|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding by Artificial Intelligence  www.mpai.community |

|  |  |
| --- | --- |
| **Public document** | |
| **N156** | 2021/02/17 |
| **Source** | Audio-Events-Data group |
| **Title** | Draft MPAI-GSA Use Cases and Functional Requirements |
| **Target** | MPAI Members |

# Introduction

Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) is an [international association](http://mpai.community/) with the mission to develop *AI-enabled data coding standards*. Research has shown that data coding with AI-based technologies is *more efficient* than with existing technologies.

The MPAI approach to developing AI data coding standards is based on the definition of *standard interfaces* of *AI Modules (AIM).* AIMs operate on input data and provide output data both of which have a standard format. AIMs can be *combined* and *executed* in an MPAI-specified *AI-Framework* called MPAI-AIF. A [Call for MPAI-AIF Technologies](https://mpai.community/standards/mpai-aif/) [2] against functional requirements [1] is currently open.

While AIMs must expose standard interfaces to be able to operate in an MPAI AI Framework, their performance may differ depending on the technologies used to implement them. MPAI believes that *competing* developers striving to provide more performing *proprietary* and *interoperable* AIMs will promote *horizontal markets* of *AI solutions* that build on and further promote AI *innovation*.

The MPAI standardisation model is currently hard to implement because in many cases the data used do not have well-defined format or unambiguous semantics. This document lays down a plan to achieve the goal of achieving the desired standardisation. It does that by introducing four representative Use Cases that use AIMs to understand and compress the results of high-throughput experiments combining genomic/proteomic and other data - for instance from video, motion, location, weather, medical sensors are identified. These are used to derive AI Modules, their input/output types and the type of data format standardisation required to achieve the goal.

The Use Cases are

1. Integrative analysis of ‘omics datasets
2. Smart Farming
3. Genomics and phenotypic/spatial data
4. Genomics and behaviour

This document is to be read in conjunction with the MPAI-GSA Call for Technologies (CfT) [as it provides the functional requirements of all the technologies that have been identified as required to implement the current MPAI-GSA Use Cases. Respondents to the MPAI-GSA CfT should make sure that their responses are aligned with the functional requirements expressed in this document.

This document is structured in 7 chapters, including this Introduction.

|  |  |
| --- | --- |
| Chapter 2 | briefly introduces the AI Framework Reference Model and its six Components |
| Chapter 3 | briefly introduces the 4 Use Cases. |
| Chapter 4 | presents the 4 MPAI-CAE Use Cases with the following structure   1. Reference architecture 2. AI Modules 3. I/O data of AI Modules 4. Technologies and Functional Requirements |
| Chapter 5 | outlines a possible solution |
| Chapter 6 | gives suggested references |
| Chapter 7 | gives a basic list of relevant terms and their definition |

# The MPAI AI Framework (MPAI-AIF)

Most MPAI applications considered so far can be implemented as a set of AIMs – AI, ML and even traditional Data Processing (DP)-based units with standard interfaces assembled in suitable topologies to achieve the specific goal of an application and executed in an MPAI-defined AI Framework. MPAI is making all efforts to identify processing modules that are re-usable and upgradable without necessarily changing the inside logic. MPAI plans on completing the development of a 1st generation AI Framework called MPAI-AIF in July 2021.

The MPAI-AIF Architecture is given by

*Figure 1*.



*Figure 1 – The MPAI-AIF Architecture*

Where

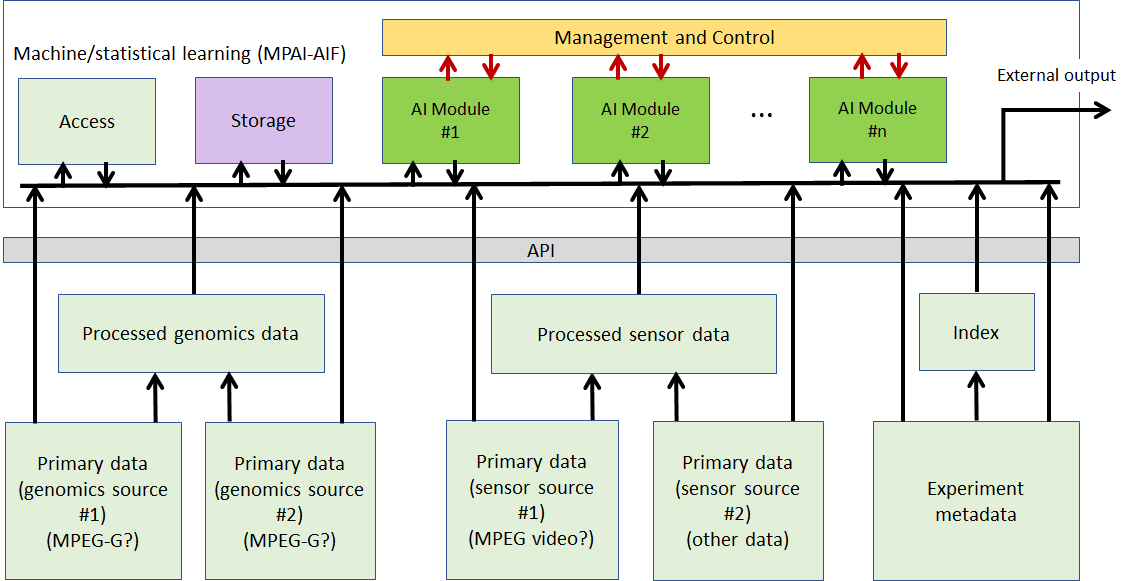
1. *Management and Control* manages and controls the AIMs, so that they execute in the correct order and at the time when they are needed.
2. *Execution* is the environment in which combinations of AIMs operate. It receives external inputs and produces the requested outputs both of which are application specific interfacing with Management and Control and with Communication, Storage and Access.
3. *AI Modules* (AIM) are the basic processing elements receiving processing specific inputs and producing processing specific outputs.
4. *Communication* is required in several cases and can be implemented, e.g., by means of a service bus and may be used to connect with remote parts of the framework
5. *Storage* encompasses traditional storage and is used to e.g., store the inputs and outputs of the individual AIMs, data from the AIM’s state and intermediary results, shared data among AIMs.
6. *Access* represents the access to static or slowly changing data that are required by the application such as domain knowledge data, data models, etc.

