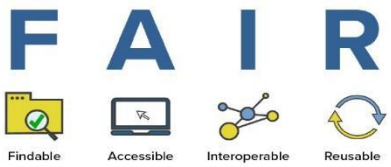
Funded by
the European Union

AGILE data tracking guidelines



Contents

1. **Purpose and scope** 1
2. Why unique identifiers, not self-given names 2
3. How the identifiers link 3
4. Images 3
5. Statistics 4
6. Workflows 5
7. Models 5
8. Tracking Samples and Scripts (optional but highly recommended) 6
9. Golden rules 7
10. Where data are stored, and when: the three stages 8
11. References and resources 9

1. Purpose and scope

Every image, dataset, workflow and model produced in AGILE gets a short, unique identifier built on the producer's Doctoral Candidate (DC) number, such as DC12_001. Self-given descriptive names are not used as identifiers. These guidelines explain why, and how to do it for each data class. The AGILE Data Registry (Excel template, one tab per class) is the place where every identifier is recorded together with its description.

Data class	What it is	Identifier	Registry tab
Images	Every acquired image or image dataset	DCx_001	Images
Statistics	Measurements from several images, consolidated in one dataframe	Statistics_DCx_001	Statistics
Workflows	Image-analysis pipelines	Workflow_DCx_001	Workflows
Models	Trained machine-learning / AI models	Model_DCx_001	Models
Samples (optional)	Slides and specimens prepared for imaging	Slide_DCx_001	Samples
Scripts (optional)	Code that turns statistics into plots and tests	Script_DCx_001	Scripts

x = your DC number in AGILE; 001 = a running number you count yourself, separately for each class.

2. Why unique identifiers, not self-given names

A short identifier points to a full description in the registry; a long descriptive name tries to be the description and fails. Tracking all data this way serves four aims:

- **Avoid data loss** — data can be found again years later, by anyone, which saves time and money.
- **Prevent fraud and errors** — every number in a figure can be traced back to its raw image and analysis.
- **Allow reuse** — images, statistics, workflows and models can be reused by other DCs of AGILE or other researchers instead of being produced again.
- **Enable publication** — journals increasingly require source images and source data in public repositories (BioImage Archive, BioStudies, Dryad, Zenodo), each with a DOI.

Data sharing in bioimaging is now within reach, but it depends on data being well documented from the start (Uhlmann et al. 2024).

What goes wrong with self-given names

Self-given name (avoid)	Problem
Root_H1-1RFP_H1-2GFP_3h1Crispr_ExpAug2024_3daysCIM_Glycerol_63x_5xAverage.TIF	Too long; information belongs in the registry, not the file name
Image 002_segmented_deconvolved_crop_trial_23.TIF	Processing history in the name; spaces; not unique across people
Test.JPG, Test_new.JPG	Says nothing; collides with everyone else's "Test"
Coty_November2023_epi_dark_cry2_5dag1hlight.lif	Every person encodes conditions differently; cannot be searched consistently

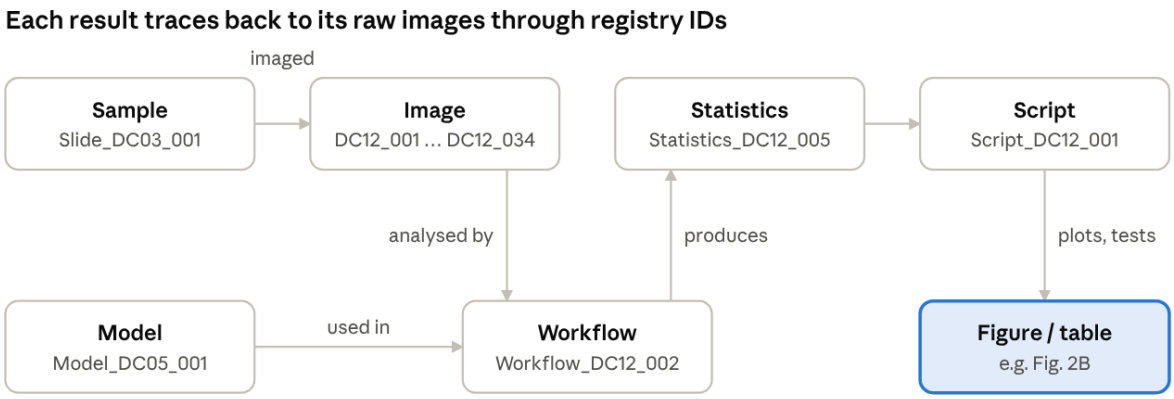
Long names cause concrete problems:

- Files may be rejected when uploading to a repository.
- Source data behind graphs become unreadable, e.g. a column of entries like Root_wt_Renaissance_confocal_January2023_nucleus26 next to each value.
- Dataframes with several descriptors per object, several objects per image and several channels become unmanageable.
- Image galleries become cumbersome.

