md File Reference

## 19.7 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Home.md File Reference

## 19.8 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Mask.md File Reference

## 19.9 /Users/jeromezhang/Documents/development/python/PySAR.wiki/pysarApp.md File Reference

## 19.10 /Users/jeromezhang/Documents/development/python/PySAR.wiki/UNAVCO-InSA↩ R-Archive.md File Reference

## 19.11 /Users/jeromezhang/Documents/development/python/PySAR.wiki/Web-Viewer.md File Reference

## 19.12 /Users/jeromezhang/Documents/development/python/PySAR/pysar/__init__.py File Reference

### Namespaces

- pysar

### Variables

- bool miami_path = True

## 19.13 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_datetime.py File Reference

### Namespaces

- pysar._datetime

### Functions

- def yyyymmdd2years (dates)
- def yymmdd2yyyymmdd (date)
- def yyyymmdd (dates)
- def yymmdd (dates)
- def igram_date_list (igramFile)
- def read_date_list (date_list_file)
- def date_index (dateList)
- def date_list2tbase (dateList)
- def date_list2vector (dateList)
- def auto_adjust_xaxis_date (ax, datevector, fontSize=12)

## 19.14 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_gmt.py File Reference

### Namespaces

- pysar._gmt

### Functions

- def write_gmt_simple (lons, lats, z, fname, title='default', name='z', scale=1.0, offset=0, units='meters')

## 19.15 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_network.py File Reference

### Namespaces

- pysar._network

### Functions

- def read_pairs_list (date12ListFile, dateList=[ ])
- def write_pairs_list (pairs, dateList, outName)
- def read_igram_pairs (igramFile)
- def read_baseline_file (baselineFile, exDateList=[ ])
- def date12_list2index (date12_list, date_list=[ ])
- def get_date12_list (File)
- def igram_perp_baseline_list (File)
- def threshold_perp_baseline (igramIdxList, perpBaseList, perpBaseMax=800, perpBaseMin=0)
- def threshold_temporal_baseline (igramIdxList, tempBaseList, tempBaseMax=365, seasonal=1, tempBase↩ Min=0)
- def pair_sort (pairs)
- def pair_merge (pairs1, pairs2)
- def select_pairs_all (dateList)
- def select_pairs_delaunay (tempBaseList, perpBaseList, normalize=1)
- def select_pairs_sequential (dateList, num_incr=2)
- def select_pairs_hierarchical (tempBaseList, perpBaseList, tempPerpList)
- def select_pairs_mst (tempBaseList, perpBaseList, normalize=1)
- def select_pairs_star (dateList, m_date)
- def plot_network (ax, pairs_idx, date8List, bperpList, plot_dict={})
- def plot_perp_baseline_hist (ax, date8List, bperpList, plot_dict={})
- def auto_adjust_yaxis (ax, dataList, fontSize=12)

## 19.16 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_pysar_↩ utilities.py File Reference

### Namespaces

- pysar._pysar_utilities

### Functions

- def incidence_angle (atr, dimension=2)
- def which (program)
- def get_file_stack (File, maskFile=None)
- def nonzero_mask (File, outFile='Mask.h5')
- def spatial_average (File, mask=None, box=None, saveList=False)
- def temporal_average (File, outFile=None)
- def get_file_list (fileList)
- def print_progress (iteration, total, prefix='calculating:', suffix='complete', decimals=1, barLength=50)
- def glob2radar (lat, lon, geomapFile='geomap ∗.trans', rdrFile=None)
- def radar2glob (az, rg, geomapFile='geomap ∗.trans', rdrFile=None)
- def radar_or_geo (File)
- def check_variable_name (path)
- def hillshade (data, scale)
- def date_list (h5file)
- def YYYYMMDD2years (d)
- def design_matrix (h5file)
- def timeseries_inversion (igramsFile, timeseriesFile)
- def timeseries_inversion_FGLS (h5flat, h5timeseries)
- def timeseries_inversion_L1 (h5flat, h5timeseries)
- def Baseline_timeseries (igramsFile)
- def dBh_dBv_timeseries (igramsFile)
- def Bh_Bv_timeseries (igramsFile)
- def stacking (File)
- def yymmdd2YYYYMMDD (date)
- def yyyymmdd (dates)
- def yymmdd (dates)
- def make_triangle (dates12, igram1, igram2, igram3)
- def get_triangles (h5file)
- def generate_curls (curlfile, h5file, Triangles, curls)

## 19.17 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_readfile.py File Reference

### Namespaces

- pysar._readfile

### Functions

- def read (File, box=(), epoch='')
- def read_attribute (File, epoch='')
- def check_variable_name (path)
- def read_template (File, delimiter='=')
- def read_roipac_rsc (File)
- def read_gamma_par (File)
- def read_isce_xml (File)
- def merge_attribute (atr1, atr2)
- def read_float32 (File, box=None)
- def read_complex_float32 (File, real_imag=False)

- def read_real_float32 (File)
- def read_complex_int16 (File, box=None, real_imag=False)
- def read_dem (File)
- def read_real_int16 (File)
- def read_flag (File)
- def read_GPS_USGS (File)
- def read_multiple (File, box='')

### Variables

- list multi_group_hdf5_file = ['interferograms','coherence','wrapped','snaphu_connect_component']
- list multi_dataset_hdf5_file = ['timeseries']
- list single_dataset_hdf5_file = ['dem','mask','rmse','temporal_coherence', 'velocity']

## 19.18 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_remove_surface.py File Reference

### Namespaces

- pysar._remove_surface

### Functions

- def remove_data_surface (data, mask, surf_type='plane')
- def remove_data_multiple_surface (data, mask, surf_type, ysub)
- def remove_surface (File, surf_type, maskFile=None, outFile=None, ysub=None)

## 19.19 /Users/jeromezhang/Documents/development/python/PySAR/pysar/_writefile.py File Reference

### Namespaces

- pysar._writefile

### Functions

- def write (args)
- def write_float32 (args)
- def write_complex64 (data, outname)
- def write_real_int16 (data, outname)
- def write_dem (data, outname)
- def write_real_float32 (data, outname)
- def write_complex_int16 (data, outname)

## 19.20 /Users/jeromezhang/Documents/development/python/PySAR/pysar/add.py File Reference

### Namespaces

- pysar.add

### Functions

- def add (data1, data2)
- def usage ()
- def main (argv)

## 19.21 /Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attribute.py File Reference

### Namespaces

- pysar.add_attribute

### Functions

- def usage ()
- def main (argv)

