



HAL
open science

Exploitation of historical analog seismological records by image processing and machine learning

Polina Lemenkova

► **To cite this version:**

Polina Lemenkova. Exploitation of historical analog seismological records by image processing and machine learning. Doctoral. Bruxelles, Belgique, Belgium. 2023, pp.68. hal-04612863

HAL Id: hal-04612863

<https://hal.science/hal-04612863v1>

Submitted on 14 Jun 2024

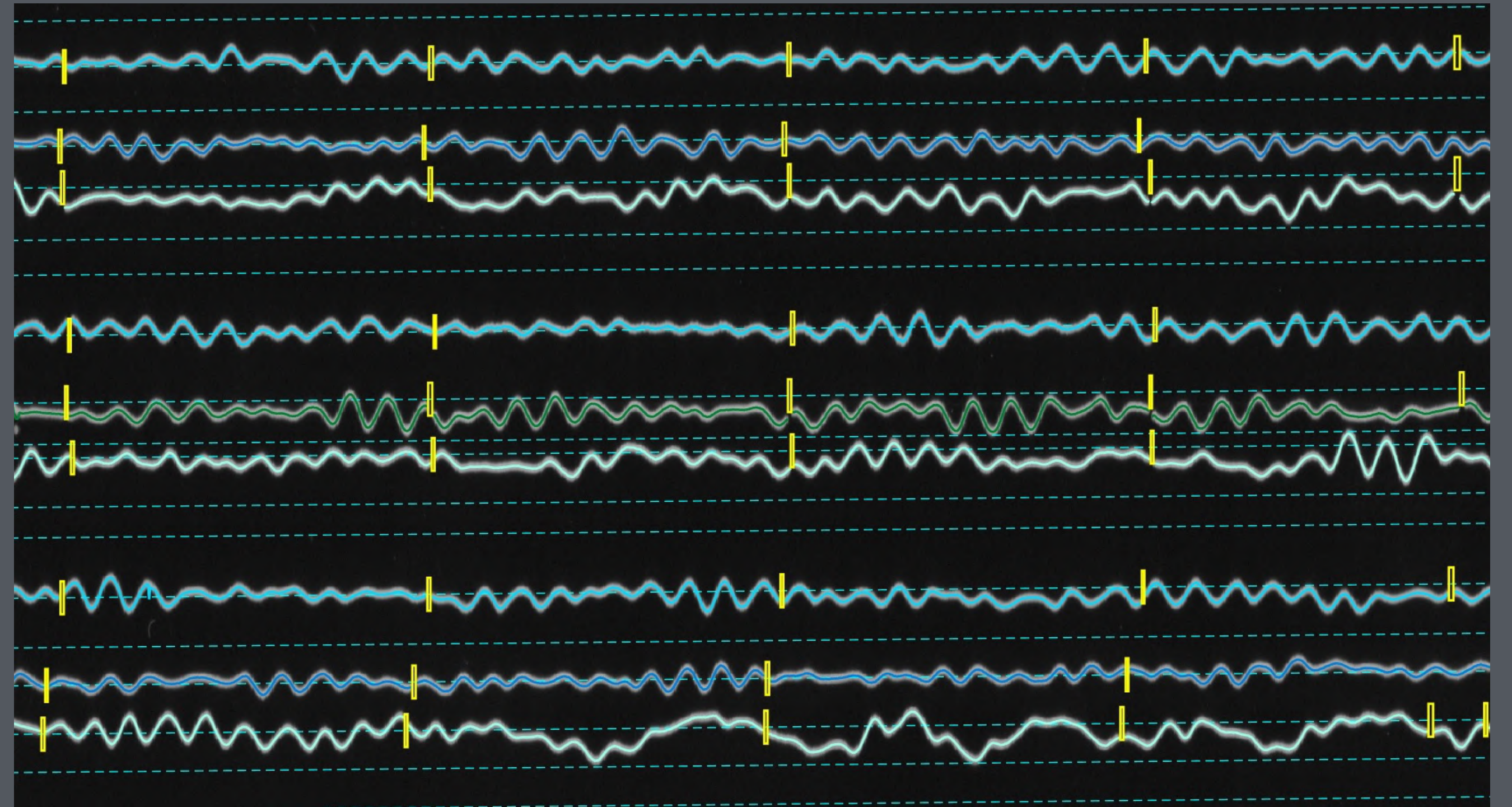
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Exploitation of historical analog seismological records by image processing and machine learning



SUPERVISORS Prof. Dr. Olivier DEBEIR (ULB) and Dr. Thomas LECOCQ (Royal Observatory of Belgium, Department of Seismology and Gravimetry, co-promoteur)

PRESENTATION PLACE Université Libre de Bruxelles, École polytechnique de Bruxelles (Brussels Faculty of Engineering), Laboratory of Image Synthesis and Analysis (LISA).

DATE 02.V.2023

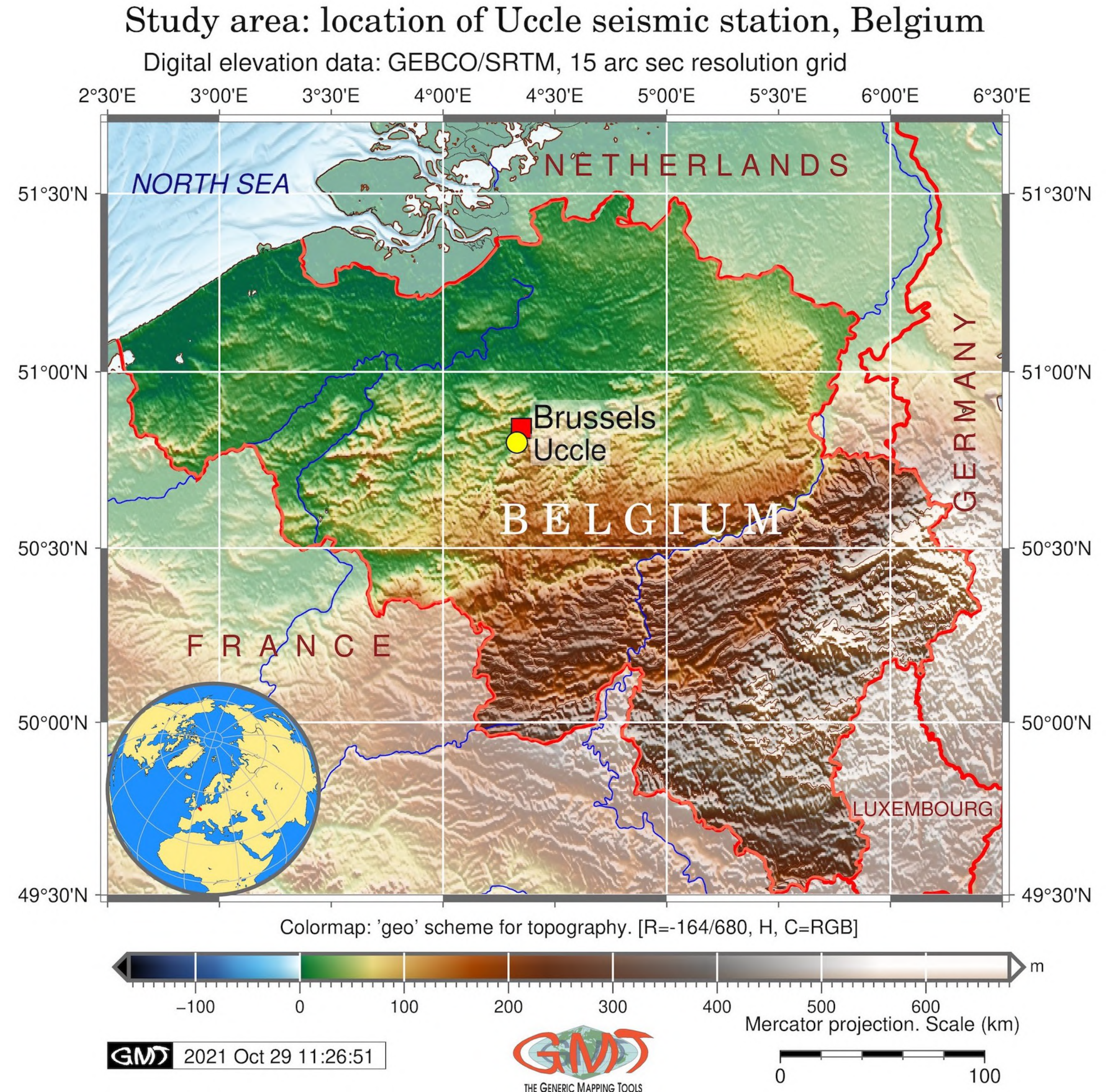
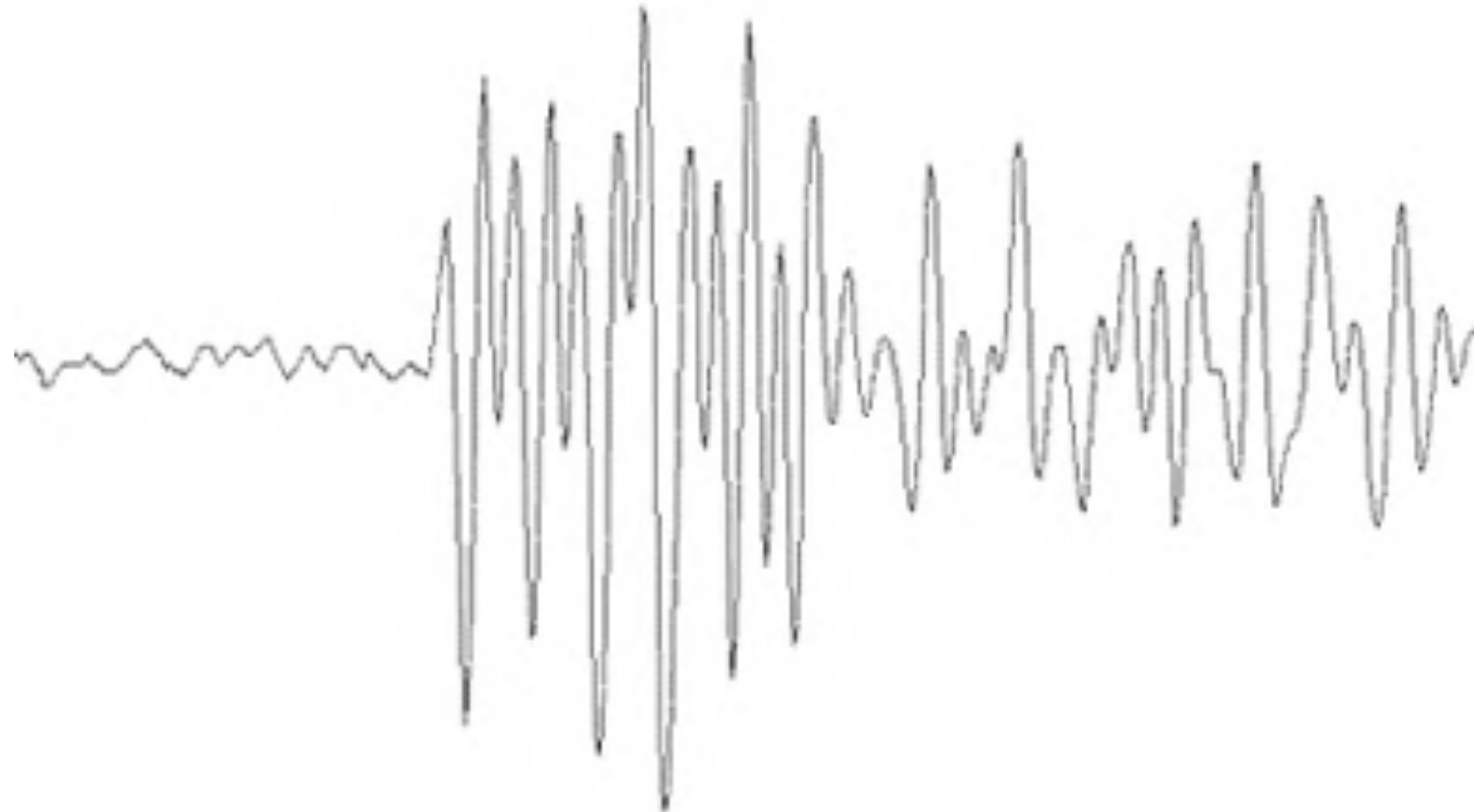
PRESENTER Polina Lemenkova

Part 1.

Project Objectives and Goals.
Data (Seismograms), Data Source (ROB)
Instruments (Galitzine Seismograph).

Research Object and Problem

- Study object => historical scanned seismograms in TIFF format from the archives of Royal Observatory of Belgium (ROB), Department of Seismology & Gravimetry.
- Study area => Uccle station (see map).
- Study problem => to digitise large archive of the old paper-based seismograms from ROB quickly, accurately and automatically.



Publication results of the Part 1 of the PhD project:

De Plaen, R. S. M.; Lecocq, T.; Lemenkova, P. ; Debeir, O.; Arduhin, F.; De Carlo, M. Extracting Microseismic Ground Motion From Legacy Seismograms. *In: Proceedings of the Third European Conference on Earthquake Engineering and Seismology, 2022-09-04: Bucharest, Romania. Conspress, Ed. 1, pp. 3507-3513. Publié, 2022-09-09.*



3rd EUROPEAN CONFERENCE ON
EARTHQUAKE ENGINEERING & SEISMOLOGY
BUCHAREST, ROMANIA, 2022

Extracting Microseismic Ground Motion From Legacy Seismograms

De Plaen Raphael - Seismology-Gravimetry, Royal Observatory of Belgium, Brussels, Belgium, raphael.deplaen@gmail.com

Lecocq Thomas - Seismology-Gravimetry, Royal Observatory of Belgium, Brussels, Belgium, thomas.lecocq@seismology.be

Polina Lemenkova - Université Libre de Bruxelles, Brussels Faculty of Engineering, Laboratory of Image Synthesis and Analysis, Brussels, Belgium, polina.lemenkova@ulb.be

Olivier Debeir - Université Libre de Bruxelles, Brussels Faculty of Engineering, Laboratory of Image Synthesis and Analysis, Brussels, Belgium, olivier.debeir@ulb.be

Fabrice Arduhin - Laboratoire d'Océanographie Spatiale, Institut Francais de Recherche pour l'Exploitation de la Mer, Plouzané, France, fabrice.arduhin@ifremer.fr

Marine De Carlo - Laboratoire d'Océanographie Spatiale, Institut Francais de Recherche pour l'Exploitation de la Mer, Plouzané, France, marine.de.carlo@ifremer.fr

3rd EUROPEAN CONFERENCE ON
EARTHQUAKE ENGINEERING & SEISMOLOGY
September 4-9, 2022

Home Conference Partners Afternoons@3ECEEES Fees Travel@3ECEEES Contact Us

THE GRAND EVENT IN SEISMOLOGY

September 4-9
Bucharest, 2022

A joint event of the
17th European Conference on Earthquake
Engineering & 38th General Assembly of the
European Seismological Commission

International Conference Centre, Bucharest,
Romania.

TICKETS

Abstract

Before digital recordings became available in the 1970s, the ground motion was recorded using ink on white paper, scratching black-smoked paper, or light on photographic paper. While those analog seismic records offer unique continuous observations from the last century, most of them are now stacked and archived in boxes and potentially exposed to physical decay and permanent loss. To preserve those records and ultimately subject them to modern methods of analysis, it is time-sensitive to scan and digitize them. Here, we worked on a method for automatic digitization of paper seismograms using image processing and machine learning to extract microseismic ground-motion periods and amplitudes. We implemented the method on legacy data recorded at the Royal Observatory of Belgium to extract power spectral densities for major storms during the last century, which are compared with modeled microseisms levels computed using a numerical ocean wave model. This further shows how digitizing analog seismograms does not only preserve the scientific legacy but also makes new research possible by bringing analog data to the digital age.

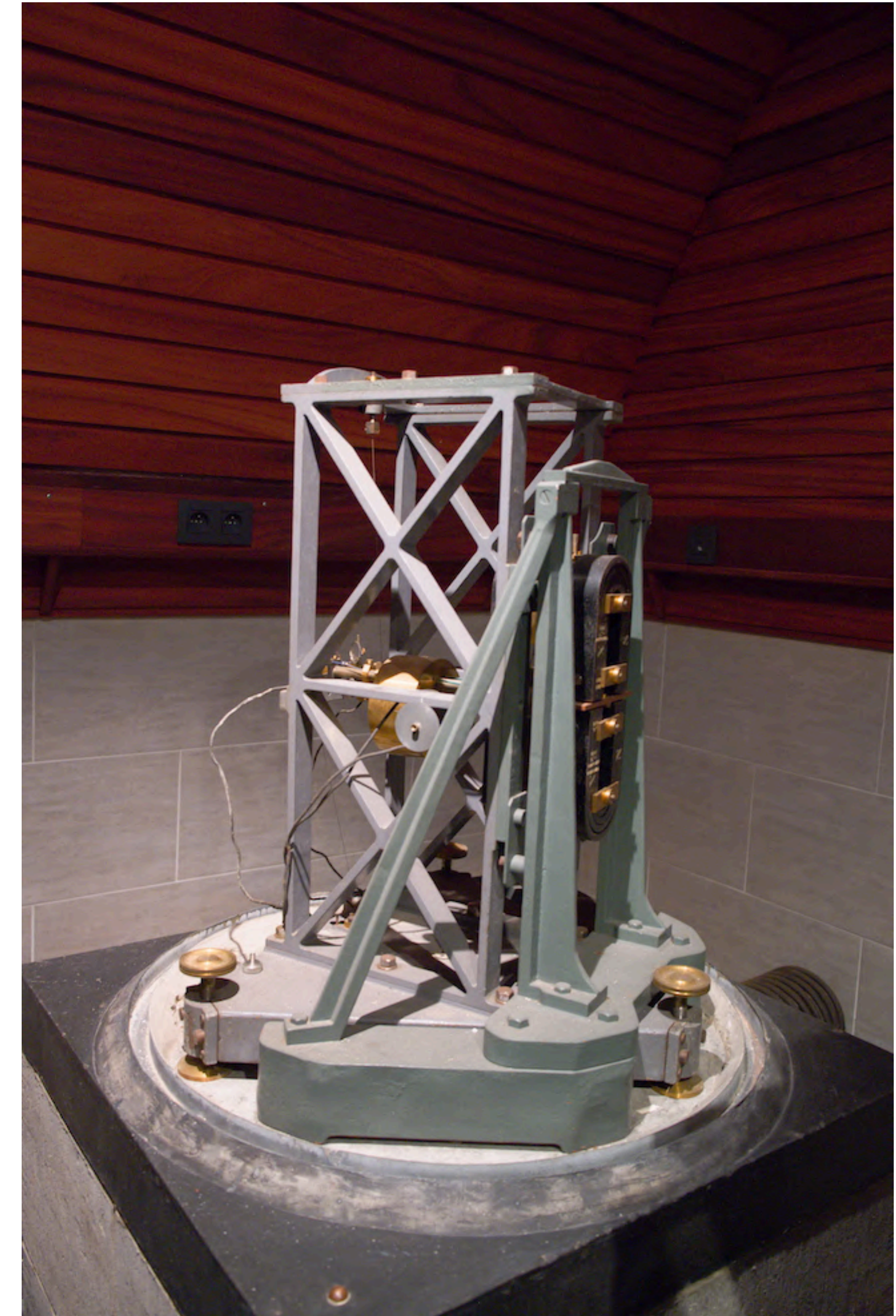
Keywords: analog seismograms; seismic noise; oceanic storms; digitalization; machine learning; oceanic climate

1. Introduction

The Royal Observatory of Belgium (ROB) has been operating seismic stations since the late 19th century with the first instrument, a von Rebeur-Ehlertr triple horizontal pendulum

Data and Instrument

- ✳️ There are various types of seismometers used in geophysics. In this study we used archived seismograms recorded in 1954 by the Galitzine seismometer in Uccle station.
- ✳️ Currently dataset included a collection of 145 images from 1 January 1954 to 12 March 1954
- ✳️ The period will be gradually enlarged as soon as other seismograms are scanned to cover 70 last years.
- ✳️ The most of the images are monochrome (B/W). Some other images are scanned in colour (RGB).
- ✳️ Some of the images are well preserved, some have distortions and defects visible on the aged paper

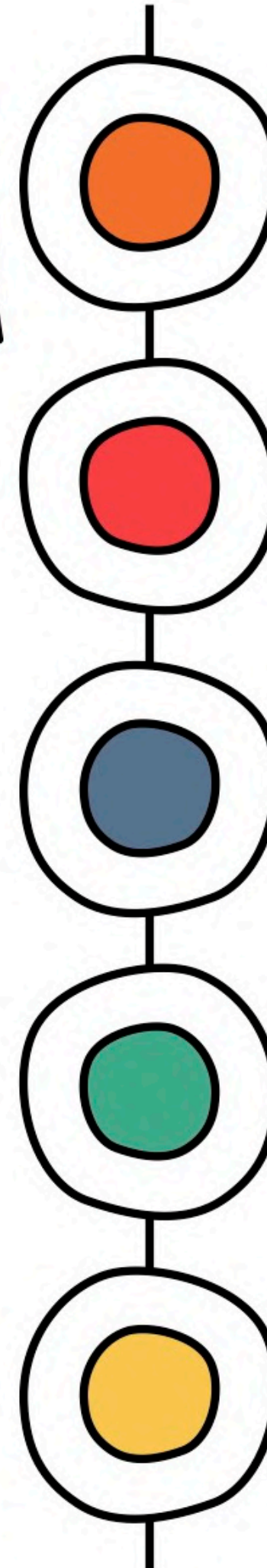


*Instrument used for data capture in 1954:
Horizontal Galitzine seismometer located in UCC.
Image source: courtesy of ROB. Photo: Raphaël S. M. De Plaen*

Research Questions

What are the best ML/DL approaches for automatic processing of seismograms? – Develop an effective repetitive workflow.

Methods & Approaches



How to assess and interpret signals using seismic data in a semi-automated and automated regimes?

Algorithms

How does seismic activity differs by years in Uccle station area (time-series analysis of large data 1950s+)?

Data Analysis

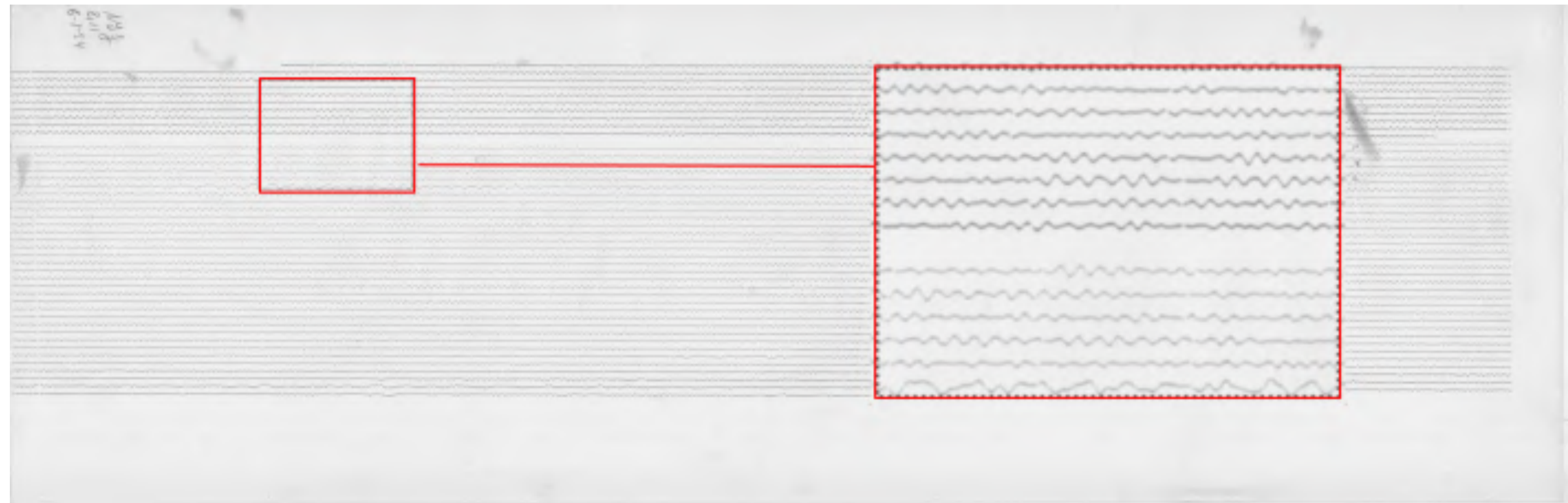
Processing & Modelling

How to effectively process and model big datasets of archived seismograms with minimised human workflow and maximised automation?

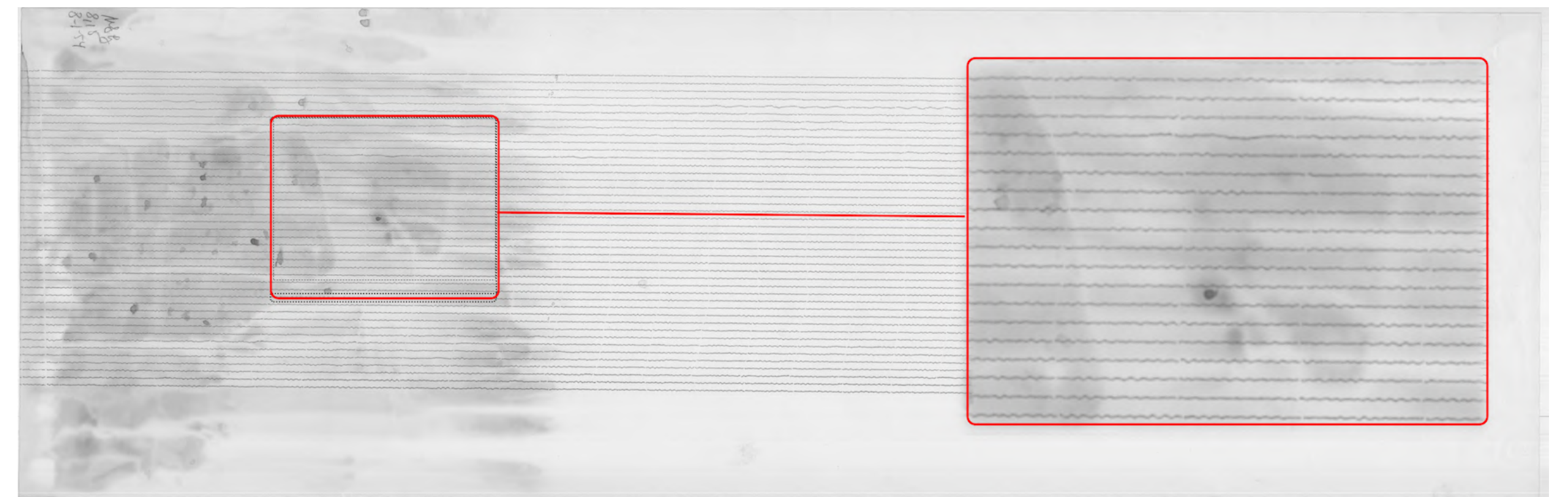
ML/DL

How to effectively train machine to identify timing for fully automated data processing (hours and minute marks/gaps)?

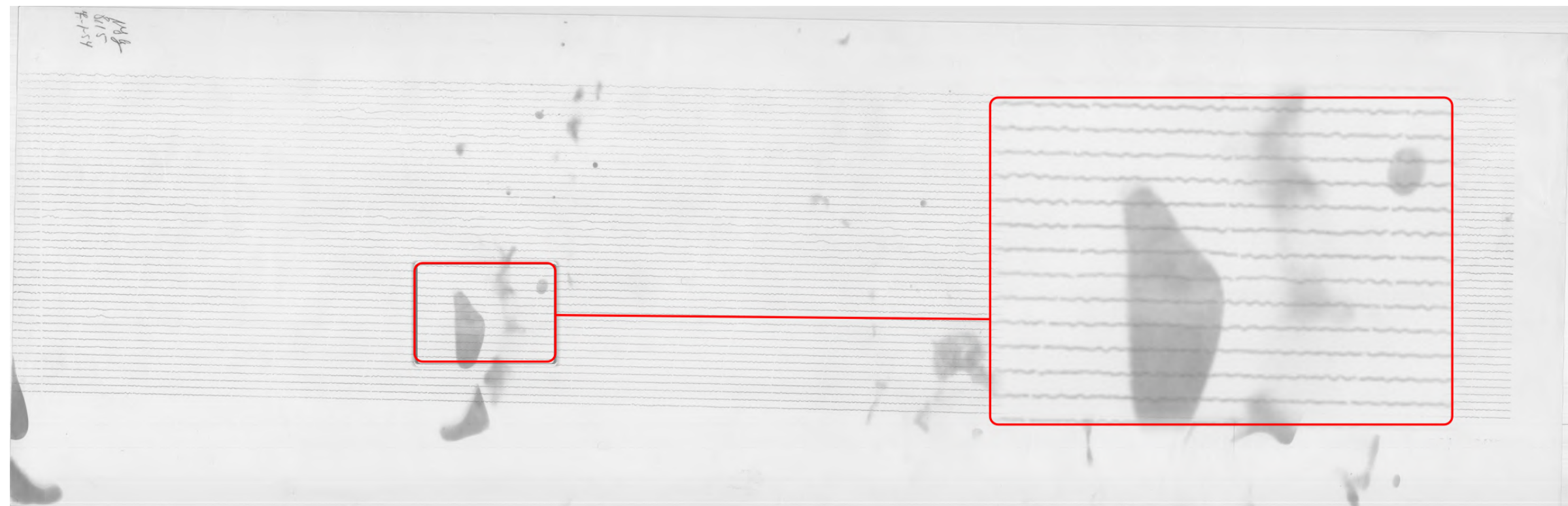
Examples of the raw data: paper-based seismograms



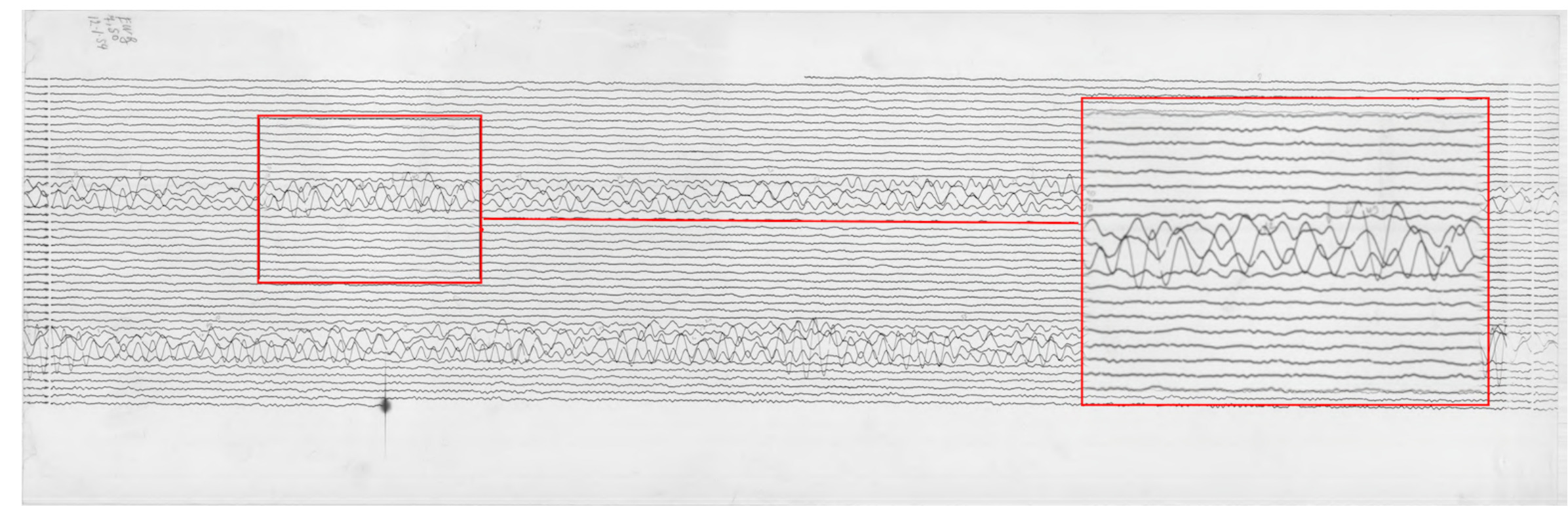
Empty records between the lines of seismic traces with enlarged fragment of seismogram. Here: UCC19540106Gal_N_0811.TIFF



Continuous noise dark background with blurred traces => lack of contrast for image recognition. Here: UCC19540108Gal_N_0815.TIFF



Partially spotted image caused by storage, with enlarged fragment of seismogram. Here: UCC19540107Gal_N_0815.TIFF

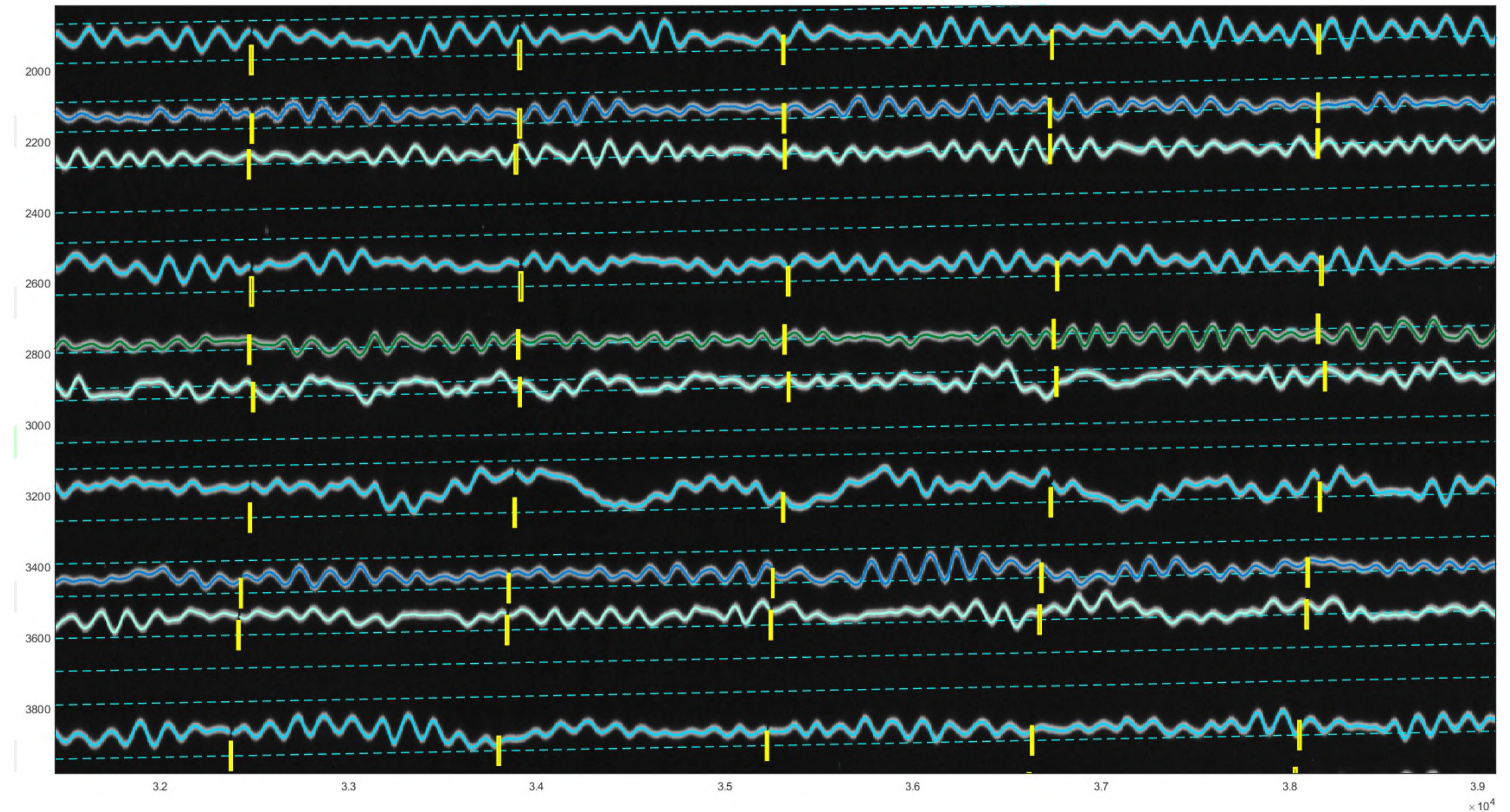


Overlapped traces => problems for recognition of trace direction during vectorising. Here: UCC19540112Gal_E_0750.TIFF

PROBLEMS ARE CAUSED BY TECHNIQUES OF OLD SEISMOGRAM RECORDING + TIME (SPOTS, BLURS, BROKEN PAPER, ETC.)

Actuality, Importance and Research Tasks

- *Manual digitising cannot provide accurate and rapid data processing for developing digitised big dataset of archived seismograms*
- *Seismic data cannot be processed manually and require **automatization and programming approaches***
- *We need to process big archives of seismic data from ROB effectively and quickly but accurately and precisely*
- *We need to analyse data with minimised human labour to derive information on earthquakes and ground motion*

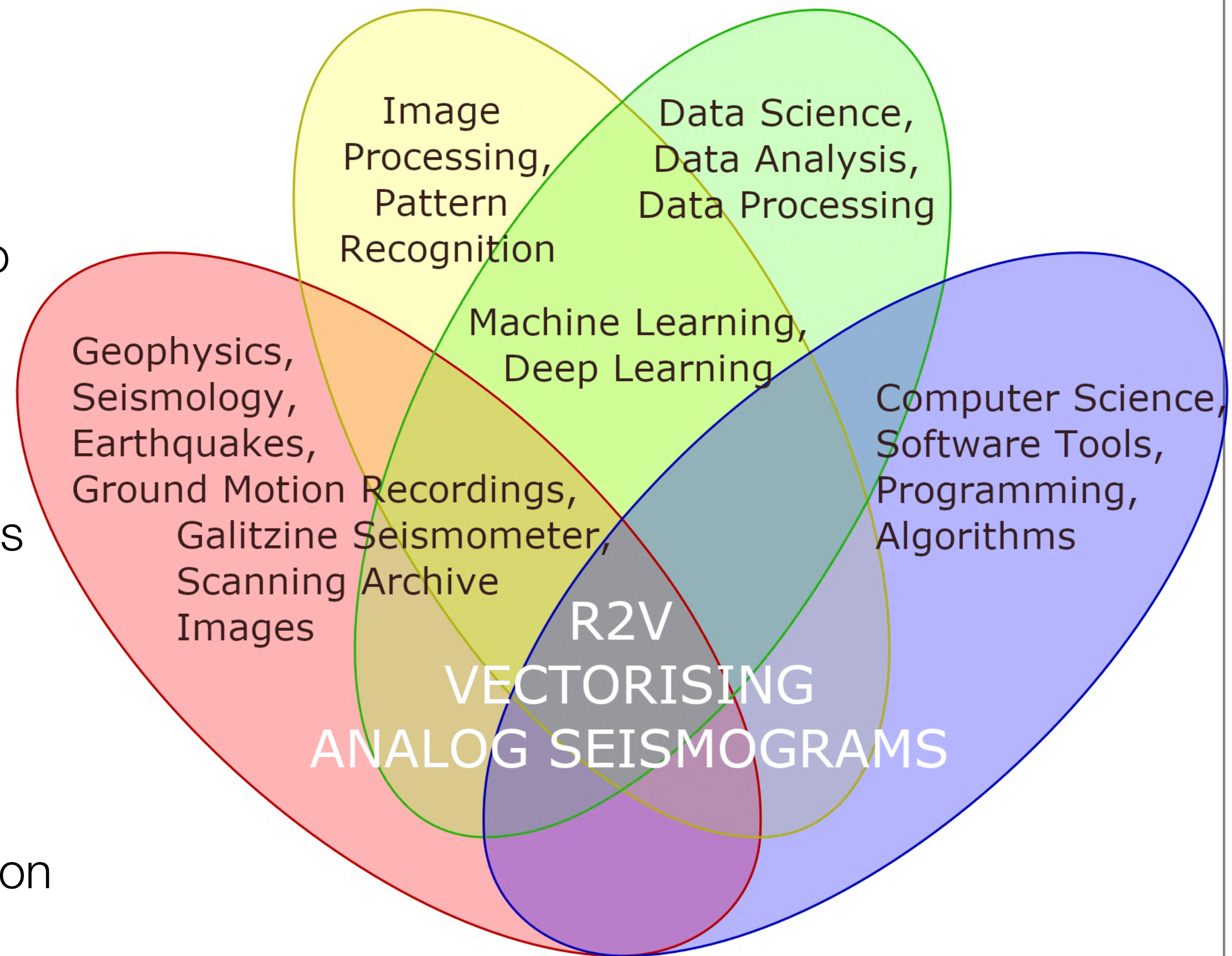


Example of the digitised seismograms using DigitSeis

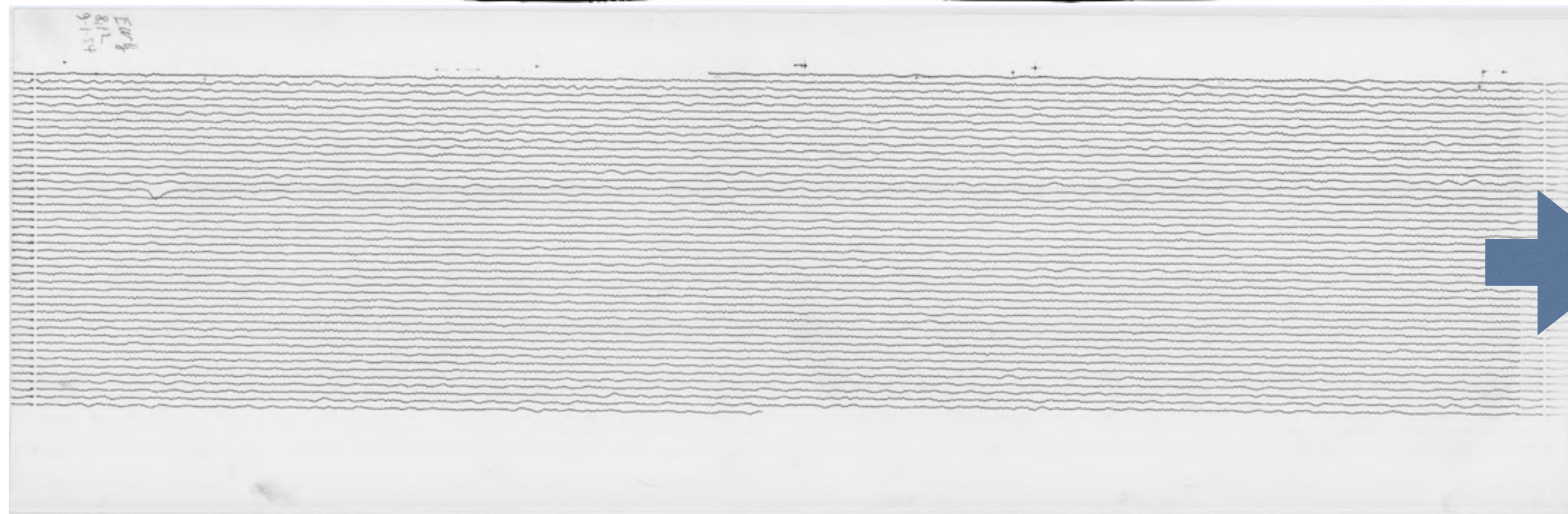
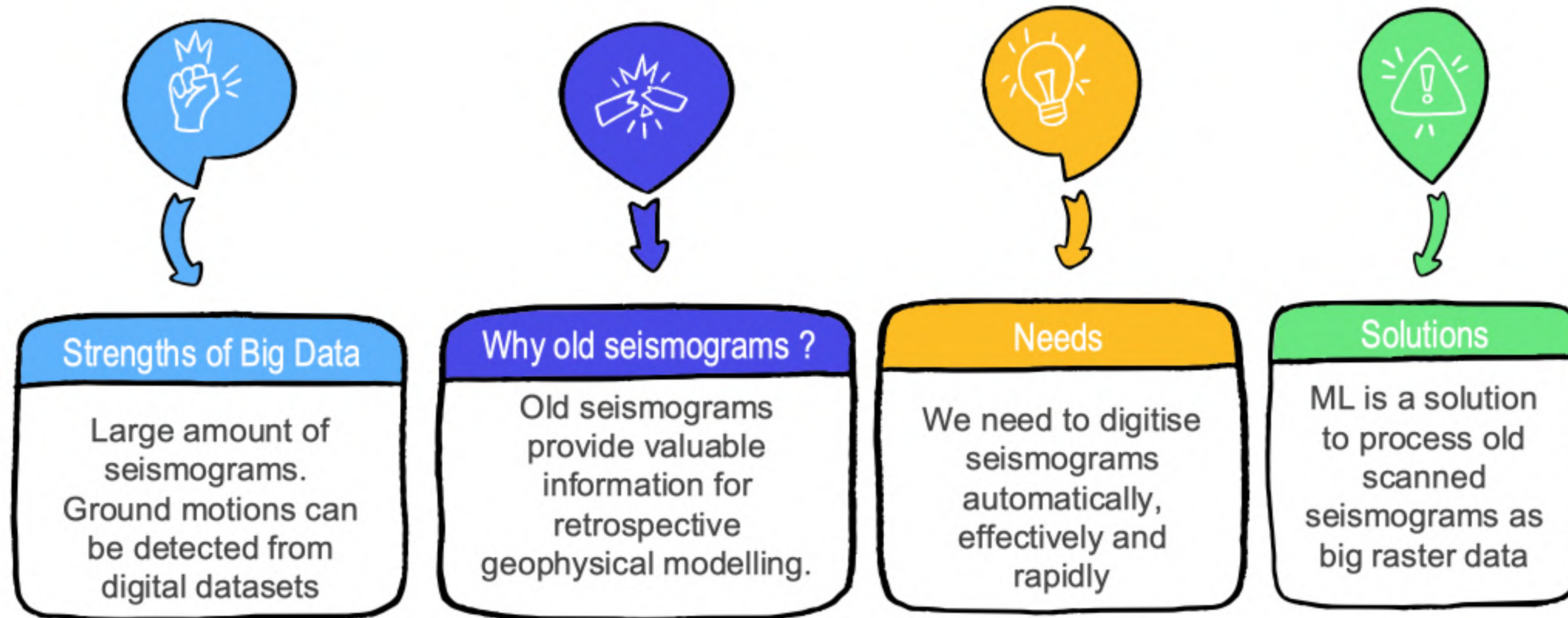
So far there are no existing integrated studies of digitising seismograms in big data volumes by ML methods. Only selected software exist (e.g. DigitSeis, SKATE, Teseo)

Interdisciplinary Nature of Project

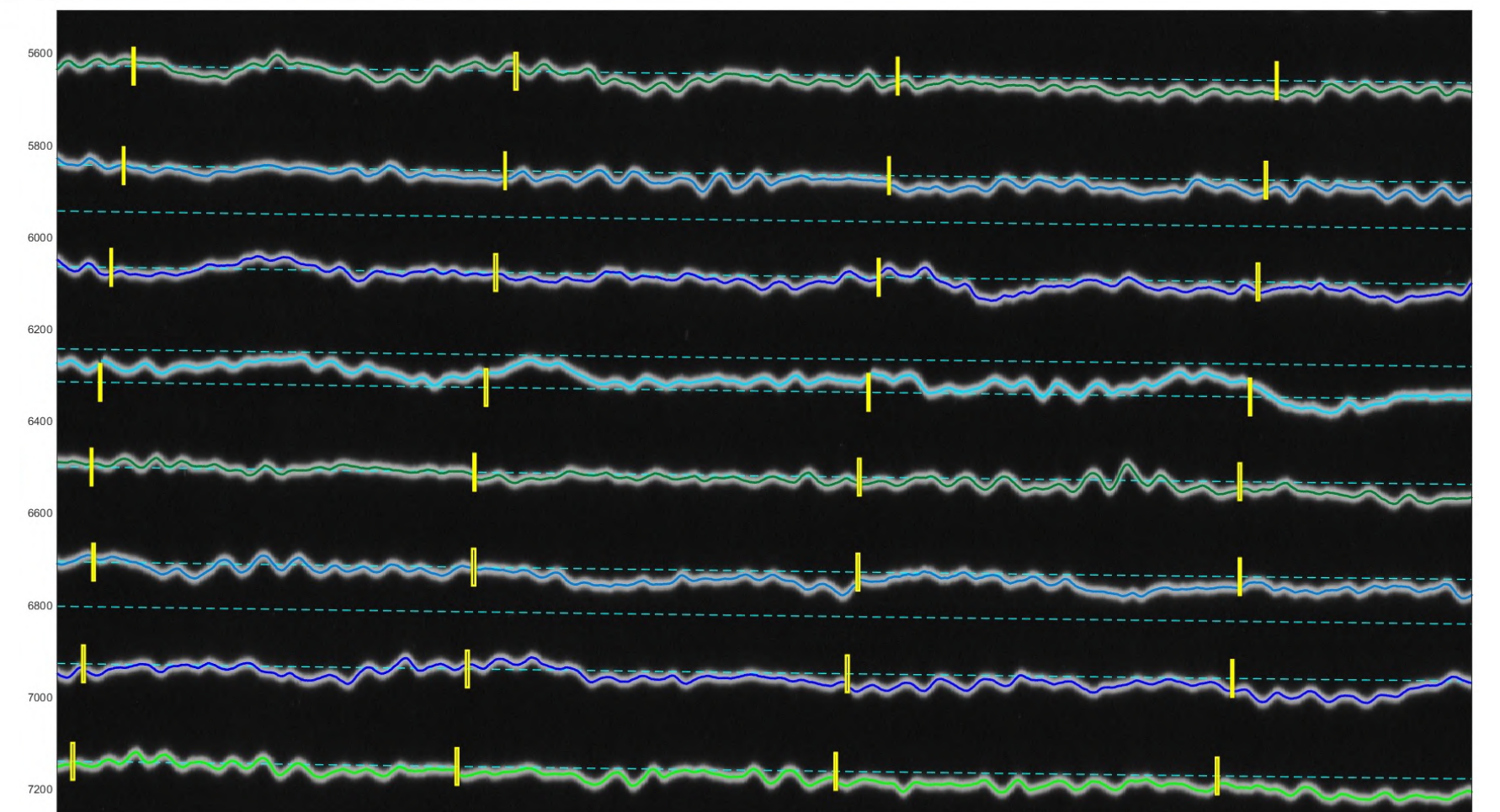
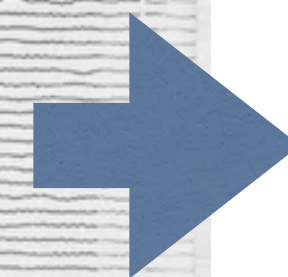
- Complexity of geophysical data processing requires integrated approaches
- The multi-disciplinary aspects of this PhD project consist of tight links between the two disciplines : **computer science** (software development, programming algorithms and tools) and **geophysics** (seismology).
- Applying ML to digitising seismograms brings new possibilities and benefits in seismology.
- Advantages of ML =>> accurate and rapid digitising of the scanned images, rapid processing of historical seismograms, improved techniques of automated recognition of signals and data interpreting.



