DRG4 FOOD TOOLBOX

QC-Tech360 Documentation

Faculty of Economics and Management

DEPARTMENT OF INFORMATION TECHNOLOGIES FEM CZU PRAGUE

Czech University of Life Sciences Prague

QC-Tech360 Documentation

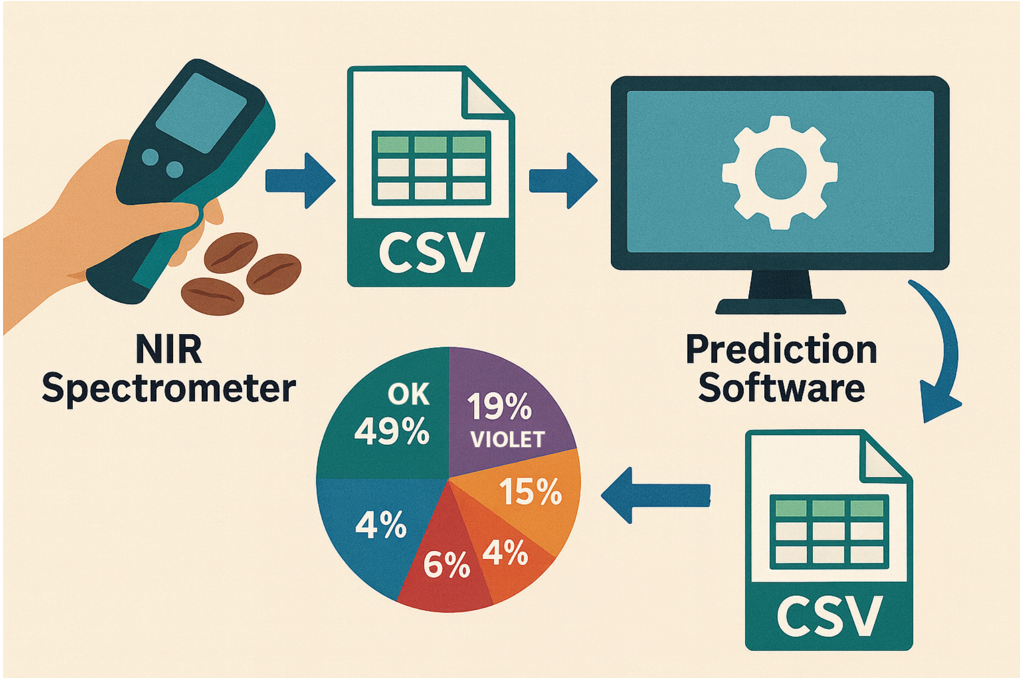
Documentation for the Quality Control module for Cacao Tech 360

Last updated: 30.10.2025

Please note that the document is updated regularly.

• Overview QC-Tech360

The Quality Control (QC) module of the CacaoTech tracking and tracing solution enables rapid and automated assessment of cacao quality directly at the farm gate. Using a portable NIR sensor, it provides non-destructive, objective evaluations that help reduce waste while improving both time and cost efficiency. Designed for robustness, it operates fully offline and remains interoperable with CacaoTech360 as well as other traceability platforms, making it a scalable and future-proof solution for quality verification across the cacao value chain.



•.1 DRG Tools Used

NAME (Toolbox Database - drg4foodtoolbox.eu)	Description and Application in the Pilot
Core Data Science & Machine Learning	
TensorFlow (https://www.tensorflow.org/)	Underlying python framework used during development of AI models for NIRS calibration
NumPy (https://numpy.org/)	Numerical computing library
Pandas (https://pandas.pydata.org/)	Data manipulation and analysis
Scikit-learn (https://scikit-learn.org/)	Machine learning algorithms and tools
SciPy (https://scipy.org/)	Scientific computing library
Machine Learning Optimization	
Optuna (https://optuna.org/)	Hyperparameter optimization framework
Visualization	
Matplotlib (https://matplotlib.org/)	Plotting and visualization
Seaborn (https://seaborn.pydata.org/)	Statistical data visualization
Utilities	
Joblib (https://joblib.readthedocs.io/)	Serialization and parallel computing
Pydantic (https://docs.pydantic.dev/)	Data validation and settings management