## 19.22 /Users/jeromezhang/Documents/development/python/PySAR/pysar/add_attributes↵_insarmaps.py File Reference

### Classes

- class InsarDatabaseController

### Namespaces

- pysar.add_attributes_insarmaps

### Functions

- def usage ()
- def parse_file_for_attributes (file)
- def build_parser ()
- def main (argv)

## 19.23 /Users/jeromezhang/Documents/development/python/PySAR/pysar/asc_desc.py File Reference

### Namespaces

- pysar.asc_desc

### Functions

- def usage ()
- def corners (h5V1)
- def nearest_neighbor (x, y, tbase, pbase)
- def nearest (x, X)
- def find_row_column (Lon, Lat, h5file)
- def get_lat_lon (h5file)
- def main (argv)

## 19.24 /Users/jeromezhang/Documents/development/python/PySAR/pysar/baseline_↩ error.py File Reference

### Namespaces

- pysar.baseline_error

### Functions

- def to_percent (y, position)
- def usage ()
- def main (argv)

## 19.25 /Users/jeromezhang/Documents/development/python/PySAR/pysar/baseline_↩ trop.py File Reference

### Namespaces

- pysar.baseline_trop

### Functions

- def to_percent (y, position)
- def usage ()
- def main (argv)

## 19.26 /Users/jeromezhang/Documents/development/python/PySAR/pysar/convert2mat.py File Reference

### Namespaces

- pysar.convert2mat

### Functions

- def usage ()
- def yyyymmdd2years (date)
- def main (argv)

## 19.27 /Users/jeromezhang/Documents/development/python/PySAR/pysar/correct_↩ dem.py File Reference

### Namespaces

- pysar.correct_dem

### Functions

- def usage ()
- def main (argv)

## 19.28 /Users/jeromezhang/Documents/development/python/PySAR/pysar/correlation_↩ with_dem.py File Reference

### Namespaces

- pysar.correlation_with_dem

### Functions

- def usage ()

### Variables

- amp
- dem = dem[int(suby[0]):int(suby[1]),int(subx[0]):int(subx[1])]
- demRsc
- h5data = h5py.File(File)
- dset = h5data['velocity'].get('velocity')
- data = dset[0:dset.shape[0],0:dset.shape[1]]
- suby = sys.argv[3].split(':')
- subx = sys.argv[4].split(':')
- ndx = ∼np.isnan(data)
- C1 = np.zeros([2,len(dem[ndx])])

## 19.29 /Users/jeromezhang/Documents/development/python/PySAR/pysar/dem_error.py File Reference

### Namespaces

- pysar.dem_error

### Functions

- def usage ()
- def main (argv)

## 19.30 /Users/jeromezhang/Documents/development/python/PySAR/pysar/diff.py File Reference

### Namespaces

- pysar.diff

### Functions

- def diff (data1, data2)
- def usage ()
- def main (argv)

## 19.31 /Users/jeromezhang/Documents/development/python/PySAR/pysar/drop_turbulence.py File Reference

### Namespaces

- pysar.drop_turbulence

### Functions

- def circle_index (atr, circle_par)
- def usage ()

  *Usage ####################################.*
- def main (argv)

  *Main Function ################################.*

## 19.32 /Users/jeromezhang/Documents/development/python/PySAR/pysar/filter_spatial.py File Reference

### Namespaces

- pysar.filter_spatial

### Functions

- def usage ()
- def filter (data, filtType, par)
- def multilook (ifg, lksy, lksx)
- def main (argv)

## 19.33 /Users/jeromezhang/Documents/development/python/PySAR/pysar/filter_temporal.py File Reference

### Namespaces

- pysar.filter_temporal

### Functions

- def get_data (h5timeseries)
- def usage ()
- def main (argv)

## 19.34 /Users/jeromezhang/Documents/development/python/PySAR/pysar/gamma_↩ view.py File Reference

### Namespaces

- pysar.gamma_view

### Functions

- def usage ()
- def main (argv)

## 19.35 /Users/jeromezhang/Documents/development/python/PySAR/pysar/generate_↩ mask.py File Reference

### Namespaces

- pysar.generate_mask

### Functions

- def usage ()
- def main (argv)

## 19.36 /Users/jeromezhang/Documents/development/python/PySAR/pysar/geocode.py File Reference

### Namespaces

- pysar.geocode

### Functions

- def geomap4subset_radar_file (radar_atr, geomap_file)
- def geocode_data_roipac (data, geomapFile, outname)

  *Geocode one data ########################.*
- def geocode_attribute (atr_rdr, atr_geo)
- def geocode_file_roipac (infile, geomap_file, outfile=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 19.37 /Users/jeromezhang/Documents/development/python/PySAR/pysar/igram_closure.py File Reference

### Namespaces

- pysar.igram_closure

### Functions

- def usage ()
- def main (argv)

## 19.38 /Users/jeromezhang/Documents/development/python/PySAR/pysar/igram_inversion.py File Reference

### Namespaces

- pysar.igram_inversion

### Functions

- def usage ()
- def main (argv)

## 19.39 /Users/jeromezhang/Documents/development/python/PySAR/pysar/image_math.py File Reference

### Namespaces

- pysar.image_math

### Functions

- def operation (data, operator, operand)

  *Sub Functions ######################### Operation ####################.*
- def add (data1, data2)

  *Image Add ####################.*
- def diff (data1, data2)

  *Image Diff ####################.*
- def usage ()

  *Usage ######################.*
- def main (argv)

  *Main Functions ########################.*

## 19.40 /Users/jeromezhang/Documents/development/python/PySAR/pysar/incidence_↩ angle.py File Reference

### Namespaces

- pysar.incidence_angle

### Functions

- def usage ()
- def main (argv)

## 19.41 /Users/jeromezhang/Documents/development/python/PySAR/pysar/info.py File Reference

### Namespaces

- pysar.info

### Functions

- def print_attributes (atr, sorting=True)
- def print_hdf5_structure (File)

  *By andrewcollette at* `https://github.com/h5py/h5py/issues/406`.
- def print_timseries_date_info (dateList)
- def usage ()
- def main (argv)

## 19.42 /Users/jeromezhang/Documents/development/python/PySAR/pysar/insar_vs_gps.py File Reference

### Namespaces

- pysar.insar_vs_gps

### Functions

- def readGPSfile (gpsFile, gps_source)
- def nearest (x, tbase, xstep)
- def find_row_column (Lon, Lat, lon, lat, lon_step, lat_step)
- def usage ()
- def main (argv)

### Variables

- Stations

  *finding the raw an column of the reference gps station and referencing insar data to this pixel*
- Lat
- Lon
- Ve
- Se
- Vn
- Sn
- Vu
- Su
- idxRef = Stations.index(refStation)
- IDYref
- IDXref
- insarData = insarData - insarData[IDYref][IDXref]