In MPAI-GSA, data can be of three types:

* *Primary*, i.e., the original unprocessed high-throughput content (such as DNA sequencing or video data)
* *Secondary*, i.e., the results of the pre-processing of primary data (such as gene expression estimates or features extracted from video) – applications will typically use these as input rather than primary data
* *Metadata* specifying additional information about the biological sample or experiment (such as sample content, cell types and barcodes, collection time and place).

The API provides uniform access to data; in particular, it standardises the definition of the semantics of the different data sources.

*Figure 2* is an alternative view of MPAI-AIF showing the different role of the 3 types of data.



*Figure 2 – The MPAI-AIF Architecture highlighting 3 data types*

The possibility of implementing genomic workflows integrated with different data sources whose processing concurs to achieving the desired result relies on the availability of standard and machine-actionable data formats.

# Use Cases

Integrative Genomic/Sensor Analysis uses AI to understand and compress the results of high-throughput experiments combining genomic/proteomic and other data - for instance from video, motion, location, weather, medical sensors. So far, the following application areas, ranging from personalised medicine to smart farming, have been considered.

## Integrative analysis of ‘omics datasets

In one possible realisation of this use case, one would like to correlate a list of genomic variants present in humans and having a known effect on health (metadata) with the variants present in a specific individual (secondary data). Such variants are derived from sequencing data for the individual (primary data) on which some variant calling workflow has been applied. Additional information derived from transcriptomics (RNA-sequencing, secondary data) might be taken into account. The list of variants could potentially be used to get to a personalised therapy.

Notably, there is an increasing number of companies doing just that as their core business. Their products differ by: the choice of the primary processing workflow (how to call variants from the sequencing data for the individual); the choice of the machine learning analysis (how to establish the clinical importance of the variants found); and the choice of metadata (which databases of variants with known clinical effect to use).

## Genomics and phenotypic/spatial data

As an example, we take single-cell RNA sequencing. The primary data sources is RNA-sequencing performed at the same time on a number (typically hundred of thousands) of different cells – while bulk RNA sequencing mixes together RNAs coming from several thousands of different cells, in single-cell RNA sequencing the RNAs coming from each different cell are separately barcoded, and hence distinguishable. The DNA barcodes for each cell would be metadata here. Cells can then be clustered together according to the expression patterns present in the secondary data (vectors of expression values for all the species of RNA present in the cell) and, if sufficient metadata and spatial information is present, clusters of expression patterns can be associated with different types/lineages of cells – the technique is typically used to study tissue differentiation. A number of complex algorithms exist to perform primary analysis (statistical uncertainty in single-cell RNA-sequencing is much bigger than in bulk RNA-sequencing) and, in particular, secondary AI-based clustering/analysis. Again, expressing those algorithms in terms of MPAI-GSA would make them much easier to describe and much more comparable. External commercial providers might provide researchers with clever modules to do all or part of the machine learning analysis.

## Genomics and behaviour

In a typical application of this use case, one would like to correlate animal behaviour (typically of lab mice) with their genetic profile (case of knock-down mice). Another application might be correlating genetic variants with the reaction to drug administration (typically encountered in neurobiology), possibly monitored in real-time with functional MRI scans. Hence primary data would be video data from cameras tracking the animal and/or data from an MRI scanner; secondary data would be processed video data in the form of primitives describing the animal’s movement, well-being, activity, weight, etc.; and metadata would be a description of the genetic background of the animal (for instance, the name of the gene which has been deactivated) or a timeline with the list and amount of drugs which have been administered to the animal. Again, there are several companies providing software tools to perform some or all of such analysis tasks – they might be easily reformulated in terms of MPAI-GSA applications.