Development Tools	
Pytest (https://docs.pytest.org/)	Testing framework
Black (https://pypi.org/project/black/)	Code formatter
Isort (https://pycqa.github.io/isort/)	Import sorter
Mypy (https://mypy-lang.org/)	Static type checker
Ruff (https://docs.astral.sh/ruff/)	Fast Python linter and formatter
Package Management & Build Tools	
setuptools (https://pypi.org/project/setuptools/)	Python packaging
uv (https://docs.astral.sh/uv/getting-started/features/)	Fast Python package installer and resolver
Version Control	
Git (https://git-scm.com/)	Code versioning
Explainability & Model Interpretation	
Dalex (https://dalex.drwhy.ai/)	Considered to use later in AI models' development to understand how the model interprets the NIR signal
Explainable AI (https://insights.sei.cmu.edu/blog/what-is-explainable-ai/)	Article, read, considered while developing the AI models within the project
AI Explainability 360 (https://aix360.res.ibm.com/)	Considered to use later in AI models' development to understand how the model interprets the NIR signal
Regulations & Ethics	
AI Security and Privacy Guide (https://owasp.org/www-project-ai-security-and-privacy-guide/)	Article, considered for final deployment of the AI models
EU guidelines for trustworthy AI (https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai)	Considered, considered while developing the AI-based solution within the project
AI Act (https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai)	Considered, taken into account for the development and deployment of the AI models
EU AI Act (https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence)	Considered, taken into account for the development and deployment of the AI models
The AI Act Explorer (https://artificialintelligenceact.eu/ai-act-explorer/)	Used to explore the AI acts mentioned above

●.2 Current Status

Minimum Viable Product (MVP)

Technology Readiness Level (TRL) – **6**

- Technology demonstrated in a relevant environment.
- See description at EU and NASA.

User-Centred Customization and Onboarding

TRL6 means that each of the existing systems and models will have to be validated for the intended use as specified by the user. Consequently, new models might need to be co-developed for the well-defined use (e.g. for other cacao qualities or for other crops, fruits, vegetables or even spices) with the user in order to deliver the results as per the expectations. Once the use-case is defined and validated, user might require technical training before s-he will be provided with the customized software and models relevant for his/her use-case.

Your Path to Quality Control: A 3-Step Onboarding Process

We have designed a simple and transparent process to get you started with QC-Tech360.

Step 1: Explore and Define Your Use-Case

Begin by exploring our platform and defining your specific needs. We provide materials to help you understand the capabilities of QC-Tech360.

- **Project Overview:** Learn more at our project portal FARM2FORK
- **Promotional Video:** Watch a quick introduction to the technology
- **Software Demo**: See the software in action
- **Interactive Demo:** Access a demo version of the software with a sample dataset

  QCtech360 - Model runner - https://github.com/kit-pef-czu-cz/QCtech360
  QCtech360 - Model example - https://cacaotech.eu/docs/cacao-predication.zip

After reviewing these materials, we will work with you to formally define and validate your use-case.

Step 2: Licensing and Implementation

Once your use-case is validated, we will provide a customized solution tailored to your requirements.

- **Software Licence:** You will be provided with a licensing agreement that specifies the terms, conditions, and scope of use for the software and predictive models. The cost of the solution depends on the complexity and scale of your use-case.
- **Software Delivery**: You will receive instructions to download the custom application, complete with the pre-validated models relevant to your needs.
- **Hardware Delivery (MicroNIR, VIAVI):**
  https://www.viavisolutions.com/en-us/osp/products/micronir-spectrometers
  Users are not required to purchase their own hardware unless the use-case has been validated. To validate the case, hardware will be enabled from the providers (i.e. CZU).

Step 3: Training and Support (Recommended)

To ensure you achieve the best possible prediction accuracy and seamlessly integrate QC-Tech360 into your workflow, we highly recommend our training and support packages.

- **User Training:** We offer training sessions on how to operate the system, perform optimal sample scanning, and interpret the results correctly.

- **Ongoing Support**: For long-term success, you can opt for our support services, which include regular model maintenance, performance monitoring, and quality prediction control.

For more details, a personalized consultation, or a quote based on your use-case, please contact:
Jana Kholova at **kholova@pef.czu.cz**

● Architecture

This solution supports end-to-end acquisition, processing, and modeling of spectral data, with a focus on classification- and regression-based quality control, where the goal is to assign each sample to a specific category or trait value based on its spectral profile. The system is modular, transparent, and designed for flexibility in both development and deployment.