  *Stations, gpsData = redGPSfile(gpsFile) idxRef=Stations.index(refStation) Lat,Lon,Vn,Ve,Sn,Se,Corr,Vu,Su = gpsData[idxRef,:] IDYref,IDXref=find_row_column(Lon,Lat,lon,lat,lon_step,lat_step)*
- stationsList = Stations
- look_n = float(h5file['velocity'].attrs['LOOK_REF1'])
- look_f = float(h5file['velocity'].attrs['LOOK_REF2'])
- tuple theta = (look_n+look_f)/2.
- heading = float(h5file['velocity'].attrs['HEADING'])
- list unitVec = [np.cos(heading)∗np.sin(theta),-np.sin(theta)∗np.sin(heading),-np.cos(theta)]
- string gps_comp_txt = ' projecting three gps components to LOS'
- list gpsLOS_ref = unitVec[0]∗Ve[idxRef]+unitVec[1]∗Vn[idxRef]+unitVec[2]∗Vu[idxRef]

- tuple Sr = ((unitVec[0]**2)*Se[idxRef]**2+(unitVec[1]**2)*Sn[idxRef]**2+(unitVec[2]**2)*Su[idx↩
Ref]**2)**0.5
- h5coh = h5py.File(coherenceFile)
- kh5coh = h5coh.keys()
- dset = h5coh[kh5coh[0]].get(kh5coh[0])
- Coh = dset[0:dset.shape[0],0:dset.shape[1]]
- list InSAR = [ ]
- list GPS = [ ]
- list InSAR1 = [ ]
- list GPS1 = [ ]
- list InSAR2 = [ ]
- list GPS2 = [ ]
- list coherence = [ ]
- list GPSx = [ ]
- list GPSy = [ ]
- list GPSx1 = [ ]
- list GPSy1 = [ ]
- list GPSx2 = [ ]
- list GPSy2 = [ ]
- list GPS_station = [ ]
- list GPS_std = [ ]
- idx = Stations.index(st)
- list gpsLOS = unitVec[0]*Ve[idx]+unitVec[1]*Vn[idx]+unitVec[2]*Vu[idx]
- tuple Sg = ((unitVec[0]**2)*Se[idx]**2+(unitVec[1]**2)*Sn[idx]**2+(unitVec[2]**2)*Su[idx]**2)**0.5
- tuple S = (Sg**2+Sr**2)**0.5
- IDY
- IDX
- insar_velocity = -insarData[IDY][IDX]
- string InSAR_GPS_Copmarison = 'yes'
- string NoInSAR = 'yes'
- lt = len(InSAR)
- SAD = np.sum(np.abs(InSAR-GPS),0)/lt
- C1 = np.zeros([2,len(InSAR)])
- Cor = np.corrcoef(C1)[0][1]
- minV = np.min([InSAR,GPS])
- maxV = np.max([InSAR,GPS])
- fig = plt.figure()
- ax = fig.add_subplot(111)
- yerr
- xerr
- fmt
- ms
- fontsize
- xy
- xytext
- color
- majorLocator = MultipleLocator(5)
- minorLocator = MultipleLocator(1)
- which
- length
- width
- string figName = 'InSARvsGPS_errorbar.png'

## 19.43 /Users/jeromezhang/Documents/development/python/PySAR/pysar/insarmaps_query.py File Reference

### Classes

- class BasicHTTP

### Namespaces

- pysar.insarmaps_query

### Functions

- def buildURL (args)
- def build_parser ()
- def main ()

## 19.44 /Users/jeromezhang/Documents/development/python/PySAR/pysar/l1.py File Reference

### Namespaces

- pysar.l1

### Functions

- def l1mosek (P, q)
- def l1mosek2 (P, q)
- def l1 (P, q)
- def l1blas (P, q)

### Variables

- bool __MOSEK = True
- task = env.Task(0,0)
- x = zeros(n, float)

## 19.45 /Users/jeromezhang/Documents/development/python/PySAR/pysar/load_data.py File Reference

### Namespaces

- pysar.load_data

### Functions

- def auto_path_miami (inps, template_dict={})

    *Sub Functions ####################################.*
- def mode (thelist)

    *Find Mode (most common) item in the list #############.*
- def check_file_size (fileList, mode_width=None, mode_length=None)
- def check_existed_hdf5_file (roipacFileList, hdf5File)
- def load_roipac2multi_group_h5 (fileType, fileList, hdf5File='unwrapIfgram.h5', pysar_meta_dict=None)
- def roipac_nonzero_mask (unwFileList, maskFile='Mask.h5')
- def copy_roipac_file (targetFile, destDir)
- def cmdLineParse ()
- def main (argv)

    *Main Function ################################.*

### Variables

- string EXAMPLE

    *Usage ################################.*
- string TEMPLATE

## 19.46 /Users/jeromezhang/Documents/development/python/PySAR/pysar/load_dem.py File Reference

### Namespaces

- pysar.load_dem

### Variables

- demFile = sys.argv[1]
- ext = os.path.splitext(demFile)[1]
- amp
- dem
- demRsc
- outName
- h5 = h5py.File(outName,'w')
- group = h5.create_group('dem')
- dset = group.create_dataset('dem', data=dem, compression='gzip')

## 19.47 /Users/jeromezhang/Documents/development/python/PySAR/pysar/lod.py File Reference

### Namespaces

- pysar.lod

### Functions

- def correct_lod_file (File, outFile=None)
- def usage ()
- def main (argv)

## 19.48 /Users/jeromezhang/Documents/development/python/PySAR/pysar/look_angle.py File Reference

### Namespaces

- pysar.look_angle

### Functions

- def usage ()
- def main (argv)

## 19.49 /Users/jeromezhang/Documents/development/python/PySAR/pysar/los2enu.py File Reference

### Namespaces

- pysar.los2enu

### Functions

- def usage ()
- def main (argv)

## 19.50 /Users/jeromezhang/Documents/development/python/PySAR/pysar/mask.py File Reference

### Namespaces

- pysar.mask

### Functions

- def mask_matrix (data_mat, mask_mat)
- def update_mask (mask, inps_dict=None)
- def mask_file (File, maskFile, outFile=None, inps_dict=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 19.51 /Users/jeromezhang/Documents/development/python/PySAR/pysar/match.py File Reference

### Namespaces

- pysar.match

### Functions

- def corners (atr)
- def nearest (x, X)
- def manual_offset_estimate (matrix1, matrix2)
- def match_two_files (File1, File2, outName=None, manual_match=False, disp_fig=False)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 19.52 /Users/jeromezhang/Documents/development/python/PySAR/pysar/mean_spatial.py File Reference

