

# Supplementary material 2: Phase Jump Detector

Viotto, Sofia<sup>1\*</sup> and Bookhagen, Bodo<sup>1</sup>

\*Corresponding author

<sup>1</sup>Institute of Geosciences, University of Potsdam

E-mails: viotto1@uni-potsdam.de

## 1 Introduction

The differential interferometric synthetic aperture radar (DInSAR) technique utilizes the phase difference between two SAR scenes acquired at different times. The differential phase (known as an interferogram) provides valuable information about the surface, enabling the detection and measurement of ground motion. Accurately retrieving the magnitude of the movement depends on several steps throughout the DInSAR processing chain. This description focuses on scene alignment, the first and crucial step.

The purpose of scene alignment is to match common objects in an image pair while ensuring that these objects are correctly positioned on the Earth's surface. Typically, alignment is performed throughout coregistration and using precise knowledge of orbital parameters, together with topographic information (Sansosti et al., 2006). However, the accuracy requirements for this procedure vary depending on the SAR data acquisition mode, and finer alignment methods may be needed. Thus, the accuracy demands for the Terrain Observation with Progressive Scans (TOPS) mode are significantly higher, compared to the Stripmap mode (e.g., Prats-Iraola et al., 2012; Yagüe-Martínez et al., 2016). The free data with global coverage provided by the Sentinel-1 SAR mission, is mostly acquired in Interferometric Wide-Swath (TOPS) mode (Torres et al., 2012).

In TOPS mode, the swath coverage is divided into subbands, which are captured in bursts (De Zan and Monti Guarnieri, 2006). To generate a continuous DInSAR product, TOPS

bursts must be accurately matched and aligned in both the azimuth (along-track) and range (across-track) directions. Azimuth alignment must be better than 0.001 pixel of an azimuth cell size (Prats-Iraola et al., 2012; Wang et al., 2017; Fattahi et al., 2017). This accuracy requirement is significantly stricter than that for the range direction (i.e., 0.1 pixel of a range cell; Yagüe-Martínez et al., 2016).

Poor azimuth alignment introduces a ramp (i.e., a phase bias) within the TOPS burst interferogram, as well as a linear ramp along the along-track direction over the entire interferogram (De Zan and Monti Guarnieri, 2006). Since these ramps are cyclic among consecutive bursts (i.e., they reset at each burst boundary), they are manifested as phase jumps along the overlapping areas of consecutive bursts (cf. Figure1) (Fattahi et al., 2017).

