

# Declarative, YAML-Based Workflows for Reproducible and Scalable Microbiome Analysis in the mia Ecosystem

Dattatray S. Mongad  
Postdoctoral Researcher (SYS-LIFE),  
Department of Computing, University of Turku, Finland



**UNIVERSITY  
OF TURKU**



**Funded by  
the European Union**

# Motivation

- Analysis workflows often start as scripts or notebooks. Over time, they become difficult to track, reproduce, and share.

## Initial days

One dataset  
One script  
Few parameters  
Few figures  
Clear report



## Few months later...

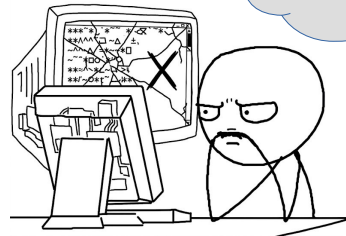
New metadata  
Thresholds changed  
Several model versions  
Some steps rerun, some not  
Figures scattered  
Final outputs hard to trace



The  
metadata  
changed  
again

I only reran  
one step...  
I think

Wait... which  
parameters  
made this  
figure?



# What breaks when workflows keep evolving?



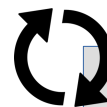
## Hidden parameter choices

Thresholds, transformations, and model settings are buried inside scripts.



## Scattered outputs

Tables, plots, reports, and objects spread across folders.



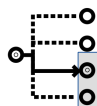
## Unnecessary reruns

Steps are recomputed even when upstream inputs are unchanged.



## Unclear dependencies

It becomes hard to trace which step produced which result.



## Difficult variant comparison

Alternative filters, models, and datasets are hard to compare systematically.



## Local-to-HPC gap

Analysis code that works locally may not scale cleanly to parallel or HPC execution.

# Proposed idea: describe the workflow, then execute it

**YAML analysis plan**  
“What should run?”

**miaPlan engine**  
“How to translate?”

**targets pipeline**  
“What needs rerun?”

**output + report**  
“What should be produced?”

## YAML file

- analysis steps
  - input data
  - outputs
  - function arguments
- reports
- provenance

### Readable

Written as clear recipe

### Shareable

One YAML shared with collaborators.

### Less scripting

Define once, reuse across workflows

### Traceable

Parameters, steps, and outputs are recorded in one place.

### Reproducible

The same plan can be rerun to regenerate results.

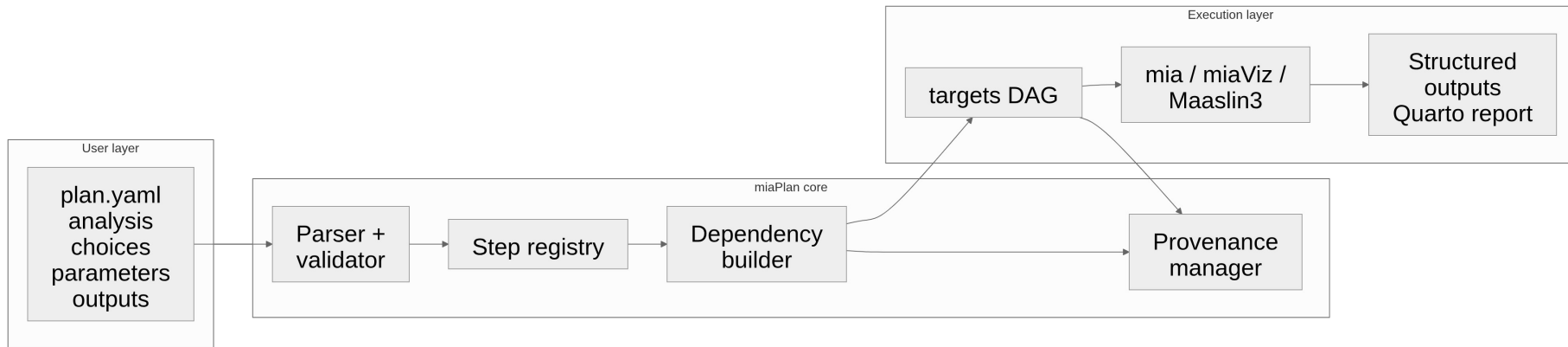
# What the YAML Plan Could Contain?

## plan.yaml

```
project: microbiome_xxx
input:
  object: data/tse.rds
steps:
  - filter_taxa
  - transform_clr
  - alpha_diversity
  - ordination
  - association_testing
outputs:
  tables: true
  plots: true
  report: quarto
function:
params:
```

- **Data source:** Where the microbiome object or raw data comes from.
- **Analysis decisions:** Filtering, transformations, diversity, ordination, and association steps.
- **Parameters:** Thresholds, methods, variables, models, and output options.
- **Step order and dependencies:** Which step should run before another.
- **Expected outputs:** Tables, plots, intermediate objects, and Quarto report.

# Proposed Architecture



## User layer

- `plan.yaml` defines analysis plan
- Optional Shiny interface to help users create/edit YAML
- Focus: readable, shareable workflow recipe

## miaPlan core

- Parse and validate the YAML
- Map step names to R functions
- Build dependencies between steps
- Track parameters and provenance

## Execution layer

- Compile steps into a {targets} pipeline and run
- Cache intermediate results and rerun only affected steps
- Save tables, plots, objects, and Quarto reports

10

Microbiome cohort analysis using  
TreeSummarizedExperiment  
Goal: diversity, ordination, and  
metadata association testing

*When one parameter changes, the  
workflow should know what needs to  
rerun and what can be reused.*



# Expected Benefits for the Community

## **Reproducibility**

Clear path from data to result

## **Collaboration**

Share the complete analysis plan

## **Scalability**

Rerun only affected steps

## **Standardization**

Use consistent workflow structure and outputs

### **For Researchers:**

- ✓ Easy reruns, clear reports and fewer lost analysis decisions

### **For bioinformaticians:**

- ✓ modular workflow design
- ✓ reusable templates
- ✓ cleaner debugging

### **For consortia/larger cohorts:**

- ✓ Standardized workflow
- ✓ Comparable analysis versions
- ✓ HPC-ready execution



# Development Plan

## Define the minimal YAML schema

- input data
- filtering
- transformation
- diversity analysis
- outputs

## Build the miaPlan core

- Read & validate  
YAML
- map steps to R  
functions
- build step  
dependencies

## Connect with {targets}

- caching
- incremental reruns
- dependency  
tracking
- parallel execution

## Outputs and reports

- intermediate .rds  
objects
- result tables
- plots
- Quarto report
- provenance record

## Future extensions

Shiny  
YAML  
builder

miaDash

Workflow  
templates

Plugin  
system

Container  
support

# Thank you!



Team Leo Lahti: <https://datascience.utu.fi/>

June 04, 2026

EuroBioC2026

## Brainstormers:

- Rasmus Hindström
- Nitin Bayal
- Giulio Benedetti
- Tuomas Borman



**UNIVERSITY  
OF TURKU**



**Funded by  
the European Union**



10 / 10