### Namespaces

- pysar.mean_spatial

### Functions

- def circle_index (atr, circle_par)
- def Usage ()

  *Usage ####################################.*
- def main (argv)

  *Main Function #################################.*

## 19.53 /Users/jeromezhang/Documents/development/python/PySAR/pysar/modify_↩ network.py File Reference

### Namespaces

- pysar.modify_network

### Functions

- def nearest_neighbor (x, y, x_array, y_array)

  *Sub Function ################################.*
- def manual_select_pairs_to_remove (File)
- def update_inps_with_template (inps, template_file)
- def modify_file_date12_list (File, date12_to_rmv, outFile=None)
- def cmdLineParse ()
- def main (argv)

  *Main Function ################################.*

### Variables

- string EXAMPLE

  *Usage ################################.*
- string TEMPLATE

## 19.54 /Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/delayTimeseries.py File Reference

### Classes

- class timeseries

### Namespaces

- delayTimeseries

### Functions

- def write_to_h5 (dataset, outName, groupName, h5withAttributes)
- def nearest_valid (xr, yr, data_flat, rows, cols)

## 19.55 /Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/dloadUtil.py File Reference

### Namespaces

- dloadUtil

### Functions

- def download_modis (inps)
- def download_atmosphereModel (inps)
- def daterange (start_date, end_date)
- def get_date (f)
- def pwv2zwd (pwv)
- def zwd2swd (zwd, theta)
- def read_modis (file)

## 19.56 /Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/get_modis_v3.py File Reference

### Namespaces

- get_modis_v3

### Functions

- def usage ()
- def main ()

### Variables

- out = sys.stdout
- start_time_main = time.time()
- time_elapsed = time.time() - start_time_main

## 19.57 /Users/jeromezhang/Documents/development/python/PySAR/pysar/modis/troposphere_uncertainty.py File Reference

### Namespaces

- troposphere_uncertainty

### Functions

- def createParser ()
- def cmdLineParse (iargs=None)
- def velocity_uncertainty_vs_distance (inps)
- def statistics (inps)
- def estimate_seasonal (inps)
- def velocity_uncertainty (realtive_std_file, inps)
- def download (inps)
- def main (iargs=None)

## 19.58 /Users/jeromezhang/Documents/development/python/PySAR/pysar/multi_transect.py File Reference

### Namespaces

- pysar.multi_transect

## Functions

- def usage ()
- def dms2d (Coord)
- def gps_to_LOS (Ve, Vn, theta, heading)
- def check_st_in_box (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def check_st_in_box2 (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def line (x0, y0, x1, y1)
- def dist_point_from_line (m, c, x, y, dx, dy)
- def get_intersect (m, c, x, y)
- def readGPSfile (gpsFile, gps_source)
- def redGPSfile (gpsFile)
- def redGPSfile_cmm4 (gpsFile)
- def nearest (x, tbase, xstep)
- def find_row_column (Lon, Lat, lon, lat, lon_step, lat_step)
- def get_lat_lon (h5file)
- def nanmean (data, args)
- def nanstd (data, args)
- def get_transect (z, x0, y0, x1, y1)
- def get_start_end_point (Xf0, Yf0, Xf1, Yf1, L, dx, dy)
- def point_with_distance_from_line (Xf0, Yf0, Xf1, Yf1, L)
- def point_on_line_with_distance_from_beginning (Xf0, Yf0, Xf1, Yf1, L)
- def read_fault_coords (Fault_coord_file, Dp)
- def main (argv)
- def onclick (event)

## Variables

- lat
- lon
- lat_step
- lon_step
- lat_all
- lon_all
- Fault_lon
- Fault_lat
- int Num_profiles = len(Fault_lon)-1
- list FaultCoords = [Fault_lat[Np],Fault_lon[Np],Fault_lat[Np+1],Fault_lon[Np+1]]
- list Lat0 = FaultCoords[1]
- list Lat1 = FaultCoords[3]
- Length
- Width
- Yf0
- Xf0
- Yf1
- Xf1
- y0 = yc[1]
- x0 = xc[1]
- y1
- x1
- fig = plt.figure()
- ax = fig.add_subplot(111)
- list xc = [ ]
- list yc = [ ]

- cid = fig.canvas.mpl_connect('button_press_event', onclick)
- length = int(np.hypot(x1-x0, y1-y0))

  *try: mf=float(Yf1-Yf0)/float((Xf1-Xf0)) # slope of the fault line cf=float(Yf0-mf∗Xf0) # intercept of the fault line df0=dist←_point_from_line(mf,cf,x0,y0,1,1) #distance of the profile start point from the Fault line df1=dist_point_from_←line(mf,cf,x1,y1,1,1) #distance of the profile end point from the Fault line*
- x
- y
- zi = z[y.astype(np.int), x.astype(np.int)]
- lat_transect = lat_all[y.astype(np.int), x.astype(np.int)]
- lon_transect = lon_all[y.astype(np.int), x.astype(np.int)]
- float dx = float(h5file[k[0]].attrs['X_STEP'])∗6375000.0∗np.pi/180.0
- float dy = float(h5file[k[0]].attrs['Y_STEP'])∗6375000.0∗np.pi/180.0
- tuple DX = (x-x0)∗dx
- tuple DY = (y-y0)∗dy
- D = np.hypot(DX, DY)
- mf
- cf
- def df0_km = dist_point_from_line(mf,cf,x0,y0,dx,dy)
- transect = np.zeros([len(D),ntrans])
- list XX0 = [ ]
- list YY0 = [ ]
- m = float(y1-y0)/float((x1-x0))
- c = float(y0-m∗x0)
- float m1 = -1.0/m
- float dp = 1.0
- float X0 = i∗dp/np.sqrt(1+m1∗∗2)+x0
- float Y0 = m1∗(X0-x0)+y0
- float X1 = i∗dp/np.sqrt(1+m1∗∗2)+x1
- float Y1 = m1∗(X1-x1)+y1
- transect_lat = np.zeros([len(D),ntrans])
- transect_lon = np.zeros([len(D),ntrans])
- m_prof_edge
- c_prof_edge
- string gpsFile = 'Nogps'
- insarData = z
- fileName
- fileExtension
- Stations
- Lat
- Lon
- Ve
- Se
- Vn
- Sn
- idxRef = Stations.index(refStation)
- IDYref
- IDXref
- stationsList = Stations
- h5file_theta = h5py.File(incidence_file,'r')
- dset = h5file_theta['mask'].get('mask')
- theta = dset[0:dset.shape[0],0:dset.shape[1]]
- float heading = 193.0∗np.pi/180.0
- list unitVec = [np.cos(heading)∗np.sin(theta),-np.sin(theta)∗np.sin(heading),0]
- def gpsLOS_ref = gps_to_LOS(Ve[idxRef],Vn[idxRef],theta[IDYref,IDXref],heading)
- list GPS = [ ]

