Adversarial Domain Adaptation Enables Knowledge Transfer Across Heterogeneous RNA-Seq Datasets

Oral Presentation

Kevin Dradjat [1], ***Massinissa Hamidi*** [1], ***Blaise Hanczar*** [1]

1. *Université d'Evry Paris-Saclay*

Abstract

Accurate phenotype prediction from RNA sequencing (RNA-seq) data is essential for diagnosis, biomarker discovery, and personalized medicine. Deep learning models have demonstrated strong potential to outperform classical machine learning approaches, but their performance relies on large, well-annotated datasets. In transcriptomics, such datasets are frequently limited, leading to over-fitting and poor generalization. Knowledge transfer from larger, more general datasets can alleviate this issue. However, transferring information across RNA-seq datasets remains challenging due to heterogeneous preprocessing pipelines and differences in target phenotypes.

In this study, we propose a deep learning-based domain adaptation framework that enables effective knowledge transfer from a large general dataset to a smaller one for cancer and tissue type classification. The method learns a domain-invariant latent space by jointly optimizing classification and domain alignment objectives. To ensure stable training and robustness in data-scarce scenarios, the framework is trained with an adversarial approach with appropriate regularization. Both supervised and unsupervised approach variants are explored, leveraging labeled or unlabeled target samples.

The framework is evaluated on three large-scale transcriptomic datasets (TCGA, ARCHS4, GTEx) to assess its ability to transfer knowledge across cohorts. Experimental results demonstrate consistent improvements in cancer and tissue type classification accuracy compared to non-adaptive baselines, particularly in low-data scenarios.

Overall, this work highlights domain adaptation as a powerful strategy for data-efficient knowledge transfer in transcriptomics, enabling robust phenotype prediction under constrained data conditions.

URL

https://doi.org/10.48550/arXiv.2603.08062

BeeRNA: tertiary structure-based RNA inverse folding using Artificial Bee Colony

Oral Presentation

Mehyar MLAWEH [1], ***Tristan Cazenave*** [1], ***Inès Alaya*** [1]

1. *LAMSADE, Université Paris Dauphine*

Abstract

The Ribonucleic Acid (RNA) inverse folding problem, designing nucleotide sequences that fold into specific tertiary structures, is a fundamental computational biology problem with important applications in synthetic biology and bioengineering. The design of complex three-dimensional RNA architectures remains computationally demanding and mostly unresolved, as most existing approaches focus on secondary structures. In order to address tertiary RNA inverse folding, we present BeeRNA, a bio-inspired method that employs the Artificial Bee Colony (ABC) optimization algorithm. Our approach combines base-pair distance filtering with RMSD-based structural assessment using RhoFold for structure prediction, resulting in a two-stage fitness evaluation strategy. To guarantee biologically plausible sequences with balanced GC content, the algorithm takes thermodynamic constraints and adaptive mutation rates into consideration. In this work, we focus primarily on short and medium-length RNAs (< 100 nucleotides), a biologically significant regime that includes microRNAs (miRNAs), aptamers, and ribozymes, where BeeRNA achieves high structural fidelity with practical CPU runtimes. The lightweight, training-free implementation will be publicly released for reproducibility, offering a promising bio-inspired approach for RNA design in therapeutics and biotechnology.

URL

https://arxiv.org/abs/2511.21781

Circulating DNA reveals nucleosome occupancy patterns that are associated with nucleosome-DNA affinity and are affected in cancer

Oral Presentation

Marianne RICHAUD [1], ***Ekaterina Pisareva*** [1], ***Alain Thierry*** [1], ***Jacques Colinge*** [1]

1. *Institut de recherche en cancérologie de Montpellier*

Abstract

The study of cell-free circulating DNA (cirDNA) fragments (fragmentomics) from liquid biopsies has received increasing attention. By constructing an atlas of these well-positioned nucleosomes, which we called WPNA, we found that their occupancy was associated with histone-DNA affinity, as evidenced by codon usage bias and differences in cirDNA fragment sizes. Moreover, WPNA nucleosome occupancy was different in healthy and cancer samples, thus allowing developing a high performance machine learning approach for cancer detection (specificity and sensitivity >0.95 for seven cancer types). Cancer influenced WPNA nucleosome occupancy in a global manner, although distinct cancer types retained specific features. WPNA nucleosome occupancy at transcription factor binding sites revealed shared, pan-cancer regulation of transcriptional programs involved in hematopoietic cell differentiation and neutrophil biology, the main cirDNA sources. This work provides new fundamental insights into cirDNA and DNA sequence using cirDNA as a physical readout. It also bares translational significance by disclosing a new high-performance strategy for cancer detection from liquid biopsies.

URL

https://www.biorxiv.org/content/biorxiv/early/2025/10/09/2025.10.08.681110.full.pdf

Closing the sampling-scoring gap: a MassiveFold study in CASP16

Oral Presentation

Nessim Raouraoua [1], ***Thomas Binet*** [2], ***Marc Lensink*** [1], ***Guillaume Brysbaert*** [1]

1. *Univ. Lille, CNRS UMR 8576-UGSF-Unité de Glycobiologie Structurale et Fonctionnelle, 59000 Lille, France,* ***2.*** *US 41 – UAR 2014 – PLBS, University of Lille, CNRS, Inserm, CHU Lille, Institut Pasteur of Lille, 59000 Lille, France*

Abstract

The CASP15-CAPRI experiment demonstrated that massive-scale sampling, combined with diversity-inducing parameters (AFsample by Björn Wallner), significantly improves protein structure prediction. However, the original AlphaFold2 architecture presents a major bottleneck for large-scale sampling by being limited to single-process calculations. To address these computational limitations, we collaborated with Björn Wallner to create MassiveFold [1], a tool that enables massive sampling by parallelizing structure prediction on GPU SLURM-based clusters.

In the 2024 CASP16 protein structure competition, we conducted a systematic experiment involving massive sampling for all protein targets [2]. Beyond our baseline submission (group 'Brysbaert'), we contributed these extensive datasets to the CASP Phase 2 'MassiveFold scoring' challenge. This collective effort aimed to provide the community, and specifically scorers, with a diverse pool of decoys to identify the 'hidden gems' generated by MassiveFold. Our analysis highlights a sampling-scoring gap: while these high-accuracy models exist within the ensembles, they are not systematically ranked first by standard AF2 confidence scores. We show that a perfect scoring would have ranked first in CASP16 with our MassiveFold set, highlighting an urgent need for advanced scoring methods. Furthermore, our data indicates that while 'easy' targets gain little from increased sampling, 'hard' targets consistently benefit from it. A standard AlphaFold2 run can serve as a diagnostic to predict which targets require massive sampling, allowing for a significant reduction in total computing time through selective resource allocation. Finally, we present the evolution of MassiveFold (v1.6.1), which now integrates AlphaFold3, PPI discovery, ligand screening, and support for local (non-SLURM) environments.

References:

[1] Nessim Raouraoua, Claudio Mirabello, Thibaut Véry, Christophe Blanchet, Björn Wallner, Marc F. Lensink and Guillaume Brysbaert. Nature Computational Science 4, 824–828 (2024).

[2] Nessim Raouraoua, Marc F. Lensink, Guillaume Brysbaert, Proteins: Structure, Function, and Bioinformatics (2025): 1–7, (2025).

URL

https://onlinelibrary.wiley.com/doi/10.1002/prot.70040

https://github.com/GBLille/Massivefold

Cluefish: a workflow for comprehensive biological interpretation of transcriptomic data series

Oral Presentation

Ellis Franklin [1], ***Elise Billoir*** [1], ***Philippe Veber*** [2], ***Jérémie Ohanessian*** [1], ***Marie Laure Delignette-Muller*** [2], ***Sophie Prud'homme*** [1]

1. *Université de Lorraine, CNRS, LIEC, F-57000 Metz, France,* ***2.*** *Université de Lyon, CNRS, VetAgro Sup, LBBE, F-69622 Villeurbanne, France*

Abstract

Transcriptomic "data series" — such as time-course or dose-response experiments — present significant challenges in both analytical modelling and biological interpretation. While specialised pipelines exist to characterise these series, a gap remains in tools dedicated to making sense of the extensive transcript lists they generate. To address this, functional enrichment analysis is often employed to associate these lists with simplified biological pathways. A commonly used method is Over-Representation Analysis (ORA), which uses statistical tests to determine if a pathway contains a disproportionately large number of deregulated genes compared to sampling among genes uniformly. However, standard ORA often yields broad, redundant results representing only a small fraction of deregulated genes, frequently overlooking smaller, more specific pathways.

To address these limitations, we developed Cluefish (CLUstering, Enrichment, and FISHing), an open-source semi-automated R workflow for untargeted exploration of transcriptomic data series. Cluefish applies ORA to pre-clustered protein-protein interaction networks, using clusters as anchors to identify smaller and more specific pathways. The workflow incorporates two innovative steps: cluster merging to reduce functional redundancy, and "lonely gene fishing", which recovers isolated deregulated genes through shared pathway context. Together, these features enable a more complete exploration of the data by maximising the proportion of the deregulated gene list included in the interpretation.

We tested Cluefish using an in-house dose-response dataset of zebrafish embryos exposed to dibutyl phthalate (known endocrine disruptor), as well as two publicly available toxicology datasets for rat and poplar. Compared to the standard approach, Cluefish allowed for the interpretation of a larger portion of the data and revealed disrupted pathways that would otherwise be overlooked. Coupled with dose-response modelling from DRomics, Cluefish outputs enabled the formulation of hypotheses supported by multiple concordant elements, which are not only biologically coherent but also of broader scientific interest.

URL

https://doi.org/10.1093/nargab/lqaf103

Combining phenotypic similarity and network propagation to improve performance and clinical consistency of rare disease diagnosis

Oral Presentation

Maroua CHAHDIL [1], ***Carolina Fabrizzi*** [2], ***Marc Hanauer*** [2], ***Caterina Lucano*** [2], ***Ana Rath*** [2], ***David Lagorce*** [2], ***Laurent Tichit*** [3]

1. *INSERM, US14 – Orphanet, Plateforme Maladies Rares, and Aix Marseille Univ, CNRS, I2M,* ***2.*** *INSERM, US14 – Orphanet, Plateforme Maladies Rares, 96 rue Didot, 75014, Paris, France,* ***3.*** *Aix Marseille Univ, CNRS, I2M, Marseille, France*

Abstract

Achieving timely diagnosis for rare diseases (RDs) remains challenging due to phenotypic heterogeneity and incomplete clinical data. Unlike our previous phenotype–genotype-based method developed during the SOLVE-RD project (Lagorce et al. 2024), and in contrast to existing phenotype-based tools that rarely exploit Orphanet's hierarchical classification, we present a phenotype-based computational pipeline that ranks candidate ORPHAcodes (Orphanet's unique, time-stable RD identifiers) from patient phenotypes by integrating a clinical consistency metric. Following the Orphanet Nomenclature, this metric quantifies the number of shared Groups of Disorders (GDs) between two ORPHAcodes. GDs are defined as sets of ORPHAcodes sharing clinical features. More shared GDs reflect greater clinical coherence.

The pipeline (Fig. 1) combines phenotype-based similarity with Random Walk with Restart (RWR). Patient-disease similarity is computed using asymmetric aggregation of the Human Phenotype Ontology (HPO) terms, incorporating removal of subsumed terms and integration of Orphanet HPO frequency annotations. Among 24 combinations of six similarity measures and four aggregation functions, Resnik with FunSimMaxAsym performed best.

By exploring the patient node's local neighbourhood, RWR identifies indirectly related diseases through network connectivity rather than direct annotation overlap, mitigating incompletely annotated cases and improving the clinical consistency of top-ranked candidates; although some confirmed diagnoses occasionally ranked lower, their GDs appeared prominently, supporting diagnostic utility.

We benchmarked our method on 139 expert-curated SOLVE-RD cases representing 78 distinct ORPHAcodes. Compared with the SOLVE-RD baseline (RSD), our approach achieved a harmonic mean rank of 4.64 for confirmed diagnoses (versus 7.97) and retrieved the correct RD within the top 10 positions for 39% of patients (versus 29%). The damping factor ($\alpha = 0.3$) modulated the trade-off between ranking accuracy and clinical coherence.

By integrating phenotype-based similarity with RWR, our pipeline enhances clinical consistency among top-ranked candidates by considering the Orphanet nomenclature, providing a starting point to assist clinicians in developing diagnostic hypotheses.

Code: github.com/Orphanet/OrphaScape_SimWalk

URL

https://hal.science/hal-05517071v1

Exploration of Multiconformers to Extract Information About Structural Deformation Undergone by a Protein Target: Illustration on the Bcl-xL Target

Oral Presentation

Marine Baillif [1], ***Eliott Tempez*** [1], ***Anne Badel*** [1], ***Leslie Regad*** [1]

1. *Université Paris Cité - BFA CNRS UMR 8251 - IsPP INSERM U1133*

Abstract

Analyzing how proteins change conformation upon ligand binding or mutations is essential for understanding molecular recognition and structure-based drug design. Traditionally, these changes are assessed by comparing apo and holo structures using RMSD. This parameter provides only a global measure of structural deviation and does not allow the localization of conformational changes. To overcome these limitations, we previously developed **SA-conf (Regad et al., 2017),** a tool that quantifies backbone conformational variability across sets of protein structures https://owncloud.rpbs.univ-paris-diderot.fr/owncloud/index.php/s/uOdItX27jUfgQM6. It is based on **HMM-SA (Camproux et al., 2004),** a structural alphabet that encodes local 3D geometry into sequences of structural letters where each encodes the geometry of consecutive four-Cα fragments.

In this study, we aimed to demonstrate the applicability of SA-conf for analyzing structural variability in Bcl-xL (B-cell lymphoma–extra-large), a key anti-cancer target. To this end, SA-conf was applied to a dataset of 130 crystallographic chains, including apo, holo, wild-type, and mutant forms. Our analysis revealed that **Bcl-xL shows** high structural plasticity. Structurally conserved positions were identified, notably in the protein fold and binding site anchor residues. While most mutations had limited local impact, some induced **long-range conformational rearrangements,** suggesting allosteric effects. We showed that ligand binding was the main driver of conformational changes, with rearrangements both near and distal to the binding site. By integrating SA-conf results with residue flexibility analysis, we established a structural mapping of the binding pocket with three important regions: (i) a **conserved anchoring core**, (ii) a**moderately plastic central region,** and (iii) a **flexible periphery** contributing to ligand specificity. These insights highlight SA-conf's ability to capture **functionally relevant backbone adaptations**. SA-conf offers a powerful framework for analyzing structural ensembles, identifying conformational signatures, and guiding **rational design of selective ligands** for dynamic protein targets. This study was previously reported in Baillif et al., Molecules, 2025

URL

https://www.mdpi.com/1420-3049/30/16/3355

Knowledge graph mining linking endometriosis and pollutants

Oral Presentation

Meije Mathé [1], ***Guillaume Laisney*** [2], ***Olivier Filangi*** [2], ***Franck Giacomoni*** [2], ***Maxime Delmas*** [3], ***German Cano-Sancho*** [4], ***Fabien Jourdan*** [5], ***Clément Frainay*** [1]

1. *Toxalim, Université de Toulouse, INRAE, ENVT, EI-Purpan,* ***2.*** *INRAE,* ***3.*** *Idiap Research Institute,* ***4.*** *Oniris, INRAE, Laberca,* ***5.*** *MetaboHUB, National Infrastructure of Metabolomics and Fluxomics*

Abstract

Endometriosis, a chronic inflammatory disease affecting 5–15% of individuals assigned female at birth, involves the presence of endometrial-like tissue outside the uterus. While genetic and hormonal factors are established contributors, environmental exposure to persistent organic pollutants (POPs), especially organochlorine compounds (OCCs), may also play a role, though the underlying mechanisms remain unclear due to fragmented evidence in the literature.

To address this, we developed Kg4j, an open-source Java framework for constructing question-centric biomedical knowledge graphs (KGs). Kg4j leverages FORVM, a large-scale RDF graph with 82 million associations between small molecules and biomedical concepts. Using ontology-based filtering, Kg4j extracts a locally hosted Neo4j subgraph, tailored to specific research questions, providing relevant literature-supported associations.

We applied Kg4j to explore links between OCC exposure and endometriosis risk. The resulting KG (2,706 nodes, 23,243 edges) was validated against a systematic review, achieving 95.4% recall and revealing new hypothetical associations. Centrality analysis highlighted hormone-related entities and steroid metabolites, supporting the hypothesis that endocrine disruption and altered steroid metabolism contribute to pathogenesis. Entities related to reproductive and neoplastic processes also emerged as central, suggesting potential links to cancer and environmental exposures.

Exploration of chains of concepts and chemical entities within the knowledge graph, from concept of interest to central nodes, revealed multi-step connections between OCC exposure and endometriosis-related processes. These interpretable paths enable hypothesis generation about potential mechanisms through which environmental pollutants may influence endometriosis development. The framework now supports metabolomics data integration, to extract KGs driven by both data and literature. This integration allows observed metabolic disruptions to be connected to known biological pathways, improving mechanistic insight and supporting more precise hypothesis generation.

This approach demonstrates how KG-based literature mining can transform fragmented knowledge into mechanistic hypotheses, offering a scalable tool for uncovering biomarkers, pathways, and therapeutic targets in complex diseases.

URL

https://doi.org/10.64898/2026.03.02.709027

Phage evolutionary relationships emerge from protein language model-based proteome representation

Oral Presentation

Swapnesh Panigrahi [1], ***Mireille Ansaldi*** [1], ***Nicolas Ginet*** [1]

1. *Laboratoire de Chimie Bactérienne*

Abstract

Bacteriophage taxonomy remains a challenging task due to the propensity of viruses for recombination and the lack of universal gene markers. Additionally, the rapid expansion of metagenomic datasets from diverse niches demands reliable and scalable approaches for classifying both known and novel phages. In this study, we introduce **Hierarchical Viruses (HieVi)**, a scalable framework that leverages protein language model (pLM) embeddings to compare and classify phages at the proteome level.

HieVi employs the ESM-2 model to generate vectorial representations for every protein in a phage genome. By treating phages as a "bag of proteins", we create a mean phage representation (MPR) in a 2,560-dimensional space. Using a curated dataset of 24,362 phages (INPHARED), we demonstrate that MPRs cluster effectively at the genus level. Hierarchical density-based clustering of these MPRs further reveals a multi-scale organization of phages that aligns remarkably well with ICTV taxonomic rankings, achieving an Adjusted Mutual Information score greater than 0.9 for families such as *Herelleviridae* at both genus and subfamily levels. Notably, HieVi demonstrates that highly mosaic lambdoid phages are clustered within a single, distinct branch and supports the recent ICTV promotion of *Autographiviridae* to the rank of order (*Autographivirales*). These results demonstrate that pLM-based representations can capture evolutionary relationships without relying on multiple sequence alignments.

Unlike traditional tools, HieVi is highly scalable because its vector-based formulation can leverage vector databases for rapid lookups. This enables contextual placement of new query genomes without reprocessing the entire database, facilitating the high-throughput analysis of large metagenomic datasets. The code and package are publicly available on Zenodo and GitHub, ensuring the tool is accessible and reusable. This framework shows that phage representation using protein language models contains phylogenetic information by encoding shared genes between phages, thus paving the way for advanced synteny-aware genome language models.

URL

[article] https://doi.org/10.1093/nargab/lqaf134

[App] https://huggingface.co/spaces/pswap/hievi
[Visualization] https://pswapnesh.github.io/HieVi/imgvr_galaxy.html

Revisiting SIF abstraction rules with SPARQL for querying BioPAX

Oral Presentation

Cécile Beust [1], ***Olivier Dameron*** [1], ***Nathalie Théret*** [2], ***Emmanuelle Becker*** [1]

1. *Univ Rennes, Inria, CNRS, IRISA - UMR 6074, Rennes, F-35000, France,* ***2.*** *Univ Rennes, Inria, CNRS, IRISA - UMR 6074, Rennes, F-35000, France ; Univ Rennes, Inserm, EHESP, Irset, UMR S1085,Rennes, F-35000, France*

Abstract

BioPAX (Biological PAthway eXchange) is a Semantic Web-based standard for representing biological pathways, offering fine-grained descriptions of molecular interactions, metabolic processes, and signaling networks. However, BioPAX' complexity poses challenges for downstream analyses and reasoning tasks, requiring abstraction methods to simplify representation and analysis. The Simple Interaction Format (SIF) is a widely used format addressing this issue by reducing BioPAX's mechanistic interactions into binary relationships between proteins and small molecules. Currently, Paxtools and ChiBE are the state-of-the-art tools for abstracting BioPAX to SIF, relying on 14 abstraction rules implemented as Java graph patterns. However, SIF rules are ambiguously documented, leading to inconsistencies and potential misinterpretations.

Our first contribution is a systematic analysis of SIF rules ambiguity, revealing three meanings of SIF rules descriptions: (1) diagrams and textual descriptions from PathwayCommons, (2) Paxtools pattern function descriptions (comments in the code), and (3) raw Paxtools pattern Java code. We identified that each subsequent meaning introduces additional constraints not specified in previous meanings, creating a gap between SIF rules documentation and implementation in Paxtools.

Our second contribution is the development of a transparent and reproducible BioPAX-to-SIF abstraction method introducing 14 SPARQL queries formalizing each SIF rule, and clearly documented through diagrams. Using our SPARQL queries, we quantified the impact of SIF rules ambiguity on a pathway example.

Our third contribution is the production of large scale SIF abstractions of two pathway databases, PathBank and Reactome, demonstrating scalability and efficiency of our queries.

