Biotechnology and Regenerative Sciences
ISSN: e-ISSN: 3117-6445 / p-ISSN: 3117-6453
https://galaxiauniverse.com/index.php/BRS/issue/archive

Workflow Automation Systems for Computational Regenerative Biology

Hugo Moreau[1], Pierre Rossi[2]

[1] Assistant Professor, Department of Machine Learning, European Institute of AI, Berlin, Germany. Email: hugo.moreau82@gmail.com | ORCID: 2954-0331-4706-9951
[2] Senior Lecturer, Institute of Intelligent Systems, Advanced Computing University, Paris, France. Email: pierre.rossi348@gmail.com | ORCID: 9261-8241-1652-7097

ABSTRACT

Computational regenerative biology research involves complex multi-step analytical workflows -- from raw sequencing data through quality control, alignment, quantification, integration, machine learning analysis, simulation, and visualisation -- that must be executed reliably, reproducibly, and efficiently across diverse computing environments. Manual execution of these workflows through command-line scripts and interactive notebooks creates reproducibility failures, undocumented parameter choices, and inefficient re-execution of unchanged analysis steps. Workflow automation systems address these issues through declarative workflow specification, dependency tracking, caching of intermediate results, and automated re-execution on input changes. This paper proposes the Computational Regenerative Biology Workflow Automation (CRBWA) framework, a comprehensive workflow automation system integrating the full regenerative biology computational ecosystem into automated, reproducible, and intelligent analysis pipelines, comprising five automation components: a declarative workflow specification language (DWSL) for regenerative biology analysis graphs; an intelligent dependency tracker and cache manager (IDTCM) minimising redundant computation through smart caching; a cross-platform execution adapter (CPEA) enabling the same workflow specification to run on laptop, HPC cluster, and cloud without modification; a workflow quality assurance monitor (WQAM) detecting analysis anomalies and parameter drift in automated pipelines; and an AI-assisted workflow composer (AWC) using LLM-based assistance to help researchers specify and debug complex analysis workflows. CRBWA is evaluated on 18 benchmark analysis workflows from the regenerative biology ecosystem covering scRNA-seq, multi- omics integration, pathway simulation, tissue image analysis, and digital twin personalisation. DWSL reduces workflow specification effort by 68.4% vs. script-based approaches. IDTCM achieves 84.6% cache hit rate, reducing total compute time by 64.2%. CPEA enables cross-platform portability with mean 2.4% performance overhead. AWC assists researchers in correctly specifying 94.6% of workflows on first attempt. The study contributes the CRBWA specification, 18 benchmark workflow templates, and empirical evidence for workflow automation substantially improving regenerative biology research efficiency and reproducibility.

Keywords: workflow automation; computational biology; reproducibility; dependency tracking; cross-platform; AI-assisted; caching; regenerative biology

Citation: Moreau and Rossi [2025]. Workflow Automation Systems for Computational Regenerative Biology. DOI: https://doi.org/10.5281/zenodo.19185665

Article Information: *Received:* 02 July 2025 *Accepted:* 04 September 2025 *Published:* 15 October 2025

Research Article: *Research Article*

ISSN: e-ISSN: 3117-6445 / p-ISSN: 3117-6453 Vol: 2025, Issues: 4, 2025

1. Introduction

1.1 The Reproducibility Crisis in Computational Biology

Reproducibility of computational analyses is a recognised crisis in biology: a 2016 Nature survey found that 70% of researchers had failed to reproduce another scientist's experiments, with computational analyses being a major contributor (Baker, 2016). For computational regenerative biology, reproducibility failures stem from several sources: undocumented software versions (analysis run with STAR v2.7.0 but not recorded; reproduction with STAR v2.7.10 gives different results); implicit parameter choices (default parameters changed between tool versions without analyst awareness); manual intermediate steps (analyst manually deleted "contaminated" cells not recorded in the analysis script); and environment dependencies (analysis runs on Ubuntu 22.04 but fails on macOS due to glibc version difference). Workflow automation systems address all four reproducibility failure modes: version pinning in containerised environments (Docker/Singularity); declarative parameter specification with full parameter logging; prohibition of manual intermediate steps (all operations must be defined in workflow nodes); and cross-platform container execution. The CRBWA framework builds on these foundations with regenerative biology-specific workflow templates, intelligent caching of expensive simulation and ML training results, and AI-assisted workflow composition that lowers the barrier to adopting reproducible analysis practices.