- list GPS_station = [ ]
- list GPSx = [ ]
- list GPSy = [ ]
- list GPS_lat = [ ]
- list GPS_lon = [ ]
- idx = Stations.index(st)
- IDY
- IDX
- def gpsLOS = gps_to_LOS(Ve[idx],Vn[idx],theta[IDY,IDX],heading)
- string NoInSAR = 'yes'
- list DistGPS = [ ]
- list GPS_in_bound = [ ]
- list GPS_in_bound_st = [ ]
- list GPSxx = [ ]
- list GPSyy = [ ]
- list gx = GPSx[i]
- list gy = GPSy[i]
- string check_result = 'True'
- def check_result2 = check_st_in_box2(gx,gy,x0,y0,x1,y1,X0,Y0,X1,Y1)
- def dg = dist_point_from_line(m,c,gx,gy,1,1)
- axes
- nrows
- ms

    *ax.fill_between(D/1000.0, (avgInSAR-stdInSAR)∗1000, (avgInSAR+stdInSAR)∗1000,where=(avgInSAR+stdInS↩ AR)∗1000>=(avgInSAR-stdInSAR)∗1000,alpha=1, facecolor='Red')*
- avgInSAR = np.array(nanmean(transect,axis=1))
- stdInSAR = np.array(nanstd(transect,axis=1))
- fig2
- axes2
- string FaultLine = 'None'
- string figName = 'transect_area_'+str(Np)+'.png'

    *Temporary To plot DEM try: majorLocator = MultipleLocator(5) ax.yaxis.set_major_locator(majorLocator) minor↩ Locator = MultipleLocator(1) ax.yaxis.set_minor_locator(minorLocator)*
- mfc
- linewidth
- string matFile = 'transect'+str(Np)+'.mat'
- dictionary dataset = {}
- color

    *ax.plot(D/1000.0, avgInSAR∗1000, 'r-')*
- alpha
- fontsize
- int lbound = np.nanmin(transect)∗1000

    *lower and higher bounds for diplaying the profile*
- int hbound = np.nanmax(transect)∗1000
- string ylim = 'no'
- string xlim = 'no'

## 19.59 /Users/jeromezhang/Documents/development/python/PySAR/pysar/multilook.py File Reference

### Namespaces

- pysar.multilook

### Functions

- def multilook_matrix (matrix, lks_y, lks_x)

  *Sub Functions ##############################################.*
- def multilook_attribute (atr_dict, lks_y, lks_x)
- def multilook_file (infile, lks_y, lks_x, outfile=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 19.60 /Users/jeromezhang/Documents/development/python/PySAR/pysar/plot/plot_tropcor_phase_elevation.py File Reference

### Namespaces

- plot_tropcor_phase_elevation

### Variables

- string workDir = '/scratch/projects/insarlab/yzhang1/KyushuT80F245_246JersD/TSSAR'
- string demFile = 'radar_4rlks.hgt'
- string timeseriesFile = 'timeseries_demCor.h5'
- string timeseriesFile2 = 'timeseries_demCor_tropHgt.h5'
- string maskFile = 'Mask_tempCoh_dis.h5'
- string tropHgtFile = 'tropHgt.h5'
- string ecmwfFile = 'ECMWF.h5'
- string epoch = '19980926'
- dem
- dem_atr
- data
- atr
- data2
- atr2
- tropHgt
- atr3
- ecmwf
- atr4
- mask
- msk_atr
- ndx = np.nan
- list dataList = [data,data2,-tropHgt,-ecmwf]
- fig
- axes
- nrows
- ncols
- sharex
- True
- sharey
- figsize
- int i = 0
- ms
- bbox_inches
- dpi

## 19.61 /Users/jeromezhang/Documents/development/python/PySAR/pysar/plot_atmDrop.py File Reference

### Namespaces

- pysar.plot_atmDrop

### Variables

- list projectList = ['AlosAT422','AlosAT423','AlosDT72','AlosDT73']
- string projectDir = '/Users/jeromezhang/Documents/insarlab/Kyushu/Volcanoes/Kuju'
- numProject = len(projectList)
- fig = plt.figure(figsize=(12,12))
- ax1 = fig.add_subplot(211)
- ax2 = fig.add_subplot(212)
- offset = range(1,numProject+1)
- fl = open(projectDir+'/'+projectList[i]+'/spatialMean_sum_Seeded_ts.txt','r')

  *Read txt file.*
- list lines = [ ]
- int lineNum = 0
- list dateList6 = [ ]
- list meanList = [ ]
- list pixList = [ ]
- line_s = line.split()
- dateList = ptime.yyyymmdd(dateList6)
- dates = np.array(dates)
- datevector
- idxMean = max(enumerate(meanList),key=lambda x: x[1])[0]
- list idxPix = pixList $<$ 0.7
- sc1 = ax1.scatter(dates, np.tile(offset[i],lineNum), c=meanList, s=22$**$2, alpha=0.3, vmin=0.0, vmax=1.0)

  *Plot.*
- c
- s
- alpha
- vmin
- vmax
- sc2 = ax2.scatter(dates, np.tile(offset[i],lineNum), c=pixList, s=22$**$2, alpha=0.3, vmin=0.0, vmax=1.0)
- fontsize
- cbar = fig.colorbar(sc2)
- bbox_inches
- transparent

## 19.62 /Users/jeromezhang/Documents/development/python/PySAR/pysar/plot_network.py File Reference

### Namespaces

- pysar.plot_network

### Functions

- def cmdLineParse ()
- def main (argv)

    *Main Function ################################.*

### Variables

- string BL_LIST
- string DATE12_LIST
- string EXAMPLE

## 19.63 /Users/jeromezhang/Documents/development/python/PySAR/pysar/pysar2insarmaps.py File Reference

### Namespaces

- pysar.pysar2insarmaps

### Functions

- def project_name_from_path (path)
- def sorted_ls (path)
- def rev_sorted_ls (path)
- def get_H5_filename (path)
- def build_parser ()
- def main ()

## 19.64 /Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp.py File Reference

### Namespaces

- pysar.pysarApp

### Functions

- def check_isfile (File)
- def check_subset_file (File, inps_dict, outFile=None, overwrite=False)
- def check_geocode_file (geomapFile, File, outFile=None)
- def subset_dataset (inps, geo_box4geo, pix_box4rdr)
- def create_subset_dataset (inps, pix_box=None, geo_box=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string LOGO
- string TEMPLATE
- string EXAMPLE
- string UM_FILE_STRUCT

## 19.65 /Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp_cmd.py File Reference

### Namespaces

- pysar.pysarApp_cmd

### Functions

- def check_isfile (File)
- def check_subset_file (File, inps_dict, outFile=None, overwrite=False)
- def check_geocode_file (geomapFile, File, outFile=None)
- def subset_dataset (inps, geo_box4geo, pix_box4rdr)
- def create_subset_dataset (inps, pix_box=None, geo_box=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string LOGO
- string TEMPLATE
- string EXAMPLE
- string UM_FILE_STRUCT

## 19.66 /Users/jeromezhang/Documents/development/python/PySAR/pysar/pysarApp_orig.py File Reference

### Namespaces

- pysar.pysarApp_orig

### Functions

- def find_filename (template, option, workDir='.')