Project Motivation, Strengths and Challenges



Old scanned raster seismogram (TIFF file)



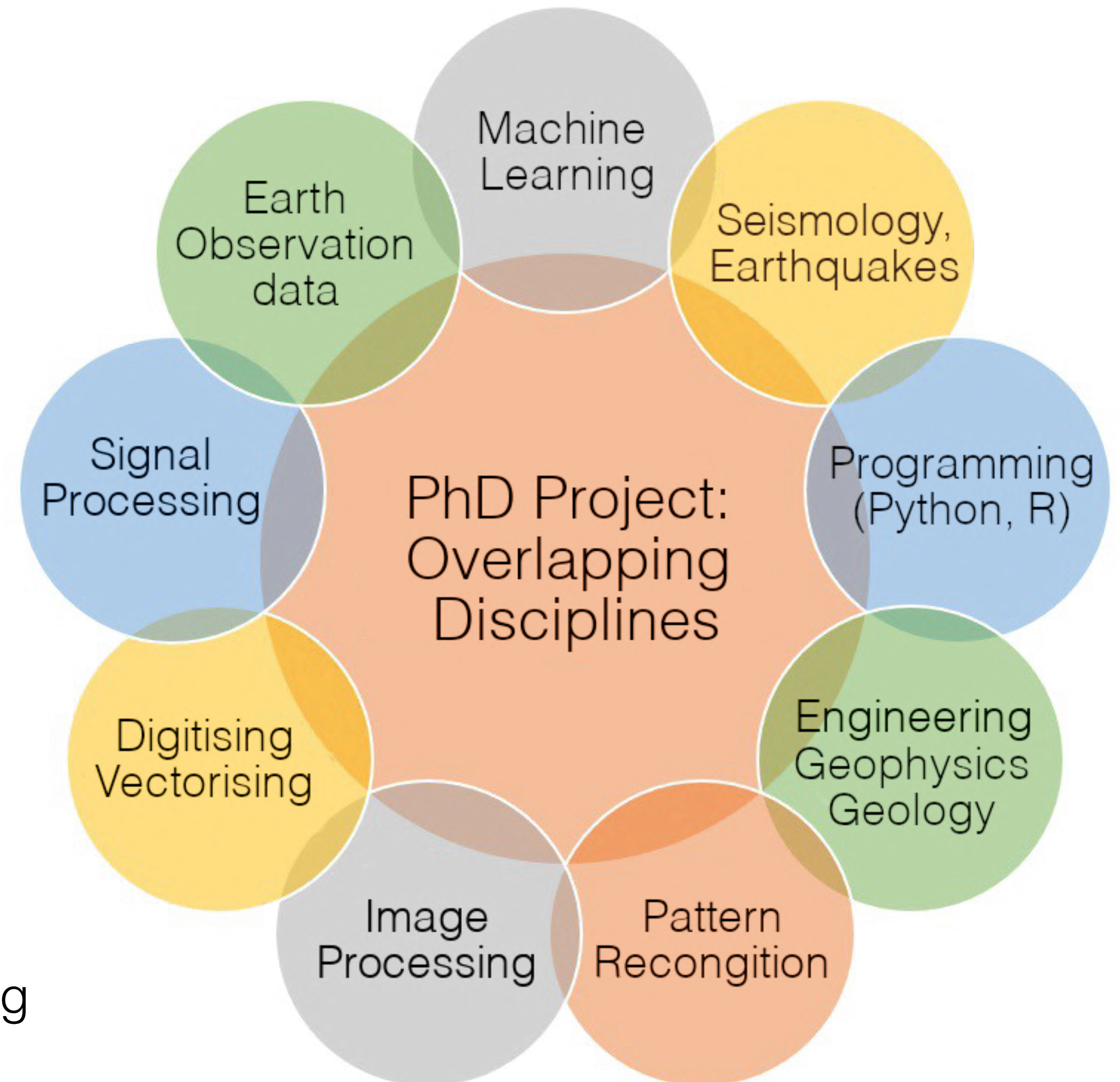
Fragment of the vectorised output (*DigitSeis*)

Various Approaches in One Study: Overlapping Disciplines

Our project presents an interdisciplinary research combining overlapping scientific clusters and engineering disciplines (image processing, geophysics, ML and data science).

A multi-disciplinary project integrates 3 major scientific clusters and several disciplines as subsections for vectorising seismograms:

1. Image Processing, Pattern Recognition, Computer Science, Programming, ML
2. Earth Observation data (ROB, Uccle archive), Geophysics and Seismology, Geology, Earthquake Engineering
3. Data Science, Data Analysis, Signal Processing Algorithms of Digitising & Vectorising



Goals and Objectives of my PhD Project

PhD Thesis

Research

Methods

Datasets

Vectorising

Education

PhD activities: 60 ETCS credits, thesis, presentations, conferences, seminars

Publications

Writing journals articles with my supervisors (Drs. O.Debeir and T. Lecocq) to publish in journals

Algorithms

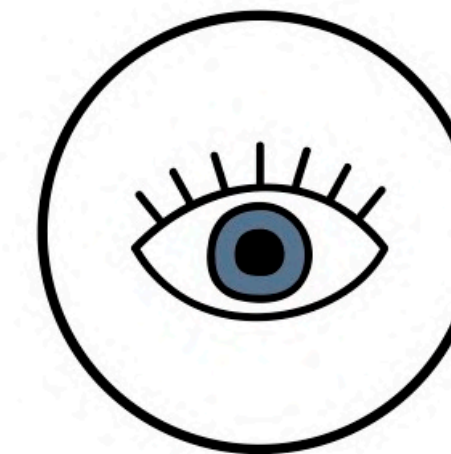
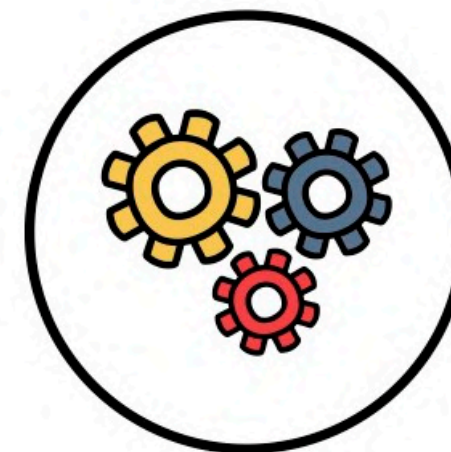
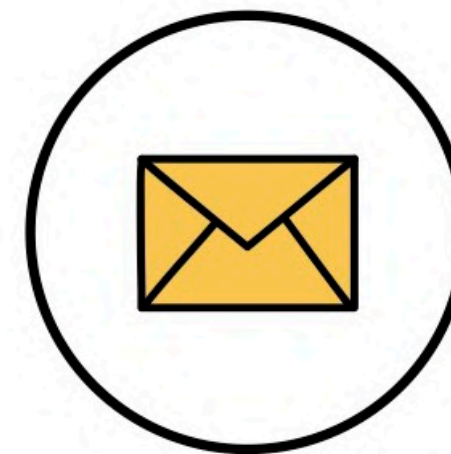
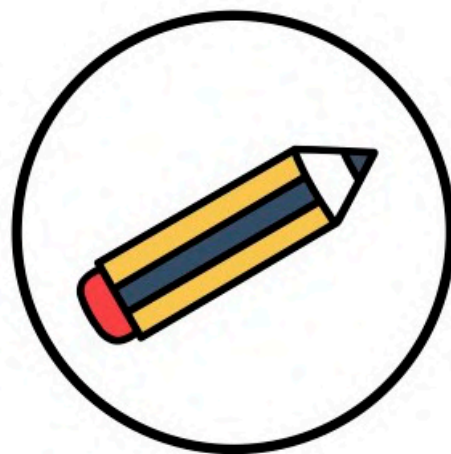
Developing and testing advanced approaches to automated vectorising

Data Analysis

Data-driven techniques of ML to analyse large seismic archives of ROB (1950s+)

Image Processing

Automated recognition of seismic traces by R2V techniques applied for ROB archived data



1

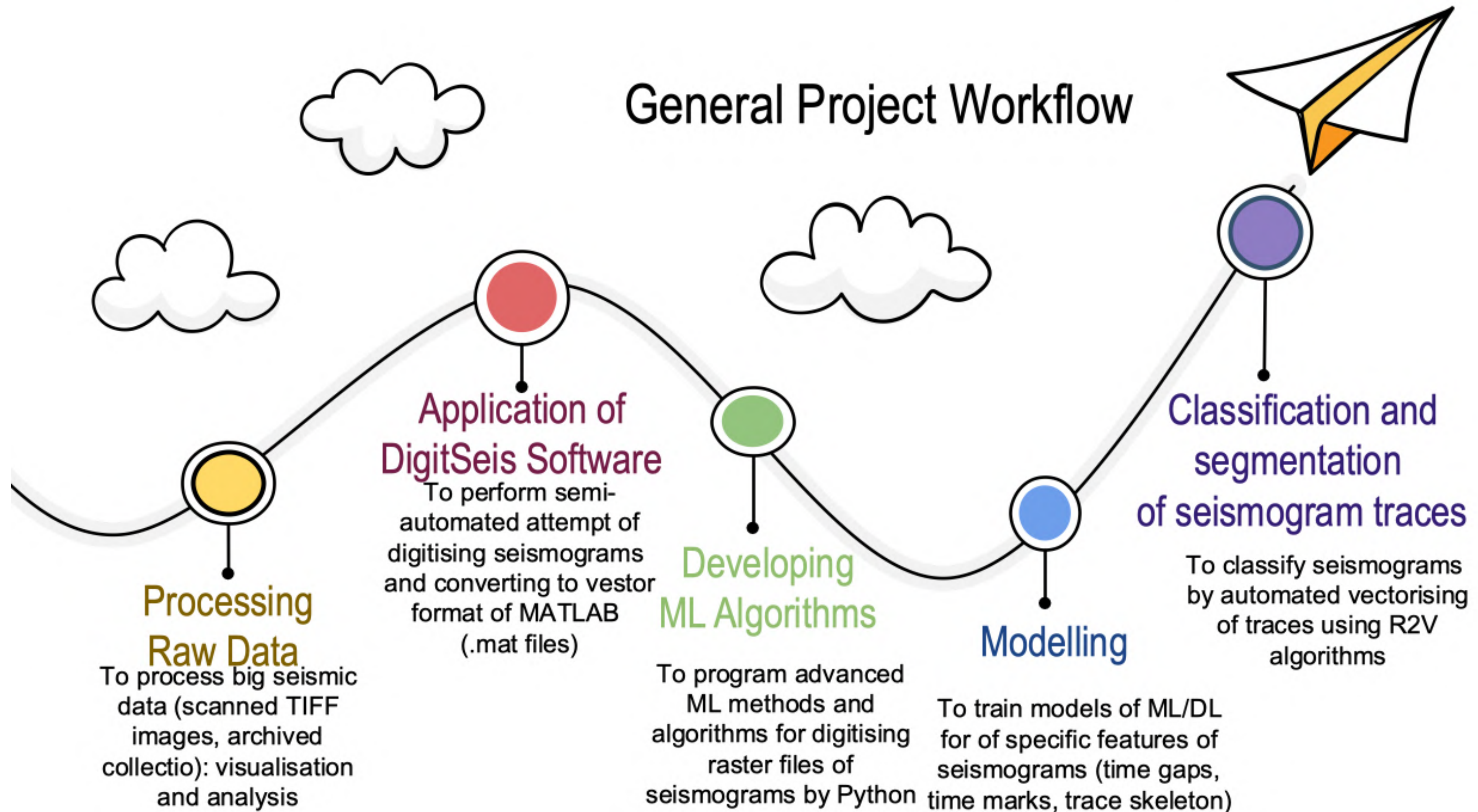
2

3

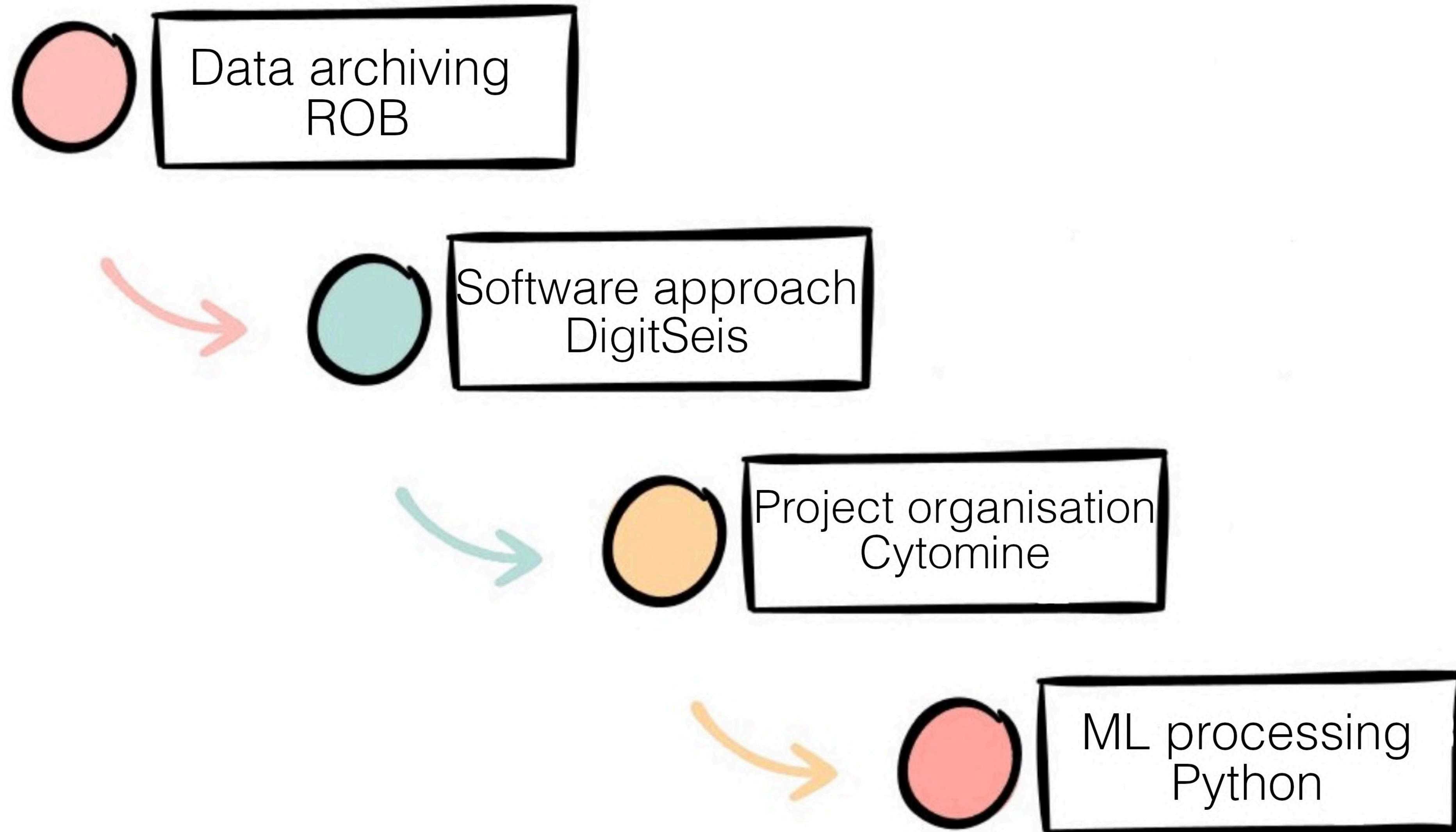
4

5

Activities Towards Achieving Project Goals



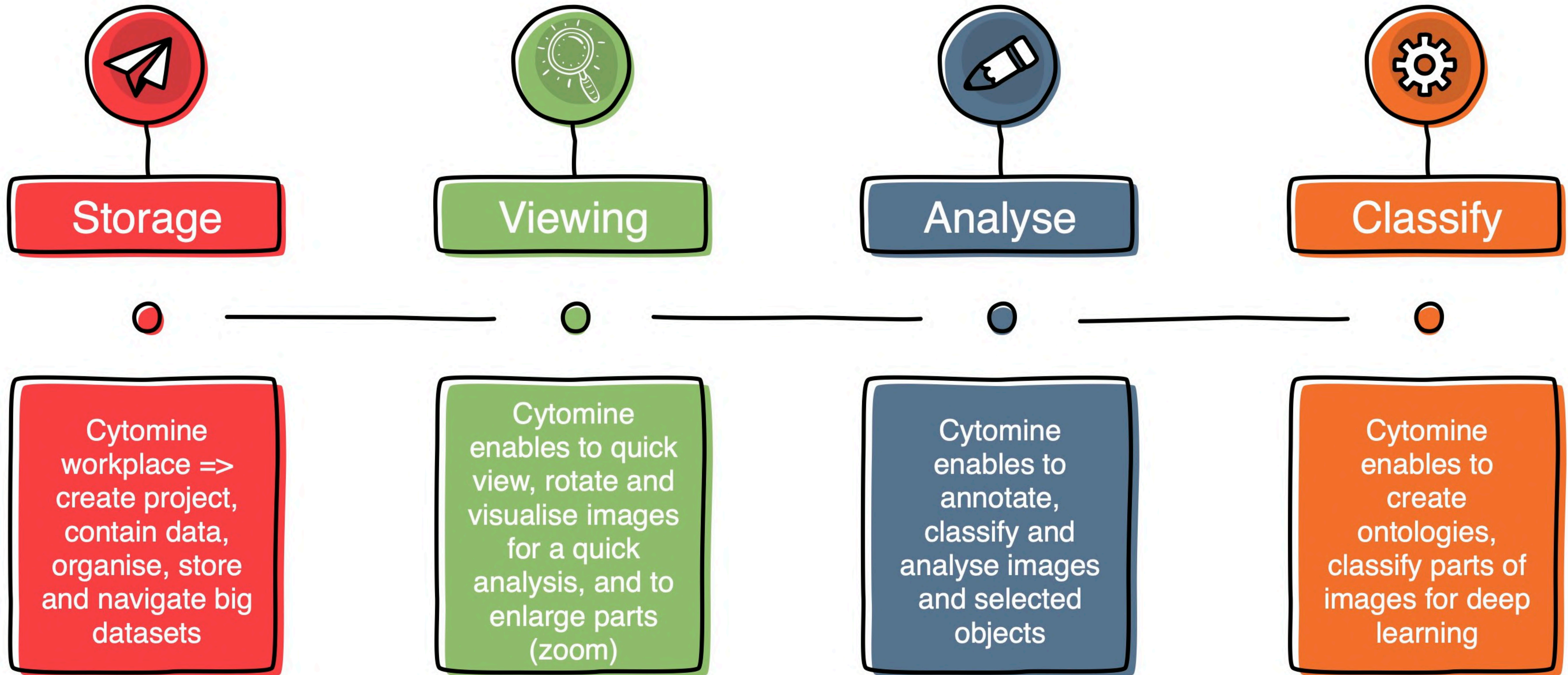
Summary of Project Milestones and Approaches



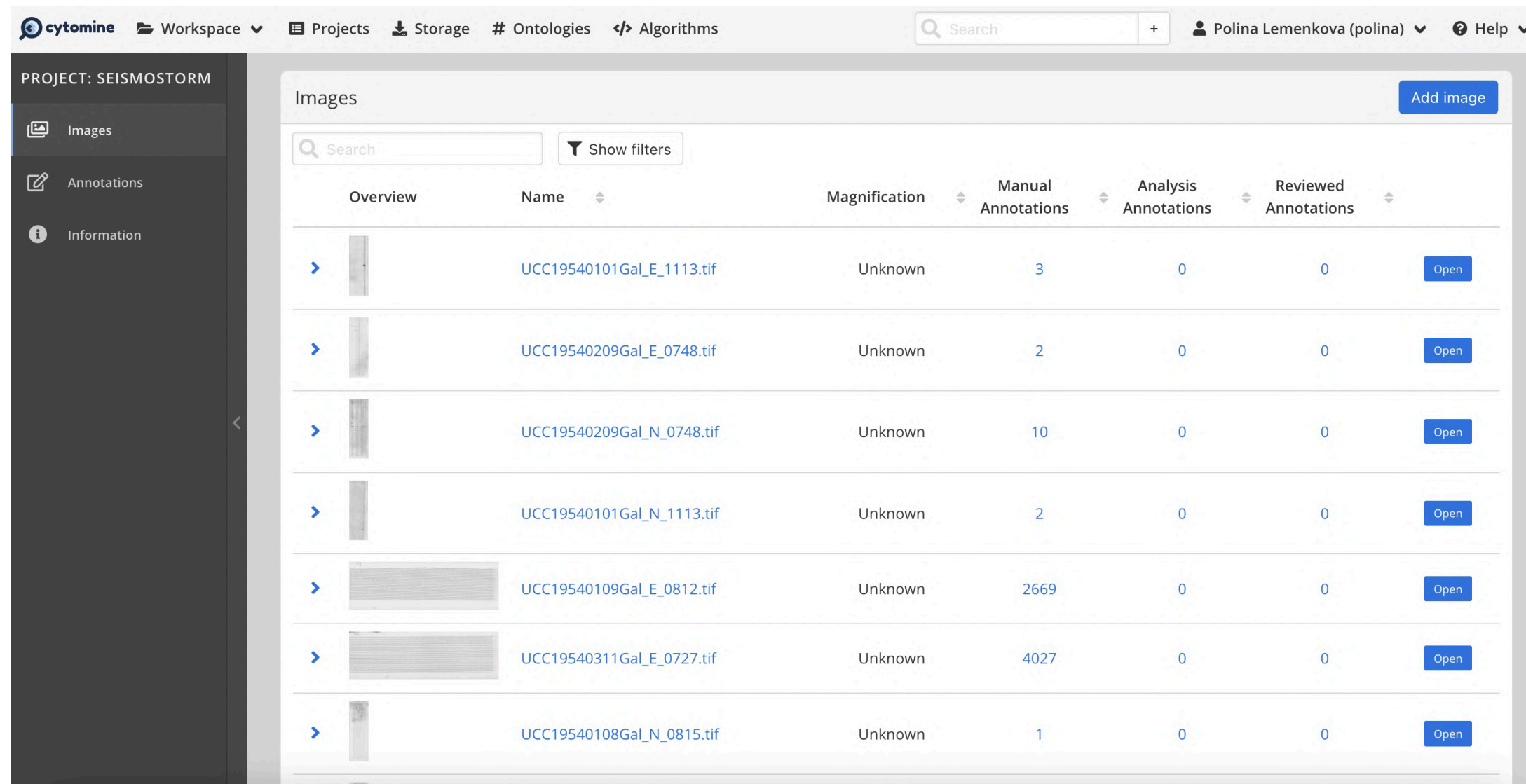
Part 2.

Data Management:
Using *Cytomine* as a Workspace
for Data Storage,
Organising, and Control

Why using *Cytomine* for Processing *Seismostorm* Project ?



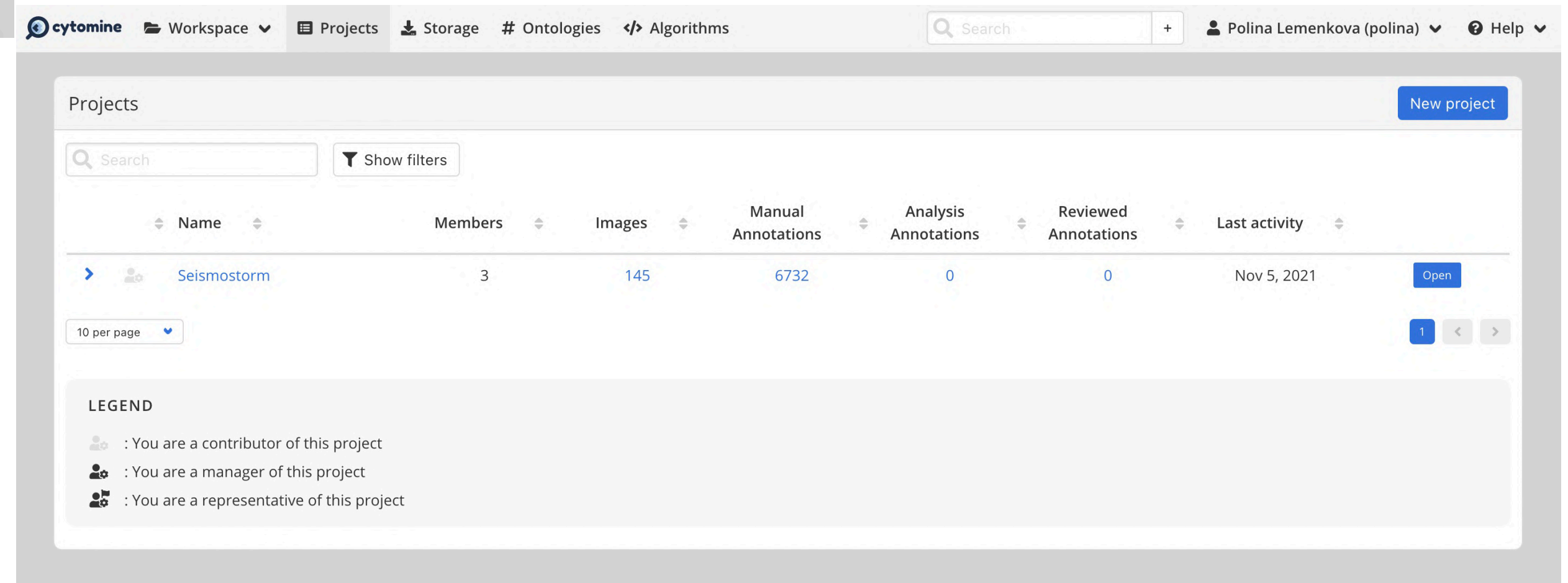
Cytomine for data storage, sharing and analysis



View of the Seismostorm project and file browsing system

- The workspace containing seismic dataset is shared by users (collaborators of Seismostorm)
- Navigating in Cytomine =>> paths and hierarchical structure of the project

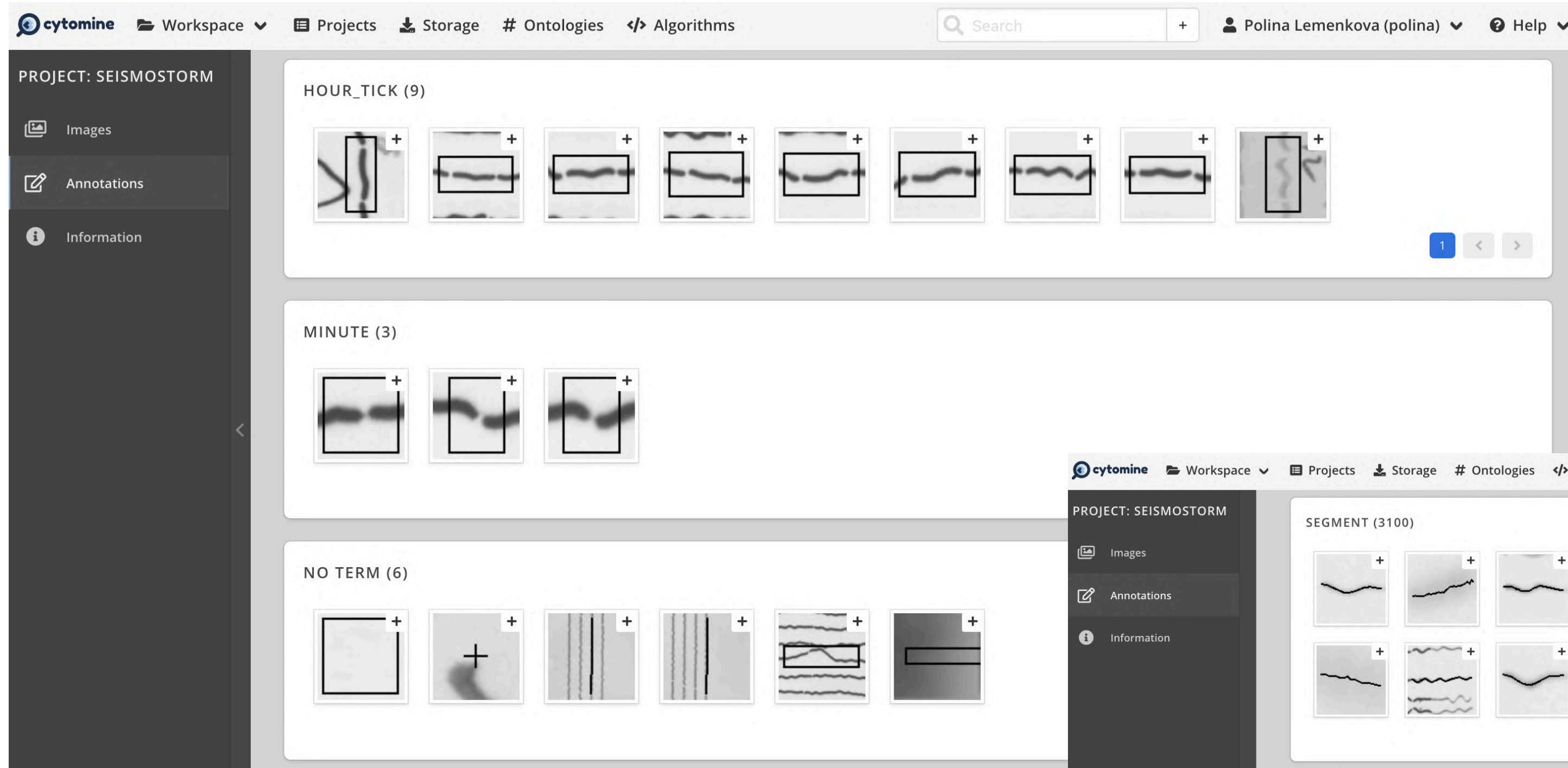
- Data were placed on the Cytomine environment (Cytomine), developed by the ULiège team.
- We uploaded our TIFF images into our project.
- Originally designed as a tool for biomedical image processing, Cytomine is adopted in this study for geophysical data processing using seismograms.
- The dataset contains 145 files recorded in 1954 by Galitzine seismometer.



Content of files in the Seismostorm project in Cytomine

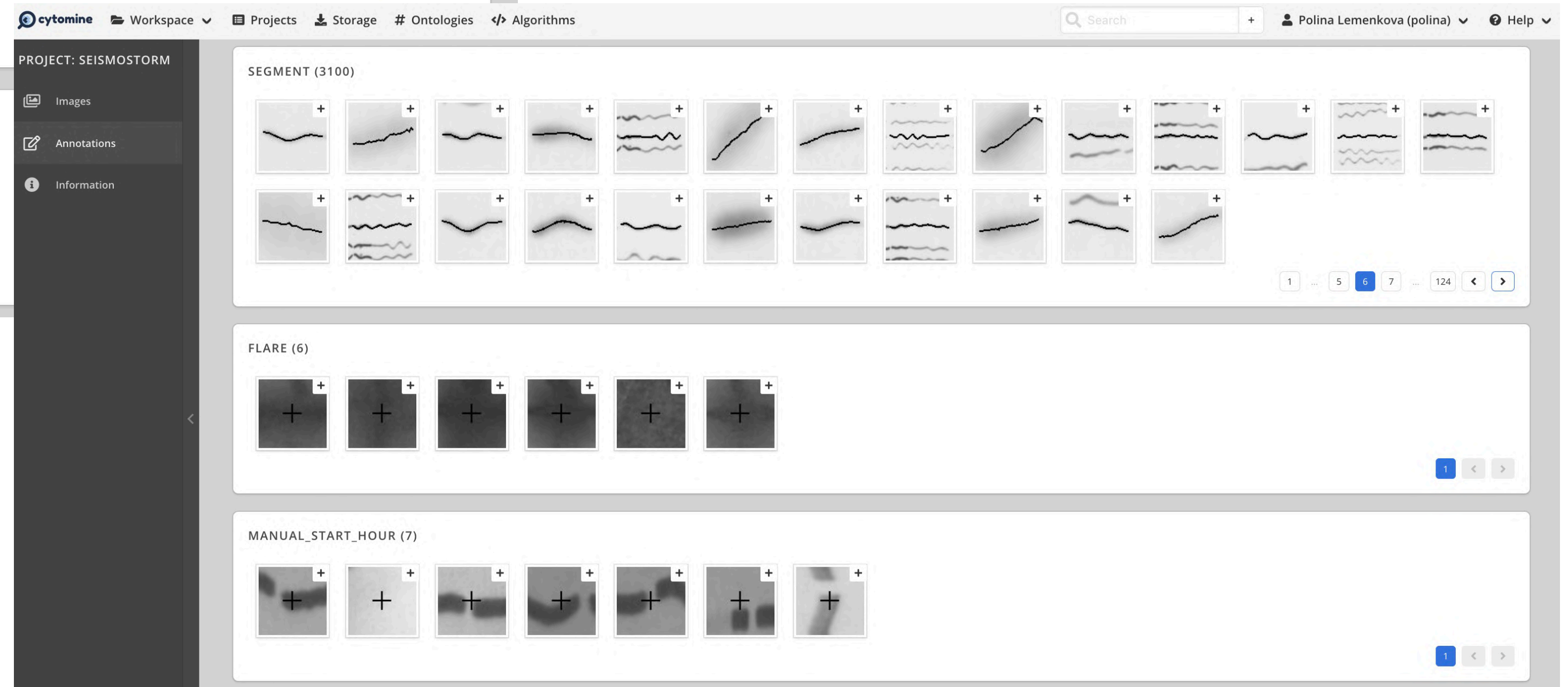
Cytomine is an image analysis workspace to contain, organise, visualise, annotate and analyse images.