Overall, our SPARQL-based approach provides a transparent, scalable and FAIR-compliant method for BioPAX-to-SIF abstraction. While Paxtools and ChiBE remain valuable for BioPAX manipulation, our queries offer a modular, well-documented alternative for users seeking clarity in SIF abstraction rules. The scalability and efficiency of the SPARQL queries demonstrate their potential for large-scale pathway analysis. SPARQL queries are publicly available on GitHub: https://github.com/CecileBeust/BioPAX-To-SIF-SPARQL.git

URL

https://zenodo.org/records/18980848

Spatial transcriptomic analysis reveals region-specific glial activation during epileptogenesis

Oral Presentation

Adrien Dufour [1], ***Christophe Le Priol*** [1], ***Baptiste Porte*** [1], ***Ronan Jouanard*** [1], ***Julien Maurizio*** [2], ***Anne-Elodie Receveur*** [1], ***Stéphane Auvin*** [1], ***Juliette Van Steenwinckel*** [1], ***Pierre Gressens*** [1], ***Andrée Delahaye-Duriez*** [1]

1. *INSERM,* ***2.*** *INOVARION*

Abstract

Temporal lobe epilepsy (TLE) is a common neurological disorder that often develops after an initial brain insult such as status epilepticus (SE). The transition from a healthy brain to a chronically epileptic state, known as epileptogenesis, involves complex molecular and cellular remodeling across multiple brain regions. Although transcriptomic studies have provided important insights into this process, most approaches rely on bulk or single-cell sequencing methods that lack spatial context. Here, we use spatial transcriptomics combined with integrative bioinformatic analyses to characterize the spatiotemporal transcriptional landscape of epileptogenesis.

Spatial transcriptomic profiles were generated using the 10x Genomics Visium platform from coronal brain sections of a lithium–pilocarpine rat model of SE at four time points (5, 10, 20, and 40 days post-induction), covering both latent and chronic phases. The dataset includes 16 samples and thousands of spatial capture spots per section, each representing the transcriptome of local cellular neighborhoods. Sequencing data were processed using Space Ranger and analyzed in R using Seurat, including SCTransform normalization, multi-slice integration, unsupervised spatial clustering, differential expression analysis, and pathway enrichment. Spatial cell composition was inferred using reference-based deconvolution with CARD and a mouse brain single-cell atlas.

Unsupervised clustering reconstructed major anatomical structures, including cortex, hippocampus, thalamus, white matter tracts, and caudate–putamen, demonstrating that transcriptomic signatures alone can recover brain architecture. Differential expression analysis identified more than 8,000 genes significantly dysregulated following SE, with enrichment in pathways related to immune activation, synaptic remodeling, and neuronal plasticity.

Spatial pathway analysis revealed pronounced reactive astrogliosis and microglial activation extending beyond the hippocampus into white matter tracts and thalamic nuclei during the latent phase of epileptogenesis. These results highlight the value of spatial transcriptomics and integrative bioinformatics workflows for uncovering region-specific molecular programs underlying neurological disease progression.

URL

https://doi.org/10.1186/s40478-026-02224-y

Spatiotemporal regulation of cell cycle states within the complex tumor microenvironment

Oral Presentation

Gianni Zanardelli [1], ***Olivier Tassy*** [1], ***Maulik Nariya*** [1], ***Nacho Molina*** [2]

1. *IGBMC,* ***2.*** *IE university*

Abstract

Dysregulation of the cell cycle is a hallmark of cancer, driving uncontrolled proliferation and contributing to tumor progression, metastasis, and therapy resistance. In addition to active cycling, many tumor cells can enter a quiescent (G0) state, halting proliferation without undergoing terminal differentiation. Quiescence has emerged as a key mechanism of therapeutic failure, allowing subpopulations of cancer cells to survive treatment and later initiate relapse. Furthermore, the tumor microenvironment (TME) plays a pivotal role in shaping cancer cell behavior, including their proliferative potential. Composed of a heterogeneous mix of stromal, immune, and malignant cells, the TME provides spatially organized signals that influence cell fate decisions. While immune and metabolic dimensions of the TME have been extensively studied, its role in regulating cell cycle states remains poorly understood. Particularly, the spatial distribution of proliferating versus quiescent tumor cell states, and how this organization contributes to intratumoral heterogeneity and therapeutic resistance, has yet to be systematically investigated.

In this work, we use publicly available spatial transcriptomics datasets to characterize the effect of the TME on cell cycle states in melanoma, breast, and lung tumors. Based on transcriptional profiles, we assigned discrete cell cycle phases to the cells within the TME. Interestingly we identified proliferating and quiescent subpopulations of cancer cells in the tumor. We further examined the spatial localization of these subpopulations and uncovered clear spatial patterning based on proliferating and quiescent cellular states. To refine this understanding, we develop CyclEM, a computational tool based on the Expectation-Maximization algorithm, which performs trajectory analysis, assigns a continuous cell cycle phase to proliferating cells, and reveals gene expression dynamics of key cell cycle regulators within the TME. Our approach enables a spatially resolved, high-resolution view of tumor cell cycle dynamics, uncovering heterogeneity in proliferation potential that may underlie differential responses to therapy and disease progression.

URL

https://www.biorxiv.org/content/10.64898/2025.11.28.691200v1

Supporting Workflow Reproducibility by Linking Bioinformatics Tools across Papers and Executable Code

Oral Presentation

Clémence Sebe [1], ***Olivier Ferret*** [2], ***Aurélie Névéol*** [1], ***Sarah Cohen-Boulakia*** [1]

1. *Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique,* ***2.*** *Université Paris-Saclay, CEA, List, F-91120*

Abstract

Motivation: The rapid growth of biological data has intensified the need for transparent, reproducible, and well-documented computational workflows. The ability to clearly connect the steps of a workflow in the code with their description in a paper would improve workflow understanding, support reproducibility, and facilitate reuse. This task requires the linking of Bioinformatics tools in workflow code with their mentions in a published workflow description.

Results: We present CoPaLink, an automated approach that integrates three components: Named Entity Recognition (NER) for identifying tool mentions in scientific text, NER for tool mentions in workflow code, and entity linking grounded on Bioinformatics knowledge bases. We propose approaches for all three steps achieving a high individual F1-measure (84 - 89) and a joint accuracy of 66 when evaluated on Nextflow workflows using Bioconda and Bioweb Knowledge bases. CoPaLink leverages corpora of scientific articles and workflow executable code with curated tool annotations to bridge the gap between narrative descriptions and workflow implementations.

Availability: The code is available at https://gitlab.liris.cnrs.fr/sharefair/copalink-experiments and https://gitlab.liris.cnrs.fr/sharefair/copalink. The corpora are also available at https://doi.org/10.5281/zenodo.18526700, https://doi.org/10.5281/zenodo.18526760 and https://doi.org/10.5281/zenodo.18543814.

URL

Link to preprint : https://arxiv.org/abs/2603.08195

Software CoPaLink : https://gitlab.liris.cnrs.fr/sharefair/copalink

Oral Full Proceedings

Benchmark Bias and Conformational Dynamics in Allosteric Site Prediction

Victor Pryakhin[1], Malika Smail-Tabbone[1], Yasaman Karami[2,*]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] Université de Lorraine, Inria, F-54000 Nancy, France
[*] corresponding author: yasaman.karami@inria.fr

Abstract

Allosteric site prediction plays a critical role in modern drug discovery, offering opportunities to target regulatory regions with high specificity. However, most existing computational approaches rely on static protein structures and pocket detection tools such as *fpocket*, thereby overlooking conformational dynamics essential for allosteric regulation. Here, we present AlloDyn, a framework that integrates static pocket descriptors with dynamic features derived from both all-atom molecular dynamics (MD) simulations of apo-state proteins and AlphaFlow-generated conformational ensembles. By capturing structural flexibility, solvent accessibility, and residue–residue communication patterns at the pocket level, our approach enables a dynamic-aware representation of candidate allosteric sites. Importantly, we identify a systematic bias in current benchmarking practices, showing that applying *fpocket* to holo structures without removing bound allosteric modulators introduces data leakage and leads to artificially inflated performance estimates. When evaluated on properly preprocessed datasets, dynamic feature augmentation significantly improves prediction performance over static baselines. Furthermore, we demonstrate that AlphaFlow-generated ensembles achieve performance comparable to MD-derived features at a fraction of the computational cost, providing a scalable alternative for conformational sampling. Benchmarking on the D24 dataset shows that AlloDyn achieves the best balance between precision and recall, yielding the highest F1 score and MCC among evaluated methods. We show that current benchmarks overestimate performance due to data leakage, and that incorporating dynamics is key to accurate and scalable allosteric site prediction.

Introduction

Allostery is a fundamental regulatory mechanism in which the binding of a molecule (*effector* or *modulator*) at one site of a protein (the *allosteric site*) induces functional changes at a distant site, often the *active site*. This long-range intramolecular communication enables proteins to modulate their activity in response to cellular signals. Allosteric regulation is widespread across biological systems and plays a central role in controlling enzymatic catalysis, metabolic pathways, and signal transduction [1]. Importantly, effectors can act as activators or inhibitors, finely tuning protein function without directly interacting with the substrate-binding site [2].

Allosteric sites often lie outside conserved active regions, making them attractive yet challenging targets for drug design. Unlike orthosteric drugs, allosteric modulators can offer enhanced specificity and reduced toxicity [3, 4]. However, the structural diversity of allosteric mechanisms across protein families has long complicated efforts to systematically identify allosteric sites.

The systematic collection of experimental annotations in recent years has greatly improved the accessibility of curated data on allosteric proteins. The Allosteric Database (ASD) [5] is the most comprehensive, with derived subsets such as ASBench [6] and CASBench [7] providing filtered, benchmarking-oriented collections. AlloBench [8] was recently introduced to facilitate standardized evaluation of predictive models. The growing volume and accessibility of annotated allosteric data have enabled the development of data-driven computational approaches for allosteric site prediction, offering scalable and generalizable alternatives to experimental identification.

Computational methods for allosteric site prediction can be broadly divided into two categories. *Residue-level* methods aim to identify individual allosteric residues within a protein, typically by combining structural, evolutionary, and dynamic features at the per-residue level, as in AR-Pred [9], which couples dynamics features using elastic network models with evolutionary information, NACEN [10], which exploits residue contact energy networks, and more recently a nanoenvironment descriptor-based approach [11]. The advent of protein language models (PLMs) has further enabled *sequence-based* residue prediction without requiring a three-dimensional structure, as demonstrated by AlloFusion [12], which leverages PLM embeddings, and Kannan et al. [13], which uses PLM attention maps to identify allosteric residues directly from sequence. *Pocket (or site)-level* methods, by contrast, operate on cavities detected on the protein surface and classify them as allosteric or non-allosteric, offering a more directly actionable output for drug design. While residue-level approaches provide finer mechanistic insight, pocket-level methods are better suited for identifying druggable sites and are the focus of the present work.

Pocket-level approaches have diversified over the past decade, yet most machine learning-based models still rely on static protein conformations. Many such methods use geometry-based pocket detection tools such as *fpocket* [14] to derive physicochemical features and employ classifiers, as implemented in *Allosite* [15] and the *PASSer* family [16–18]. Other recent strategies extend these approaches by incorporating complementary structural descriptors and advanced feature selection, as in *MEF-AlloSite* [19]. Beyond static representations, several methods rely on coarse-grained dynamic analysis. *AllositePro* [20] uses normal mode analysis based on an elastic network model to evaluate protein dynamic changes upon allosteric ligand binding; *AlloPred* [21] applies normal mode perturbation analysis to capture flexibility changes, and *ESSA* (Essential Site Scanning Analysis) [22] identifies residues whose perturbation strongly alters the dispersion of global motions within elastic network-based normal modes. A related perturbation-based approach, *APOP* [23], evaluates the allosteric potential of fpocket-detected pockets by stiffening residue–residue interactions within each cavity in a Gaussian network model and ranking them based on induced shifts in global mode frequencies combined with hydrophobicity descriptors. More recently, *AllosES* [24] expanded this concept by integrating transfer entropy–based coupling derived from Gaussian network models, followed by *AlloEF* [25] and *ZHMolEReP* [26], two residue-level methods that combine dynamic and energetic features (transfer entropy and energetic frustration for the former, perturbation response scanning and free energy approximation for the latter), both requiring active site information as input.

While the coarse-grained dynamic analyses described above go beyond purely static representations, they rely on simplified physical models and a single input structure, which may not fully capture the conformational heterogeneity central to allosteric regulation. Molecular dynamics (MD) simulations offer a principled route to capture such dynamics, sampling the conformational space of a protein at atomic resolution over time. However, the computational cost of all-atom MD simulations remains substantial, particularly at the scale of large benchmark datasets. The recent emergence of generative models for conformational sampling, such as AlphaFlow [27], offers a promising and scalable alternative. More broadly, the growing interest in dynamics-aware machine learning, including methods that incorporate conformational flexibility into binding interface prediction [28], underscores the potential of integrating protein dynamics into allosteric site modelling.

In this study, we present AlloDyn, which is, to the best of our knowledge, the first dynamic-aware method for allosteric site prediction that integrates static *fpocket* descriptors with dynamic features derived from full-atom MD simulations of apo (modulator-free) protein structures, allowing direct comparison with a static-only baseline. We further evaluate AlphaFlow [27], a diffusion-based generative model that constructs conformational ensembles directly from protein sequences, as a computationally tractable alternative to MD simulations. In addition, we identify and characterize a systematic bias introduced when *fpocket* is applied to holo structures without prior removal of the allosteric modulator, leading to data leakage and artificially inflated performance estimates. AlloDyn is evaluated against several state-of-the-art allosteric site prediction methods on the D24 benchmark set [20], demonstrating competitive and consistent classification performance.

Materials and Methods

Allosteric Dataset Construction and Annotation

The dataset used in this study was prepared following a pipeline inspired by PASSerRank [18], which was developed to curate a high-quality, structurally consistent, and sequence-diverse collection of protein–modulator complexes for allosteric site prediction. Protein structures and annotations were obtained from the 2023 release of the Allosteric Database (ASD2023) [5], which contains over 2,400 experimentally validated protein–modulator entries. To ensure structural reliability, we retained only X-ray structures with resolution better than 3.0 Å, and excluded entries with incomplete allosteric sites or ambiguous modulator annotations. Pocket identification was performed using *fpocket* [14], a widely used geometry-based method. We explored two preprocessing strategies prior to pocket detection. In the first, *fpocket* was applied directly to the full structure using a flag to restrict detection to the chain annotated as allosteric in the ASD, without any prior cleaning. In the second, solvent molecules and ions were removed, the allosteric modulator was extracted from the chain, and *fpocket* was then applied to the cleaned, isolated chain. For each protein, the geometric centers (*centroids*) of the predicted pockets and the modulator were computed. The pocket with the shortest centroid-to-centroid distance to the modulator (*pocket-modulator distance*) was labeled as allosteric, and all others as non-allosteric. This modulator-based labeling strategy was adopted as residue-level allosteric site annotations are not consistently available across all entries in ASD. If the minimum distance exceeded 10 Å (*default threshold*), the entry was excluded, as such cases typically reflect failures in pocket detection [18]. Redundancy reduction was applied to obtain a sequence-diverse dataset, using

pairwise sequence identity filtering with a 30% threshold. After detection, labeling, and filtering the final dataset consisted of 353 proteins, comprising 8,847 pockets in total, of which 353 were labeled as allosteric and 8,494 as non-allosteric (Figure 1-(1)).

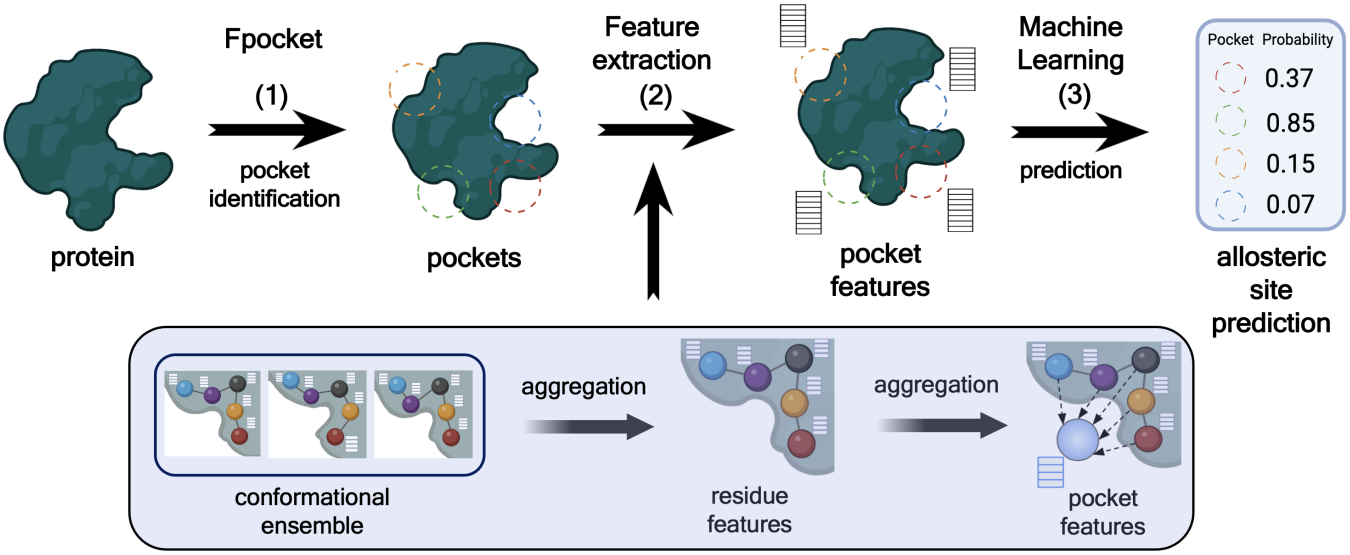

Figure 1: **Overview of the AlloDyn pipeline.** (1) *fpocket* detects pockets on the input protein structure. (2) Static features come directly from *fpocket* descriptors; dynamic features come from conformational ensembles generated by MD simulations or AlphaFlow, aggregated across the ensemble into residue-level features and then into pocket-level features. (3) XGBoost combines both feature sets and outputs an allosteric probability score for each pocket.

Generation of Conformational Ensembles

Molecular Dynamics simulations. For each of the 353 protein chains, we performed three replicates of 500-ns MD simulations, starting from their respective experimental structures. All bound ligands were removed, while ions were retained. Missing residues in regions with continuous gaps were modeled using MODELLER [29]. MD simulations were performed following the protocol described in [30], with identical simulation settings. Simulations were performed using GROMACS 2024.2 [31] with the CHARMM36m force field [32]. Systems were neutralized and adjusted to physiological ionic strength (150 mM NaCl) by adding Na^+ and Cl^- ions. Each system was solvated in a dodecahedral box of explicit TIP3P water model with a minimum buffer distance of 12 Å from the solute. Hydrogen atoms were added, and histidine protonation states were assigned using Reduce [33]. Energy minimization was carried out using the steepest descent algorithm for 5000 steps to remove steric clashes. This was followed by a multi-step equilibration phase in the NPT ensemble at 310 K, during which positional restraints were applied to protein and lipid heavy atoms and gradually released over 0.375 ns. Pressure was maintained at 1 bar using the Berendsen barostat during equilibration [34]. Production simulations were performed in the NPT ensemble with a time step of 2 fs. For each system, three independent replicates of 500 ns were generated using different initial velocities. Temperature was maintained at 310 K using the V-rescale thermostat [35], and pressure was controlled at 1 atm using the C-rescale barostat under isotropic conditions [36]. Covalent bonds involving hydrogen atoms were constrained using the LINCS algorithm [37], and long-range electrostatic interactions were treated with the Particle Mesh Ewald method [38]. Coordinates were recorded every 100 ps. For downstream analysis,

the three MD replicates were combined into a single concatenated trajectory for each protein, which served as the basis for dynamic feature extraction.

Generative models. As an alternative to classical MD simulations, we explored the use of AlphaFlow [27], a diffusion-based generative model that produces conformational ensembles directly from protein sequences. We employed its distilled version, which offers a favorable tradeoff between accuracy and computational efficiency. For consistency with MD-derived ensembles, we used the same curated protein sequences as input to AlphaFlow. For each protein entry, 25 conformations were generated. To ensure consistent atom and residue indexing, the sequences were aligned to their corresponding experimental structures prior to feature extraction.

Static and Dynamic Features Construction

Fpocket features

Each detected pocket was described using 19 descriptors computed by *fpocket*. These features capture geometric, physicochemical, and topological properties of the pocket, and include descriptors such as volume, solvent-accessible surface area (SASA), polarity, hydrophobicity, charge, and alpha-sphere statistics. Although one of these descriptors encodes residue flexibility via B-factors, all fpocket descriptors are derived from a single static structure rather than from conformational ensembles, and are referred to as *static* or *fpocket* features throughout this work.

Dynamic features

We computed dynamic descriptors from conformational ensembles to capture various aspects of pocket dynamics (Figure 1-(2)). Pockets were defined on the experimental structures using *fpocket* and represented as fixed sets of residues. Dynamic descriptors were computed by tracking these residues across conformational ensembles at the pocket level. Dynamic features were organised into three groups as described below (full list of features is available in Supplementary Table S1).

Compactness and shape features All features in this group were computed from the Cα atom coordinates of pocket residues using MDAnalysis [39]. For each frame, the pocket centroid was computed as $\mathbf{c}_t = \frac{1}{N}\sum_i \mathbf{r}_i^t$ over the N Cα atoms of the pocket. Based on this definition, we extracted nine descriptors capturing pocket geometry and variability across the ensemble. These include the standard deviation of the centroid position, as well as the mean and standard deviation of the minimum, average, and maximum Cα-to-centroid distances. In addition, mean pairwise inter-residue distances were computed per frame and summarised by their mean and standard deviation across the ensemble. Pocket compactness was quantified using the root mean square radius, $r_{\mathrm{rms}}(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{r}_i^t - \mathbf{c}_t\|^2}$, computed per frame and summarised by its mean and standard deviation. Residue-level flexibility was captured through per-residue RMSF values, aggregated as mean and standard deviation across pocket residues. Finally, pocket shape was characterised per frame by the eigenvalues of the 3×3 spatial covariance matrix of Cα positions, $\mathbf{C}_t = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{r}_i^t - \mathbf{c}_t)(\mathbf{r}_i^t - \mathbf{c}_t)^\top$. The resulting eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ describe the principal axes of the instantaneous spatial distribution of pocket residues, and were used to derive anisotropy $A = \lambda_3/\lambda_1$ and sphericity $S = (\lambda_1\lambda_2\lambda_3)^{1/3}/(\lambda_1+\lambda_2+\lambda_3)$, both summarised by their mean and standard deviation over the ensemble (17 features).