  *Sub Functions ######################################## find accurate file name for template input #############.*
- def check_subset (inName, subset, option='yx', workDir='.')

  *update the subset of input file ########################*
- def check_geocode (inName, geomapFile, workDir='.')

  *update geocoding of input file ########################*
- def check_mask (inName, maskFile, workDir='.')

  *update masking of input file ##########################*
- def usage ()

  *Usage Function ##########################.*
- def cmdLineParse ()
- def main (argv)

  *Main Function #####################################.*

## 19.67 /Users/jeromezhang/Documents/development/python/PySAR/pysar/quality_map.py File Reference

### Namespaces

- pysar.quality_map

### Functions

- def usage ()
- def main (argv)

## 19.68 /Users/jeromezhang/Documents/development/python/PySAR/pysar/reconstruct_igrams.py File Reference

### Namespaces

- pysar.reconstruct_igrams

### Functions

- def reconstruct_igrams_from_timeseries (h5timeseries, h5igrams)
- def usage ()
- def main (argv)

## 19.69 /Users/jeromezhang/Documents/development/python/PySAR/pysar/reference_epoch.py File Reference

### Namespaces

- pysar.reference_epoch

### Functions

- def yymmdd2yyyymmdd (date)
- def usage ()
- def main (argv)

## 19.70 /Users/jeromezhang/Documents/development/python/PySAR/pysar/remove_dates.py File Reference

### Namespaces

- pysar.remove_dates

### Functions

- def usage ()
- def main (argv)

## 19.71 /Users/jeromezhang/Documents/development/python/PySAR/pysar/remove_plane.py File Reference

### Namespaces

- pysar.remove_plane

### Functions

- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 19.72 /Users/jeromezhang/Documents/development/python/PySAR/pysar/rewrap.py File Reference

### Namespaces

- pysar.rewrap

### Functions

- def usage ()
- def rewrap (unw)
- def main (argv)

## 19.73 /Users/jeromezhang/Documents/development/python/PySAR/pysar/save_gmt.py File Reference

### Namespaces

- pysar.save_gmt

### Functions

- def get_geo_lat_lon (atr)
- def usage ()
- def main (argv)

  *Main Function ####################################.*

## 19.74 /Users/jeromezhang/Documents/development/python/PySAR/pysar/save_kml.py File Reference

### Namespaces

- pysar.save_kml

### Functions

- def rewrap (unw)
- def usage ()
- def main (argv)

## 19.75 /Users/jeromezhang/Documents/development/python/PySAR/pysar/save_unavco.py File Reference

### Namespaces

- pysar.save_unavco

### Functions

- def metadata_pysar2unavco (pysar_meta_dict, dateList)
- def cmdLineParse ()
- def main (argv)

### Variables

- INT_ZERO = np.int16(0)
- FLOAT_ZERO = np.float32(0.0)
- CPX_ZERO = np.complex64(0.0)
- string EXAMPLE

## 19.76 /Users/jeromezhang/Documents/development/python/PySAR/pysar/save_unw.py File Reference

### Namespaces

- pysar.save_unw

### Functions

- def usage ()
- def main (argv)

## 19.77 /Users/jeromezhang/Documents/development/python/PySAR/pysar/seed_data.py File Reference

### Namespaces

- pysar.seed_data

### Functions

- def nearest (x, tbase, xstep)

    *Sub Functions ################################################.*
- def seed_file_reference_value (File, outName, refList, ref_y='', ref_x='')
- def seed_file_inps (File, inps=None, outFile=None)
- def seed_attributes (atr_in, x, y)
- def random_select_reference_yx (data_mat)
- def manual_select_reference_yx (stack, inps)
- def select_max_coherence_yx (corFile, mask=None)
- def print_warning (next_method)
- def read_seed_template2inps (template_file, inps=None)
- def read_seed_reference2inps (reference_file, inps=None)
- def usage ()

    *Usage ################################################.*
- def cmdLineParse ()
- def main (argv)

    *Main Function ##########################################.*

## 19.78 /Users/jeromezhang/Documents/development/python/PySAR/pysar/simulation.py File Reference

### Namespaces

- pysar.simulation

### Functions

- def usage ()
- def main (argv)

## 19.79 /Users/jeromezhang/Documents/development/python/PySAR/pysar/spatial_↩ average.py File Reference

### Namespaces

- pysar.spatial_average

### Functions

- def cmdLineParse ()
- def main (argv)

    *Main Function ################################.*

### Variables

- string EXAMPLE

  *Usage ####################################.*

## 19.80 /Users/jeromezhang/Documents/development/python/PySAR/pysar/subset.py File Reference

### Namespaces

- pysar.subset

### Functions

- def coord_geo2radar (geoCoord, atr, coordType)

  *Example: 300 = coord_geo2radar(32.104990, atr,'lat') [1000,1500] = coord_geo2radar([130.5,131.4],atr,'lon')*
- def coord_radar2geo (radarCoord, atr, coordType)

  *Inputs: radarCoord : coordinate (list) in row/col in int atr : dictionary of file attributes coordType : coordinate type: row, col, y, x.*
- def check_box_within_data_coverage (pixel_box, atr_dict)
- def subset_attribute (atr_dict, subset_box)
- def get_coverage_box (atr)
- def read_subset_template2box (templateFile)
- def subset_box2inps (inps, pix_box, geo_box)
- def get_box_overlap_index (box1, box2)
- def subset_input_dict2box (subset_dict, meta_dict)
- def box_pixel2geo (pixel_box, meta_dict)
- def box_geo2pixel (geo_box, meta_dict)
- def subset_file (File, subset_dict, outFile=None)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE

## 19.81 /Users/jeromezhang/Documents/development/python/PySAR/pysar/sum_epochs.py File Reference

### Namespaces

- pysar.sum_epochs

### Functions

- def usage ()
- def main (argv)

## 19.82 /Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_↩ average.py File Reference

### Namespaces

- pysar.temporal_average

### Functions

- def usage ()