Creating ontologies in *Cytomine* for objects recognition



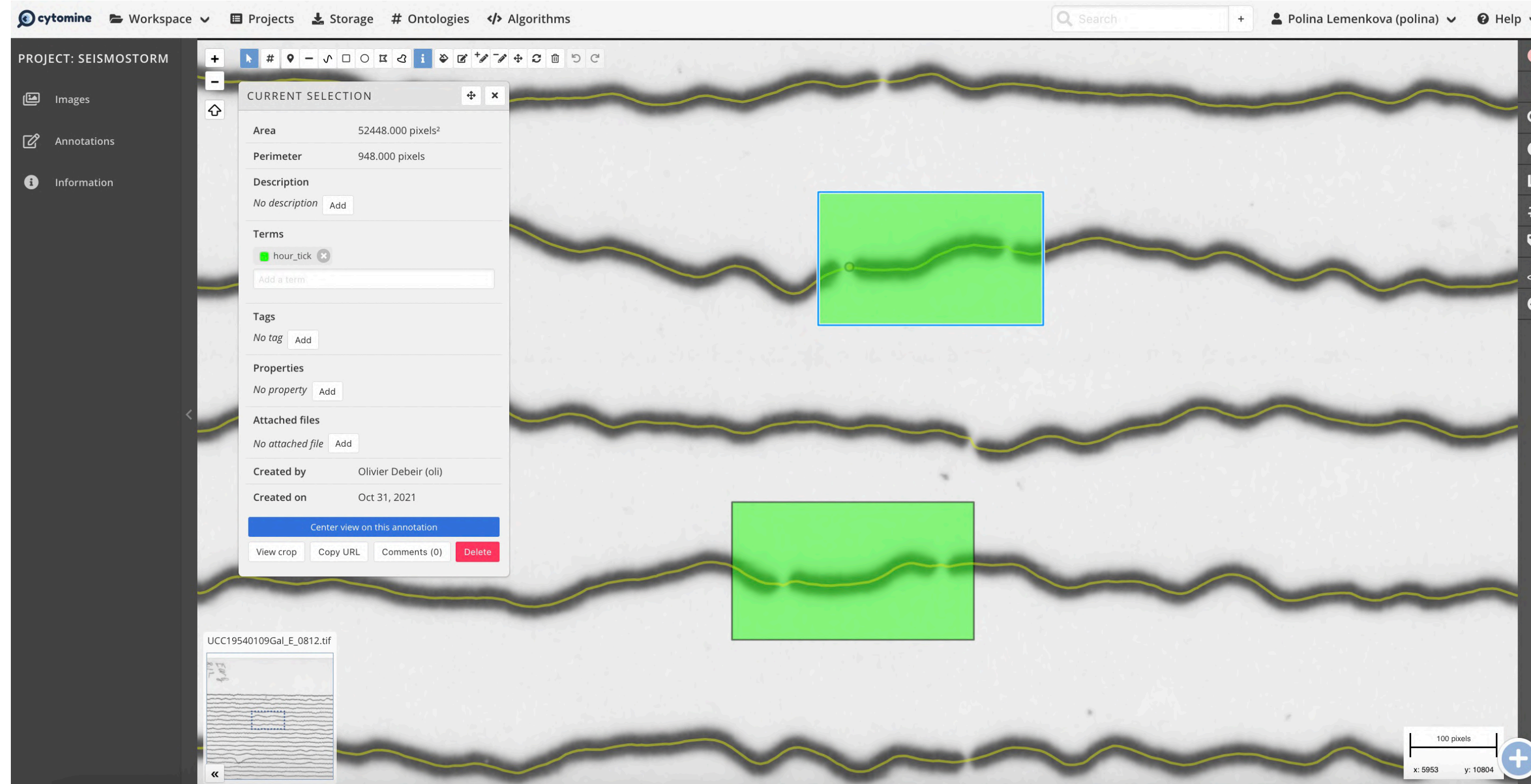
- Ontologies generated in Seismostorm project in Cytomine enable to class shapes for automated recognition
- Segments, start hours ticks and flares detected as object classes on the scanned images

Hour ticks, minute ticks and various categories detected as object classes on the images

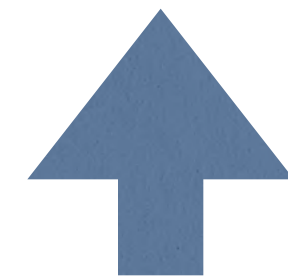
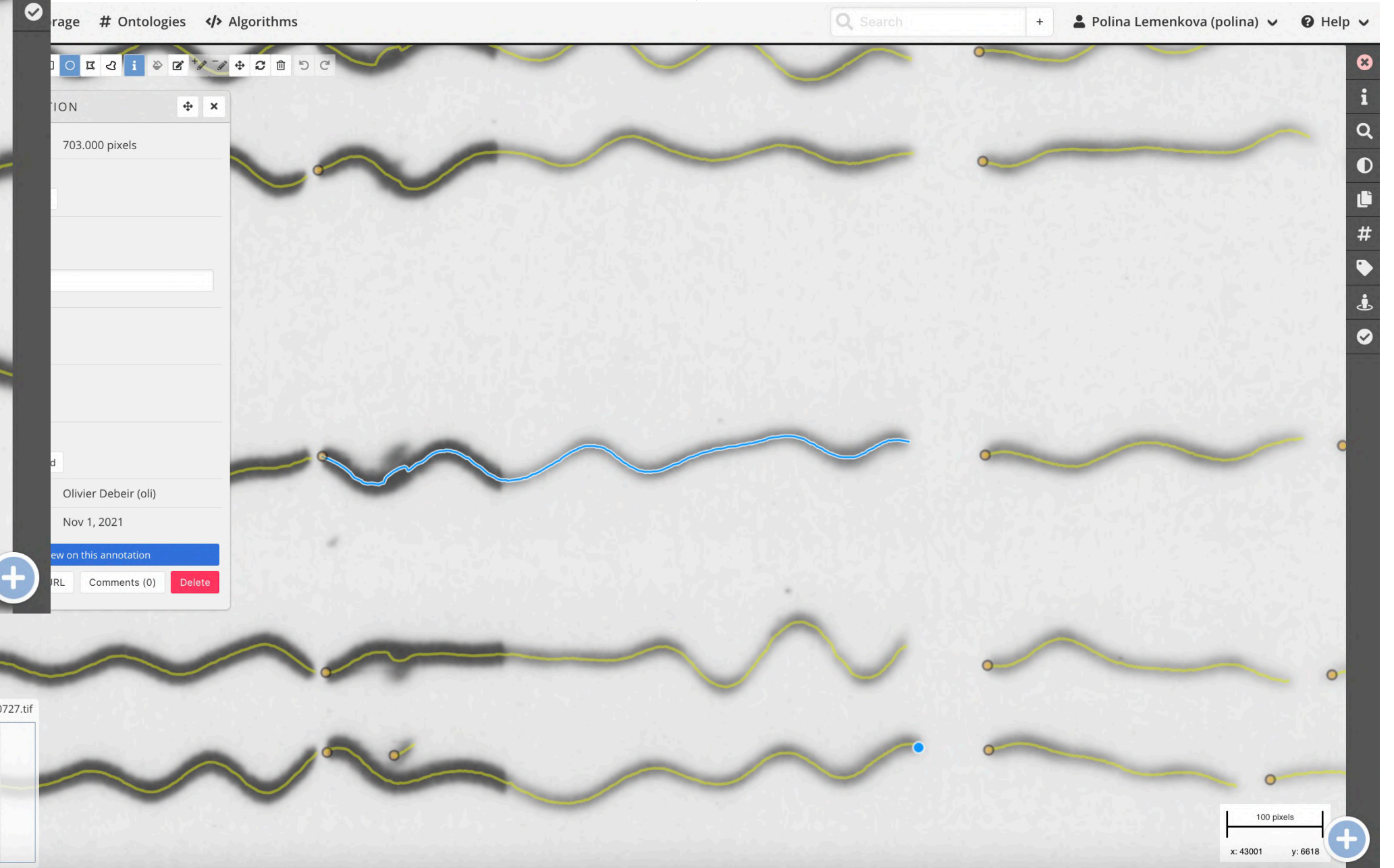
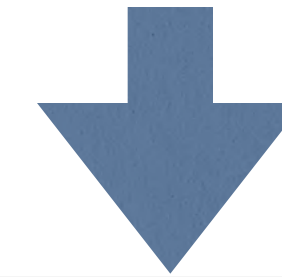


Examples of the detected and annotated object classes on the scanned seismograms

Examples of detecting cases in seismograms in *Cytomine*



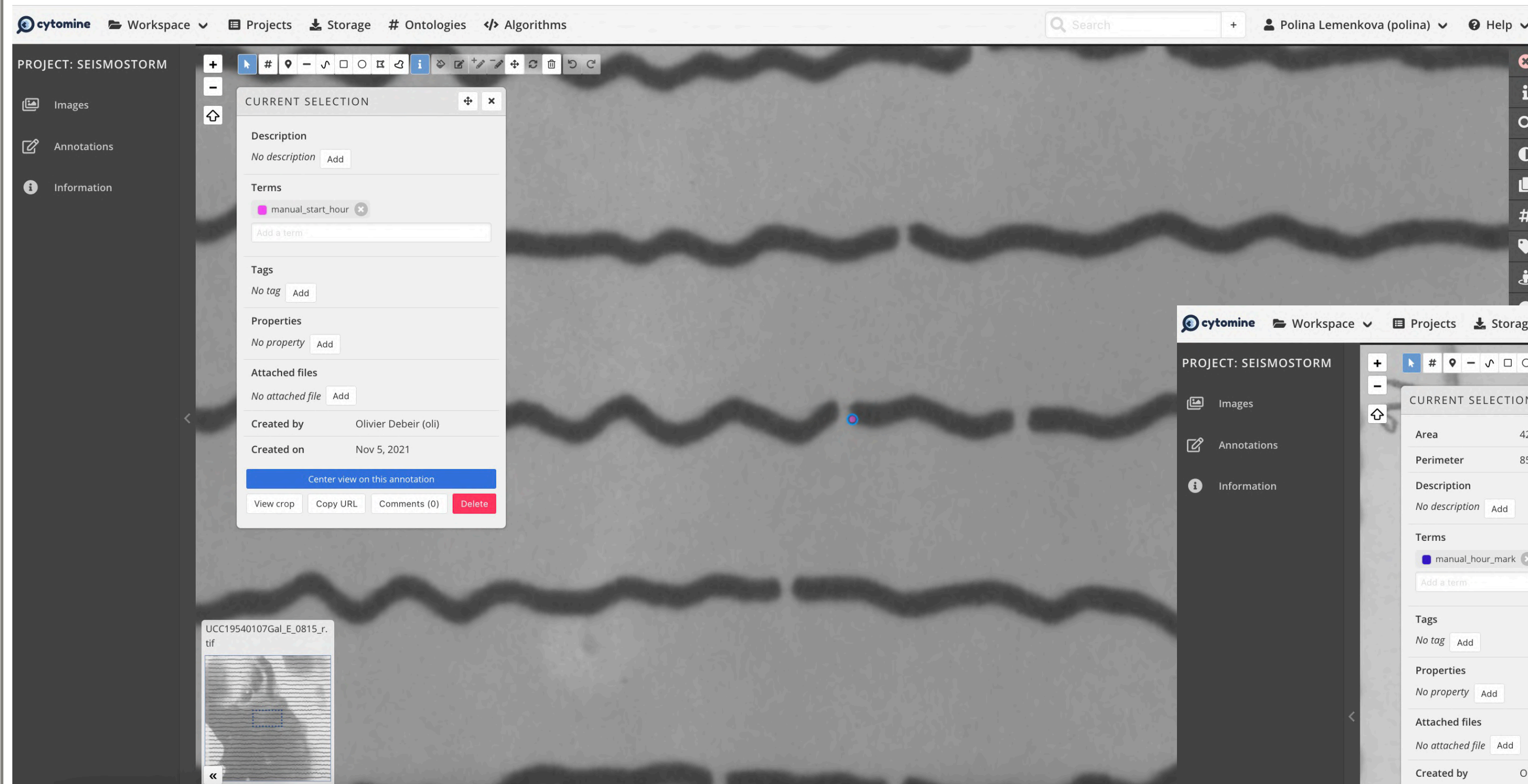
Segments separated as fragments on the trace lines



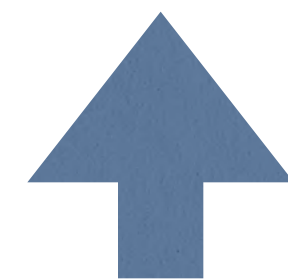
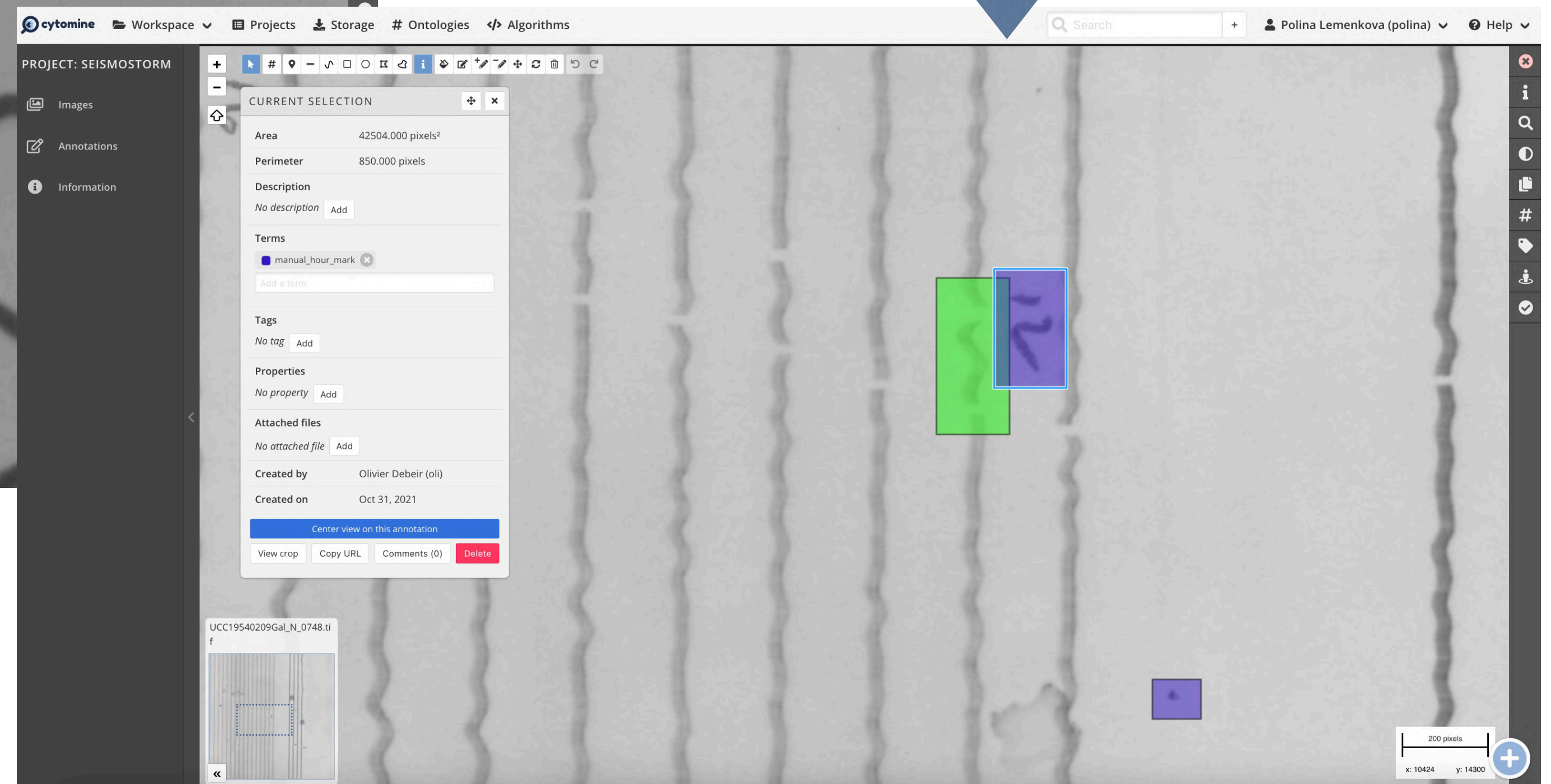
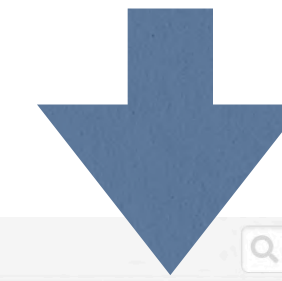
Hour ticks on the seismograms recorded by the seismometer drum

Examples of the detected and annotated object classes on the scanned seismograms

Examples of marking time gaps (minutes/hours) on seismograms in *Cytomine*



Manual hour marks for handwritten annotations on the old scanned image

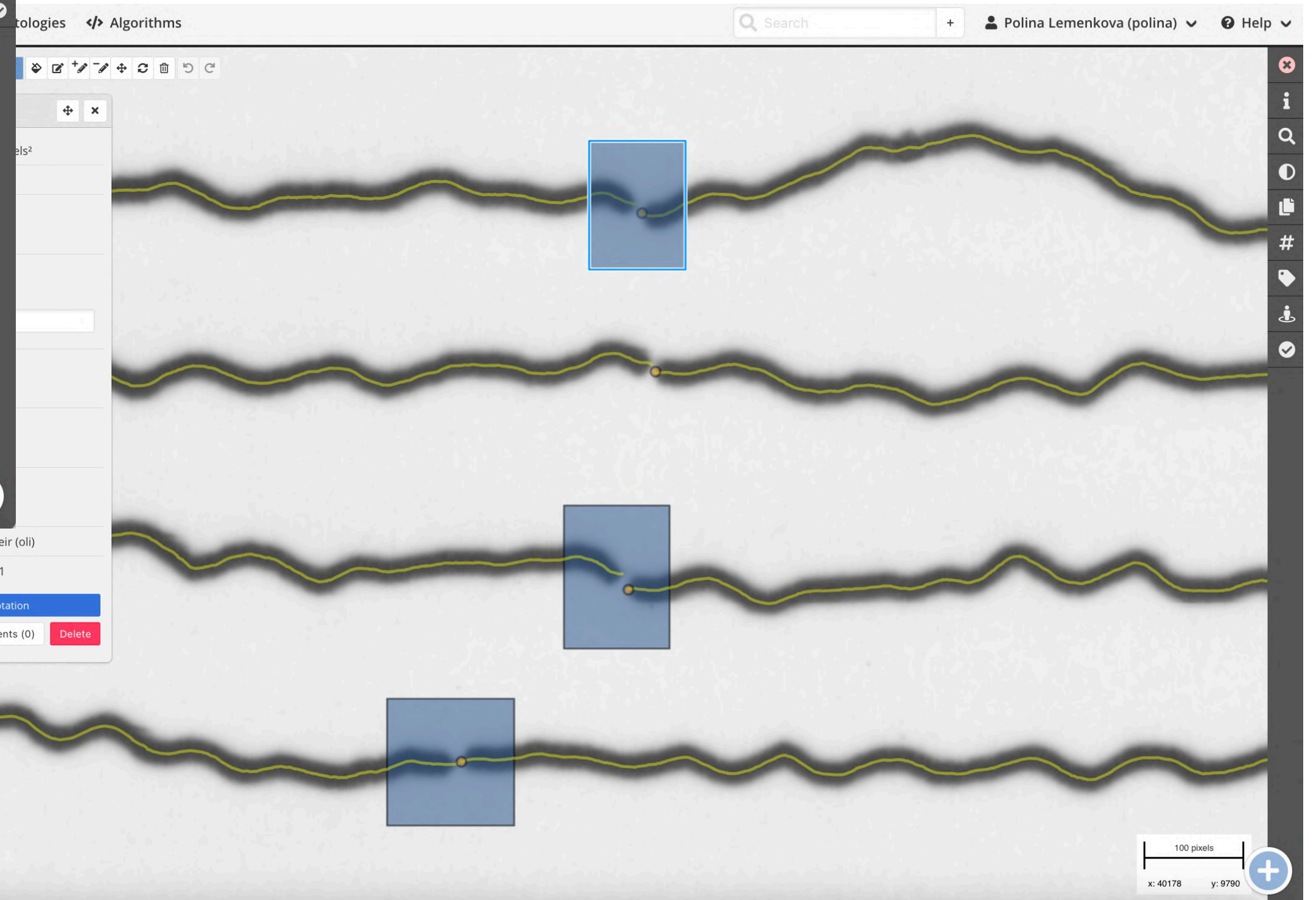
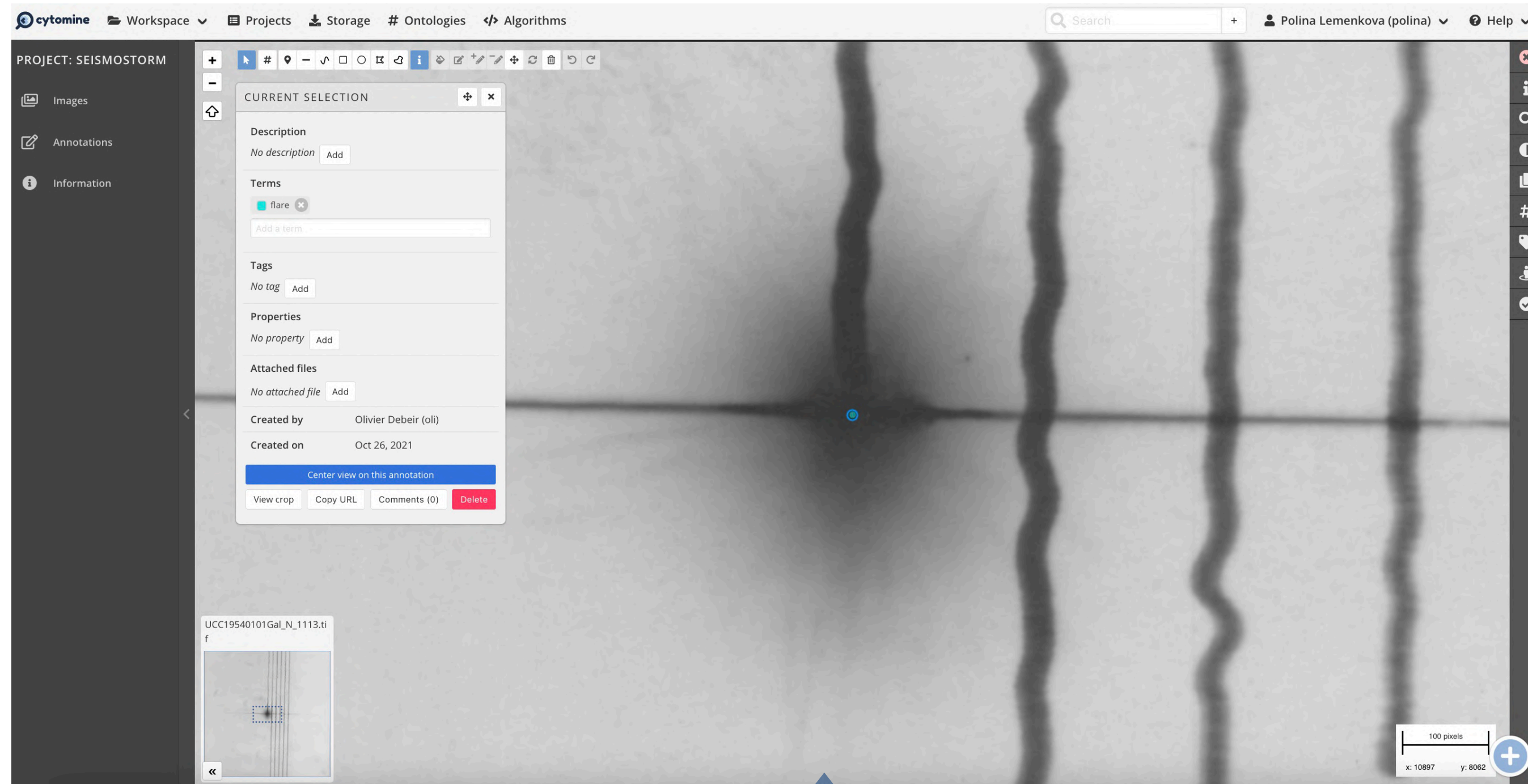
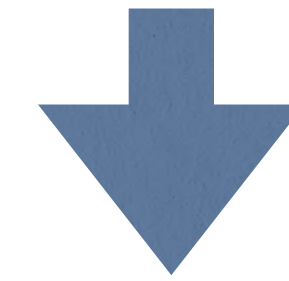


Manual ticks for the start hours on the partially spotted image

Examples of the annotation classes on the raw data: scanned analog seismograms from the Uccle station.

Examples of marking time gaps (minutes/hours) on seismograms in *Cytomine*

Minute marks detected, recognised and classified using 'ontologies' of Cytomine on the TIFF files



Flares detected on the old scanned raster images of the analog seismograms

Examples of the annotation classes on the raw data: scanned analog seismograms from the Uccle station.

Part 2.

Application of *DigitSeis* Software
for Vectorising Seismograms

Article

Computer Vision Algorithms of DigitSeis for Building a Vectorised Dataset of Historical Seismograms from the Archive of Royal Observatory of Belgium

Polina Lemenkova ^{1,*}, Raphaël De Plaen ², Thomas Lecocq ² and Olivier Debeir ¹

¹ Laboratory of Image Synthesis and Analysis (LISA), École Polytechnique de Bruxelles (Brussels Faculty of Engineering), Université Libre de Bruxelles (ULB), Building L, Campus de Solbosch, Avenue Franklin Roosevelt 50, BE-1050 Brussels, Belgium

² Royal Observatory of Belgium, Seismology & Gravimetry (OD2), Avenue Circulaire 3, BE-1180 Uccle, Belgium

* Correspondence: polina.lemenkova@ulb.be; Tel.: +32-471-86-04-59

Abstract: Archived seismograms recorded in the 20th century present a valuable source of information for monitoring earthquake activity. However, old data, which are only available as scanned paper-based images should be digitised and converted from raster to vector format prior to reuse for geophysical modelling. Seismograms have special characteristics and specific features recorded by a seismometer and encrypted in the images: signal trace lines, minute time gaps, timing and wave amplitudes. This information should be recognised and interpreted automatically when processing archives of seismograms containing large collections of data. The objective was to automatically digitise historical seismograms obtained from the archives of the Royal Observatory of Belgium (ROB). The images were originally recorded by the Galitzine seismometer in 1954 in Uccle seismic station, Belgium. A dataset included 145 TIFF images which required automatic approach of data processing. Software for digitising seismograms are limited and many have disadvantages. We applied the DigitSeis for machine-based vectorisation and reported here a full workflow of data processing. This included pattern recognition, classification, digitising, corrections and converting TIFFs to the digital vector format. The generated contours of signals were presented as time series and converted into digital format (mat files) which indicated information on ground motion signals contained in analog seismograms. We performed the quality control of the digitised traces in Python to evaluate the discriminating functionality of seismic signals by DigitSeis. We shown a robust approach of DigitSeis as a powerful toolset for processing analog seismic signals. The graphical visualisation of signal traces and analysis of the performed vectorisation results shown that the algorithms of data processing performed accurately and can be recommended in similar applications of seismic signal processing in future related works in geophysical research.

Keywords: seismology; Galitzine seismometer; horizontal component; analogue seismogram; digitising; earthquake recording; ground motions; historical seismograms; seismic waves

1. Introduction

1.1. Background

The seismicity of the Earth represents a physical phenomenon resulting from the tectonic processes of energy accumulation and release [1]. This fundamental concept of the Earth's physics is reflected in the movements on the surface that have a different intensity of the vibrations caused by the fluctuations of energy [2]. In geodynamics, the seismicity is the result of the self-organisation of the Earth which responds to the lithosphere movements [3]. Measuring seismic signals using seismometers enables to evaluate ground motions of the Earth associated with the earthquakes of various magnitude or ambient seismic noise [4]. Evaluating seismic ambient noise is effective in different contexts, including imaging and monitoring the Earth's interior, seismic mapping at a global or regional scale, environmental



Citation: Lemenkova, P.; De Plaen, R.; Lecocq, T.; Debeir, O. Computer Vision Algorithms of DigitSeis for Building a Vectorised Dataset of Historical Seismograms from the Archive of Royal Observatory of Belgium. *Sensors* **2023**, *23*, 56.

<https://doi.org/10.3390/s23010056>

Academic Editor: Marcin Woźniak

Received: 17 October 2022

Revised: 4 December 2022

Accepted: 15 December 2022

Published: 21 December 2022

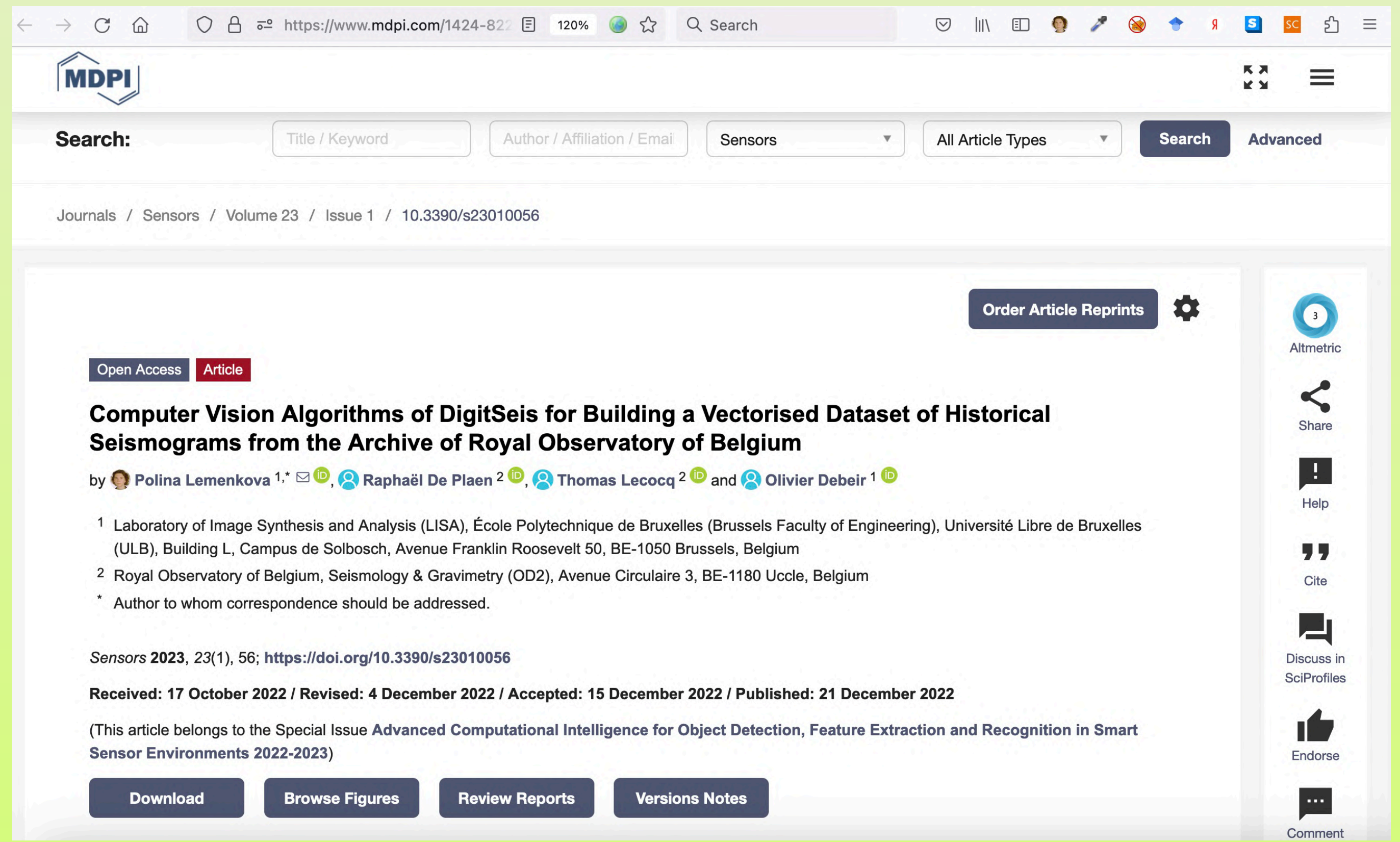


Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Publication results of the Part 2 of the PhD project:

Lemenkova, P.; De Plaen, R.; Lecocq, T.; Debeir, O. Computer Vision Algorithms of DigitSeis for Building a Vectorised Dataset of Historical Seismograms from the Archive of Royal Observatory of Belgium. *Sensors* **2023**, *23*, 56. <https://doi.org/10.3390/s23010056>

Journal metrics: *Scopus*, *WoS*, Journal Ranking in JCR: Q2 (Instruments & Instrumentation) / CiteScore - Q1 (Instrumentation)



The screenshot shows the MDPI article page. At the top, there is a search bar and navigation options. The article title is prominently displayed, followed by the authors' names and ORCID icons. Below the authors, there are buttons for 'Open Access' and 'Article'. The abstract and keywords are visible. On the right side, there are social media sharing icons (Altmeter, Share, Help, Cite, Discuss in SciProfiles, Endorse, Comment) and a button for 'Order Article Reprints'. At the bottom, there are buttons for 'Download', 'Browse Figures', 'Review Reports', and 'Versions Notes'. The page also displays the journal information: *Sensors* **2023**, *23*(1), 56; <https://doi.org/10.3390/s23010056>. The publication dates are: Received: 17 October 2022 / Revised: 4 December 2022 / Accepted: 15 December 2022 / Published: 21 December 2022. A note at the bottom states: (This article belongs to the Special Issue *Advanced Computational Intelligence for Object Detection, Feature Extraction and Recognition in Smart Sensor Environments 2022-2023*).

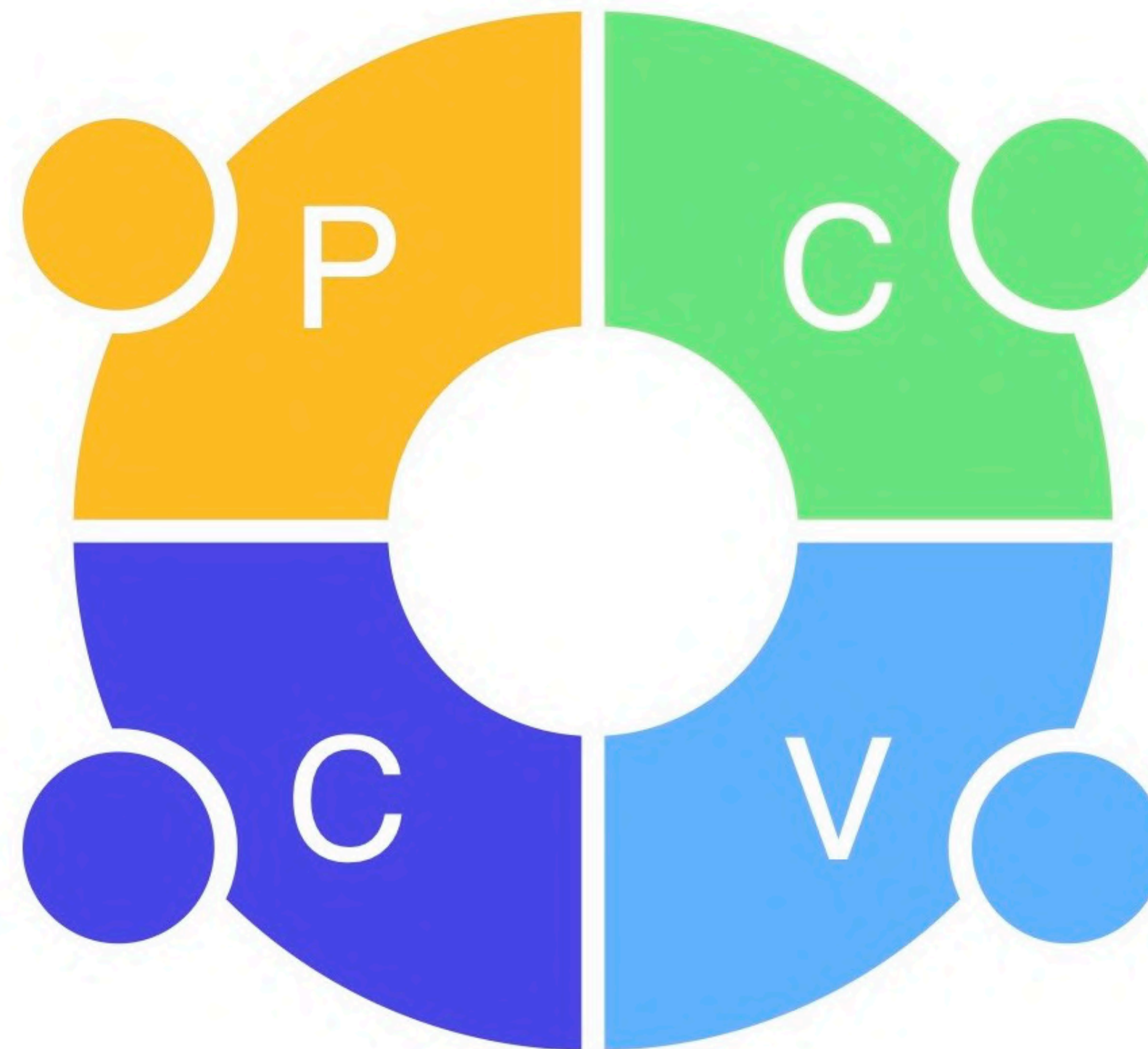
Research Approach of DigitSeis: Major Steps of Seismic Data Processing

Pre-processing

Image loading (export TIFF files), Identifying time gaps (1 minute breaks of trace lines), often 12-20 pixels.

Correcting

Manual corrections of the misclassified segments by enlarging small sections. Converting to .mat format.



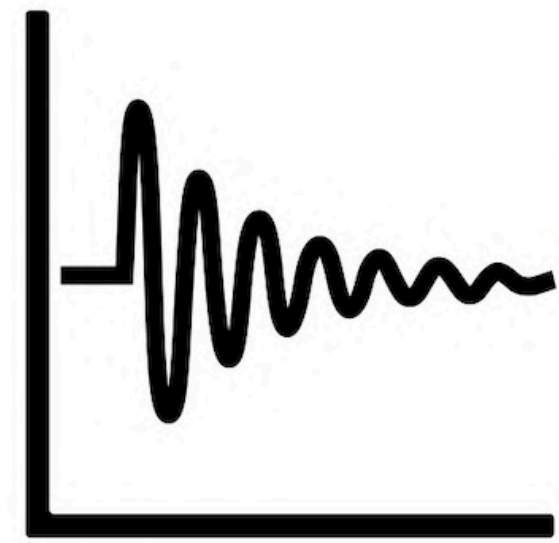
Classification

Discriminating traces from time gaps and noise (handwritten annotations), object recognition by image analysis (thresholds).

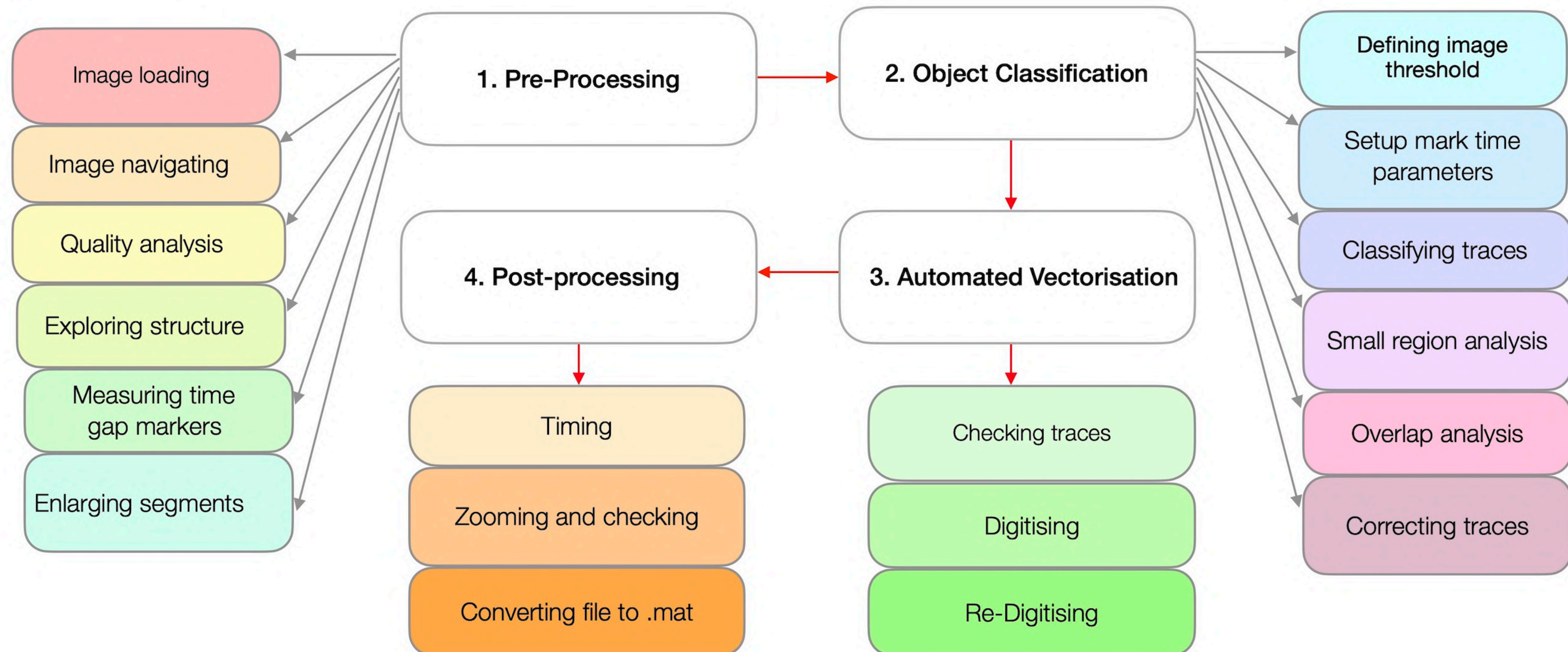
Vectorisation

Automatic discrimination of traces, noise, time gaps and background (algorithm of golden-section search).

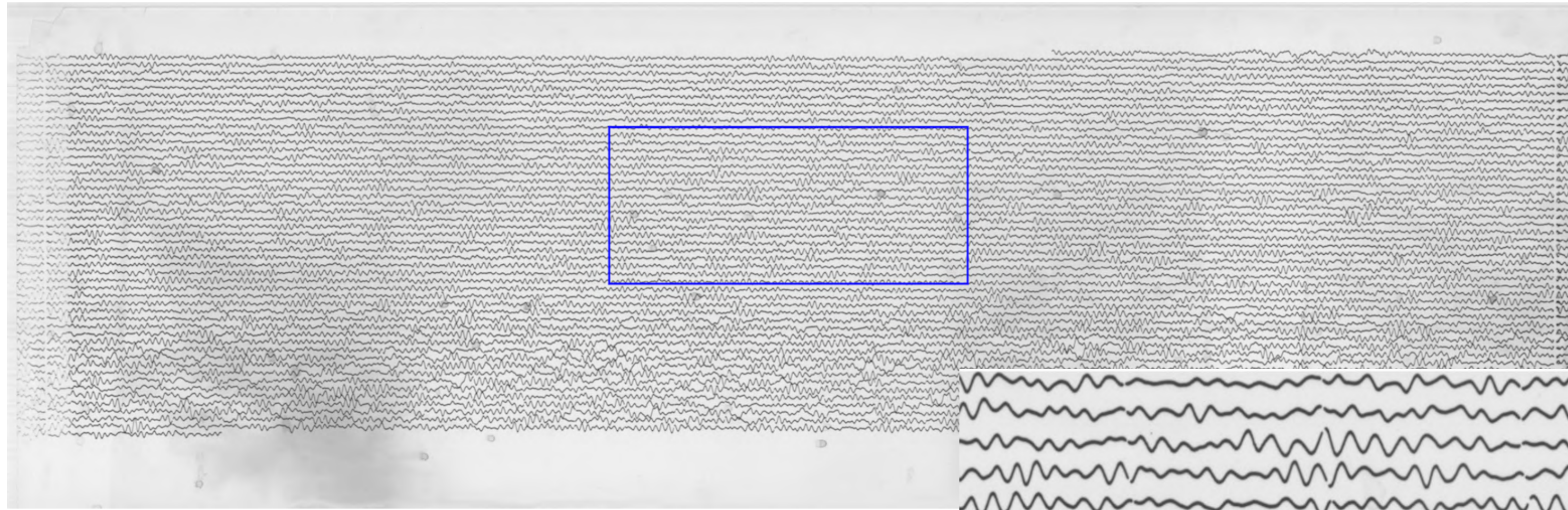
Workflow of *DigitSeis* Software for Vectorising



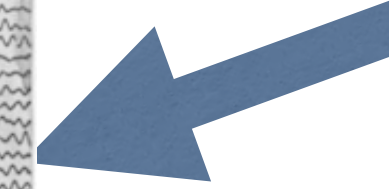
Workflow for processing seismograms in DigitSeis software



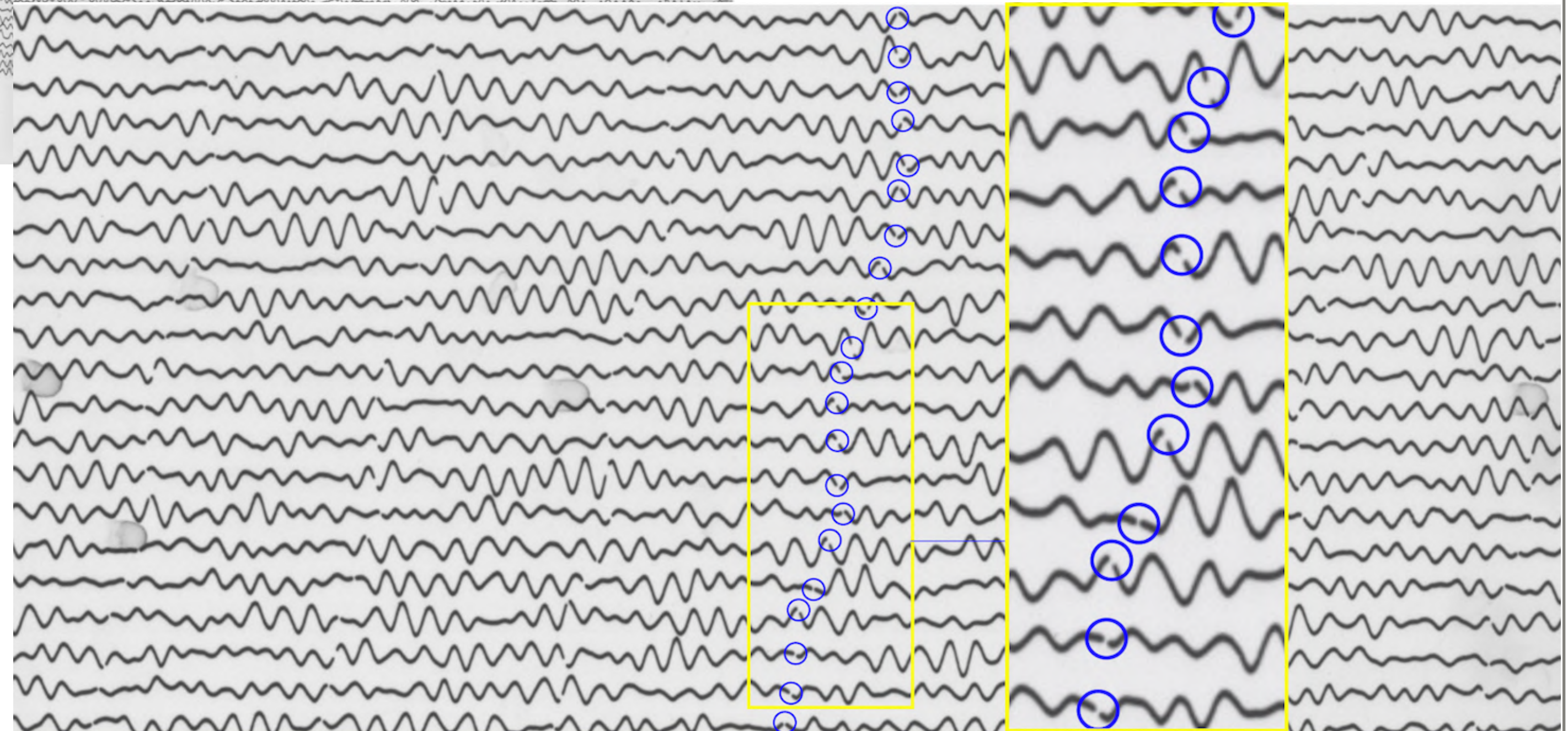
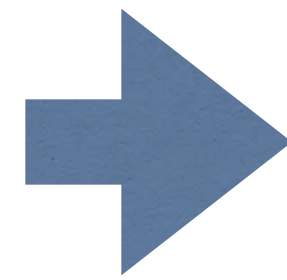
Examples of the identified time gaps on the raw TIFF images



Original scanned seismogram
(UCC19540116Gal_E_0820.tif)



