This document is currently under revision by the European Commission (EC) and has not yet been validated or approved by the EC. The content provided herein is subject to change, and the information presented may not represent the final position or official stance of the EC.

This document is being shared for informational purposes only and is not to be considered an official or authoritative source of information from the European Commission. Any decisions, actions, or interpretations based on the content of this document should be taken with caution, as the content may be subject to modification or revision by the EC.

The EC accepts no liability for any inaccuracies, errors, or omissions in this document, and any reliance on its content is at the user's own risk. It is recommended to verify the information provided in this document with official EC publications or communications before making any decisions or drawing any conclusions based on its content.

Please note that the content in this document may be confidential or sensitive in nature and should be treated as such. Unauthorized dissemination, distribution, or use of this document is strictly prohibited.

By accessing and reviewing this document, you acknowledge and accept the terms of this disclaimer.

EIS
Exploration
Information
System

D 3.6: EIS QGIS Plugin Beta Version

User Manual and Technical Specifications

Version 1.0

Lead Beneficiary: GISPO
04 / 2024

Niko Aarnio[1], Timo Aarnio[1], Miikka Kallio[1], Mika Sorvoja[1], Johanna Torppa[2], Bijal Chudasama[2], Ina Storch[3], Peggy Hielscher[3], Michael Steffen[3], Andreas Knobloch[3]

[1]GISPO
[2]GTK
[3]Beak Consultants GmbH

Funded by
the European Union

Disclaimer

The content of this report reflects only the author's view. The European Commission is not responsible for any use that may be made of the information it contains.

Funded by
the European Union

Document information

Grant Agreement / Proposal ID	101057357
Project Title	Exploration Information System
Project Acronym	EIS
Scientific Coordinator	Vesa Nykänen (vesa.nykanen@gtk.fi) – GTK
Project starting date (duration)	1 May 2022 (36 months)
Related Work Package	WP 3
Related Task(s)	Task 3.4
Lead Organisation	GISPO
Contributing Partner(s)	GTK, BEAK, UTU
Due Date	30.04.2024
Submission Date	02.05.2024
Dissemination level	PU

History

Date	Version	Submitted by	Reviewed by	Comments
01.02.2024	0.1	Andreas Knobloch		Template
30.04.2024	0.2	Niko Aarnio		Draft
02.05.2024	0.9	Andreas Knobloch		Revision
02.05.2024	1.0	Hafsa Munia		Submission

Table of contents

List of tables

List of figures

Abbreviations and Acronyms

Acronym	Description
WP	Work Package
GIS	Geographic Information System
EIS	Exploration Information System
GUI	Graphical User Interface

Summary

The Deliverable D3.6 provides a description of the contents and general structure of the EIS QGIS Plugin beta release and a short user manual.

Keywords

Software Design, Mineral Prospectivity Modelling, Mineral Predictive Mapping, QGIS, Artificial Intelligence, QGIS, Toolkit, Wizard, Python, Beta release

1. Executive summary

This document describes the first public beta release of the "EIS QGIS Plugin", which is a plugin for the QGIS software for conducting Mineral Prospectivity Mapping (MPM). The plugin relies on the previously released "EIS Toolkit" Python library and functionalities provided by QGIS.

The installation of the "EIS QGIS plugin" includes both the "EIS Toolkit" and the "EIS Wizard". The "EIS Wizard" works as a graphical user interface (GUI) for the "EIS Toolkit", providing a more user-friendly way of utilizing the functionalities offered by the library. The "EIS Wizard" becomes available in QGIS after the installation of the "EIS QGIS Plugin". The installation of the "EIS QGIS Plugin" also installs the "EIS Toolkit" as the library of tools, required to run the functionalities within the "EIS Wizard".

The technical documentation for "EIS QGIS Plugin" is included in this document.

The "EIS QGIS Plugin" is still actively developed, so minor issues might be found. The next release is planned for October 2024 – with the Deliverable D3.7 "EIS QGIS Plugin Final Version"

Any found issues and feedback are kindly requested to be reported as issues to the GitHub-repository of the plugin in https://github.com/GispoCoding/eis_qgis_plugin.

2. Introduction

2.1 General remarks

This document will provide background information about the beta release of "EIS QGIS Plugin". The functionality of the plugin relies mostly on the previously released "EIS Toolkit" which is a Python library and in addition it provides some tools for visualizing data. The beta release marks the start of beta testing for the plugin and another (stable/final) release is set for October 2024.

2.2 EIS WP3 and Task 3.4

The main objective of WP3 is the development of a GIS (Geographical Information System) based Exploration Information System (EIS) for predictive mapping of mineral resources. EIS does not have a strict definition but can be characterized as an environment for performing data analysis and modelling, for managing data and other information, and for representing results in various forms.

Task 3.4 consists of creation and implementation of the EIS QGIS Plugin, which provides the graphical user-interface (GUI) for utilizing the tools provided by the previously developed EIS Toolkit –Python library. In addition to providing a GUI for EIS Toolkit, the plugin guides the end user by providing a Wizard-like functionality for conducting the different steps that are part of mineral prospectivity mapping analysis.

3. User manual

3.1 Introduction

This chapter outlines how to use "EIS QGIS Plugin", what are the system requirements, and how it can be installed.

The "EIS QGIS Plugin" contains the "EIS Wizard" and the EIS Processing Algorithms as Provider through the Processing Toolbox in QGIS. The Processing Algorithms are all the processes provided by "EIS Toolkit" and can be run individually for any suitable data and the Wizard provides a guided workflow.

3.2 EIS QGIS Plugin System requirements

The minimum requirement for "EIS QGIS Plugin" is a QGIS 3.X installation. Any QGIS version that is 3.0 or newer is attempted to be supported, although this might be changed in the future if supporting the older versions cause unbearably many compatibility issues. The newest QGIS Long Term Release (QGIS 3.34.5 'Prizren' at the time of writing) is always the recommended one and the primary focus of support.

The primary website to download QGIS and to find more detailed help regarding QGIS installation is www.qgis.org.

To use "EIS QGIS Plugin", the user needs to install "EIS Toolkit" in a Python environment. The details of this are discussed more in the following chapters. In the future, compiled binaries might be released for "EIS Toolkit", which would allow users to use "EIS Toolkit" without separate installations and manual interaction with Python.

Technical system requirements for the "EIS QGIS Plugin" and the "EIS Toolkit" have not been determined, but a recent computer with a common operating system should work. GPU acceleration is not used yet so all the calculations and processing are done using CPU. There is no strict memory requirement but 16 GB or more is recommended for smooth operation.

3.3 EIS QGIS Plugin Installation guide

3.3.1 Installing the plugin

The most up-to-date installation guide is always on the front-page of "EIS QGIS Plugin" GitHub repository (https://github.com/GispoCoding/eis_qgis_plugin)

Installing QGIS plugins is a straightforward process in general. Only a folder that contains the plugin source files needs to be placed under a correct folder for QGIS to find them, although getting the plugin as a .zip file or installing directly from the official QGIS Python Plugins Repository are often the easiest methods. At the moment of beta release, "EIS QGIS Plugin" is not yet released in the official QGIS Python Plugins Repository.

To install "EIS QGIS Plugin", the user needs to get the .zip package that contains the source files. Currently, the only website that can be used for this is the GitHub repository of "EIS QGIS Plugin". There, under "Releases", the user can download the latest eis_qgis_plugin.zip.

When the user has successfully downloaded eis_qgis_plugin.zip, they should open QGIS. In QGIS, they open the Plugins menu and click the top entry that is "Manage and Install Plugins...". A new window opens, where the menu entry "Install from ZIP" should be selected. What remains is finding the downloaded .zip file and pressing

installing. If no errors occurred, “EIS” provider should be visible in the Processing Toolbox of QGIS and the “EIS Wizard” icon appear in the Plugins Toolbar.

3.3.2 Configuring EIS Toolkit

While most QGIS plugins work immediately after installation, the “EIS QGIS Plugin” requires an additional step: Installing and configuring the “EIS Toolkit”. For further information about the reasons for this, refer to the “Technical specifications” chapter. Furthermore, installation of “EIS Toolkit” itself is not described in this document, only the configuring after installation. Installation of “EIS Toolkit” is described in Deliverable D3.5 “EIS Toolkit Final Version”

When the user has successfully installed “EIS QGIS Plugin” and “EIS Toolkit”, they need to specify the Python environment where “EIS Toolkit” was installed. This is done in the “Settings” page of “EIS Wizard”, where the first section includes input fields and selections related to “EIS Toolkit”. While Docker setup is technically supported, it is not recommended at this stage; instead, the recommended approach is to use a Conda environment or Python venv. Regardless of the type of virtual environment, the user needs to specify the **root** folder of the virtual environment in the “Python venv directory” input field in the settings page for “EIS QGIS Plugin” to find “EIS Toolkit”. To check if “EIS Toolkit” is found, the user can press the “Verify” button.

3.4 EIS Wizard User Guide

The EIS Wizard User Guide is provided as **Appendix 1.**

4. Technical specifications

4.1 Description of development environment

To develop the "EIS QGIS Plugin", the minimum requirement was just a QGIS installation and any text editor.

QGIS comes with its own Python environment for Windows and uses the system Python in macOS and Linux operating systems. Depending on QGIS version, a set of Python packages is installed to the environment, most notably PyQGIS – the Python QGIS API. Most work in QGIS plugin development revolves around using classes and functions of PyQGIS.

To enhance the development process, some additional tools were used. For quick deployment of code changes, QGIS Plugin tools (https://github.com/GispoCoding/qgis_plugin_tools) package was used in combination with the Plugin Reloader QGIS plugin. For code linting and formatting, Ruff (https://github.com/astral-sh/ruff) was used. As a good development practice, the deploying and formatting were performed in a virtual environment.

For developers using Windows, additional actions are often needed for the IDE (such as Visual Studio Code) to be aware of PyQGIS. The following site guides any fresh QGIS plugin developers through this process: https://www.gispo.fi/en/blog/getting-started-with-qgis-plugin-development-in-2022/.

4.2 Description of technical architecture

4.2.1 EIS QGIS Plugin file structure

The following list attempts to give a glimpse to the file structure of the "EIS QGIS Plugin". Most important modules and folders for this purpose are listed, but some are omitted to keep the list size manageable. Folders are marked with **bold** font, other entries represent Python modules or other files:

- Main file (*plugin.py*)
- **EIS Processing**
    - **Algorithms**
- **Exploratory analysis**
- **Filtering**
- **Prediction**
- **Raster processing**
- **Transformations**
- **Utilities**
- **Evaluation**
- **Vector processing**
- EIS Processing Algorithm
- EIS Provider
- EIS Toolkit invoker
- **EIS Wizard**
- **Mineral system proxies**

- Proxy processing
- Proxy view
- **Modeling**
- **Fuzzy modeling**
- **Machine learning**
- Model manager
- Model utils
- **Plots**
- **Utils**
- Model feedback
- Settings manager
- Wizard main
- Wizard proxies
- Wizard plot
- Wizard modeling
- Wizard history
- Wizard settings
- Wizard about
- Settings
- Utils
- **Resources**
- Proxies JSON
- **UI files**
- **Icons**
- **Libraries**
- **Seaborn**

4.2.2 EIS Wizard architecture

In technical terms, the “EIS Wizard” is a QWidget that is either inside a QDialog or a QgsDockWidget. One main script instructs the high-level creation of “EIS Wizard” when the user presses the EIS Wizard icon in QGIS.