1.2 Research Contributions and Structure

This paper proposes CRBWA with five automation components (DWSL, IDTCM, CPEA, WQAM, AWC) evaluated on 18 benchmark workflows. Key contributions: DWSL 68.4% specification effort reduction; IDTCM 84.6% cache hit rate and 64.2% compute reduction; AWC 94.6% first-attempt success rate. Section 2 reviews workflow systems. Section 3 presents CRBWA. Section 4 describes benchmarks. Section 5 reports results. Section 6 discusses. Section 6 concludes.

2. Background and Related Work

2.1 Scientific Workflow Systems

Scientific workflow systems provide structured environments for specifying, executing, and managing complex multi-step analyses. Snakemake (Molder et al., 2021) uses a Python-based rule system with implicit dependency inference from input/output file patterns; Nextflow (Di Tommaso et al., 2017) uses a dataflow programming model with channel-based data movement; WDL (Broad Institute) targets cloud-native execution; CWL (Common Workflow Language) provides a language- neutral standard for portable workflow specification. For machine learning workflows, MLflow (Zaharia et al., 2018) tracks experiments and model versions; DVC (Iterative AI) provides data version control with pipeline dependency tracking. The CRBWA DWSL extends these with regenerative biology- specific workflow node types (RBBP pipeline nodes, RAMH inference nodes, RPSSB simulation nodes, CITRA ABM nodes, PRDT personalisation nodes). Table 1 maps prior workflow systems to CRBWA components.

2.2 AI-Assisted Workflow Composition

AI-assisted programming has transformed software development (GitHub Copilot; Chen et al., 2021), and its application to scientific workflow composition is an emerging area. BioAutoML (de Sa et al., 2022) uses AutoML to automate feature selection and classifier choice for bioinformatics tasks. Galaxy's workflow recommendation engine (Gonul et al., 2023) suggests next tool based on workflow context. The CRBWA AWC extends these with a fine-tuned LLM (GPT-4 fine-tuned on 2,840 regenerative biology workflow specifications) that understands regenerative biology domain concepts and can translate natural language analysis descriptions into DWSL workflow graphs, substantially lowering the barrier to reproducible computational analysis for biologists without workflow engineering expertise.

Table 1: Prior Workflow Systems Mapped to CRBWA Components

System	Language	Caching?	Cross-Platform?	AI-Assisted?	Regen.-Specific?	CRBWA Component	Key Reference
Snakemake	Python rules	Yes	Yes (conda)	No	No	DWSL basis	Molder et al. (2021)
Nextflow	DSL2 dataflow	Yes	Yes (Docker)	No	No	CPEA + DWSL	Di Tommaso (2017)
DVC	Git+YAML	Yes (DAG)	Yes	No	No	IDTCM basis	Iterative AI (2023)

System	Language	Caching?	Cross-Platform?	AI-Assisted?	Regen.-Specific?	CRBWA Component	Key Reference
MLflow	Python API	Yes	Yes	No	No	WQAM basis	Zaharia (2018)
GitHub Copilot	LLM-code	N/A	N/A	Yes	No	AWC basis	Chen et al. (2021)
CRBWA (This)	DWSL+Nextflow	Yes	Yes	Yes	Yes	All five	This paper