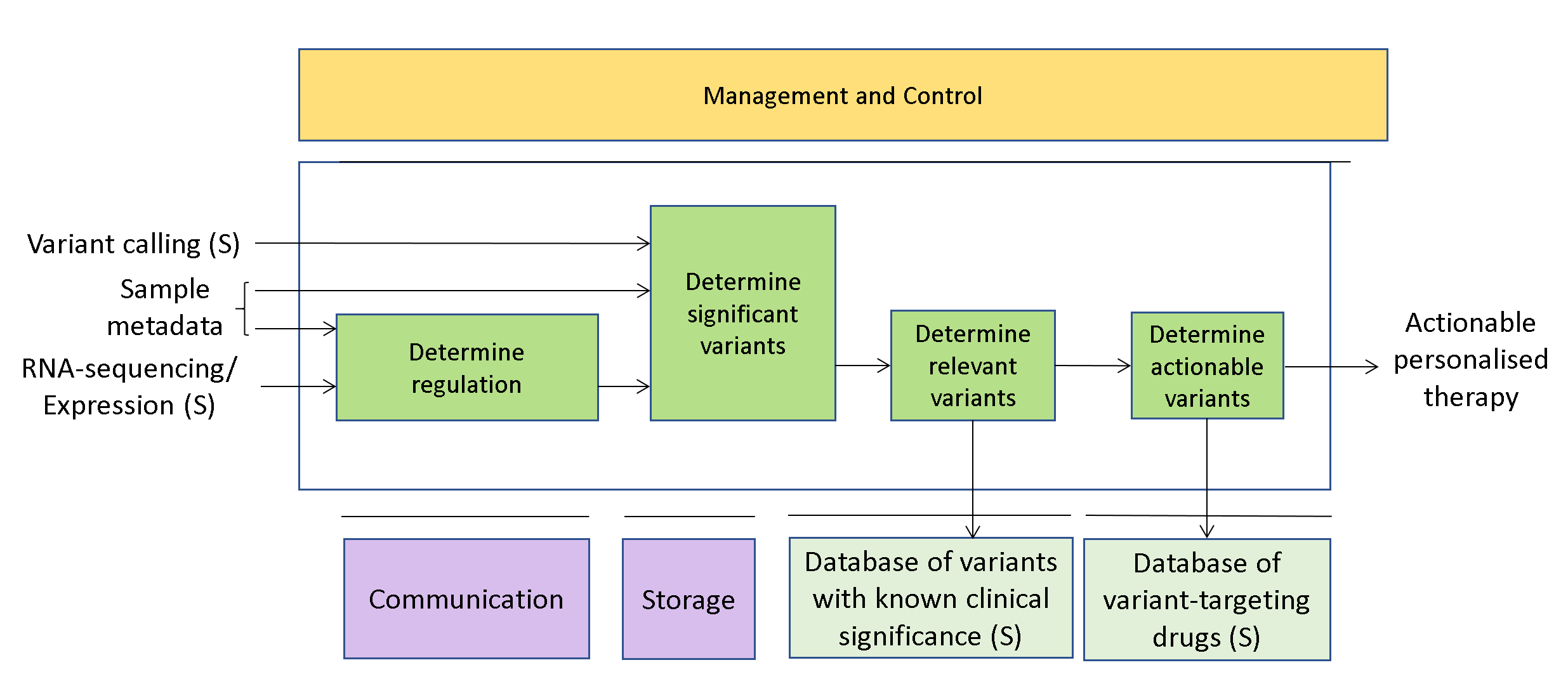
## Smart Farming

During the past few years, there has been an increasing interest in data-rich techniques to optimise livestock and crop production (so called “smart farming”). The range of techniques is constantly expanding, but the main ideas are to combine molecular techniques (mainly high-throughput sequencing and derived protocols, such as RNA-sequencing, ChIP-sequencing, HiC, etc.; and mass-spectrometry – as per the ‘omics case at point 2) and monitoring by images (growth rate under different conditions, sensor data, satellite-based imaging) for both livestock species and crops. So this use case can be seen as a combination of cases 2 and 4. Primary sources would be genomic data and images; secondary data would be vectors of values for a number of genomic tags and features (growth rate, weight, height) extracted from images; metadata would be information about environmental conditions, spatial position, etc. A growing number of companies are offering services in this area – again, having the possibility of deploying them as MPAI-GSA applications would open up a large arena where academic or commercial providers would be able to meet the needs of a number of customers in a well-defined way.

# Functional Requirements

## Integrative analysis of ‘omics datasets

### Reference architecture



*Figure 3 – An example of Integrative analysis of ‘omics datasets*

### AI Modules

*Table 1 – AI Modules of* *Integrative analysis of ‘omics datasets*

|  |  |
| --- | --- |
| **AIM** | **Function** |
| Determine regulation | To determine the structure and quantitative details of gene regulation |
| Determine significant variants | To determine the sequence and placement of genomic variants that are significant to regulation |
| Determine relevant variants | To determine the sequence and placement of genomic variants that have known clinical significance |
| Determine actionable variants | To determine the sequence and placement of genomic variants that are known targets of existing drugs |

### I/O interfaces of AI Modules

*Table 2 – I/O data of AIMs*

|  |  |  |
| --- | --- | --- |
| **AIM** | **Input Data** | **Output Data** |
| Determine regulation | Sample metadata  RNA-sequencing (P)  Expression (S)  Genomic functional annotation | Regulation model  Genomic functional annotation |
| Determine significant variants | DNA-sequencing (P)  Genomic variants (S)  Sample metadata  Regulation model  Genomic functional annotation | Significant variants |
| Determine relevant variants | Significant variants  Variants with known clinical significance | Relevant variants |
| Determine actionable variants | Relevant variants  Variant-targeting drugs | Personalised therapy |