●.1 Development Technologies & Frameworks

We use a modern Python-based tech stack designed for scalable data analysis, preprocessing, and machine learning model development.The key components include:

- **Programming Language:** Python 3.11
- **Data Processing:** Efficient tabular and numerical processing frameworks (Pandas, NumPy, SciPy, Kennard-Stone)
- **Machine Learning:** Established libraries for classification tasks and domain-specific algorithms (scikit-learn, TensorFlow)
- **Spectral Data Handling:** Specialized techniques for spectroscopy signal normalization and correction (e.g., SciPy signal processing, Savitzky-Golay filters via scikit-learn, custom SNV (Standard Normal Variate) implementation, Derivative Transformation)
- **Configuration Management:** Type-safe and structured configuration modeling (Pydantic), ruff for linting and formatting
- **Project Structure:** Modular codebase using virtual environments, type annotations, and reusable components (Docker for containerized build and deployment; setuptools/pip dependency management)
- **Architecture is Modular** to ensure solution flexibility, reproducibility, maintainability, and adaptation to other tasks

●.2 QC-Tech360 Modules & Components

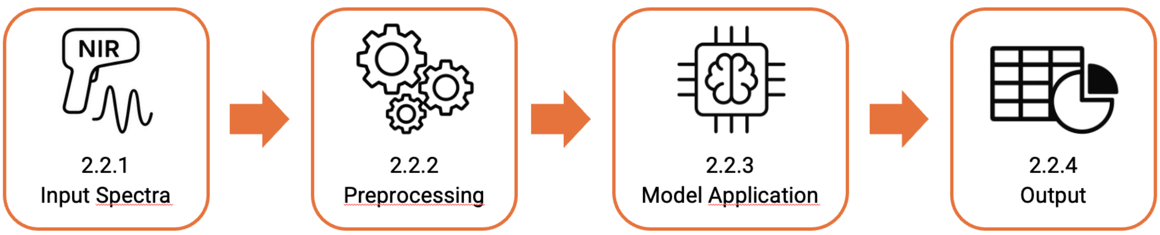


Diagram #1 shows the system architecture and points out to the following section where each part of the process is described in more details.

●.2.1 Input Spectra

Use the input data generated by the MicroNIR Pro SW from VIAVI. The spectral acquisition method is described in section 3, and the installation of the QC-Tech360 prediction system is described in section 4; the whole system operation is described in section 5.

- **Input Format**: The user provides input as a .csv file containing spectral data collected via the MicroNIR device. Each row represents one scan, and each column corresponds to a wavelength-specific reflectance value.
    - **File format**: Exported as .csv (ASCII) from MicroNIR software
        - **Separator:** Semicolon (`;`)
        - **Decimal Format:** European format (comma `,` as decimal separator)
        - **Encoding:** UTF-8 or ASCII
    - **File Structure:**
        - **ID Column:** Bean unique identifier format: `*XXX-XXX-BEAN_TYPE-CATEGORY-Origin-Details--N.sam*`
        - **Example:** `*004-001-CUT-VIOLET-IvoryCoastBulk-x150MT--1.sam*`
        - **Spectral Columns**: Wavelength measurements (typically 125 columns)
- **Spectral Acquisition**: This data is typically generated directly from the NIR sensor device (VIAVI MicroNIR, details in section 4). It should follow the expected structure: numeric reflectance values per wavelength, one sample per row.
- **Input Validation:**
  Before running any transformations or predictions, the input undergoes an automatic validation process:
    - The system compares each new scan against a simple validation model of invalid/bad scans
    - If the scan falls outside expected waveform, it is flagged
    - Flagged inputs are returned with a label “BAD scan” in the output file
- **Validation model Details**:
    - Outlier detection is applied using statistical methods (e.g., Mahalanobis distance, Z-score)
    - Configurable thresholds that can be adjusted based on data requirements
    - AI model is under development

●.2.2 Preprocessing - Spectral Transformation

This step handles the cleaning and normalization of spectral data to prepare it for model prediction. It ensures that all inputs are standardized and ready for meaningful interpretation.

- **Main Preprocessing Operations**:
    - Smoothing (e.g., Savitzky-Golay filter) reduces high-frequency noise
    - Normalization (e.g., SNV): standardizes intensity across samples
    - Derivative computation: enhances spectral features for better pattern recognition

- **Configurable Pipeline**: The preprocessing steps are modular and customizable: the exact combination can vary depending on the prediction task (e.g., CUT test vs. origin classification)
- **Implementation Details**:
  - Libraries: SciPy, custom implementations of SNV
  - Transformations are applied in a fixed sequence defined in the config file
- **Output of this step**: Preprocessed spectral data is ready for prediction

●.2.3 Model Application

Once the input spectrum has passed preprocessing and normalization, it is passed to a pre-trained machine learning model that performs classification based on learned patterns.

- **Validated Model:**
  - The model was trained on labeled spectral data with reference values from laboratory wet chemistry tests
  - Model type: currently a classification model (e.g., Random Forest, or TensorFlow-based classifier), chosen based on best performance during training
- **How prediction works**:
  - The processed input is passed to the model
  - The model returns a prediction label (e.g., "OK") and a confidence score
  - If the input data is flagged as invalid (e.g,. detected as a bad scan earlier), the model step is skipped, and the result is returned as "BAD scan"
- **Current Classification Tasks Implemented**:
  - CUT test classification: detects whether the bean passes a quality test
  - Others in progress – see *Description of prediction models* for details.
- Model inference runs locally using embedded model weights in the .exe package.