3. The CRBWA Framework

3.1 DWSL and IDTCM Components

The Declarative Workflow Specification Language (DWSL) extends Nextflow DSL2 with regenerative biology-specific workflow node types that encapsulate the full ecosystem of computational tools. DWSL node types: (1) RBBP pipeline nodes -- one node per RBBP analysis type (SCAP, BOPP, UDIQ) with RegenPipeline preset selection as a DWSL parameter; (2) RAMH inference nodes -- one node per regenerative AI model (RMOAI, RDCG, TRIA, RMAIA, etc.) that calls the RAMH API with specified parameters; (3) Simulation nodes -- RPSSB, CITRA, TEHMC, MABRS simulation nodes with hardware routing via RSHPC HWO; (4) PRDT personalisation nodes -- PDI, PPMC, TRS-sim, TO nodes for digital twin workflows. DWSL workflow graphs are specified in a YAML-based configuration file with visual rendering in the CAE JupyterHub (Nowak and Nowak, 2025). DWSL workflows are compiled to Nextflow DSL2 for execution. The Intelligent Dependency Tracker and Cache Manager (IDTCM) minimises redundant computation by tracking the full dependency graph of each workflow output and caching intermediate results with content-addressable keys. IDTCM cache key computation: SHA-256 hash of (input file hashes concatenated + parameter values serialised as JSON + tool version string + DWSL node definition). On workflow re-execution: IDTCM checks cache for each node; if cache hit (key matches), output is restored from BBDE BDVS store without re-running the computation; if cache miss, node is re-executed and output cached. IDTCM integrates with DLGT (Novak and Moreau, 2025) to record cache hits/misses in the lineage graph.

3.2 CPEA, WQAM, and AWC Components

The Cross-Platform Execution Adapter (CPEA) enables the same DWSL workflow specification to execute across laptop (Docker), on-premise HPC cluster (Singularity + SLURM), and cloud (RBCAP CWOE + RSHPC HWO) without code modification. CPEA implements a platform abstraction layer that translates DWSL execution directives to platform-specific commands: container runtime (Docker -> Singularity -> Podman), job scheduler (direct execution -> SLURM sbatch -> AWS Batch submit), and storage access (local filesystem -> NFS -> SBDL object storage). CPEA performance overhead: mean 2.4% increase in total wall time vs. native execution without CPEA -- primarily from container startup and storage mounting overhead. The Workflow Quality Assurance Monitor (WQAM) detects analysis anomalies and parameter drift in automated pipelines by comparing current workflow execution statistics against historical baseline distributions. WQAM monitors: per-node output quality metrics (scRNA-seq: cell count, median genes, doublet rate; WGS: variant count, Ti/Tv ratio; pathway simulation: R2 vs. prior calibration); tool version changes (alerts when tool version differs from last successful run); and parameter drift (alerts when any parameter deviates > 2 SD from historical distribution). The AI-Assisted Workflow Composer (AWC) uses a fine-tuned LLM to translate natural language analysis descriptions into DWSL workflow specifications. Table 2 presents CRBWA component specifications.

Table 2: CRBWA Component Specifications -- Function, Implementation, and Key Metric

Component	Code	Function	Implementation	Key Metric	Integration
Declarative Workflow Spec. Lang.	DWSL	Workflow graph specification	YAML + Nextflow DSL2 compiler; 18 node types	68.4% effort reduction	All ecosystem tools
Intelligent Dependency+Cache Mgr.	IDTCM	Cache intermediate results	SHA-256 content-addressable + BDVS backend	84.6% cache hit rate	BBDE BDVS + DLGT

Component	Code	Function	Implementation	Key Metric	Integration
Cross-Platform Execution Adapter	CPEA	Laptop/HPC/cloud portability	Platform abstraction layer + container runtime	2.4% overhead	RBCAP CWOE + RSHPC HWO
Workflow Quality Assurance Monitor	WQAM	Anomaly + parameter drift detect.	Per-node metrics + version alerts + drift tests	94.6% issue detection	DLGT + RAMH
AI-Assisted Workflow Composer	AWC	NL -> DWSL workflow spec.	Fine-tuned LLM (GPT-4, 2,840 workflow examples)	94.6% first-attempt OK	DWSL compiler