  *Usage ####################################.*
- def main (argv)

  *Main Function ################################.*

## 19.83 /Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_↩ coherence.py File Reference

### Namespaces

- pysar.temporal_coherence

### Functions

- def date_list (h5file)
- def design_matrix (h5file)
- def usage ()
- def main (argv)

## 19.84 /Users/jeromezhang/Documents/development/python/PySAR/pysar/temporal_↩ derivative.py File Reference

### Namespaces

- pysar.temporal_derivative

### Functions

- def usage ()
- def main (argv)

## 19.85 /Users/jeromezhang/Documents/development/python/PySAR/pysar/timeseries2velocity.py File Reference

### Namespaces

- pysar.timeseries2velocity

### Functions

- def yyyymmdd2years (date)
- def update_inps_from_template (inps, template_file)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE
- string TEMPLATE
- string DROP_DATE_TXT

## 19.86 /Users/jeromezhang/Documents/development/python/PySAR/pysar/transect.py File Reference

### Namespaces

- pysar.transect

### Functions

- def get_scale_from_disp_unit (disp_unit, data_unit)
- def read_lonlat_file (lonlat_file)
- def manual_select_start_end_point (File)
- def transect_yx (z, atr, start_yx, end_yx, interpolation='nearest')
- def transect_lalo (z, atr, start_lalo, end_lalo, interpolation='nearest')
- def transect_list (fileList, inps)
- def cmdLineParse ()
- def main (argv)

  *Main ###################################.*

### Variables

- string EXAMPLE

## 19.87 /Users/jeromezhang/Documents/development/python/PySAR/pysar/transect_legacy.py File Reference

### Namespaces

- pysar.transect_legacy

### Functions

- def dms2d (Coord)
- def gps_to_LOS (Ve, Vn, theta, heading)
- def check_st_in_box (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def check_st_in_box2 (x, y, x0, y0, x1, y1, X0, Y0, X1, Y1)
- def line (x0, y0, x1, y1)
- def dist_point_from_line (m, c, x, y, dx, dy)
- def get_intersect (m, c, x, y)
- def readGPSfile (gpsFile, gps_source)
- def redGPSfile (gpsFile)
- def redGPSfile_cmm4 (gpsFile)
- def nearest (x, tbase, xstep)
- def find_row_column (Lon, Lat, lon, lat, lon_step, lat_step)
- def get_lat_lon (atr)
- def nanmean (data, args)
- def nanstd (data, args)
- def get_transect (z, x0, y0, x1, y1, interpolation='nearest')

  *Option: interpolation : sampling/interpolation method, including: 'nearest' - nearest neighbour, by default 'cubic' - cubic interpolation 'bilinear' - bilinear interpolation.*
- def Usage ()
- def main (argv)
- def onclick (event)

### Variables

- fig = plt.figure()
- ax = fig.add_subplot(111)
- list xc = [ ]
- list yc = [ ]
- cid = fig.canvas.mpl_connect('button_press_event', onclick)
- list x0 = xc[1]
- list y0 = yc[1]
- mf = float(Yf1-Yf0)/float((Xf1-Xf0))
- cf = float(Yf0-mf∗Xf0)
- def df0 = dist_point_from_line(mf,cf,x0,y0,1,1)
- def df1 = dist_point_from_line(mf,cf,x1,y1,1,1)
- int mp = -1./mf
- x1 = int((df0+df1)/np.sqrt(1+mp∗∗2)+x0)
- y1 = int(mp∗(x1-x0)+y0)
- string Info_aboutFault = 'No'
- length = int(np.hypot(x1-x0, y1-y0))
- x
- y

- zi = z[y.astype(np.int), x.astype(np.int)]
- lat_transect = lat_all[y.astype(np.int), x.astype(np.int)]
- lon_transect = lon_all[y.astype(np.int), x.astype(np.int)]
- int earth_radius = 6371e3;
- float dx = float(atr['X_STEP'])∗np.pi/180.0∗earth_radius∗np.sin(np.mean(lat)∗np.pi/180)
- float dy = float(atr['Y_STEP'])∗np.pi/180.0∗earth_radius
- tuple DX = (x-x0)∗dx
- tuple DY = (y-y0)∗dy
- D = np.hypot(DX, DY)
- df0_km
- transect = np.zeros([len(D),ntrans])
- list XX0 = [ ]
- list YY0 = [ ]
- m = float(y1-y0)/float((x1-x0))
- c = float(y0-m∗x0)
- float m1 = -1.0/m
- list X0 = i∗dp/np.sqrt(1+m1∗∗2)+x0
- float Y0 = m1∗(X0-x0)+y0
- X1 = i∗dp/np.sqrt(1+m1∗∗2)+x1
- float Y1 = m1∗(X1-x1)+y1
- transect_lat = np.zeros([len(D),ntrans])
- transect_lon = np.zeros([len(D),ntrans])
- m_prof_edge
- c_prof_edge
- gpsFile
- insarData = z
- fileName
- fileExtension
- Stations
- Lat
- Lon
- Ve
- Se
- Vn
- Sn
- idxRef = Stations.index(refStation)
- Length
- Width
- lat
- lon
- lat_step
- lon_step
- lat_all
- lon_all
- IDYref
- IDXref
- stationsList = Stations
- h5file_theta = h5py.File(incidence_file,'r')
- dset = h5file_theta['mask'].get('mask')
- theta = dset[0:dset.shape[0],0:dset.shape[1]]
- float heading = 193.0∗np.pi/180.0
- list unitVec = [np.cos(heading)∗np.sin(theta),-np.sin(theta)∗np.sin(heading),0]
- def gpsLOS_ref = gps_to_LOS(Ve[idxRef],Vn[idxRef],theta[IDYref,IDXref],heading)
- list GPS = [ ]
- list GPS_station = [ ]

- list GPSx = [ ]
- list GPSy = [ ]
- list GPS_lat = [ ]
- list GPS_lon = [ ]
- idx = Stations.index(st)
- IDY
- IDX
- def gpsLOS = gps_to_LOS(Ve[idx],Vn[idx],theta[IDY,IDX],heading)
- string NoInSAR = 'yes'
- list DistGPS = [ ]
- list GPS_in_bound = [ ]
- list GPS_in_bound_st = [ ]
- list GPSxx = [ ]
- list GPSyy = [ ]
- list gx = GPSx[i]
- list gy = GPSy[i]
- string check_result = 'True'
- def check_result2 = check_st_in_box2(gx,gy,x0,y0,x1,y1,X0,Y0,X1,Y1)
- def dg = dist_point_from_line(m,c,gx,gy,1,1)
- axes
- nrows
- ms