Why DC numbers and not name initials

Initials are not unique across a network (two DCs can share them) and they say nothing about the network. The DC number is assigned once, never repeats within AGILE, and stays valid after the DC has left, so DC14_001 is unique across the whole consortium. Good: DC14_001, DC14_002 … DC14_1245. Parts of a split dataset: DC14_036-1, DC14_036-2 …

3. How the identifiers link



How AGILE records link · 6 data classes leading to a figure

Read the arrows backwards from a figure: the Script names the Statistics dataset it used, the dataset lists its images and the workflow, the workflow names its model, and each image names its slide. Numbers are not shared between classes; the connection is always an ID written in a cross-reference column.

4. Images

Rule: every image file or dataset gets an ID DCx_001 and one row in the Images tab of the registry.

Why?

- Journals ask for the source images behind quantifications and illustrations , deposited in a repository such as the BioImage Archive, IDR or Dryad.
- Repositories require each dataset to be tagged with dataset name, biosample (genotype, treatment), specimen, imaging method and image-processing method according to the REMBI guidelines. Completing the registry at acquisition facilitates the job at publication instead of days of reconstruction.
- Collaborations within AGILE: The registry lets you and other DCs select images by genotype, organ, treatment, labeling method or microscope without opening files.

What goes in?

- Images used for qualitative analysis, illustration or testing a setup.
- Images used for quantitative analyses
- Raw and processed images (used versioning -a, -b, -c or -1, -2, -3..)
- Images borrowed from other authors and re-used/re-processed

Important - Do not keep /Delete images that are uninformative or undocumented.

How?

1. **Proprietary files containing multiple images** (.lif, .czi, .nd2, .ims): microscope software often creates a folder, or "experiment" containing several images. Name the containing folder with a proper DC identifier, split images for quantification, name them as daughter images. For instance, if the DC14_023.lif file contains 10 images, they will be named DC14_023-1..DC14_023-10
2. **Single-image files**: alternatively, you can create one folder/experiment per 3D, or 4D image. It facilitates the downstream work as it saves versioning numbers and keep the identifier simple.
3. **Tip to** rename multiple images: using Bulk Rename Utility.
4. **Register** the ID in your Image registry template (see instructions in the xls file;msca-agile.eu/Resources/Data-Management)
5. **Link** the image to its Sample (Slide_DCx_…), the Workflow used to analyse it and the Statistics dataset that holds its measurements.
6. **Never reuse** an ID. A discarded image keeps its row with Deleted = y.

Recommended: register existing images retrospectively, so that older data can still be published and reused.

5. Statistics

Rule: every consolidated dataframe of image measurements gets an ID Statistics_DCx_001 and one row in the Statistics tab.

Why?

- A plot or a statistical test is only as trustworthy as the trail back to its raw images. Journals now ask for the source data behind each graph (e.g. as csv/xls in BioStudies).
- One dataframe usually pools the measurements of many images. Without an ID and a list of the images it contains, nobody can tell later which images were included or excluded.
- Short image IDs keep the dataframe readable: one column "Image ID" with DC12_017 instead of a 60-character file name repeated on thousands of rows.

How?

1. **Inside the dataframe**, every row (one measured object, and channel if relevant) carries the Image ID it was measured on, plus genotype, treatment where applicable, measured variables (intensity, volume, sphericity, distances …).
2. **Name the file** with its ID, e.g. Statistics_DC14_005.csv.
3. **Register** it in the Statistics tab of the AGILE data registry: the images included as ranges (e.g. DC14_001-034; DC14_056-064), the number of images, the Workflow ID(s) that produced the measurements, genotypes, treatments, objects and variables measured, date.
4. **Record exclusions** in the comments (e.g. DC15_004 excluded, low signal-to-noise), so the selection is transparent.
5. **Link forward**: list the Script ID(s) that uses the dataframe to make plots or tests.
6. **Store locally** first then repositories for long term - see storage strategy **section 10**

6. Workflows

Rule: every image-analysis pipeline, and every new version of it, gets an ID Workflow_DCx_001 and one row in the Workflows tab.