At the moment of beta release of the “EIS QGIS Plugin”, the “EIS Wizard” contains 6 tab pages:

- **Mineral System Proxies**
- **EDA (Exploratory Data Analysis)**
- **Modeling**
- **History**
- **Settings**
- **About.**

What page is visible is controlled by clicking a linked menu item on the left side of the interface. When user clicks one of these menu items, the currently shown page of the QStackedWidget on the right side is changed to match the selected menu item. What is perhaps the most important to understand about the high-level implementation of “EIS Wizard” is that all the pages exist simultaneously when “EIS Wizard” is open. In other words, the pages do not get destroyed when switching to another page and the user can except to find the page in the same state when returning to it (before closing EIS Wizard). This is how almost everything in “EIS Wizard” works, so also tabs have their own interfaces that don't get destroyed by switching tabs.

In general, each unique view in “EIS Wizard” is defined in a corresponding .ui file created with Qt Desginer software. In Python scripts, the .ui file is loaded and sometimes widgets are added to the view (such as the custom ModelDataTable widget). In addition, connections between interface components are created and functionalities defined.

Since PyQGIS is a very object-oriented library, it makes sense to follow the paradigm when implementing QGIS plugins. Consequently, EIS QGIS Plugin consists largely of different classes that interact with each other.

4.2.2.1 Mineral System Proxies

Mineral system proxies dialog has 2 types of views: Mineral system proxies list and Proxy processing. The former changes based on selected mineral system and study scale, and there are multiple versions of the latter, some of which include multiple pages. The available proxies, data related to them and the linked proxy procesing page are defined in a JSON file (*proxies.json)*. This means that a knowledgable user can make modifications to the JSON file should they want to make changes (although the "custom" mineral system can suit better for most cases where more freedom is wanted). The proxy processing pages are custom interfaces that are linked to the following processing algorithms: Distance computation, Distance to anomaly, IDW interpolation and Kriging interpolation.

4.2.2.2 EDA (Exploratory Data Analysis)

Currently, EDA page includes only plots. The included plots are barplot, boxplot, ECDF, histogram, KDE, lineplot, pairplot, parallel coordinates and scatterplot. All the plots are defined as their own classes that inherit from a base class called EISPlot. All but one of the plot classes consist almost fully of reading the input data and parameters and delivering them to Seaborn plotting library that produces the plot. The plot with more customized code and that directly uses matplotlib is the parallel coordinates plot, which was not available in Seaborn at the time of beta release.

As most everything else in “EIS Wizard” relies on EIS Processing algorithms and “EIS Toolkit”, it should be explained why the plot tools are (re)defined in “EIS Wizard”. The answer is quite simple: to make the plots more dynamic and allow easy plotting without having to save every plot to disk. While it is not yet implemented, linking data points in plots and the corresponding points/areas in a map representation is also possible only if the plots are produced within the plugin.

4.2.2.3 Modeling

The modeling page contents depend on the selected model. For machine learning models, the contents are largely the same: only the model-specific parameters vary by model. *Data preparation*, *Training*, *Testing* and *Application* tabs are shown for these models. Each of the tabs are defined in their own Python scripts and UI files, and each tab (for a selected model) is linked to a matching EIS Processing Algorithm.

When a machine learning model is trained, information about it is saved to the QGIS project file by a class called ModelManager. Essentially, there is a minimal model database in the project file that makes it possible to test, apply and inspect previously trained models in separate sessions (so after QGIS is closed). At the time of beta release, the model database is still being developed and does not include all information that could be useful to save.

Fuzzy overlay has its own unique tabs: Fuzzy memberships and Fuzzy overlay. Fuzzy memberships handles processing the membership datasets, and the computations are currently performed using the various "Fuzzify raster" QGIS native processing algorithms. This might be changed in the future.

Many of the modeling tabs have a table input widget that is a customized widget. The widget is called ModelDataTable and defined in *model_data_table.py*.

4.2.2.4 History

The history page is very minimal currently. It allows the user to view and delete trained models. It was essential to include in its minimal form in the beta release so that users have a way to delete the saved models.

4.2.2.5 Settings

The settings page collects all things related to settings as the name indicates. The page is a collection of QGroupBoxes and nested widgets. Most important setting is specifying “EIS Toolkit” installation – without it, any computational tools won't run. All settings are saved to QGIS project through QgsSettings class, and all saving and retrieving of settings values is handled by EISSettingsManager in a centralized way. All the settings are just key-value pairs.

4.2.2.6 About

The about page includes only text, images and links that were seen important to include in the plugin.

4.2.3 EIS Processing algorithms architecture

EIS Processing algorithms is a collection of QGIS processing algorithms. Almost every tool from “EIS Toolkit” is mapped one-to-one with EIS Processing algorithms. This means essentially that “EIS Toolkit” is available in QGIS in a toolkit form, not only as a set of computations behind “EIS Wizard”.

All processing algorithms are grouped under one processing provider called "EIS". This provider appears in the Processing Toolbox of QGIS when “EIS QGIS Plugin” is installed. Under the provider, each tool is categorized under a group that matches a module in “EIS Toolkit”.

Each processing algorithm is defined in its own Python file, but common functions and properties are abstracted to a base class called EISProcessingAlgorithm. Since the processing algorithms do not carry out the computations themselves and the base class handles the common parts, it is straightforward and easy to add new EIS Processing algorithms should they be added to EIS Toolkit. The best way to go about this is to take some existing processing algorithm file as the basis and change the name, parameter types and descriptions accordingly.

4.2.4 EIS QGIS Plugin – EIS Toolkit integration

The “EIS QGIS Plugin” relies on “EIS Toolkit” for the execution of its computational tools. In some sense, “EIS Wizard” is the "frontend" and “EIS Toolkit” the "backend", as software components are sometimes categorized.

In the beginning of the project, it was thought that “EIS Toolkit” could be installed in the Python environment of QGIS. This would be the ideal option, as “EIS QGIS Plugin” could then easily find and use “EIS Toolkit”, and data objects could be shared between PyQGIS functions that run in the plugin and the computational tools of EIS Toolkit. However, it soon became evident that the Python environment of QGIS is inflexible and cannot run the heavy packages of EIS Toolkit with complex dependencies, such as Tensorflow. This is the reason “EIS Toolkit” needs to be installed in a separate - ideally otherwise empty - Python environment, such as Conda environment or venv, and that “EIS QGIS Plugin” would need to request computations from the isolated backend runner.

The distilled technical description for “EIS QGIS Plugin” – “EIS Toolkit” communication is the following: EIS Processing Algorithms of “EIS QGIS Plugin” assemble a detailed command for the command-line interface (CLI) of “EIS Toolkit” and start a separate Python process using the “subprocess” module of Python. To get into more detail about the communication, the command-line interface of “EIS Toolkit” and the Toolkit invoker of “EIS QGIS Plugin” should be described.

4.2.4.1 EIS CLI

The “EIS Toolkit” CLI has **one matching function per tool**, excluding a few exceptions where one-to-many or many-to-one mapping was chosen due to some technicalities. Each CLI function handles necessary file opening, running the linked tool and either writing output file or sending result information back via the stdout-stream.

All the functions are collected in one module called *cli.py*. The EIS CLI can be accessed from the command-line by running the module (**python -m eis_toolkit**) or with a script shortcut (**eis**) followed by a function to run and its parameters. An example command to run resampling: **eis resample-raster-cli --input-raster path/to/my/raster.tif --output-raster my_output_raster.tif --resolution 50 --resampling-method bilinear**

The CLI was created for two main reasons: 1) the need to run additional functions before and/or after the tool itself - such as reading and writing raster files - when accessing the tools from external programs 2) to define a clear entry point for external programs. An additional benefit of creating a CLI is that it functions as an alternative way to use “EIS Toolkit”; users get the opportunity to run “EIS Toolkit” tools from their terminal program.

4.2.4.2 Toolkit invoker

Toolkit invoker is not any entity exposed to end users but is a central agent in implementation of “EIS QGIS Plugin” – “EIS Toolkit” communication. In a nutshell, it is the responsibility of Toolkit invoker to manage all things related to “EIS QGIS Plugin” – “EIS Toolkit” communication **on the plugin side**. This includes tasks such as

- Verifying the correctness of “EIS Toolkit” setup
- Translating parameter values of EIS processing algorithms to command-line arguments
- Assembling the full path to the Python executable that should be used to run “EIS Toolkit”
- Opening the Python subprocess
- Reading from the I/O connection
- Communicating with the user in QGIS using the feedback messages

EIS CLI and Toolkit invoker act as a pair that facilitates the communication between the two programs. Since there is nothing "QGIS specific" about the EIS CLI, it is possible to utilize “EIS Toolkit” in other software in a very similar way as in QGIS, if tailored scripts that do the job of Toolkit invoker are created for these other software.