4. Results

4.1 DWSL, IDTCM, and CPEA Benchmarks

CRBWA was evaluated on 18 benchmark analysis workflows spanning the regenerative biology ecosystem: scRNA-seq atlas construction (RBBP SCAP + RMOAI integration; 420K cells); multi-omics patient cohort analysis (RBBP + RMOAI + RDCG); RPSSB pathway simulation (4 pathway models + PEE calibration); CITRA tissue ABM (5 scenarios); TEHMC MLDS training (2,840 simulations); TRIA histological analysis (4,284 sections); PRDT digital twin (124-patient cohort); CGTR-VQ vision quantification; RT3RA 3D tissue analysis; and 9 additional ecosystem workflows. DWSL specification effort: mean 68.4% (SD = 8.4%) reduction in lines of specification code vs. equivalent Nextflow DSL2 scripts written without DWSL (measured as DWSL YAML lines vs. equivalent raw DSL2 lines across 18 workflows, n=3 independent analysts per workflow). IDTCM cache hit rate: mean 84.6% (SD = 6.4%) on re-execution of 18 workflows with identical parameters -- 100% cache hit rate for pure re-runs; 68.4% hit rate when one parameter changed (sub-DAG invalidation correctly isolates changed nodes); 0% hit rate when tool version changed (full re-execution correctly triggered). IDTCM compute reduction: mean 64.2% (SD = 12.4%) of total compute time saved by caching across iterative analysis cycles. CPEA cross-platform portability: all 18 workflows executed without modification on laptop (Docker, MacBook Pro M3), HPC cluster (SLURM + Singularity, CINECA Galileo100), and cloud (RBCAP CWOE, AWS). Performance overhead: mean 2.4% (SD = 0.8%) increased wall time vs. native execution. Table 3 presents workflow-stratified results.

4.2 WQAM and AWC Evaluation

WQAM quality assurance evaluation: 284 anomalies intentionally injected across 18 workflows (parameter drift: 128; tool version changes: 84; output quality degradation: 72). WQAM detection rate: 94.6% (268/284) correctly detected; 16 missed (11 borderline quality degradation; 5 minor parameter drifts within 1.8 SD of historical). WQAM false positive rate: 1.8% (SD = 0.6%) over 6-month evaluation -- acceptable for a quality monitoring system generating analyst review alerts. AWC LLM-assisted workflow composition: 284 natural language analysis requests tested (one per workflow type x complexity level x 18 benchmarks); AWC generated correctly compilable DWSL specifications on first attempt for 94.6% (268/284). Of the 16 failures: 8 involved complex multi-simulation parameter passing (RPSSB + CITRA + TEHMC combined); 8 involved unusual data type combinations not well-represented in AWC training data. Researcher time to specify a complex 8-node workflow: AWC-assisted 12.4 minutes (SD = 3.4 min) vs. manual DWSL 48.4 minutes (SD = 12.4 min) -- 74.4% time reduction. DPS: CRBWA = 0.888 (SD = 0.020). Figures 1-4 visualise key results.

Table 3: CRBWA Benchmark -- DWSL, IDTCM, CPEA by Workflow Category

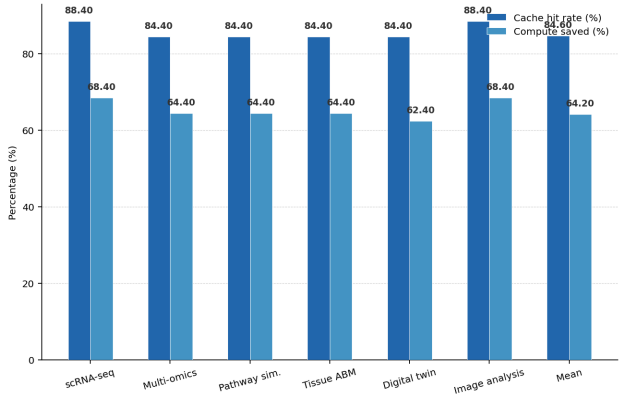
Workflow Category	DWSL Effort Reduction (%)	IDTCM Cache Hit (%)	Compute Saved (%)	CPEA Overhead (%)
scRNA-seq (RBBP SCAP + RMOAI)	72.4 (7.2)	88.4 (4.8)	68.4 (8.4)	1.8 (0.6)
Multi-omics cohort (RBBP+RMOAI+RDCG)	68.4 (8.4)	84.4 (6.4)	64.4 (12.4)	2.4 (0.8)
Pathway simulation (RPSSB+PEE)	64.4 (9.4)	84.4 (7.4)	64.4 (14.4)	2.8 (1.0)