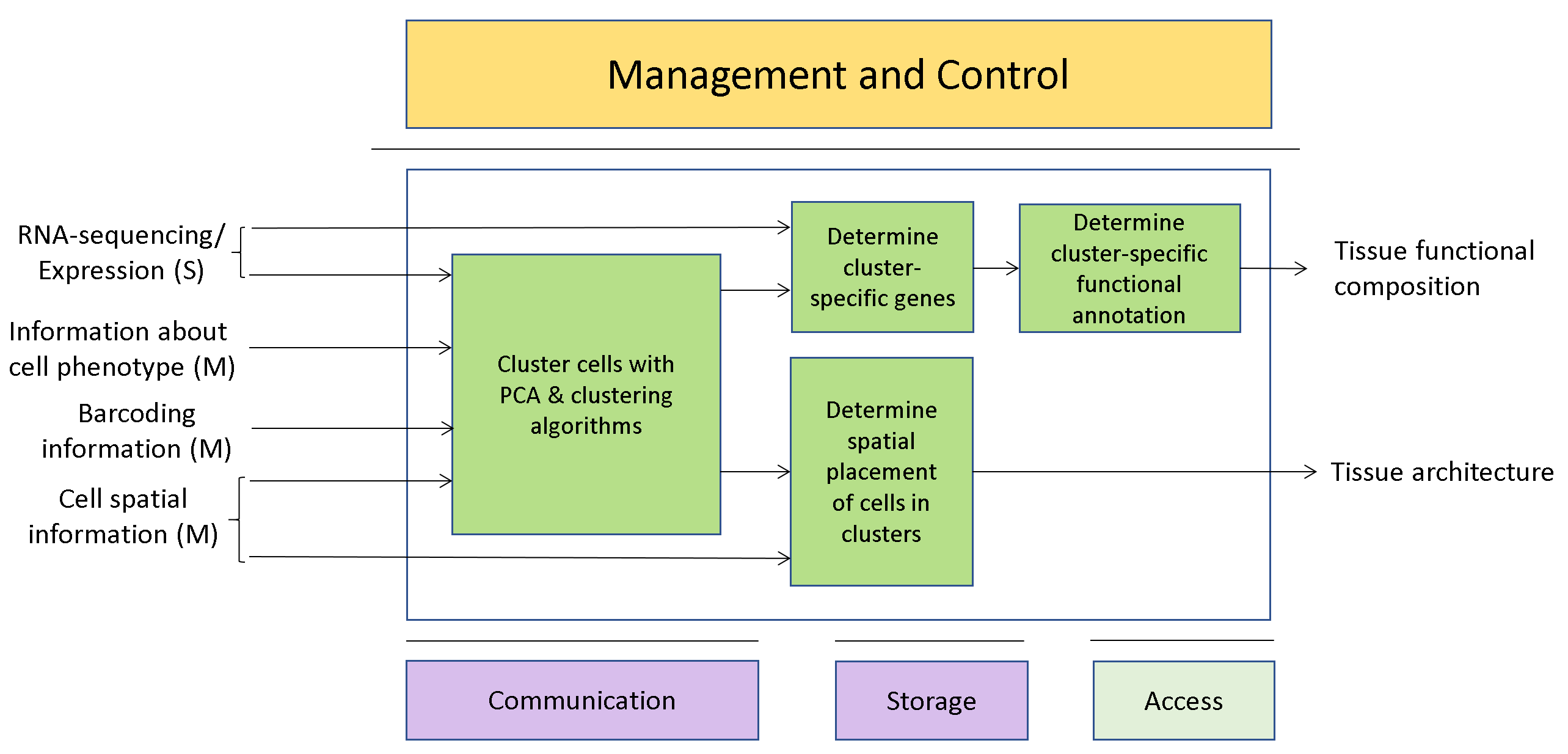
### Technologies and Functional Requirements

*Table 3 – Data types and formats*

|  |  |
| --- | --- |
| **Data type** | **Format** |
| DNA-sequencing (P) | FASTQ/SAM |
| Expression (S) | Tabular/Matrix |
| Genomic functional annotation | GTF/GFF |
| Genomic variants (S) | VCF |
| Personalised therapy | Tabular/JSON/Ontology |
| Regulation model | Tabular/JSON/Ontology |
| Relevant variants | VCF |
| RNA-sequencing (P) | FASTQ/SAM |
| Sample metadata | Tabular/JSON/Ontology |
| Significant variants | VCF |
| Variants with known clinical significance | VCF |
| Variant-targeting drugs | Tabular/JSON/Ontology |

## Genomics and phenotypic/spatial data

### Reference architecture



*Figure 4 – An example of Genomics and Phenotypic/spatial data*

### AI Modules

*Table 4 – AI Modules of* *Genomics and phenotypic/spatial data*

|  |  |
| --- | --- |
| **AIM** | **Function** |
| Cluster cells with PCA and clustering algorithms | To cluster cells based on their phenotype and RNA expres­sion profile |
| Determine cluster-specific genes | To determine expressed genes whose expression pattern characterises the cluster |
| Determine spatial placement of cells in clusters | To determine where the cells belonging to each cluster are placed in space |
| Determine cluster-specific functional annotation | To determine a functional annotation that takes into account genomic regulation for each cluster |

### I/O interfaces of AI Modules

*Table 5 – I/O data of Genomics and phenotypic/spatial data AIMs*

|  |  |  |
| --- | --- | --- |
| **AIM** | **Input Data** | **Output Data** |
| Cluster cells with PCA and clustering algorithms | Sample metadata  Cell phenotypes  Barcoding information  Cell spatial information  RNA-sequencing (P)  Expression (S)  Genomic functional annotation | Cell clustering |
| Determine cluster-specific genes | Sample metadata  RNA-sequencing (P)  Expression (S)  Cell clustering  Genomic functional annotation | Cluster-specific genes |
| Determine spatial placement of cells in clusters | Cell spatial information  Cell clustering | Tissue architecture |
| Determine cluster-specific functional annotation | Cluster-specific genes  Genomic functional annotation | Tissue functional composition |