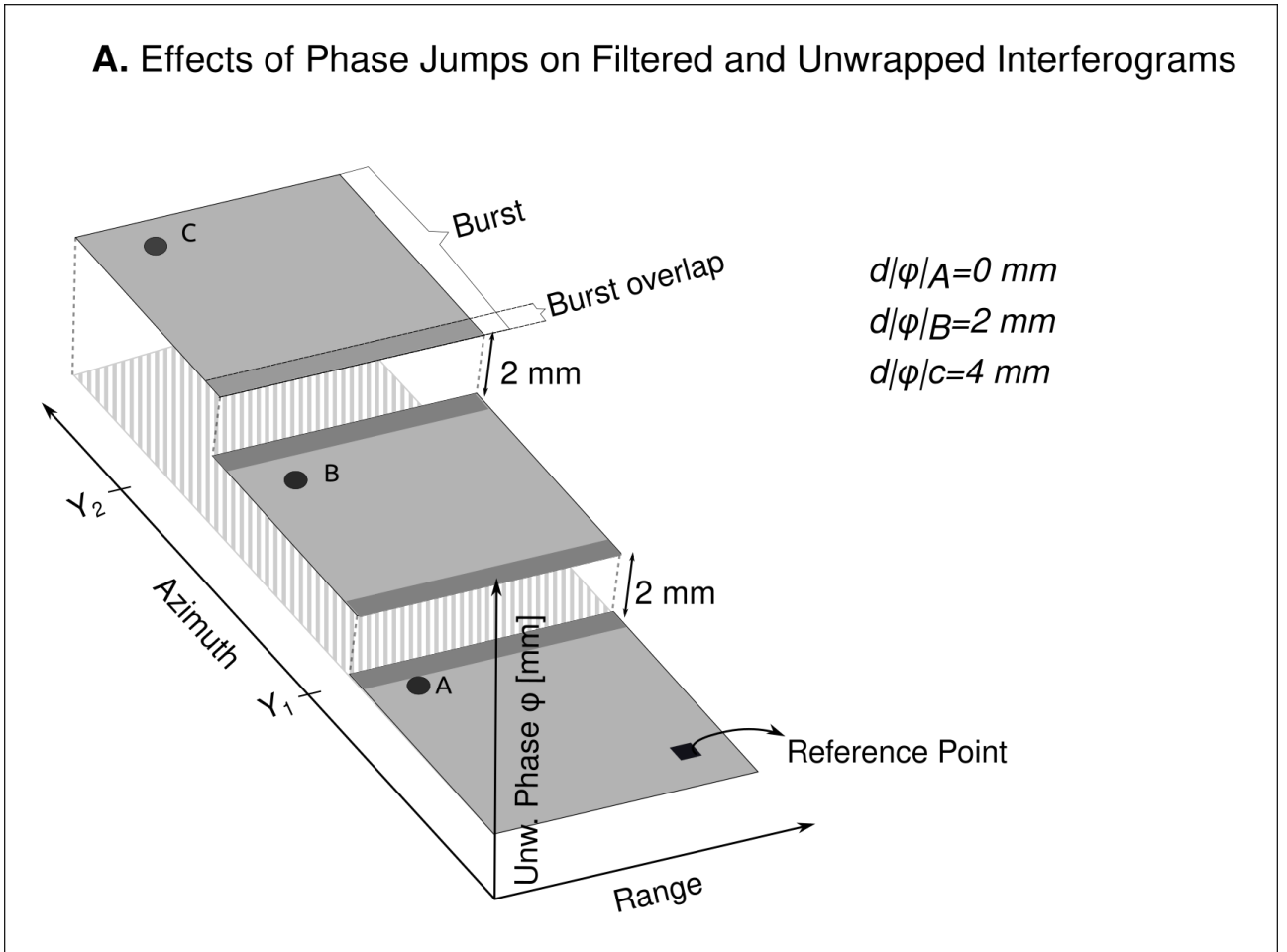


Figure 1: Conceptual drawing of the effects of phase jumps on filtered and unwrapped interferograms.

The azimuth misalignment is minimized through specific processing methods, such as Enhanced Spectral Diversity (ESD) (Prats-Iraola et al., 2012) and its variants (e.g., Fattahi et al., 2017; Wang et al., 2017). However, azimuth alignment may still be suboptimal, leading to small-

33 magnitude phase jumps that remain noticeable at burst boundaries. Subsequent steps, such as  
34 multilooking and filtering of the wrapped phase, may obscure these jumps, but they will still  
35 be visible in the unwrapped phase.

36 Identifying and quantifying phase jumps is a time-consuming process, particularly in datasets  
37 containing thousands of interferograms. Furthermore, manual inspection does not allow mea-  
38 surement of the magnitude of these jumps. To address this, we present an algorithmic approach  
39 that helps users to 1) detect phase jumps at burst boundaries; 2) measure their magnitude; 3)  
40 examine outliers; and 4) identify pairs or dates for exclusion from interferometric analysis. In  
41 the following, we discuss its implementation.

## 42 2 Detection of Phase Jump at Burst Boundary

43 Building on the ideas of Zhong et al. (2014) and Wang et al. (2017), we developed an algo-  
44 rithm to detect systematic phase jumps in unwrapped interferograms and quantify them using  
45 statistical techniques. We focus on the residual effects of phase jumps at burst boundaries,  
46 which are measured exclusively in the unwrapped phase. These phase jumps, when observed  
47 in the wrapped phase and in the absence of filtering or multilooking, resemble a Heaviside step  
48 function near the burst overlap. However, since filtering or multilooking is often applied before  
49 phase unwrapping, these phase jumps in unwrapped interferograms are more likely to resemble  
50 a sigmoid function.

51 The detection of phase jumps at burst boundaries is based on five key principles:

- 52 1. **Parsimony principle:** In a well-behaved unwrapped phase signal, the difference between  
53 adjacent cells is expected to be small. In other words, the values of adjacent cells should  
54 follow a smooth variation in the absence of significant deformation gradients. Phase jumps  
55 fail to follow this principle, which constitutes the key point of the phase jump detection.
- 56 2. **Systematic phase jumps:** for a given pair, a phase jump in the burst overlap area is  
57 systematic. This means that all cells along a specific azimuth coordinate (corresponding  
58 to the burst overlap) exhibit a phase shift. One can think of this phase jump as an abrupt  
59 disruption in phase continuity.