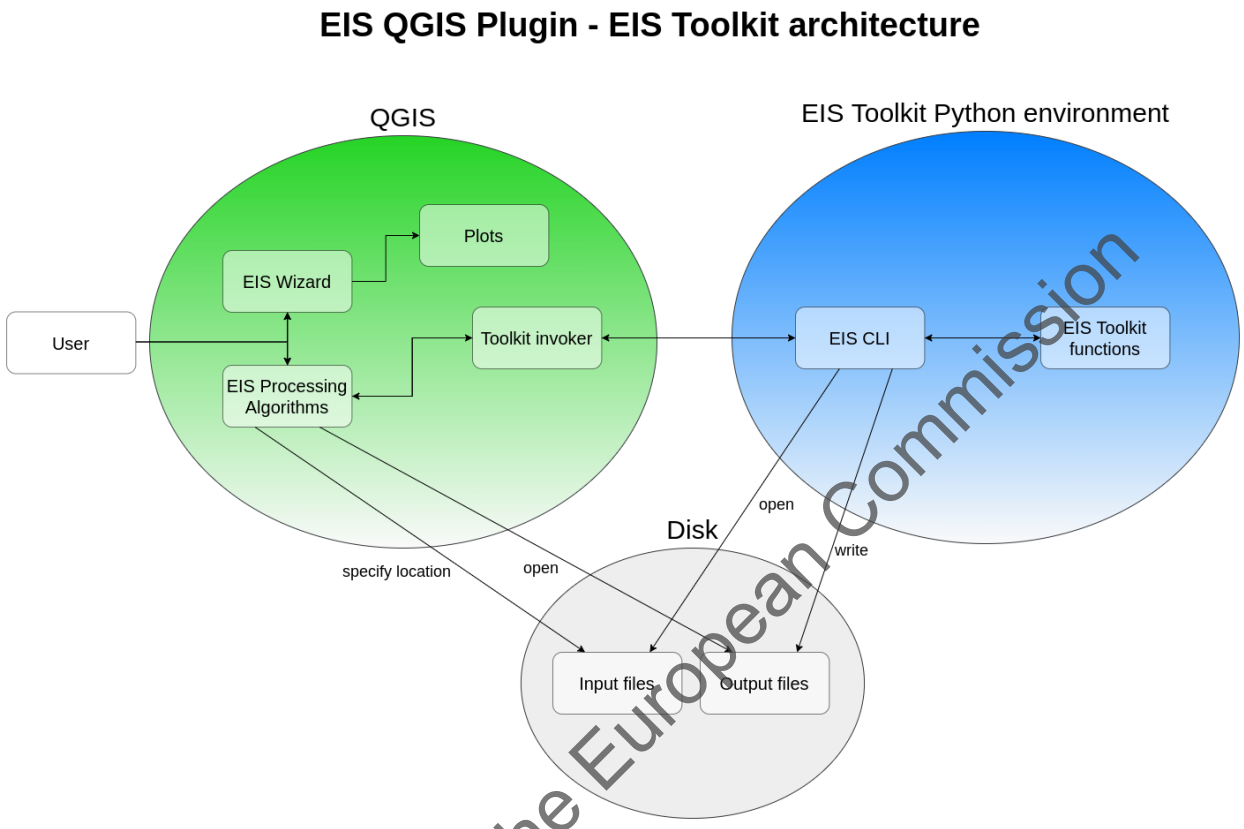


Figure 1: Schematic overview of EIS QGIS Plugin – EIS Toolkit architecture and interaction

5. Initial testing

5.1 Listing of initial testers

The initial testing is conducted by a small group of voluntary testers within the project who have not been actively involved in the development of “EIS QGIS Plugin”.

The first EIS Internal User Training Workshop is planned for 28 and 29 May 2024 to take place in Helsinki, in person.

During the 2 days, participants from EIS WP4 will be introduced to the “EIS Wizard” and its functionalities with a tutorial run-through on the first day and then first practical testing of the “EIS Wizard” with the actual data from the EIS test sites.

Currently, the following 4 training instructors for the EIS Wizard are assigned:

Table 1: List of EIS Wizard Training Instructors

	First name	Last name	Organisation
1	Niko	Aarnio	Gispo
2	Johanna	Pesonen	Geological Survey of Finland (GTK)
3	Bijal	Chudasama	Geological Survey of Finland (GTK)
4	Ina	Storch	Beak Consultants GmbH

For the initial testing, the following 18 trainees / testers are planned to take place in the EIS Internal User Training Workshop in May 2024:

Table 2: List of Initial EIS Wizard Trainees / Testers

	First name	Last name	Organisation
1	Gonzalo	Ares	Instituto de Geociencias (CSIC-UCM)
2	Iana	Bacquet	TALGA
3	Nolukholo Sinovuyo	Busakwe	University of Free State (UFS)
4	John	Carranza	University of Free State (UFS)
5	Percy	Clark	GoldenPET
6	Anne	Dabosville	BRGM
7	Guillaume	Bertrand	BRGM
8	Alex	Vella	BRGM
9	Helena	Hulkki	Geological Survey of Finland (GTK)
10	Hafsa	Munia	Geological Survey of Finland (GTK)
11	Tero	Niiranen	Geological Survey of Finland (GTK)
12	Vesa	Nykänen	Geological Survey of Finland (GTK)
13	Sari	Romppanen	Geological Survey of Finland (GTK)
14	Martiya	Sadeghi	Geological survey of Sweden (SGU)
15	Krisztian	Szentpeteri	Geological Survey of Finland (GTK)
16	Andreas	Knobloch	Beak Consultants GmbH
17	Javad	Sheikh	University of Turku / Department of Computing
18	Jiří	Zachariáš	Faculty of Science, Charles University

After this initial training, further testing will be continued independently by the listed internal users from EIS.

Furthermore, an EIS External User Training Workshops is planned for either PDAC 2025 or EGU 2025 conference.

5.2 Guidelines for testers to report bugs and problems

Development of “EIS QGIS Plugin” is done in GitHub. All testers with a GitHub account should use the “EIS QGIS Plugin” repository to report bugs and to send modification suggestions. The most practical way to do this is to use *GitHub issues*, which are widely used in programming projects for reporting bugs and development related tasks.

The “EIS QGIS Plugin” GitHub page has basic instructions for installing and using the plugin, and the initial testers should refer to guidance found in there. In addition, a more detailed “Getting started with testing” document is sent to all initial testers.

Since the development of “EIS QGIS Plugin” continues throughout the testing phase, new additions and bugfixes occur frequently. The testers are advised to visit the GitHub page often to get the most updated version of “EIS QGIS Plugin” and find the changelogs.

6. Features to be implemented in next 6 months until final release

The planned features to be implemented are divided here into three categories: general features, EIS Wizard features and EIS Processing algorithm features. For each of these categories, the items are listed, briefly described and labelled as either "In consideration" or "Decided" to reflect the status of the planned feature. The "Decided" label indicates that this feature is truly intended to implement and will be if no critical obstacles occur. The "In consideration" indicates that while the feature is seen desirable, the details of it are still to be decided or it might not be ready before the final (the 1.0) release of "EIS QGIS Plugin".

6.1 General features

Feature	Description	Status
Enable canceling execution of EIS Toolkit tools	Processing algorithms or computations in EIS Wizard cannot be cancelled currently, except by closing QGIS. This feature would bring a lot of convenience, since sometimes long executions need to be cancelled.	Decided
Release EIS QGIS Plugin in the official QGIS Python Plugins Repository	EIS QGIS Plugin should be released in the official QGIS plugin repository to make installation easier and so that QGIS users can find the plugin.	Decided
Add EIS Toolkit installer	If EIS Toolkit installer was implemented in EIS Wizard, it could simplify the installation process greatly. However, the feasibility of this is still to be assessed. Furthermore, the need of this feature depends also on whether an EIS Toolkit binary (compiled, executable file) can be produced.	In consideration
Improve Docker support	Docker containers with EIS Toolkit installation can be configured and used in EIS Wizard already, but it is not thoroughly investigated how well it performs in various platforms. In addition, no EIS Toolkit Docker images meant for end users are currently available, only an image for development purposes.	In consideration
Add automatic visualization settings for outputs	Some outputs from EIS QGIS Plugin computations could be stylized automatically or based on settings made by the user. For example, discretized data could be visualized automatically using a certain palette. However, this is not a critical feature and thus might not be implemented in the near future.	In consideration

Name output layers more descriptively	Currently, most of the tools simply name the output layer either “Output raster” or “Output vector”. This is not descriptive, and if the user does not manually rename an output immediately, they might end up forgetting and mixing some outputs.	In consideration
Harmonize naming conventions within the plugin	Some concepts are referred to with multiple names, at least QGIS vector layer attributes (attribute/field/column). To avoid confusion, names could be harmonized as much as possible.	In consideration

6.2 EIS Wizard features

Feature	Description	Status
Implement Co-VMS mineral system for proxy processing	The Co-VMS system is one of the three main mineral systems targeted in the EIS project and therefore a necessary inclusion in EIS Wizard.	Decided
Implement Li-Pegmatites mineral system for proxy processing	The Li-Pegmatites system is one of the three main mineral systems targeted in the EIS Project and therefore a necessary inclusion in EIS Wizard.	Decided
Redesign custom mineral system in mineral system proxies	Currently, when the user selects custom mineral system, all mineral system proxies available in the database (proxies.json) are listed without labeling them with importances. It has been suggested that the custom mineral system would instead allow the user to manually add their own mineral system proxies and assign proxy processing workflows to them.	In consideration
Add Weights of Evidence in EIS Wizard modeling	Weights of Evidence model is already implemented in EIS Toolkit and should be added to EIS Wizard to extend the set data-driven models.	Decided
Add MLP (Multilayer perceptron) in EIS Wizard modeling	MLP model is already implemented in EIS Toolkit and should be added to EIS Wizard to extend the set of deep learning models. Before this is implemented, MLP processing algorithm needs to be implemented.	Decided
Add CNN (Convolutional neural network) in EIS Wizard modeling	CNN model development in EIS Toolkit is in progress and will be finished soon. After that, CNN should be added to EIS Wizard to extend the set of deep learning models. Before this is implemented, CNN processing algorithm needs to be implemented.	Decided

Add BNN (Bayesian neural network) in EIS Wizard modeling	BNN model development in EIS Toolkit is in progress and will be finished soon. After that, BNN should be added to EIS Wizard to extend the set of deep learning models. Before this is implemented, BNN processing algorithm needs to be implemented.	Decided
Add selection of training area in EIS Wizard modeling	Allowing users to select a training area as a parameter in machine learning model training has been suggested. This would bring convenience to the training process. However, clipping datasets with the available tools is already possible before training the model and therefore this is not a critical addition.	In consideration
Extend modeling history	Modeling history in EIS Wizard saves currently very minimal amount of information about the models and should be extended to allow inspecting training and evaluation metrics of trained models.	Decided
Improve model data preparation tab	Model data preparation tab exists and has functional buttons to open various data preparation tools, but refining the window will be done. At least the set of offered tools will be assessed for each model individually, and additionally the user interface might be redesigned.	Decided
Improve model result evaluation capabilities	Currently, basic model evaluation using metric scores, such as accuracy and squared errors, is possible, but expanding the evaluation to include all offered tools of EIS Toolkit should be done. Additionally, comparison between models in single plots or summary tables is seen highly desirable.	Decided
Change how plotting in EDA works	Plotting in the EDA window works currently, but it has been suggested that all the different plot interfaces should share selected datasets. This change could make plotting more intuitive for some users and is considered to be implemented.	In consideration
Add exploratory methods in EIS Wizard, including at least K-means, DBSCAN, PCA and Local Moran's I	The EDA window should be extended with various exploratory methods, such as K-means clustering, DBSCAN and Local Moran's I.	Decided
Add density scatterplot linked to map layer	A density scatterplot that would be linked to the map layer of its data could provide a powerful way to inspect multiband raster data or vector data with multiple features. 3 features/bands at a time could be visualized as a 3D scatterplot, where the user could select cluster of points and the selection would be also	In consideration

	displayed on the map layer. Implementation details of this feature are still to be investigated.	