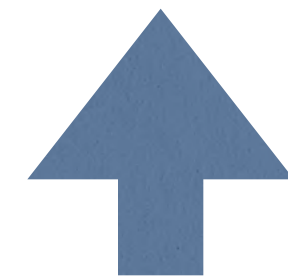
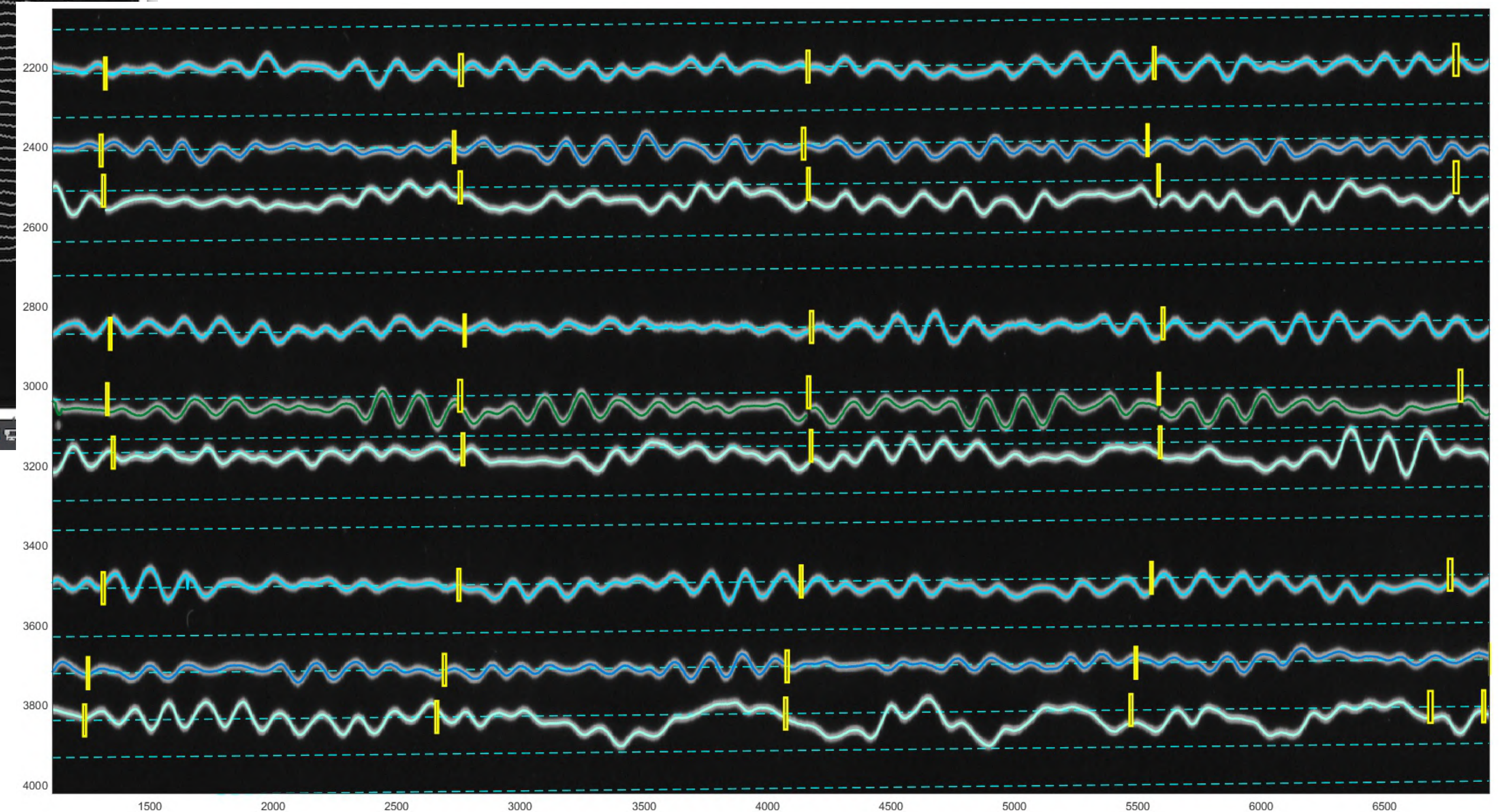
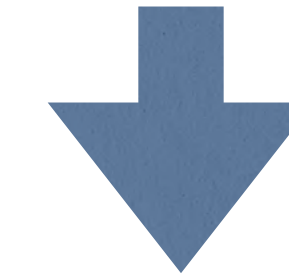
- Enlarged fragment of image
- Time gaps indicating minutes breaking the trace line
- Zoomed segment separating the trace line between each other (tiny white gaps breaking traces)



Examples of marking time gaps (minutes/hours) on seismograms in *Cytomine*



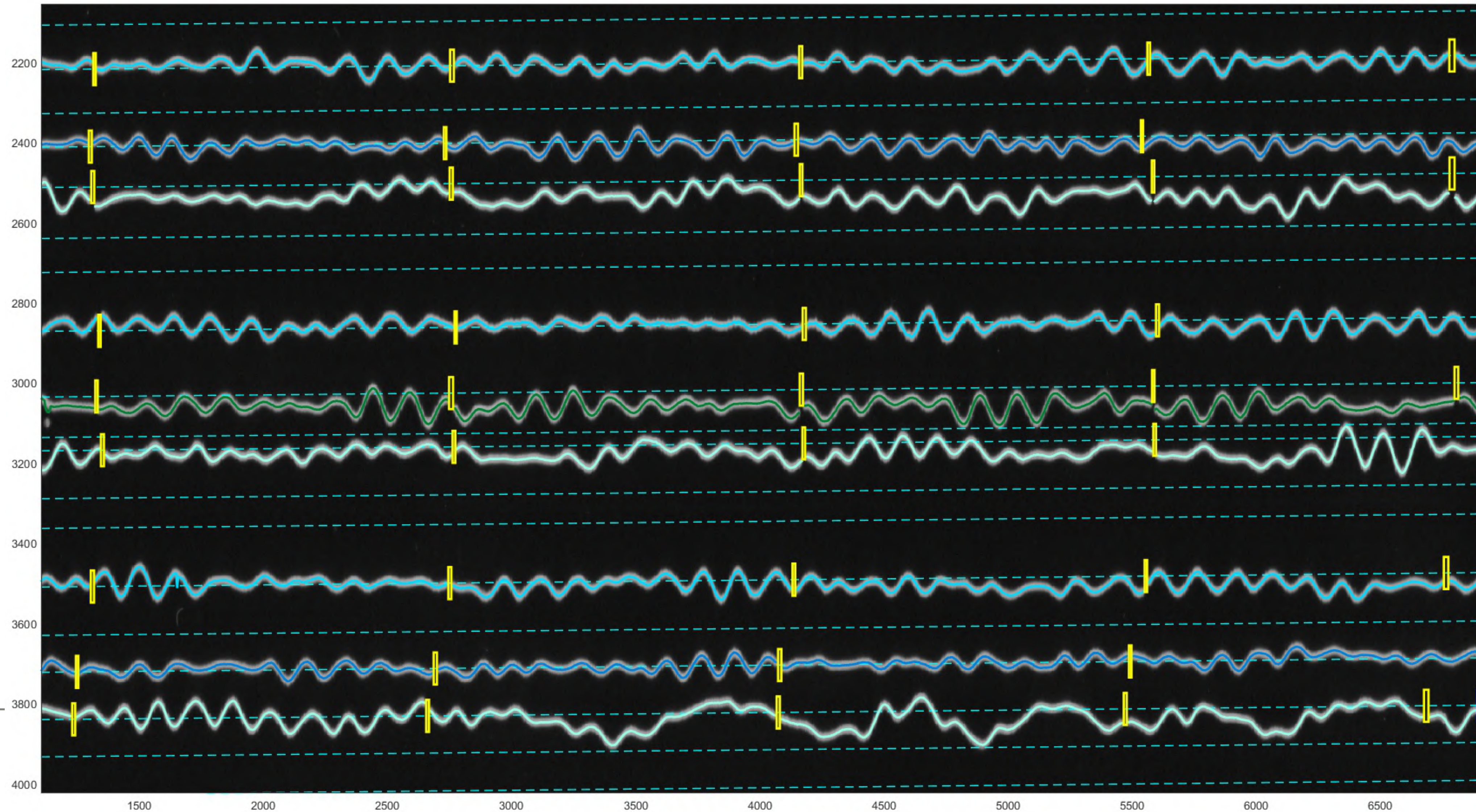
Fragment of the resulting digitised output (enlarged) showing seismic traces (horizontal curve lines) and 1-minute time gaps (small vertical dashed lines)



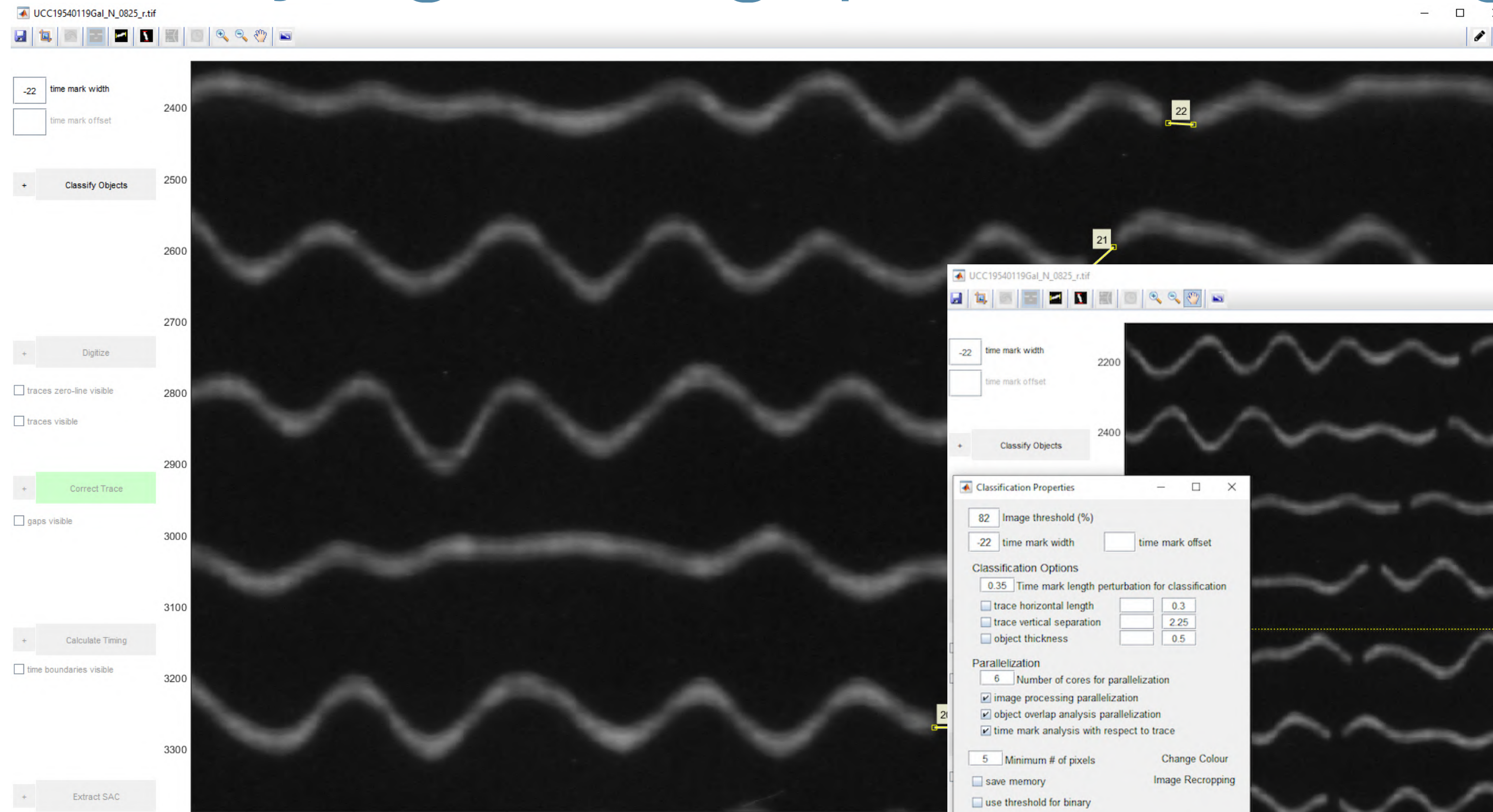
Seismogram processed by DigitSeis

Examples of the annotation classes on the raw data: scanned analog seismograms from the Uccle station.

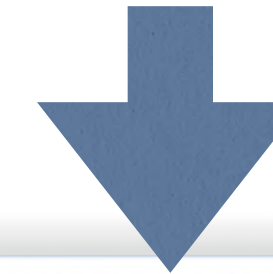
Example of the digitised image with minute time gaps
Here: fragment of UCC19540311Gal_E_0727.mat)



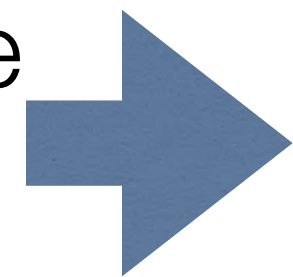
Identifying time gaps on seismogram using *DigitSeis*



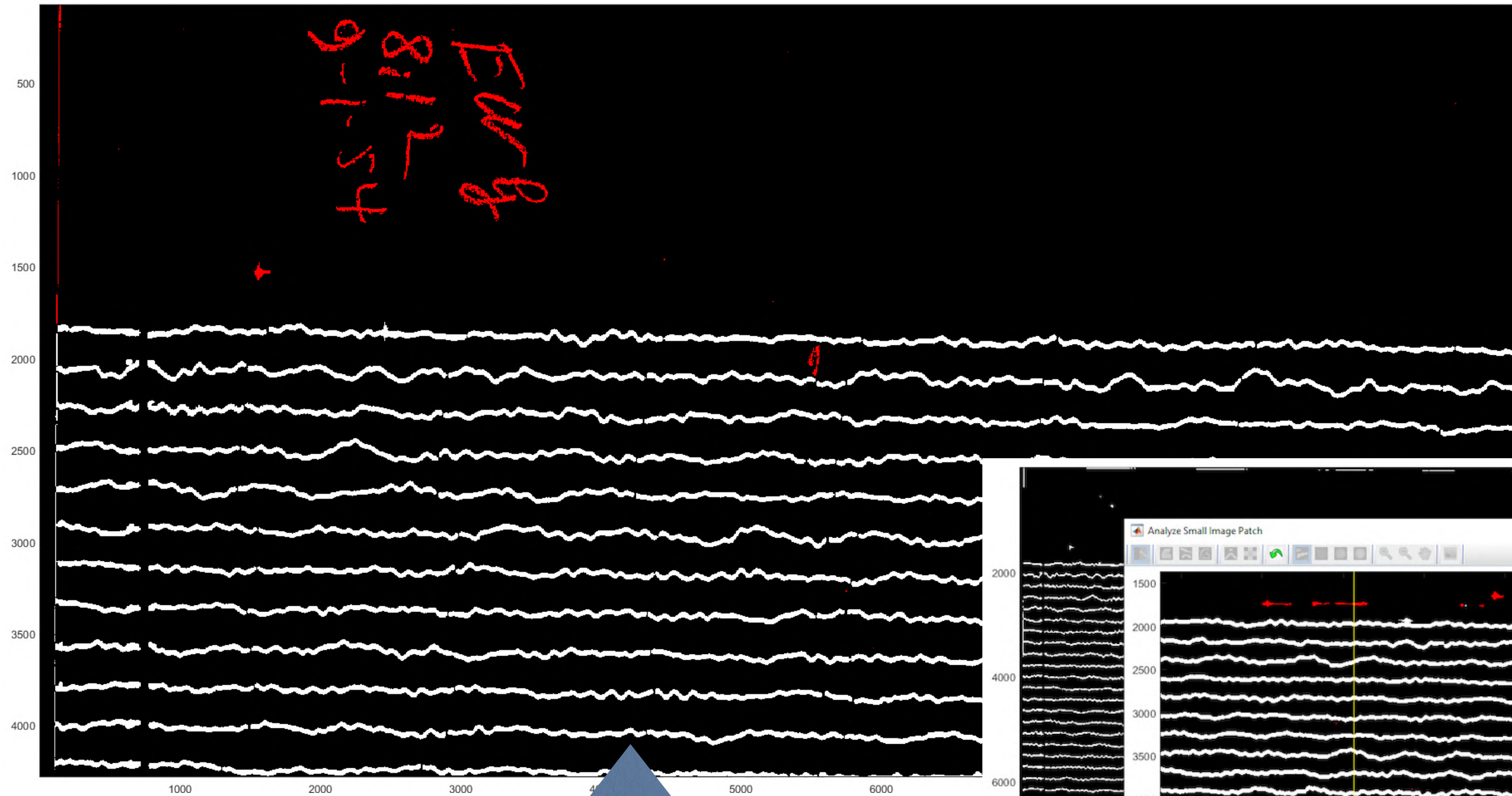
Indicating time marks on seismograms as -22 and preparing image for classification



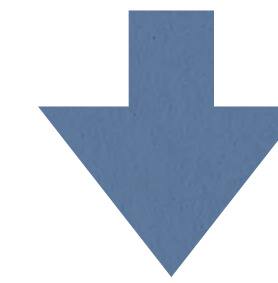
- Identifying time marks on seismograms by measuring time gap between records. Here: UCC19540119Gal_N_0825.tif



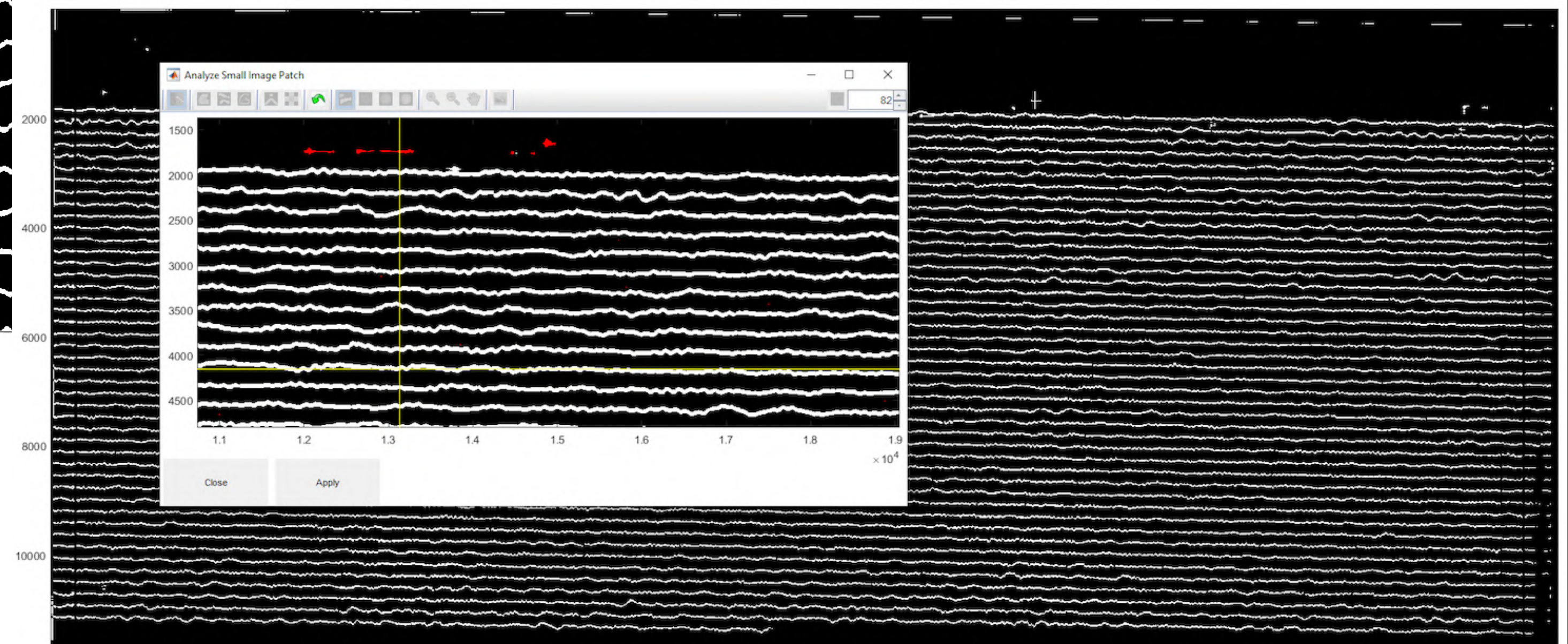
Identifying noise and annotations on seismogram using *DigitSeis*



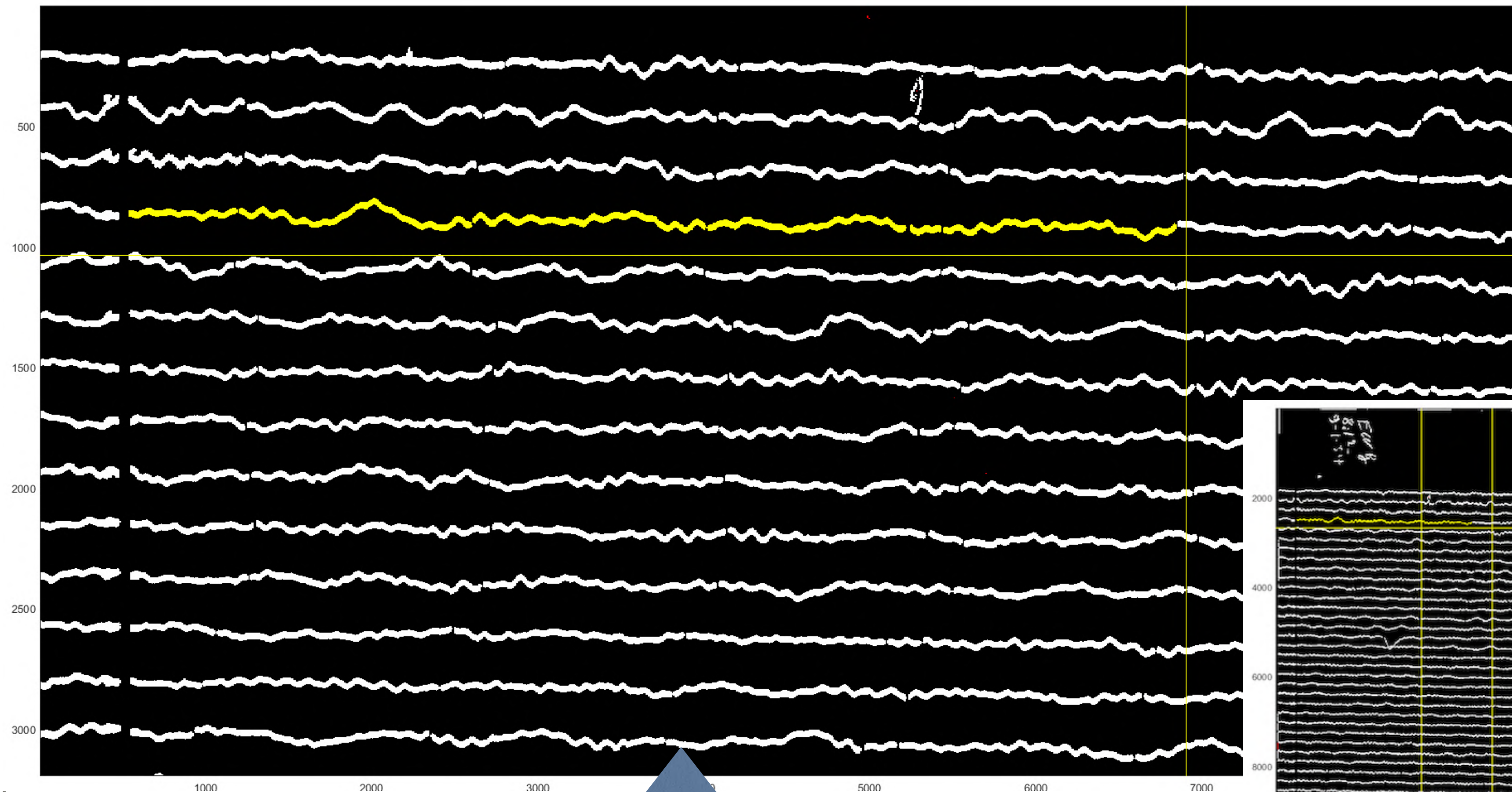
Small region analysis used for defining a smaller area of interest for closer examination of a border region of the seismogram



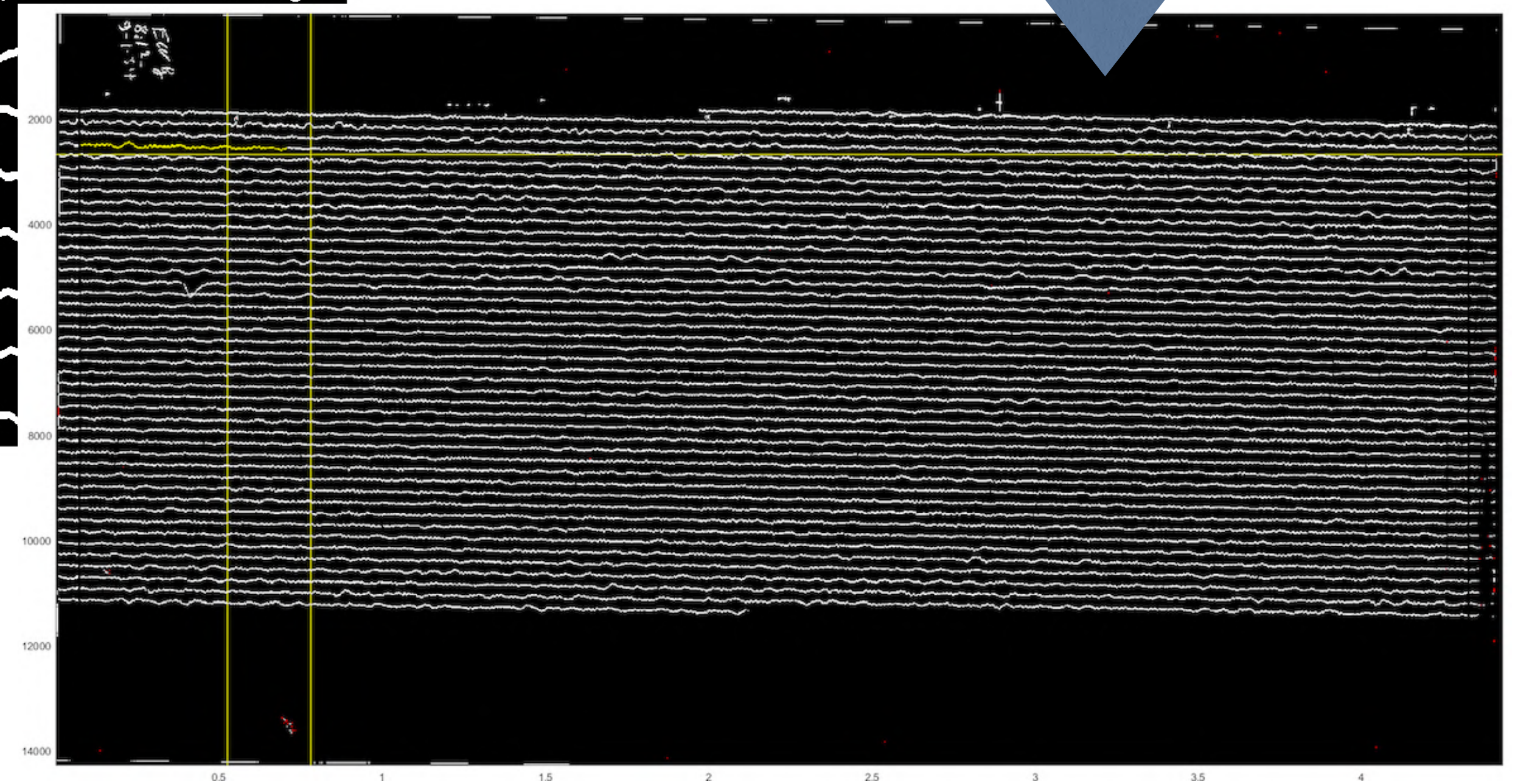
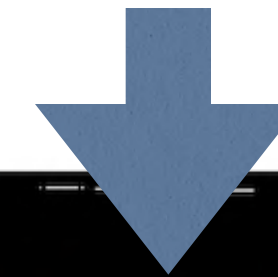
- Results of the classified seismogram with shown identified object categories.
- Traces are vector white lines while noise is red-coloured objects, automatically recognised (here: handwritten annotations)



Digitised segments of the trace lines in *DigitSeis*



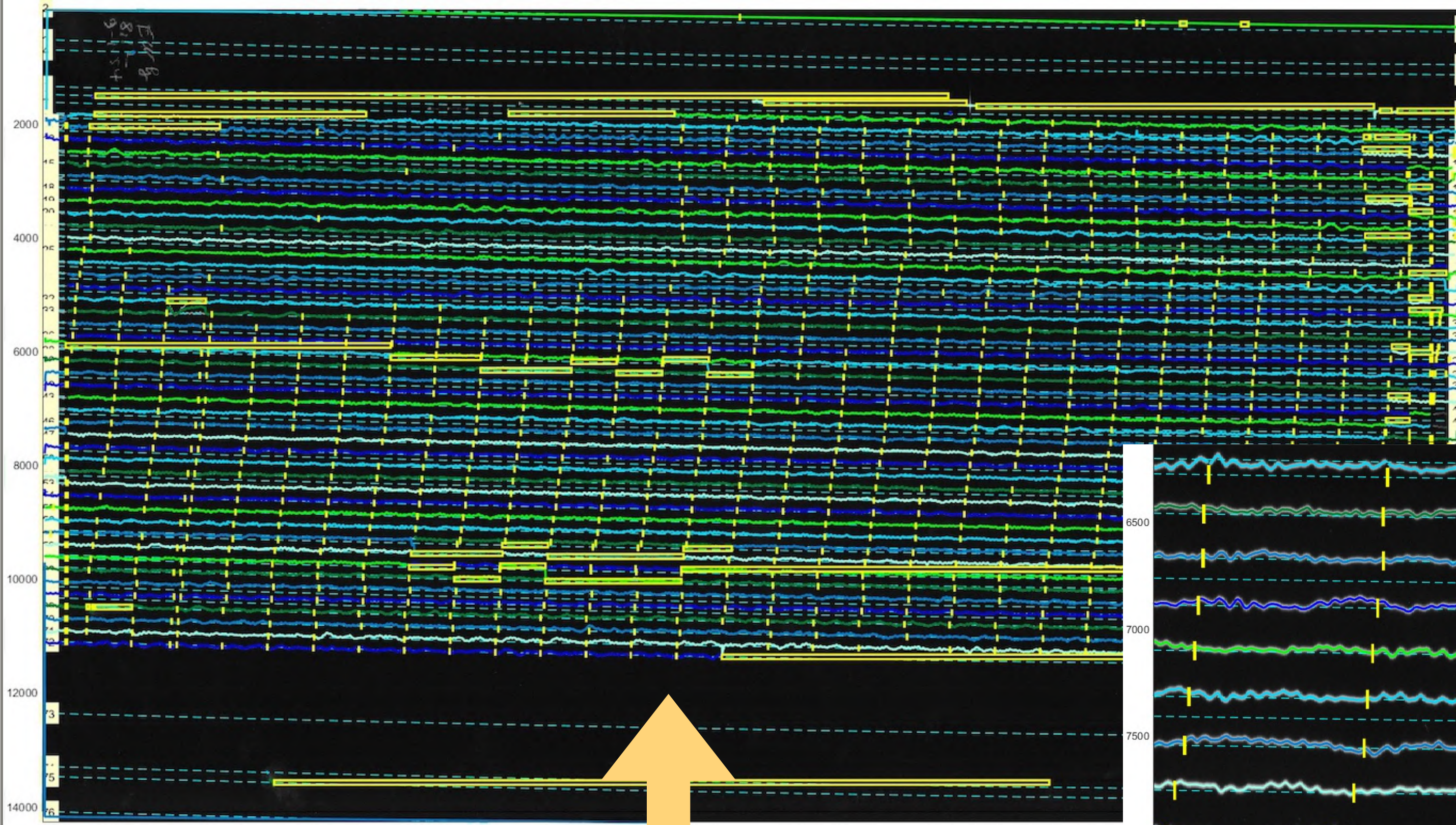
- Classified seismogram with traces saved in binary format 0-1.
- Here: example of file UCC19540109Gal_E_0812.tif (January 9, 1954.)



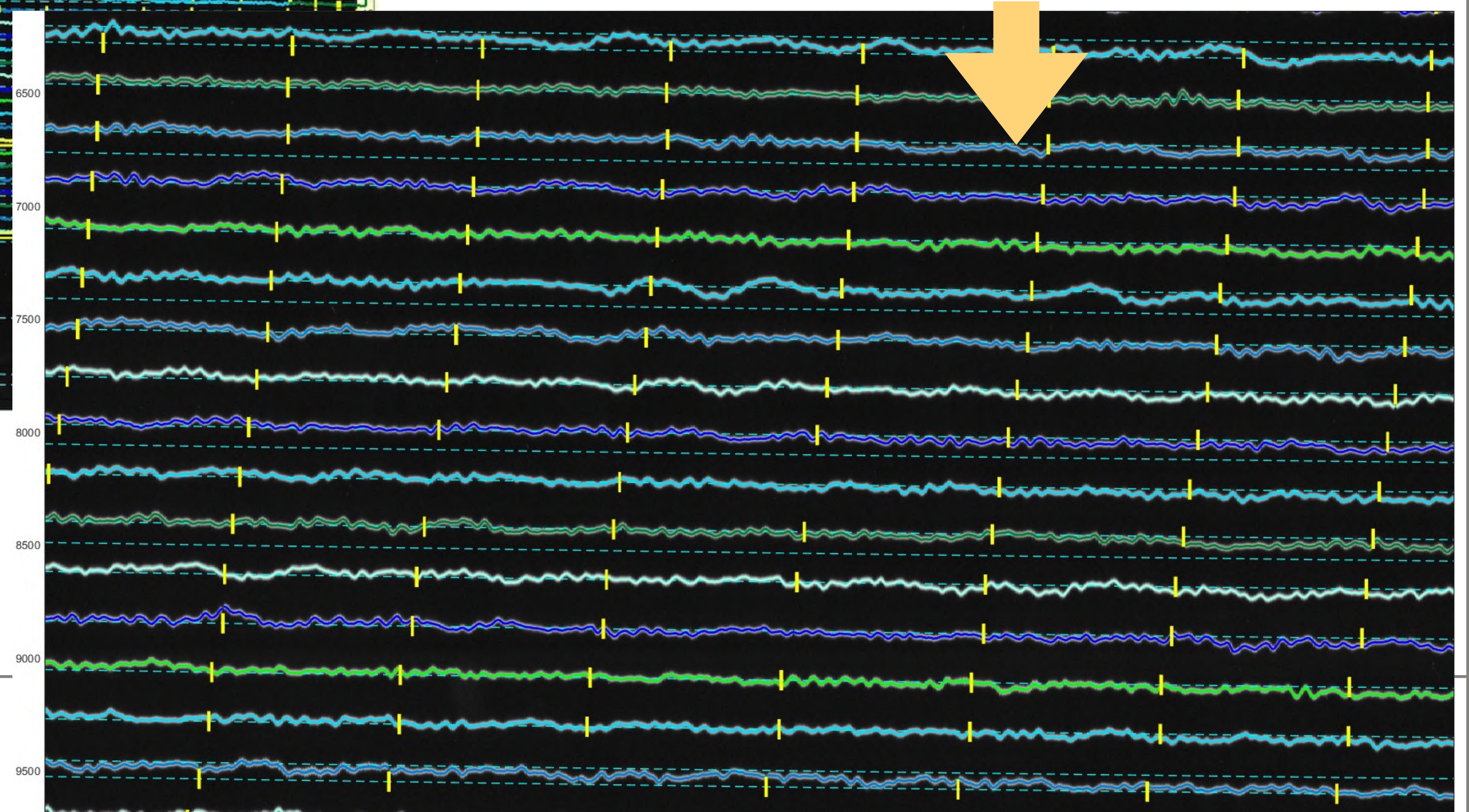
- Results of the classified image with shown yellow segments of the identified trace (enlarged fragment).
- Here: example of the file UCC19540109Gal_E_0812.tif

Digitised traces after classification in *DigitSeis*

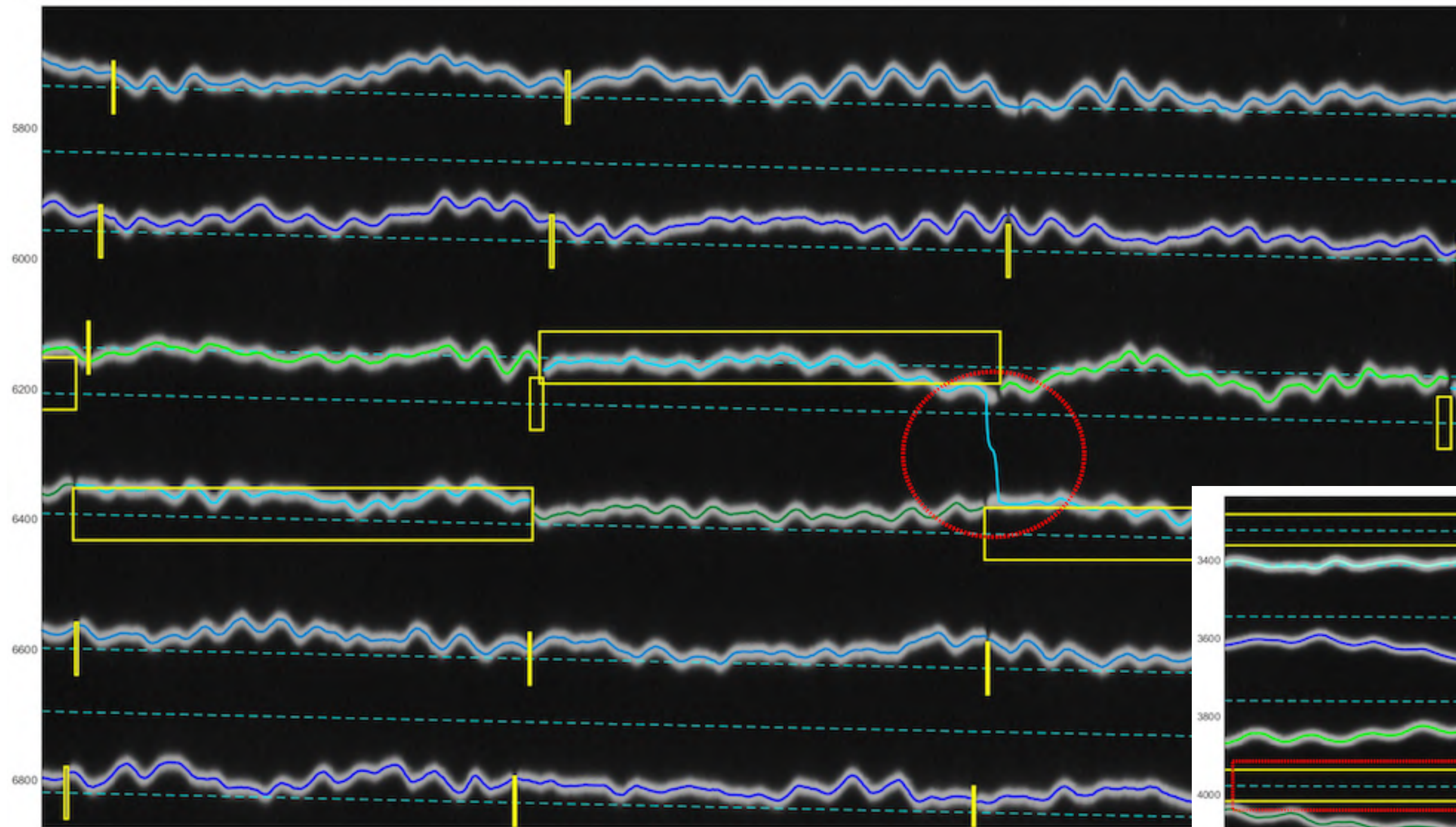
- Enlarged view of the automatically recognised digitised traces displayed by lines of various colours,
- Zero-lines for each trace are visualised as cyan-coloured dashed lines, numbered from top to bottom.
- Vertical yellow dashes are time gaps



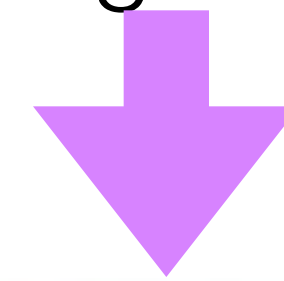
- Some time gaps (upper left part of the image) were not identified and not recognised automatically between the trace and dark background.
- In these cases, gaps required manual correction to identify time intervals.



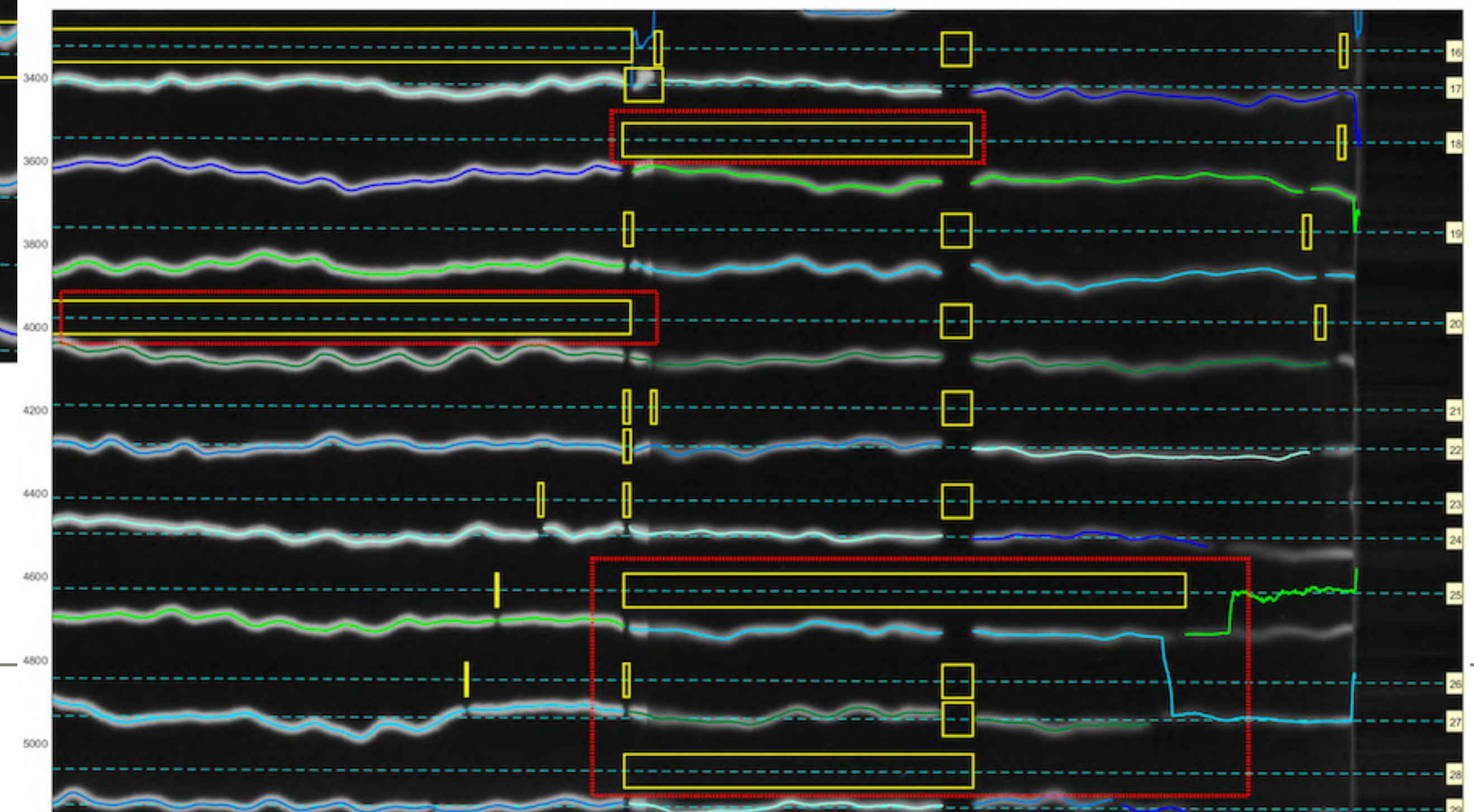
Identified traces for selective correction and re-digitising using Correct Trace mode



- Detected misclassifications caused erroneous digitising.
- The gaps on the zero-lines (small yellow boxes) show the gaps that existed in the old paper in the original image itself.