3. **Consistency across interferograms:** phase jumps consistently appear at the same coordinates across multiple interferograms. This is based on the fact that interferometry with TOPS mode requires perfectly synchronized bursts (De Zan and Monti Guarnieri, 2006). Although this principle applies to Sentinel-1 (Torres et al., 2012) and TerraSAR-X (Prats-Iraola et al., 2012) missions, it may not hold for non-synchronous TOPS acquisitions such as the SAOCOM mission (e.g., Euillades et al., 2024).

4. **Significance of phase jumps:** If significant enough, phase jumps at burst boundaries can produce residuals that remain detectable in the unwrapped phase. This is true even after the multilooking and filtering steps have been applied to the wrapped phase. However, the detectability and prominence of phase jumps may vary depending on how consecutive TOPS-interferogram bursts are stitched together (e.g., averaging across overlapping areas, pasting the first burst onto the following one, etc.).

5. **Prevalence of phase jumps at burst boundaries in stacks:** Prevalence implies that most of the data have a certain percentage of pairs with notorious or significant enough phase jumps at burst boundary coordinates.

To address the problem of detecting phase jumps at burst overlap, we divide it into two subproblems. Firstly, we identify the coordinates where interferogram bursts were merged. Secondly, we measure their magnitude, aiming to detect pairs with large phase jumps.

## 3 Identifying Burst Boundary Coordinates

### 3.1 Data Selection and Preparation

We perform a first-order selection of the pairs that can be evaluated. Pairs with extremely low coherence (we define this as a median coherence below 0.4) cannot be effectively assessed, as they lack sufficiently large areas to detect phase jumps.

The remaining pairs are masked based on their corresponding coherence. Cells with a coherence value lower than a predefined value (typically 0.75) are masked out throughout the entire assessment. The masking procedure aims to remove local phase jumps caused by processes

other than azimuth misalignment (e.g., speckle noise, deformation).

Let us consider a data set of  $R$  pairs of interferograms with dimensions  $(Y \times X)$  in radar coordinates, where  $Y$  is the total number of rows and  $X$  is the total number of columns. For a given pair  $r$ , the unwrapped phase  $(\phi)$  at a cell location  $(i,j)$  is considered reliable if the coherence  $(\gamma_{r,i,j})$  meets the following condition (Eq. (1)):

$$\phi_{r,i,j} = \begin{cases} \phi_{r,i,j}, & \text{if } \gamma_{r,i,j} > 0.75 \\ \text{No Data,} & \text{otherwise} \end{cases} \quad (1)$$

Note that  $i$  is a coordinate along the azimuth direction ( $i \in Y$ ) and  $j$  along the range direction ( $j \in X$ ).

### 3.2 Quantifying Phase Gradient

We calculate the absolute gradient of the unwrapped phase  $(|\partial\phi/\partial Y|)$  along the azimuth direction  $Y$  (hereafter referred to as phase gradient). The phase gradient is derived as follows:

$$\left| \frac{\partial\phi_{(r,i,j)}}{\partial Y} \right| = \begin{cases} |\phi_{(r,i,j)} - \phi_{(r,i+1,j)}|, & \text{if } i < Y \\ 0, & \text{if } i = Y \end{cases} \quad (2)$$

Note that when the row  $i$  equals the total number of rows  $Y$ , the phase gradient equals zero. In this way, the phase gradient maintains the same number of rows and columns  $(Y \times X)$  as the unwrapped phase.

### 3.3 Classifying the Phase Gradient

To systematically distinguish cells with significant phase gradients, we classify the phase gradient. The classified gradient  $(\text{clas}_{r,i,j}^{\text{grad}})$ , is defined by the following rules:

$$clas_{r,i,j}^{grad} = \begin{cases} \text{No Data,} & \text{if } \left| \frac{\partial \phi_{(r,i,j)}}{\partial Y} \right| = \text{No Data} \\ 1, & \text{if } \left| \frac{\partial \phi_{(r,i,j)}}{\partial Y} \right| > M_r \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where  $M_r$  is the median phase gradient per pair.