SASA-based features Pocket solvent-accessible surface area is computed per frame using MDTraj [40]. Both absolute (Å^2) and relative (fraction of total protein SASA) values are recorded, with mean, standard deviation, range, and coefficient of variation computed for each, leading to a total of 8 features per pocket.

Dynamic coupling features Pairwise couplings between all protein residues are quantified using four matrices. The *dynamic cross-correlation* (Pearson) measures the normalised cross-covariance of C_α positional fluctuations [41]:

$$\mathrm{DCC}_{ij} = \frac{\langle \Delta\mathbf{r}_i \cdot \Delta\mathbf{r}_j \rangle}{\sqrt{\langle |\Delta\mathbf{r}_i|^2 \rangle \langle |\Delta\mathbf{r}_j|^2 \rangle}}, \qquad \Delta\mathbf{r}_i = \mathbf{r}_i - \langle \mathbf{r}_i \rangle \tag{1}$$

Mutual information (MI) and *generalised correlation* (GC) capture both linear and nonlinear couplings [42]. Under the Gaussian approximation:

$$\mathrm{MI}_{ij} = \frac{1}{2}\left[\ln\det(\Sigma_i) + \ln\det(\Sigma_j) - \ln\det(\Sigma_{ij})\right], \tag{2}$$

where Σ_i, Σ_j are the covariance matrices of residues i and j, and Σ_{ij} is their joint covariance matrix. GC [42] is derived from MI as:

$$\mathrm{GC}_{ij} = \sqrt{1 - \exp\left(-\frac{2}{3}\,\mathrm{MI}_{ij}\right)}, \tag{3}$$

Finally, the *communication propensity* (CP) quantifies the variance of the inter-residue distance across the ensemble [43]:

$$\mathrm{CP}_{ij} = \langle (d_{ij} - \bar{d}_{ij})^2 \rangle, \tag{4}$$

where d_{ij} is the instantaneous distance between C_α atoms of residues i and j. GC, MI, and CP were computed using ComPASS [44]. For each matrix, three sets of pocket-level statistics are derived: *intra-pocket* couplings (mean, max, and std of all residue pairs within the pocket), *global connectivity* (mean and std of couplings between pocket residues and the rest of the protein), and *inter-pocket* couplings (statistics over couplings between this pocket and every other detected pocket).

Model Training and Evaluation

Classifier and hyperparameter optimization. We trained gradient-boosted decision tree models using XGBoost [45], selected for its speed, scalability, and effectiveness on imbalanced classification tasks (Figure 1-(3)). To prevent data leakage and ensure generalization across protein structures, we employed a group-aware splitting strategy: both the train/test split (80/20) and the 5-fold cross-validation for hyperparameter tuning were performed such that all pockets from the same protein were consistently assigned to a single partition. Hyperparameters were optimized via grid search, with model selection based on the average F1 score across cross-validation folds. To address the substantial class imbalance between allosteric and non-allosteric pockets, the `scale_pos_weight` parameter, which controls the weight assigned to the positive class, was included in the hyperparameter search space. The best-performing configuration was used to retrain the model on the full training set and evaluated on the independent test set.

Evaluation of dynamic feature contribution. To obtain robust performance estimates, the full training and evaluation pipeline was repeated across 50 random group-based train/test splits. This procedure was applied to the static baseline model, as well as to models augmented with dynamic features from MD

simulations or AlphaFlow-generated conformational ensembles. To ensure a fair comparison between MD- and AlphaFlow-derived features, only entries with fewer than 862 residues were retained, as this represented the upper length limit for AlphaFlow, resulting in 339 entries (7,779 pockets, non-allosteric-to-allosteric ratio ∼22:1). Performance was evaluated primarily using the F1 score, given the strong class imbalance; precision, recall, MCC, AUC-ROC, and AUC-PR were additionally computed.

Benchmarking against state-of-the-art methods. To compare our approach with existing methods, we used the D24 benchmark set from AllositePro [20]. Entries sharing high sequence identity with D24 proteins were removed from our training dataset using pairwise sequence identity filtering at a 30% threshold, yielding a filtered training set of 293 entries (6,704 pockets, class ratio ∼22:1). The final model was trained on this filtered dataset using 5-fold group-aware cross-validation to select optimal hyperparameters, followed by retraining on the full filtered set. This model was then evaluated on D24 alongside PASSer (ensemble [16], AutoML [17], and ranking-based [18] variants), AllosES [24], and APOP [23]. D24 structures were processed using the same pipeline applied to our main dataset: ligands, solvent molecules, and ions were removed, and for each entry, only the single functional chain annotated as allosteric in the AllositePro set was retained (Supplementary Table S2). The pocket closest to the bound modulator was labeled as the allosteric site, resulting in exactly one positive entry per structure. For all methods, precision, recall, F1-score, MCC, AUC-ROC, and AUC-PR were computed; for methods providing only ranking scores (APOP, PASSer-ranking), all positively ranked pockets were treated as predicted positives prior to metric computation.

Results

Modulator Presence Introduces a Systematic Bias in Fpocket Descriptors

When *fpocket* is applied directly to a holo structure without prior removal of the allosteric modulator, the resulting pocket descriptors can differ substantially from those obtained after pre-cleaning. We refer to these two strategies as *biased* (no pre-cleaning) and *unbiased* (pre-cleaned structure) throughout this work. We identified two sources of inconsistency introduced by the biased approach. First, in cases where the allosteric modulator is a modified amino acid, *fpocket* may incorporate it into the pocket definition or even construct a pocket around it, directly biasing the pocket geometry and composition. Second, and more systematically, we observed that while *fpocket* does not consider the modulator during pocket *detection*, it appears to include it during *feature calculation*. In particular, SASA values are computed over the full structure including the modulator, which artificially reduces the solvent-exposed area of the allosteric pocket in the biased case. Since the *fpocket* score and druggability score are derived from regression formulas that incorporate SASA, this propagates into a systematic upward shift of these scores for allosteric pockets in the biased dataset. This effect is clearly visible in the score distributions (Figure 2A): while non-allosteric pockets show nearly identical distributions between the biased and unbiased configurations, allosteric pockets exhibit a pronounced rightward shift in *fpocket* score and druggability score under the biased configuration. Manual inspection of individual entries (Figure 2B-C, Supplementary Table S3) confirmed that the primary differences between biased and unbiased descriptors are concentrated in *fpocket* score, druggability score, and SASA-related features, while geometric descriptors such as volume, alpha-sphere statistics, and flexibility remained largely unchanged.

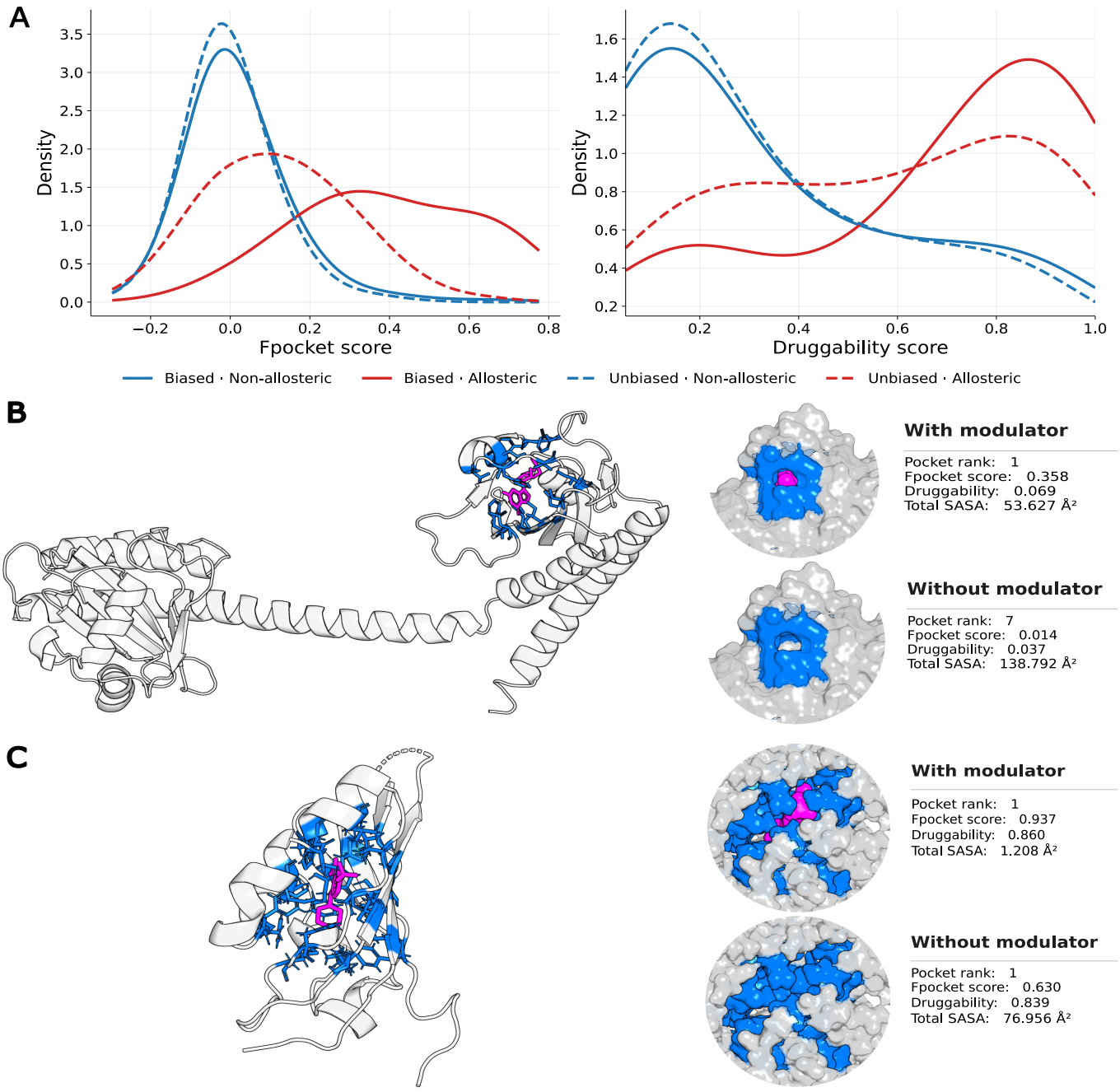

Figure 2: **Comparison of fpocket descriptors between biased and unbiased configurations. (A)** Kernel density estimates of fpocket score and druggability score across all detected pockets, split by allosteric and non-allosteric labels. **(B)** 1MC0 and **(C)** 3F1O allosteric binding sites shown in biased (modulator present) and unbiased (modulator removed) configurations. Left: full protein structure (cartoon) with allosteric residues (marine sticks) and modulator (magenta sticks). Right: zoomed surface views of the allosteric pocket (marine) with (top) and without (bottom) the modulator (magenta surface). Removal of the modulator affects the *fpocket* score (and potentially the pocket rank), druggability score, and total SASA, illustrating the systematic bias introduced when *fpocket* is applied to holo structures.

The impact of this data leakage is directly reflected in model performance (Table 1): models trained on the biased dataset achieve substantially higher F1 scores than those trained on the unbiased dataset.

However, this apparent performance gain is artifactual: feature importance analysis (Supplementary Figure S1) confirms that the *fpocket* score is the dominant predictor in the biased setting, disproportionately driving classification due to its artificially inflated values for allosteric pockets. This inflated performance does not reflect the model's ability to identify genuine allosteric features, and cannot be used as a reliable estimate of predictive performance.

Table 1: **Model performance on biased and unbiased datasets, averaged over 50 random seeds.** For each dataset, results are reported for the *fpocket*-only baseline and AlloDyn models augmented with MD- or AlphaFlow-derived (AF) dynamic features. Values are reported as mean ± standard deviation of test set performance across the 50 splits.

Source	Model	Precision	Recall	F1	MCC	AUC ROC	AUC PR
Biased	Baseline	0.57 ± 0.05	0.65 ± 0.09	0.60 ± 0.05	0.59 ± 0.05	0.94 ± 0.02	0.61 ± 0.06
	AlloDyn (Fpocket+MD)	0.54 ± 0.03	0.69 ± 0.09	0.61 ± 0.04	0.59 ± 0.05	0.95 ± 0.01	0.59 ± 0.04
	AlloDyn (Fpocket+AF)	0.56 ± 0.05	0.68 ± 0.08	0.61 ± 0.05	0.60 ± 0.05	0.95 ± 0.02	0.62 ± 0.05
Unbiased	Baseline	0.46 ± 0.06	0.41 ± 0.06	0.43 ± 0.05	0.41 ± 0.05	0.86 ± 0.03	0.40 ± 0.06
	AlloDyn (Fpocket+MD)	0.48 ± 0.06	0.42 ± 0.05	0.45 ± 0.05	0.43 ± 0.05	0.86 ± 0.02	0.41 ± 0.06
	AlloDyn (Fpocket+AF)	0.48 ± 0.06	0.42 ± 0.06	0.45 ± 0.05	0.43 ± 0.05	0.87 ± 0.02	0.40 ± 0.06

Taken together, these findings indicate that applying *fpocket* to uncleaned holo structures introduces a data leakage: allosteric pockets receive artificially inflated scores due to the presence of the modulator, making them easier to classify for reasons unrelated to genuine structural features. The unbiased preprocessing strategy, in which the modulator is removed prior to pocket detection, eliminates this artifact and provides a more reliable basis for model training and evaluation.

Dynamic Feature Augmentation Improves Allosteric Site Prediction

To evaluate the contribution of dynamic features to allosteric site prediction, we compared models trained on *fpocket* descriptors alone (baseline) to those augmented with dynamic features derived from MD simulations or AlphaFlow-generated conformational ensembles, using the unbiased dataset. Model training and evaluation were repeated across 50 random group-based train/test splits. Statistical significance of performance differences was assessed using two-tailed paired t-tests on matched splits, after confirming normality of all metric distributions with the Shapiro–Wilk test ($p > 0.05$); effect sizes were quantified using Cohen's d (more details in Supplementary Table S4).

Both AlloDyn variants significantly outperformed the static baseline across multiple metrics (Table 1). For F1 and MCC, improvements were statistically significant ($^{**}p < 0.01$) with small effect sizes ($|d| \approx 0.4$). For AUC-PR, the MD-augmented model reached a medium effect size ($|d| = 0.56$, $^{***}p < 0.001$). Models trained on MD- and AlphaFlow-derived features performed comparably across all metrics, with no statistically significant difference between the two conformational sources for F1 and MCC ($p > 0.98$, $|d| \approx 0.00$). This indicates that AlphaFlow-generated ensembles can serve as a computationally efficient alternative to MD simulations within the current classification framework.

Comparison with State-of-the-Art Methods

AlloDyn was evaluated on the D24 benchmark set against PASSer (ensemble [16], AutoML [17], and ranking-based [18] variants), AllosES [24], and APOP [23]. For this evaluation, the AlloDyn model was trained using AlphaFlow-generated conformational ensembles as the source of dynamic features. All methods were evaluated using the same set of classification metrics (Table 2). For methods providing only ranking scores (APOP, PASSer-ranking), all positively ranked pockets were treated as predicted positives when computing classification metrics. The number of evaluated pockets (N) varies across methods. AlloDyn and PASSer operate on the same pocket set ($N = 645$). The small discrepancy for APOP ($N = 641$) arises from its mandatory removal of non-standard amino acids during preprocessing, which introduces gaps in the sequence and alters the alpha-sphere clustering step in *fpocket*, resulting in slightly different pocket definitions for a small number of entries. For AllosES, the use of an older version of *fpocket* in combination with non-standard amino acid removal led to a substantially different pocket set ($N = 305$).

Table 2: **Performance comparison of allosteric site prediction methods on the D24 benchmark.** Metrics are reported at the pocket level. N denotes the number of evaluated pockets. Best values per column are shown in bold.

Method	Precision	Recall	F1	MCC	AUC-ROC	AUC-PR	N
AlloDyn	0.52	0.50	**0.51**	**0.49**	**0.93**	0.44	645
APOP	0.09	**0.95**	0.17	0.23	**0.93**	0.52	641
AllosES	0.24	0.79	0.37	0.37	0.86	**0.54**	305
AllosES*	**0.60**	0.27	0.38	0.39	0.90	0.41	645
PASSer-automl	0.50	0.27	0.35	0.35	0.91	0.32	645
PASSer-ensemble	0.45	0.23	0.30	0.31	0.86	0.38	645
PASSer-rank	0.21	0.14	0.17	0.15	0.84	0.28	645

*Re-evaluated with fpocket v.4 to match the pocket set used by all other methods ($N = 645$).

Figure 3 illustrates representative prediction outcomes on the D24 benchmark. Panel **A** shows a prediction example for 4OYA_A, human soluble adenylate cyclase, demonstrating the discriminative power of AlloDyn: while AllosES and PASSer variants assign the correct allosteric pocket a lower score and instead prioritize a distant, non-allosteric cavity, AlloDyn correctly identifies it as the top prediction. Panels **B**–**C** illustrate the fundamental limitation imposed by the pocket detection step: for 5J94_A and 4NHV_A, no *fpocket*-detected pocket falls within the 10 Å labelling threshold of the allosteric modulator, meaning all pockets were labelled as non-allosteric and the correct site could not be recovered by any method. Detailed predictions for all D24 entries across all compared methods are provided in Supplementary Figures S2–S4.

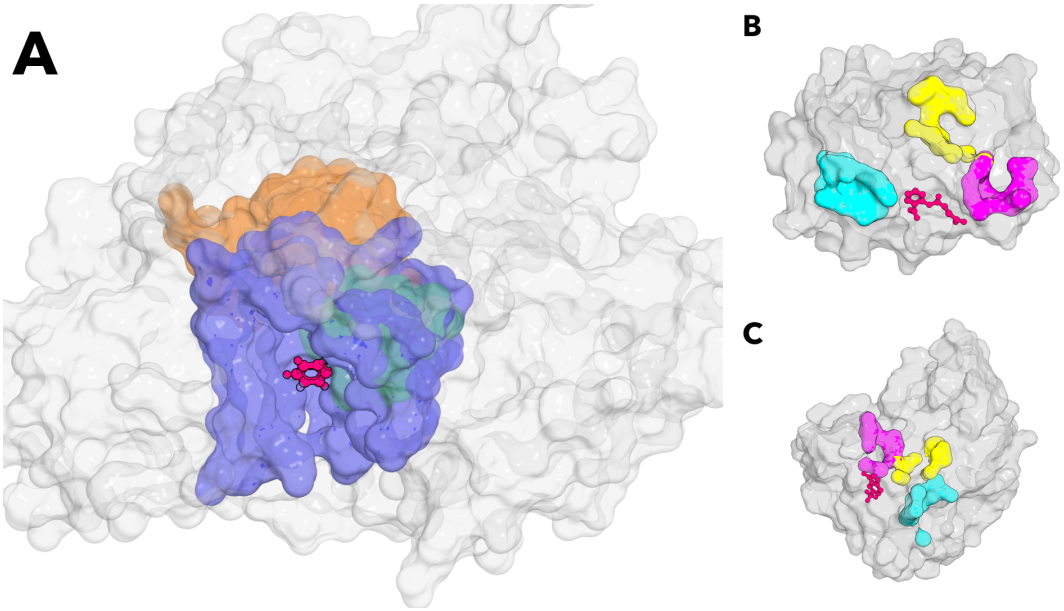

Figure 3: **Representative allosteric pocket predictions on the D24 benchmark. (A)** Prediction comparison for 4OYA_A. The correct allosteric pocket is ranked first by AlloDyn (green, buried pocket, 2.5 Å from modulator) and APOP (blue, larger binding-site representation). The difference in pocket extent between the two methods arises from APOP's removal of modified residues during preprocessing. AllosES and PASSer variants assign a lower rank to the correct allosteric pocket, instead prioritizing a distant pocket (orange, 15.3 Å from the modulator) as their top prediction. Modulator is shown in pink (sticks). **(B–C)** Failure cases for 5J94_A and 4NHV_A, respectively. No *fpocket*-detected pocket falls within the 10 Å labelling threshold of the allosteric modulator (pink sticks), showing that prediction performance is fundamentally constrained by the pocket detection step in these cases. The three closest detected pockets are shown: the nearest (magenta), second nearest (cyan), and third nearest (yellow). Protein surface is shown in light gray.

Among all evaluated methods, AlloDyn achieves the best balance between precision and recall, yielding the highest F1 score (0.51) and MCC (0.49, Table 2). APOP attains the highest recall (0.95) together with a competitive AUC-PR (0.52), but at the cost of very low precision (0.09), indicating that a large fraction of detected pockets are predicted as allosteric, leading to numerous false positives. Similarly, AllosES achieves the highest AUC-PR (0.54), suggesting strong ranking performance across thresholds, but its lower precision results in reduced F1 and MCC scores compared to AlloDyn. PASSer variants show moderate and consistent performance across metrics, with PASSer-AutoML performing best within the family. These results indicate that while APOP and AllosES are effective at assigning elevated scores to allosteric pockets, they tend to overpredict positive sites. In contrast, AlloDyn provides a more balanced tradeoff between sensitivity and specificity, leading to more reliable threshold-based classification performance. In practical applications, where experimental validation of predicted allosteric sites is costly, reducing false positives while maintaining competitive recall may represent a more useful operating regime.

Discussion

This study investigated whether dynamic features derived from conformational ensembles can improve allosteric site prediction beyond what is achievable with static *fpocket* descriptors alone. Using a curated

dataset of annotated allosteric and non-allosteric pockets, we trained and evaluated XGBoost classifiers on static features, conformational ensemble-derived dynamic features (MD or AlphaFlow-generated). Our results show that dynamic feature augmentation significantly improves classification performance when structures are properly preprocessed, while also revealing several methodological bottlenecks that limit the utility of dynamic descriptors in the current framework.

Pocket detection constitutes the first and most critical step in our pipeline, and its accuracy directly determines the upper bound of downstream prediction performance. When *fpocket* is applied to holo structures without prior removal of the allosteric modulator, SASA-based descriptors are computed over the full structure including the modulator, artificially inflating the *fpocket* score and druggability score for allosteric pockets. This constitutes a data leakage that leads to overoptimistic performance estimates. Removing the modulator prior to pocket detection eliminates this artifact and provides a more reliable basis for model training. Beyond preprocessing, *fpocket* may simply fail to detect the correct allosteric pocket in some cases, as illustrated by the two D24 entries (5J94_A and 4NHV_A) where no detected pocket fell within the labelling threshold of the modulator. In such cases, no downstream method can recover the correct site, highlighting the fundamental dependency of our approach on pocket detection quality.