●.2.4 Output – Predictions

The results are the final output of the module - shown to the user or stored for further use. They include the prediction and status information.

●.2.4.1 Fermented Cacao Beans

- **Prediction Output:** The core output is a classification result - e.g., the bean passes the CUT test. Prediction label (e.g. "OK", "other")
- **Format:** CSV files with timestamped names
- **Data Fields:**
  - **lot_ID**: Batch identifier
  - **ID**: Unique sample ID, aligned with input naming convention
  - **predicted_class**: Binary class (0 or 1)
  - **class_probability**: Confidence score for the predicted class (0.0 - 1.0)
  - **predicted_origin**: Predicted country of origin
  - **origin_probability**: Confidence score for origin classification
  - **moisture_content**: Predicted moisture percentage or range (e.g., below 10)
  - **moisture_probability**: Confidence score for moisture prediction

    - **fat_content**: Predicted fat percentage
    - **fat_probability**: Confidence score for fat prediction
    - **datetime_measurement**: Time when the sample was measured (UTC)
    - **datetime_prediction**: Time when the prediction was made (UTC)
    - **serial_number**: Serial number of the NIR device used for scanning
- **Error Handling**:
    - If the input was flagged "BAD scan" during Validation or preprocessing, the predicted_class column will include this label instead of a prediction
    - Helps prevent misinterpretation of unreliable data
- **Integration Possibilities**:
    - Results can be directly uploaded to tracking software
    - See section *6 Interaction with tracking-tracing software - CacaoTech360* for details.

●.2.4.2 Cacao Pulp

- Will be added later

● Measurement and generating spectral data using VIAVI MicroNIR

●.1 MicroNIR description & measurement

●.1.1 Device Overview

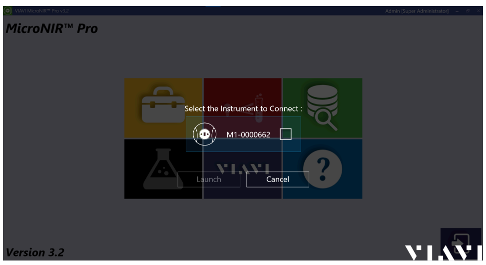
- **Purpose**: Portable NIR spectrometer for rapid, non-destructive quality assessment of cacao beans and pulp in field and lab conditions.
- **Key features**:
  - Handheld device for on-site use
  - ~1s per bean scan time
- **VIAVI documentation**:
  - https://www.viavisolutions.com/en-us/literature/micronir-onsite-w-data-sheets-en.pdf
- **MicroNIRS OnSite-W by VIAVI**
  - https://www.viavisolutions.com/en-us/osp/products/micronir-onsite-w

- **_MicroNIR Pro Software User Manual Version 3_2-ED000124-01_Rev011**

  - _MicroNIR Pro Software User Manual Version 3_2-ED000124-01_Rev011.pdf

●.1.2 Connection & Setup

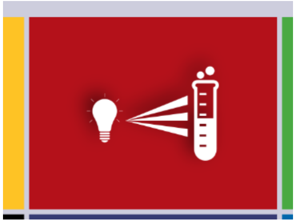
- **Power on:** Ensure device is ON before pairing
- **Connection options**:
  - USB-A cable
  - Bluetooth (turn on PC Bluetooth, find device, connect)
- **Login**:
  - Username: admin
  - Password: micronir



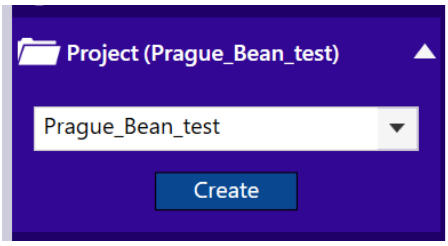
- **Pre-measurement:** Check connection and clean lens before calibration

●.1.3 Measurement Procedure

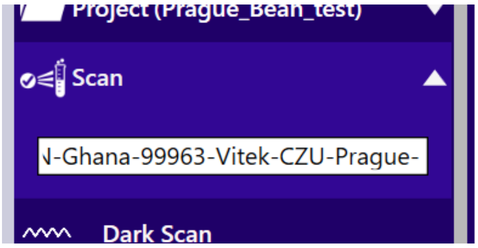
1. **Creating a project**:
   a. Go to developer's space



b. Create new project



c. Name the scan using the FAIR-compliant naming convention. (See Appendix A: Naming Convention Details - CUT Test for detailed instructions and examples).