Note that in Eq. (3), cells identified with 1 have a significant phase gradient, while cells with 0 do not. According to Eq. (3), the gradient is considered significant if its value is greater than the median phase gradient of the pair.

### 3.4 Summarizing the Classified Gradient on a Row-Wise Basis

From the classified phase gradient, we derive a new property to identify phase jumps on a row-wise basis. The phase-gradient intensity is the proportion of cells along a row with significant phase gradient (i.e., cells marked with 1 in Eq. (3)). We only consider unmasked cells. We refer to the phase-gradient intensity as intensity, and it is expressed as:

$$intensity_{r,i}^* = \frac{\sum_{j=1}^{j=X} clas_{r,i,j}^{grad}}{C_{r,i}} \times 100 \quad (4)$$

With  $intensity_{r,i}$  the intensity and  $C_{r,i}$  the number of unmasked cells per row.

Note that, for a given pair,  $intensity_{r,i}$  and  $C_{r,i}$  variables have dimension (Y), which means that the X dimension has been compressed.

Eq. (4) provides an indicator of the systematic nature of the phase jump. The closer the value is to 100%, the greater the likelihood that the row corresponds to a phase jump at burst overlap. Conversely, local phase jumps will have lower intensity values. Although the intensity allows comparing different rows regardless of the number of unmasked cells, it also assigns the same relevance to rows with an extremely low cells. This issue is addressed in the next section.

### 3.5 Detecting the Coordinates of Burst Overlap

We rely on the intensity to detect the overlap of the bursts and their coordinates. It should be noted that we detect these coordinates rather than deduce them from the regularity of the burst.

Before detecting the coordinates, it is necessary to preprocess the intensity data. Preprocessing includes applying row-reliability masking followed by data detrending.

The reliability of a row is determined by its number of unmasked cells ( $C_{r,i}$ ). In most cases, this number is smaller than the total number of cells per row (i.e.,  $C_{r,i} < X$ ). However, the fewer cells a row contains, the less reliable it is. To set an appropriate threshold to define reliable rows, we apply a dynamic selection approach.

Dynamic selection compares the percentile of the distribution of unmasked cells ( $C$ ), with the percentage  $p_k$  of the total cells per row  $X$ . Both, percentile and percentage, are calculated for a predefined value  $k$  (typically  $k$  equals 25). At the end, the threshold  $t$  is set to be equal to the largest value ( $t = P_k$  if  $P_k > p_k$ , otherwise  $t = p_k$ ).

Once the threshold of row reliability is established, we mask the intensity as follows:

$$intensity_{r,i} = \begin{cases} \text{No Data,} & \text{if } C_{r,i} < t \\ intensity_{r,i}, & \text{otherwise} \end{cases} \quad (5)$$

In a posterior stage, the masked intensity is detrended. The trend arises from decreasing the coherence along the azimuth direction, which results in an increase of the intensity. We remove this trend by dividing each row by its median intensity among pairs.

We obtain the gradient along the  $Y$  dimension from the detrended intensity. This gradient allows us to obtain the coordinates of peaks in the intensity. The gradient is computed as Eq. (6).

$$\frac{\partial intensity_{r,i}}{\partial Y} = \begin{cases} intensity_{r,i}^d - intensity_{r,i+1}^d, & \text{if } i < Y \\ 0, & \text{if } i = Y \end{cases} \quad (6)$$

140 where  $intensity_{r,i}^d$  is the detrended intensity, and  $\frac{\partial intensity_{r,i}}{\partial Y}$  represents its gradient along the  
 141  $Y$  dimension. To avoid confusion with the gradient defined in Eq. (2), we will refer to it as the  
 142 gradient of intensity.

143 We find the  $i$ -coordinates of peaks in the gradient of intensity. A peak is detected if the  
 144 gradient  $\frac{\partial intensity_{r,i}}{\partial Y}$  is three times larger than the standard deviation, i.e., where  $\frac{\partial intensity_{r,i}}{\partial Y} >$   
 145  $3 \times \text{std}(\frac{\partial intensity_r}{\partial Y})$ . This stage is known as *coarse phase-jump coordinates detection*.

146 We analyze the set of  $i$ -coordinates detected in the previous step. We denominate this stage  
 147 as *phase-jump coordinates refinement*. Ideally, the same coordinates should appear in all pairs  
 148 if they align with the burst overlaps. However, this may not be the case if the masks fail to  
 149 identify and remove unreliable rows.