Why?

- The same images analysed with different parameters give different numbers. Without a recorded workflow, results cannot be reproduced, defended in review or compared between DCs.
- Undocumented processing is a major source of image mishandling in published papers (Miura & Nørrelykke 2021). Journals increasingly expect the image-analysis steps to be reported, ideally with code or parameter files (Schmied et al. 2024).
- A documented workflow can be reused by other DCs on new images, saving development time.

How?

1. **Name the aim** (e.g. measure H1.1 enrichment at the nuclear periphery) and the software: Imaris, Fiji, Python or other, with version.
2. **Describe each major step** with method and key parameters, or n.a. if not used: denoising, channel shift correction, field (flat-field) correction, deconvolution, nucleus segmentation, spot segmentation, other objects, measurements exported.
3. **Store the executable part** where it cannot be lost or silently changed and allow to follow versioning, such as Git Repository. Store Imaris creation parameters, Fiji macro, Python scripts etc; see also storage strategy **section 10**.
4. **Link** the Model ID(s) used in any step and the Image IDs the workflow was validated on.
5. **Version it**: a change in parameters that affects results = a new version or a new Workflow ID, never an overwrite.

7. Models

Rule: every trained machine-learning or AI model, and every re-trained version, gets an ID Model_DCx_001 and one row in the Models tab.

Why?

- A model is defined by its training data. Applied to images unlike those it was trained on (another microscope, tissue or voxel size), it can fail. The registry records its domain of validity.
- Results obtained with a model can only be reproduced with the exact same model file and version.
- Trained models are valuable shared assets in AGILE: one DC's nucleus-segmentation model can serve others, if it is findable and documented. The BioImage Model Zoo (Ouyang et al. 2022) shows how standardised model descriptions make models reusable across tools.

The MIFA guidelines set out what metadata and formats annotated training datasets need for reuse in AI (Zulueta-Coarasa et al. 2025); recording training images by ID in the registry is the first step towards that.

How?

1. **Describe the model**: task (nucleus or spot segmentation, denoising, pixel classification …), type (deep learning, random forest, fine-tuned generalist model), architecture or method (StarDist, Cellpose, Noise2Void, Imaris ML, ilastik …) and framework with version.
2. **Record the training data** as Image IDs or ranges, the number of images and annotated objects, and who annotated them how.
3. **Record validation**: the validation images, the metric (e.g. average precision at IoU 0.5) and its value.
4. **State input requirements**: channels, voxel size, 2D or 3D.
5. **Store the model file** at a stable location; use a Git Repository for versioning; share final models with a DOI (Zenodo) or on the BioImage Model Zoo, and list the Workflows that use it.
6. **Re-trained = new version**, never an overwrite.

8. Tracking Samples and Scripts (optional but highly recommended)

These two classes are optional in AGILE, but they close the chain from the plant to the figure.

Samples — Slide_DCx_001

Why? Image quality and interpretation depend on how the specimen was prepared (fixation, nuclei extraction, antibody and dilution, mounting medium). Recording it once per slide avoids repeating it for every image and lets you trace a problem back to a preparation batch.

How?

1. Give each slide or specimen an ID Slide_DCx_001 and label the physical slide or box with it.
2. Register it in the Samples tab: experiment type, dates, tissue and plant line, growth conditions, treatment, mounting, counterstain, targets with primary and secondary antibodies (code, batch, dilution, incubation).
3. Use the Slide ID in the Sample column of the Images tab, and list the Image IDs acquired from the slide.

Scripts — Script_DCx_001

Why? The step from a dataframe to a plot or a p-value involves choices (subsetting, outlier removal, normalisation, choice of test) that change the result. "For every result, keep track of how it was produced" and "always store raw data behind plots" are two of the ten rules for reproducible computational research (Sandve et al. 2013).

How?