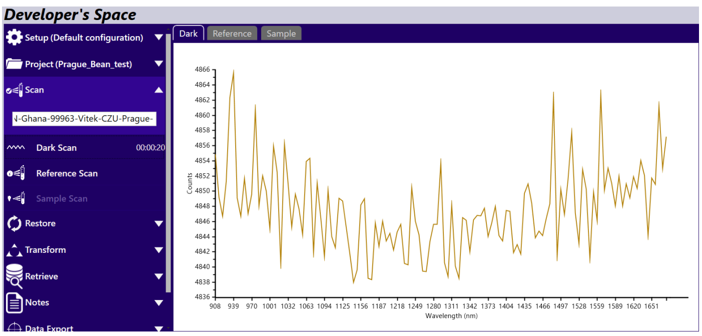


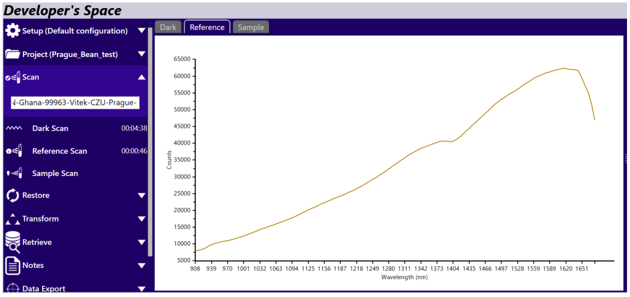
2. **Calibration:**

Before taking measurements, the device must be calibrated. This is a critical step to ensure the accuracy of the spectral data. The process involves two scans:

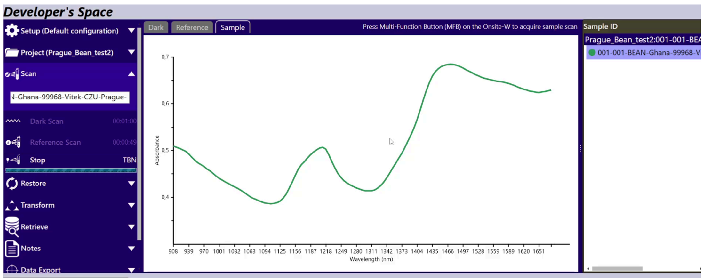
a. Dark Scan (standing position glass base). This measures the instrument's baseline noise.
b. Reference Scan (standing position on glass base). This calibrates the instrument against a known standard.

For detailed, step-by-step instructions on performing the calibration procedure, see VIAVI MircroNirs Pro Software User Manual, Section 4.3 ("Scan"). The link is provided in **Section 3.1.1** of this document