150 To refine the coordinates, we separate them into  $b$  groups. The number of groups corresponds to  
 151 the expected number of burst overlaps (i.e.,  $b = B - 1$ ). Each group is centered on a theoretical  
 152  $i$ -coordinate matching a burst overlap position. For each burst overlap number, the central  
 153 coordinate is given by the size of the burst ( $\sim (Y/B)$ ) multiplied by the burst overlap number.  
 154 Thus, the central coordinate at a given burst overlap number  $n$  is:

$$i_n^c = \text{int}(Y/B) * n \quad (7)$$

155 Where  $i_n^c$  is the central coordinate, and  $\text{int}$  denotes integer division. The burst number  $n$  ranges  
 156 from 1 to  $b$ .

157 We establish the limits of each group as half of the burst coordinates after and before its central  
 158 coordinate (i.e.,  $i_n^c \pm (\text{int}(Y/B) * 0.5)$ ).

159 Within each group, we store the most frequent coordinate found. Ideally, this refinement  
 160 ends with  $b$  coordinates. In practice, if the pairs have generally low coherence, the number of  
 161 coordinates found may be less than  $b$ .

### 162 3.6 Estimating Ramp by Phase Jump

163 We measure the phase gradient (Eq. (2)) on a row-wise basis as the median phase gradient  
 164 per row ( $pj_{r,i}$ ). We examine these medians only at the  $i$ -coordinates identified as burst overlap



during the refinement stage.

To estimate the overall ramp introduced by the phase jumps at the burst overlap,  $pj_r$ , we compute the average of the phase jumps measured and multiply it by the number of burst overlaps. Thus:

$$pj_r \sim \frac{\sum pj_{r,i}}{n_f} \times b \quad (8)$$

With  $n_f$  is the number of measured phase jumps at burst overlap, and this number can be less than the number of burst overlaps (i.e.,  $n_f < b$ ). Note that if both values are equal (i.e.,  $n_f = b$ ), the ramp of the phase jumps is the sum of the gradients per row at a given pair.

## 4 Detecting and Excluding Pairs

Finally, we detect dates or pairs that should be excluded from further processing. In this case, we establish a threshold for the maximum ramp (Eq. (8)) that a user can reasonably accept (typically around half a centimeter). In this way, we identify pairs that exhibit a phase-jump-derived ramp exceeding the predefined threshold. All these pairs should be excluded from time series processing.

## 5 Code Implementation

The algorithm is implemented in Python. It relies on the Miami InSAR Time-series software in Python (MintPy) (Yunjun et al., 2019). In addition, we use the *xarray* open-source package (Hoyer and Hamman, 2017) for labeling, organizing the data, and generating outputs. Other essential tools include *NumPy* (Harris et al., 2020) and *Matplotlib* (Hunter, 2007).

We called the Python script `calculate_phasejumps_from_mintpystack.py` and it is available on github [https://github.com/UP-RS-ESP/PhaseJumps\\_InSAR](https://github.com/UP-RS-ESP/PhaseJumps_InSAR).

## 5.1 Input

The program requires as input a stack of pairs in *HDF5* format, generated using MintPy. The stack contains a wrapped phase, an unwrapped phase, coherence, and a connected component mask for each pair.

The stacks must be provided in radar coordinates and should cover the full dimension along the azimuth direction, meaning no subset should be applied along azimuth when loading the data with MintPy. Otherwise, the phase-jump coordinate refinement stage will fail. Additionally, if the stack results from the processing of two or three sub-swaths, an area of interest should be specified. The current implementation is not optimized for analyzing interferograms derived from the processing of multiple sub-swaths.

Input parameters to run the program are summarized in Table 1.