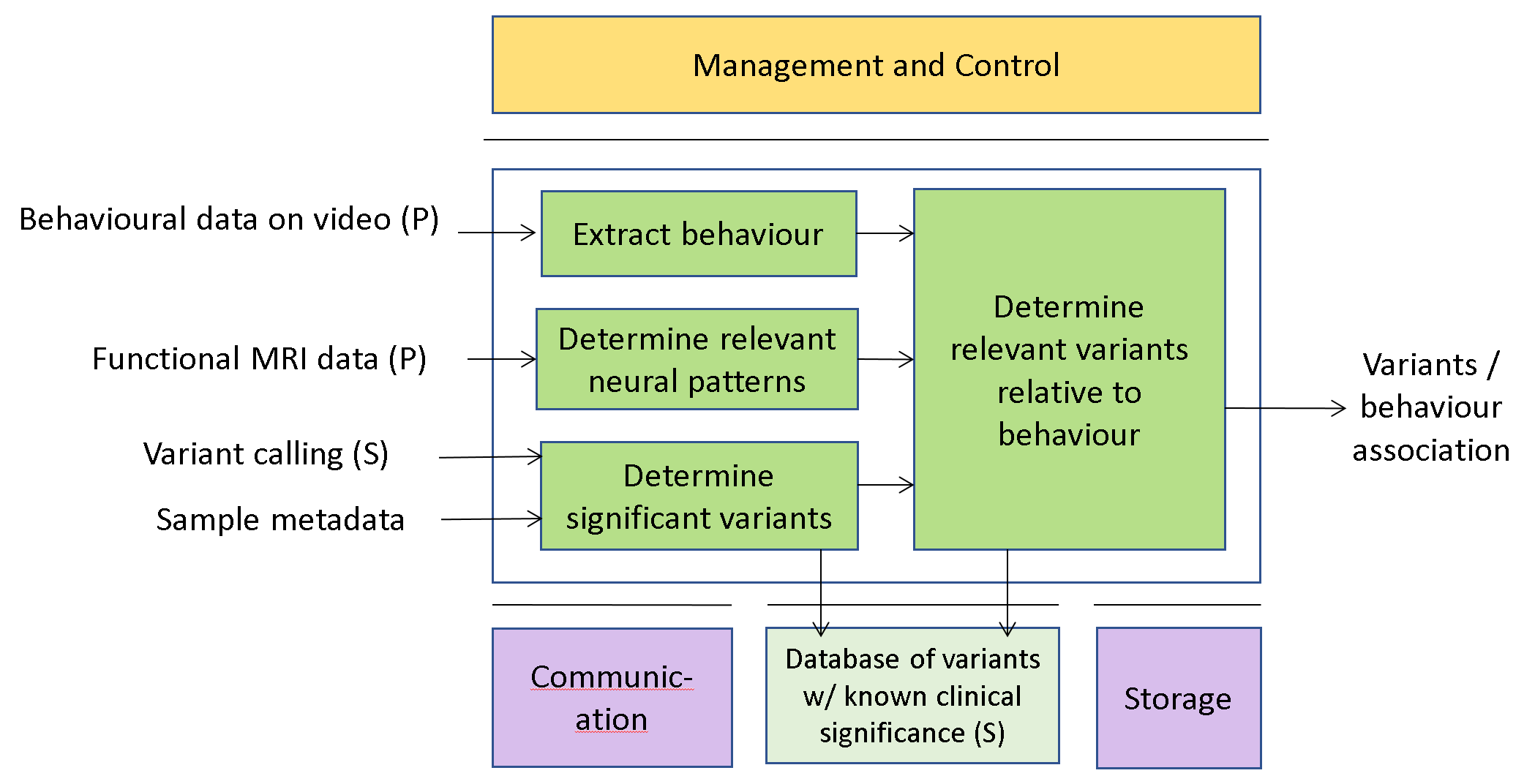
### Technologies and Functional Requirements

*Table 6 – Data types and formats*

|  |  |
| --- | --- |
| **Data type** | **Format** |
| Barcoding information | Tabular/JSON/Ontology |
| Cell clustering | Tabular/JSON/Ontology |
| Cell phenotypes | Tabular/JSON/Ontology |
| Cell spatial information | Tabular/JSON/Ontology |
| Cluster-specific genes | Tabular/JSON/Ontology |
| Expression (S) | Tabular/Matrix |
| Genomic functional annotation | GTF/GFF |
| RNA-sequencing (P) | FASTQ/SAM |
| Sample metadata | Tabular/JSON/Ontology |
| Tissue architecture | Tabular/JSON/Ontology |
| Tissue functional composition | Tabular/JSON/Ontology |

## Genomics and behaviour

### Reference architecture



*Figure 5 – An example of Genomics and Behaviour*

### AI Modules

*Table 7 – AI Modules of Genomics and behaviour*

|  |  |
| --- | --- |
| **AIM** | **Function** |
| Extract behaviour | To extract behavioural patterns (actions, ROIs, time series, repetition) out of video data |
| Determine relevant neural patters | To determine neural pattern which are relevant to training activities/responses to drugs |
| Determine significant variants | To determine the sequence and placement of genomic variants that have known clinical significance |
| Determine clinically relevant variants relative to behaviour | To determine the sequence and placement of genomic variants that have known clinical significance and are likely to be associated with behavioural traits |

### I/O interfaces of AI Modules

*Table 8 – I/O data of Genomics and behaviour AIMs*

|  |  |  |
| --- | --- | --- |
| **AIM** | **Input Data** | **Output Data** |
| Extract behaviour | Behavioural data recorded on video (P) | Behavioural patterns extracted from video (S) |
| Determine relevant neural patters | Functional MRI data (P) | Topology of neural patterns (S) |
| Determine significant variants | DNA-sequencing (P)  Genomic variants (S)  Sample metadata  Variants with known clinical significance | Significant variants |
| Determine clinically relevant variants relative to behaviour | Behavioural patterns extracted from video (S)  Topology of neural patterns (S)  Significant variants  Variants with known clinical significance | Clinically relevant variants correlated with behaviour |