6.3 EIS Processing algorithm features

MLP processing algorithm	MLP processing algorithm will be added to extend the set of offered processing algorithms.	Decided
CNN processing algorithm	CNN processing algorithm will be added to extend the set of offered processing algorithms.	Decided
BNN processing algorithm	BNN processing algorithm will be added to extend the set of offered processing algorithms.	Decided
Autoencoder for image segmentation processing algorithm	Autoencoder for image segmentation processing algorithm will be added to extend the set of offered processing algorithms.	Decided
Feature importance processing algorithm	Feature importance processing algorithm will be added to extend the set of offered processing algorithms.	Decided
Summarize label metrics binary processing algorithm	Summarize labels metrics binary processing algorithm will be added to extend the set of offered processing algorithms.	Decided
Summarize probability metrics processing algorithm	Summarize probability metrics processing algorithm will be added to extend the set of offered processing algorithms.	Decided
Cell-based association processing algorithm	Cell-based association processing algorithm will be added to extend the set of offered processing algorithms.	Decided
Mini-Unet processing algorithm	Mini-Unet processing algorithm might be added to extend the set of offered processing algorithms. The addition depends on whether Mini-Unet will be successfully implemented in EIS Toolkit.	In consideration
SOM processing algorithm	SOM processing algorithm might be added to extend the set of offered processing algorithms. The addition depends on whether SOM will be successfully implemented in EIS Toolkit.	In consideration

Mahalanobis similarity processing algorithm	Mahalanobis similarity processing algorithm might be added to extend the set of offered processing algorithms. The addition depends on whether Mahalanobis similarity is decided as a useful addition regardless of its limitations.	In consideration

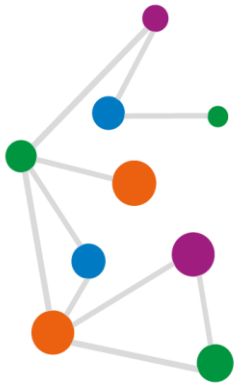
7. Conclusion

The first beta release of the "EIS QGIS Plugin" on GitHub has been done and the work continues with finalising the functionality and fixing issues potentially found during the testing phase that starts with this release. It is likely that minor changes will be needed in "EIS Toolkit" as well to address already surfaced performance issues.

After and during the testing phase, reported issues will be addressed before the final/stable release that is set for October 2024.

8. References

Nykänen, V., Lahti, I., Niiranen, T., & Korhonen, K. (2015). Receiver operating characteristics (ROC) as validation tool for prospectivity models — A magmatic Ni–Cu case study from the Central Lapland Greenstone Belt, Northern Finland. Ore Geology Reviews, 71, 853-860. doi:10.1016/j.oregeorev.2014.09.007



EIS
Exploration
Information
System

D 3.6 EIS QGIS Plugin

Appendix 1: EIS Wizard User Manual

Version 1.0

Niko Aarnio [1], Timo Aarnio[1], Miikka Kallio [1], Mika Sorvoja[1], Johanna Torppa[2], Bijal Chudasama[2], Ina Storch[3], Peggy Hielscher[3], Michael Steffen[3], Andreas Knobloch[3]

[1]GISPO
[2]GTK
[3]Beak Consultants GmbH

Funded by
the European Union

Table of contents

Abbreviations and Acronyms

Acronym	Description
GIS	Geographic Information System
EIS	Exploration Information System
GUI	Graphical User Interface
EDA	Exploratory Data Analysis
MPM	Mineral Prospectivity Mapping
KDE	Kernel Density Estimate
ECDF	Estimator of the Cumulative Distribution Function

1. Introduction

EIS Wizard is developed to guide you through the entire workflow of mineral prospectivity analysis. It is part of EIS QGIS Plugin, that consists of EIS Wizard and EIS Processing Algorithms. EIS Processing Algorithms comprise of a set of individual functions for processing and exploring data, and for predictive modeling and evaluation of results. EIS Wizard uses these tools as part of guided workflows that are common in predictive modeling of mineral deposits.

2. Using EIS Wizard

The EIS Wizard icon appears in the QGIS toolbar once EIS Plugin is installed (**Figure 1**). Open EIS Wizard by clicking the icon.

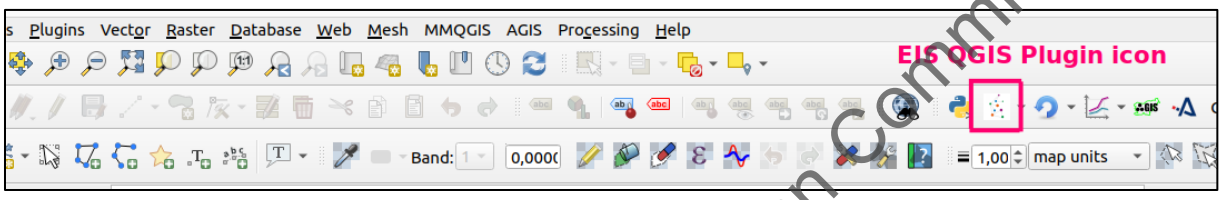


Figure 1: Icon for "EIS QGIS Plugin" in QGIS Toolbar.

EIS Wizard consists of separate views (**Figure 2**) for each step of prospectivity analysis (*Mineral system proxies, EDA, Modeling*), a page for settings (*Settings*) and an information page about the wizard (*About*). In the following sections we describe the contents and functionalities in each of these pages (except for the *About* page, which is self-explanatory), and tell how you might navigate through an MPM workflow using EIS Wizard.

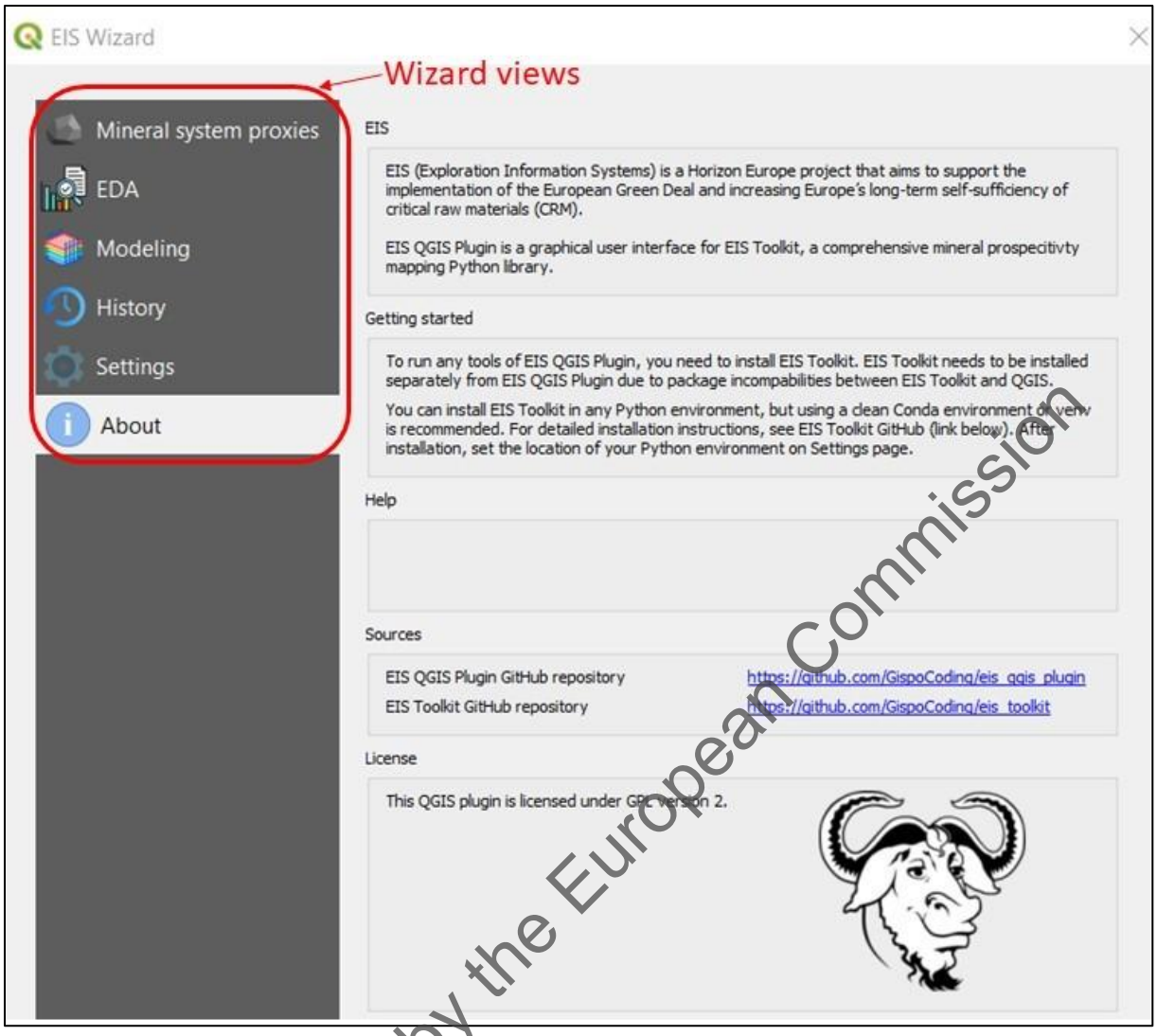


Figure 2: EIS Wizard user interface and list of views within the tab page "About".

2.1 Mineral System Proxies

The Mineral system proxies view is the first page that you should open once you start an MPM project. The view helps you to select and generate data for modeling the occurrence of a certain type of mineral system. The view presents a list of proxies, divided into tabs for each mineral system critical component (3) which are i) *Source*, ii) *Active/structural pathways*, iii) *Depositional processes*, iv) *Mineralisation, remobilization, modification*.

2.1.1 Proxies view

The mineral system is selected at the top of the view (1). The options are i) *IOCG,* ii) *Li-Pegmatites* and iii) *Co-VMS*. It is also possible not to use any predefined mineral system by selecting the option *Custom*. Scale of the study area is also needed (2), as different proxies are important at different scales. Options are i) *Regional*, ii) *Camp* and iii) *Deposit* scale. Importance of the proxies at different scales are represented by the color of the "*" characters before proxy name. Red marks high importance, orange moderate importance and green low importance.