Table 1: Description of input parameters to run the program. Parameters indicated by **m** are mandatory, whereas parameters indicated with **o** are optional

| Parameter            | Description   |
|----------------------|---|
| - <i>-inDir</i>      | [ <b>m</b> ] Working directory containing <i>input/ifgramStack.h5</i> .<br>Format: <i>string</i>  |
| - <i>-n-burst</i>    | [ <b>m</b> ] Number of bursts processed to generate the interferograms. Format: <i>integer</i>  |
| - <i>-pct</i>        | [ <b>o</b> ] Value $k$ to define the row reliability masking threshold. Value in range 0.0-1.0. Format: <i>float</i> . Default: 0.25 (Eq. (5))                                |
| - <i>-cmin</i>       | [ <b>o</b> ] Minimum coherence of pixels to be assessed (in range 0 to 1). Format: <i>float</i> . Default: 0.75 (Eq. (1))   |
| - <i>-msk-avgCoh</i> | [ <b>o</b> ] Use an additional mask based on the average coherence of the stack. The option is automatically activated in case of poor coherence. Default: <i>Deactivated</i> |
| - <i>-pj-thr</i>     | [ <b>o</b> ] Maximum accumulated phase jump per pair in milliliter. Format: <i>float</i> . Default: 5   |

| Parameter          | Description (mandatory, optional)  |
|--------------------|--|
| - <i>-sub-x</i>    | [o] Area of interest along the range direction $(x_0 \ x_1)$ . Specify it in case of stacks generated after merging two or three subswaths. Format: <i>integer integer</i> . |
| - <i>-pair</i>     | [o] Perform calculations only for a pair. Format: <i>YYYYMMDD_yyyymmdd</i>   |
| - <i>-plot-ind</i> | [o] Plot every interferogram from the stack, altogether with the intensity mask. Default: <i>Deactivated</i>   |

## 5.2 Output

Output are saved in the working directory, within the subdirectory called *pj\_evaluation*. Figures are stored in the subdirectory *pj\_evaluation/figs*.

Output arrays are saved in netCDF format. The files have dimensions  $(pair \times Y)$ , with *pair* the pair name ( $pair = R$ ) and *Y* the azimuth coordinate. Dimension *pair* contains the pair names stored in string format as *YYYYMMDD\_yyyymmdd* (year, month, day). Dimension *Y* contains azimuth row label (in radar coordinates), with integer format.

Output naming convention of arrays follows the rule "variable\_unit" (Table 2). File name suffixes have the following meanings:

- *\_pct*: percentage
- *\_cts*: counts
- *\_mm*: millimeter

In case a pair is selected (with *pair* parameter), then all output arrays end with the pair name in format *YYYYMMDD\_yyyymmdd*.

Table 2: Description of nc output files generated by the algorithm. Output indicated with \* are only generated in case of activating the corresponding function

| File Name                   | Description  |
|-----------------------------|--|
| <i>intensity_pct.nc</i>     | Intensity per row ( Eq. (4)). Dimensions: ( <i>pair</i> $\times$ <i>Y</i> ).<br>Format: <i>Int16</i> , No Data: -999   |
| <i>coherence_cts.nc</i>     | Number of unmasked cells per row (i.e., number of pixels with coherence $> cmin$ ), after coherence masking on cell-wise basis (Eq. (1)). Dimensions: ( <i>pair</i> $\times$ <i>Y</i> ).<br>Format file: <i>Int16</i> , No Data: -999                |
| <i>median_az_grad_mm.nc</i> | Median absolute gradient per row (i.e. along azimuth direction) in millimeter unit. Dimensions: ( <i>pair</i> $\times$ <i>Y</i> ).<br>Format: <i>float32</i>   |
| <i>maskCoh_cts.nc</i>       | [*] Number of unmasked cells per row (coherence $> cmin$ ), from the average coherence of the data set. Saved if the option <i>msk-avgCoh</i> is activated. Dimensions: ( <i>pair</i> $\times$ <i>Y</i> ). Format file: <i>Int16</i> , No Data: -999 |

210 Additional output files are provided in text format (*.txt*). They contain all relevant information  
211 for the evaluation, such as statistics as well as the results of the phase jump evaluation (Table  
212 3).

Table 3: Description of output text files generated by the algorithm

| File Name                          | Description   |
|------------------------------------|---|
| <i>stats_absolute_gradient.txt</i> | File containing summarized statistics (average, median, standard deviation) of coherence and absolute phase gradient (Eq. (2)), as well as the temporal baseline per pair from the stack. |

| File Name  | Description  |
|--|--|
| <i>magnitude_phase_jumps.txt</i>                           | File containing the overall magnitude of the phase jump, which is estimated with Eq. (8). Note that this magnitude is estimated only from pairs that are not excluded from the assessment (i.e., pairs with a median coherence $> 0.4$ ).  |
| <i>exclude_listdate12_interferograms_by_phase_jump.txt</i> | File containing all pairs that should be excluded from the assessment. This file includes pairs that were excluded from the assessment (median coherence below 0.4). Pairs are presented in two ways: by pair name (format YYYYMMDD_yyyymmdd) and by index that the pair has in the stack. |
| <i>exclude_dates_by_phase_jumps.txt</i>                    | File containing dates that should be excluded from the assessment if more than 50% of their pairs exhibit a significant phase jump   |