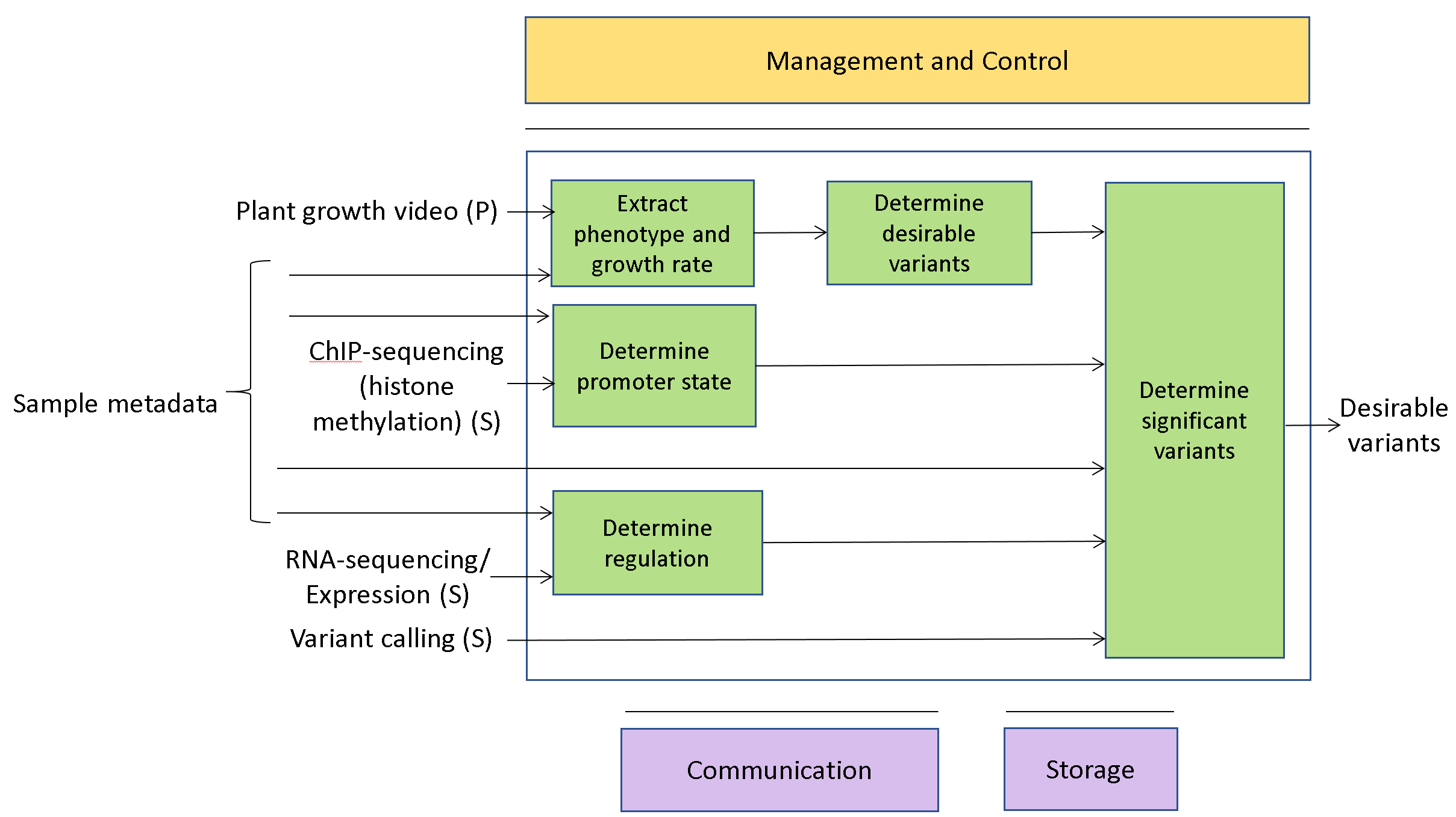
### Technologies and Functional Requirements

*Table 9 – Data types and formats*

|  |  |
| --- | --- |
| **Data type** | **Format** |
| Behavioural data recorded on video (P) | MPEG |
| Behavioural patterns extracted from video (S) | MPEG “subtitles”  Tabular/JSON/Ontology |
| Clinically relevant variants correlated with behaviour | VCF  Tabular/JSON/Ontology |
| DNA-sequencing (P) | FASTQ/SAM |
| Functional MRI data (P) | MRI-like formats |
| Genomic variants (S) | VCF |
| Sample metadata | Tabular/JSON/Ontology |
| Significant variants | VCF |
| Topology of neural patterns (S) | MPEG “subtitles”  Tabular/JSON/Ontology |
| Variants with known clinical significance | VCF |

## Smart Farming

### Reference architecture



*Figure 6 – An example of Smart Farming*

### AI Modules

*Table 10 – AI Modules of Smart Farming*

|  |  |
| --- | --- |
| **AIM** | **Function** |
| Extract phenotype and growth rate | To extract plant phenotype and growth rate from video data |
| Determine promoter state | To determine the state of gene promoters in relation to gene regul­ation |
| Determine regulation | To determine the structure and quantitative details of gene regulation |
| Determine desirable variants | To determine the sequence and placement of genomic variants that are correlated with a higher growth rate |
| Determine significant variants | To determine the sequence and placement of genomic variants that confer desirable traits |