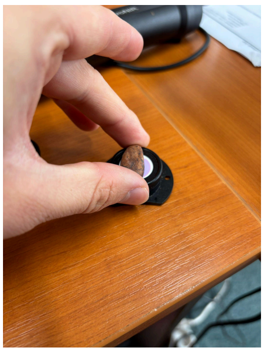


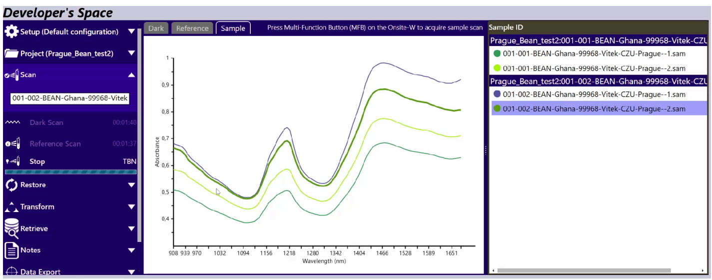


3. **Scanning**:
    a. Click Sample Scan and press button on device

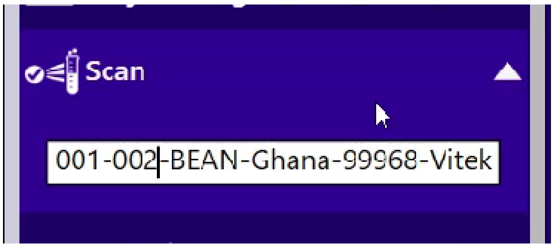


    b. Position bean, twist 180°, rescan



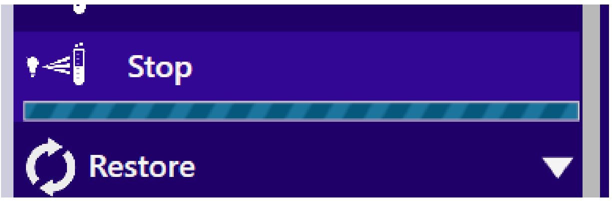


    c. Manually rename scans for new beans

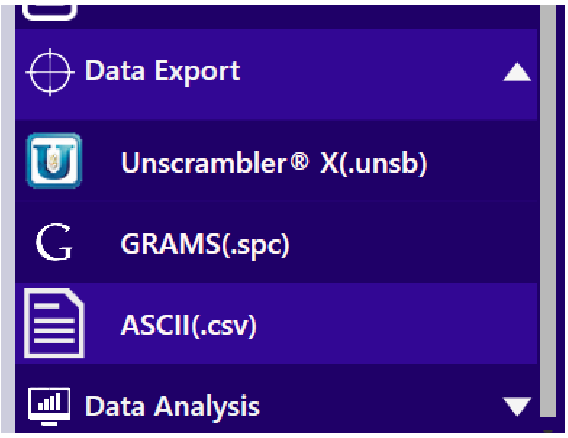


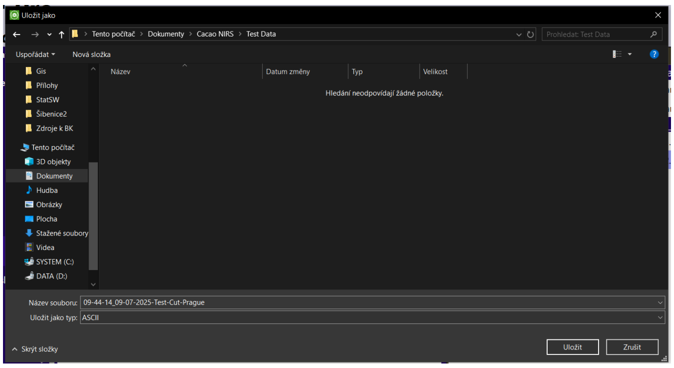
•.1.4 Data Export

- During long sessions -> software may request lens cleaning -> STOP -> clean -> redo Dark & Reference scans



- **When finished**:
  a. STOP
  b. Export ASCII .csv
  c. Save in a predefined location





● Prediction Application Installation

●.1 Setup

Instructions for installation and launching the application.

●.1.1 Prerequisites

What the user needs before running the app

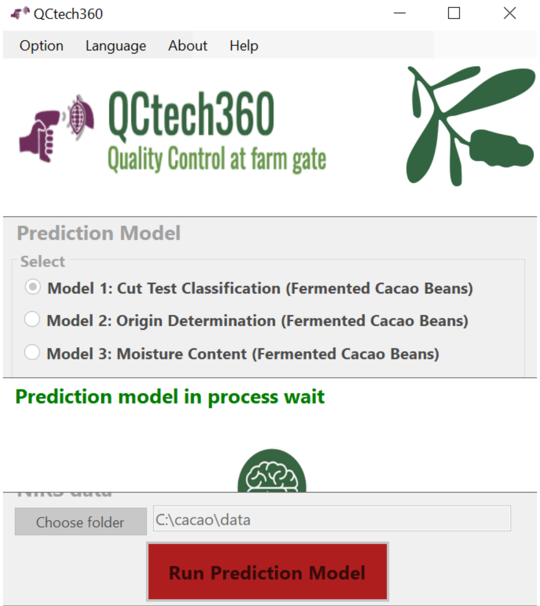
- **Supported OS**:
  - Windows 10 or later (64-bit)
- **Additional:** No Python installation required (all dependencies bundled)

●.1.2 Model validation and installation procedure

Steps to get the app up and running (also see section 1.2)

1. **Software demo - hands on modeling**
   Users can access the demo version of the software with the basic models and testing dataset here:
   QCtech360 - Model runner - https://github.com/kit-pef-czu-cz/QCtech360
   Contains the source code of the application that runs the basic quality prediction model (doesn't include the model *per-se* as its too large to place in the GitHub repository. Model is here:
   QCtech360 - Model example - https://cacaotech.eu/docs/cacao-predication.zip (this link is also at the GitHub link above).

   Start by loading the QCtech360 Model example .zip file, extract the content and save the folder called "cacao" and "QCtech360 - model runner" to your disk (C:).
   Open the "QCtech360.exe" stored in the folder "QCtech360 - model runner"
   This opens the QCtech360 UI on your desk-top (it can take a minute):



Now you need to disable the demo mode in the “Option” field
Then you click “Run Prediction Model” (this can take a minute)
When the results are ready, the application will notify you and the results are automatically visualized on a pie chart while the details of the prediction are available as a .csv file in the folder “cacao” you originally stored at your C:\ drive.
! note: at the moment, the application runs only the “cut test” prediction regardless of the model you would choose.