### Coherence-based Row Reliability

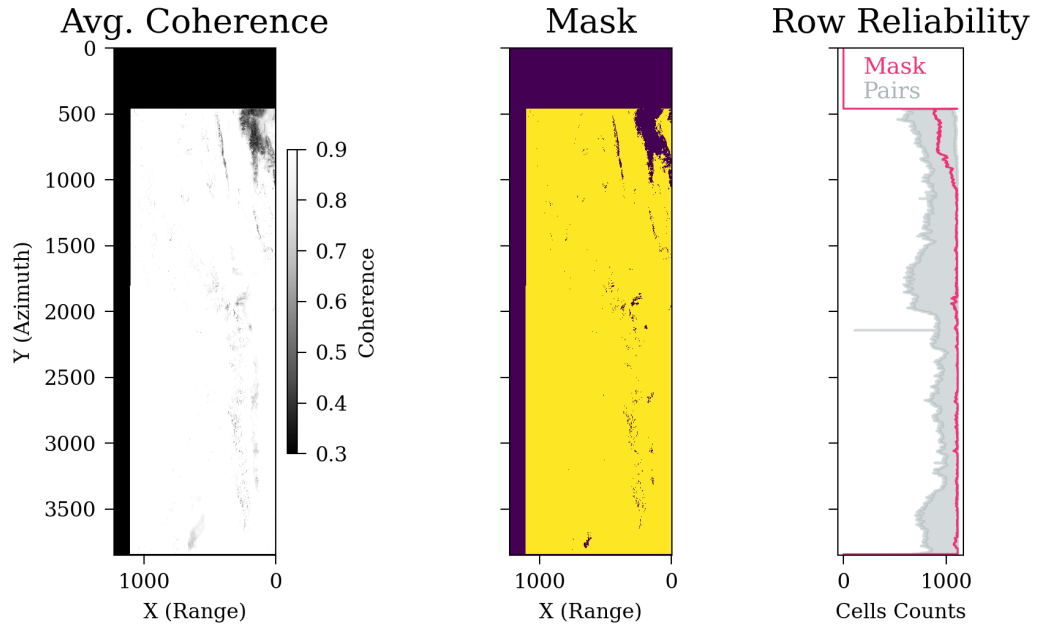


Figure 2: Example Output for the generation of the coherence mask.