### I/O interfaces of AI Modules

*Table 11 – I/O data of Smart Farming AIMs*

|  |  |  |
| --- | --- | --- |
| **AIM** | **Input Data** | **Output Data** |
| Extract phenotype and growth rate | Video of growing plants  Sample metadata | Phenotype and growth rate |
| Determine promoter state | ChIP-seq of histone methylation  Sample metadata | Promoters state |
| Determine regulation | RNA-sequencing (P)  Expression (S)  Sample metadata  Genomic functional annotation | Genomic regulation |
| Determine desirable variants | DNA-sequencing (P)  Genomic variants (S)  Sample metadata | Variants that are correlated with desirable phenotypes |
| Determine significant variants | DNA-sequencing (P)  Genomic variants (S)  Sample metadata  Variants that are correlated with desirable phenotypes  Promoters state  Genomic regulation | Variants that confer desirable phenotypes |

### Technologies and Functional Requirements

*Table 12 – Data types and formats*

|  |  |
| --- | --- |
| **Data type** | **Format** |
| ChIP-seq of histone methylation | FASTQ/SAM |
| DNA-sequencing (P) | FASTQ/SAM |
| Expression (S) | Tabular/Matrix |
| Genomic functional annotation | GFF/GTF |
| Genomic regulation | Tabular/JSON/Ontology |
| Genomic variants (S) | VCF |
| Phenotype and growth rate | MPEG “subtitles” |
| Promoters state | Tabular/JSON/Ontology |
| RNA-sequencing (P) | FASTQ/SAM |
| Sample metadata | Tabular/JSON/Ontology |
| Variants that are correlated with desirable phenotypes | VCF |
| Variants that confer desirable phenotypes | VCF  Tabular/JSON/Ontology |
| Video of growing plants | MPEG |

# Data formats

Broadly speaking, the data formats identified by the Use Cases fall under three categories:

1. Genomic/sequencing/proteomic data
2. Video/audio/sensor data
3. Metadata and other data that is weakly structured. Examples would be drugs databases, pathway/metabolic/growth models, behavioural annotations, information about samples and experiments, and the I/O of secondary analysis themselves. Such information is often presented in tabular format, but without a defined way of associating the rows/columns with their semantics (see, e.g., differential regulation for RNA-sequencing experiments).

In the following sections such categories are analysed in more detail in order to identify solutions that are available and others that require additional investigation.

## Genomic/sequencing/proteomic data

|  |  |  |
| --- | --- | --- |
| **Data type** | **Format** | **Identified solution** |
| Sequencing reads | FASTA | MPEG-G parts 1/2 |
| Genomic references |
| Genomic assemblies |
| Sequencing reads | FASTQ |
| Aligned data | SAM |
| Genomic functional annotations | GFF/GTF | MPEG-G part 6 |
| Genomic variants | VCF |
| Genomic tracks | BigWig |
| Genomic assemblies | Graph formats |
| Genomic contacts | (Sparse) Matrix Formats |
| Expression data | Tabular |

### Genomic assemblies, assembly graphs

|  |  |
| --- | --- |
| **Data type** | Assembly graphs  Graph-like genome references |
| **Usage domain** |  |
| **Data semantics** | Express a string graph (set of sequences which are partially overlapping), such as NCBI’s ASN |
| **Requirements** | Ability to represent and query string graph, either standalone or as a combination of formats |
| **Action** | FASTA for the edges combined with a tabular representation of nodes  add all details |

### Proteomic/spatial proteomic data

|  |  |
| --- | --- |
| **Data type** |  |
| **Usage domain** | - |
| **Semantics** | - |
| **Requirements** | - |
| **Action** |  |

### Smart farming data

|  |  |
| --- | --- |
| **Data type** |  |
| **Usage domain** | - |
| **Semantics** | - |
| **Requirements** | - |
| **Action** |  |

## Video/audio/sensor data

### Metadata

|  |  |  |
| --- | --- | --- |
| **Data type** | **Format** | **Identified solution** |
| Experiment recording | Audio/Video Formats | MPEG video/audio formats. |
| Association between events and AV streams | Subtitle-like formats | MPEG video/audio file formats.  Common with MPAI-CAE? |

### Location/satellite

|  |  |
| --- | --- |
| **Data type** | Images with associated spatial metadata |
| **Usage domain** | Experiment recording |
| **Semantics** | Coordinates and elevations of points on curves (approximated with polygons) on the surface of the Earth and collection metadata |
| **Requirements** | Ability to represent the point where the experiment is carried out with an accuracy adequate to the application (which might vary – from lab to smart farming)  Ability to represent a polygon surrounding e.g., s plot |
| **Action** | GIS? |

### Location/satellite

|  |  |
| --- | --- |
| **Data type** | data sequences of positions of objects moving on the surface of the Earth |
| **Usage domain** | - |
| **Semantics** | Coordinates and elevations of trajectories of moving objects (e.g., animals) on the surface of the Earth |
| **Requirements** | Ability to record the trajectory of an object as a dunction of time and relevant metadata |
| **Action** | GeoJSON? |

### MRI-like data

|  |  |
| --- | --- |
| **Data type** | 3D or 4D images, together with experimental meta-data |
| **Usage domain** | Data from (functional-, connectomic) MRI experiments |
| **Semantics** | Density of voxels (e.g., amount of water, diffusion speed) |
| **Requirements** | Ability to represent voxel of spatial imaging information, possibly as a function of time |
| **Action** | Existing format for medical imaging (PACS?) for static voxels  ?? for dynamic voxels  call for metadata schema |