Despite the dominance of static descriptors, dynamic feature augmentation led to a statistically significant improvement in F1 score on the unbiased dataset. Notably, some static *fpocket* descriptors (in particular the *fpocket* score and druggability score) remained among the top predictors across all settings, reflecting their strong discriminative power even in the absence of dynamic information. These results suggest that dynamic features are most informative when the allosteric pocket is accurately identified, highlighting the dependency of dynamic descriptors on the quality of the initial pocket definition.

Dynamic features were computed by anchoring to the set of pocket-forming residues identified in the experimental structure. While this provides a consistent reference across conformations, it is an approximation: the true pocket boundary may shift during simulation, and transient or cryptic regions may not be captured by residues defined on a single static structure. More sophisticated pocket tracking strategies, such as ensemble-based pocket detection or adaptive residue mapping across conformations, could improve the accuracy of dynamic descriptors. Furthermore, obtaining pocket-level features requires a double aggregation - from the conformational ensemble to per-residue features, and then from residues to the pocket level - which inevitably discards spatial and temporal resolution. More expressive architectures, such as graph neural networks with learnable aggregation, could better preserve this information and more fully exploit the rich structural information contained in conformational ensembles.

Correlation-based features capture residue-residue communication patterns that are conceptually well-suited to allosteric regulation. However, their computation requires knowledge of which residues belong to each pocket, and the current approach uses all detected pockets for inter-pocket correlation calculations due to the absence of explicit orthosteric site annotations in the dataset. This introduces noise in the inter-pocket and global connectivity features, as non-functional pockets are treated on equal footing with biologically relevant ones. Future work incorporating orthosteric site annotations would allow more targeted and interpretable correlation descriptors.

Despite known structural imperfections such as occasional atomic clashes, models trained on AlphaFlow-derived dynamic features achieved performance comparable to those using MD-derived ensembles, with no statistically significant differences. A current limitation of AlphaFlow is that it operates on single protein chains, whereas MD simulations can accommodate multimeric complexes. For proteins whose allosteric regulation depends on inter-chain interactions, this represents a meaningful constraint. Nevertheless, for monomeric systems, AlphaFlow offers a practical and computationally efficient alternative to MD for conformational sampling in this context.

Overall, our results demonstrate that integrating dynamic features into allosteric site prediction is a promising direction, but its effectiveness is conditioned on reliable pocket detection, accurate structural annotations, and sufficiently expressive feature representations. Expanding annotated datasets with orthosteric site information and developing architectures capable of learning directly from raw conformational ensembles are key directions for future work.

Data and code availability

The source code and training datasets are publicly available on GitLab at https://gitlab.inria.fr/vpryakhi/AlloDyn.

Acknowledgements

YK was supported by the French National Research Agency (ANR) under the France 2030 grant reference number ANR-24-RRII-0002 operated by the Inria Quadrant Program. Computations were performed using resources from the Grid'5000 and MBI computing platforms. This work was granted access to the HPC resources of IDRIS under the allocation 2025-A0180714660 granted to Y.K. made by GENCI.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Nina M. Goodey and Stephen J. Benkovic. Allosteric regulation and catalysis emerge via a common route. *Nat. Chem. Biol.*, 4(8):474–482, 2008.

2. Mingyu Li, Xiaobin Lan, Xun Lu, and Jian Zhang. A Structure-Based Allosteric Modulator Design Paradigm. *Health Data Sci.*, 3:0094, 2023.

3. Ashok Kumar Grover. Use of Allosteric Targets in the Discovery of Safer Drugs. *Med. Princ. Pract.*, 22(5):418–426, 2013.

4. Nan Wu, Léonie Strömich, and Sophia N. Yaliraki. Prediction of allosteric sites and signaling: Insights from benchmarking datasets. *Patterns*, 3(1):100408, 2022.

5. Jixiao He, Xinyi Liu, Chunhao Zhu, Jinyin Zha, Qian Li, Mingzhu Zhao, Jiacheng Wei, Mingyu Li, Chengwei Wu, Junyuan Wang, Yonglai Jiao, Shaobo Ning, Jiamin Zhou, Yue Hong, Yonghui Liu, Hongxi He, Mingyang Zhang, Feiying Chen, Yanxiu Li, Xinheng He, Jing Wu, Shaoyong Lu, Kun Song, Xuefeng Lu, and Jian Zhang. ASD2023: Towards the integrating landscapes of allosteric knowledgebase. *Nucleic Acids Res.*, 52(D1):D376–D383, 2024.

6. Wenkang Huang, Guanqiao Wang, Qiancheng Shen, Xinyi Liu, Shaoyong Lu, Lv Geng, Zhimin Huang, and Jian Zhang. ASBench: Benchmarking sets for allosteric discovery. *Bioinformatics*, 31(15):2598–2600, 2015.

7. A Zlobin, D Suplatov, K Kopylov, and V Švedas. Casbench: a benchmarking set of proteins with annotated catalytic and allosteric sites in their structures. *Acta Naturae*, 11(1 (40)):74–80, 2019.

8. Dibyajyoti Maity and Baofu Qiao. AlloBench: A Data Set Pipeline for the Development and Benchmarking of Allosteric Site Prediction Tools. *ACS Omega*, 10(17):17973–17982, 2025.

9. Sambit K. Mishra, Gaurav Kandoi, and Robert L. Jernigan. Coupling dynamics and evolutionary information with structure to identify protein regulatory and functional binding sites. *Proteins*, 87(10):850–868, 2019.

10. Wenying Yan, Guang Hu, Zhongjie Liang, Jianhong Zhou, Yang Yang, Jiajia Chen, and Bairong Shen. Node-Weighted Amino Acid Network Strategy for Characterization and Identification of Protein Functional Residues. *J. Chem. Inf. Model.*, 58(9):2024–2032, 2018.

11. Folorunsho Bright Omage, José Augusto Salim, Ivan Mazoni, Inácio Henrique Yano, Luiz Borro, Jorge Enrique Hernández Gonzalez, Fabio Rogerio De Moraes, Poliana Fernanda Giachetto, Ljubica Tasic, Raghuvir Krishnaswamy Arni, and Goran Neshich. Protein allosteric site identification using machine learning and per amino acid residue reported internal protein nanoenvironment descriptors. *Comput. Struct. Biotechnol. J.*, 23:3907–3919, 2024.

12. Jiabin Huang, Dongliang Guo, Yapeng Liu, Yanfen Wang, and Mengya Lv. Allofusion: Allosteric Site Prediction Based on Language Models and Multi-Feature Fusion. *J. Chem. Inf. Model.*, 65(16):8858–8870, 2025.

13. Gokul R. Kannan, Brian L. Hie, and Peter S. Kim. Single-sequence, structure free allosteric residue prediction with protein language models. *bioRxiv*, 2024.

14. Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinform.*, 10(1):168, 2009.

15. Wenkang Huang, Shaoyong Lu, Zhimin Huang, Xinyi Liu, Linkai Mou, Yu Luo, Yanlong Zhao, Yaqin Liu, Zhongjie Chen, Tingjun Hou, and Jian Zhang. Allosite: A method for predicting allosteric sites. *Bioinformatics*, 29(18):2357–2359, 2013.

16. Hao Tian, Xi Jiang, and Peng Tao. PASSer: Prediction of allosteric sites server. *Mach. Learn.: Sci. Technol.*, 2(3):035015, 2021.

17. Sian Xiao, Hao Tian, and Peng Tao. PASSer2.0: Accurate Prediction of Protein Allosteric Sites Through Automated Machine Learning. *Front. Mol. Biosci.*, 9:879251, 2022.

18. Hao Tian, Sian Xiao, Xi Jiang, and Peng Tao. PASSerRank: Prediction of allosteric sites with learning to rank. *J. Comput. Chem.*, 44(28):2223–2229, 2023.

19. Sadettin Y. Ugurlu, David McDonald, and Shan He. MEF-AlloSite: An accurate and robust Multimodel Ensemble Feature selection for the Allosteric Site identification model. *J. Cheminform.*, 16(1):116, 2024.

20. Kun Song, Xinyi Liu, Wenkang Huang, Shaoyong Lu, Qiancheng Shen, Lu Zhang, and Jian Zhang. Improved Method for the Identification and Validation of Allosteric Sites. *J. Chem. Inf. Model.*, 57(9):2358–2363, 2017.

21. Joe G Greener and Michael Je Sternberg. AlloPred: Prediction of allosteric pockets on proteins using normal mode perturbation analysis. *BMC Bioinform.*, 16(1):335, 2015.

22. Burak T. Kaynak, Ivet Bahar, and Pemra Doruker. Essential site scanning analysis: A new approach for detecting sites that modulate the dispersion of protein global motions. *Comput. Struct. Biotechnol. J.*, 18:1577–1586, 2020.

23. Ambuj Kumar, Burak T. Kaynak, Karin S Dorman, Pemra Doruker, and Robert L. Jernigan. Predicting allosteric pockets in protein biological assemblages. *Bioinformatics*, 39(5):btad275, 2023.

24. Fangrui Hu, Fubin Chang, Lianci Tao, Xiaohan Sun, Lamei Liu, Yingchun Zhao, Zhongjie Han, and Chunhua Li. Prediction of Protein Allosteric Sites with Transfer Entropy and Spatial Neighbor-Based Evolutionary Information Learned by an Ensemble Model. *J. Chem. Inf. Model.*, 64(15):6197–6204, August 2024.

25. Jilong Zhang, Xiaohan Sun, Zhixiang Wu, Jingjie Su, Xinyu Zhang, and Chunhua Li. AlloEF: An Ensemble Model for Protein Allosteric Site Identification Based on Transfer Entropy and Energetic Frustration. *J. Phys. Chem. B*, 130(19):4970–4981, 2026.

26. Xing Ke, Haoquan Liu, Jian Wang, Wenxin Guo, He Feng, and Yunjie Zhao. ZHMolEReP: An Energy Response Strategy for Protein Allosteric Site Prediction. *J. Chem. Inf. Model.*, 2026.

27. Bowen Jing, Bonnie Berger, and Tommi Jaakkola. AlphaFold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

28. Omid Mokhtari, Sergei Grudinin, Yasaman Karami, and Hamed Khakzad. DynamicGT: A dynamic-aware geometric transformer model to predict protein-binding interfaces in flexible and disordered regions. *Cell Syst.*, 17(1), 2026.

29. Andrej Šali and Tom L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of molecular biology*, 234(3):779–815, 1993.

30. Omid Mokhtari, Emmanuelle Bignon, Hamed Khakzad, and Yasaman Karami. DynaRepo: The repository of macromolecular conformational dynamics. *Nucleic Acids Res.*, 54(D1):D393–D401, 2026.

31. David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E. Mark, and Herman J. C. Berendsen. GROMACS: Fast, flexible, and free. *J. Comput. Chem.*, 26(16):1701–1718, 2005.

32. Jing Huang, Sarah Rauscher, Grzegorz Nawrocki, Ting Ran, Michael Feig, Bert L De Groot, Helmut Grubmüller, and Alexander D MacKerell. CHARMM36m: An improved force field for folded and intrinsically disordered proteins. *Nat. Methods*, 14(1):71–73, 2017.

33. J.Michael Word, Simon C. Lovell, Jane S. Richardson, and David C. Richardson. Asparagine and glutamine: Using hydrogen atom contacts in the choice of side-chain amide orientation. *J. Mol. Biol.*, 285(4):1735–1747, 1999.

34. Herman J. C. Berendsen, Johan P. M. Postma, Wilfred F. van Gunsteren, A. Dinola, and Jan R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81:3684–3690, 1984.

35. Giovanni Bussi, Davide Donadio, and Michele Parrinello. Canonical sampling through velocity rescaling. *J. Chem. Phys.*, 126(1):014101, 2007.

36. Mattia Bernetti and Giovanni Bussi. Pressure control using stochastic cell rescaling. *J. Chem. Phys.*, 153(11):114107, 2020.

37. Berk Hess, Henk Bekker, Herman J. C. Berendsen, and Johannes G. E. M. Fraaije. LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.*, 18(12):1463–1472, 1997.

38. Darrin M. York, Tom A. Darden, and Lee G. Pedersen. The effect of long-range electrostatic interactions in simulations of macromolecular crystals: A comparison of the Ewald and truncated list methods. *J. Chem. Phys.*, 99(10):8345–8348, 1993.

39. Richard Gowers, Max Linke, Jonathan Barnoud, Tyler Reddy, Manuel Melo, Sean Seyler, Jan Domański, David Dotson, Sébastien Buchoux, Ian Kenney, and Oliver Beckstein. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In *Python in Science Conference*, pages 98–105, Austin, Texas, 2016.

40. Robert T. McGibbon, Kyle A. Beauchamp, Matthew P. Harrigan, Christoph Klein, Jason M. Swails, Carlos X. Hernández, Christian R. Schwantes, Lee-Ping Wang, Thomas J. Lane, and Vijay S. Pande. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophys. J.*, 109(8):1528–1532, 2015.

41. Toshiko Ichiye and Martin Karplus. Collective motions in proteins: A covariance analysis of atomic fluctuations in molecular dynamics and normal mode simulations. *Proteins*, 11(3):205–217, 1991.

42. Oliver F. Lange and Helmut Grubmüller. Generalized correlation for biomolecular dynamics. *Proteins*, 62(4):1053–1061, 2006.

43. Yasaman Karami, Elodie Laine, and Alessandra Carbone. Dissecting protein architecture with communication blocks and communicating segment pairs. *BMC Bioinform.*, 17(S2):S13, 2016.

44. Sneha Bheemireddy, Roy González-Alemán, Emmanuelle Bignon, and Yasaman Karami. Communication Pathway Analysis within Protein-Nucleic Acid Complexes. *J. Chem. Theory Comput.*, 21(17):8255–8266, 2025.

45. Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

DrMab : Framework to Track Mutations of concern in Respiratory Viruses

Jérôme BOURRET[1,2,3], Marie-Anne RAMEIX-WELTI[1,3], and Frédéric LEMOINE[1,2,3]

1 Institut Pasteur, National Reference Center for Respiratory Viruses, Université Paris Cité, Paris F-75015, France

2 Institut Pasteur, Bioinformatics and Biostatistics Hub, Université Paris Cité, Paris F-75015, France

3 Institut Pasteur, Molecular Mechanisms of Multiplication of Pneumovirus, Université Versailles St-Quentin en Yvelines, Paris-Saclay INSERM UMR 1173 (2I), Assistance Publique des Hôpitaux de Paris, Paris, France

Corresponding Author: jerome.bourret@pasteur.fr

Keywords

Drug resistance, respiratory viruses, surveillance, epidemics, mutations, workflows.

1. Introduction

Respiratory viruses such as Influenza viruses and Respiratory Syncytial Virus (RSV) have long been major causes of hospitalizations due to acute respiratory infections, with SARS-CoV-2 emerging as a major global health threat in recent years. Tracking the emergence, evolution, and spread of these viruses is crucial for anticipating epidemic progression and implementing measures to mitigate public health impacts.

Recent progress in genomic surveillance has enabled the establishment of highly robust systems for monitoring epidemics. For instance, full-genome viral sequencing facilitates near real-time tracking of viral evolution through the construction and continuous updating of phylogenetic trees. These capabilities further support the development of standardized lineage nomenclatures for tracking viral clades of concern, detecting evolutionary events such as recombination and reassortment (in segmented viruses), and monitoring the emergence of concerning mutations.

These mutations of concern typically fall into several categories: host-adaptive mutations (e.g., facilitating transmission of influenza viruses from avian to human hosts), immune evasion mutations, or antiviral resistance mutations (conferring reduced susceptibility to therapeutic agents, often known as Drug Resistance Mutations or DRMs). Surveillance systems must enable the rapid and accurate detection of these mutations through routine genomic sequencing to facilitate timely public health response on an individual (e.g. for custom treatments) or a global scale (e.g. surveillance of mutational profile in the population). Crucially, monitoring must capture these variants whether they dominate viral populations or remain at low frequencies, as both contexts inform predictions regarding treatment outcomes and evolutionary trends.

In this context, bioinformatics tools play a crucial role. They must i) accurately analyze complete genome sequences from routine surveillance data (and integrate themselves easily in already existing workflows), ii) detect early indicators such as low-frequency mutations, and iii) rely on complete, flexible, and regularly updated mutation databases.

Several bioinformatics tools have been developed to identify mutations of concern, typically by aligning query sequences to reference genomes and matching the detected variations against curated catalogs of known mutations. However, most of these approaches fail to satisfy every requirement outlined above. Specifically, tools like `Flu/RSVSurver` (FluSurver, n.d.), `Sierra` (Tzou et al., 2022), and `BMA` (Salvatierra & Florez, 2016) operate exclusively on consensus sequences, precluding the analysis of low-frequency minor (intra-host) variants. Meanwhile, specialized tools including `SABRes` (Fong et al., 2023), `HerpesDRG` (Charles et al., 2024), and `DR_SEQUAN` (Garriga & Menéndez-Arias, 2006) are limited to specific viral species with restricted configurability.

To overcome these limitations, we developed DrMab, a comprehensive and flexible framework that analyzes viral whole-genome viral data (already assembled genomes or raw reads) to identify mutations of concern at the viral population level. To do so, it leverages a curated database of mutations or a user provided list of mutations.

2. DrMab framework

DrMab functionalities

DrMab comprises two core modules. The first is a manually curated and continuously updated database of mutations of concern for SARS-CoV-2 (based on the Cov-RDB database; Tzou et al., 2022), influenza viruses (based on the WHO guidelines ; WHO, n.d.), several avian influenza (list of mutations curated by experts and based on literature review) and RSV (based on the Virus French Resistance database ; Virus French Database, n.d.), reflecting current knowledge regarding variant functional impacts. In total, 806 mutations for 18 strains of viruses are recorded on our database. The second module consists of a Nextflow-based computational workflow that processes FASTQ, FASTA, and VCF inputs alongside configurable resources, including the mutation database and corresponding reference genomes.

The workflow is designed to limit manual intervention, and executes the following steps, depending on the kind of inputs: For FASTQ files, the pipeline first selects the most appropriate reference genome for each sample and maps the reads with minimap2 (Li, 2021). It then executes two concurrent variant-calling procedures: one to detect minor variants (intra-host variants) and another to generate a consensus sequence. The consensus - generated with iVar (Grubaugh et al., 2019) - is aligned with MAFFT (Katoh et al., 2019) to the reference coordinate system (against which database mutations are annotated) to identify major variants for comparison against the curated database. Simultaneously, minor variant coordinates - estimated with iVar - are converted to this standardized reference framework prior to database screening. When VCF files are provided, the workflow bypasses alignment and variant calling, proceeding directly to coordinate standardization and extraction of mutations of concern. In all cases, nucleotide sequences of all ORFs as defined in a reference gff annotation file (even when they are overlapping or subjected to frameshift) are translated into amino acids. This is done using a dedicated tool for minor variants and with goalign (Lemoine & Gascuel, 2021) for consensus sequences, and accounts for multiple minor variants occurring within individual codons and appropriately handles degenerate nucleotides codes. Figure 1 illustrates the simplified FASTQ processing subworkflow.

The DrMab workflow generates two primary outputs:

- A list of minor variants of concern: these are low frequency mutations
- A list of dominant variants of concern: these are present in the consensus sequence

Both types are classified as:

- "True" (the specific mutation of concern),
- "Positional" (mutations at the same site as a known variant but with a different amino acid), or
- "Potential" (inferred from degenerated codons, applicable only to dominant variants).

In these outputs, each entry includes additional descriptive metadata such as:

- The type of mutation (DRM, vaccine-induced immunity escape, host-adaptation), as described on the curated databases.
- The variant's predicted functional consequences (level of reduced sensitivity, modifications of cellular tropism, premature stop codon potentially making the protein non-functional, etc.).

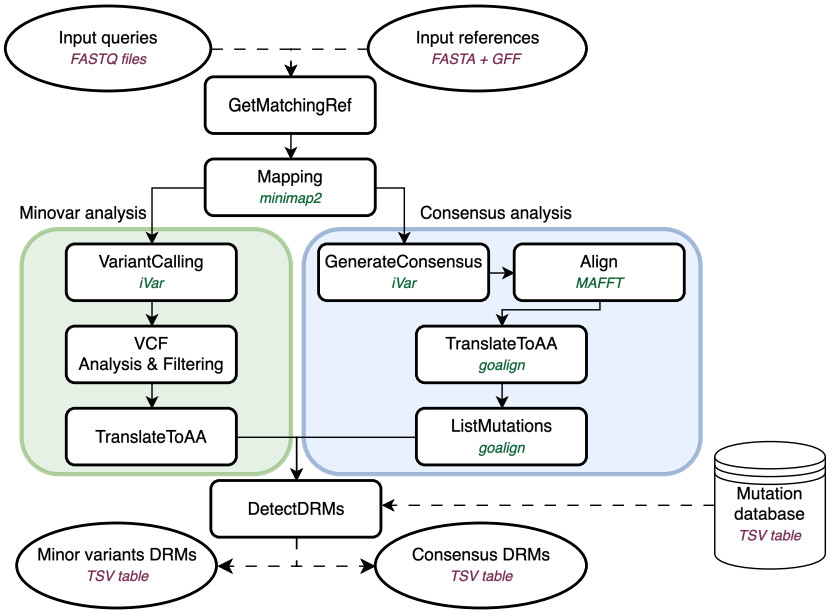

Figure 1 `DrMab` FASTQ Workflow.

`DrMab` implementation

The `DrMab` workflow is developed using Nextflow, a choice that guarantees reproducibility, scalability, and component reusability. By isolating each analytical step within dedicated software environments (containers), the pipeline ensures straightforward setup, deployment, and maintenance.

Designed with modularity in mind, `DrMab` offers distinct subworkflows tailored to specific input types (FASTA, FASTQ, and VCF) and includes dedicated pipelines for analyzing detected DRMs. It provides specialized analytical branches for RSV, SARS-CoV-2, and Influenza datasets, all of which utilize a common foundation of base modules to ensure consistency and reusability across workflows.