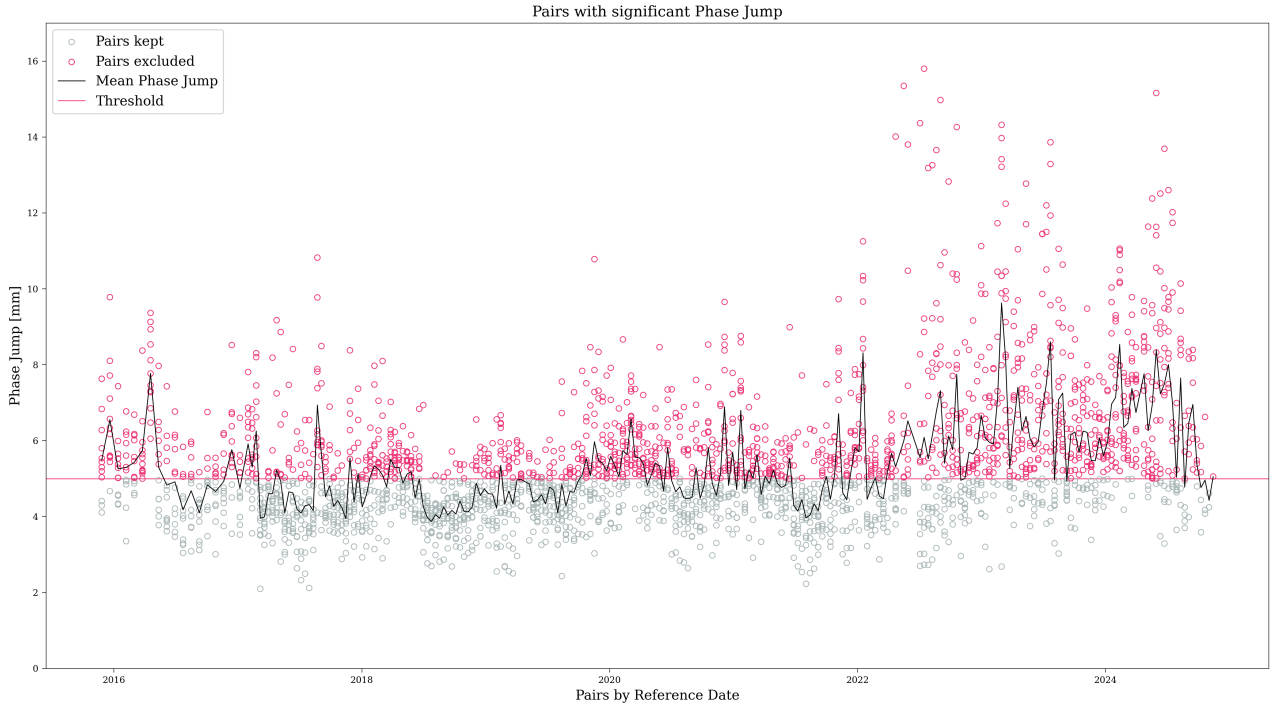


Figure 3: Example Output for a time series showing estimated phase jump magnitude for every date.

## References

- De Zan, F. and Monti Guarnieri, A. (2006). Topsar: Terrain observation by progressive scans. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2352–2360.
- Euillades, J. A., Roa, Y. L., Euillades, L. D., Euillades, P. A., Rosell, P. A., Solarte, E. A., and Perna, S. (2024). Interferometric assessment of saocom-1 topsar data. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5.
- Fattahi, H., Agram, P., and Simons, M. (2017). A network-based enhanced spectral diversity approach for tops time-series analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):777–786.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585:357–362.

227 Hoyer, S. and Hamman, J. (2017). xarray: N-d labeled arrays and datasets in python. *Journal*  
228 *of Open Research Software*.

229 Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engi-*  
230 *neering*, 9(3):90–95.

231 Prats-Iraola, P., Scheiber, R., Marotti, L., Wollstadt, S., and Reigber, A. (2012). Tops interfer-  
232 ometry with terrasar-x. *IEEE Transactions on Geoscience and Remote Sensing*, 50(8):3179–  
233 3188.

234 Sansosti, E., Berardino, P., Manunta, M., Serafino, F., and Fornaro, G. (2006). Geometrical  
235 sar image registration. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2861–  
236 2870.

237 Torres, R., Snoeij, P., Geudtner, D., Bibby, D., Davidson, M., Attema, E., Potin, P., Rommen,  
238 B., Floury, N., Brown, M., Traver, I. N., Deghaye, P., Duesmann, B., Rosich, B., Miranda,  
239 N., Bruno, C., L’Abbate, M., Croci, R., Pietropaolo, A., Huchler, M., and Rostan, F. (2012).  
240 Gmes sentinel-1 mission. *Remote Sensing of Environment*, 120:9–24. The Sentinel Missions  
241 - New Opportunities for Science.

242 Wang, K., Xu, X., and Fialko, Y. (2017). Improving burst alignment in tops interferometry  
243 with bivariate enhanced spectral diversity. *IEEE Geoscience and Remote Sensing Letters*,  
244 14(12):2423–2427.

245 Yagüe-Martínez, N., Prats-Iraola, P., Rodríguez González, F., Brcic, R., Shau, R., Geudtner,  
246 D., Eineder, M., and Bamler, R. (2016). Interferometric processing of sentinel-1 tops data.  
247 *IEEE Transactions on Geoscience and Remote Sensing*, 54(4):2220–2234.

248 Yunjun, Z., Fattahi, H., and Amelung, F. (2019). Small baseline insar time series analysis:  
249 Unwrapping error correction and noise reduction. *Computers & Geosciences*, 133:104331.

250 Zhong, H., Tang, J., Zhang, S., and Zhang, X. (2014). A quality-guided and local minimum  
251 discontinuity based phase unwrapping algorithm for insar/insas interferograms. *IEEE Geo-*  
252 *science and Remote Sensing Letters*, 11(1):215–219.