2. **User access to software licence and relevant models**
   At the moment, each of the existing models will have to be validated for the intended use as specified by the user (see section 1.2). New models might need to be co-developed with the user in order to deliver the results as per the user expectations. Once the use-case is defined and validated, user might require customized technical training.
3. **User training**
   For the best prediction accuracy, and according to the intended use, the user might require the training to scan the material and use the model.
4. **User will be provided**
   After defining the use-case and validating prediction models, user will be provided
   a. Instructions to download the relevant application and validated model
   b. Consequently, user will download the `Predictor.zip` file from the latest release validated and customized for the particular use and context
   c. Size of the application is approximately 900MB (includes all dependencies)
5. **Extract the Archive**
   a. The extracted archive should have structure similar to the following:

```
Predictor/
├── PredictionsOO.exe          # Binary classification executable
├── PredictionsOVS_MWI.exe     # Multi-class classification executable
├── data/                      # Place CSV files here
```

```
├── models/                         # Model files (pre-installed)
├── results/                        # Output results folder
└── README.md                       # Quick start guide
```

6. **Launch the application**
   a. Double-click on the desired executable:
      i. PredictionsOO.exe
      ii. PredictionsOVS_MWI.exe

●.2 Running the prediction application

●.2.1 Prepare Input Data

Place the input .csv files into the *data* folder. Ensure the files are formatted according to the specifications described in **Section 2.2.1**.

●.2.2 Launch the Application

Run the appropriate .exe file (e.g., CacaoPredictionsOO.exe) based on the required prediction task (see *Description of prediction models* section).

●.2.3 Check the Results

The application will process the input files and save the output files with timestamps in the results folder. The structure of the output is detailed in **Section 2.2.4**.

●.2.4 Troubleshooting

- For common issues related to the physical measurement process (e.g., calibration errors), refer to **Chapter 3**.
- If the application fails to produce an output, first verify that the input data format strictly follows the requirements in **Section 2.2.1**.

● Description of prediction models

This section provides a description of the predictive machine learning models that form the core of the QC-Tech360 system. These models analyze spectral data acquired by the MicroNIR device to perform rapid, objective assessments of key quality characteristics of cacao beans and pulp. Each model is trained on an extensive dataset correlated with laboratory reference values.

●.1 Fermented Cacao Beans

These models are designed to analyse spectra obtained from the surface or the cut of whole fermented and dried cacao beans.

●.1.1 Model 1: Cut Test Classification (Fermentation Quality)

Model Purpose: To automate and objectify the "Cut Test," a standard procedure for assessing the degree of cacao bean fermentation based on the color of its cross-section. The model determines the quality of fermentation.
Model Type: Classification models (e.g., Random Forest, TensorFlow-based neural networks) are used to assign beans to quality categories.
Predicted Categories: Binary Classification

- **OO** Model (CacaoPredictionsOO.exe):
  - **O**K: Fully fermented beans
  - **O**THERS: Beans with fermentation defects (purple, slaty, moldy, etc.)
- **OVS-MWI** Model (CacaoPredictionsOVS_MWI.exe):
  - **O**K: Fully fermented
  - **V**IOLET: Partially fermented
  - **S**LATY: Unfermented
  - **M**OULDY: Affected by mold
  - **W**SPOTS: White Spots
  - **I**NSECT: Insect damage

Data Processing Specifics: To achieve high accuracy, spectral preprocessing is applied, including normalization (e.g., SNV) to correct for light scattering and the use of spectral derivatives to enhance characteristic peaks associated with chemical changes during fermentation.

Current Accuracy:

- for **OO:** 80-85%
- for **OVS-MWI:** 93-95%

Status: MVP

●.1.2 Model 2: Origin Determination (In development)

Model Purpose: To predict the country or region of origin of cacao beans based on unique "fingerprints" in their spectra, which are influenced by cultivar, soil composition, and climatic conditions.
Model Type: Multi-class classification.
Predicted Categories: Country or region of origin (e.g., Ghana, Ivory Coast, Ecuador, etc.).
Status: Research and evaluation phase.

●.1.3 Model 3, 4 - Moisture and Fat content

Model Purpose: Regression models to predict continuous numerical values.
Moisture Content: Predicting the percentage of moisture.
Model Type: Regression
Fat Content: Predicting the percentage of cocoa butter.
Status: Research and evaluation phase

●.2 Models for Cacao Pulp (Quantitative Prediction)

These models are designed to analyse spectra obtained from whole cacao pulp (fruit) when harvested.

●.2.1 Model 1: Prediction Models for pH and Brix Determination

Model Purpose: To predict the acidity level (pH) and sugar content (Brix) in the pulp. These parameters are critical indicators of the stage and correctness of the fermentation process.
Model Type: Regression models and classification to certain levels.
Predicted Values:

- pH level (e.g., 4.5)
- Brix level (e.g., 12%)

Status: Research and evaluation phase.