1. Give each script or notebook an ID Script_DCx_001 and put it in the file name or header.
2. Register it in the Scripts tab: software (R, Python, Matlab, Julia, other) with key packages, and each step used: subsetting, QC, normalisation, aggregation, statistical test, plot.

3. List the Statistics ID(s) it was run on and the output it produced (figure panel, supplementary table).
4. Keep scripts under version control (use a Git repository) and give the commit or a Zenodo DOI.

9. Golden rules

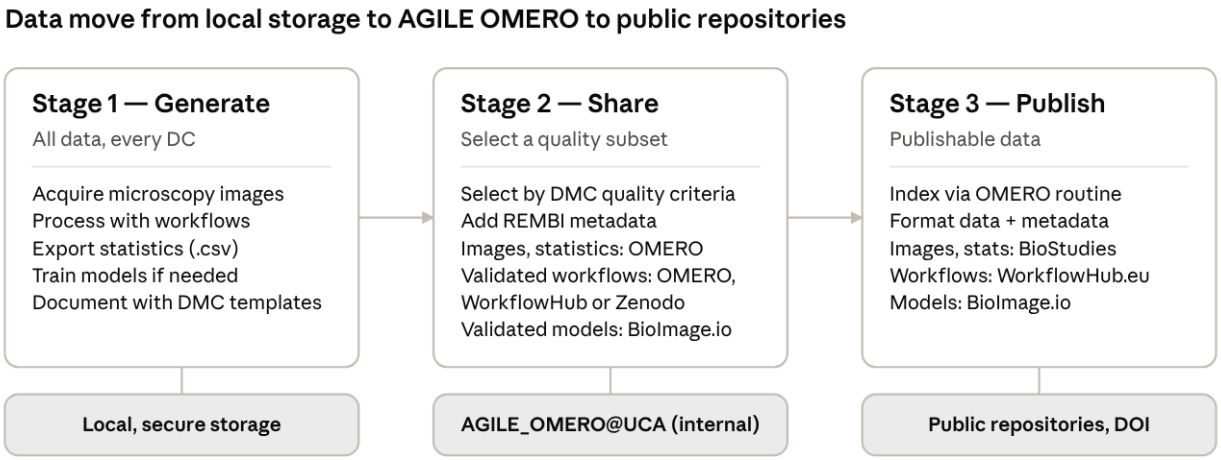
1. **Plan before you start**: experimental design and replicates, imaging platform, analysis workflow and software, which data and how they will be normalised.
2. **One ID per record**, built on your DC number, never on name initials or descriptive names.
3. **Register at creation**, not at the end of the project: the row is written when the image is acquired, the dataframe consolidated, the workflow fixed or the model trained.
4. **Never reuse or recycle an ID**; mark discarded records as Deleted = y.
5. **Fill the cross-reference columns** (Sample, Image IDs, Workflow, Model, Statistics, Script IDs): they are what links a figure back to its raw data.
6. **New version = new record** for workflows, models and scripts.
7. **Store files under their ID** at a shared, backed-up location, and give that location in the registry.
8. **Use the drop-down terms** of the registry; add missing terms to the Lists tab so that everyone uses the same wording.

Where each class lives

Class	During the project	For publication
Samples	Registry (Samples tab)	Methods, reagent list, protocol (e.g. STAR Protocols)
Images	Registry (Images tab), file server or the AGILE OMERO server at UCA (internal)	Image repository: BioImage Archive, IDR, Dryad
Workflows and Models	Registry, parameter files, Git repository	Methods, supplements, WorkflowHub, Zenodo DOI, BioImage Model Zoo
Statistics and Scripts	Registry, csv files, Git repository (follow versioning)	Source data (e.g. BioStudies), Zenodo, supplements, Git repository with DOI

10. Where data are stored, and when: the three stages

AGILE data pass through three stages, defined in the AGILE Data Management Plan: all data are generated and documented locally (Stage 1), a quality-selected subset is shared within AGILE on the OMERO server at UCA (Stage 2), and publishable data are consolidated and deposited in public repositories (Stage 3). The identifiers and registry described in this document are the thread that runs through all three.



AGILE data handling · 3 stages, from the AGILE Data Management Plan

Who does what

The Data Management Committee (DMC: C. Baroux, C. Tatout, J.-M. Burel, F. Pontvianne) provides guidelines, templates and infrastructure; the PIs and DCs implement them with DMC support.