`DrMab` is publicly available at https://gitlab.pasteur.fr/cnrvir/pipelines/drmab/.

Together, these features establish `DrMab` as a versatile, modular platform for variant-of-concern analysis in the context of epidemic monitoring.

3. Use cases

`DrMab` is currently used by the National Reference Center for Respiratory Viruses to detect DRMs and other concerning mutations in influenza, RSV, and SARS-CoV-2 samples collected in France. We evaluated `DrMab` on two distinct datasets: 1) 386 French H1N1pdm samples collected during the 2024-2025 epidemic season, and 2) 858 French RSV samples from the 2025 POLYRES Cohort (Fourati et al., 2026). We finally compared the results obtained with DrMab and FLU/RSVSurver.

Analysis of the H1N1pdm dataset

The H1N1pdm dataset revealed rare and sporadic emergences of resistance mutations, with no clear spatial nor temporal pattern (Table 1). Most of these mutations (except H275Y) confer only mild or low resistance to treatments targeting NA or PA segments. Our findings show no evidence of fixation of these mutations, indicating that such mutational profiles were likely outcompeted and eliminated from viral populations due to fitness costs.

Table 1 H1N1pdm influenza samples from France's 2024-2025 season with detected DRMs in the NA or PA segments. The "Resistance" column specifies the affected antiviral treatment and its resistance level (in parentheses), while the "Mutation" column shows the frequency of each mutation in sequencing reads (as a percentage). Sample origins are listed by French administrative region. The bold line indicates the minor variant not detected by FLUSurver (written as H275X instead of H275Y).

ID (EPI_ISL)	Mutation (frequency)	Resistance against (level)	Location	Date
19850563	**NA:H275Y(0.46)**	**Oseltamivir, Peramivir(strong), zanamivir, laninamivir(low)**	**Basse-Normandie**	**2025-03-21**

19850562	NA:H275Y(1)	Oseltamivir, Peramivir (strong), zanamivir, laninamivir (low)	Basse-Normandie	2025-02-21
19669923	NA:I223T(1)	Oseltamivir (mild), Zanamivir, Peramivir, Laninamivir (low)	Île-de-France	2024-12-20
19690751	PA:I38V(1)	Baloxavir (low)	Nord-Pas-de-Calais	2024-12-24
19554791	NA:S247N(1)	Oseltamivir, Zanamivir, Peramivir, Laninamivir (low)	Alsace	2024-10-29
19655854	NA:S247N(1)	Oseltamivir, Zanamivir, Peramivir, Laninamivir (low)	Alsace	2024-12-02
19669949	NA:S247N(1)	Oseltamivir, Zanamivir, Peramivir, Laninamivir (low)	Nord-Pas-de-Calais	2024-12-17

Analysis of the RSV 2025 dataset

Regarding the RSV dataset, we aimed to evaluate the introduction of Nirsevimab in France by comparing mutational profiles between breakthrough samples (treated infants that are nonetheless infected by the virus ; n = 419) and untreated control samples (untreated infants ; n = 439) during the 2024–2025 epidemic season. Among breakthrough samples, `DrMab` identified DRMs in 2 RSV-A and 23 RSV-B samples, including 7 minor variants present at low frequencies (<50%). The lack of evolutionary clustering in these resistance patterns (no clear grouping by clade) suggests sporadic emergence (convergence) rather than lineage-driven propagation (Figure 2). Intriguingly, only one DRM was detected nearly twelve months post-treatment, indicating that while most resistance mutations are rapidly eliminated in the absence of continued selective pressure, rare instances of persistence may occur without achieving broader epidemic spread (Fourati et al., 2026).

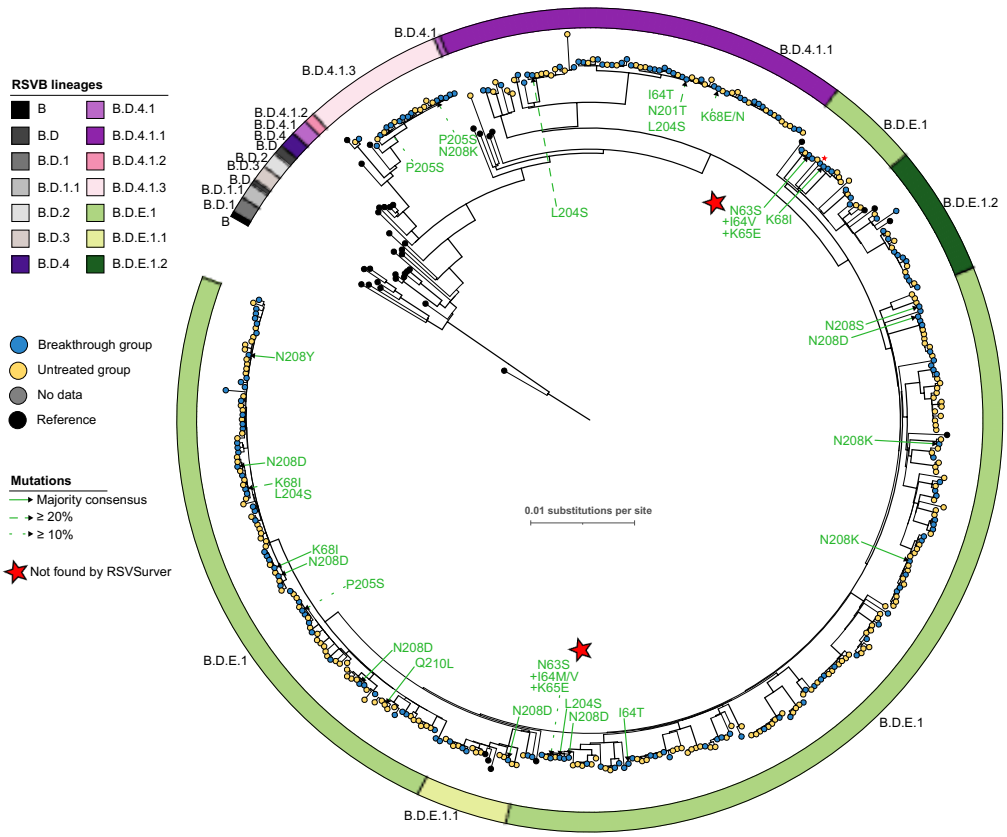

Figure 2 RSVB Phylogeny of RSVB samples from the POLYRES study. This phylogeny has been inferred from the full-length genome sequences with a Maximum-Likelihood approach (GTR model, bootstrap of 1000). Tip color indicates the breakthrough (blue), unexposed (yellow), or external sequences used as a context for this analysis (black). The colored ribbon around the tree indicates the RSV lineages to which the tips belong. Branches with a bootstrap value of 90% or higher are highlighted with a thicker line. Green lines pinpoint individuals with mutations of interest on epitope Φ of the F protein. Green dashed lines pinpoint individuals where those mutations are found in minor populations (>10% and >20% as represented in the figure). Resistance-associated substitutions co-occurring in the same viral population are separated with a « + » sign. Red stars show the co-occurring mutations not detected by RSVSurver. This phylogeny has been inferred with `IQ-TREE` (v2.2.5). The full version of this Figure can be found in Fourati et al., study (2026).

4. Conclusion

Altogether, these results show that `DrMab` is a robust and adaptable framework designed for routine and large-scale surveillance of clinically relevant mutations in major respiratory viruses, as well as for sporadic viral population / cohort analyses. `DrMab` requires few inputs, with only a predefined list of concerning mutations and a reference genome that can be both customized by the user, making it a generic framework adaptable to many viruses. Our validation across two datasets demonstrated its accuracy in detecting key mutations.

The National Reference Center for Respiratory Viruses now routinely uses `DrMab` to detect DRMs in influenza, RSV, and SARS-CoV-2 samples from France. Beyond standard surveillance, the tool is also applied in targeted scenarios, such as tracking resistance in treated patients.

Future development will focus on identifying co-occurring mutations within viral populations through haplotype estimation, as certain influenza DRMs exhibit enhanced resistance when combined (e.g., with tools such as `Haploflow`, `Virus-VG` or `SAVAGE` (Baaijens et al., 2017, 2019; Fritz et al., 2021). Additionally, we plan to include phylogenetic placements of query sequences on reference trees, enabling evolutionary interpretations of the detected variants. Finally, we plan to extend `DrMab` to support other viruses through updates of the database. `DrMab` is available as a standalone workflow for local or high-performance computing environments and will soon be accessible via a dedicated web platform.

References

Baaijens, J. A., Aabidine, A. Z. El, Rivals, E., & Schönhuth, A. (2017). De novo assembly of viral quasispecies using overlap graphs. *Genome Research*, *27*(5), 835–848. https://doi.org/10.1101/gr.215038.116

Baaijens, J. A., Van der Roest, B., Köster, J., Stougie, L., & Schönhuth, A. (2019). Full-length *de novo* viral quasispecies assembly through variation graph construction. *Bioinformatics*, *35*(24), 5086–5094. https://doi.org/10.1093/bioinformatics/btz443

Charles, O. J., Venturini, C., Goldstein, R. A., & Breuer, J. (2024). HerpesDRG: a comprehensive resource for human herpesvirus antiviral drug resistance genotyping. *BMC Bioinformatics*, *25*(1), 279. https://doi.org/10.1186/s12859-024-05885-5

FluSurver. (n.d.). *FluSurver website*. Http://Flusurver.Bii.a-Star.Edu.Sg.

Fong, W., Rockett, R. J., Agius, J. E., Chandra, S., Johnson-Mckinnon, J., Sim, E., Lam, C., Arnott, A., Gall, M., Draper, J., Maddocks, S., Chen, S., Kok, J., Dwyer, D., O'Sullivan, M., & Sintchenko, V. (2023). SABRes: in silico detection of drug resistance conferring mutations in subpopulations of SARS-CoV-2 genomes. *BMC Infectious Diseases*, *23*(1), 303. https://doi.org/10.1186/s12879-023-08236-6

Fourati, S., Reslan, A., Bourret, J., Casalegno, J.-S., Rahou, Y., Softic, L., Chollet, L., Trémeaux, P., Imbert-Marcille, B.-M., Pillet, S., Veyrenche, N., Boudet, A., Deroche, L., Burrel, S., Mouna PharmD, L., Cocherie, T., Schnuriger, A., Pronier, C., Handala PharmD, L., … Rameix Welti, M.-A. (2026). *Real-world emergence of nirsevimab resistance in RSV-B breakthrough infections on behalf of the POLYRES investigators**. https://ssrn.com/abstract=5427106

Fritz, A., Bremges, A., Deng, Z.-L., Lesker, T. R., Götting, J., Ganzenmueller, T., Sczyrba, A., Dilthey, A., Klawonn, F., & McHardy, A. C. (2021). Haploflow: strain-resolved de novo assembly of viral genomes. *Genome Biology*, *22*(1), 212. https://doi.org/10.1186/s13059-021-02426-8

Garriga, C., & Menéndez-Arias, L. (2006). DR_SEQAN: a PC/Windows-based software to evaluate drug resistance using human immunodeficiency virus type 1 genotypes. *BMC Infectious Diseases*, *6*, 44. https://doi.org/10.1186/1471-2334-6-44

Grubaugh, N. D., Gangavarapu, K., Quick, J., Matteson, N. L., De Jesus, J. G., Main, B. J., Tan, A. L., Paul, L. M., Brackney, D. E., Grewal, S., Gurfield, N., Van Rompay, K. K. A., Isern, S., Michael, S. F., Coffey, L. L., Loman, N. J., & Andersen, K. G. (2019). An amplicon-based sequencing framework for accurately measuring intrahost virus diversity using PrimalSeq and iVar. *Genome Biology*, *20*(1), 8. https://doi.org/10.1186/s13059-018-1618-7

Katoh, K., Rozewicki, J., & Yamada, K. D. (2019). MAFFT online service: multiple sequence alignment, interactive sequence choice and visualization. *Briefings in Bioinformatics*, *20*(4), 1160–1166. https://doi.org/10.1093/bib/bbx108

Lemoine, F., & Gascuel, O. (2021). Gotree/Goalign: toolkit and Go API to facilitate the development of phylogenetic workflows. *NAR Genomics and Bioinformatics*, *3*(3). https://doi.org/10.1093/nargab/lqab075

Li, H. (2021). New strategies to improve minimap2 alignment accuracy. *Bioinformatics*, *37*(23), 4572–4574. https://doi.org/10.1093/bioinformatics/btab705

Salvatierra, K., & Florez, H. (2016). Biomedical Mutation Analysis (BMA): A software tool for analyzing mutations associated with antiviral resistance. *F1000Research*, *5*, 1141. https://doi.org/10.12688/f1000research.8740.2

Tzou, P. L., Tao, K., Pond, S. L. K., & Shafer, R. W. (2022). Coronavirus Resistance Database (CoV-RDB): SARS-CoV-2 susceptibility to monoclonal antibodies, convalescent plasma, and plasma from vaccinated persons. *PloS One*, *17*(3), e0261045. https://doi.org/10.1371/journal.pone.0261045

Virus French Database. (n.d.). *Virus French Database - RSV resistance list*. Https://Virusfrenchresistance.Org/Virus-French-Resistance-Rsv/.

WHO. (n.d.). *WHO - Influenza Antiviral susceptibility*. Https://Www.Who.Int/Teams/Global-Influenza-Programme/Laboratory-Network/Quality-Assurance/Antiviral-Susceptibility-Influenza.

Fast and robust graph construction from KEGG metabolic and genomic data

Florent Cabret[1], Ronan Bocquillon[1], and Emmanuel Néron[1]

https://doi.org/10.5281/zenodo.19036695

Abstract

The Kyoto Encyclopedia of Genes and Genomes (KEGG) enables the modelling of biological systems as integrated graphs, where directed metabolic networks capture reaction flows and undirected genomic graphs encode chromosomal proximity of enzyme coding genes. However, constructing these graphs at the scale of complete bacterial organisms remains challenging due to two limitations: web based API access is constrained by rate limits triggering temporary IP blocks, and existing tools are typically restricted to the scale of pathway analysis. Here we present a framework for the fast and robust construction of whole organism metabolic and genomic graphs from KEGG data. Our approach combines a concurrency model with proxy rotation to overcome web access bottlenecks, specialised parsers for tab-delimited and KGML formats, and an embedded analytical database for efficient storage. For graph construction, we introduce optimised methods that handle real world data complexities, including multiple genomic intervals per gene and uncertain boundary annotations. This enables, to the best of our knowledge, the first systematic construction of integrated graphs at the scale of complete bacterial genomes and entire metabolisms. By overcoming previous scalability barriers, our framework provides the foundation for applying advanced trail finding algorithms to reveal how metabolic function and genomic organisation co-evolve across the bacterial domain, moving beyond isolated pathways towards whole-organism comparative analysis.

Keywords: Systems Biology, Metabolic Networks and Pathways, Genome Bacterial, Data Collection, Graph Theory

[1]Laboratoire d'Informatique Fondamentale et Appliquée de Tours, 64 Avenue Jean Portalis, 37200 Tours, France

Correspondence
ronan.bocquillon@univ-tours.fr,emmanuel.neron@univ-tours.fr

Introduction

The Kyoto Encyclopedia of Genes and Genomes (KEGG) was established in 1995 as a knowledge base for the systematic analysis of gene functions, with the ambitious goal of linking genomic information to higher order functional insights (Ogata et al., 1999). From its inception, KEGG has been structured around three core databases: the PATHWAY database for the graphical representation of biochemical pathways, the GENES database for consistently annotated gene catalogs from complete genomes, and the LIGAND database for chemical compounds and enzymes. By integrating genomic, chemical, and systemic functional information, KEGG serves as an indispensable reference for interpreting sequence data and understanding cellular processes, providing a foundational resource for applications ranging from the reconstruction of metabolic pathways to the analysis of gene expression profiles.

The utility of KEGG extends far beyond simple data retrieval. As demonstrated in studies such as Zaharia et al., 2019, KEGG data enables sophisticated integrative analyses that simultaneously consider metabolic pathways and genomic context. Their CoMetGeNe pipeline leverages KEGG's rich annotations to identify conserved metabolic and genomic patterns across multiple species, revealing how evolutionary pressures shape the organization of enzyme coding genes and their corresponding reactions. Such approaches exemplify the growing recognition that biological function emerges not from isolated components but from the complex interplay between metabolic networks and genomic architecture.

Accessing KEGG data programmatically has been facilitated through various software tools over the years. The KEGG API provides RESTful web services that allow automated retrieval of database entries, forming the backbone for numerous analysis pipelines. Early tools like KEGGgraph (Zhang and Wiemann, 2009) introduced graph based approaches to pathway analysis in R, capturing pathway topology for the first time. Specialized query systems such as IsoKEGG (Sultana et al., 2010, 2014) further expanded the possibilities by enabling logic based pathway queries using subgraph isomorphism, allowing researchers to retrieve pathways based on structural patterns. Integration tools like BiKEGG (Jamialahmadi et al., 2016) have bridged KEGG with other databases such as BiGG, facilitating cross database analyses. More recently, packages such as kegg_pull (Huckvale and Moseley, 2023) have improved accessibility through robust Python implementations that handle multiprocessing and fault tolerant retrieval. However, these tools share fundamental limitations inherent to web-based data access, namely rate limits that can result in temporary IP blocks, combined with a lack of built-in proxy support for large-scale concurrent requests. While kegg_pull's authors provide recommendations for sleep times to mitigate throttling, such measures necessarily trade speed for reliability, and the underlying constraints of operating within KEGG's API limits remain unchanged.

Despite these persistent constraints on data access, a particularly powerful paradigm that has emerged from KEGG's pathway representation is the modeling of biological systems as graphs. In this framework, metabolic pathways are represented as directed graphs where vertices correspond to enzymes (or reactions) and arcs represent substrate-product relationships between sequential reactions. Concurrently, genomic context can be modeled as an undirected graph where edges connect genes that are physically close on the chromosome. This dual graph representation (Zaharia et al., 2019) captures the fundamental insight that enzymes catalyzing successive reactions are often encoded by neighboring genes, reflecting evolutionary optimization of co-expression and functional coordination. The graph abstraction enables rigorous computational approaches: searching for maximum span trails in the metabolic graph whose vertex sets induce a single connected subgraph in the genomic graph reveals functionally coherent modules that transcend traditional operon definitions. These trails can subsequently be used for cross-species comparison, by grouping them based on shared reactions or conserved gene neighborhoods, to reconstruct phylogenetic relationships and trace the evolutionary history of metabolic pathways.

Despite the conceptual elegance of graph based representations, existing implementations have been constrained by computational complexity and scope. Early algorithms such as HNET (Zaharia et al., 2019) demonstrated the feasibility of identifying conserved metabolic genomic patterns but were limited to pathway by pathway analysis. Subsequent work by Ahmed Sidi,

2022; Ahmed Sidi et al., 2025 developed more efficient exact methods based on integer linear programming and constraint programming, demonstrating improved scalability on individual pathway instances. However, the $\mathcal{NP}$-hard complexity of the underlying problems imposed fundamental scalability limitations, preventing exact methods from extending beyond isolated pathway boundaries to encompass entire metabolic networks. Moreover, the reliance on KEGG's predefined pathway boundaries potentially fragments biologically meaningful patterns that span multiple pathways.

Recent algorithmic advances by Cabret et al., 2025 have improved this situation. Their binary integer programming model for finding maximum coverage trails in metabolic networks subject to genomic constraints achieves computational speedups of nearly 3000 times compared to previous approaches, allowing it to scale to networks with thousands of vertices. This development now makes it possible to begin systematic construction and analysis of integrated metabolic genomic graphs at the scale of complete bacterial genomes and full metabolisms. By considering all reactions and genes simultaneously, rather than focusing on individual pathways, this approach may help uncover broader organizational principles and evolutionary patterns that previous computational limitations made difficult to access. Additionally, their approach to generating biologically realistic benchmark instances through gamma-distributed degree sampling provides a way to create validation data that reflects true biological network properties. This offers an alternative to relying solely on simplified random graph models.

The convergence of rich KEGG data resources, sophisticated graph theoretic modeling frameworks, and increasingly more powerful algorithmic solvers creates an unprecedented opportunity to understand how metabolic function and genomic organization co-evolve across the bacterial domain. Realizing this opportunity, however, requires robust infrastructure for constructing integrated metabolic genomic graphs from KEGG data that can leverage these new computational capabilities. In this paper, we present a novel approach that addresses this need through two key contributions:

- Section 2 details a faster and more robust methodology for retrieving KEGG data, which overcomes the bottlenecks of web-based access through a concurrency model with proxy rotation and efficient parsing;
- then Section 3 introduces a graph construction framework operating at the scale of whole metabolisms and complete genomes, introducing optimized methods for building both directed metabolic graphs and undirected genomic graphs that handle real-world data complexities.

By moving beyond previous pathway-centric methods, this combined framework enables the application of advanced trail finding algorithms to complete bacterial systems.

Retrieving data from KEGG

To enable the scalable and robust construction of integrated metabolic genomic graphs at the scale of complete bacteria, our data retrieval pipeline is engineered as a cohesive system of four interdependent components. The process begins by programmatically contacting the KEGG REST API through a dedicated interface, after which the returned data, ranging from tab-delimited records to complex XML based KGML files, is parsed using specialized libraries tailored to each format. To overcome the inherent limitations of web-based access, including rate limits and the risk of temporary IP blocks, a sophisticated concurrency model manages thousands of parallel requests through a lock-free proxy rotation scheme, ensuring both speed and reliability. Finally, all retrieved and parsed data is efficiently stored in an embedded analytical database, where a carefully designed schema and bulk insertion strategy preserve referential integrity while maximizing throughput. Together, these elements form the foundation for acquiring the comprehensive, high quality datasets required for downstream graph analysis.

Concurrency model

The large number of independent web requests required to retrieve complete KEGG data for thousands of organisms makes concurrency essential. We therefore implement a thread pool to

manage these requests. Since most of the time in each request is spent waiting for a response from the KEGG server, the pool is deliberately oversized, and we create 25 times the number of logical processor threads. This over-subscription ensures that while some threads are blocked on I/O, others can continue making progress, thereby maximising network utilisation.