●.2.2 Prediction Model for Cultivar Classification

Model Purpose: To predict the original cultivar - CCN or National.
Model Type: Binary classification.
Predicted Values:
Status: In the research and evaluation phase.

● Interaction with tracking-tracing software - CacaoTech360

●.1 Purpose of Integration

- Ensure traceability from measurement (MicroNIR scan) to final product certification
- Integrate prediction results into the TT-system (tracking-tracing) alongside metadata (farmer, location, crop ID)
- Make data accessible via dashboard and available for downstream processes (e.g., reporting, certification)

●.2 Upload Workflow

1. Login to the CacaoTech Web App
2. Upload spectral data (.csv from MicroNIR device)
3. Select scan type: e.g., CUT test, Origin, Fat, Moisture
4. Assign batch to correct fermentation lot / bean category
5. Manually input metadata (if not auto-filled):
    a. Farmer ID
    b. Crop Year
    c. Origin
    d. Certifier
    e. Number of scans
6. Preview assigned records
7. Submit to server for validation

●.3 Validation & Feedback

- System checks for
    - Valid format and naming (FAIR-compliant ID)
    - Missing fields or conflicts
    - Metadata consistency
- UI provides:
    - Green checkmarks for validated entries
    - Error indicators for incomplete/missing data
- User can edit and re-submit if needed

●.4 Integration with TT-System

- After submission:
    - The scan and its predicted result (e.g., OK/ssss class) are linked to the corresponding crop entry in the TT-system
    - Result appears in the Tracking Dashboard along with:
        - Farmer details

        - Bean quality classification
        - Certifier info
        - Date of scanning
- Allows for auditing, reporting, and export to external systems

● Appendix

●.1 A: Naming Convention Details - CUT Test

To ensure data is FAIR-compliant, each scan must be named according to the following structure. The components are separated by a hyphen (-).

Structure: ItemID-BeanID-Type-Cultivar-Lot-Author-Company-Location-SampleID
Component Breakdown:

- **ItemID**: A unique identifier for the lot or fruit batch being tested.
  - Example: 001
- **BeanID**: A unique number for the specific bean being scanned within the lot.
  - Example: 002
- **Type**: The type of test being performed.
  - Example: CUT (for a cut test), BEAN (for a whole bean scan).
- **Cultivar**: The variety of the cacao bean.
  - Example: Ghana
- **Lot**: The specific lot identifier provided by the supplier.
  - Example: 99963
- **Author**: The initials or name of the person performing the scan.
  - Example: Vitek
- **Company**: The name of the organization conducting the test.
  - Example: CZU
- **Location**: The city or lab where the measurement is taken.
  - Example: Prague
- **SampleID**: The sample number automatically generated by the MicroNIR software (usually 1 or 2 for each bean).

Full Example:
A scan named ***001-002-BEAN-Ghana-99963-Vitek-CZU-Prague-1*** represents: The first scan (SampleID=1) of the second bean (BeanID=002) from lot 99963 (Lot=99963), which is a Ghana cultivar. The bean test (Type=BEAN) was performed by Vitek (Author=Vitek) for CZU (Company=CZU) in Prague (Location=Prague). The entire batch has the ID 001 (ItemID=001).

●.2 B: Naming convention Details - Brix and Ph

To ensure data is FAIR-compliant, each scan of this type must be named according to the following structure. The components are separated by a hyphen (-).
Structure: FarmerID-FruitID-Type-Category-Cultivar-Lot--SampleID
Component breakdown:

- **FarmerID:** A unique identifier for the lot or fruit batch being tested.
  - Example: Market01
- **FruitID**: A unique number for the specific bean being scanned within the lot.
  - Example: 002
- **Type**: The type of test being perform
  - Example: Surface, Pulp, TwoSeeds, SevenSeeds

- **Category**: Category of test based on the Type
  - Example: **BrixPH (**Surface), **BrixPH** (Pulp), **PH** (TwoSeeds), **Brix (**SevenSeeds)
- **Cultivar**: The variety of the cacao bean.
  - Example: CCN, National
- **Lot:** The specific lot identifier provided by the supplier.
  - Example: 99966
- **Author**: The initials or name of the person performing the scan.
  - Example: Vitek
- **Company**: The name of the organization conducting the test.
  - Example: CZU
- **Location**: The city or lab where the measurement is taken.
  - Example: Prague
- **SampleID**: (round) generated by NIRS SW - 6 and more samples per

Full Example:
A scan named Sanjose-2-SevenSeeds-Brix-CCN-x-8.sam ……