## Metadata/weakly structured data

### Metadata

|  |  |
| --- | --- |
| **Usage domain** | All use cases |
| **Semantics** | Metadata about the collection of experimental data |
| **Requirements** | Ability to describe:   * Sample * Collector * Collection data and place * Collection or generation experimental methodology * Generating experiment * Relations of the sample with its generating experiment (time series, hierar­chical sub-category) |
| **Possible solutions** |  |
| **Action** |  |

### Models (metabolic, behaviours)

|  |  |
| --- | --- |
| **Usage domain** | All use cases |
| **Semantics** | A model generated out of experimental data and describing relations between samples and/or other biological concepts |
| **Requirements** | Ability to describe:   * Scope of the model * Relations between the different components of the model (cluster, sets, graphs, conditions) * Relations between model components and time |
| **Possible solutions** |  |
| **Action** |  |

### Audio/video events

|  |  |
| --- | --- |
| **Usage domain** | Video/audio/sensor |
| **Semantics** | Describing features extracted from 2-3-4D video/audio/sensor data |
| **Requirements** | Ability to describe:   * The nature/ontology of the event * Spatial/temporal characteristics of the event (ROI, duration) * Placement of the event within 2-3-4D video/audio/sensor streams |
| **Possible solutions** |  |
| **Action** |  |

### Secondary inputs/Outputs of AIMs

|  |  |
| --- | --- |
| **Usage domain** | All use cases |
| **Semantics** | Describing secondary inputs, or outputs, of AIMs in terms of components and ontologies |
| **Requirements** | Ability to describe:   * The inputs/outputs in terms of their components (spatial/temporal dimensions, combination of channels) * The ontology of each component/channel.   Partially in common with MPAI-AIF? |
| **Possible solutions** |  |
| **Action** |  |

# Possible solution

All categories of data considered above can be represented as a tree-like data structure (which could be expressed in JSON-like format) combined with an ontology expressing the nature of the nodes of the tree. Interestingly, the typical data to be described could very well be represented with the data model employed by a number of non-SQL databases, such as MongoDB.

For instance, in the case of the outputs of an AIM expressing differential regulation estimated from an RNA-sequencing experiment and other data, the representation might be something like:

* For each time point:
  + Time
  + For each sample:
    - **For each feature in the payload:**
      * **Name**
      * **Unit of measurement**
      * **Ontology**
    - Sample name
    - Sample collection time
    - Sample collection place
    - More information about the sample (collector, etc.)
    - Category of sample
    - For each gene:
      * Estimated expression value
* For each couple of sample sets:
  + Set of samples 1
  + Set of samples 2
  + **For each result of the experiment:**
    - **Name**
    - **Unit of measurement**
    - **Ontology**
  + For each DR gene:
    - Estimated log-fold change
    - Estimated FDR/p-value for the fold-change

The meta-information about the data structure (in red) might be stored separately or embedded in the data structure itself. Given that information, it would be possible to query such data structures.

This suggests that defining an association between each example and an adequate meta-data schema might be sufficient to provide a satisfactory solution.

# Terms and definitions

|  |  |  |
| --- | --- | --- |
| **Term** | **Acr** | **Definition** |
| ‘omics |  |  |
| Access |  | Static or slowly changing data that are required by an application such as domain knowledge data, data models, etc. |
| AI Framework | AIF | The environment where AIM-based workflows are executed |
| AI Module | AIM | The basic processing elements receiving processing specific inputs and producing processing specific outputs |
| Abstract Syntax Notation | ASN |  |
| ChIP-sequencing |  |  |
| Cluster |  |  |
| Communication |  | The infrastructure that connects the Components of an AIF |
| Data Processing | DP | A legacy technology that may be used to implement AIMs |
| Delivery |  | An AIM that wraps data for transport |
| DNA sequencing |  |  |
|  | DR |  |
| Execution |  | The environment in which AIM workflows are executed. It receives external inputs and produces the requested outputs both of which are application specific |
| Expression |  |  |
| FASTA |  |  |
| FASTQ |  |  |
| FDR |  |  |
| Gene |  |  |
| Gene expression |  |  |
| GFF/GTF |  |  |
| HiC |  |  |
| JSON |  |  |
| Knowledge Base |  | Structured and unstructured information made accessible to AIM (especially DP-based) |
| Management and Control |  | Manages and controls the AIMs in the AIF, so that they execute in the correct order and at the time when they are needed |
| Mass-spectrometry |  |  |
| Metadata |  |  |
| Magnetic resonance imaging | MRI |  |
| NCBI |  |  |
| PACS |  |  |
| Principal Component Analysis | PCA |  |
| Phenotype |  |  |
| Primary data |  | The original unprocessed high-throughput content |
| Regulation |  |  |
| RNA sequencing |  |  |
| Region of interest | ROI |  |
| Sequence Alignment Map | SAM |  |
| Secondary data |  | The results of the pre-processing of primary data |
| Sequencing |  |  |
| Structured Query Language | SQL |  |
| Storage |  | Storage used to e.g., store the inputs and outputs of the individual AIMs, data from the AIM’s state and intermediary results, shared data among AIMs |
| Transcriptomics |  |  |
| Variant |  |  |
| Variant Call Format | VCF |  |
| Workflow |  |  |

# References

1. MPAI-AIF Use Cases and Functional Requirements
2. MPAI-AIF Call for Technologies