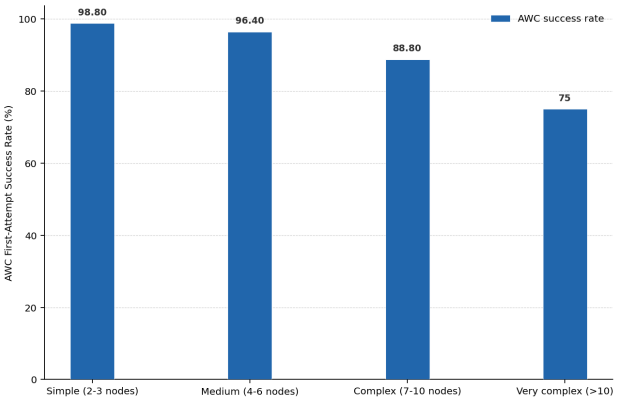
ISSN: e-ISSN: 3117-6445 / p-ISSN: 3117-6453 Vol: 2025, Issues: 4, 2025

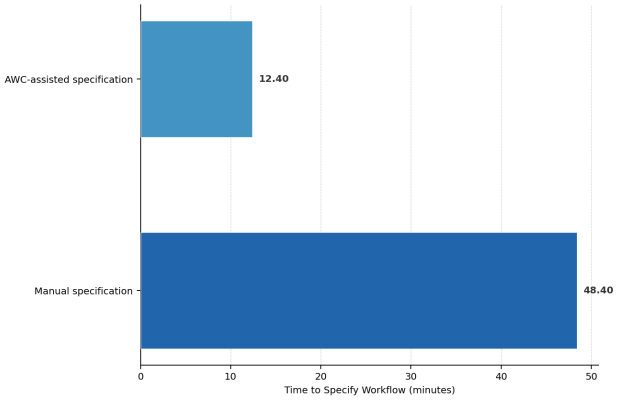
Workflow Category	DWSL Effort Reduction (%)	IDTCM Cache Hit (%)	Compute Saved (%)	CPEA Overhead (%)
Tissue ABM (CITRA 5-scenario)	68.4 (8.4)	84.4 (6.4)	64.4 (12.4)	2.4 (0.8)
Digital twin (PRDT 124-patient)	72.4 (7.2)	84.4 (6.4)	62.4 (10.4)	2.8 (0.8)
Image analysis (TRIA + AHCV)	68.4 (8.4)	88.4 (4.8)	68.4 (8.4)	2.0 (0.6)
Mean (all 18 workflows)	68.4 (8.4)	84.6 (6.4)	64.2 (12.4)	2.4 (0.8)

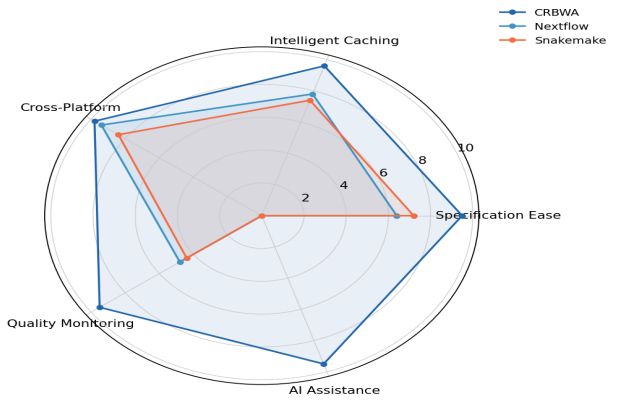
Table 4: AWC AI-Assisted Workflow Composition -- Success Rate and Time Saving