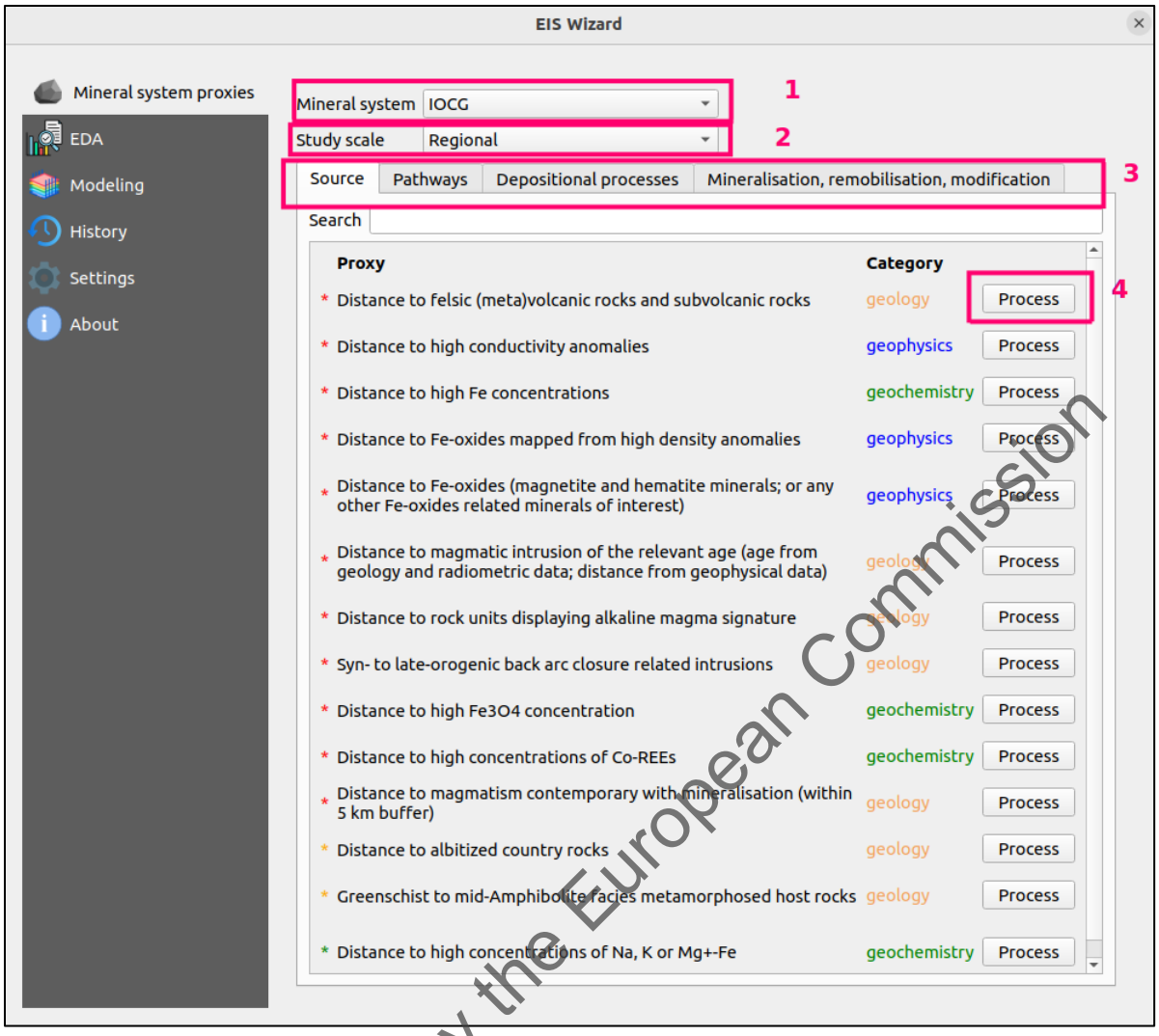


Figure 3: **EIS wizard with tab page for selection of "Mineral systems proxies".**

2.1.2 Proxy processing view

By pressing the *Process* button on the right side of the Proxies view page (**Figure 4**), a new page opens for processing your input data to represent the corresponding proxy. There are currently 2 different workflows that are mapped to the proxies: Compute distance to features, and Interpolate + Compute distance to anomaly. Each of these workflows has its own page.

Distance to features tool (Figure 4) is used to compute distance to points, lines or polygons. By selecting only part of the features in the data set using an expression in the *Selection* parameter distance to, e.g., anomalous points can be computed.

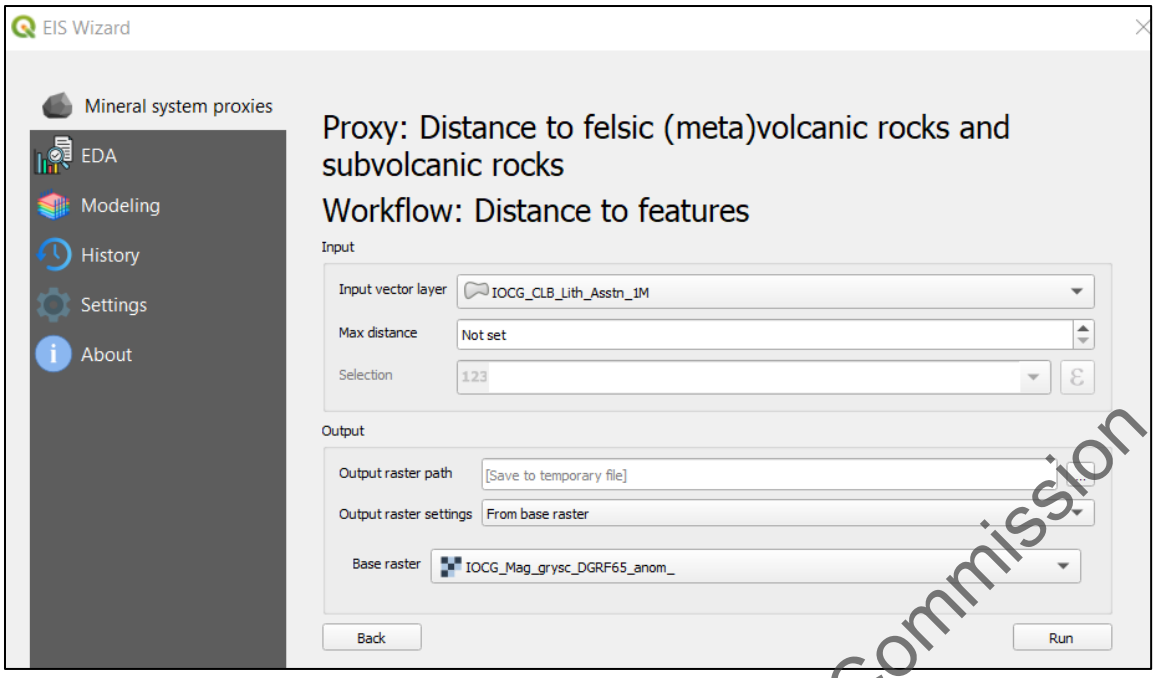


Figure 4: **Distance to features tool computes distances to lines, points or polygons.**

In case of computing distance to anomalous raster values, the raster is optionally generated first using the *Interpolation* view (**Figure 5**). Methods for interpolation to choose from are inverse distance weighting (IDW) and kriging.

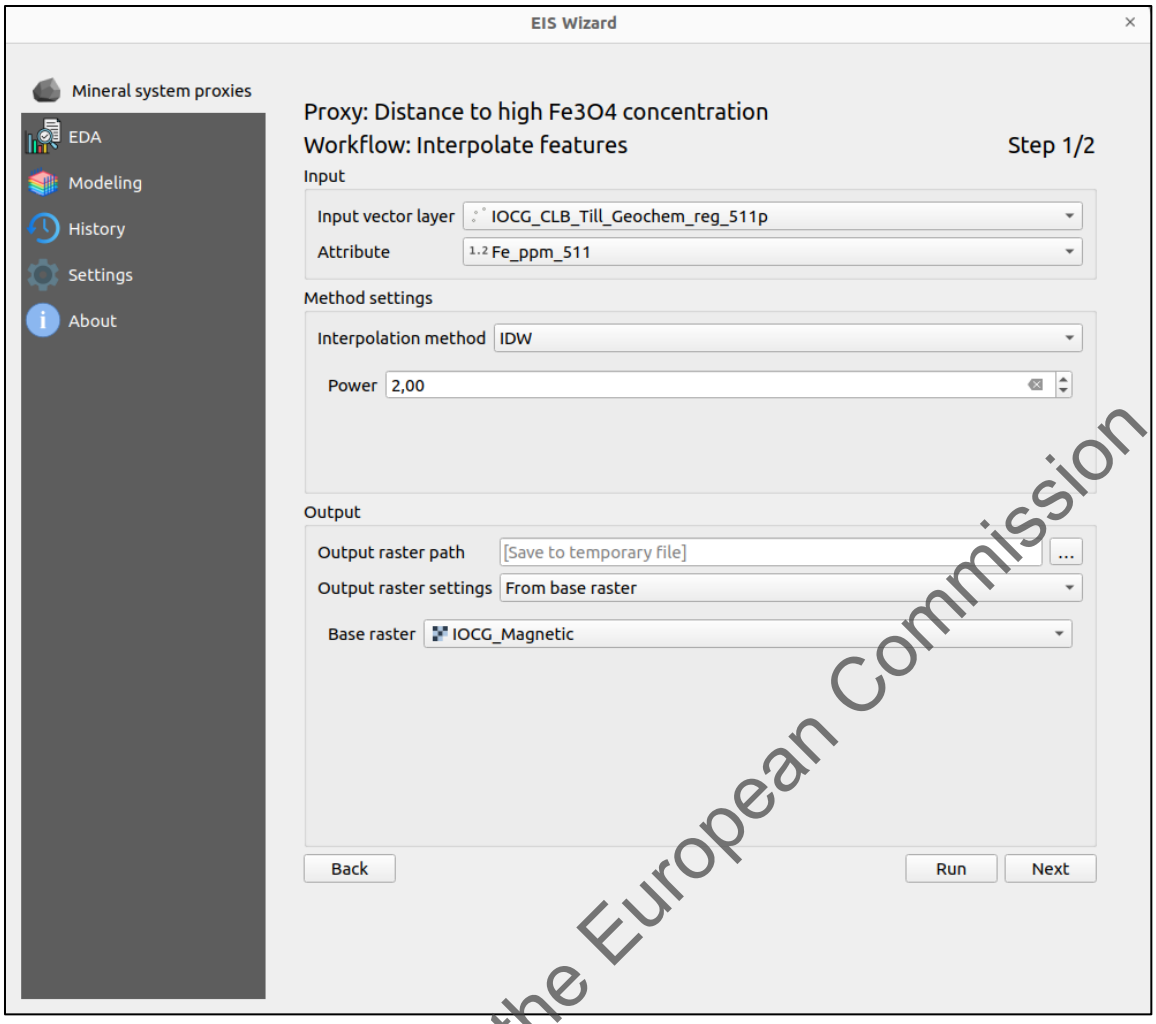


Figure 5: Interpolation view offers tools to interpolate point data.

Distance to the anomalies in the raster is then computed on the second view of the workflow called *Define anomaly* (**Figure 6**) accessed by pressing the *Next* button on the *Interpolation* view. Anomaly can be defined as values higher or lower than a threshold, when one threshold value is given. It is also possible to provide two threshold values, when anomaly is defined either as in-between or outside these values.