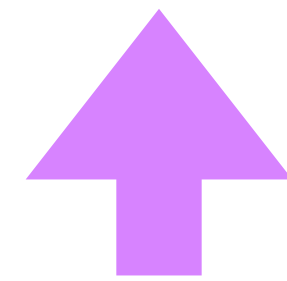
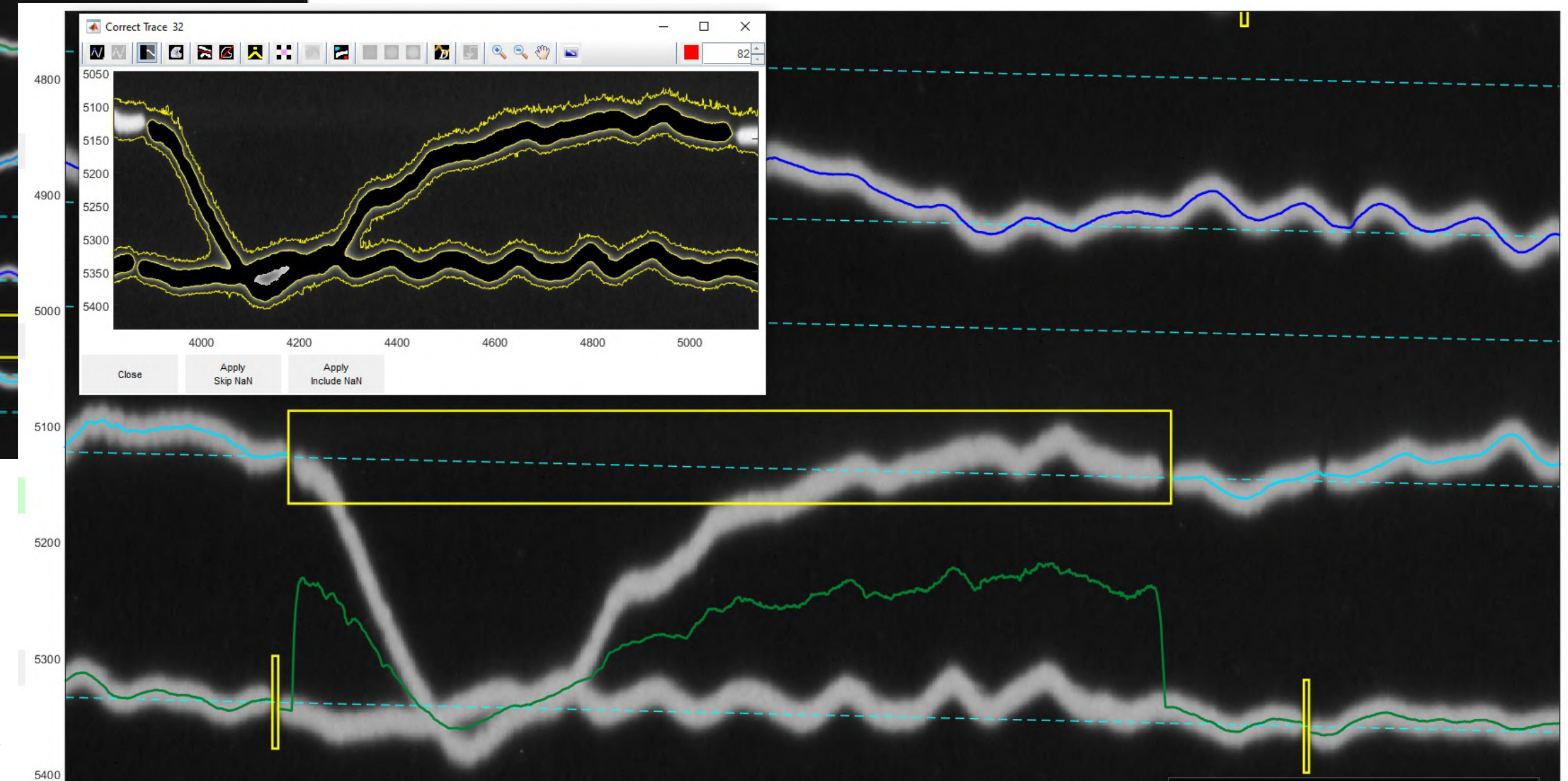
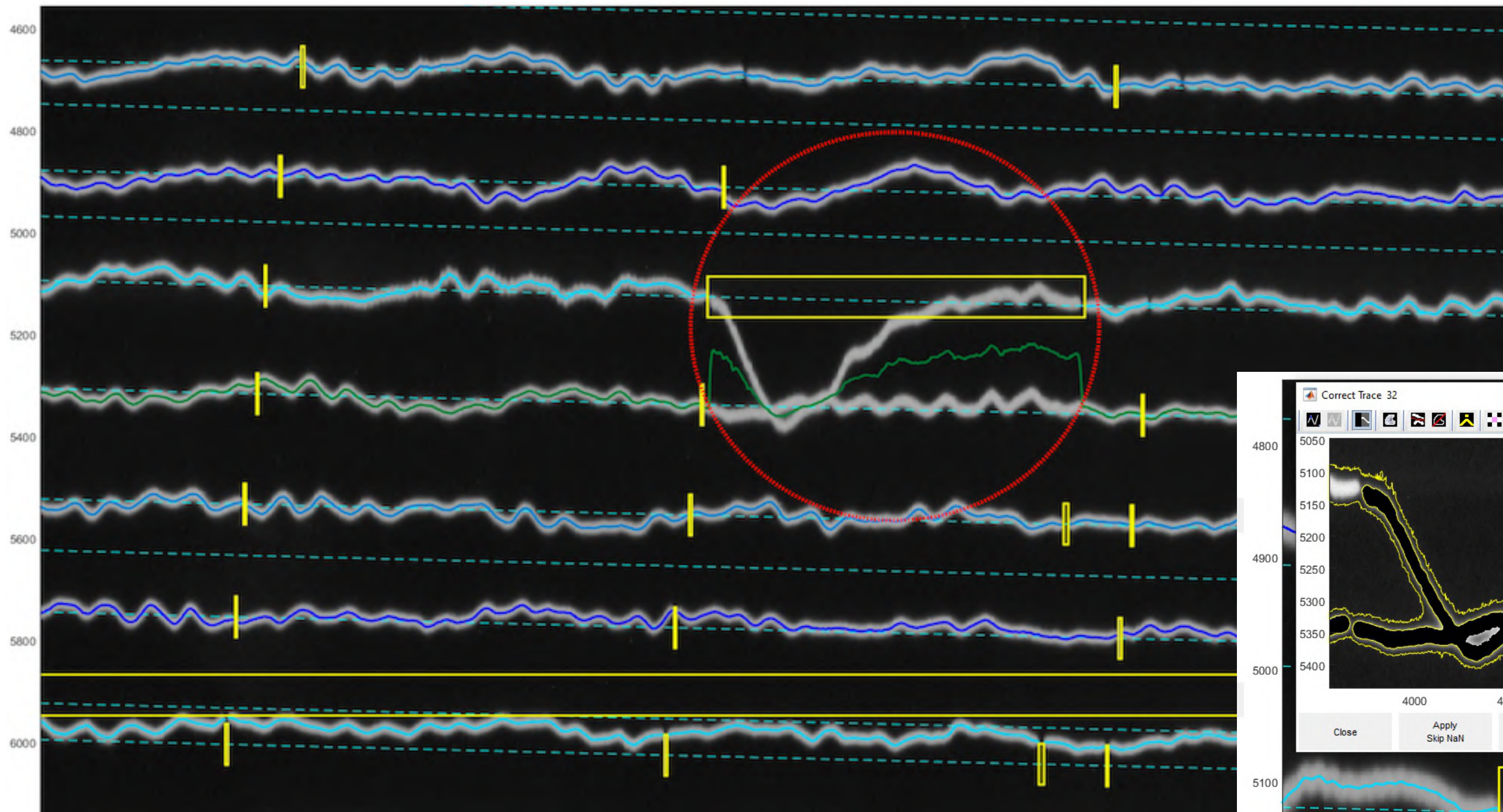
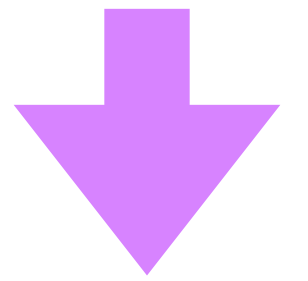


- Identified wrong vector direction of line crossing individual traces



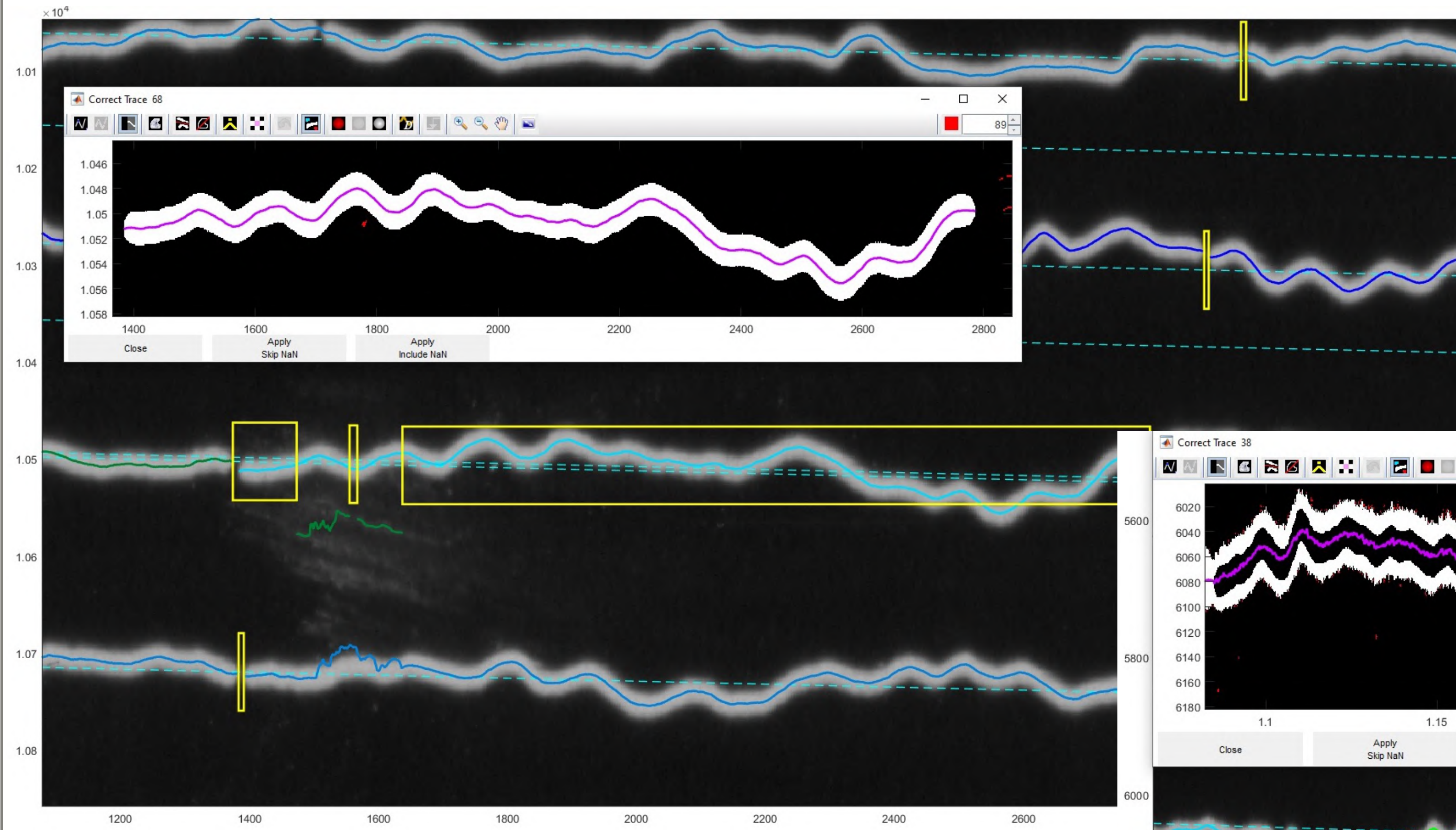
Identified traces for selective correction and re-digitising using Correct Trace mode

- Enlarged view of the manually corrected entangled traces. Correcting misclassified traces with wrong direction based on colour and geometric pixel's characteristics.

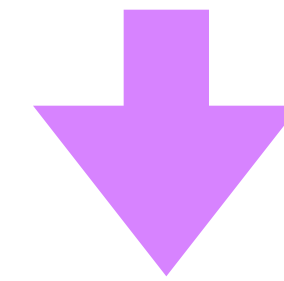


- Overlap of line traces unrecognised during digitising: one segment of trace went steeply downwards and merged with another trace

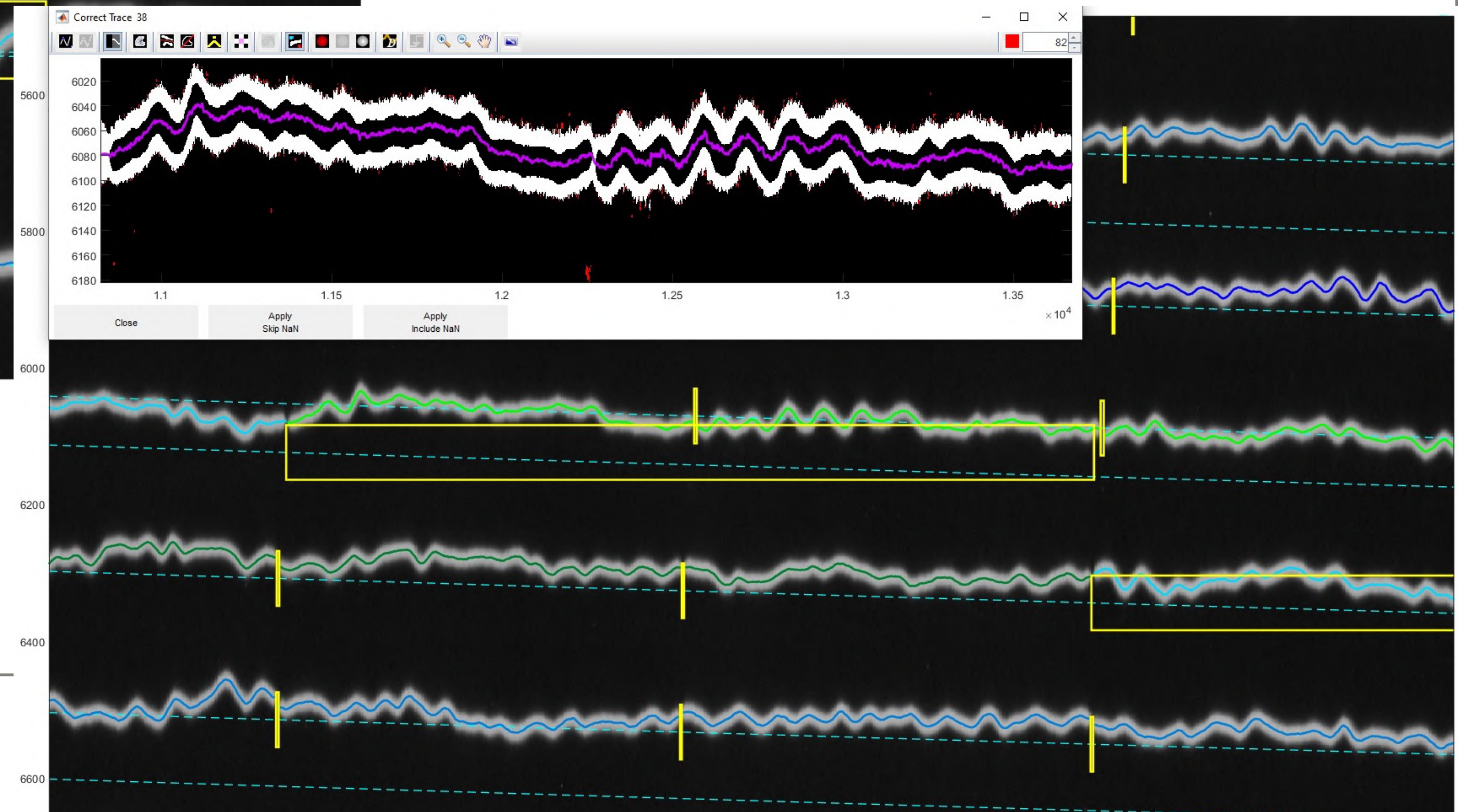
Identified traces for selective correction and re-digitising using Correct Trace mode



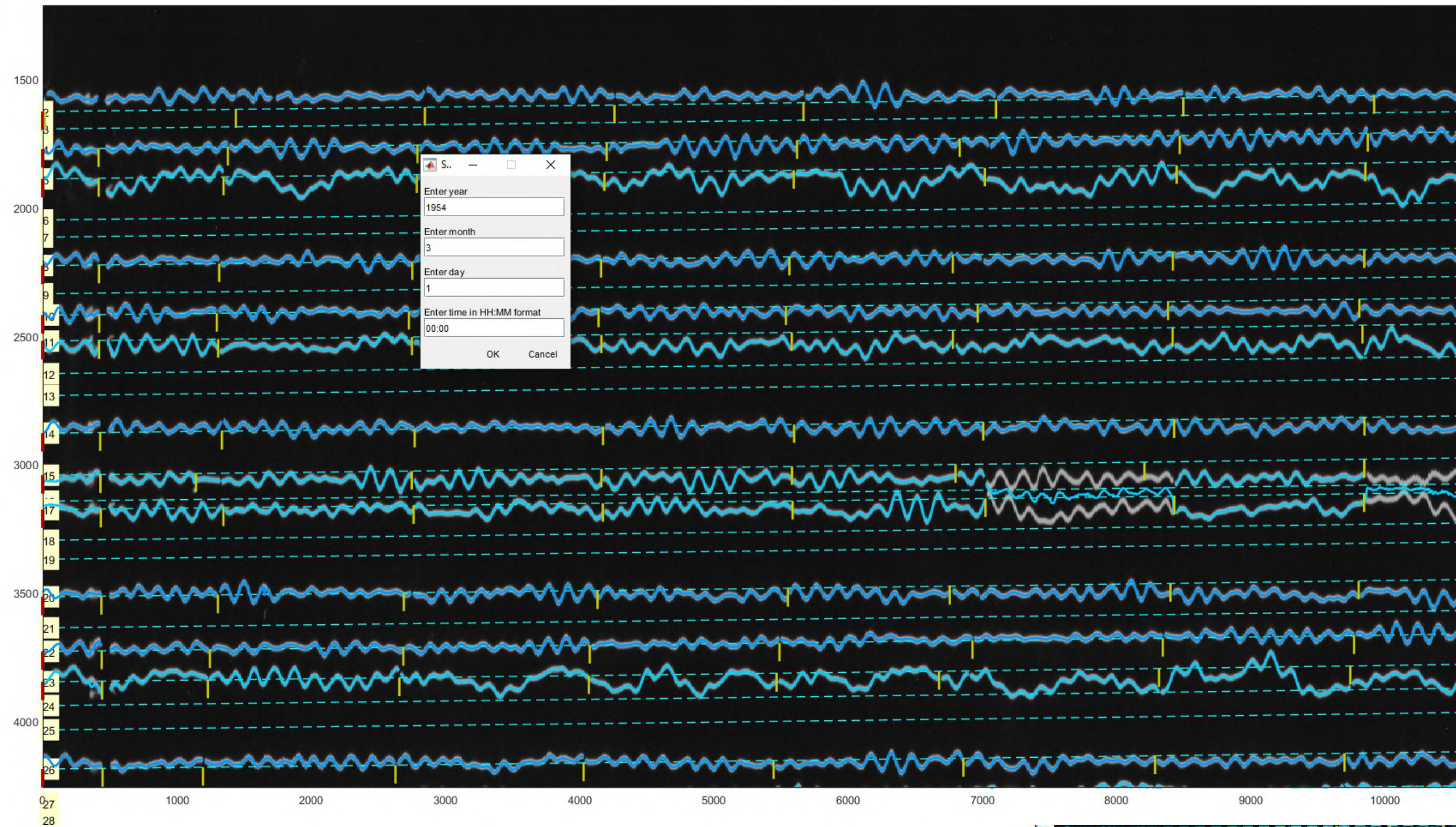
- Reclassification of the selected segment and digitising the centroid of the trace line (purple-coloured). Correcting trace for the selected segments



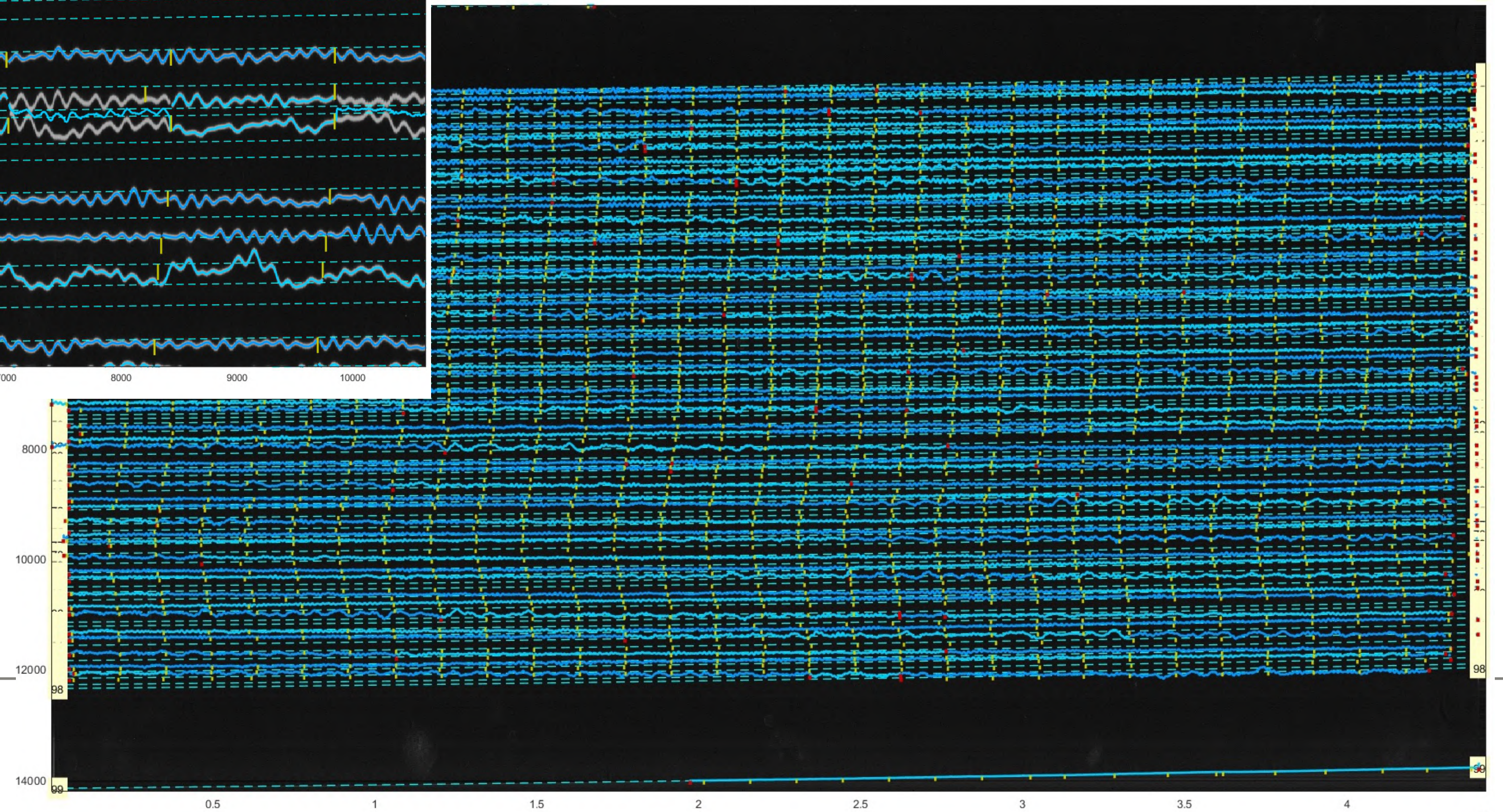
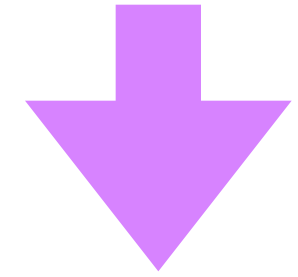
- Merging the trace initially broken into the three separate parts (three small yellow boxes)



Seismogram image with adjusted timing. Here: UCC19540311Gal_E_0727.mat



- Time markers at 1-minute intervals on each 30-minute trace.



- Timing setup using time display increment
- Yellow vertical small dash lines - minute marks

Validating Results of MATLAB File in Python: Post-Processing



DATA ANALYSIS

Reasons

The misclassified pixels were induced by the distortions of paper => erroneous recognition of pixels in the edge (red dots)



IDENTIFYING TIME INTERVALS

Hours and minutes

Hours marks are aligned, as the rotation speed of drum (2/h). Second gaps are slanted due to the rotation of drum.



STATISTIC ANALYSIS

Data analysis

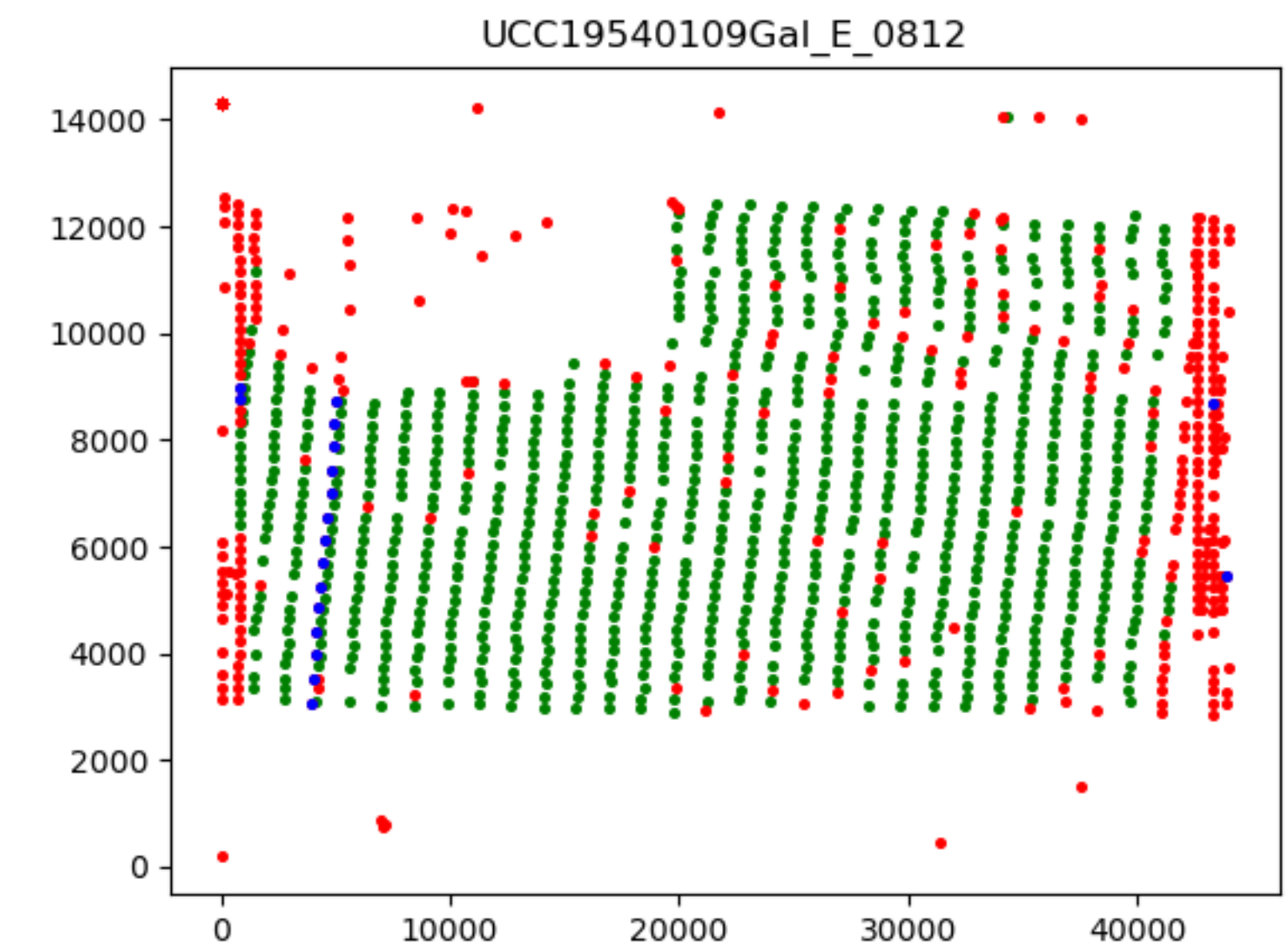
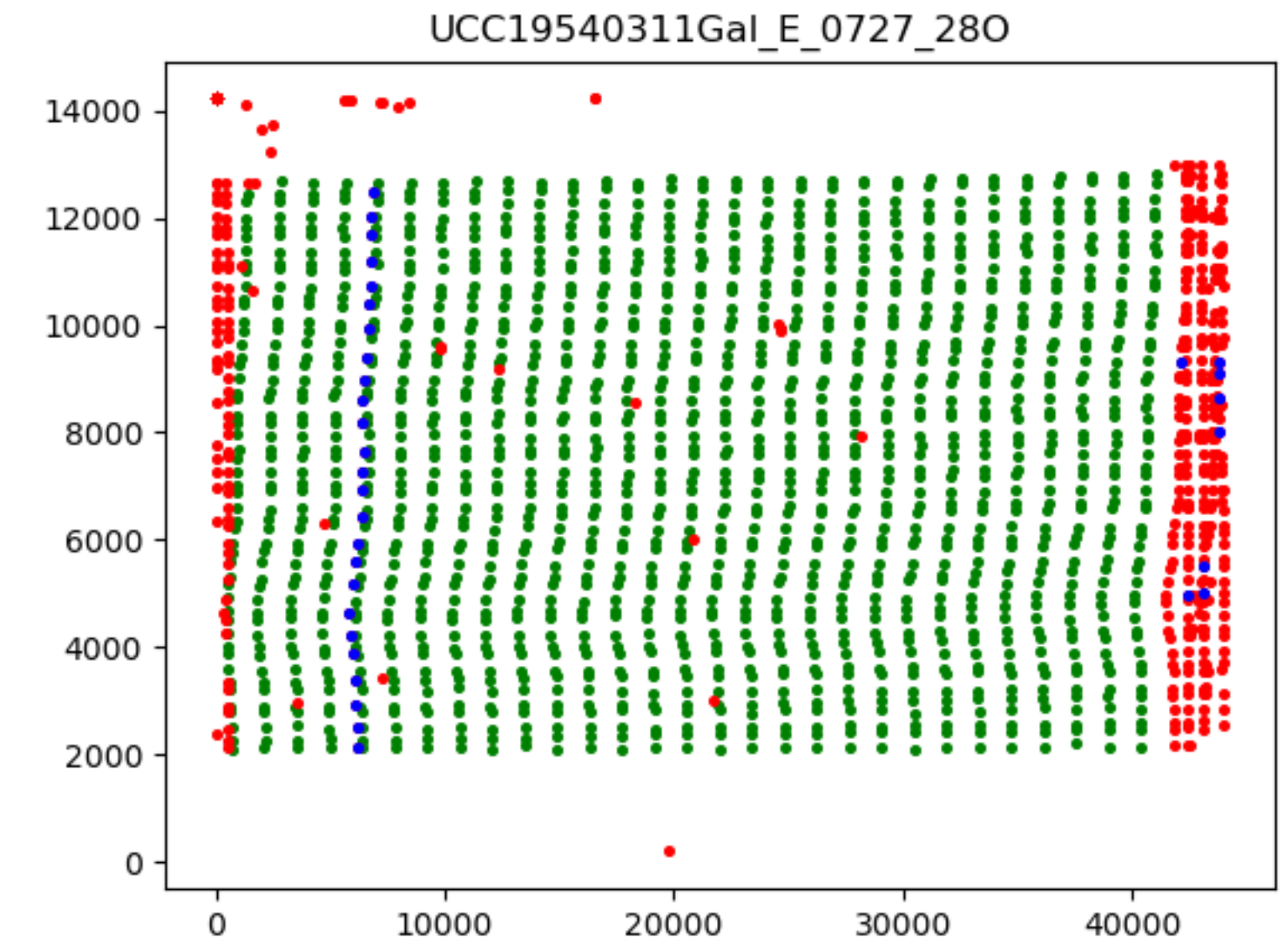
Visualizing frequency of segment length: size of various segments in a trace per hours of recording



CHECK AND PLOTTING

Quality check

The data were kept as valid the segment inside the range $[.8 * Styp, 1.1 * Styp]$.
Green dots - starting position of segments



Controlling digitising results using Python (Matplotlib library).

Blue dots shown the starting position of the hours segments.

Green dots show the minute marks.

Red dots show the noise and edge dots.

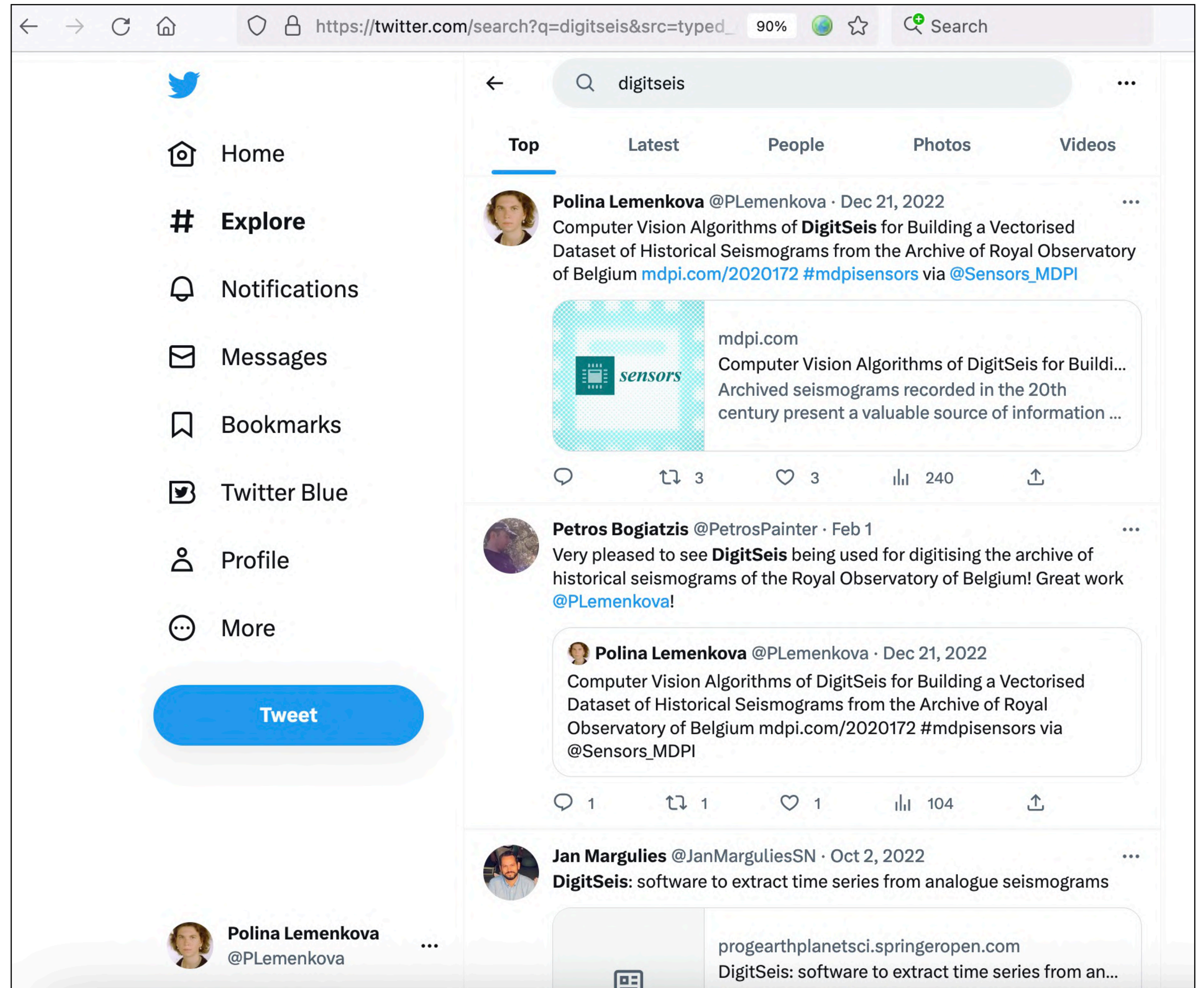
Correctly identified time gaps controlled by Python's Matplotlib

Quality control for time gaps: missed marks in unrecognised segments.

Publication in *Sensors* and continue to Python

Despite the endorsement of our article published in *Sensors* by *DigitSeis* Developer (Dr. Petros Bogiatzis and their team), we need to develop more advanced tools using Python ML algorithms.

The ML Python-based part of this project are introduced in the following section - Part 4.



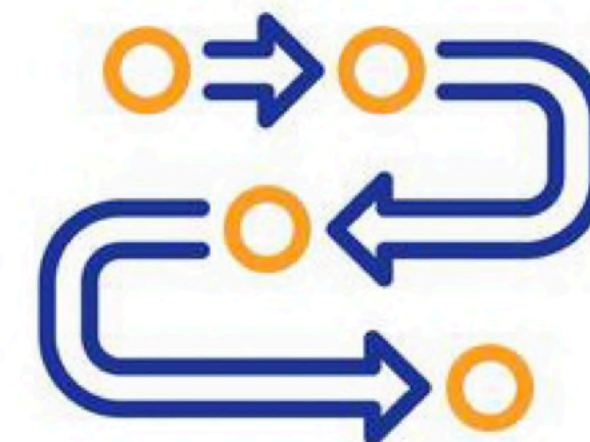
Part 4.

Using *Python* for Automatic Data Processing

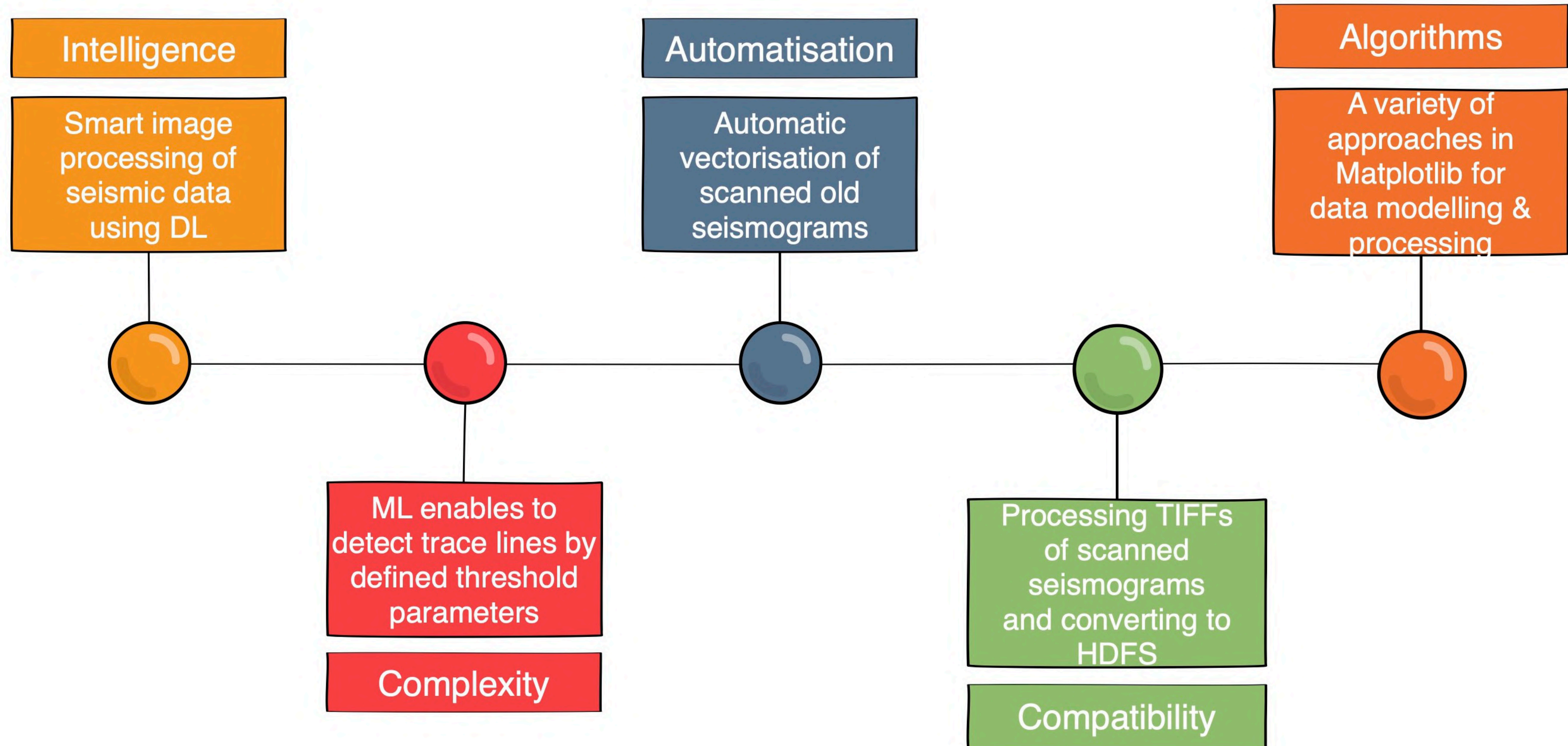
Methodology of the Part 4 of the PhD project based on submitted publication:

Journal article (2) – submitted: Lemenkova, P.; De Plaen, R.; Lecocq, T.; Debeir, O. A Python-based framework for automated vectorisation of the analog seismograms recorded in Uccle seismic station, Belgium. **2023 (expected 10 ECTS)**

- Machine Learning (ML) in vectorising analog seismograms
 - *ML*: Automatic and intelligent data analysis: detecting trace lines using threshold parameters by Python
 - *Image processing*: segmentation, classification of seismograms (separating lines from noise)
 - *Data visualisation and plotting*
 - *Data analysis and interpretation*
- Advanced methods => solve problem of efficient processing of big massifs of old scanned files (TIFFs) for geophysical modelling and data interpretation for seismology research
- Developing new advanced ML algorithms by Python to digitise seismograms and convert them in vector format automatically



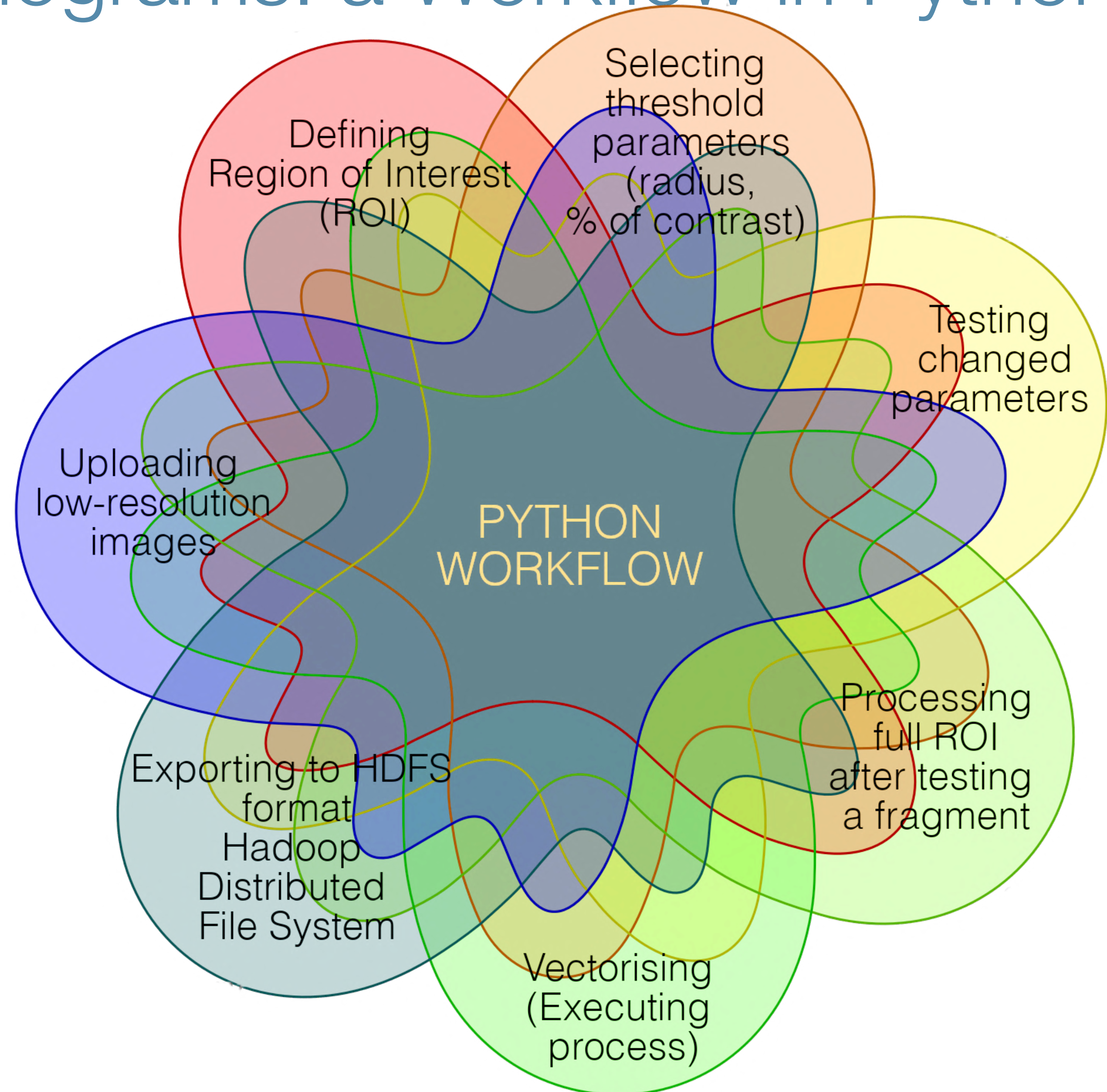
Why Python in Vectorising Seismograms?