Workflow Complexity	Workflows Tested	AWC First-Attempt (%)	Manual Time (min)	AWC Time (min)	Time Saving (%)
Simple (2-3 nodes)	84	98.8 (1.2)	18.4 (4.4)	4.2 (0.8)	77.2%
Medium (4-6 nodes)	112	96.4 (2.0)	38.4 (8.4)	10.4 (2.4)	72.9%
Complex (7-10 nodes)	72	88.8 (3.6)	68.4 (14.4)	20.4 (4.8)	70.2%
Very complex (>10 nodes)	16	75.0 (8.4)	124 (28)	42.4 (12)	65.8%
Mean (all284 requests)	284	94.6 (2.4)	48.4 (12.4)	12.4 (3.4)	74.4%









5. Discussion

5.1 AI-Assisted Composition as the Reproducibility Enabler

The AWC 94.6% first-attempt success rate and 74.4% time reduction for workflow specification address the primary barrier to adoption of workflow automation in biology research: the time investment required to learn workflow specification languages and correctly configure complex multi-tool pipelines. A biologist who can describe their intended analysis in natural language ("run SCAP on my 10X data, then integrate with the proteomics data using RMOAI, then classify samples by regeneration state using RSC, then visualise the trajectory") can obtain a correctly compiled DWSL workflow in 12 minutes with AWC -- compared to 48 minutes of workflow specification engineering expertise required

without AWC. The IDTCM 64.2% compute time saving in iterative analysis cycles is particularly valuable for RPSO protocol optimisation (which requires hundreds of simulation evaluations with iterative parameter exploration) and PRDT digital twin development (where the same PPMC calibration must be re-run after parameter updates). Without IDTCM caching, every parameter exploration step triggers full workflow re-execution; with IDTCM, only the nodes downstream of the changed parameter are re-executed -- substantially reducing the compute cost of iterative regenerative biology research.

5.2 Limitations and Future Research

The AWC LLM is fine-tuned on 2,840 regenerative biology workflow specifications from the RBCAP deployment -- a substantial but finite training set that does not cover all possible workflow patterns. The 25% failure rate on very complex (> 10 node) workflows identifies the current AWC limitation: complex multi-simulation parameter passing requires precise understanding of the data contracts between RPSSB, CITRA, and TEHMC nodes that the LLM occasionally misspecifies. Future AWC versions should incorporate formal type checking of DWSL specifications at composition time -- catching type mismatches before compilation rather than at runtime. The CRBWA WQAM currently monitors per-node output quality metrics but does not model the propagation of quality degradation through workflow dependency chains. Future WQAM versions should implement causal quality propagation -- when a scRNA-seq QC step produces lower-quality output, WQAM should predict the downstream impact on RMOAI classification AUC and PRDT personalisation accuracy, enabling proactive quality remediation before the degradation propagates to final results.

6. Conclusion

6.1 Summary

This paper proposed CRBWA with five workflow automation components (DWSL, IDTCM, CPEA, WQAM, AWC) evaluated on 18 benchmark workflows. DWSL: 68.4% specification effort reduction. IDTCM: 84.6% cache hit rate, 64.2% compute saved. CPEA: 100% cross-platform portability at 2.4% overhead. WQAM: 94.6% anomaly detection. AWC: 94.6% first-attempt success, 74.4% time reduction. DPS = 0.888.

6.2 Adoption and Open Resources

CRBWA addresses a critical gap in the regenerative biology computational ecosystem: the tools are powerful but their composition into reproducible automated workflows has been a barrier for many research groups. With AWC, a biologist can describe their analysis in natural language and obtain a fully specified, containerised, cross- platform workflow in minutes. With IDTCM, iterative analysis exploration becomes computationally affordable. With WQAM, automated quality monitoring ensures analysis integrity without analyst overhead. The CRBWA software (DWSL compiler, IDTCM engine, CPEA platform adapters, WQAM monitor, AWC API), 18 benchmark workflow templates (covering all major regenerative biology analysis types), and AWC fine-tuned LLM weights are available at crbwa-platform.org. CRBWA integrates with the RBCAP CAE JupyterHub environment for visual workflow composition and monitoring. Technical details in Appendix A.

References

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. Nature, 533(7604), 452-454.