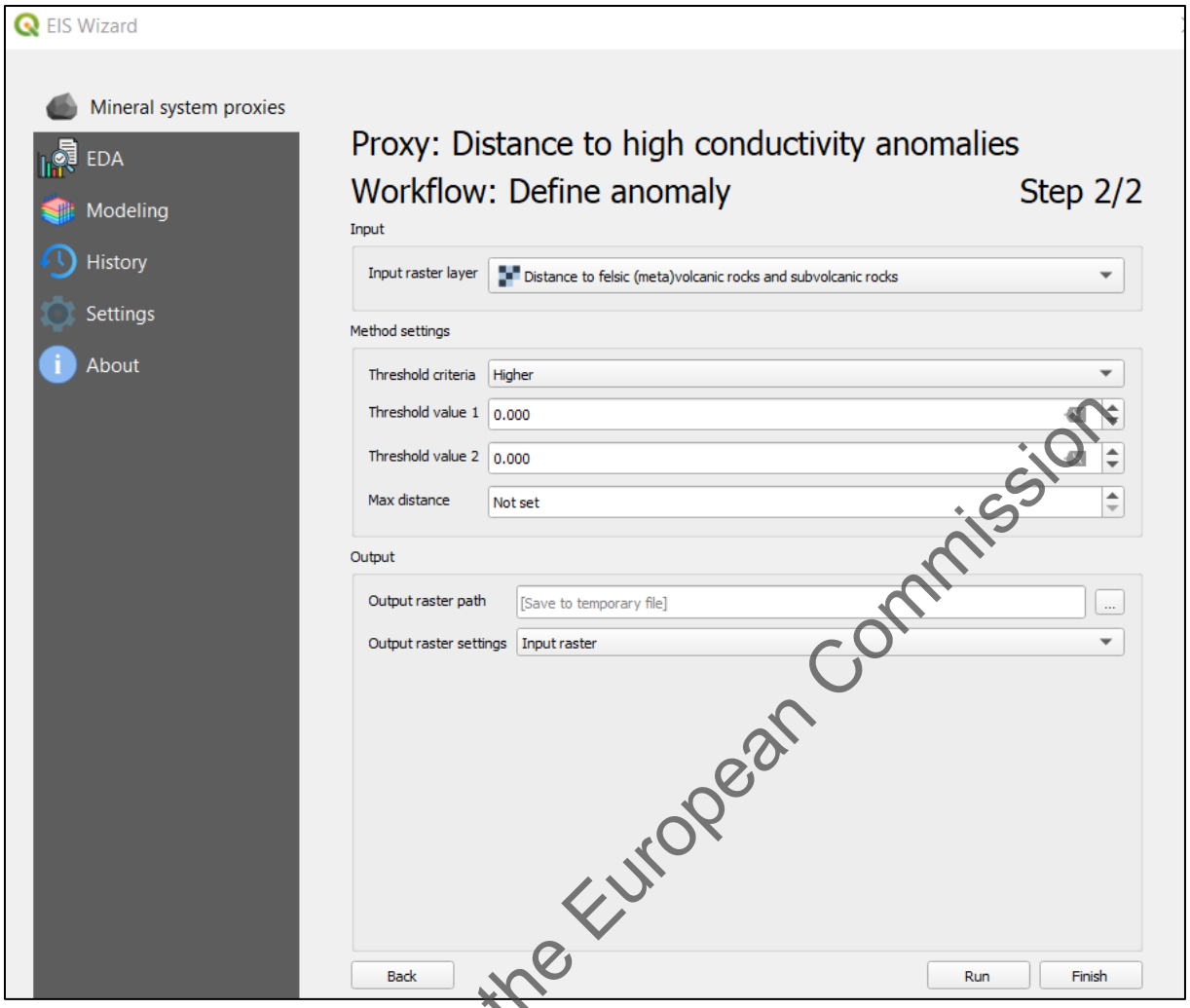


Figure 6: **The "Define anomaly" tool computes distance to anomalous areas in rasters in the "Mineral system proxies" tab page of EIS Wizard**

2.1.3 Using Mineral system proxies

The Mineral system proxies page was included in EIS Wizard to implement a Mineral systems approach for mineral exploration in EIS QGIS Plugin. You can utilize the collected information about mineral system proxies to prepare data for modelling in a knowledge-driven fashion. However, in case you intend to use machine learning models, you can also try other methods for data selection and preparation, such as the feature importance tool.

How you decide which proxies you prepare will likely be a combination of different factors: The importance of the proxy, what data you have available and which proxies you assume will be the most important for your specific study. Those that decide to use the *Custom* mineral system will rely on their own knowledge of the relevant proxies but can still take advantage of the proxy list representation we have created.

Often an MPM workflow is iterative, and when already modeling, you might decide you have too few proxy datasets prepared and come back to prepare more proxies. Because of this, EIS Wizard was designed to make it easy to jump back and forth between the steps by not restricting the user to linear, tunnel-like workflow.

2.2 Exploratory Data Analysis (EDA)

The EDA view facilitates data exploration with only plots at the time of EIS QGIS Plugin beta release. With the plots, you can visualize distributions and relationships between variables to identify (anomaly) thresholds, correlation and other patterns of interest.

2.2.1 Plots

The most common plotting methods are included in EIS Wizard to facilitate basic data exploration. Plots specific to modeling result evaluation are not included here but in the Modeling page. You can visualize datasets with the following plot types in EIS Wizard:

- Histogram
- KDE
- ECDF
- Scatterplot
- Lineplot
- Barplot
- Boxplot
- Pairplot
- Parallel coordinates plot

The EDA plot window layout is assembled so that you should traverse the content vertically from top to bottom. As the first thing at the top of the window, you can select the desired plot type from the available plots listed above. When the plot type is changed, the data selection and parameter widgets are automatically updated to match the selected plot type.

After selecting your plot type, you should select your data. Depending on the plot, raster and vector layers can be used as input data. The supported data types are automatically filtered, so the dropdown menu will only show layers that can be used for the plot type. For some plots, you need define only one feature or band to plot, for other multiple are required.

The parameter section consists of some generic parameters, such as color and opacity, and of several plot-specific parameters. For simple cases, like plotting a histogram to check the distribution of your data, these parameters might not need to be touched, but for stylized and/or scaled plots they are made available.

After all inputs and parameters are defined, the plot can be generated by pressing the *Create plot* button. The plot appears below in a space reserved for it. Should you want to inspect the plot in a larger window, you can press the *Open in new window* button which creates a popup window with the plot. Note that this will **transfer** the plot to the new window, so it will disappear (or become buggy) in the embedded view. For further adjustments on the plot and saving it to a file, you can use the navbar above the plot.

The *Reset* button resets only the plot parameters to their defaults and does not affect the produced plot, plot type or input data. The *Clear plot* button will remove the produced plot from the embedded plot view. Note that clicking *Clear plot* will also close the popup window with your plot if you have transferred it to a new window.

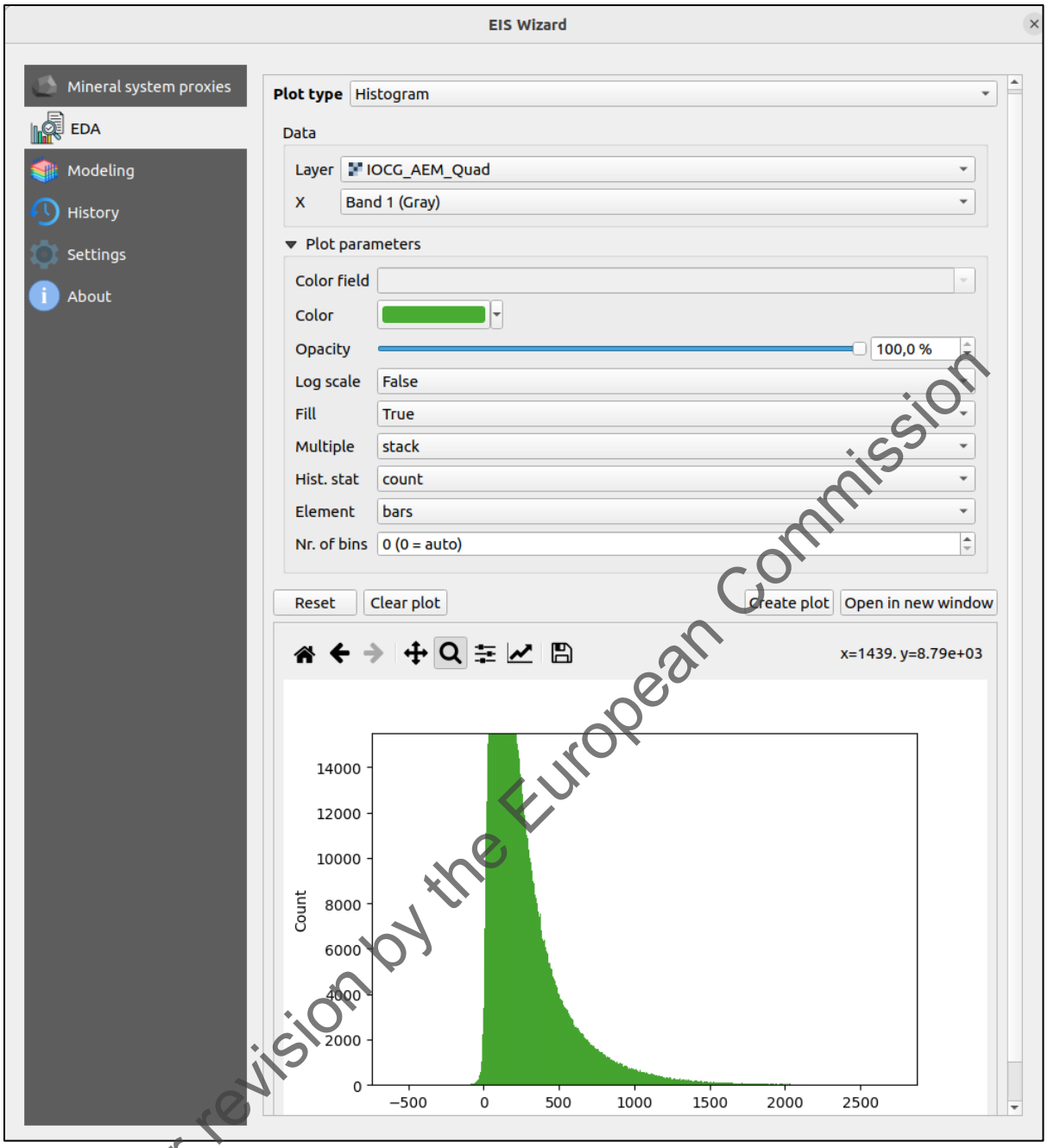


Figure 7: Plot histogram tool under "Exploratory Data Analysis" tab page in EIS Wizard.

2.3 Modeling

The *Modeling* view has tools for generating predictive models using multiple explanatory data sets and optionally observations, depending on the modeling method. Currently, four methods are implemented for classification and regression: Logistic regression, Random forests, Gradient boosting and Fuzzy logic. The method can be chosen in the top of the window.

Modeling view has four tabs: Data preparation, Training, Testing and Application

The *Data preparation* tab (**Figure 8**) offers tools for data processing methods that are commonly needed for transforming the proxy data to a form suitable for modeling. *Raster preparation* tools are used to unify raster gridding and coordinate system, *Data transformations* are used to transform data values to a form that can be

optimally handled by the specific modeling method. *CoDa transformation* tools are used for compositional data (such as geochemical concentrations).