A critical challenge when issuing many requests to the same public API is the risk of temporary IP blocks. To mitigate this, we obtain before data retrieval begins a list of public proxies drawn from monosans/proxy-list, an hourly updated repository of freely available proxies scraped from diverse sources and validated for reliability. Each proxy is then tested for reachability, requiring a successful connection attempt to the KEGG website within 10 seconds and allowing up to three retries. Proxies that pass this filter are stored in a global list alongside a flag indicating whether they are currently in use and an optional timestamp of the last failure. This list is fixed for the duration of the run and provides the pool of proxies used for all requests.

Access to this shared list must be efficient and must not become a bottleneck. We therefore implement a lock-free proxy rotation scheme. Lock-free algorithms guarantee system-wide progress even if individual threads are delayed, and they avoid the overhead of kernel-space locking. Specifically we use the test and test-and-set idiom. A thread repeatedly reads the state of a randomly chosen proxy; only if the proxy appears free does it attempt to claim it with an atomic exchange. If the exchange succeeds, the thread proceeds to use the proxy. Should the request later fail with an HTTP status of 400 or higher, which indicates a client or server error, the proxy is marked as temporarily unavailable and a 5 second timeout is recorded. This timeout value follows the recommendation of kegg_pull to avoid hammering a faulty proxy. Because the entire selection loop uses only relaxed-order loads and a single acquire-release exchange, contention on the proxy list is negligible.

Each request itself follows a robust retry policy. After a proxy has been acquired, the thread issues a GET request with a generous one minute timeout. If the request times out or returns an error status, it is retried up to three times. A failed request does not immediately discard the proxy; instead the proxy is temporarily penalised (the 5 second timeout) so that other proxies can be tried, while a faulty proxy is given time to recover.

Retrieved responses must be stored in a relational database while preserving referential integrity. Because foreign keys must point to existing primary keys, we enforce a strict ordering of insertions using the bulk synchronous parallel (BSP) model. The workflow is divided into super-steps: first an organism is inserted, then all genes and positions, then enzyme annotations, and finally the metabolic graph edges (see Figure 1). A barrier synchronises all worker threads at the end of each super-step. The barrier itself is implemented with the low-level atomic_wait and notify_all primitives. Threads that finish early start the next super-step up to the point of committing to the database; if the remaining threads have not yet reached the barrier, they first spin briefly before finally parking on the atomic flag. This hybrid approach avoids excessive context switching when the last thread is expected to arrive quickly, while still allowing long waits to block efficiently. This design guarantees that no thread ever tries to insert a row that references a still missing primary key.

To further reduce contention, each thread maintains its own database connection in thread-local storage. Instead of issuing individual INSERT statements (which would require repeated round trips and serialisation), threads accumulate rows in memory and commit them in bulk transactions. This approach minimises lock contention on the database and significantly cuts the number of inter-thread synchronisation events. Only when a barrier is reached must the threads ensure that all their buffered writes have been applied; the atomic wait barrier provides exactly that guarantee.

Contacting the API

To programmatically access the KEGG REST API, we are using libcurl, a free and portable client-side URL transfer library. It supports a wide range of protocols including HTTPS, which is required for all of the endpoints, and provides robust features such as persistent connections, thread safety, and comprehensive error handling. Its ability to seamlessly handle HTTP and SOCKS proxies, including those requiring authentication, is particularly valuable in institutional

and high-performance computing environments where direct external access may be restricted. By leveraging libcurl, our workflow can reliably retrieve genomic, pathway, and chemical data from KEGG while automatically routing traffic through necessary proxy servers, ensuring both reproducibility and resilience in diverse network settings.

Parsing CSV and XML

The KEGG API returns data in several distinct formats depending on the operation performed. Most operations including list, find, conv, and link return tab-delimited text that can be parsed line by line, while pathway data retrieved via the get operation with the kgml option is provided in the XML-based KGML (KEGG Markup Language) format. These two structural paradigms require different parsing strategies: the former demands efficient, type-safe handling of flat records, while the latter requires navigating hierarchical relationships between pathway elements. To address these requirements, we employ two specialized C++ libraries. For tab-delimited responses, we leverage the PEG based parser combinator library lexy to construct declarative grammars that map directly onto data structures. For KGML, we utilize the DOM interface provided by pugixml to traverse and extract pathway components.

Parsing Expression Grammars (PEGs) provide a formal and unambiguous method for describing machine oriented syntax using ordered choice operators (Ford, 2004). This paradigm was notably adopted by the CPython interpreter in Python 3.9 (PEP 617) to lift the LL(1) restrictions and simplify grammar maintenance. The practical advantages of this approach are now being demonstrated in production database systems, with the recently released DuckDB v1.5.0 adopting a PEG based parser to enable runtime extensibility and deliver significantly improved error messages and auto-completion. This validation in a high-performance setting reinforces our choice of a PEG based solution for the KEGG parsing task. To process KEGG's tab-delimited outputs, we implement this approach using lexy, a C++ parser combinator library. Its expressive domain-specific language embeds grammars within the code, mirroring the structure of handwritten recursive descent parsers while providing explicit control over backtracking through branch conditions. This design avoids the performance pitfalls common in traditional PEG implementations. By constructing specialized grammars, we ensure type-safe and zero-copy parsing, complete with robust error recovery to handle the irregular formatting sometimes found in biological data dumps.

To process KGML outputs, we employ pugixml, a lightweight and high performance XML parsing library. This format encodes pathway information in a hierarchical structure that defines elements such as genes, compounds, reactions, and their relations. The library's DOM interface provides intuitive traversal methods, enabling efficient navigation of the KGML structure and ensuring reliable data extraction for subsequent analysis through the retrieval of pathway components with minimal overhead.

Database management system

DuckDB is a fast analytical, in-process, open-source and portable database system. Following the paradigm popularized by SQLite, it operates embedded within a host application, eliminating external dependencies and the overhead of inter-process communication. Its architecture is specifically optimized for Online Analytical Processing (OLAP) workloads through a columnar-vectorized query execution engine, which processes data in large batches rather than row-by-row to accelerate complex aggregations and joins on massive datasets. This combination of embedded simplicity and analytical performance makes DuckDB particularly well suited for efficient data analysis within the application itself. To support this analysis, the database is structured according to the entity-relationship model illustrated in Figure 1. Such a unified storage could also benefit tools like KEGG spider (Antonov et al., 2008), which builds a global gene metabolic network by integrating all the pathways, enabling the interpretation of gene lists in a pathway-free context. Rather than treating each pathway as an independent unit, it constructs a unified network where genes are connected if their associated reactions share common compounds. It then infers network models from an input gene list by progressively connecting genes at increasing distances and assesses the statistical significance of these models through Monte Carlo simulations. This

approach offers an alternative to the directed metabolic graph construction presented later in this paper.

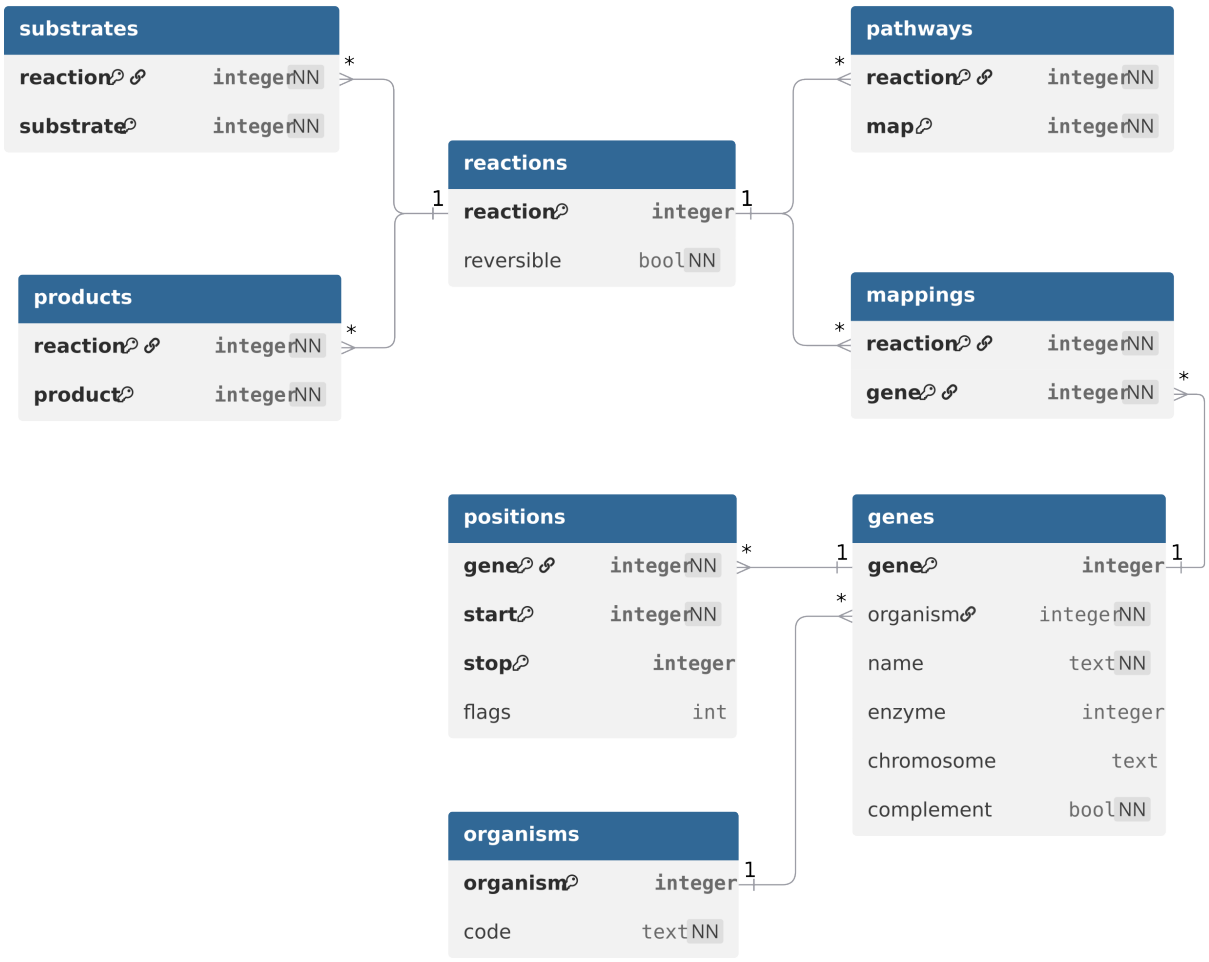

Figure 1 – Entity-relationship diagram of the database

Numerical experiments

Retrieving complete KEGG data for 1000 bacterial organisms with our approach requires approximately one hour of total runtime, with a variance of about ± 15 minutes attributable to the heterogeneous performance of public proxies (since all threads must synchronise at each super-step, a slower proxy can delay every worker). In practice, the number of proxies can be reduced to avoid placing an excessive load on the KEGG server. For smaller analyses involving at most 100 organisms, the parallel portion of the workload is substantially reduced. Amdahl's law (Amdahl, 1967) then predicts that the fixed overheads, notably the two minutes needed for initial proxy filtering, dominate the total time, making the concurrent strategy less beneficial than a simpler sequential retrieval.

Regarding long term robustness against changes in KEGG's access policies, our pipeline is designed to accommodate progressive hardening of the API. Rate limits and temporary IP blocks are already handled by proxy rotation and retries. Should KEGG introduce more advanced challenges such as CAPTCHAs or JavaScript based access gates, these can be integrated by adding a headless browser automation layer coupled with a CAPTCHA solving service. While such extensions would increase latency, they would preserve the overall architecture and concurrency model. For non interactive bulk access, KEGG also provides a paid FTP subscription that delivers flat files; our parser components (PEG for tabular data, pugixml for KGML) can ingest those offline files unchanged, bypassing API restrictions entirely. Thus, the framework is not locked into a

single acquisition strategy but can be adapted to alternative data sources or evolving server policies through interchangeable backend modules.

In terms of data quality, our pipeline issues more API calls than the method of Zaharia et al., 2019 because it fetches complete, fine-grained data for every organism instead of relying on global association tables. The additional calls are the price paid for obtaining precise reaction-enzyme associations and handling complex genomic annotations, which the earlier approach approximated or omitted.

Construction of the graphs

The core of the analysis framework lies in constructing two complementary graph representations from the retrieved KEGG data: a directed metabolic graph capturing the flow of substrates and products between reactions, and an undirected genomic graph encoding the chromosomal proximity of the associated enzyme-coding genes. These constructions follow the foundational concepts introduced by Zaharia et al., 2019, which enable the integrated analysis of metabolic and genomic organization. To achieve the necessary performance for organism scale analysis, we have developed optimized variants of these foundational constructions. The following subsections describe each graph construction in detail, using side by side comparison with the original methods to highlight how our approach overcomes previous scalability limitations.

Metabolic graph

The metabolic graph is a directed graph that models the flow of metabolites through a sequence of reactions. In Zaharia et al., 2019, its construction begins by representing both reactions and chemical compounds as nodes in a bipartite graph. Directed edges are placed from substrate compounds to the reactions they participate in, and from each reaction to its product compounds. The final directed graph is obtained by projecting this bipartite structure onto the set of reaction nodes. In this projection, two reactions become connected by a directed arc if a product of the first reaction serves as a substrate for the second. This results in a pathway centric directed graph where arcs trace feasible metabolic sequences, abstracting away the underlying compounds. KEGG provides a reversible flag as a boolean attribute for each reaction. For reactions marked as reversible, the projection naturally creates arcs in both directions because each compound is considered both a substrate and a product. Thus, two reactions linked via a reversible reaction may have arcs in both directions after projection.

In this work, we adopt a more efficient construction that bypasses the explicit bipartite graph and its projection. Instead, we maintain two hash maps: one that associates each compound with the set of reactions that produce it, and another that associates each compound with the set of reactions that consume it. For every compound, we then iterate over all producer-consumer pairs and directly add a directed arc from the producer to the consumer whenever the product of the former serves as a substrate of the latter. This approach eliminates the storage of intermediate bipartite edges and the overhead of a separate projection step. During arc creation, we also query the enzymes associated with each reaction and discard any reaction that lacks an enzyme annotation, ensuring that only reactions with genomic support are retained in the final graph. The resulting directed graph retains the same biochemical interpretation, arcs represent feasible metabolic transitions, but is constructed in a single pass with reduced memory footprint and lower time complexity, making it particularly suitable for large-scale metabolic network analysis.

Genomic graph

The genomic graph is an undirected graph that captures the spatial proximity of genes encoding the enzymes of the corresponding reactions. Zaharia et al., 2019 first build this graph by linking consecutive genes on each chromosome, forming a linear chain. For circular bacterial chromosomes, an additional edge connects the first and last gene to correctly reflect the circular topology. By default, the graph is constructed separately for genes on the forward and reverse strands, reflecting the biological observation that functionally related genes tend to reside on the same strand. This behavior can be modified to consider genes on both strands together when

a more permissive neighborhood definition is desired. To allow for gaps between functionally related genes, the shortest path distances in this initial gene graph are computed. If two genes are separated by at most a specified number δ of intermediate genes along the chromosome, an edge is added directly between them (typically, $\delta \leq 3$). Reactions are then mapped to the genes that encode their enzymes. Two reactions are linked by an undirected edge precisely when their associated genes are neighbors in the graph. This genomic graph thus encodes the condition that the genes of a metabolic trail must lie in a genomically coherent cluster.

In this work, we employ an Order Statistic Tree using as the base data structure the Logarithmic Binary Search Tree introduced by Roura, 2001. In this structure, the size of the subtree stored at each node serves the dual purpose of both maintaining the balancing property and enabling efficient rank queries. Each gene can be associated with multiple genomic intervals due to the presence of introns or uncertainties arising from high-throughput sequencing errors. In our database schema (see Figure 1), a gene can thus be associated with several position entries. Other complications include entries where the stop value is missing, in which case we set it equal to the start value. Additionally, a flag field, which we currently ignore, might indicate that the true start or stop lies outside the reported values.

The positions of all genes are inserted into the tree ordered by their start value. The data structure provides logarithmic time rank queries, which we use to compute genomic distances efficiently. For a given pair of genes, we first compute a new interval spanning from the smallest start value to the largest stop value among all intervals belonging to that gene. If these new intervals don't intersect, the genomic distance between the two genes is simply the distance between them. This value can be obtained directly using rank queries on the tree without examining individual intervals.

When they overlap, a more detailed analysis is required. We apply a sweep line algorithm on the sorted list of all interval endpoints belonging to the two genes. The algorithm maintains two pointers, each initially pointing to the first interval of the respective gene. At each step, it compares the current intervals: if they intersect, the distance is zero; otherwise, it advances the pointer corresponding to the interval with the smaller start value. This process continues until either an intersection is found or all intervals have been processed, yielding the minimal distance between any two positions belonging to the two genes. By leveraging the Order Statistic Tree for initial sorting and span checks, and the sweep line for overlapping cases, we achieve an efficient and accurate computation of gene neighbourhood distances even in the presence of multiple position intervals per gene.

It is worth noting that the genes encoding enzymes represent only a small subset of all coding genes in a genome. This specificity makes the construction of the genomic graph more computationally manageable and focuses the analysis on biologically relevant functional units. Compared to the approach of Zaharia et al., 2019, which does not account for the complexities introduced by multiple intervals per gene, missing stop values, or flag attributes indicating uncertain boundaries, our method is designed to handle these real-world data imperfections robustly. By efficiently computing distances between genes even in the presence of such complications, our approach provides a reliable foundation for downstream analyses. This capability could prove particularly useful for trail comparison if the need for less strict genomic constraints were to arise, thereby supporting scalable whole genome analyses.

To go further, the genomic graph, which captures chromosomal proximity between enzyme-coding genes, could be refined by integrating spatial proximity data derived from chromosome conformation capture techniques such as Hi-C. The contact matrix used by Gao et al., 2024 provides genome-wide interaction frequencies between genomic loci, revealing that genes physically close in the three-dimensional space of the nucleoid are more likely to be functionally associated, even when they are far apart on the linear chromosome. By incorporating these spatial interaction frequencies as additional edges in the genomic graph, the resulting network could better reflect the functional organisation of the genome, complementing our method. Moreover, manual annotations remain valuable for correcting annotation errors, resolving ambiguous gene boundaries, or incorporating experimentally validated interactions that may not be captured by high-throughput data.

Numerical experiments

To illustrate the scalability of our graph construction pipeline (see Table 1), we built metabolic and genomic graphs for six bacterial species with diverse number of reactions: the reduced genome of the secondary endosymbiont of *Trabutina mannipara* (senm), the *Blochmannia* endosymbiont of *Camponotus (Colobopsis) obliquus 757* (ben), the porcine pathogen *Glaesserella parasuis SH03* (hpas), the marine bacteria *Vibrio campbellii ATCC BAA-1116* (vac), the model organism *Escherichia coli K-12 MG1655* (eco), and the opportunistic pathogen *Klebsiella grimontii* (kgr). Construction of both graphs from the retrieved KEGG data scales with the organism complexity and consistently takes under 30 seconds. While the six species studied here span a wide range of reaction counts, the average across all bacteria is 946. The experiments confirm that finding a maximum span trail in these graphs (*i.e.*, at the scale of a complete bacterial organism) is tractable, even if it requires more time than expected. In contrast, Cabret et al., 2025 solve this same problem on a set of biologically realistic synthetic graphs of 1060 vertices in less than 3 seconds on average, and they obtain longer trails (36 reactions on average). These discrepancies highlight the computational challenge of the underlying combinatorial problem, as well as the difficulties to create an accurate model of synthetic graphs.

Bacteria name		senm	ben	hpas	vca	eco	kgr
Number of reactions		26	340	647	958	1130	1269
Metabolic graph	order	17	309	594	863	1019	1161
	size	22	538	1484	2737	3578	4071
Genomic graph	size	26	1377	2091	2489	2620	2899
Construction time (s)		0.3	4	9	15	22	26
Maximum span trail		5	10	25	18	12	12
Resolution time (s)		0.05	2	7	29	83	89

Table 1 – Statistics for different bacterial species

Conclusion

We have presented a framework for constructing integrated metabolic and genomic graphs from KEGG data at the scale of complete bacterial organisms. Our approach addresses the bottlenecks of web-based data retrieval through a concurrency model with proxy rotation. Our graph construction pipeline leverages two specialized data structures: hash maps for efficient metabolic graph assembly and order statistic trees for genomic proximity computations. To the best of our knowledge, this enables the first systematic construction of whole organism graphs that were previously limited to pathway by pathway analysis.

With these organism scale graphs and the trails they support, the next challenge becomes comparative analysis across species. Zaharia et al., 2019 provided an elegant method for identifying conserved patterns. However, their approach reintroduces a segregation between reactions and genes during comparison: trails are treated as reaction sets, temporarily decoupling the metabolic and genomic dimensions that the model itself integrates.

Although our framework is presented using KEGG, the modular design of its components (the concurrency model with proxy rotation, the PEG based tabular parser, the XML parser, and the embedded analytical database) makes them readily adaptable to other biological databases that expose REST APIs or provide structured flat files. For instance, replacing the KEGG API endpoints with those of any alternative data source would require only a thin configuration layer, namely URL templates and response grammar definitions. Similarly, the graph construction algorithms operate on generic reaction compound and gene interval data models, and can therefore be applied to any metabolic and genomic dataset that adheres to the same relational schema. This generality is explicitly supported by the use of DuckDB, which allows the user to load alternative data sources into the same table structure.

The availability of whole organism graphs now calls for comparison methods that preserve this duality. We need approaches that identify conserved substructures simultaneously in both metabolic and genomic dimensions. Such methods would reveal not only which reaction sequences persist across evolution, but also how their underlying genomic organization is maintained or rearranged. Developing these methods represents the next frontier in understanding the co-evolution of metabolism and genome architecture across the bacterial domain.

Fundings

The authors declare that they have received no specific funding for this study.

Conflict of interest disclosure

The authors declare that they comply with the PCI rule of having no financial conflicts of interest in relation to the content of the article.