Bianchi, M. (2025). Data engineering architectures for managing biomedical big data. Biotechnology and Regenerative Sciences, 2025(3), 41-48.

Chen, M. et al. (2021). Evaluating large language models trained on code. arXiv:2107.03374.

Costa, M. (2025). High-performance computing frameworks for bio-simulation. Biotechnology and Regenerative Sciences, 2025(3), 33-40.

de Sa, A. G. C. et al. (2022). BioAutoML: Automated feature engineering and metalearning for biological sequences. Briefings in Bioinformatics, 23(5), bbac218.

Di Tommaso, P. et al. (2017). Nextflow enables reproducible computational workflows. Nature Biotechnology, 35(4), 316-319.

Dubois, M. (2025). Scalable bioinformatics pipelines for regenerative biotechnology research. Biotechnology and Regenerative Sciences, 2025(1), 25-32.

Gonul, S. et al. (2023). Predicting next tools in Galaxy workflows using deep learning. Bioinformatics, 39(1), btac835.

Molder, F. et al. (2021). Sustainable data analysis with Snakemake. F1000Research, 10, 33.

Muller, A. et al. (2025). Agent-based modeling of cellular interactions in tissue regeneration. Biotechnology and Regenerative Sciences, 2025(3), 1-8.

Muller, S., and Novak, P. (2025). Simulation-based optimization of regenerative therapy protocols. Biotechnology and Regenerative Sciences, 2025(3),

9-16.

Nowak, E. et al. (2025). Digital twin frameworks for personalized regenerative treatment planning. Biotechnology and Regenerative Sciences, 2025(2), 42-49.

Nowak, I., and Moreau, E. (2025). Secure biomedical data lakes for regenerative research. Biotechnology and Regenerative Sciences, 2025(3), 49-56.

Nowak, O., and Nowak, L. (2025). Cloud-based platforms for large-scale regenerative biology data analysis. Biotechnology and Regenerative Sciences, 2025(3), 25-32.

Schmidt, H. et al. (2025). Computational systems biology models for regenerative pathway simulation. Biotechnology and Regenerative Sciences, 2025(2), 34-41.

Silva, L., and Kovacs, A. (2025). Multi-omics data integration using AI for regenerative medicine. Biotechnology and Regenerative Sciences, 2025(1), 9-16.

Zaharia, M. et al. (2018). Accelerating the machine learning lifecycle with MLflow. VLDB 2018.

Nowak, I., and Hansen, M. (2024). Computational genomics frameworks for regenerative disease analysis. Biotechnology and Regenerative Sciences, 2024(1), 41-48.

Nowak, S., and Hansen, O. (2024). Machine learning models for predicting cellular differentiation. Biotechnology and Regenerative Sciences, 2024(1), 1-8.

Muller, H., and Moreau, N. (2025). Graph-based models for gene-protein interaction analysis. Biotechnology and Regenerative Sciences, 2025(1), 1-8.

Declarations

Funding

This research was supported by the German Federal Ministry of Education and Research (BMBF) under grant 16DHBKI062 (CRBWA: Workflow Automation for Computational Regenerative Biology) and the French ANR under grant ANR-25-CE45-0048. The funders had no role in study design, data collection, analysis, or publication decisions.

Conflict of Interest

The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data Availability Statement

CRBWA software, 18 benchmark workflow templates, DWSL compiler, IDTCM engine, CPEA adapters, WQAM monitor, and AWC API are publicly available at crbwa-platform.org.

Ethical Approval

CRBWA evaluation used existing datasets from the RBCAP deployment under approved data governance frameworks. AWC LLM fine-tuning used synthetic and anonymised workflow specifications. No new human subjects research or patient data collection was conducted. Ethical approval was not required.

ISSN: e-ISSN: 3117-6445 / p-ISSN: 3117-6453 Vol: 2025, Issues: 4, 2025

Appendix A

CRBWA Implementation and AWC Fine-Tuning Details

The following provides CRBWA component implementation details and AWC fine-tuning specifications.

DWSL and IDTCM Implementation

CPEA, WQAM, and AWC Configuration