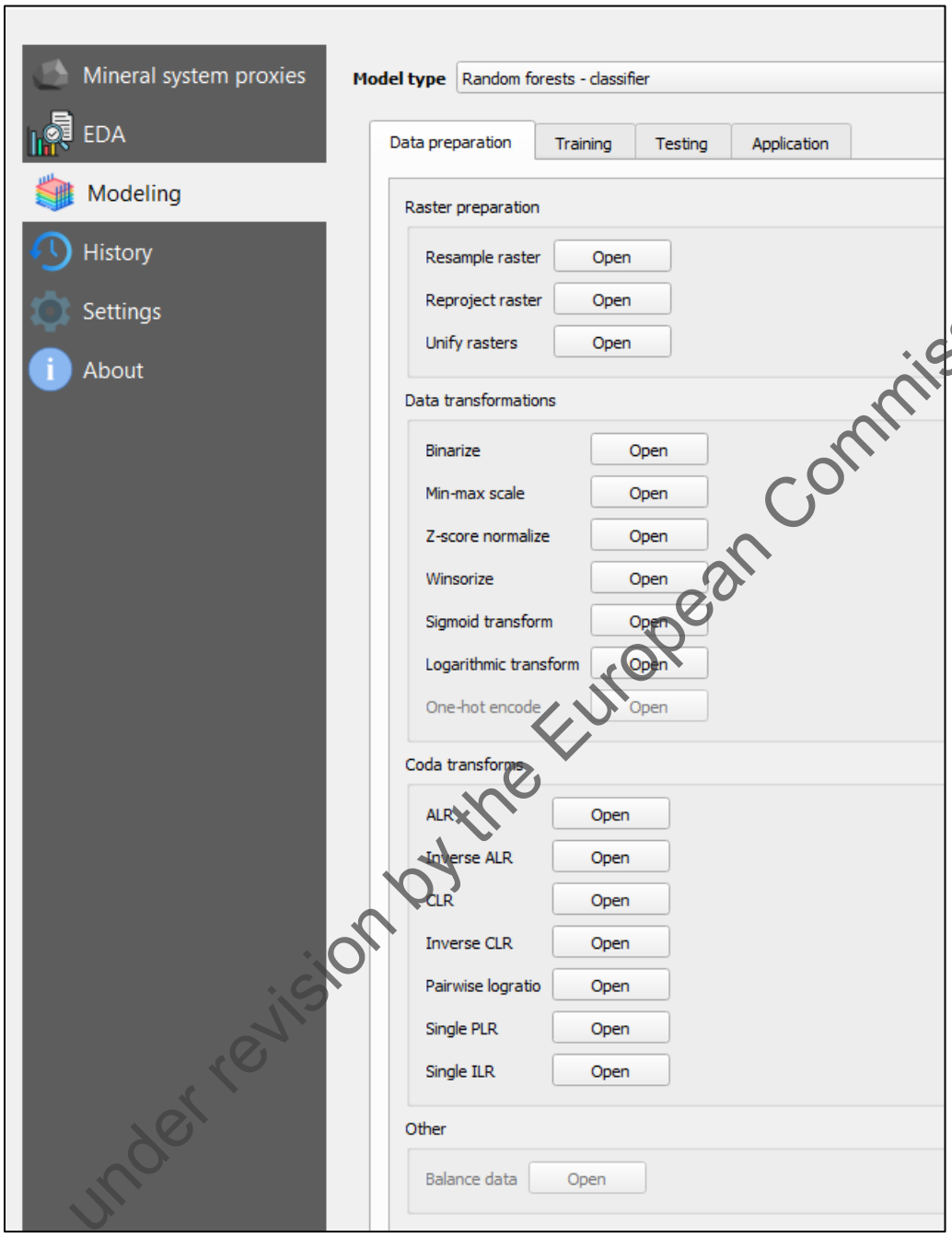


Figure 8: Data preparation tab of the "Modeling" page in EIS Wizard.

In the *Training* tab (**Figure 9**), input data and model specific parameters for training are provided. The user should also provide a name for the model and the path where to save the model. The *Tag* parameter for identifying the input data parameters is obligatory, but this can also be autogenerated by pressing the *Generate tags* button. Tags are used for matching the proxies in the correct order when applying the model to a new dataset. This means that the proxies in the new data set can be in a different order, and they are automatically matched using the tags.

The *Labels* input parameter refers to locations of known occurrences of the modeled quantity. The locations are used to generate training data for supervised modeling methods.

Model name parameter is used to identify the model, when referring to it in the testing and application phases (handled on the *Testing* and *Application* tabs as well as the *History* page).

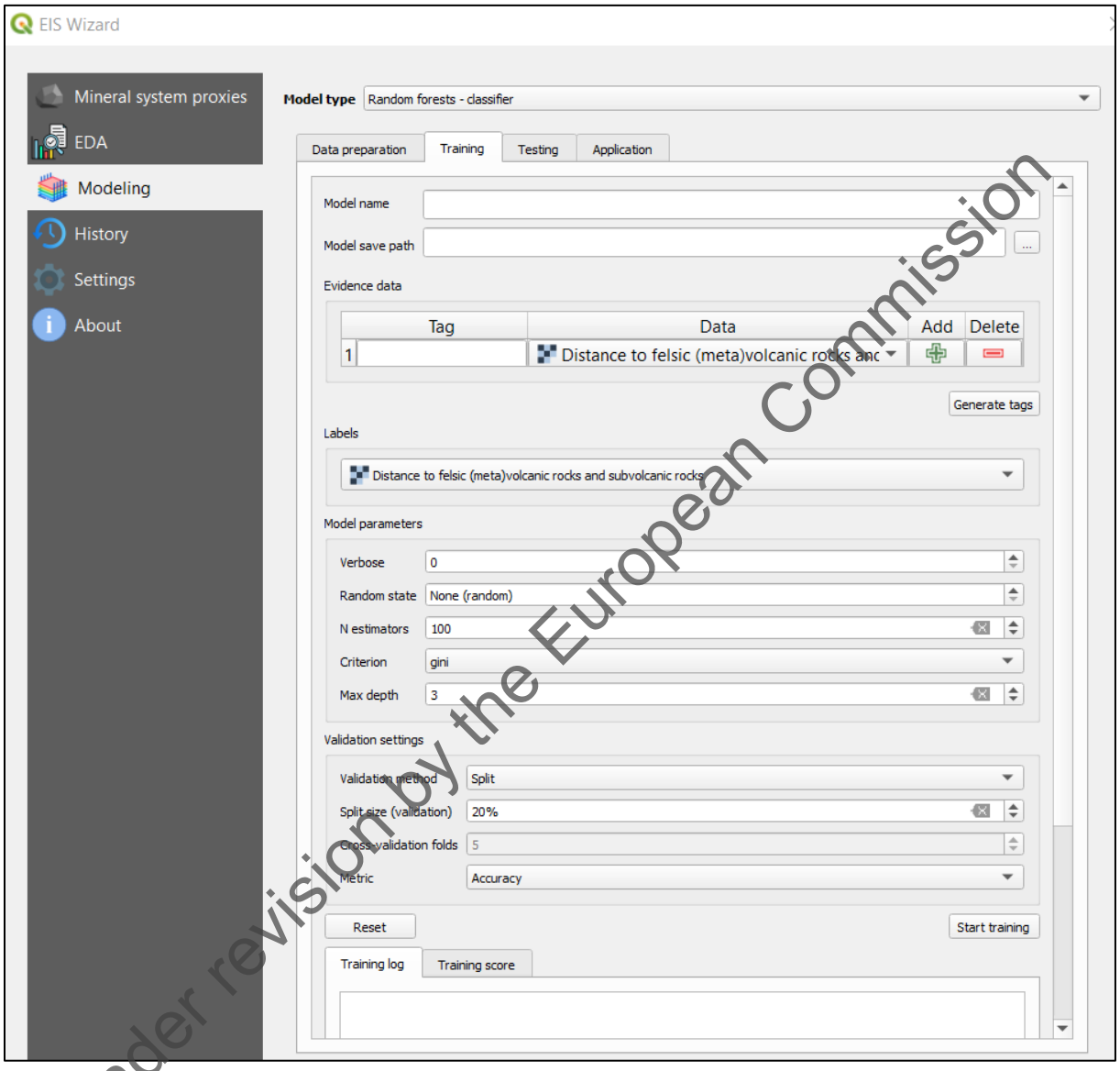


Figure 9: "Training" tab of the "Modeling" tab page in EIS Wizard.

The *Testing* tab (**Figure 10**) offers several methods to test the model using testing data. *Model name* and *Model* file parameters should be as given in the *Training* stage. Currently testing data is a set of known occurrence locations file that is given as the *Labels* input parameter.

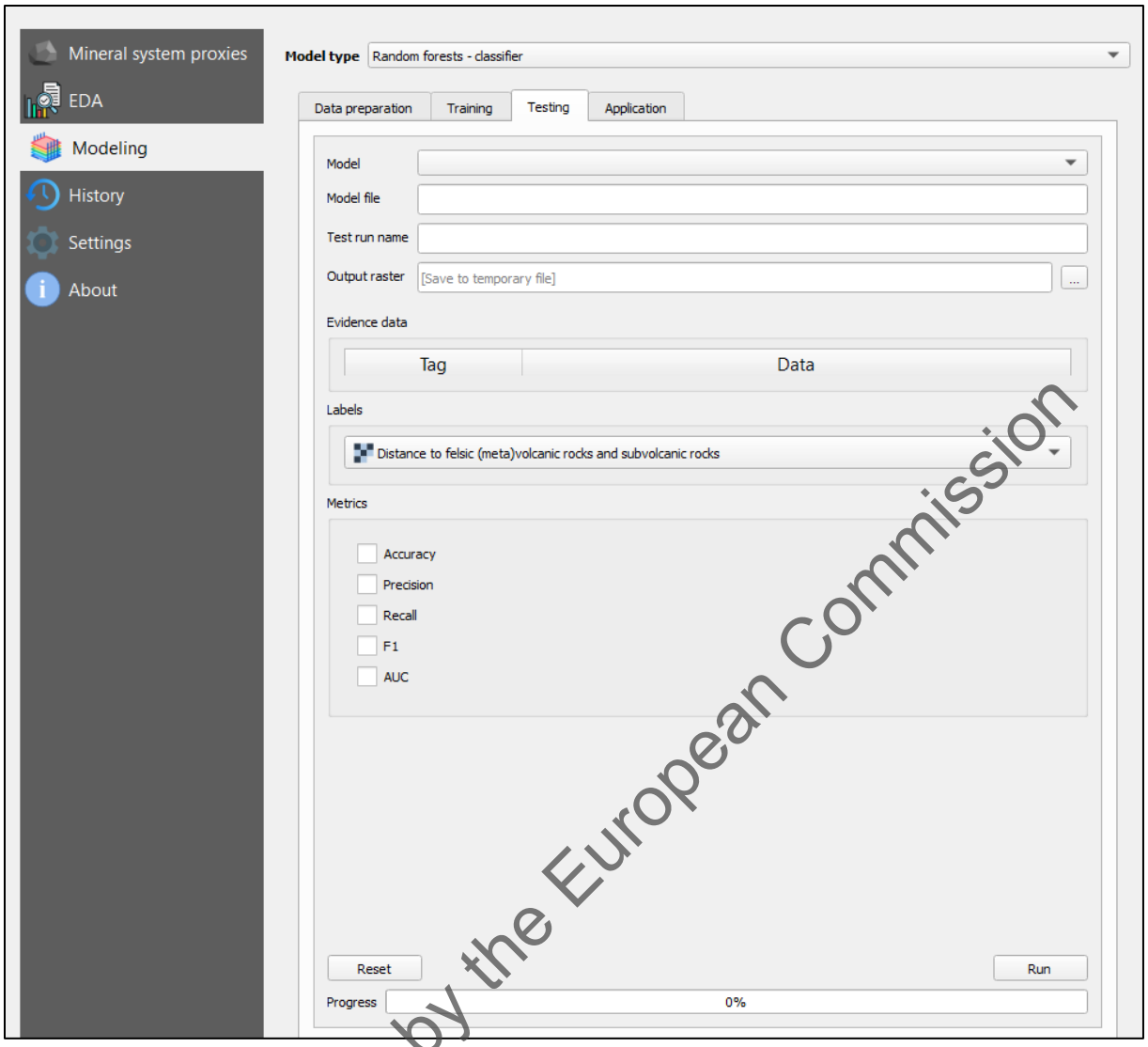


Figure 10: "Testing" tab of the "Modeling" tab page in EIS Wizard

The *Application* tab (**Figure 11**) is used for applying a model that was generated using the functionalities in the *Training* tab, to a new data set. *Model name* and *Model* file parameters should be as given in the *Training* stage.