ML for Vectorising Seismograms: a Workflow in Python

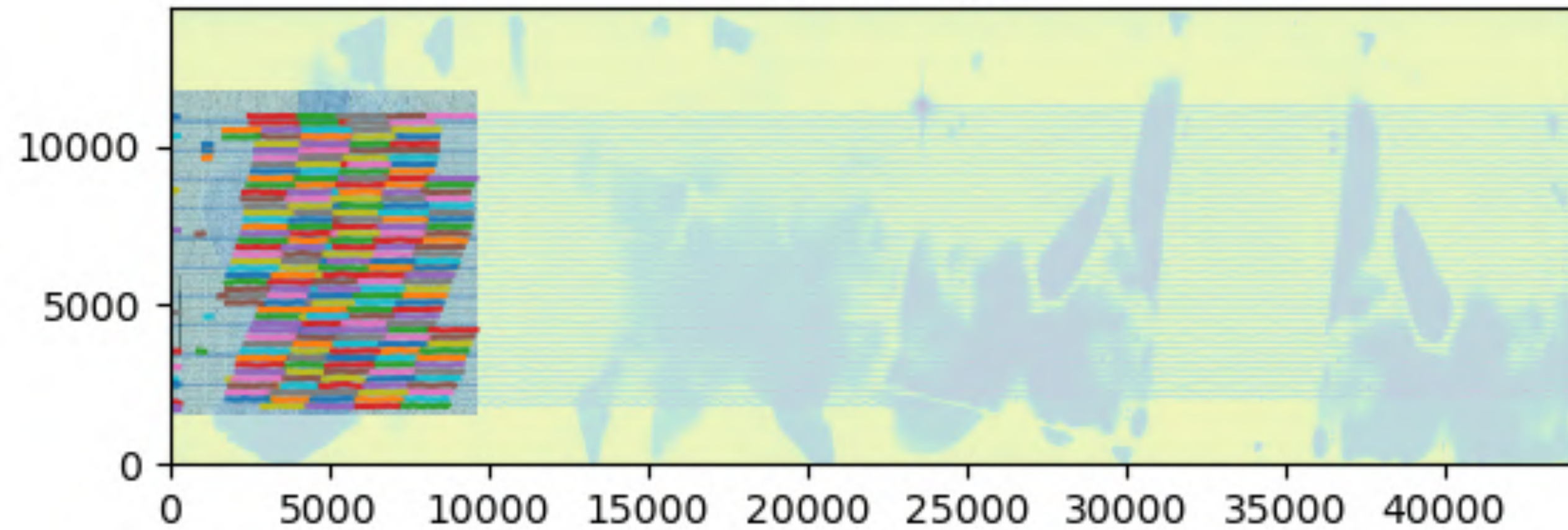
The workflow for digitising seismograms in Python includes several steps:

- Defining Region of Interest (ROI)
- Selecting threshold parameters (radius of pixels, percentage of contrast)
- Sampling several approaches with varied parameters
- Processing full ROI after testing parameters and selecting the best and optimal parameters (e.g. pixel size 30, radius 85%)
- Vectorising (executing Python script)
- Exporting the results to the HDFS format

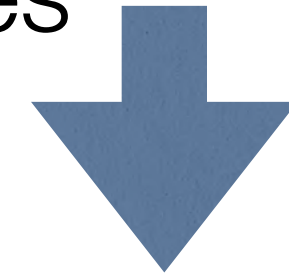


Python-based digitising of raster image (1)

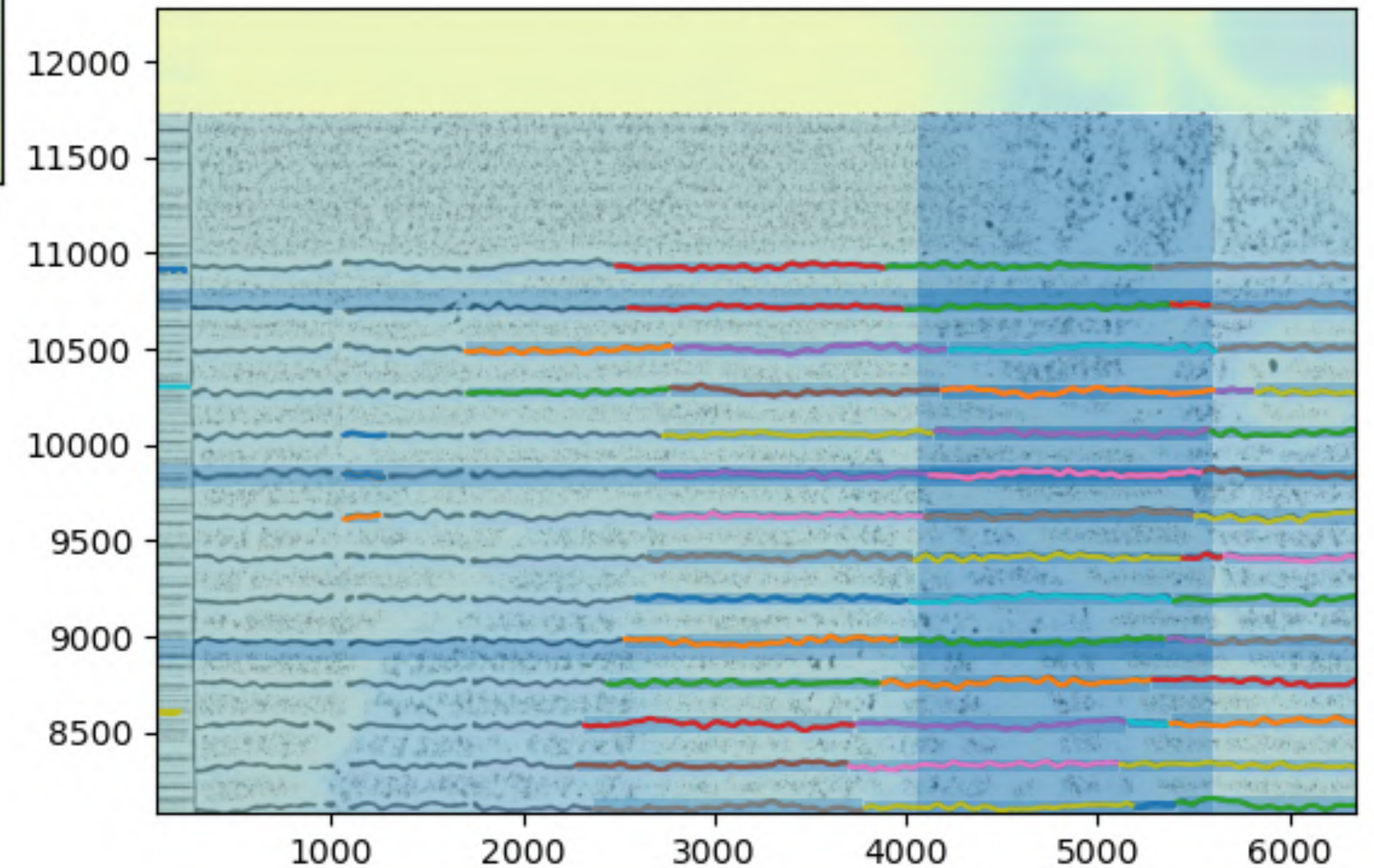
Wholeslide



Enlarged fragment of the vectorised segments of the trace lines



Wholeslide

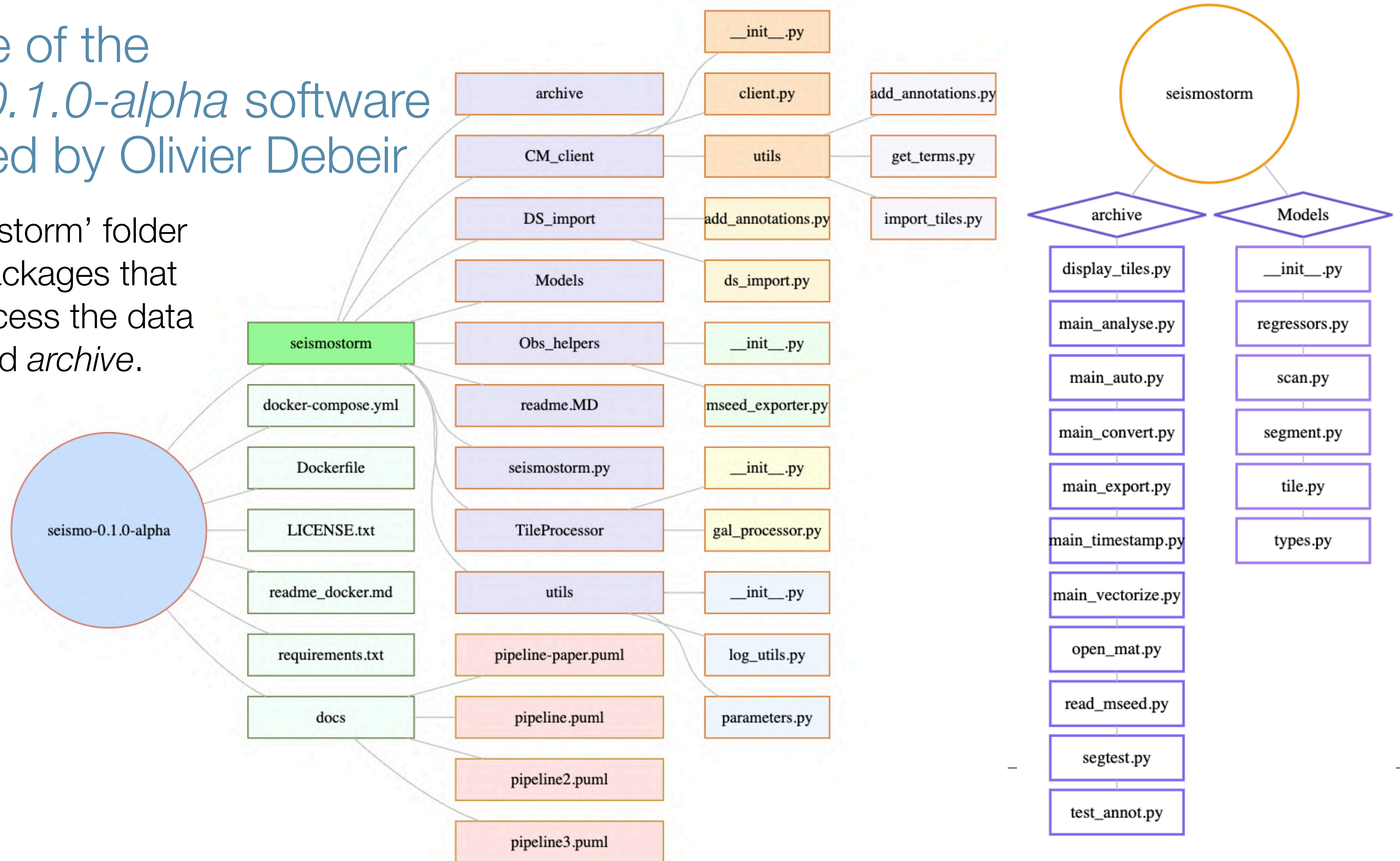


Automated vectorising of seismograms was performed using several work steps. First, the low-resolution images were grabbed by Python script from the Cytomine and used in script.

Workflow for vectorising in Python, Matplotlib library (slide 1/10)

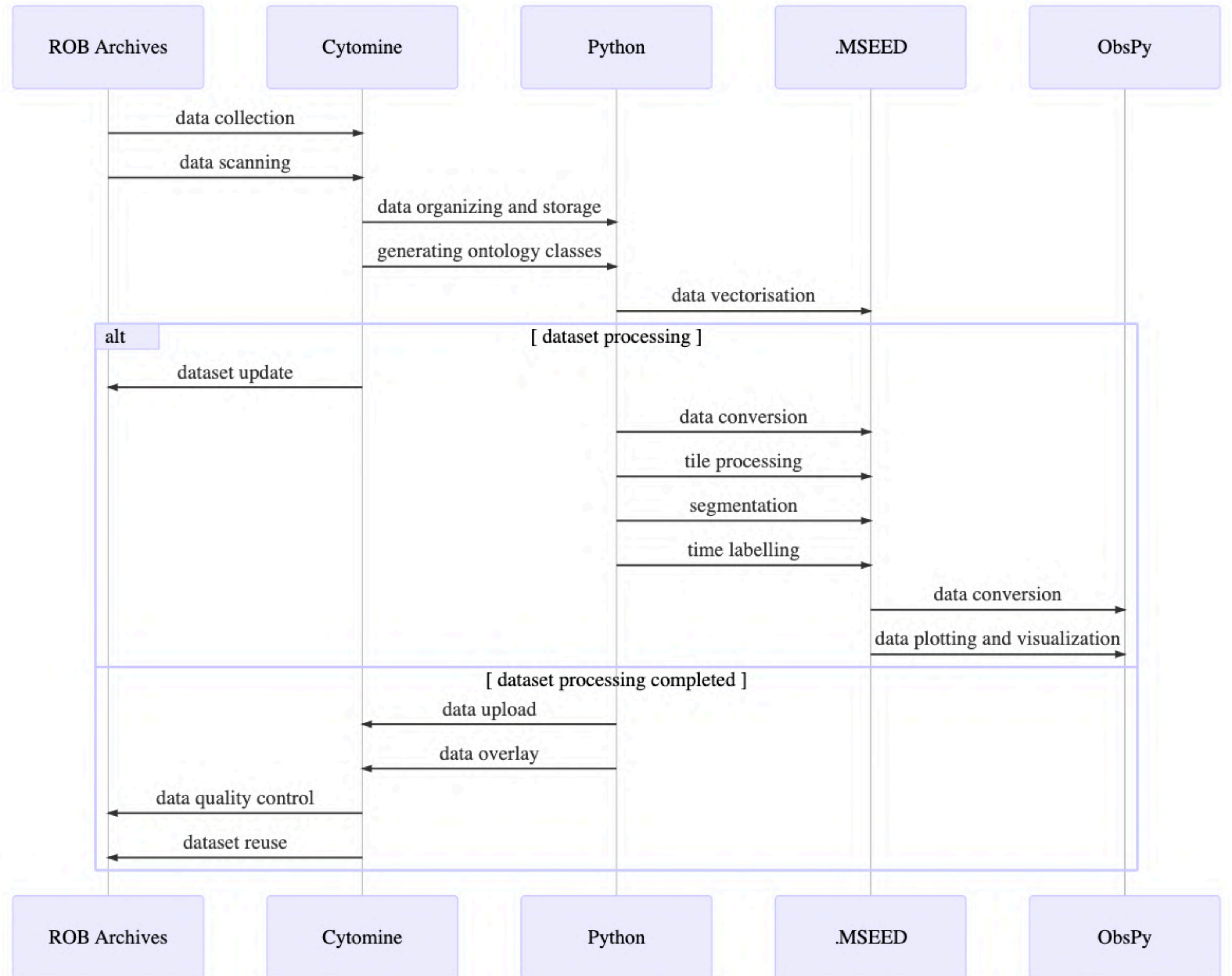
Structure of the *seismo 0.1.0-alpha* software developed by Olivier Debeir

The 'seismostorm' folder has main packages that actually process the data - *Models* and *archive*.



Workflow outline:

- *ROB* is the data source
- Cytomine workspace - storage place, management and editing system for a large dataset
- *Python* algorithms - processing and vectorising data
- *MSEED* files generated by Python as main outputs
- *ObsPy* - a Python library for visualization of the vectorised seismograms

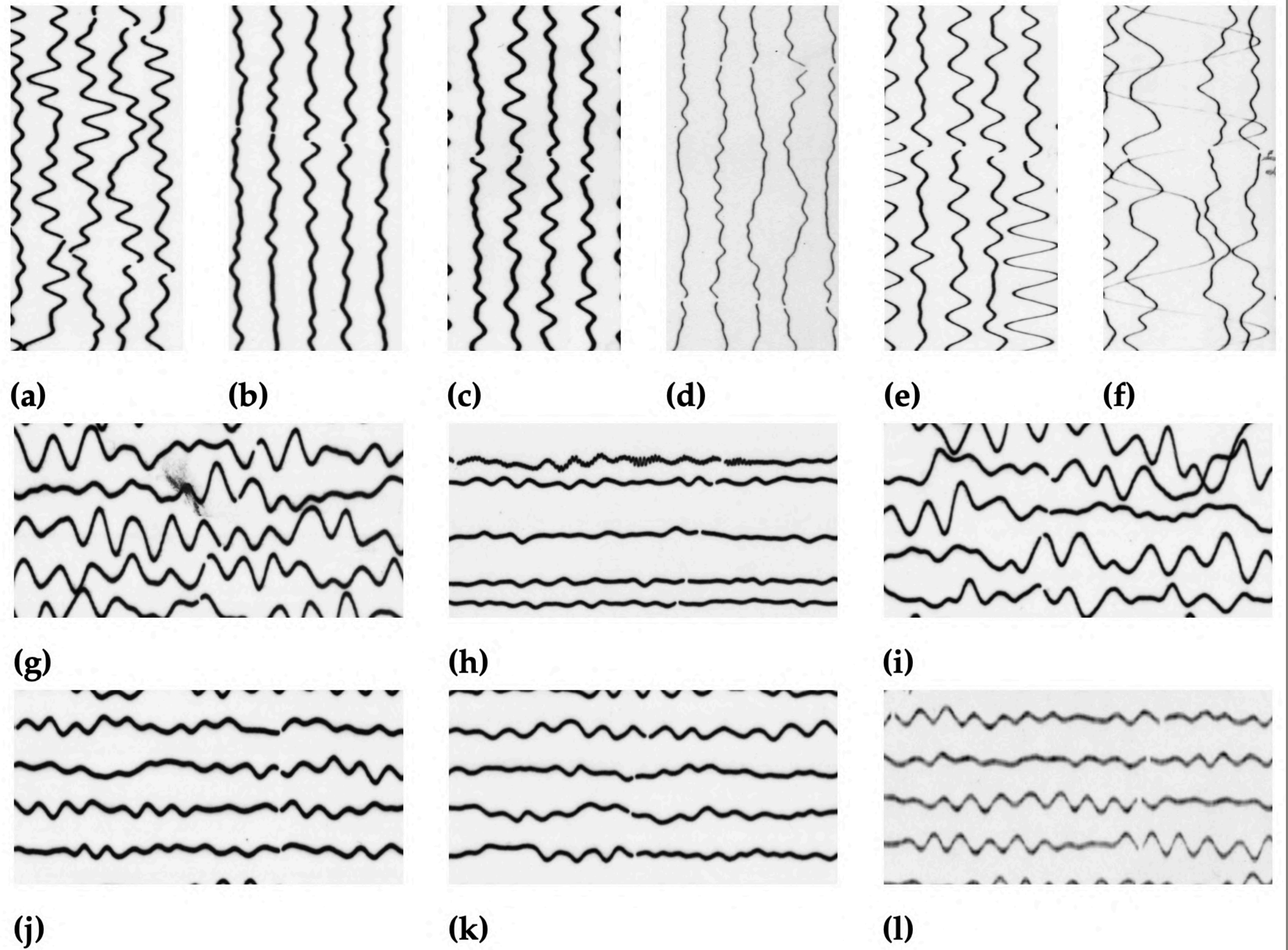


Fragments of the original scanned seismograms used as reference images for machine-based training of vectorising.

(a)-(f) Vertically oriented seismogram samples.

Cytomine IDs from left to right: 7747, 7765, 9748, 10864, 10870, 12338.

(g)-(l) Horizontally oriented seismograms. Cytomine IDs, left to right, up to bottom: 9456, 5660, 9469, 9730, 7795, 9365.



Examples of the fragment of the raster scanned paper-based seismograms used for Python processing

Workflow steps of the novel Python-based framework:

- The images were uploaded to Python by tiles; The tiles had width=4 min, height=1024 p, overlap between tiles =1.1% min. The horizontal gap of overlap (interline) is 200 pxl;
- The image was cropped to ROI to minimise the workflow: empty edges were subtracted from the image using threshold;
- Thinning of lines was done by threshold parameters: number of pixels (30), radius of target pixel's (101), intensity of grey in pixel's colour (30%);
- Vectorisation was performed as an iterative loop for each tile;
- Detecting timing gap intervals through buffering around hour and minute;
- Correcting errors for double-line vectorisation on the overlapping edges;
- Labelling time intervals.
- The comparison of the vector layer overlaid in Cytomine on the original raster image shown accurate vectorisation of the seismograms based on trace discrimination.

Examples of the fragment of the raster scanned paper-based seismograms used for Python processing

Python code for defining *Tile* class on seismograms (© O.Debeir)

Tiles and **segments** are two important class objects on the seismograms defined using characteristics of time start and end of the seismogram recording via time gaps and marks.

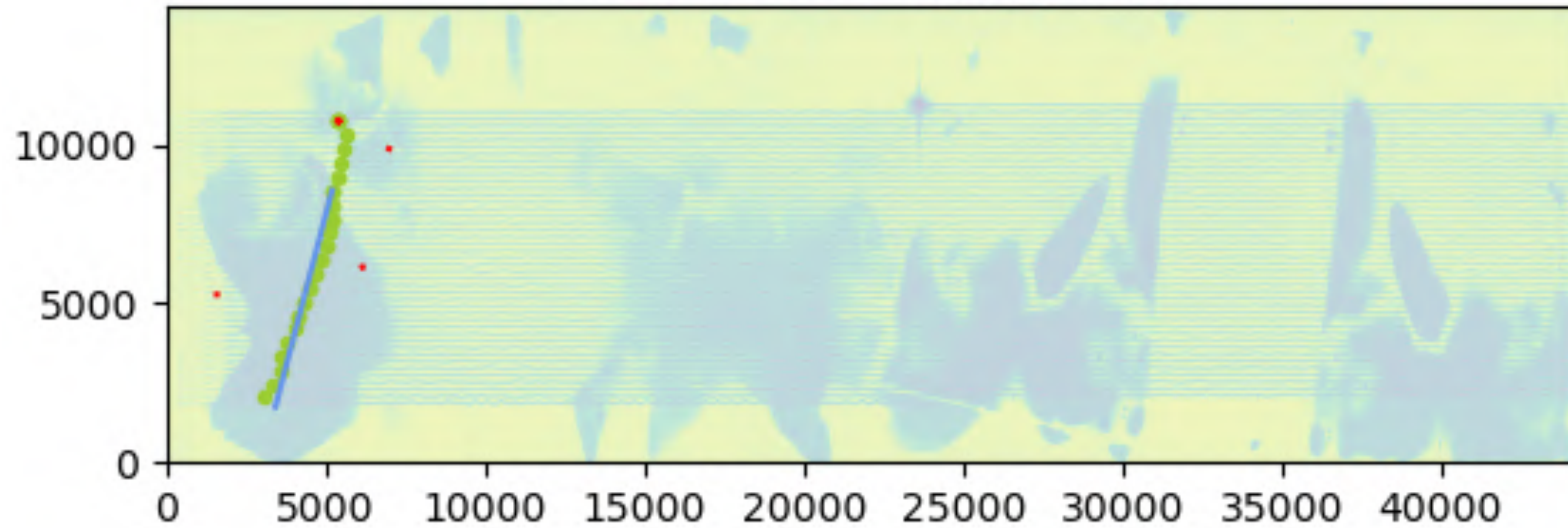
Segment is a fragment of the seismogram defined as the period of records during one minute, i.e., it is limited by two minute gaps indicating the start and the end of the minute.

Tile is defined as a sequence of records within the consecutive line on a seismogram which is interrupted and followed by the next tile and repeated iteratively on the whole seismogram.

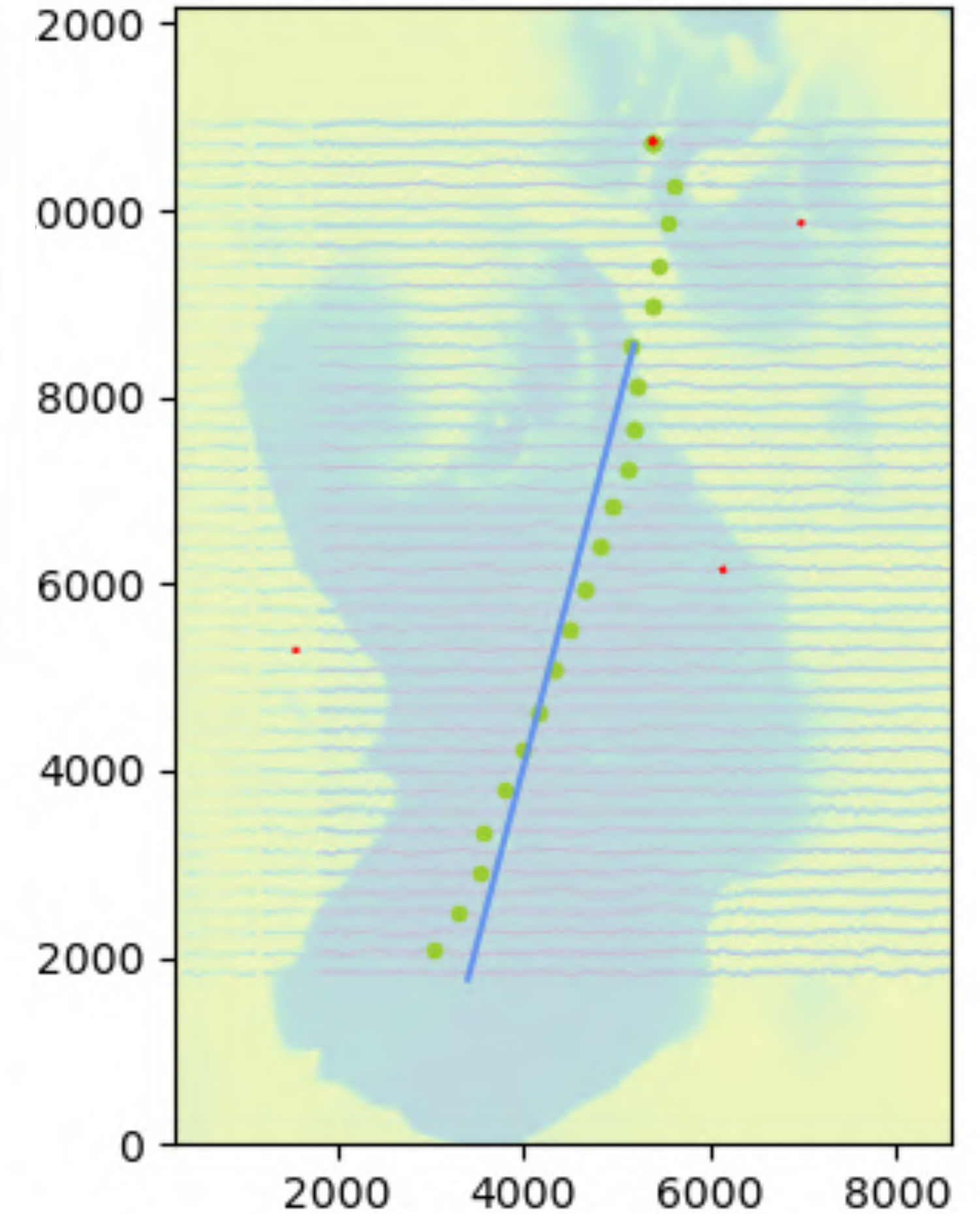
```
1 class Tile():
2     def __init__(self, param,x0, y0, w, h, sx, sy, params=None):
3         self.param = param
4         self.x0 = int(x0)
5         self.y0 = int(y0)
6         self.w = int(w)
7         self.h = int(h)
8         self.sx = sx # full image size
9         self.sy = sy
10        self.processed = False
11        self.th = None
12        self.Segments = []
13        if params:
14            self.params = params
15        else:
16            self.params = {'radius': 50, 'percentile': .3}
17    def __str__(self):
18        return f'origin({self.x0},{self.y0} w x h ({self.w},{self.h})) #segments: {len(
19            self.Segments)} proc. {self.processed}'
20
21    def insert_ima_extent(self, ima, ax):
22        # plot image at the Tile coordinates in axes
23        ax.imshow(ima, extent=[self.x0, self.x0 + self.w, self.sy - self.y0 - self.h,
24            self.sy - self.y0],
25            cmap=plt.cm.gray, alpha=.5)
26
27    def plot(self, ax, display_th=True, display_seg=True, display_tile=True, txt=None):
28        # plot Tile in gca (in wholeslide coordinates)
29        if display_tile:
30            if self.processed:
31                alpha = .5
32            else:
33                alpha = .2
34            ax.add_patch(Rectangle((self.x0, self.sy - self.y0 - self.h), self.w, self.h
35                ,
36                alpha=alpha))
37            if txt:
38                ax.text(self.x0, self.sy - self.y0 - self.h, txt)
39
40            if (self.th is not None) and display_th:
41                ax.imshow(self.th, extent=[self.x0, self.x0 + self.w, self.sy - self.y0 -
42                    self.h,
43                    self.sy - self.y0], cmap=plt.cm.gray, alpha=.5)
44            if display_seg:
45                for s in self.Segments:
46                    s.plot(ax)
```

Python-based digitising of raster image (2)

UCC19540107Gal_E_0815_r



UCC19540107Gal_E_0815_r



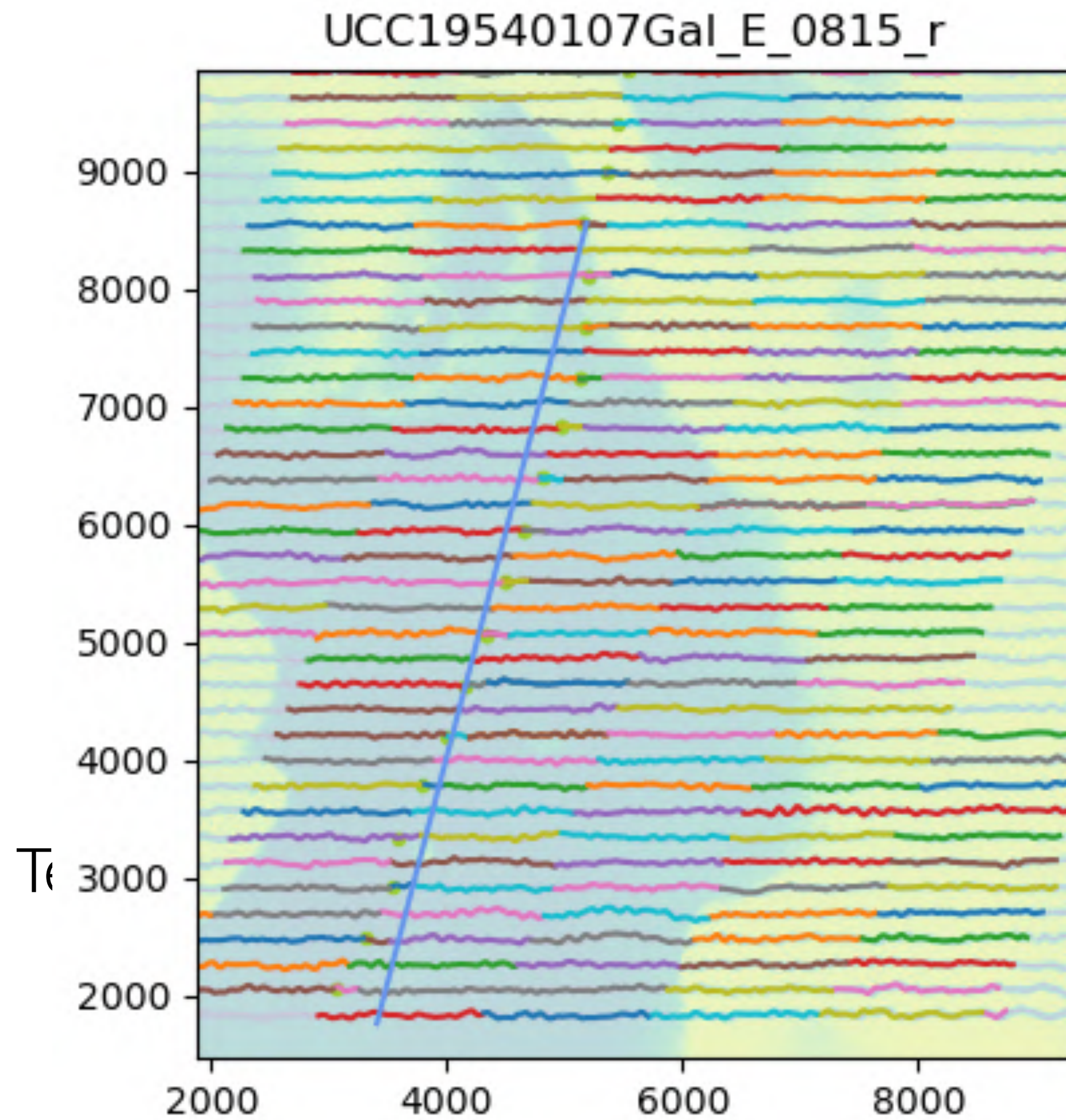
Second, the hour gaps have been detected using the indication of the repeatability of gaps (double gaps, close located next to the first minute of this hour).

Above: view of the seismogram with indicated hour gaps.

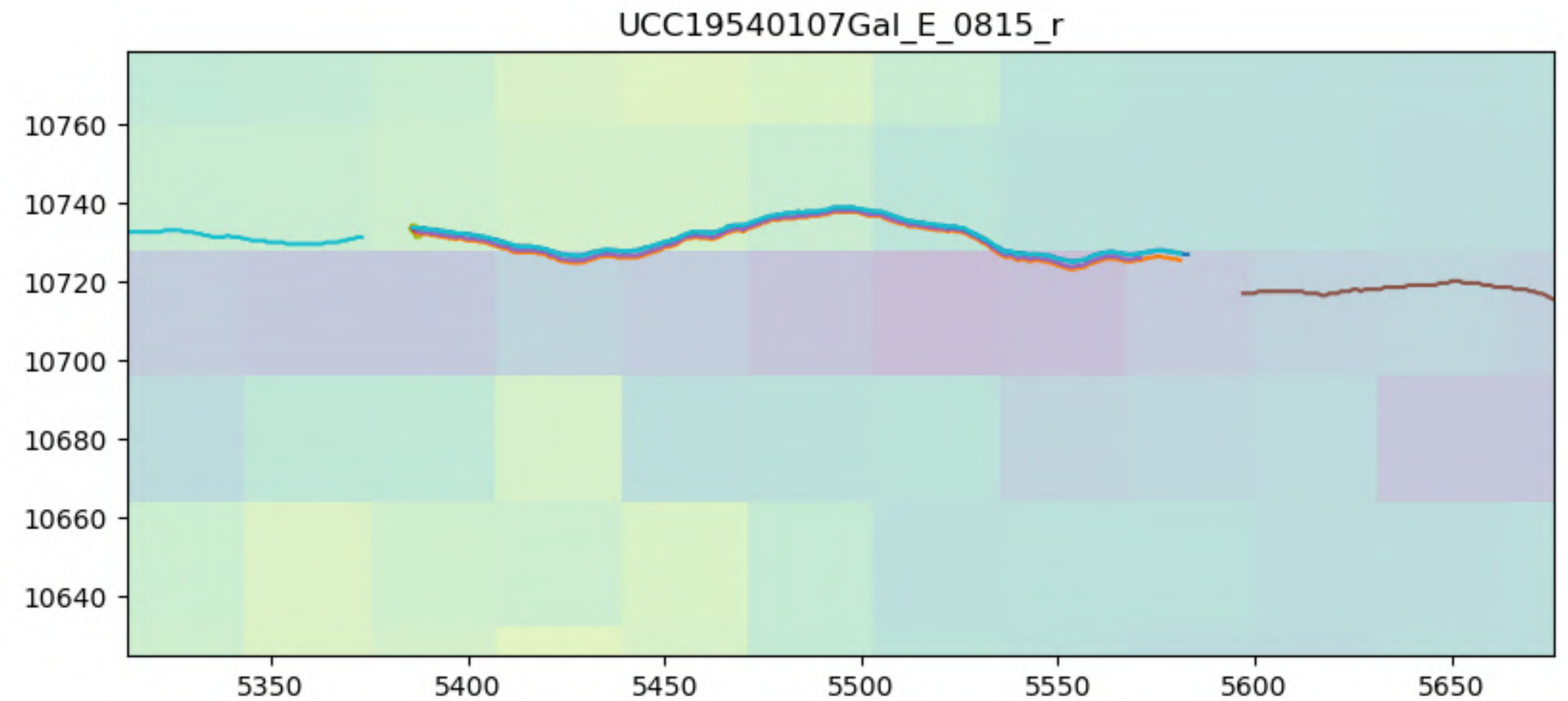
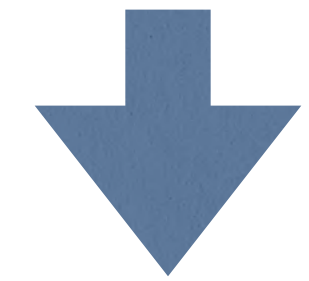
Right: enlarged fragment.

Workflow for vectorising in Python, Matplotlib library (slide 2/10)

Python-based digitising of raster image (3)

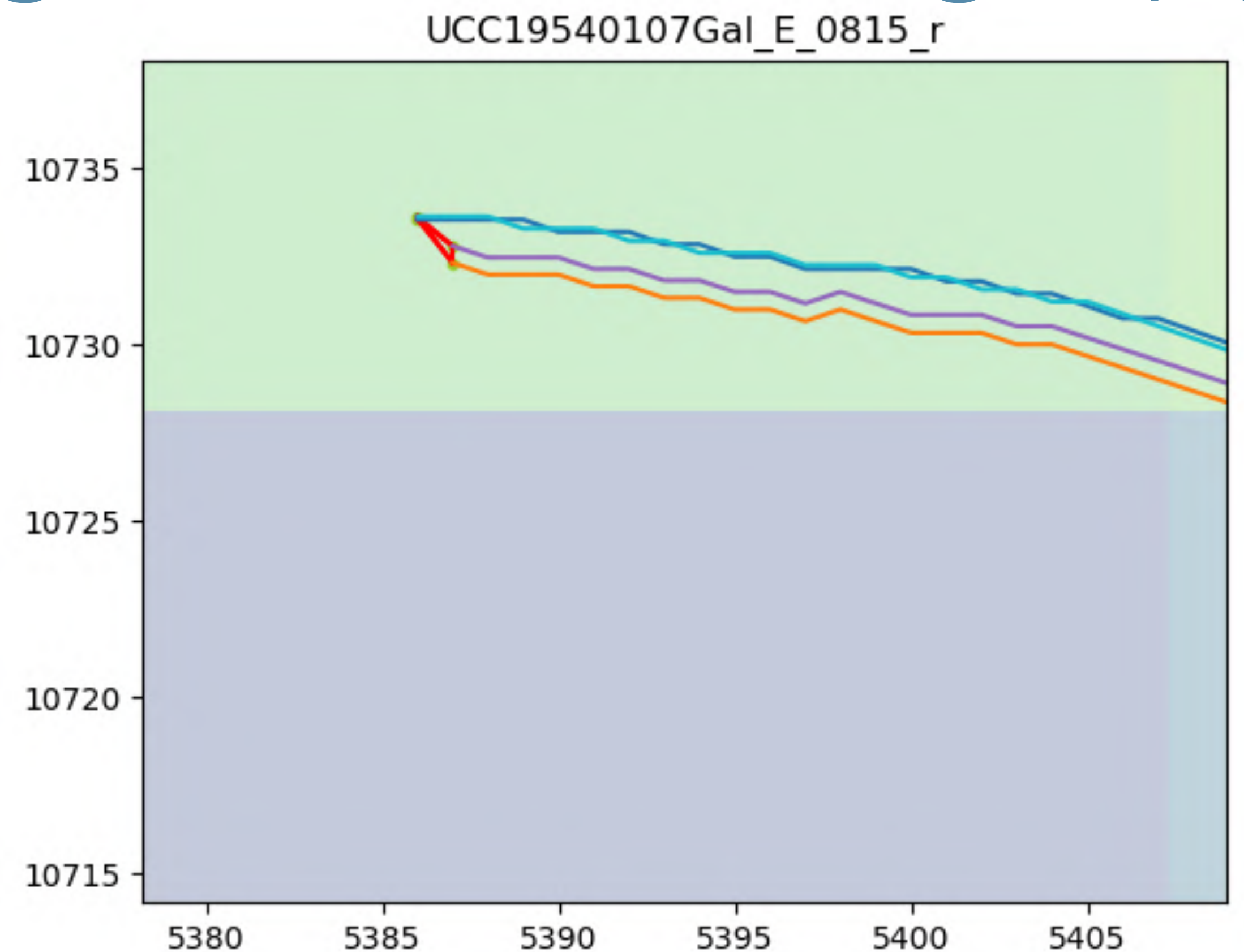
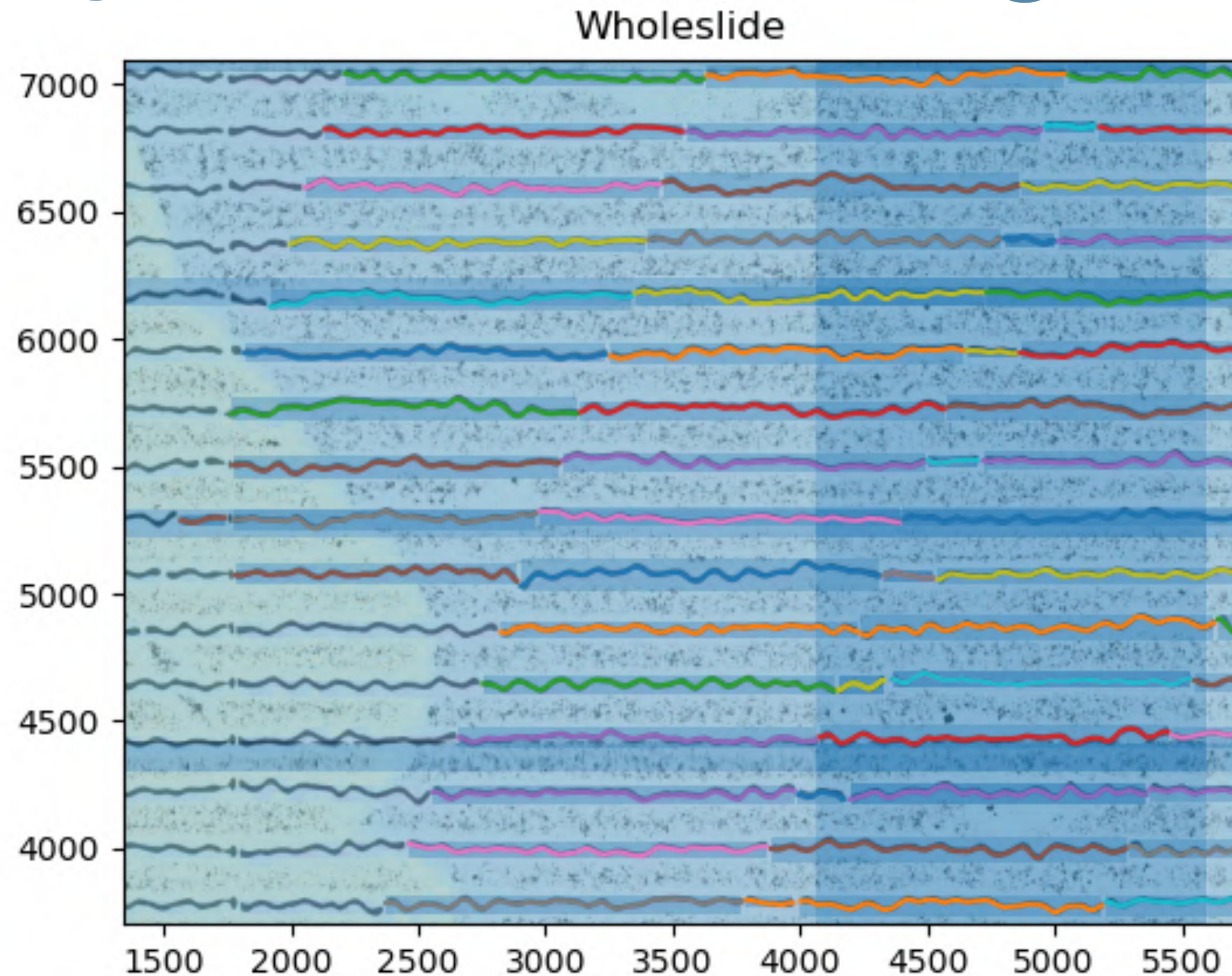


Third, the line with double vectorisation (overlapping time periods) were processed.



Workflow for vectorising in Python, Matplotlib library (slide 3/10)

Python-based digitising of raster image (4)



Left: Example of the digitised traces in Python.
Above: Example of the misclassified line, which was vectorised several times as belonging to 'neighbor' hours segments (e.g. hour 1 and hour 2).