Data, script, code, and supplementary information availability

Data, script and codes are available online (https://doi.org/10.5281/zenodo.19025626; Cabret et al., 2026).

References

Ahmed Sidi ML (2022). Nouvelles approches informatiques et mathématiques pour la résolution de problèmes biologiques. PhD thesis. Université de Tours - LIFAT. URL: https://hal.science/tel-03807522.

Ahmed Sidi ML, Bocquillon R, Cabret F, Mohamed Babou H, Dhib C, Néron E, Soukhal A, Nanne MF (2025). Constraint Programming Approaches for Finding Conserved Metabolic and Genomic Patterns. *Computers & Operations Research* **183**, 107166. https://doi.org/10.1016/j.cor.2025.107166.

Amdahl GM (1967). Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. In: *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*. AFIPS '67 (Spring). New York, NY, USA: Association for Computing Machinery, pp. 483–485. https://doi.org/10.1145/1465482.1465560.

Antonov AV, Dietmann S, Mewes HW (2008). KEGG Spider: Interpretation of Genomics Data in the Context of the Global Gene Metabolic Network. *Genome Biology* **9**, R179. https://doi.org/10.1186/gb-2008-9-12-r179.

Cabret F, Bocquillon R, Neron E (2025). Towards Mining Neighbourhood Patterns by Crossing the Metabolic and Genomic Networks at the Organism Level. Unpublished manuscript.

Cabret F, Bocquillon R, Neron E (2026). *KEGG Extractor: A Toolkit for Data Retrieval and Construction of Whole-Organism Metabolic and Genomic Graphs*. Zenodo. https://doi.org/10.5281/zenodo.19025627.

Ford B (2004). Parsing Expression Grammars: A Recognition-Based Syntactic Foundation. In: *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '04. New York, NY, USA: Association for Computing Machinery, pp. 111–122. https://doi.org/10.1145/964001.964011.

Gao Y, Ma B, Xu Q, Peng Y, Gong H, Guan A, Hua K, Langford PR, Jin H, Luo R (2024). Spatial Proximity and Gene Function: A New Dimension in Prokaryotic Gene Association Network Analysis with 3D-GeneNet. *Briefings in Bioinformatics* **25**, bbae320. https://doi.org/10.1093/bib/bbae320.

Huckvale E, Moseley HNB (2023). Kegg_pull: A Software Package for the RESTful Access and Pulling from the Kyoto Encyclopedia of Gene and Genomes. *BMC Bioinformatics* **24**, 78. https://doi.org/10.1186/s12859-023-05208-0.

Jamialahmadi O, Motamedian E, Hashemi-Najafabadi S (2016). BiKEGG: A COBRA Toolbox Extension for Bridging the BiGG and KEGG Databases†. *Molecular BioSystems(MBS)* **12**, 3459–3466. https://doi.org/10.1039/C6MB00532B.

Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M (1999). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* **27**, 29–34. https://doi.org/10.1093/nar/27.1.29.

Roura S (2001). A New Method for Balancing Binary Search Trees. In: *Automata, Languages and Programming*. Ed. by Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen. Berlin, Heidelberg: Springer, pp. 469–480. https://doi.org/10.1007/3-540-48224-5_39.

Sultana KZ, Bhattacharjee A, Jamil H (2010). IsoKEGG: A Logic Based System for Querying Biological Pathways in KEGG. In: *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 626–631. https://doi.org/10.1109/BIBM.2010.5706642.

Sultana KZ, Bhattacharjee A, Jamil H (2014). Querying KEGG Pathways in Logic. *International Journal of Data Mining and Bioinformatics* **9**, 1–21. https://doi.org/10.1504/IJDMB.2014.057772.

Zaharia A, Labedan B, Froidevaux C, Denise A (2019). CoMetGeNe: Mining Conserved Neighborhood Patterns in Metabolic and Genomic Contexts. *BMC Bioinformatics* **20**, 19. https://doi.org/10.1186/s12859-018-2542-2.

Zhang JD, Wiemann S (2009). KEGGgraph: A Graph Approach to KEGG PATHWAY in R and Bioconductor. *Bioinformatics* **25**, 1470–1471. https://doi.org/10.1093/bioinformatics/btp167.

Holograph: a generic RDF schema to handle data from agroecological holobionts

Marie Lahaye[1], Alice Mataigne[2], Edmond Berne[3,4,5], Matéo Boudet[1,2], Maria Bernard[6,7], Christophe Mougel[1], Lionel Lebreton[1], Valentin Loux[3,4], Anne-Françoise Adam-Blondon[5,8], Olivier Rué[3,4] and Fabrice Legeai[1,2]

[1] UMR 1349 IGEPP, INRAE, Le Rheu 35650, France
[2] Univ Rennes, Inria, CNRS, IRISA - UMR 6074, F-35000 Rennes, France
[3] Université Paris-Saclay, INRAE, MaIAGE, 78350 Jouy-en-Josas, France
[4] Université Paris-Saclay, INRAE, BioinfOmics, MIGALE bioinformatics facility, 78350 Jouy-en-Josas, France
[5] IFB-Core, Institut Français de Bioinformatique (IFB), CNRS, INSERM, INRAE, CEA, 94800 Villejuif, France
[6] INRAE, SIGENAE, Jouy-en-Josas, France
[7] INRAE, AgroParisTech, GABI, Université Paris-Saclay, Jouy-en-Josas, France
[8] URGI, INRAE, Université Paris-Saclay, 78026 Versailles, France

Corresponding author: marie.lahaye@inrae.fr, fabrice.legeai@inrae.fr

Keywords data integration, ontologies, holobionts, RDF, agroecology, metagenomics

Abstract *Understanding host-microbiota interactions in agroecological studies requires linking complex heterogeneous datasets, such as microbial diversity, host phenotype and environmental context. In response, we developed Holograph, a RDF (Resource Description Framework) schema dedicated to the representation of holobiont data, to provide a structured and interoperable way to store such heterogeneous information. This schema is structured around three main ontologies: SOSA, I-ADOPT and GeoSPARQL and integrates specific terms from other convenient ontologies. This schema was successfully tested and implemented on datasets describing plant and livestock animal holobionts allowing data exploration through SPARQL queries or web applications.*

Introduction

Holobionts aim to represent a host (an animal or a plant), and its microbiota as a whole. Agroecological projects studying holobionts entail identifying and characterizing the communities of living organisms associated with cultivated plants or livestock. The composition of the microbiota is generally obtained using metabarcoding or metagenomics methods, resulting in abundance tables of Amplicon Sequence Variants (ASVs) or metagenome-assembled genomes (MAGs), representing the observed taxa. In addition, in holobiont studies, many variables are collected in the field or in laboratory to accurately describe the host phenotype (including health status, growth, physiological or performance traits) or host products. Furthermore, the environment can be modelled with descriptors (e.g. presence of bioagressors, agricultural practices including diet composition, climatic data).

To properly manage and capitalize on the complexity of holobionts data, a modelization as a knowledge graph using generic and domain specific ontologies is essential. This representation enables precise semantic description of data and metadata and links information describing the host, the microbiota and the environment. Reusing terms from shared ontologies will allow interoperability between datasets, and other knowledge graphs.

In the frame of the french program MUDIS4LS (Mutualized Digital Spaces for FAIR data in Life and Health Science), we developped Holograph, a generic RDF schema based on the SOSA and I-ADOPT core-ontologies. This schema was applied to integrate data from large-scale projects: DeepImpact and ChickStress/Feed-a-Gene.

Material and methods

RDF

The Resource Description Framework (RDF) is a standard model to describe data as an oriented graph. It is divided in three elementary entities: the subject and the object linked by a predicate, all identified by a Uniform Resource Identifier (URI). This is an interoperable model thanks to URIs and the use of controlled vocabulary and ontologies. An ontology is a description and organisation of concepts to structure knowledge about topics so that actors of a same domain can share the same vocabulary. RDF data are loaded into triplestores such as Virtuoso which allows data querying through the SPARQL query language.

Ontologies

The schema partly relies on a combination of three core ontologies. The **SOSA** core ontology [1] defines classes and properties to describe observations made by sensors on a feature of interest, the observed property and temporal information. The **I-ADOPT** framework ontology[2] is designed to help with interoperability between existing variables by representing them as more specific entities that can match terms described in ontologies or taxonomies. The **GeoSPARQL**[3] ontology defines a vocabulary for representing geospatial data. We only used its two main classes (Feature and SpatialObject) as they were sufficient to describe nested locations. Our schema is also linked to the **NCBITAXON** ontology[4], which represents the NCBI organismal taxonomy. To form the rest of the schema, we extended these core ontologies using specific terms from other relevant ontologies, when available, to describe classes, properties or relations (see Table 1).

Ontology	ID	Description
Plant Ontology[5]	PO	Plant anatomy, morphology, growth and development
Uber-anatomy ontology[6]	UBERON	Cross-species anatomy ontology
Environmental ontology[7]	ENVO	Environments, ecosystems, habitats and related entities
Agronomy Ontology[8]	AGRO	Agronomic pratices, techniques, variables and related entites
FoodOn Food Ontology[9]	FOODON	Farm-to-fork ontology describing food
Relation Ontology[10]	RO	Relationship types destined to be used in multiple ontologies
Thesaurus INRAE[11]		INRAE thesaurus used to index and annotate web pages, datasets or specific terms

Tab. 1. List of the ontologies terms where extracted from to build the schema

Datasets

The model was first tested using the **DeepImpact** dataset as a plant holobiont use case. This project aims to understand the impact of soil's microbial diversity, as well as physico-chemical and environmental conditions on performance of wheat and rapeseed crops. This dataset includes hundreds of agroenvironmental descriptors (e.g. bioagressors inventories, plant phenotypic traits, agricultural practices, soil biochemistry) measured at various spatial levels (field, plot or plant) collected in 4 plots from 200 agricultural fields during 2 years (4 seasons of sampling). This dataset has been associated with climatic data extracted from the MeteoFrance SAFRAN database. In each plot, microbial commu-

nities from plant compartments (leaf, root and rhizosphere) have been sequenced (39 runs covering 6379 samples) and ASV, their taxonomic affiliations and abundances were derived from bioinformatics analyses.

The model was then applied to the dataset from the **ChickStress** and **Feed-a-Gene** projects to generalize it to both plant and livestock holobionts. Part of these projects aim to study the relationship between feed efficiency and gut microbiota in laying chickens under contrasting feeding conditions[12]. This dataset includes measurements on 57 hens on which caecum microbiotas were analysed. The hens belong to two distinct lines (R+ and R-), which are highly divergent in feed efficiency, and were fed either a control diet (CTR) or a fiber-enriched, energy-depleted diet (LE). During and at the end of the experiment, various performance (linked to feed efficiency or egg productions and quality), growth or physiological measures (including blood analysis) were recorded.

Results

Holograph schema

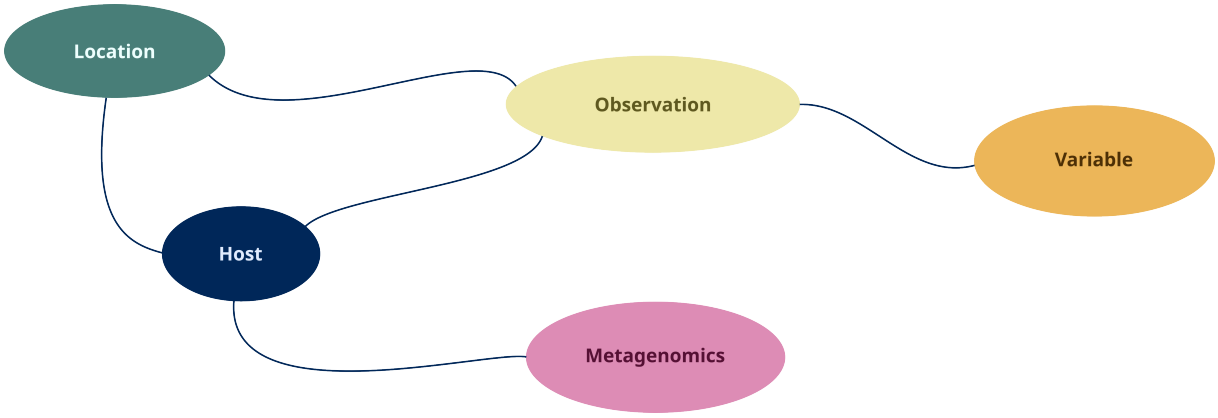

Fig. 1. Overview of the holograph schema

The holograph schema can be broken down into 5 different subparts (see Figure 1). The first one describes the **host**, either an animal or a plant, its products (eggs, in our use case), diet composition and the sampled compartments. The compartment from the chicken dataset was the caecum, term found in the UBERON ontology. In the DeepImpact project the compartments were plant organs: roots, leaves (terms found in the PO) and rhizosphere (ENVO).

These compartments can be sampled for **metagenomics** studies. As this part of the schema did not match any known ontology we developed a new model. It describes the sequenced sample, the run and the analysis, as well as the experiment and the project (if needed). Specific metadata are stored (eg: total read count, sequencing instrument or sequenced marker). Each analysis has results stored under the classes TaxonomicAssignation and Abundance, both linked to an ASV. The taxonomy has a direct link with NCBITAXON as, when instantiating the model, we extract the most accurate taxon URI found in NCBITAXON.

A host is located on a **location** which can be contained in another location. The links between the different locations are represented using the GeoSPARQL ontology. For example, a plant can be located in a plot contained in fields, and a hen is reared in a cage housed in farm. The different types of locations are terms coming from ENVO, AGRO and the Thesaurus INRAE.

Observations such as the host, the diet, the products or the locations are represented using the SOSA ontology. An observation is made by a sensor (a person or a device) on a object. A measured variable, named ObservableProperty in SOSA, is stored as a value and a unit. The observation also has a result time and is linked to a time interval during which the measurement was made (eg: the sampling season or year).

Variables are described as an ObservableProperty in the SOSA ontology. However, modelization can be quite complex (e.g. class density of a weed species at a specific phenological stage). To describe them, and allow interoperability with ontologies, we used the I-ADOPT ontology. Variables are decomposed into elementary entities: a property, an entity and a constraint on this entity. Variables can also be grouped in sets.

Data integration and exploration

To query the model, data must either be loaded directly into a triplestore (eg: Virtuoso) or through a dedicated interface, such as AskOmics (https://askomics.org/). To ease the formatting of user data into the Holograph model, we provided multiple CSV templates and adhoc scripts for conversion into the turtle standard format. Users can directly explore the ingested data with SPARQL queries into the triplestore, or through the AskOmics web interface.

Instantiating the schema with the DeepImpact dataset resulted in 20 210 804 triples and with the ChickStress/Feed-a-Gene dataset 176 190 triples. The datasets can be explored through questions, such as (for DeepImpact): *What are the taxa found in root microbiota of fields in western France under conventional agriculture in the 2nd sampling of the 2nd year*. This resulted in a SPARQL query that returned 1005 results in 4.717 seconds.

Discussion

The current Holograph schema is flexible, as terms can be added to represent other locations, products, sampling compartments. It also can be adapted to other applications: bioreactors, ecosystems or fermented foods. However, any adaptation will be met with some difficulties.

Identifying the best match for terms defined in the schema was one of the main challenge, as terms might be defined across multiple ontologies, and the most suitable descriptions may come from ontologies outside the expected domain. We tried to favor ontologies from our domain or at least generic ones, for more robustness. In particular, in the metagenomics part of the model we did not find terms in convenient ontologies or sometimes the definition of terms were not acceptable (e.g. the term sequencing run is well described in the National Cancer Institute Thesaurus OBO Edition, but this thesaurus is not relevant to agroecological study). It would be appropriate to complement existing ontologies or the thesaurus INRAE with the missing terms and their precise definitions. Consequently, filling the templates requires time to find matching terms to describe variables from the dataset. Yet, it is unavoidable to ensure data and metadata integrity. Thus, a good documentation is essential, as well as some training and support of the data producer.

Querying the model can be challenging: if the data are loaded into an or Virtuoso instance, the user needs to know how to write SPARQL queries. If using an AskOmics instance, SPARQL queries writing can be avoided but the user still needs to know and understand the schema. To answer these

problems, a new user-friendly web application dedicated to the Holograph schema is needed to allow the user to query the model easily, without having to be familiar with it as the complexity would be hidden. Contrary to AskOmics, this web interface would be specific to the schema and could be extended to other schemas using the SOSA and I-ADOPT ontologies.

Acknowledgments

This work is funded by the French National Research Agency (ANR): France 2030, Investments for the Future Program (ANR-11-INBS-0013), EQUIPEX+ (ANR-21-ESRE-0048), Alternative Crop Production and Protection (ANR-20-PCPA-0004), and BIOADAPT (ANR-13-ADAP). It has also been supported by the European Union's H2020 programme Feed-a-Gene project (No. 633531).

References

[1] Janowicz K, Haller A, Cox SJD, Le Phuoc D, Lefrançois M. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. Journal of Web Semantics. 2019;56:1-10. Available from: https://www.sciencedirect.com/science/article/pii/S1570826818300295.

[2] Magagna B, Moncoiffé G, Devaraju A, Stoica M, Schindler S, Pamment A, et al.. InteroperAble Descriptions of Observable Property Terminologies (I-ADOPT) WG Outputs and Recommendations. Zenodo; 2022. Available from: https://doi.org/10.15497/RDA00071.

[3] Nicholas J Car, Timo Homburg, Matthew Perry, John Herring, Frans Knibbe, Simon J D Cox, et al. OGC GeoSPARQL - A Geographic Query Language for RDF Data. Open Geospatial Consortium; 2023. OGC 22-047. Available from: http://www.opengis.net/doc/IS/geosparql/1.1.

[4] Schoch CL, Ciufo S, Domrachev M, Hotton CL, Kannan S, Khovanskaya R, et al. NCBI Taxonomy: a comprehensive update on curation, resources and tools. Database (Oxford). 2020 Jan;2020.

[5] Cooper L, Walls RL, Elser J, Gandolfo MA, Stevenson DW, Smith B, et al. The plant ontology as a tool for comparative plant anatomy and genomic analyses. Plant Cell Physiol. 2013 Feb;54(2):e1.

[6] Mungall CJ, Torniai C, Gkoutos GV, Lewis SE, Haendel MA. Uberon, an integrative multi-species anatomy ontology. Genome Biol. 2012 Jan;13(1):R5.

[7] Buttigieg PL, Pafilis E, Lewis SE, Schildhauer MP, Walls RL, Mungall CJ. The environment ontology in 2016: bridging domains with increased scope, semantic density, and interoperation. J Biomed Semantics. 2016 Sep;7(1):57.

[8] Devare M, Aubert C, Laporte MA, Valette L, Arnaud E, Buttigieg PL. Data-driven Agricultural Research for Development: A Need for Data Harmonization Via Semantics. In: ICBO/BioCreative; 2016. Available from: https://api.semanticscholar.org/CorpusID:41553233.

[9] Dooley DM, Griffiths EJ, Gosal GS, Buttigieg PL, Hoehndorf R, Lange MC, et al. FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. npj Sci Food. 2018 Dec;2(1):23. Available from: https://www.nature.com/articles/s41538-018-0032-6.

[10] Schaab GL. Relational Ontology. In: Runehov ALC, Oviedo L, editors. Encyclopedia of Sciences and Religions. Dordrecht: Springer Netherlands; 2013. p. 1974-5. Available from: http://link.springer.com/10.1007/978-1-4020-8265-8_847.

[11] National research institute for agriculture food and environment, National research institute for agriculture food and environment. Thésaurus INRAE. Recherche Data Gouv; 2021. Available from: https://entrepot.recherche.data.gouv.fr/citation?persistentId=doi:10.15454/J8GANU.

[12] Bernard M, Lecoeur A, Coville JL, Bruneau N, Jardet D, Lagarrigue S, et al. Relationship between feed efficiency and gut microbiota in laying chickens under contrasting feeding conditions. Scientific Reports. 2024 Apr;14(1):8210. Available from: https://doi.org/10.1038/s41598-024-58374-3.

ShareFAIR-KG, a centralised knowledge base of scientific workflows

Marie SCHMIT[1], Melvin Selim ATAY[2], Khalid BELHAJJAME[3], Ulysse LE CLANCHE[4], Emmanuel COQUERY[5], Olivier DAMERON[4], Fabien DUCHATEAU[5], Alban GAIGNARD[6,7], Mouna EL GARB[5], Jaffar GURA[3], Nicolas LUMINEAU[5], George MARCHMENT[8], Camille MAUMET[2], Clémence SEBE[8], Frédéric LEMOINE[1], Sarah COHEN-BOULAKIA[8] and Hervé MÉNAGER[1,7]

[1] Institut Pasteur, Université Paris Cité, Bioinformatics of Biostatistics Hub, F-75015 Paris, France
[2] Univ Rennes, CNRS, Inria, Inserm, IRISA UMR 6074, EMPENN — ERL U 1228, F-35000 Rennes, France
[3] LAMSADE, PSL, Paris Dauphine University, 75016 Paris, France
[4] Université Rennes, Inria, CNRS, IRISA—UMR 6074, Rennes 35000, France
[5] Université Lyon 1, INSA Lyon, CNRS, LIRIS, UMR 5205, Lyon, France
[6] Nantes Université, CNRS, INSERM, l'Institut du Thorax, F-44000 Nantes, France
[7] IFB-core, Institut Français de Bioinformatique (IFB), CNRS, INSERM, INRAE, CEA, 94800 Villejuif, France
[8] Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91405, Orsay, France

Corresponding author: herve.menager@pasteur.fr

Keywords Knowledge graphs, workflows, FAIR, SPARQL, metadata

Abstract *The emergence of high-throughput technologies in the Life Sciences has led to increasingly complex data analysis pipelines. Ensuring their transparency, reusability, and reproducibility requires adherence to the FAIR principles, which motivated the development of the ShareFAIR project. ShareFAIR aims to build a framework to make workflows more shareable, understandable, and reusable by improving their description through structure, provenance, and annotations.*

The ShareFAIR architecture consists of several modules addressing workflow structure extraction, provenance, annotation, and querying, with applications ranging from genomics to neuroimaging. At its core, this system depends on ShareFAIR-KG, a centralized and queryable knowledge graph (KG) that integrates all workflow components and tool metadata at various levels of granularity. Populated through automated harvesting and normalized into a graph-based representation, it consolidates the results and software produced in the project. With the help of ShareFAIR-KG, users can for instance research workflows implementing specific analytical steps, obtain the software packages used or wrapped in workflows or query citation and licensing information.