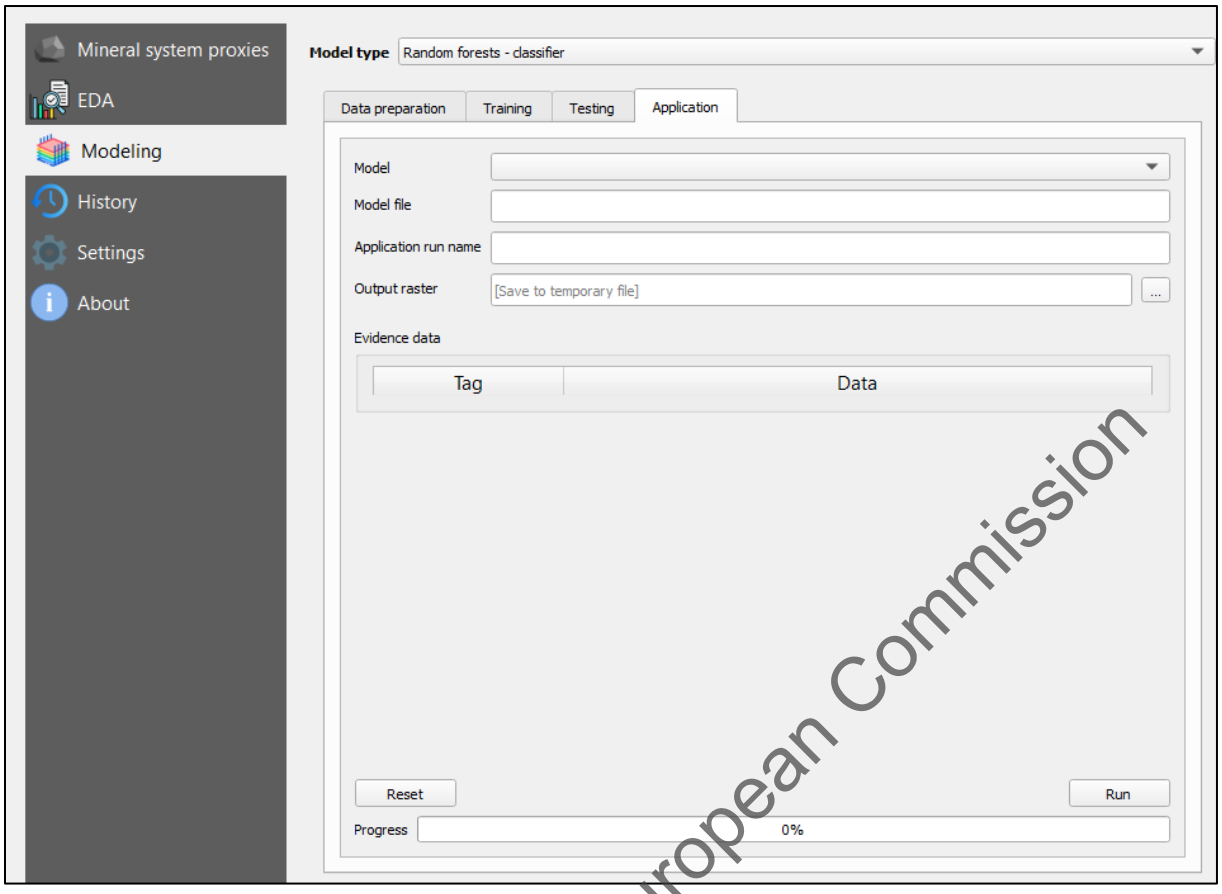


Figure 11: "Application" tab of the "Modeling" tab page in EIS Wizard.

2.4 History

The *History* view offers access to information related to previously computed models. *Model name* and *Model file* parameters should be as given in the *Training* stage.

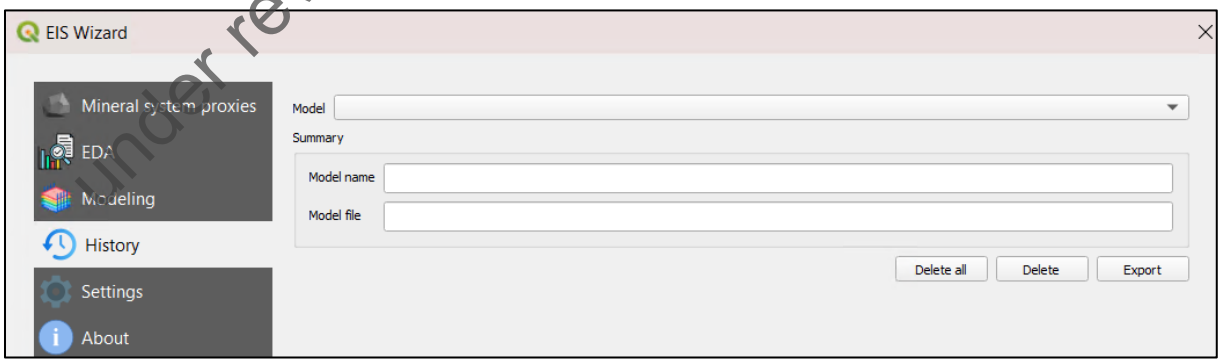


Figure 12: "History" tab page in the EIS Wizard.

2.5 Settings

The *Settings* view should be visited first to configure EIS Toolkit which is the Python library that EIS Wizard uses. In addition to this, on the *Settings* page (**Figure 13**) you can also choose whether to use the docked EIS Wizard, set a base raster, configure default colors and palette and set how produced layers are organized in the QGIS layer tree.

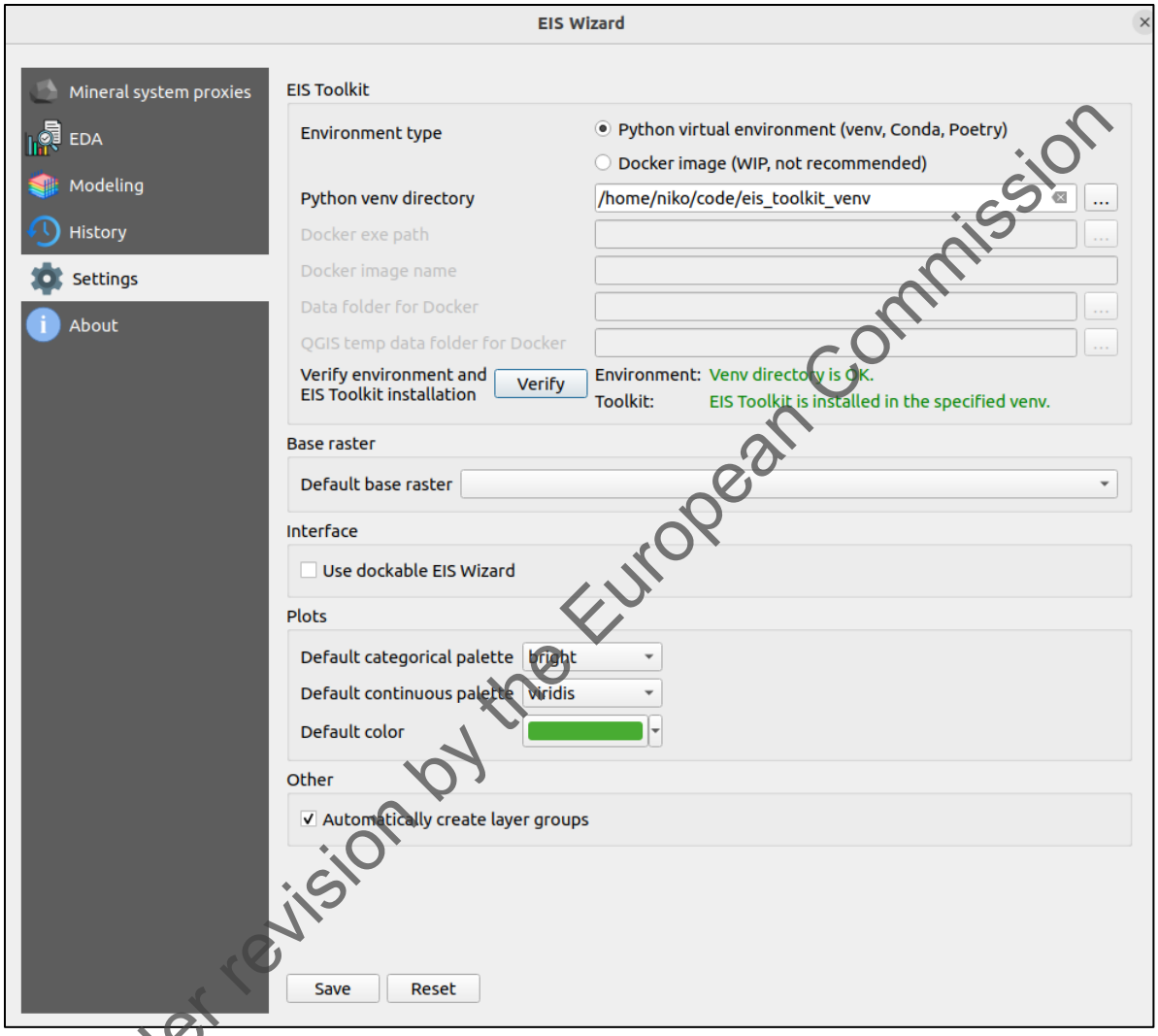


Figure 13: The "Settings" tab page in EIS Wizard.