  *ax.fill_between(D/1000.0, (avgInSAR-stdInSAR)∗1000, (avgInSAR+stdInSAR)∗1000,where=(avgInSAR+stdInS↩ AR)∗1000>=(avgInSAR-stdInSAR)∗1000,alpha=1, facecolor='Red')*
- avgInSAR = np.array(nanmean(transect,axis=1))
- stdInSAR = np.array(nanstd(transect,axis=1))
- fig2
- axes2
- string FaultLine = 'None'
- string figName = 'transect_area.png'

  *Temporary To plot DEM try: majorLocator = MultipleLocator(5) ax.yaxis.set_major_locator(majorLocator) minor↩ Locator = MultipleLocator(1) ax.yaxis.set_minor_locator(minorLocator)*
- mfc
- linewidth
- string matFile = 'transect.mat'
- dictionary dataset = {}
- color

  *ax.plot(D/1000.0, avgInSAR∗1000, 'r-')*
- alpha
- fontsize
- int lbound = np.nanmin(transect)∗1000

  *lower and higher bounds for diplaying the profile*
- int hbound = np.nanmax(transect)∗1000
- string fault_loc = 'None'
- string ylim = 'no'

## 19.88 /Users/jeromezhang/Documents/development/python/PySAR/pysar/tropcor_↩ phase_elevation.py File Reference

### Namespaces

- pysar.tropcor_phase_elevation

### Functions

- def usage ()
- def main (argv)

## 19.89 /Users/jeromezhang/Documents/development/python/PySAR/pysar/tropcor_pyaps.py File Reference

### Namespaces

- pysar.tropcor_pyaps

### Functions

- def closest_weather_product_time (sar_acquisition_time, grib_source='ECMWF')
- def get_delay (grib_file, atr, inps_dict)
- def cmdLineParse ()
- def main (argv)

### Variables

- string EXAMPLE
- string REFERENCE
- string TEMPLATE

## 19.90 /Users/jeromezhang/Documents/development/python/PySAR/pysar/tsview_mli.py File Reference

### Namespaces

- pysar.tsview_mli

### Functions

- def transect_yx (z, atr, start_yx, end_yx, interpolation='nearest')

  *Option: interpolation : sampling/interpolation method, including: 'nearest' - nearest neighbour, by default 'cubic' - cubic interpolation 'bilinear' - bilinear interpolation.*
- def transect_lalo (z, atr, start_lalo, end_lalo, interpolation='nearest')
- def transect_list (fileList, start_coord, end_coord, coord_type='radar', interpolation='nearest')
- def usage ()

  *Usage ##################################.*
- def main (argv)

  *Main ##################################.*

## 19.91 /Users/jeromezhang/Documents/development/python/PySAR/pysar/tsviewer.py File Reference

### Namespaces

- pysar.tsviewer

### Functions

- def check_yx (xsub, ysub, radius, ax, rectColor='black')
  *Sub Functions ####################################.*
- def read_dis_xy (xsub, ysub, dateList, h5file, unit='cm')
- def read_dis_lalo (lat, lon, dateList, timeseriesFile, radius=0, unit='cm')
- def update_lim (disp_min, disp_max, data_mean, data_std)
- def usage ()
  *Usage ##################################.*
- def main (argv)
  *Main Function ####################################.*

## 19.92 /Users/jeromezhang/Documents/development/python/PySAR/pysar/unavco2insarmaps.py File Reference

### Namespaces

- pysar.unavco2insarmaps

### Functions

- def get_date (date_string)
- def get_decimal_date (d)
- def convert_data (attributes, decimal_dates, timeseries_datasets, dataset_keys, json_path, folder_name, region_file_name)
- def make_json_file (chunk_num, points, dataset_keys, json_path, folder_name)
- def build_parser ()
- def main ()

### Variables

- string dbUsername = "INSERT"
- string dbPassword = "INSERT"
- string dbHost = "INSERT"

## 19.93 /Users/jeromezhang/Documents/development/python/PySAR/pysar/unwrap_↩ error.py File Reference

### Namespaces

- pysar.unwrap_error

### Functions

- def phase_bonding (data, mask, x, y)
- def usage ()
- def main (argv)

## 19.94 /Users/jeromezhang/Documents/development/python/PySAR/pysar/view.py File Reference

### Classes

- class Basemap2
  *Class ##################################################.*

### Namespaces

- pysar.view

### Functions

- def add_inner_title (ax, title, loc, size=None, kwargs)
- def auto_flip_direction (atr_dict)
- def auto_figure_title (meta_dict, inps)
- def auto_row_col_num (subplot_num, data_shape, fig_size, fig_num=1)
- def check_colormap_input (atr_dict, colormap=None)
- def check_multilook_input (pixel_box, row_num, col_num)
- def get_epoch_full_list_from_input (all_epoch_list, epoch_input_list=[ ], epoch_num_input_list=[ ])
- def plot_dem_lalo (bmap, dem, box, inps_dict)
- def plot_dem_yx (ax, dem, inps_dict)
- def round_to_1 (x)
- def scale_data2disp_unit (matrix, atr_dict, disp_unit)
- def update_plot_inps_with_display_setting_file (inps, disp_set_file)
- def update_plot_inps_with_meta_dict (inps, meta_dict)
- def update_matrix_with_plot_inps (data, meta_dict, inps)
- def plot_matrix (ax, data, meta_dict, inps=None)
- def cmdLineParse (argv)
- def main (argv)
  *Main Function ########################################.*

### Variables

- string EXAMPLE
- string PLOT_TEMPLATE

## 19.95 /Users/jeromezhang/Documents/development/python/PySAR/pysar/view_legacy.py File Reference

### Namespaces

- pysar.view_legacy

### Functions

- def add_inner_title (ax, title, loc, size=None, kwargs)

    *Sub Function ########################################.*
- def rewrap (data, atr)
- def unit_and_scale (data_unit, display_unit)
- def unit_type (unit_in)
- def orbit_direction (atr)
- def auto_flip_check (atr_dict)
- def plot_dem_lalo (bmap, dem, geo_box, demShade='yes', demContour='no', contour_step=200.0, contour↵_sigma=3.0)

    *Examples: bmap = plot_dem_lalo(bmap,dem,geo_box,'no','yes')*
- def plot_dem_yx (ax, dem, demShade='yes', demContour='no', contour_step=200.0, contour_sigma=3.0)

    *Examples: ax = plot_dem_yx(ax,dem,'no','yes')*
- def usage ()

    *Usage #######################.*
- def main (argv)

    *Main Function ########################################.*

## 19.96 /Users/jeromezhang/Documents/development/python/PySAR/README.md File Reference