Workflow for vectorising in Python, Matplotlib library (slide 4/10)

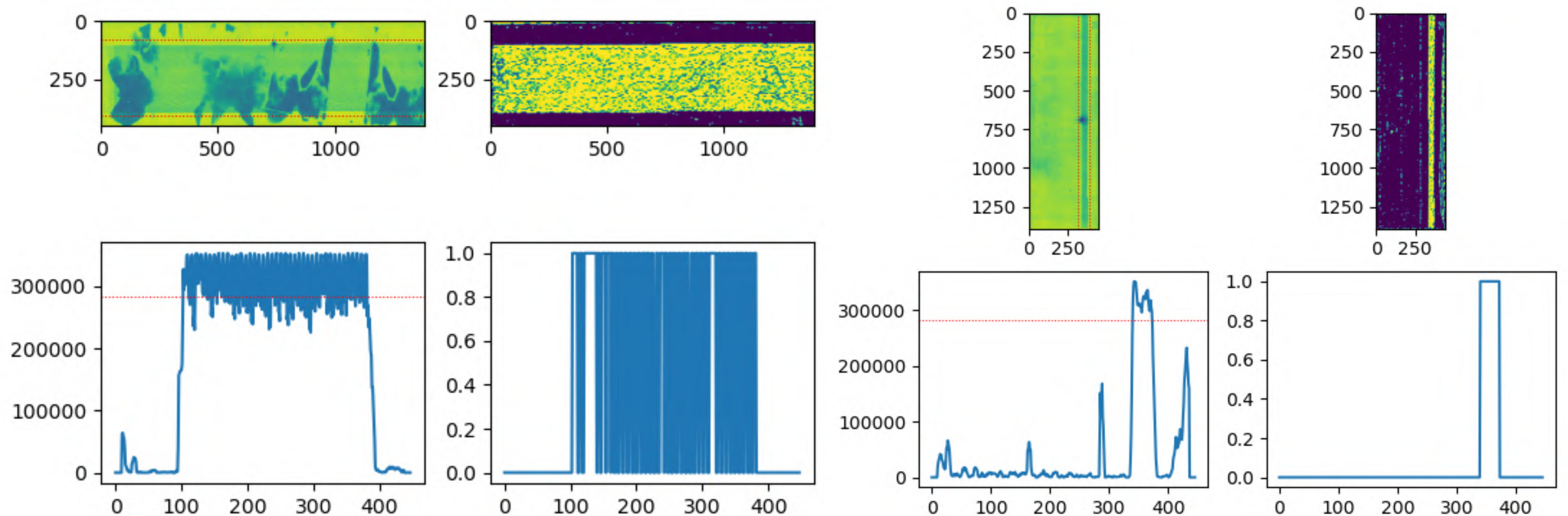
Snippet of the Python code (© O.Debeir)

```
1 from utils.log_utils import get_logger
2 my_logger = get_logger(__name__)
3 my_logger.info("Start logging")
4 id_project = 451
5 image = {'name':'UCC19540311Gal_N_0727','cytomine_id':5713} # vert std
6 cmid = image['cytomine_id']
7 # ----->
8 # Step 0- extract on single Tile
9 if False:
10     # create main Scan object
11     scan = Scan(cmid)
12     # analyse image/import lowres/find best parameters
13     scan.fetch_info()
14     scan.preproc(debug=False)
15     sample = scan.sample_fullres(w=2048,h=1024)
16     print(sample)
17     plt.imshow('sample.png',sample,cmap=plt.cm.gray)
18 # ----->
19 # Step 1- initialization of the Scan / build the grid
20 if False:
21     # create main Scan object
22     scan = Scan(cmid)
23     # analyse image/import lowres/find best parameters
24     scan.fetch_info()
25     scan.preproc(debug=True)
26     print(scan)
27     sample = scan.sample_fullres()
28     scan.autoset_params(sample, debug=True)
29     print(scan.tiles_param)
30     # prepare grid
31     scan.prep_grid(max_tiles=None)
32     # run analysis (width/ntiles)
33     scan.process(max_tiles=1,save_th=True)
34     # save Scan object
35     scan.save()
36     # display Scan object
37     fig, ax = plt.subplots()
38     scan.plot(ax,display_th=True,display_tile=True,display_seg=False)
39     plt.title(scan.name)
40     plt.show()
41 # ----->
42 # Step 2- update of the scan / digitize each tile
43 if False:
44     # load Scan object
45     scan = Scan(cmid)
46     scan.read()
47     # scan.fetch_info()
48     # sample = scan.sample_fullres()
49     # scan.autoset_params(sample, debug=True)
50     for i in range(10):
```

```
51     scan.process(max_tiles=5)
52     # update file
53     scan.save()
54     (proc, tot) = scan.get_tiles_processed()
55     fig, ax = plt.subplots()
56     scan.plot(ax,display_th=True,display_tile=True,display_seg=True)
57     plt.title(f'{scan.name}\n[{proc}/{tot}]')
58     plt.show()
59 # ----->
60 # Step 3- Re-scan some tiles
61 if False:
62     scan3 = Scan(cmid)
63     scan3.read()
64     print(scan3)
65     # invalidate a specific tile for rework
66     invalid = [84,22,49]
67     for i in invalid:
68         scan3.tiles[i].processed = False
69         scan3.tiles[i].Segments=[]
70     # reprocess tiles
71     # scan.fetch_info()
72     scan3.preproc(debug=True)
73     sample = scan3.sample_fullres()
74     scan3.autoset_params(sample, debug=False)
75     scan3.process(save_th=False)
76     scan3.save()
77     (proc, tot) = scan3.get_tiles_processed()
78     fig,ax = plt.subplots()
79     scan3.plot(ax,display_seg=True,display_th=True,display_tile=True)
80     plt.title(scan3.name)
81     plt.title(f'{scan3.name}\n[{proc}/{tot}]')
82     plt.show()
83 # ----->
84 # Step 4- display the Scan
85 if False:
86     scan3 = Scan(cmid)
87     scan3.read()
88     print(scan3)
89     # invalidate a specific tile for rework
90     # scan3.tiles[65].processed = False
91     # scan3.tiles[65].Segments=[]
92     # scan3.save()
93     (proc, tot) = scan3.get_tiles_processed()
94     fig,ax = plt.subplots()
95     scan3.plot(ax,display_seg=True,display_th=True,display_tile=True)
96     plt.title(scan3.name)
97     plt.title(f'{scan3.name}\n[{proc}/{tot}]')
98     plt.show()
```

Python code used for vectorisation of the scanned seismograms

Region of Interest: Automatic Detection (slide 1/2)

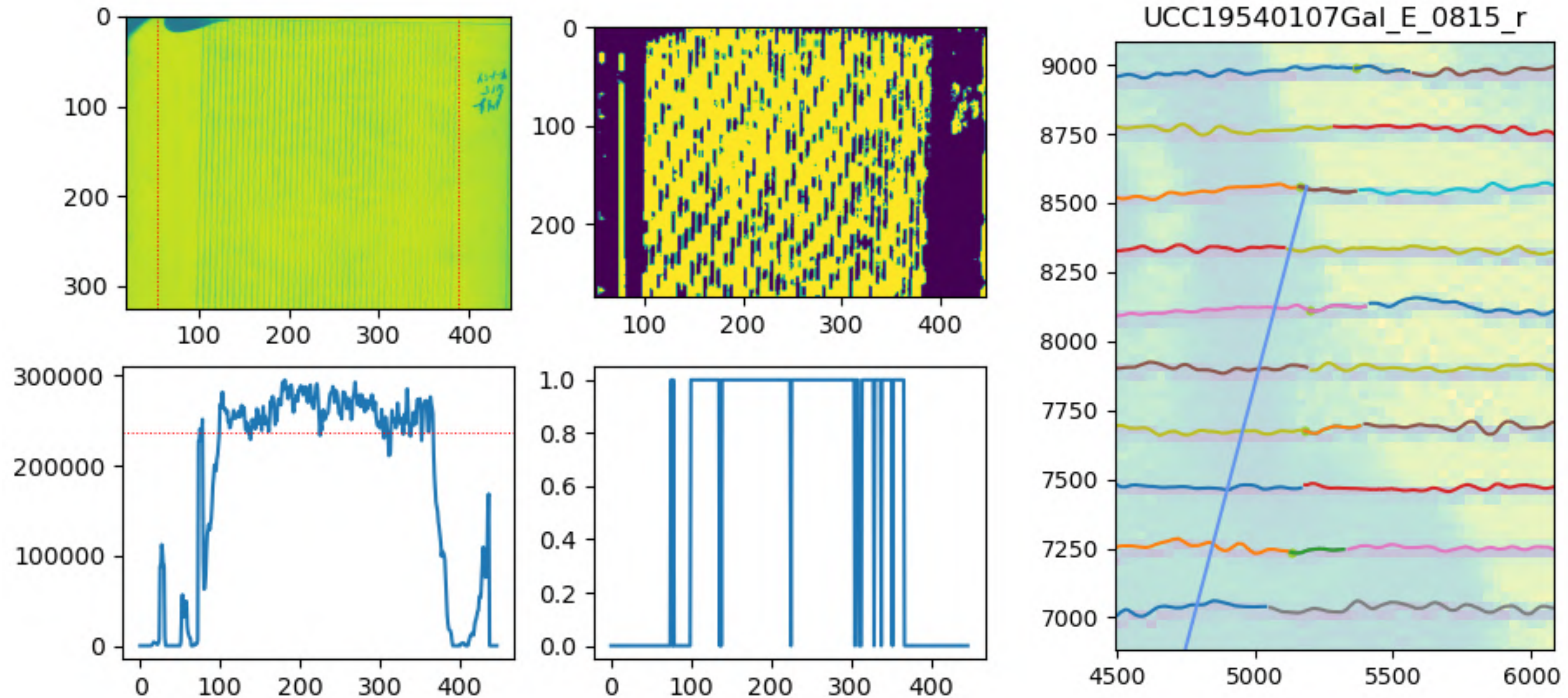


ROI detection was performed using setup of threshold for contrasting pixels on the images. As a result, the mask only included ROI between the red dashed lines (upper left image). The histograms show the value of pixels excluded from the ROI (those above the red line on the graphs).

It is possible to process images in Python both in horizontal and in vertical orientation (image on the right)

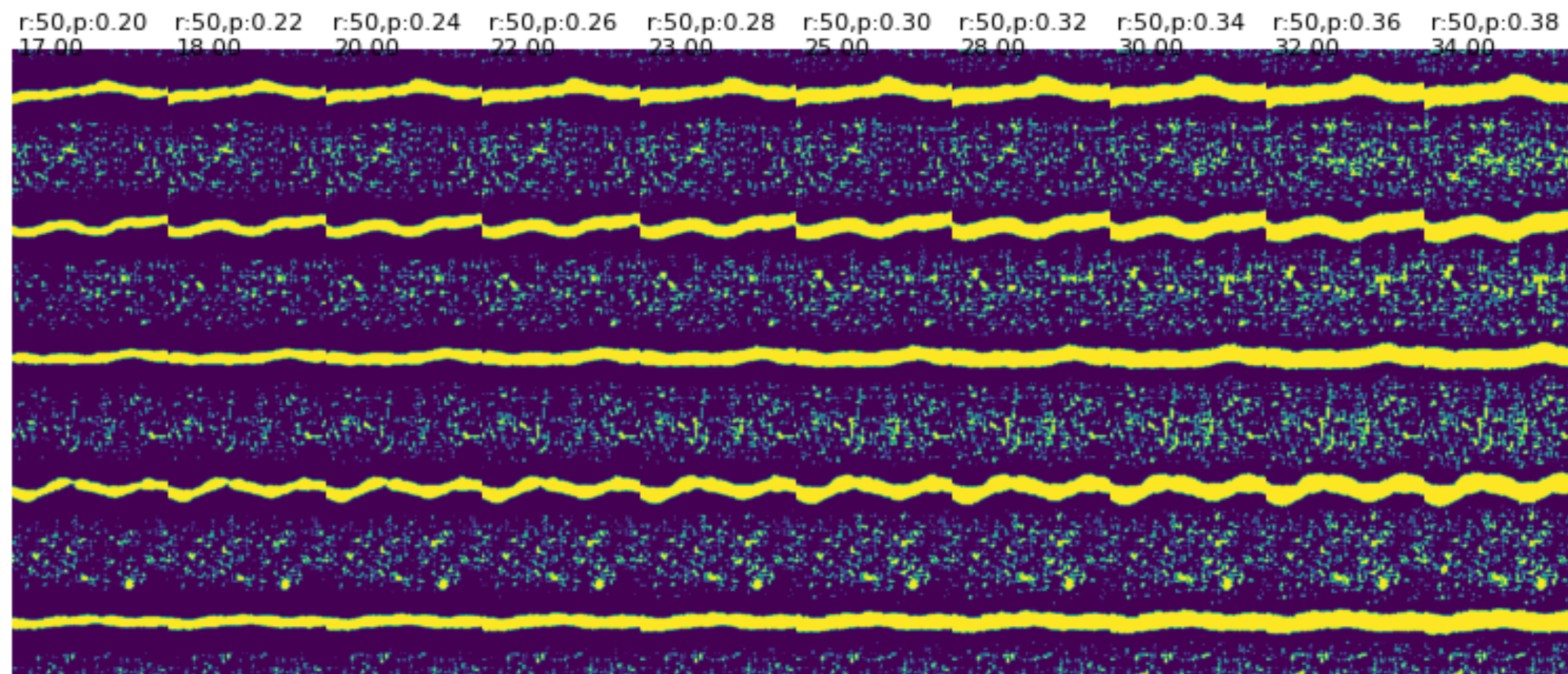
Workflow for vectorising in Python, Matplotlib library (slide 5/10)

Region of Interest: Automatic Detection (slide 2/2)



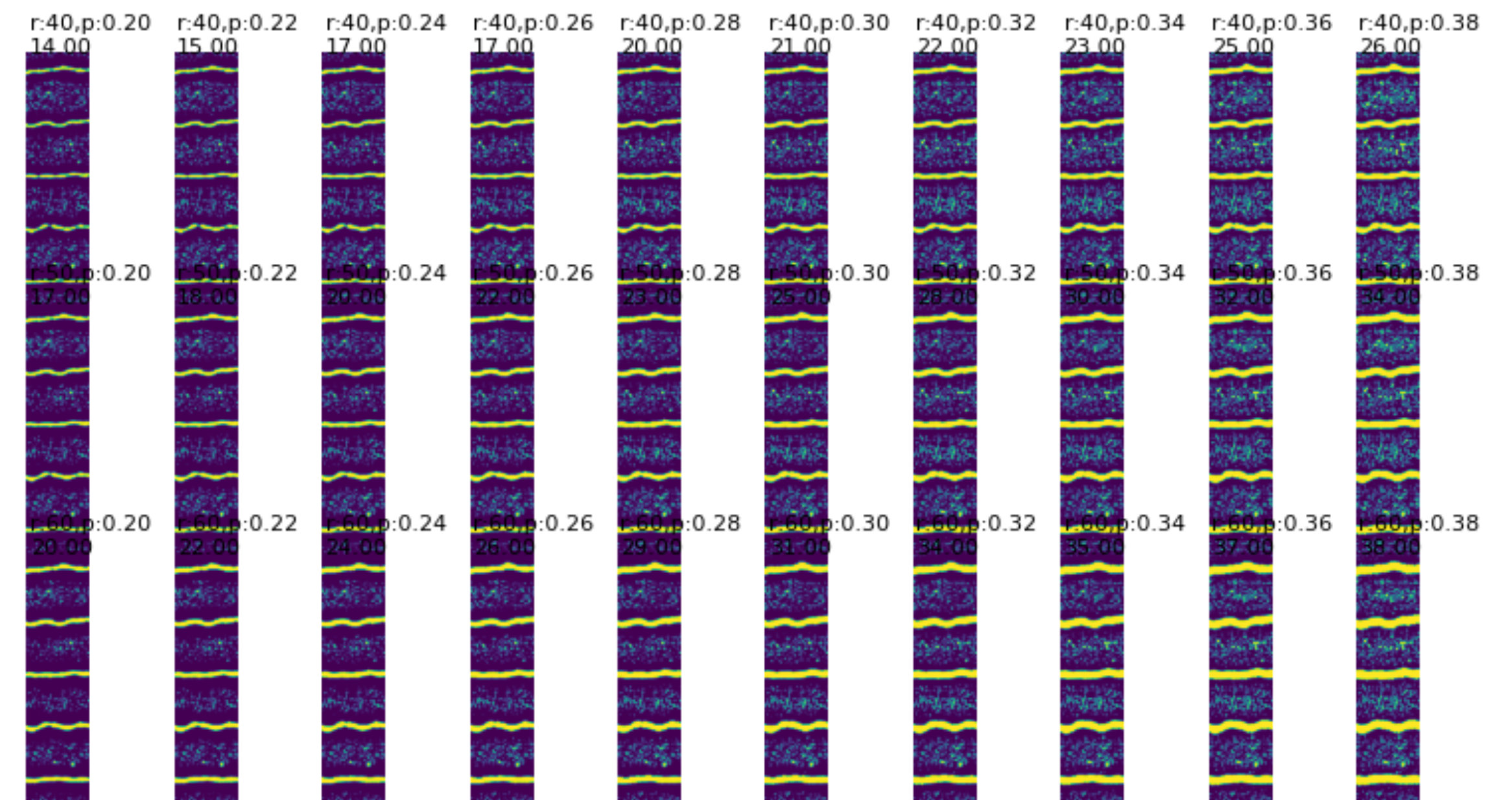
Defining ROI (between the red dashed lines) and enlarged fragment. Below: 2 histograms showing the distribution of pixels and those deleted (above the red dashed line). Right: enlarged fragment of the digitised seismogram. Workflow for vectorising in Python, Matplotlib library (slide 6/10).

Defining optimal parameters for the line thickness and radius of pixels (1)

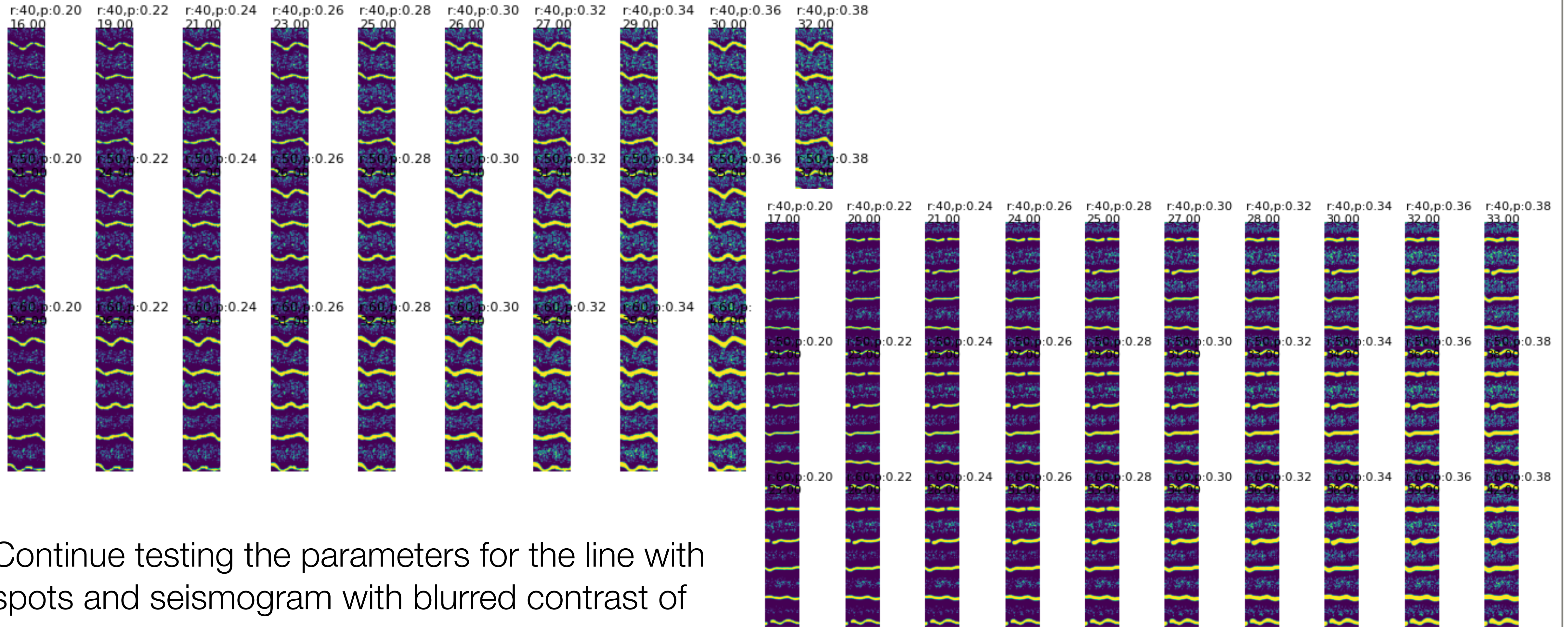


The thickness of the trace lines was defined by a series of trial tests with varied parameters. Radius of 30 pixels was defined as the optimal for the given image (it may vary through in other cases). Above: image with tested line thickness from 17 to 34 and radius of 50.

Below: image with tested thickness of the trace line from 14 to 26 pixels (upper row) and 20 to 38 pixels (lower row) and radius of 40, 50 and 60 for each corresponding row (downwards). Changed thickness of line is visible in all trial cases (yellow-coloured horizontal lines).



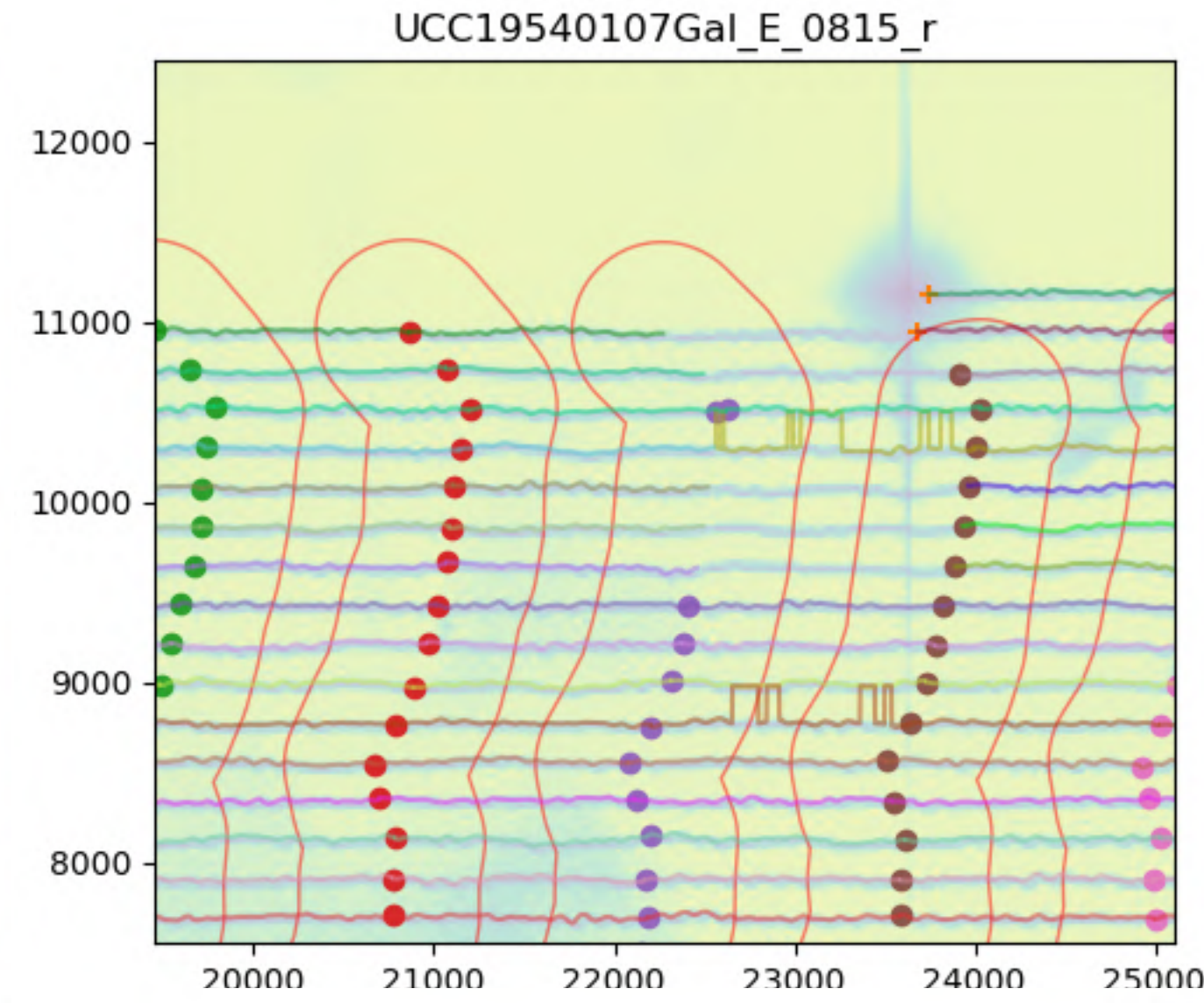
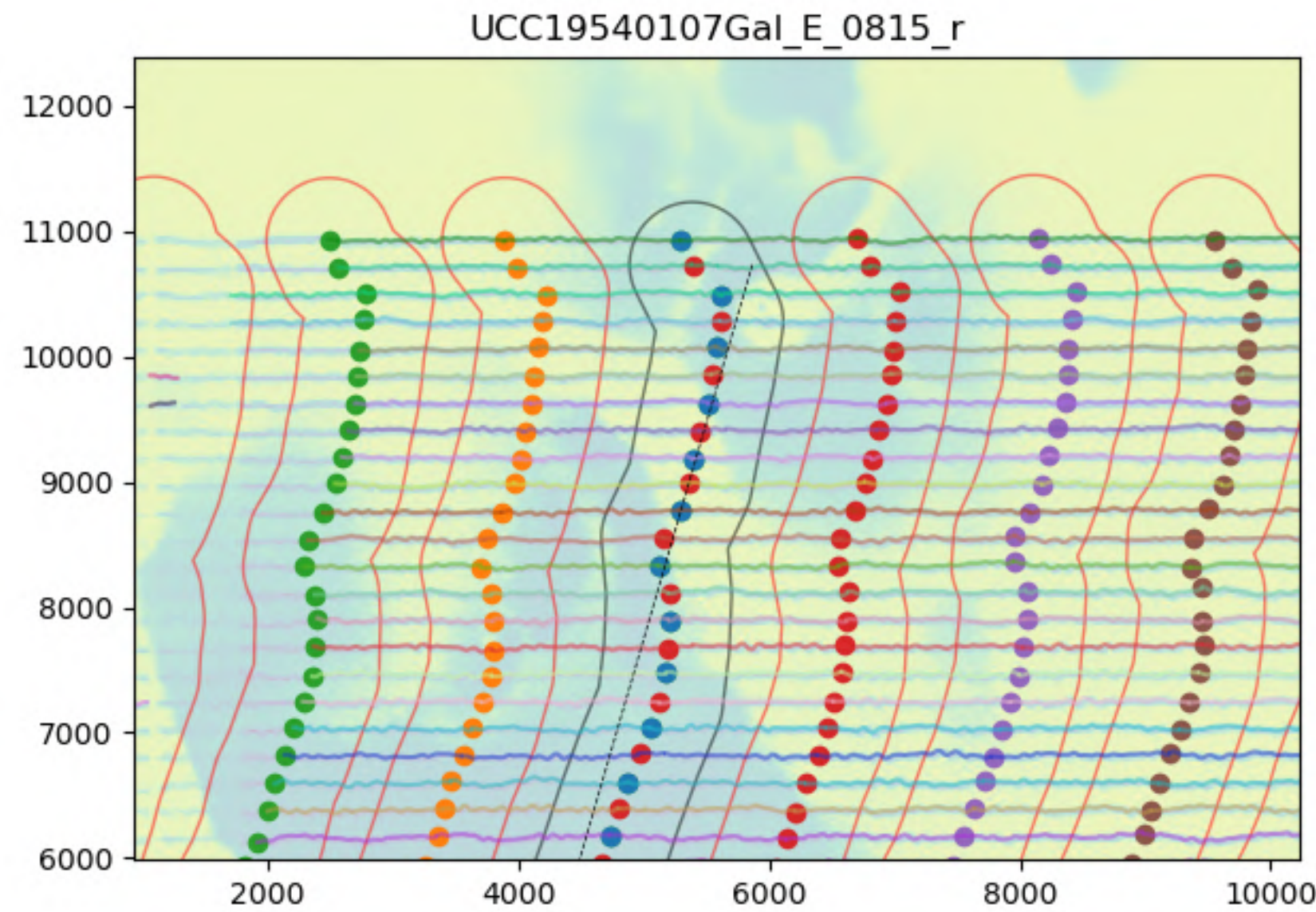
Defining optimal parameters for the line thickness and radius of pixels (2)



Continue testing the parameters for the line with spots and seismogram with blurred contrast of lines against the background

Workflow for vectorising in Python, Matplotlib library (slide 8/10)

Buffering parameters for the time gaps

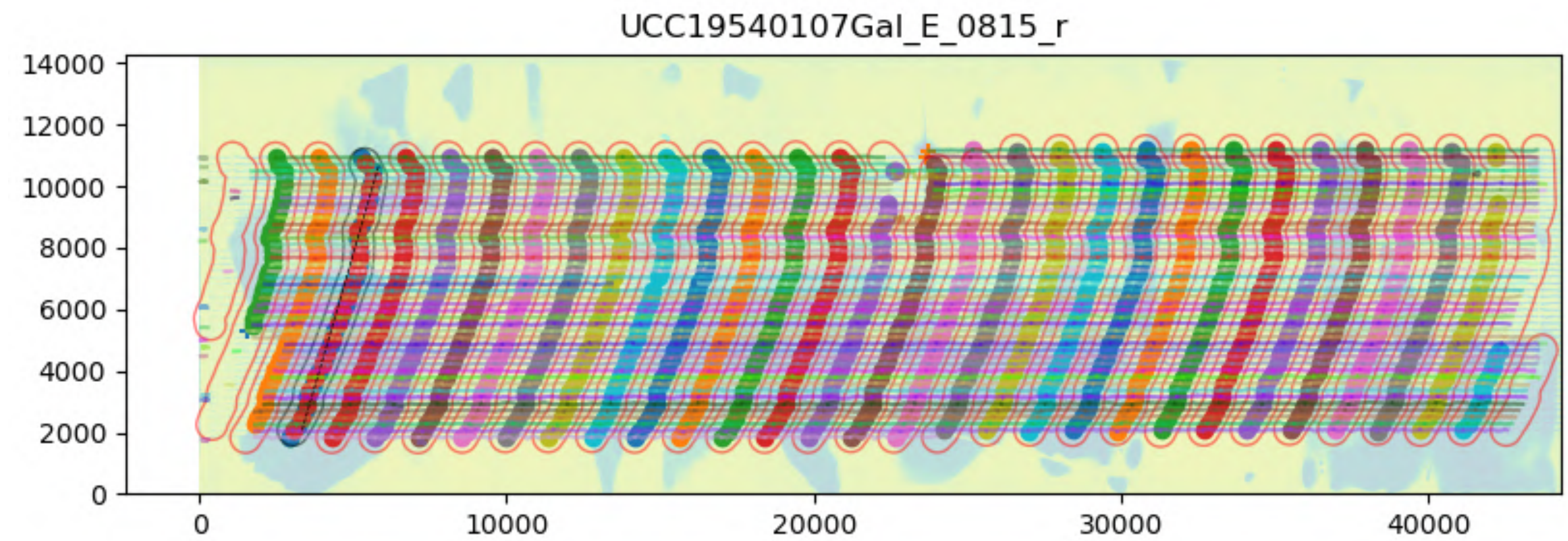


Buffering minute intervals for the one-minute gaps completed for the whole seismogram;

Buffering of missing data: minute and hour gaps

Buffering minute intervals for the one-minute gaps completed for the whole seismogram

The vectorised minute segments in seismograms were randomly coloured for a better visual contrast.



Workflow for vectorising in Python, Matplotlib library (slide 9/10)

Python code for defining hour and minute marks on the seismograms

```
1 # select hour marks only (distant from the tile border)
2     valid_h = (length > self.param.HOUR_WIDTH * 0.8) * \
3               (length < self.param.HOUR_WIDTH * 1.1) * \
4               (props['bbox-0'] > 15) * \
5               (props['bbox-1'] > 15) * \
6               (props['bbox-2'] < m) * \
7               (props['bbox-3'] < n)
8 # select minute marks only (distant from the tile border)
9     valid_m = (length > self.param.MINUTE_WIDTH * 0.9) * \
10             (length < self.param.MINUTE_WIDTH * 1.1) * \
11             (props['bbox-0'] > 15) * \
12             (props['bbox-1'] > 15) * \
13             (props['bbox-2'] < m) * \
14             (props['bbox-3'] < n)
15 # select minute complement marks only (distant from the tile border)
16     valid_c = (length > self.param.PARTIM_HOUR_WIDTH * 0.9) * \
17             (length < self.param.PARTIM_HOUR_WIDTH * 1.1) * \
18             (props['bbox-0'] > 15) * \
19             (props['bbox-1'] > 15) * \
20             (props['bbox-2'] < m) * \
21             (props['bbox-3'] < n)
```

The time gaps (hour and minute marks) were identified by dividing the line into repetitive segments with regular intervals.

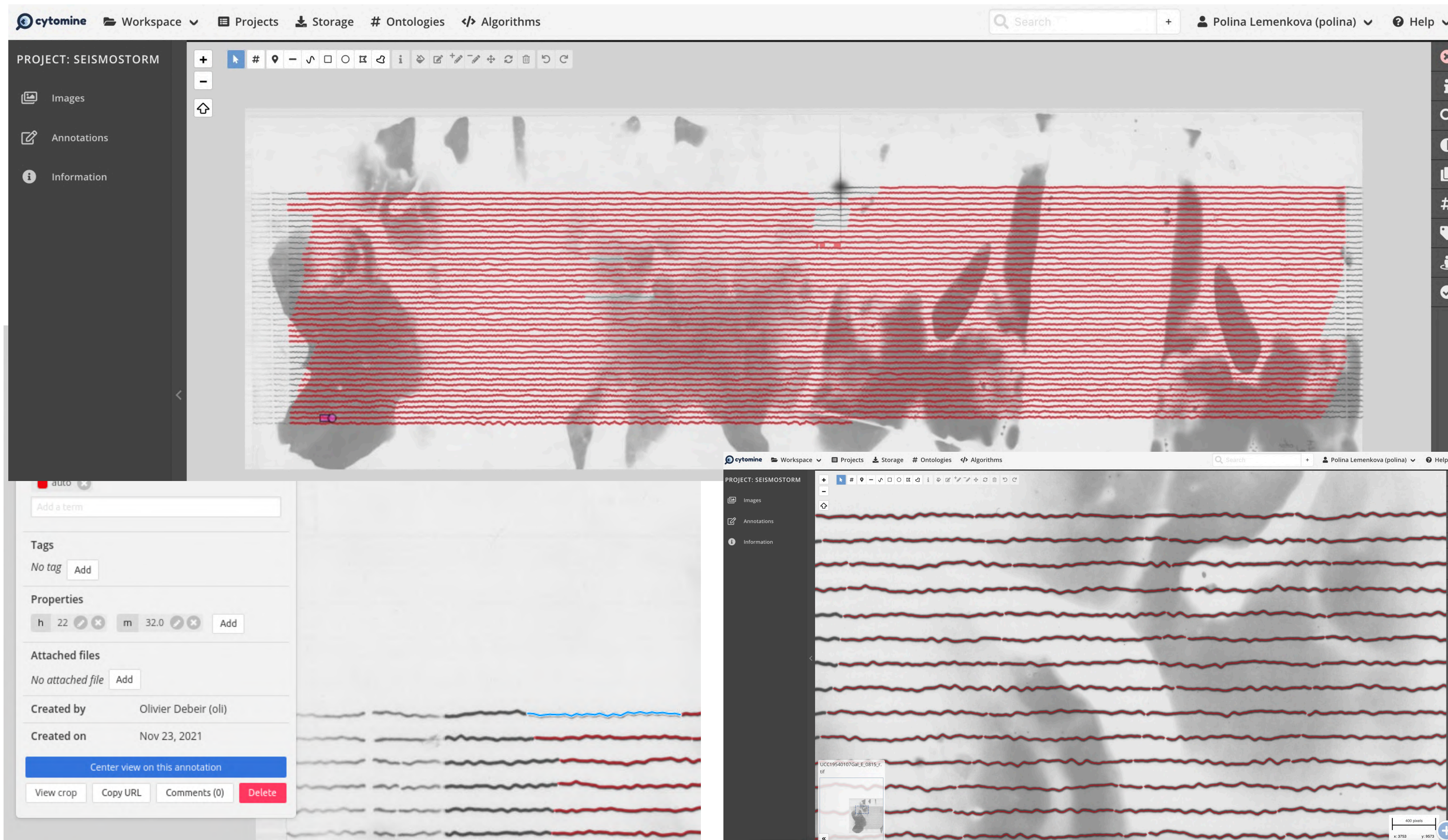
These intervals of the time gaps are identified by the code presented left.

The shape of the line as a vector geometric structure is recognised and identified by the machine vision.

The approach is based on the assessment of the connectivity of pixels constituting the line, expect for the time gaps and marks breaking the trace by minute and hour marks.

Code snippet showing algorithm for identifying the time gaps in Python (© O.Debeir).

Seismogram vectorised by Python overlain on the original image and uploaded in Cytomine

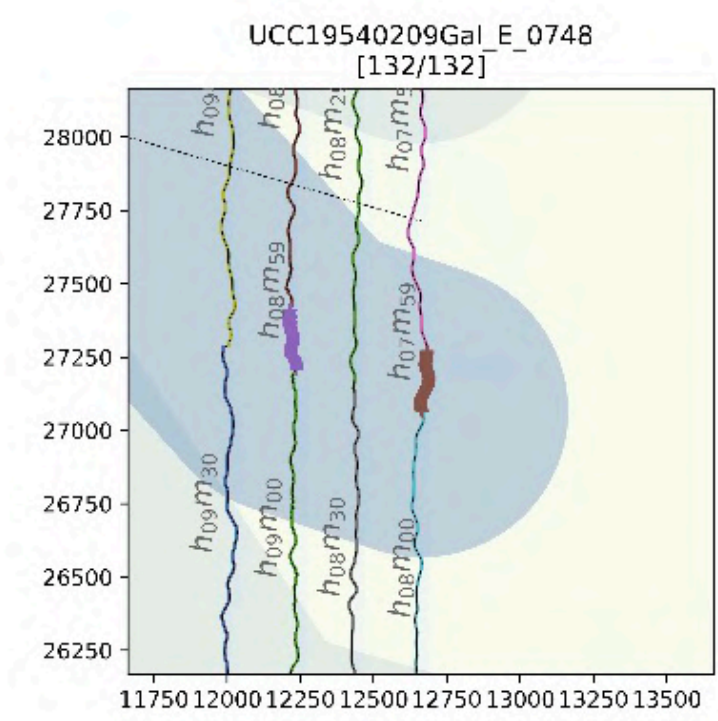


Example of the vectorised trace segments (red lines) overlaid on the spotted image

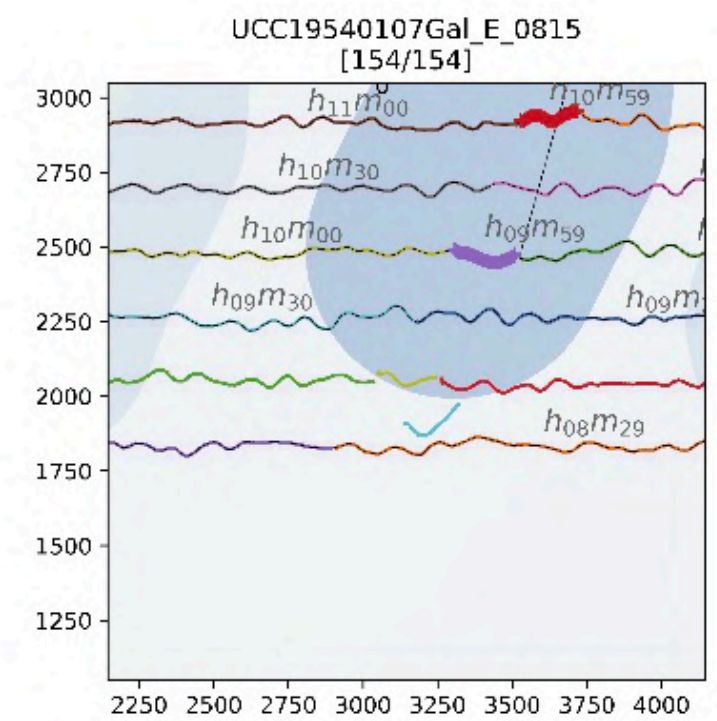
Enlarged fragment with visible distinct traces;

Enlarged fragment with visible time gaps

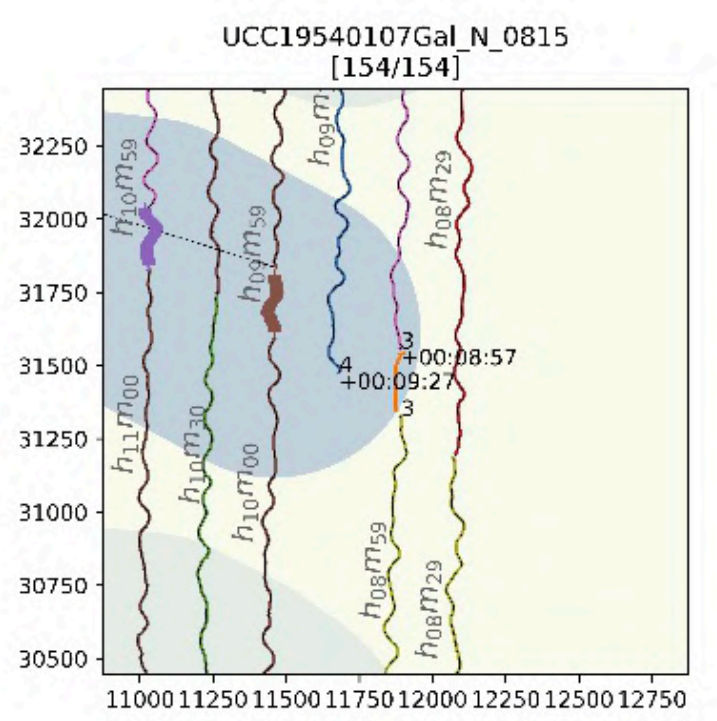
Workflow for vectorising in Python, Matplotlib library (slide 10/10)



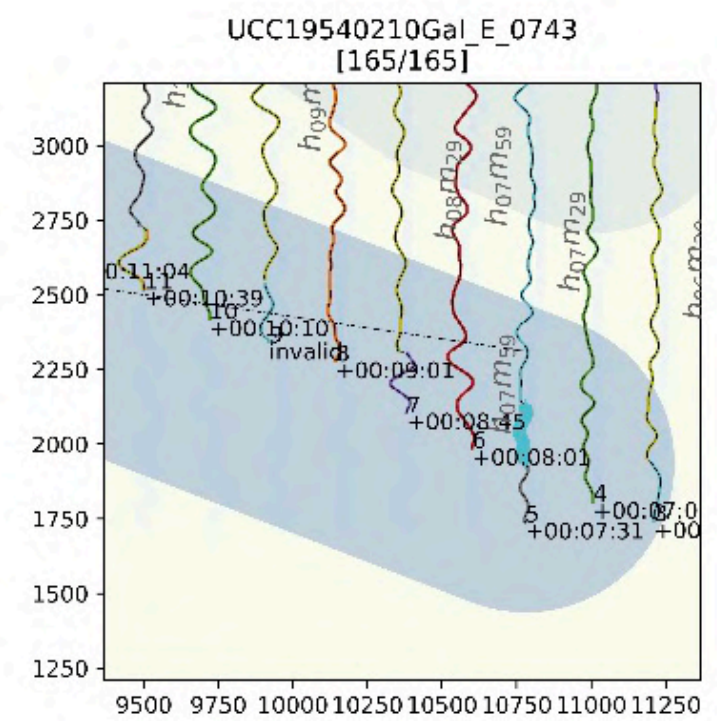
(a)