Introduction

During the last decade, an unprecedented amount of multi-scale data has been generated in the field of Life Sciences. While it offers new opportunities for biological discovery, its volume and complexity have introduced significant computational challenges. Notably, the difficulty in reproducing data analysis pipelines has become prevalent, eroding trust in scientific results and limiting the reuse of established analysis pipelines [1]. This issue is particularly relevant in bioinformatics and extends to the neuroimaging field [2].

Scientific workflows were developed to address these limitations, offering scalability, automation and reproducibility [3]. However, workflows alone are insufficient: the application of FAIR principles (Findability, Accessibility, Interoperability and Reuse) [4,5,6,7] to these pipelines is increasingly recognised

as essential for ensuring transparency, quality and reliability of computational results [4]. Other authors have highlighted the importance to develop dedicated best practices for research software that also apply to workflows [8].

The ShareFAIR project arose from this need to make scientific workflows FAIR, easier to share and to reproduce. It seeks to enhance the reliability of both datasets (e.g. finely annotated raw data and analysis results) and analysis protocols by explicitly tracking provenance relationship between data results and analytical methods. To achieve this objective, the ShareFAIR architecture includes multiple modules for the annotation, query, extraction and representation of workflow structure and execution traces. The basis of this architecture is a knowledge graph of annotated workflows from different languages and sources, gathering and allowing to query data generated in those modules. This knowledge base facilitates the discovery of workflows with their scientific context, structure and execution traces.

In this work, we present ShareFAIR-KG, a knowledge-graph of normalised FAIR workflow metadata. It leverages existing state-of-the-art and community adopted standards to link the structural representation of scientific pipelines with their execution traces, software components and their parameters. Although there are several initiatives for sharing bioinformatics workflows and software such as the WorkflowHub directory [9] or the bio.tools registry [10,11], the aim of ShareFAIR-KG is to bring them together by adding a semantic layer on top of them. Thus, ShareFAIR-KG automatically harvests metadata from multiple existing repositories and enables their discovery through complex querying. The knowlegde base supports several use cases. Researchers can explore workflows implementing specific analytical steps, facilitating the location and comparison of workflows across all supported languages. Tool developers can query software packages used or wrapped in workflows, for instance to guide the creation of new wrappers. Finally, data managers and curators can evaluate workflow FAIR compliance by querying licensing or citation information. As the groundwork of the ShareFAIR architecture, ShareFAIR-KG integrates functionalities and results from ongoing collaborative development across multiple modules of the ShareFAIR project.

This paper is organised as follows. First, we present the standards selected for workflow representation, that build the semantics of ShareFAIR-KG. Next, we explain how the knowledge graph is deployed and populated, showing examples of use cases to demonstrate its query-ability and how it explains workflows at every granularity level. Finally, we detail how the knowledge base serves as the integration layer for other modules of the ShareFAIR project, and how we plan to improve this joint work to expand its content and capabilities.

Scientific workflow and tool metadata standards

Comprehensive workflow description requires a representation model with sufficient granularity, as workflows can be studied at different scales: (i) the workflows and their scientific domains at the highest level, (ii) then their processes, either nested subworkflows or computational units calling a (iv) software with its parameters. Those components and the data-flow between them must also be considered from both the prospective (specification) and retrospective (execution traces) provenance perspective.

This representation is further subject to additional critical requirements. First, it must integrate existing ontologies, schemas or vocabularies while avoiding to redefine yet another standard. Then, it must facilitate metadata generation and exchange by adopting interoperable and reusable standards that are widely adopted by the community and already integrated by existing frameworks.

The ShareFAIR-KG semantic model [12] complies with those constraints and provides (i) scientific context and metadata for workflows and their constituent software and steps, using domain ontologies such as EDAM [13]; (ii) unique software referencing via bio.tools [10,11] identifiers; and (iii) prospective and retrospective provenance representation throught Provenance Run Crate [14], an RO-Crate profile extending WorkflowHub's workflow profile (describing workflow files and metadata), Process Run Crate (software used for file creation) and Workflow Run Crate (workflow execution by a Workflow Management System or WMS).

Populating and querying ShareFAIR-KG

ShareFAIR-KG is a knowledge base of workflow metadata following this representation model. Queryable with SPARQL, it is deployed in a Jena Fuseki server, inside a Docker container providing isolation and reproducibility.

The knowledge base integrates (i) EDAM domain-specific ontology for annotation, (ii) bio.tools registry for software identification and as a source of pre-existing EDAM annotations, (iii) WorkflowHub general metadata on workflows, (iv) metadata on the prospective and retrospective provenance of workflows from various WMS and sources, including: WorkflowHub, nf-core, Common Workflow Language (CWL) [15,16] and the *Intergalactic Workflow Commission* (IWC) library. We chose those registries and libraries for their wide-spread adoption by the community and their alignment with FAIR principles, leveraging open registries (e.g. bio.tools) and standards (e.g. RO-Crate).

ShareFAIR-KG currently contains the prospective provenance of 49 Nextflow [17] and 73 Common Workflow Language (CWL) workflows; and both types of provenance for 102 Galaxy[18] workflows[18] executed on Galaxy server. This dataset spans workflows covering very diverse domains of bioinformatics, from genome assembly to metabolite profiling. Querying this knowledge graph with SPARQL yields deeper insights into these workflows, as shown in the following use cases (further detailed in [12]):

- **Tool adoption** An example query is the search for the most frequently used tools[19] from bio.tools among workflows in the knowledge base, with their EDAM operations. Querying ShareFAIR-KG reveals that the top two by frequency are MultiQC [19] and dada2 [20], performing the operations sequencing quality control and *DNA barcoding*, *variant calling* (Table 1).
- **Workflow using specific tools** Knowing that MultiQC is the most used tool amongst workflows in ShareFAIR-KG, the workflows using it can be queried. For example, the Galaxy workflows *Genome Assembly from Hifi reads - VGP3* and *Scaffolding with Hi-C data VGP8* both use MultiQC.
- **Workflow input format** Another example of query involves retrieving workflow input and output data. For instance, the inputs of the CWL workflow *gatk4W* have the formats FASTA and FASTQ.

[18] Workflows were selected based on open-source licencing and technical compatibility with our extraction tools.
[19] Tools that are the most frequently used in workflows, not the ones which have been run the most overall through these workflows.

– **Workflow tools output formats** The last query example infers workflow outputs from bio.tools EDAM annotations. Annotations of the bio.tools software executed in the Galaxy workflow *ATAC-seq* [21] indicate that it generates BED, WIG, TSV, HTML, JSON, YAML and CSV data.

Tab. 1. Results of the SPARQL query retrieving most-used bio.tools tools, and their integration frequency in steps (# workflow steps). The names of the tools are from bio.tools. The namespace edam corresponds to http://edamontology.org/.

Tool name	bio.tools identifier	EDAM operation	EDAM operation label	# workflow steps
MultiQC	https://bio.tools/multiqc	edam:operation_3218, edam:operation_2428	Sequencing quality control, Validation	18
dada2	https://bio.tools/dada2	edam:operation_3200, edam:operation_3227	DNA barcoding, Variant calling	12
Bandage	https://bio.tools/bandage	edam:operation_3184	Sequence assembly visualisation	5

Current and Future Integration within the ShareFAIR Project

ShareFAIR-KG constitutes the foundational infrastructure of the ShareFAIR architecture, centralising output from other modules and providing data for query and representation purposes to others (Figure 1). While it already integrates several of their functionalities, ongoing efforts aim to strengthen this integration and consolidate the ShareFAIR framework. This collaboration will address key limitations by enhancing annotation depth and quality as well as expanding workflow coverage.

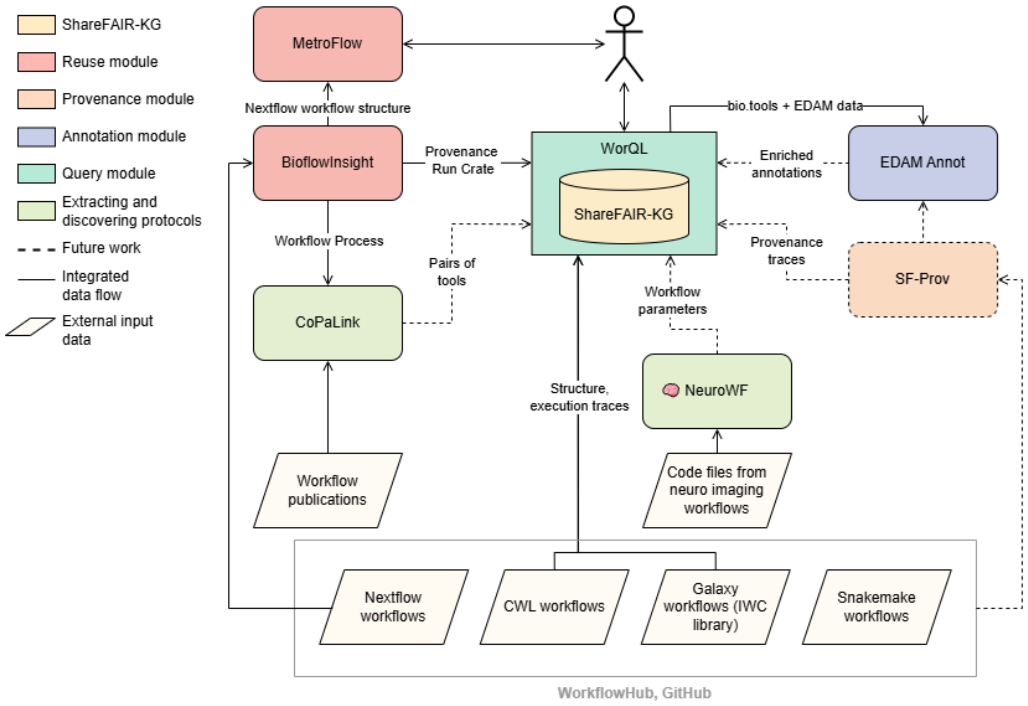

Fig. 1. ShareFAIR architecture with the software developed in different modules and the data flowing between them (type of data is specific on the arrows). Solid arrows represent current integration, dotted arrows show work in progress.

The Reuse Module aims to extract and visualise workflow structure. As part of this module, BioflowInsight [22] was developed to capture Nextflow workflow structures and generate their Provenance Run Crate metadata within ShareFAIR-KG. Additionally, MetroFlow [23] provides a unique visualization of Nextflow workflows, representing them as intuitive metro maps.

The Provenance Module ingests and exploits provenance traces captured from workflow executions. Only a limited number of WMS supports provenance capture, such as Galaxy and Nextflow with the plugin nf-prov. This module currently aligns nf-prov provenance with prospective specifications. In the short-term, our road map includes the verification and repair of the completeness of nf-prov outputs. In the longer term, we investigate multi-granular representations of provenance, including fine-grained execution traces. Both the repair and the refined provenance derivations (at different levels of granularity) will be reflected in ShareFAIR-KG.

The Annotation Module aims at inferring missing metadata to better annotate and explain scientific pipelines, as well as analyzed data. Bio.tools metadata and the EDAM ontology from ShareFAIR-KG are at the core of the EDAMAnnot [24] toolbox and were used to investigate metadata quality in terms of information content. Neuro-symbolic approaches leveraging both retrieval augmented generation and knowledge graphs are under investigation to i) increase the quality of bioinformatics tools and workflow metadata and ii) annotate analysed data for increase potential reuse.

The Query Module leverages BioFlow-Ontology [25], a comprehensive workflow representation model, to develop WorQL, a dedicated query language for workflow interrogation. WorQL employs *query relaxation mechanisms*, such that if a query yields no results, it is automatically reformulated to return meaningful matches [26]. This enhances ShareFAIR-KG accessibility, enabling users from diverse backgrounds to efficiently query workflows and fostering broader adoption. Bioflow-Ontology and the ShareFAIR-KG representation model serve complementary purposes: one extends existing standards for seamless metadata generation and interoperability, the other optimises workflow querying. The integration of this module with ShareFAIR-KG will leverage inference rules and reasoning to align both data representations.

Linking bioinformatics tools across publication and workflow code Workflow descriptions in publications and actual code often diverge. For example, the same tool may appear under different names, or can be omitted from the publication. To address this, CoPaLink [27] has been developed to extract all bioinformatics tool names from both sources (publications and code). This allows users to compare tools detected in each source and identify tools that appear in only one source. CoPaLink will be integrated in ShareFAIR-KG.

Neuroimaging Workflow Integration (NeuroWF), currently in development Besides bioinformatics workflows and provenance tools, ShareFAIR-KG will enable querying and exploring the collection of open source neuroimaging workflows. In particular, we will generate Provenance Run Crate metadata to describe functional magnetic resonance imaging (task-fMRI) pipelines of the most commonly used software tools. By using the previously described modules, NeuroWF aims to enable parameter metadata [28] querying for further analysis of neuroimaging workflows in a reproducible manner.

Conclusion

This work paves the way for FAIR sharing of scientific workflows. It leverages the results and functionalities developed in the ShareFAIR project modules to gather and normalise workflow annotations and metadata from multiple sources and languages, providing details on their scientific context, structure and execution traces to facilitate workflow discovery, understanding, sharing and reuse.

ShareFAIR-KG still faces specific constraints, mainly related to the quality and completeness of annotations and workflow descriptions. Query capabilities are currently limited by the variable depth and coverage of workflow description. Additionally, the knowledge graph is rebuilt from scratch during each update rather than maintained incrementally with workflow versioning support, limiting its scalability for evolving pipelines.

Future work will first address the lack of complete workflow annotations by finalising the integration with the other ShareFAIR modules, to integrate a larger diversity of standards and increase annotation quality, for instance by enriching bio.tools annotations. Furthermore, we plan to increase the scale of workflow integration to encompass neuroimaging workflows, while expanding the coverage of execution traces beyond Galaxy workflows. Joint efforts will also focus on improving query accessibility and user-friendliness. These advances will establish the technical foundation for the ShareFAIR framework.

Availability and implementation

All resources are openly available: **ShareFAIR-KG** with examples of queries and competency questions (GitLab, v1.0.0, GNUGPLv3.0, `swh:1:dir:4e761e3`), **EDAMAnnot** (GitHub, GNUGPLv3.0, `swh:1:dir:bf911b6`), **BioFlowInsight** (GitLab, GNUAGPLv3.0), **NeuroWF** (GitLab, MIT License, `swh:1:dir:467be46`), and **CoPaLink** (tool and experiment repositories, GNUAGPLv3.0) are hosted on GitLab and GitHub with persistent Software Heritage archives; the **knowledge graph content** (Zenodo, DOI:10.5281/zenodo.17737888) and **BioFlow-Ontology** (Zenodo, v7, CC-BY4.0) are available on Zenodo.

Acknowledgments

This work was supported in part by the National Research Agency under the France 2030 program, with reference to ANR-22-PESN-0007 (ShareFAIR project).

Generative AI Statement

During the preparation of this work, the authors used Kimi K2 (Moonshot AI) and Claude Sonnet 4.5 to improve textual clarity through reformulation, as well as to verify grammar and spelling. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Citations

References

[1] Cohen-Boulakia S, Belhajjame K, Collin O, Chopard J, Froidevaux C, Gaignard A, et al. Scientific Workflows for Computational Reproducibility in the Life Sciences: Status, Challenges and Opportunities. Future Generation Computer Systems. 2017 Oct;75:284-98.

[2] Botvinik-Nezer R, Holzmeister F, Camerer CF, Dreber A, Huber J, Johannesson M, et al. Variability in the analysis of a single neuroimaging dataset by many teams. Nature. 2020 Jun;582(7810):84-8. Available from: https://doi.org/10.1038/s41586-020-2314-9.

[3] Djaffardjy M, Marchment G, Sebe C, Blanchet R, Belhajjame K, Gaignard A, et al. Developing and reusing bioinformatics data analysis pipelines using scientific workflow systems. Computational and Structural Biotechnology Journal. 2023 Jan;21:2075-85. Available from: https://www.sciencedirect.com/science/article/pii/S2001037023001010.

[4] Goble C, Cohen-Boulakia S, Soiland-Reyes S, Garijo D, Gil Y, Crusoe MR, et al. FAIR Computational Workflows. Data Intelligence. 2020;2(1-2):108-21. Available from: https://doi.org/10.1162/dint_a_00033.
[5] Wilkinson SR, Aloqalaa M, Belhajjame K, Crusoe MR, de Paula Kinoshita B, Gadelha L, et al. Applying the FAIR principles to computational workflows. Scientific Data. 2025;12(1):328.
[6] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for Scientific Data Management and Stewardship. Scientific Data. 2016 Mar;3(1):160018.
[7] Barker M, Chue Hong NP, Katz DS, Lamprecht AL, Martinez-Ortiz C, Psomopoulos F, et al. Introducing the FAIR Principles for Research Software. Scientific Data. 2022 Oct;9(1):622.
[8] Di Cosmo R, Granger S, Hinsen K, Jullien N, Le Berre D, Louvet V, et al. CODE beyond FAIR: a roadmap for reusable research software. Scientific Data. 2025;In Press.
[9] Gustafsson OJR, Wilkinson SR, Bacall F, Pireddu L, Soiland-Reyes S, Leo S, et al.. WorkflowHub: a registry for computational workflows. arXiv; 2024. ArXiv:2410.06941 [cs]. Available from: http://arxiv.org/abs/2410.06941.
[10] Ison J, Rapacki K, Ménager H, Kalaš M, Rydza E, Chmura P, et al. Tools and data services registry: a community effort to document bioinformatics resources. Nucleic Acids Research. 2016 Jan;44(D1):D38-47. Available from: https://doi.org/10.1093/nar/gkv1116.
[11] Ison J, Ienasescu H, Chmura P, Rydza E, Ménager H, Kalaš M, et al. The bio.tools registry of software tools and data resources for the life sciences. Genome Biology. 2019 Aug;20(1):164. Available from: https://doi.org/10.1186/s13059-019-1772-6.
[12] Schmit M, Le Clanche U, Marchment G, Cohen-Boulakia S, Dameron O, Gaignard A, et al. A Standards-Based Knowledge Graph that Bridges Scientific Workflows, Run-Time Provenance, and Tool Registries. In: SWAT4HCLS 2026. Amsterdam, Netherlands; 2026. Available from: https://hal.science/hal-05517640.
[13] Gaignard A, Skaf-Molli H, Belhajjame K. Findable and reusable workflow data products: A genomic workflow case study. Semantic Web. 2020;11(5):751-63. Available from: https://journals.sagepub.com/doi/abs/10.3233/SW-200374.
[14] Leo S, Crusoe MR, Rodríguez-Navas L, Sirvent R, Kanitz A, De Geest P, et al.. Recording Provenance of Workflow Runs with RO-Crate. arXiv; 2024.
[15] Amstutz P, Crusoe MR, Tijanić N, Chapman B, Chilton J, Heuer M, et al.. Common Workflow Language, v1.0; 2016. Publisher: figshare. Available from: https://research.manchester.ac.uk/en/publications/common-workflow-language-v10/.
[16] Crusoe MR, Abeln S, Iosup A, Amstutz P, Chilton J, Tijanić N, et al. Methods included: standardizing computational reuse and portability with the common workflow language. Communications of the ACM. 2022;65(6):54-63.
[17] Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature Biotechnology. 2017 Apr;35(4):316-9. Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/nbt.3820.
[18] The Galaxy Community. The Galaxy Platform for Accessible, Reproducible and Collaborative Biomedical Analyses: 2022 Update. Nucleic Acids Research. 2022 Jul;50(W1):W345-51.
[19] Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics. 2016 06;32(19):3047-8. Available from: https://doi.org/10.1093/bioinformatics/btw354.
[20] Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. DADA2: High-resolution sample inference from Illumina amplicon data. Nature methods. 2016;13(7):581-3.
[21] Lopez-Delisle L. github.com/iwc-workflows/atacseq/main. Zenodo; 2025. Available from: https://doi.org/10.5281/zenodo.15078780.
[22] Marchment G, Brancotte B, Schmit M, Lemoine F, Cohen-Boulakia S. BioFlow-Insight: facilitating reuse of Nextflow workflows with structure reconstruction and visualization. NAR Genomics and Bioinformatics. 2024 08;6(3):lqae092. Available from: https://doi.org/10.1093/nargab/lqae092.

[23] Marchment G, Brancotte B, Cohen-Boulakia S, Gura J, Lemoine F. MetroFlow : Generating Interactive Metro-Maps from Workflow Code. Nextflow Summit 2025. 2025. Available from: https://www.youtube.com/watch?v=mgR7wQIGUDo.

[24] Le Clanche U, Cohen-Boulakia S, Cunff YL, Dameron O, Gaignard A. Assessing bioinformatics software annotations: bio.tools case-study. In: Proceedings JOBIM. Bordeaux & Online, France; 2025. p. 1-7. Available from: https://inria.hal.science/hal-05096849.

[25] El Garb M, Coquery E, Duchateau F, Lumineau N. Improving reproducibility in bioinformatics workflows with BioFlow-Model. In: ACM Conference on Reproducibility and Replicability (ACM REP '25),. Vancouver, Canada: ACM; 2025. Available from: https://hal.science/hal-05240352.

[26] Fakih G, Serrano-Alvarado P. A survey on SPARQL query relaxation under the lens of RDF reification. Semantic Web. 2024;15(6):2507-54.

[27] Sebe C, Ferret O, Névéol A, Esmailoghli M, Leser U, Cohen-Boulakia S. Supporting Workflow Reproducibility by Linking Bioinformatics Tools across Papers and Executable Code. arXiv; 2026. ArXiv:2603.08195 [cs]. Available from: http://arxiv.org/abs/2603.08195.

[28] Atay MS, Clenet B, Bannier E, Maumet C. Real-world parameter preferences in task-fMRI preprocessing pipelines. In: OHBM 2026 - Annual Meeting of the Organization for Human Brain Mapping. Bordeaux, France: OHBM; 2026. Available from: https://inria.hal.science/hal-05545909.