Stage	The DMC provides	PIs and DCs are responsible for
1 — Generate	Guidelines for image acquisition; templates for tracking images, statistics, workflows and models (this registry), available at www.msca-agile.eu	Acquiring, processing and documenting data with the templates; storing Stage 1 data securely
2 — Select & share	"Stage 2 Data Selection Guidelines" with quality criteria; the AGILE_OMERO@UCA server; tutorials on using OMERO and annotating images with REMBI metadata	Selecting, annotating and uploading the selected datasets
3 — Publish	Guidelines for depositing publishable Stage 2 data in public repositories; a routine that extracts data and metadata from the OMERO server	Selecting, annotating and uploading datasets to public repositories

Templates can be adapted to your own needs, but changes must be validated by the DMC. Following the Stage 2 guidelines is compulsory. Training webinars on the OMERO server are planned.

What happens to each data class

Class	Stage 1 — Generate	Stage 2 — Select & share	Stage 3 — Publish
Images (raw and processed)	Acquired, registered in the Images tab, stored locally and securely	Quality-selected, enriched with REMBI metadata, uploaded to AGILE_OMERO@UCA	Indexed by the OMERO routine; deposited with metadata in BioStudies / BioImage Archive
Statistics (raw and processed)	Exported as dataframes, registered in the Statistics tab, stored locally	Selected with their images; csv files and links attached in AGILE_OMERO@UCA	Deposited as source data with metadata (BioStudies)
Workflows (image and data analysis)	Developed and documented in the Workflows / Scripts tabs, code in a Git repository	Validated, robust workflows uploaded to AGILE_OMERO@UCA and/or WorkflowHub or Zenodo	Published on WorkflowHub.eu with a DOI
Models (untrained, pre-trained, re-trained)	Trained and documented in the Models tab, code in a Git repository	Validated models uploaded to BioImage.io	Published on BioImage.io (BioImage Model Zoo)

For short- and Mid-term, prefer Git repositories particularly for workflows, scripss, models to track versions. For long term, trefer WorkflowHub or Zenodo over GitHub alone for anything you share or publish: both give a DOI, GitHub does not.

11. References and resources

References

- Wilkinson MD et al. (2016) The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3: 160018.
- Sarkans U et al. (2021) REMBI: Recommended Metadata for Biological Images — enabling reuse of microscopy data in biology. *Nature Methods* 18: 1418–1422.
- Hartley M et al. (2022) The BioImage Archive – building a home for life-sciences microscopy data. *Journal of Molecular Biology* 434: 167505.
- Schmied C et al. (2024) Community-developed checklists for publishing images and image analyses. *Nature Methods* 21: 170–181.
- Miura K, Nørrelykke SF (2021) Reproducible image handling and analysis. *The EMBO Journal* 40: e105889.
- Ouyang W et al. (2022) BioImage Model Zoo: a community-driven resource for accessible deep learning in bioimage analysis. *bioRxiv* 2022.06.07.495102.
- Sandve GK et al. (2013) Ten simple rules for reproducible computational research. *PLoS Computational Biology* 9: e1003285.
- Uhlmann V, Hartley M, Moore J, Weisbart E, Zaritsky A (2024) Making the most of bioimaging data through interdisciplinary interactions. *Journal of Cell Science* 137: jcs262139.
- Zulueta-Coarasa T, …, Uhlmann V, …, Hartley M (2025) MIFA: Metadata, Incentives, Formats and Accessibility guidelines to improve the reuse of AI datasets for bioimage analysis. *Nature Methods*.

Repositories and tools

- BioImage Archive — free archive for published image data (EMBL-EBI).
- BioImage Model Zoo — sharing of trained models.
- Bulk Rename Utility — batch renaming of files.
- Image Data Resource (IDR) — curated, added-value database of published image datasets, linked to other biological data.
- Zenodo — general-purpose repository with DOI for statistics dataframes, scripts, workflows, parameter files and models.
- WorkflowHub — registry for computational and image-analysis workflows, with DOIs.
- **AGILE OMERO server at UCA** — internal to AGILE only; for storing, viewing and sharing images among consortium members during the project (not a public repository).