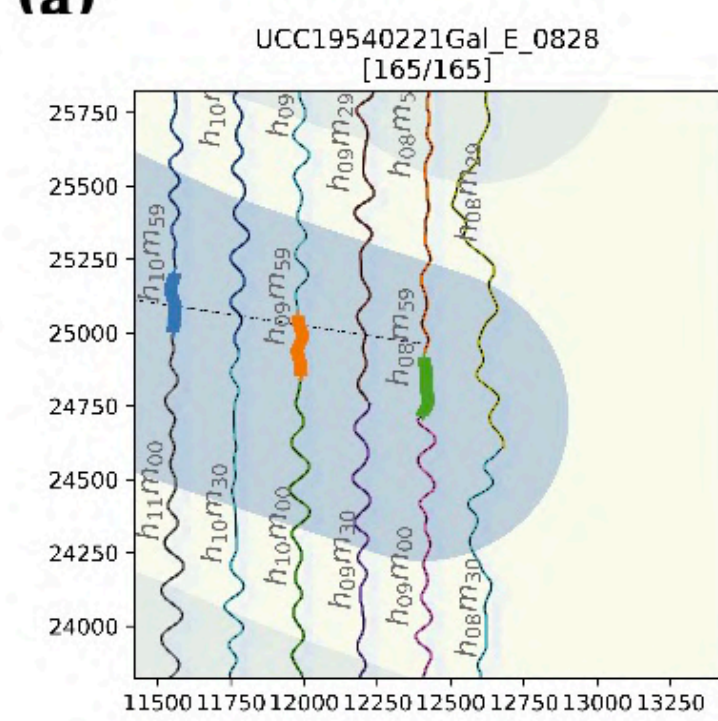
(b)



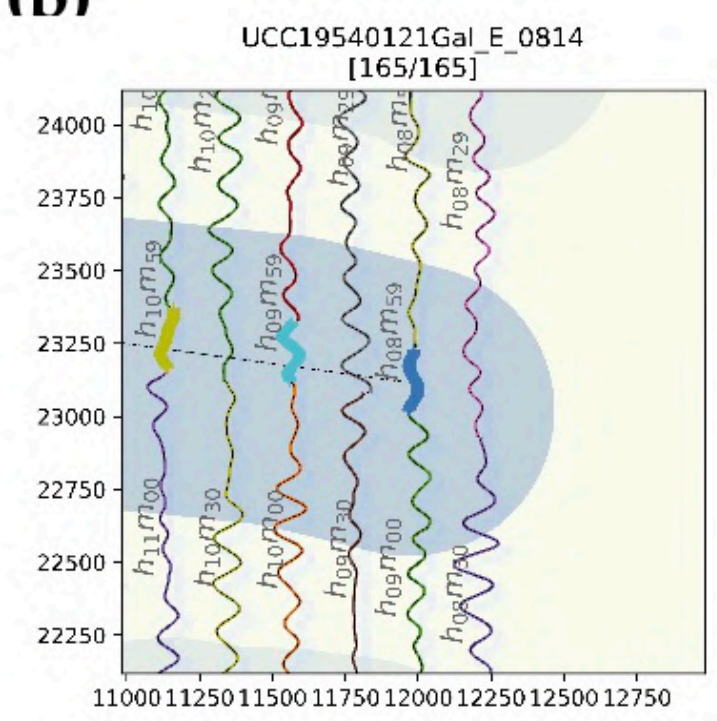
(c)



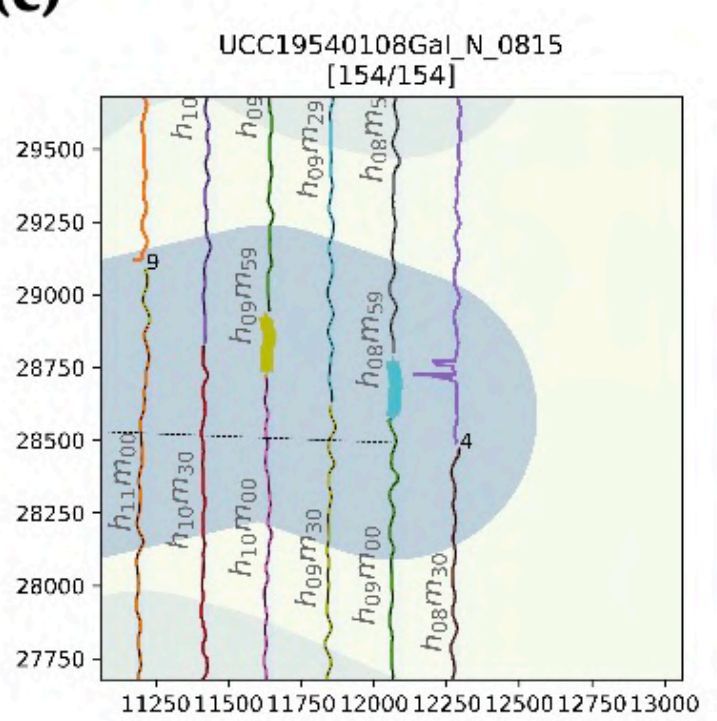
(d)



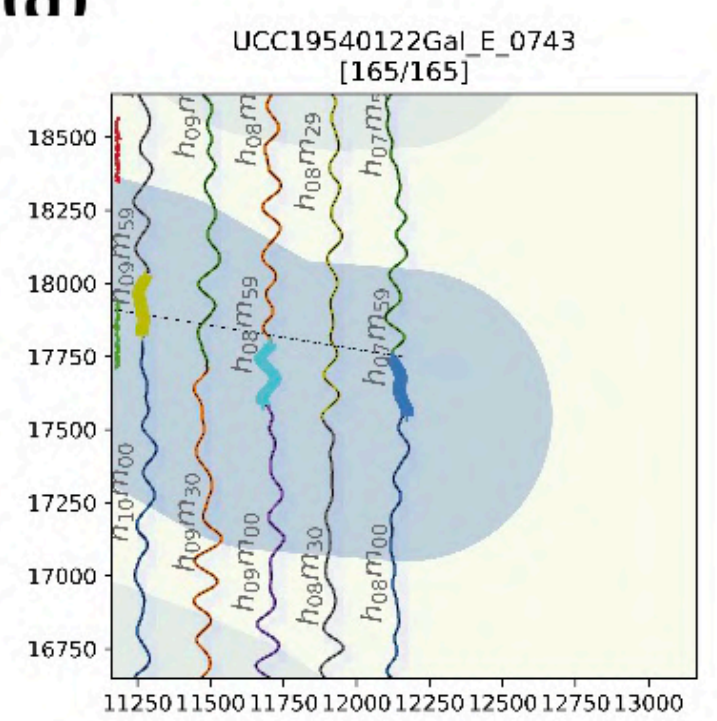
(e)



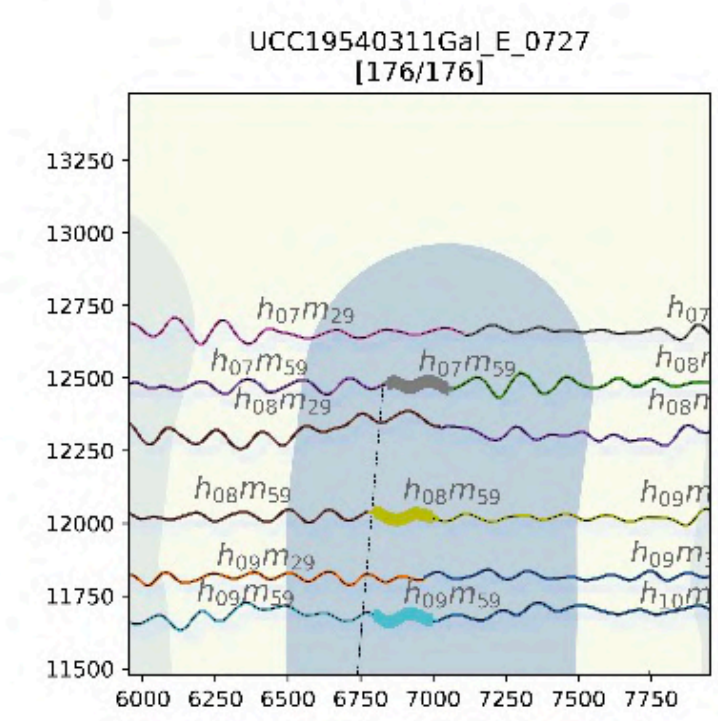
(f)



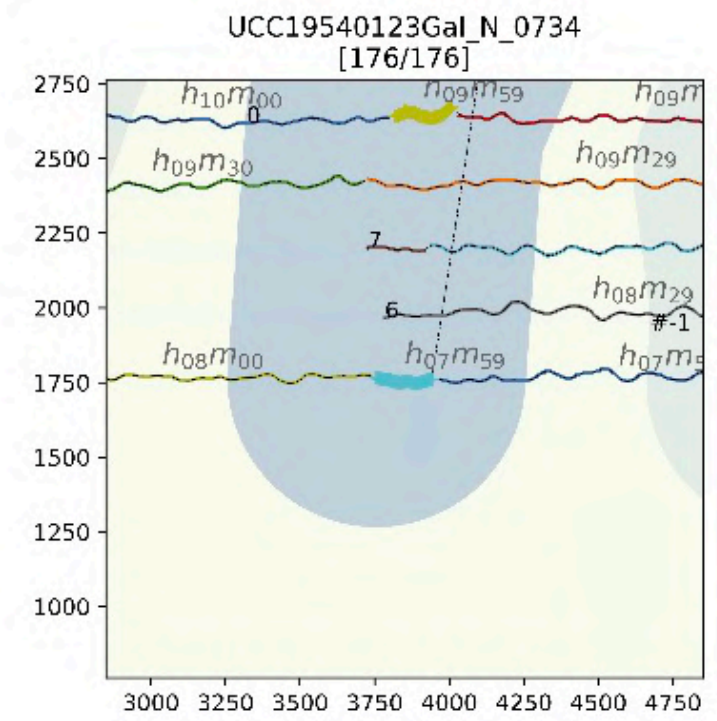
(g)



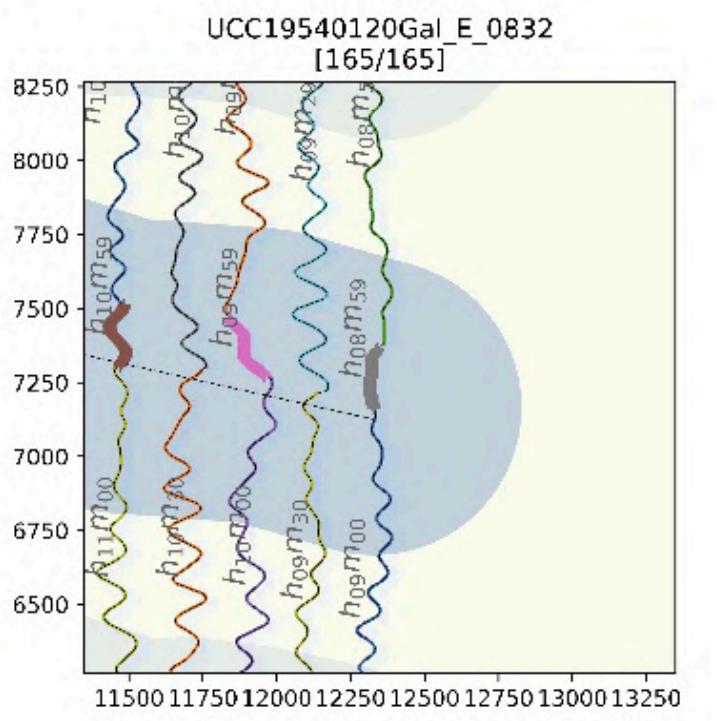
(h)



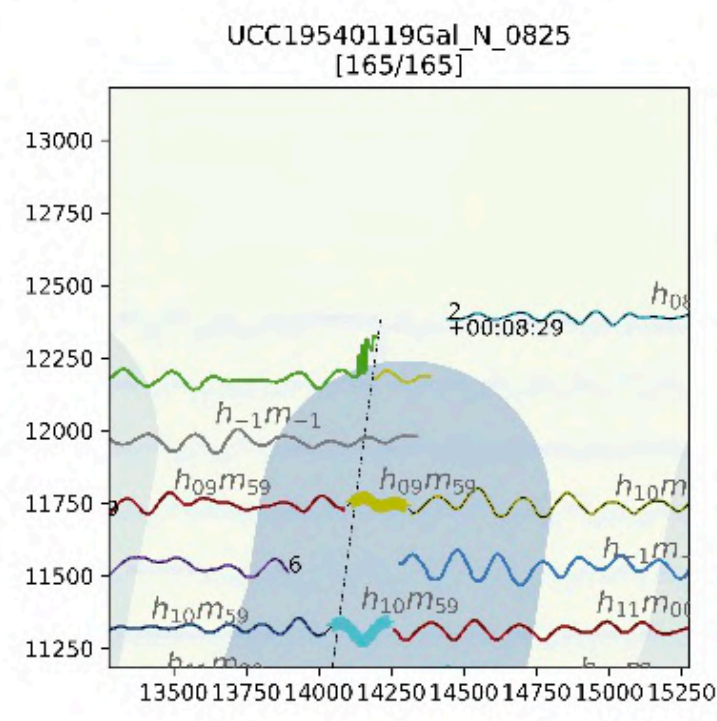
(i)



(j)



(k)



(l)

Fragments of the images generated by *seismo 0.1.0-alpha* software.

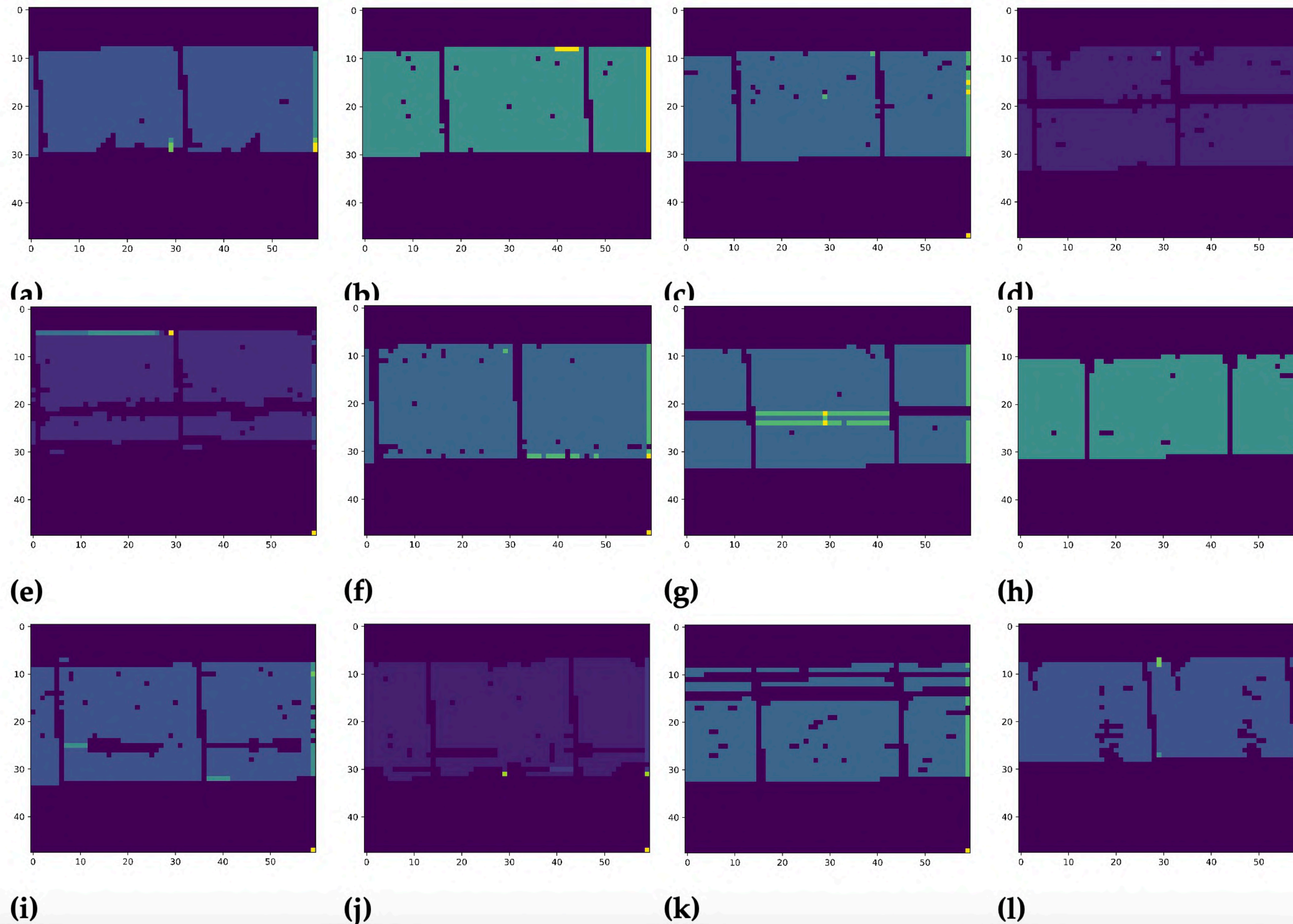
Timing of seismogram tiles with buffer zone (slate gray colour), hour marks (thick short lines coloured randomly) and minute segments (thin lines coloured randomly).

The hour marks and minutes are annotated on the graphs, e.g., 'h09m59' means hour 9 minute 59.

Cytomine IDs: a) 4433; b) 5765; c) 5779; d) 14485; e) 14391; f) 7759; g) 5673; h) 7765; i) 5660; j) 7795; k) 7747; l) 9417.

Processing image is possible both horizontal and in vertical mode.

Quality control of the seismogram processing (selected examples)



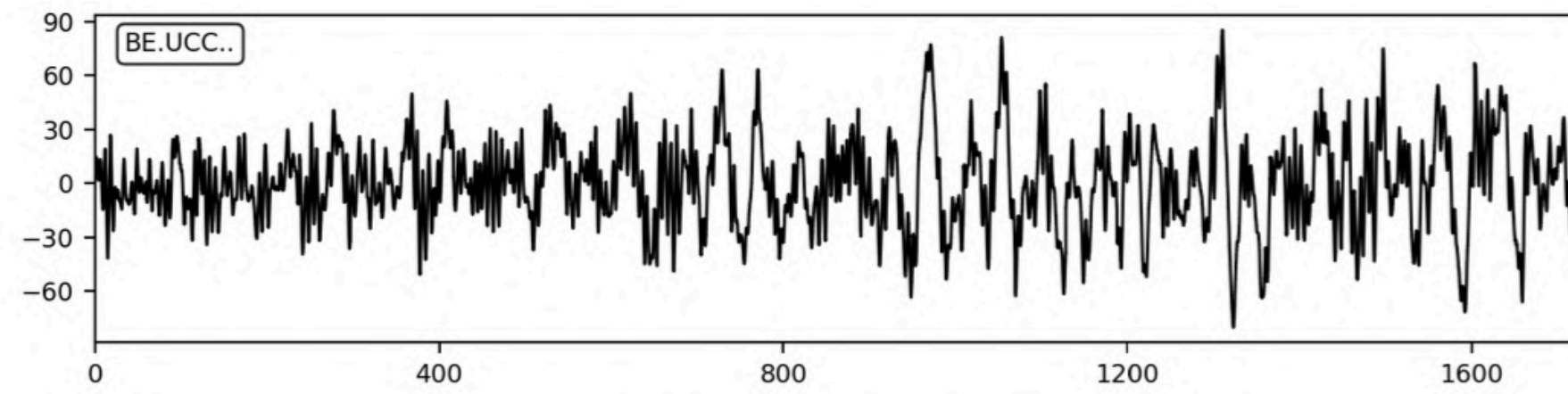
Dark purple colour signify the background with no data.

Navy blue (or aquamarine) colours in the middle of the images signify the successfully digitised traces.

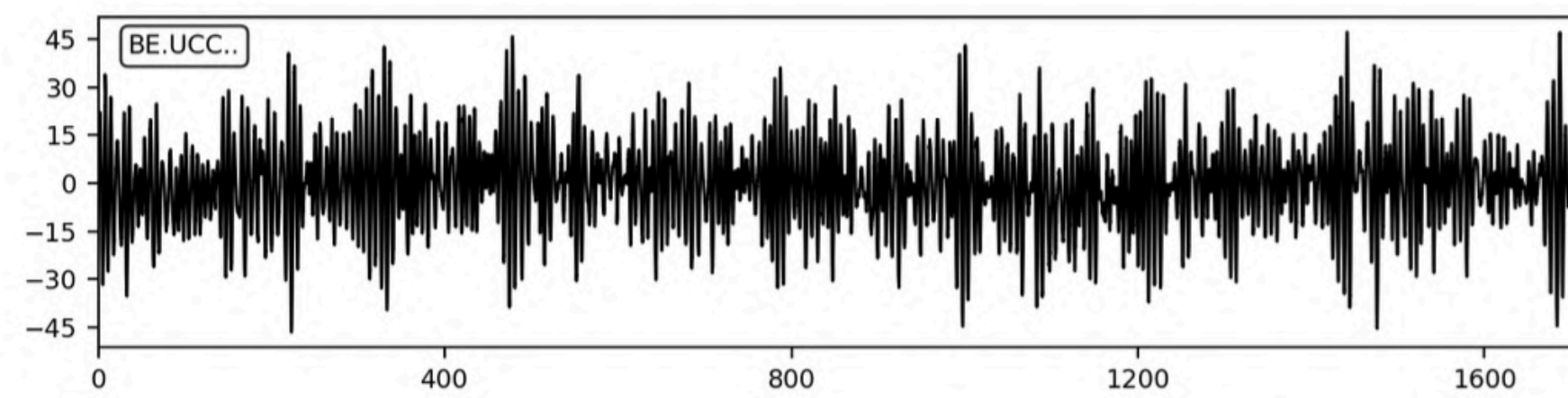
Vertical slanted lines crossing the main image mean the hour marks.

Yellow occasional pixels signify the errors and noise.

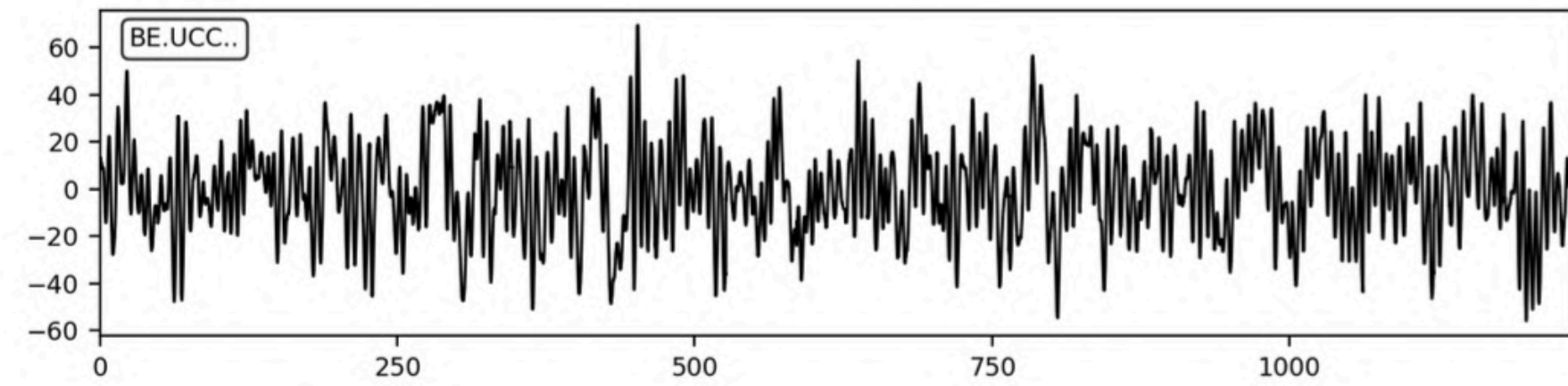
Cytomine IDs of the images: a) 5765; b) 7801; c) 10864; d) 14418; e) 14485; f) 16297; g) 16353; h) 17014; i) 17028; j) 17036; k) 19054; l) 1245452.



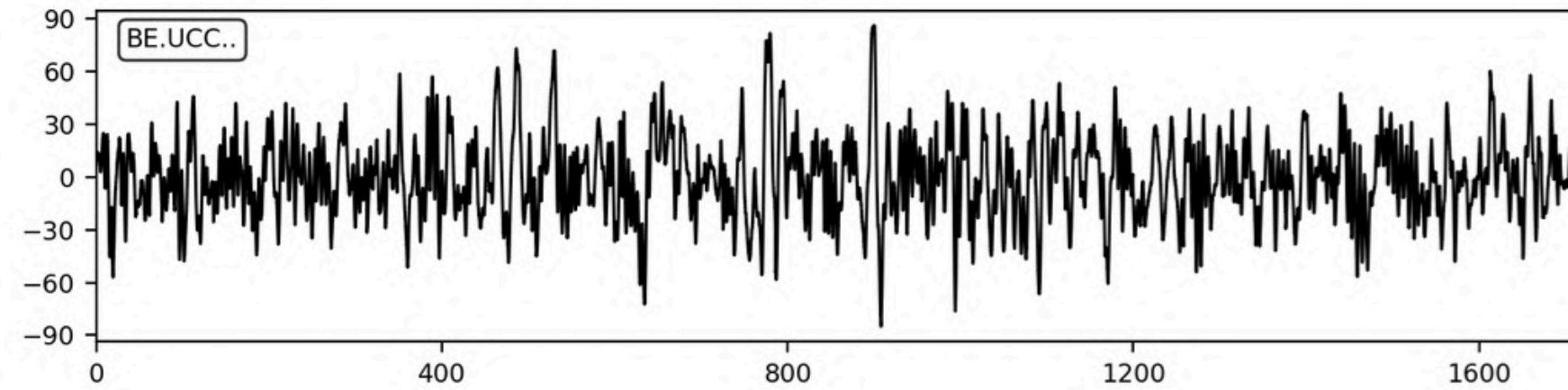
(a) traces_UCC19540224Gal_E_0800.mseed
Time in seconds relative to 1954-02-13T01:42:00



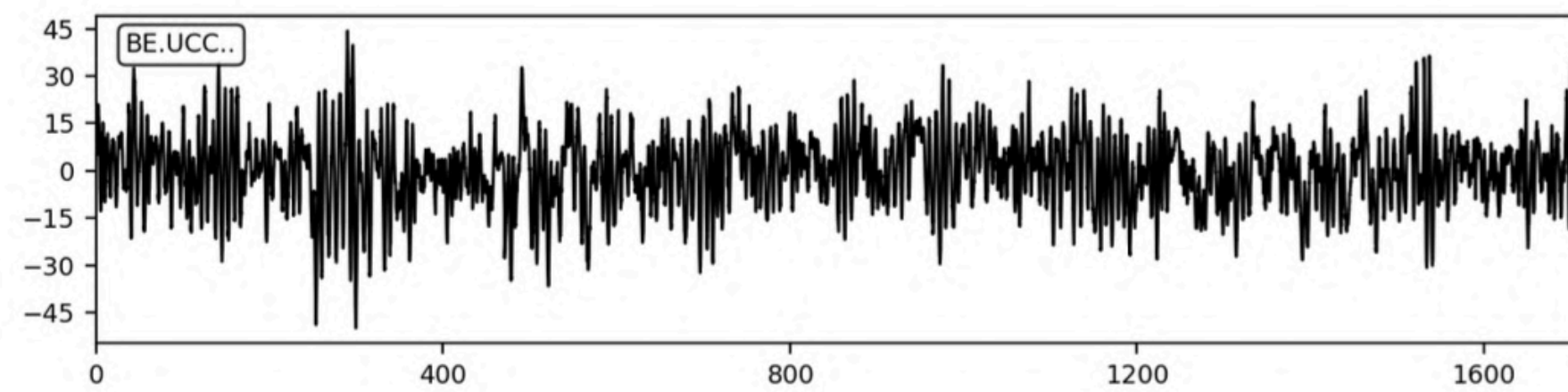
(b) traces_UCC19540310Gal_N_0907.mseed
Time in seconds relative to 1954-03-03T19:48:00



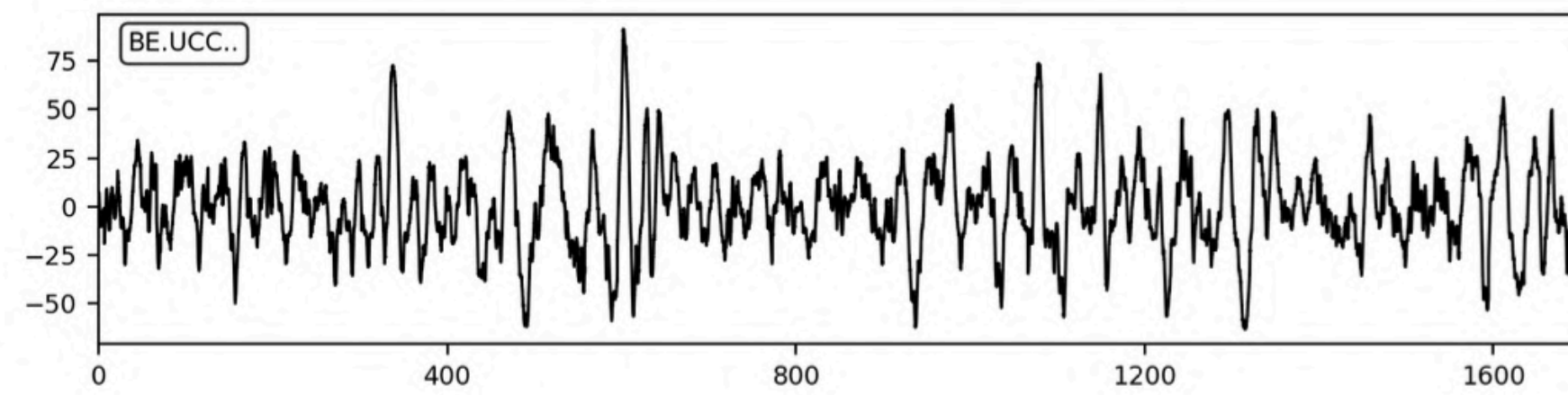
(c) traces_UCC19540212Gal_E_0758.mseed
Time in seconds relative to 1954-02-05T13:06:00



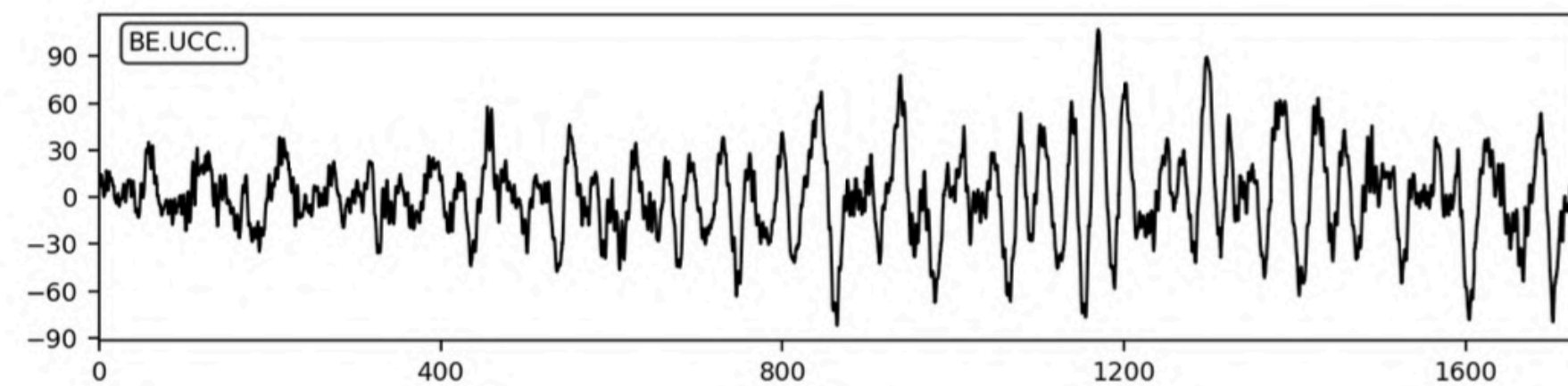
(d) traces_UCC19540303Gal_N_0833.mseed
Time in seconds relative to 1954-02-03T11:56:00



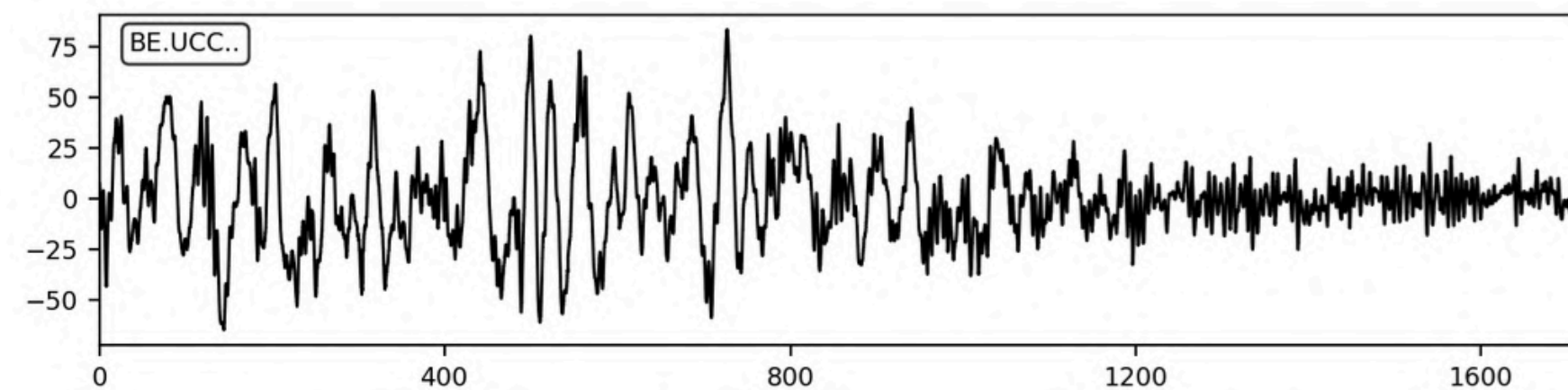
(e) traces_UCC19540205Gal_N_0742.mseed
Time in seconds relative to 1954-03-02T09:10:00



(f) traces_UCC19540203Gal_N_0821.mseed
Time in seconds relative to 1954-03-10T01:51:00

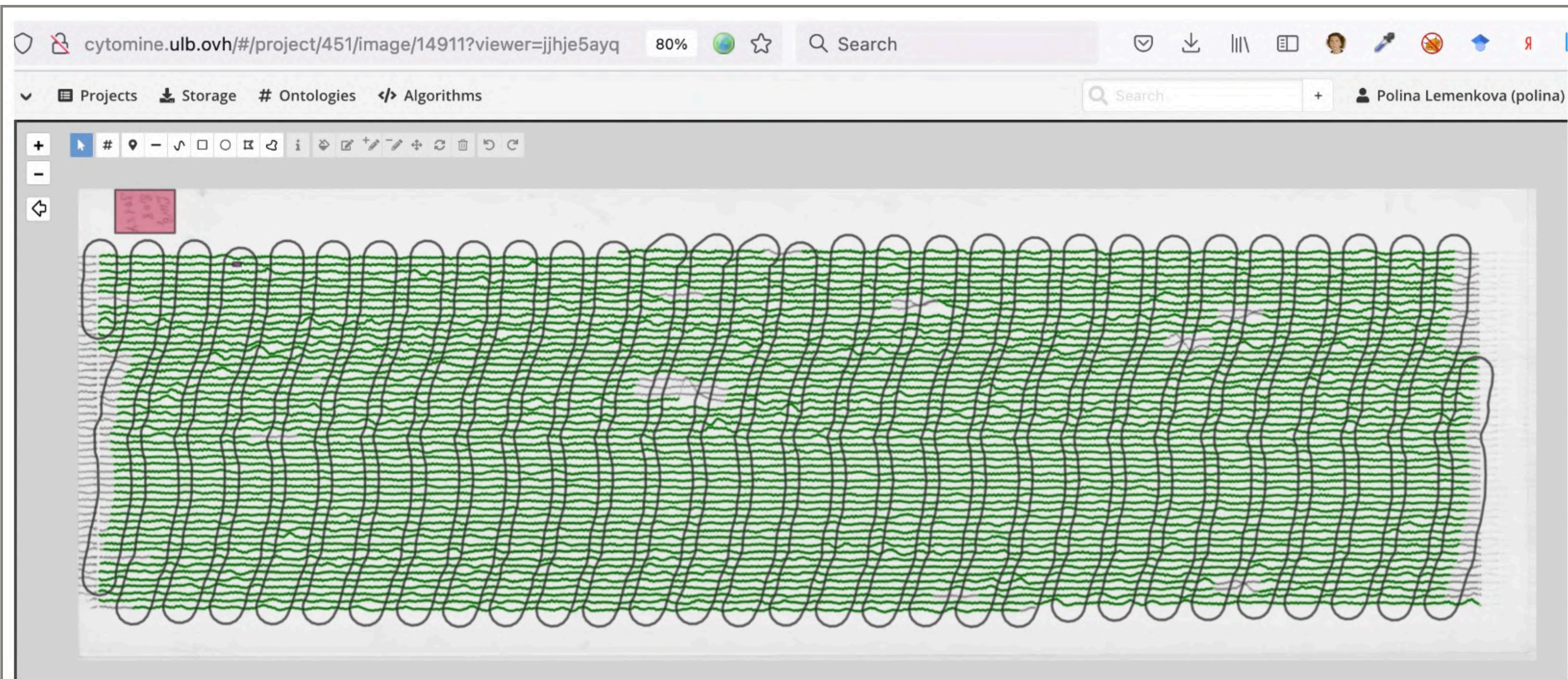


(g) traces_UCC19540302Gal_E_0805.mseed

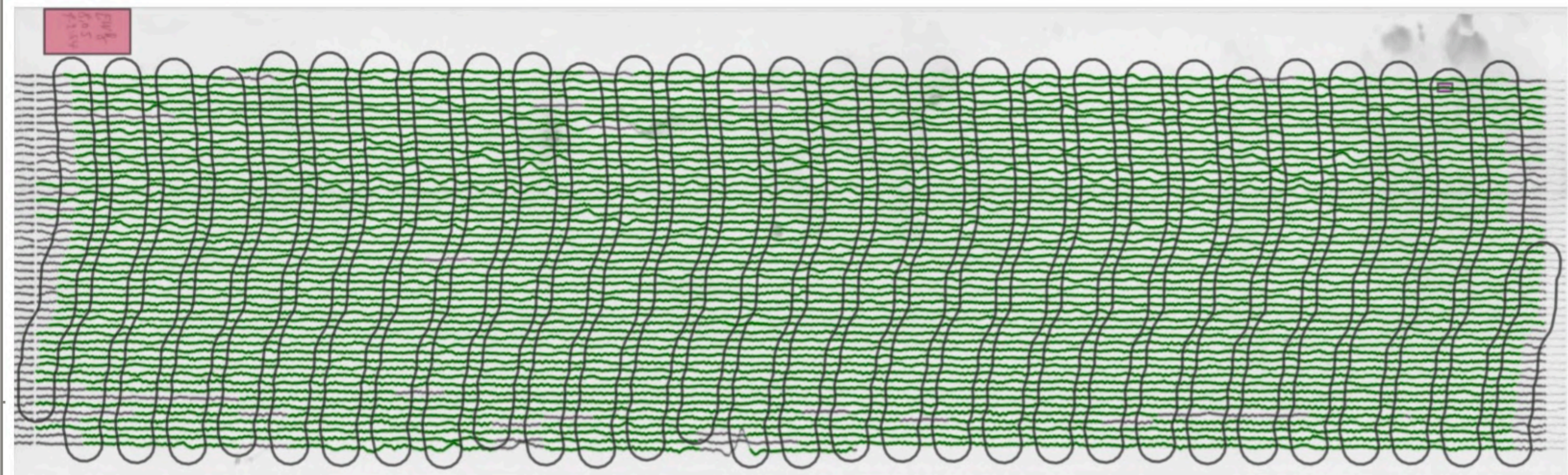


(h) traces_UCC19540309Gal_E_0817.mseed

Results (1):
Examples of the enlarged vectorised seismogram 1-minute traces processed and digitised by the Python algorithm of the *seismo 0.1.0-alpha* software developed by O. Debeir.
The vectorised data are visualised in the ObsPy library of Python using the MSEED files generated by *seismo 0.1.0-alpha*.



(a)



(b)

Results (2):

Seismogram vectorised by Python using our novel approach (green lines) overlaid on the original raster image (grey lines) and uploaded into Cytomine workspace.

(a) Image

UCC19540130Gal_E_0808
(Cytomine ID 14911).

(b) Image

UCC19540201Gal_E_0805,
Cytomine ID 16297.

Results (3):

The dataset of the raw TIFF seismograms is available in the shared repository in Zenodo <https://doi.org/10.5281/zenodo.7245119> with samples openly available for download, processing and data reuse.

It includes selected scanned seismogram images for the period of 1954 and enables the access to seismic waveform data. The original images included vertically and horizontally oriented seismograms. The images are in TIFF format and contain the original scanned seismograms from ROB data collections.

The dataset contains 45 files of seismic recordings received from the Galitzin seismometer. Image can be used for vectorisation and detection of seismic signals using presented Python workflow model.

The screenshot shows the FDSN website interface. At the top, there is a navigation bar with the FDSN logo and the text "International Federation of Digital Seismograph Networks". Below this, a breadcrumb trail reads "Home / Networks / BE: Belgian Seismic Network". The main heading is "BE: Belgian Seismic Network" with a sub-heading "FDSN Network Information" and a link "Are you the operator of this network? Update this information.".

FDSN code	BE	Network name	Belgian Seismic Network
Start year	1985	Operated by	Royal Observatory of Belgium ROR::
End year	-	Deployment region	
Description	Belgian seismic network, operated by the Royal Observatory of Belgium. The data from the superconducting gravimeters in operation at MEM and RCHB are available with network code SG: https://doi.org/10.7914/SN/SG		
Website	https://seismology.be		

Citation Information

Digital Object Identifier (DOI)	10.7914/SN/BE
Citation	Royal Observatory of Belgium. (1985). <i>Belgian Seismic Network</i> [Data set]. International Federation of Digital Seismograph Networks. https://doi.org/10.7914/SN/BE For more: DataCite (JSON XML BibTeX)

Data Access

Data Availability	Data available from: The ORFEUS Data Center (ORFEUS) : http://www.orfeus-eu.org/fdsnws/dataselect/1/ The IRIS Data Management Center (IRISDMC) : http://service.iris.edu/fdsnws/dataselect/1/
-------------------	---

Actuality and Application



Data Processing



ML Algorithms



Users



Motivation Challenge of big data in seismic studies: massive volumes of historical seismograms from ROB exist and present a source of information. Archive old data must be processed, digitised and ‘revitalised’.

Contribution This project addresses the challenges of vectorising the old seismograms which revitalise the existing archives by R2V algorithms using ML methods.

Methods Our project focuses on developing automated ML methods of vectorising seismograms with minimised human interaction and maximised programming approach in trace vectorisation.

People End-users (seismologists) will benefit from our project which includes archiving and processing data, developed Python-based algorithms and vectorised seismograms for interpreting the results.

Other activities during my research in ULB:

- Supervision of the dissertation of the Master of Science (MSc) student Alexandre Missenard with topic overlapping to mine: *'Exploitation of EQTransformer and application to Belgian seismic data'* (10 ECTS).
- Presentation at LISA with report on current research progress on December 10, 2021.
- Attended two consecutive French *in-class* courses “**French language for foreigners**” in the *ULB Langues Faculty*:
 1. **LANG-B909**, niveau B2.2, jan-may 2022 (5 ECTS);
 2. **LANG-B910** niveau C1, sep-dec 2022 (5 ECTS).



Résumé

My activities in ULB during 2021/2022 and 2022/2023 academic years:

- **Conference paper (1):** De Plaen, R. S. M.; Lecocq, T.; Lemenkova, P. ; Debeir, O.; Ardhuin, F.; De Carlo, M. Extracting Microseismic Ground Motion From Legacy Seismograms. *In: Proceedings of the Third European Conference on Earthquake Engineering and Seismology*, 2022-09-04: Bucharest, Romania. Conspress, Ed. 1, pp. 3507-3513. Publié, 2022-09-09. <https://doi.org/10.5281/zenodo.7064711> **(5 ECTS);**
- **Journal article (1):** Lemenkova, P.; De Plaen, R.; Lecocq, T.; Debeir, O. Computer Vision Algorithms of DigitSeis for Building a Vectorised Dataset of Historical Seismograms from the Archive of Royal Observatory of Belgium. *Sensors* **2023**, 23, 56. <https://doi.org/10.3390/s23010056> **(10 ECTS);**
- **Journal article (2) – submitted:** Lemenkova, P.; De Plaen, R.; Lecocq, T.; Debeir, O. A Python-based framework for automated vectorisation of the analog seismograms recorded in Uccle seismic station, Belgium. **2023 (expected 10 ECTS);**
- **Presentation (1)** at LISA, ULB with report on current research progress on *December 10, 2021*: “*Vectorising analog seismograms by techniques of machine learning for automated discriminating of seismic signal traces*” **(5 ECTS);**
- **Course (1):** LANG-B909, niveau B2.2: spring semester 2022 (jan-may) **(5 ECTS);**
- **Course (2):** LANG-B910 niveau C1: autumn semester 2022 (sep-dec) **(5 ECTS).**
- **Teaching:** Supervision of the dissertation of the MSc student Alexandre Missenard **(10 ECTS)**
- **Reviews:** my Web of Science (WoS) profile: <https://www.webofscience.com/wos/author/rid/R-8828-2018> **(10 ECTS)**

