

LCR-modules: a collection of workflows for cancer genome analysis

Kostiantyn Dreval^{1,2}, Laura K. Hilton³, Bruno M. Grande¹, Giuliano Banco¹, Krysta M. Coyle¹, Manuela Cruz¹, Sierra Gillis^{2,3}, Luke Klossok¹, Prasath Pararajalingam¹, Christopher K. Rushton¹, Haya Shaalan¹, Nicole Thomas¹, Helena Winata¹, Jasper Wong³, Jacky Yiu¹, Christian Steidl³, David W. Scott^{3,4}, Ryan D. Morin^{1,2,3,*,} 

¹Department of Molecular Biology and Biochemistry, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

²Basic and Translational Research, BC Cancer Research Institute, Vancouver, BC, V5Z 1L3, Canada

³Centre for Lymphoid Cancer, BC Cancer Research Institute, Vancouver, BC, V5Z 1L3, Canada

⁴Division of Medical Oncology, Department of Medicine, University of British Columbia, Vancouver, BC, V5Z 4E6, Canada

*Corresponding author. Department of Molecular Biology and Biochemistry, Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada.

E-mail: rdmorin@sfu.ca.

Associate Editor: Can Alkan

Abstract

Motivation: The surge of genomic data from advanced sequencing technologies is outpacing current analytical pipelines. We introduce LCR-modules, an open-source suite of bioinformatics tools designed for flexible and automated cancer genome data analysis. LCR-modules enables reproducible analysis of diverse cancer genomics data at scale. The suite comprises 49 Snakemake-based workflows organized into three levels, facilitating tasks from low-level quality control to complex cohort-level analyses. LCR-modules supports various sequencing types and integrates pipelines such as mutation calling, expression quantification, and cohort-level aggregation, ensuring flexibility and reproducibility. LCR-modules represents a significant advancement in genomic data analysis, reducing barriers in reproducibility and scalability and has already been applied to a combination of exomes and genomes from over 10 800 samples.

Availability: No new data were generated in support of this research. The source code for the LCR-modules is openly available at <https://github.com/LCR-BCCRC/lcr-modules>.

1 Introduction

High-throughput sequencing has been transformational to genetic research and the study of cancer (Liu *et al.* 2019, Satam *et al.* 2023, Mustafa 2024). Large consortia, such as The Cancer Genome Atlas (Weinstein *et al.* 2013) and ICGC (Hübschmann *et al.* 2021), and collective work from thousands of teams globally have generated vast amounts of sequencing data from cancer biopsies. Generating high-quality simple somatic mutation and copy number alteration calls involves a complex collection of algorithms (Ulrich 2017, Cremin *et al.* 2022). Bioinformatics workflows commonly rely on a variety of programming languages, which are cumbersome to maintain and difficult to scale as analyses grow in scope and complexity (Dash *et al.* 2019, Wratten *et al.* 2021). Hence, despite the opportunity afforded by a wealth of cancer sequencing data to reveal novel molecular insights, the ability to rapidly process data in a reproducible manner creates a bottleneck.

For research to keep pace with the influx of data, pipelines to automate routine analytical steps are essential. Ideally, analytical

pipelines balance versatility, tuneability, and ease of installation and maintenance while minimizing risks of errors. Modularization and version control facilitate the consistent application of analytical best practices and allow deployment of new features and improvements across projects (Davis-Turak *et al.* 2017, Houssein *et al.* 2023). Reliance on modular components helps promote consistency and reproducibility while reducing the amount of lead time in setting up a new analytical pipeline.

To support the routine analyses of short-read exome, panel-based, transcriptome, whole genome sequencing (WGS) and long-read sequencing, we developed a suite of tools collectively named Lymphoma Cancer Research modules (LCR-modules). This toolkit has been used by our team to discover drivers and molecular features in cancer, specifically focusing on lymphoid neoplasms, and perform quantitative comparisons between disease entities (Bal *et al.* 2022, Dreval *et al.* 2022, 2023, Hilton *et al.* 2023, Thomas *et al.* 2023, 2025, Sahasrabuddhe *et al.* 2025). The collection of LCR-modules includes open-source and custom bioinformatics tools utilizing the Snakemake (Mölder

et al. 2021) workflow management system, and modules for data quality control, discovery and annotation of common mutation types, analysis of B-cell receptor repertoires, and discovery of loci affected by aberrant somatic hypermutation. Individual modules are configured to provide an automated, scalable, and reproducible workflow that runs each step as dictated by the availability of new data. LCR-modules is actively maintained and openly available on GitHub.

2 Results

2.1 LCR-modules overview

LCR-modules is a collection of modules that form the basis of workflows to perform scalable and reproducible genomic analyses. The individual modules can be logically divided into three main categories representing the processing of raw, sample-level or cohort-level data (Fig. 1A). The outputs of the modules of each previous level flow, individually or in combination, to modules at the subsequent level.

An important feature of the modules at all levels includes a specifically designed set of processes to handle required reference files automatically with minimal effort from the user. These reference processes handle the download of fasta files for majority of the most common human genome builds and bed files of target regions for commonly used exome panels. While a wide range of genome builds is available to the user out-of-the-box, the reference files subworkflow also provides flexibility to add custom reference files or capture-based panels.

Level 1 modules perform low-level tasks such as adapter trimming, quality control, and alignment of sequencing files, and obtaining data from repositories such as the European Genome-phenome Archive (EGA). These modules also perform gene expression analyses, including alignment using STAR (Dobin *et al.* 2013) and calculating mRNA abundance using salmon (Patro *et al.* 2017). Level 2 modules perform routine tasks for cancer analysis, such as detecting and annotating simple somatic mutations, copy-number alterations, and structural variations. Next, the level 3 modules perform analyses that rely on cohort-level aggregation. The cohorts and data sets can be flexibly defined based on different clinical characteristics through a set of configuration files. The modules at this level operate on the outputs of level 2 modules and perform tasks such as aggregation of individual files into cohort-level merges. Example workflows include analyses of mutation signatures, identification of significantly mutated genes, and sample classification into genetic subgroups.

Each of the modules has required and optional components (Fig. 1B). Required components include: the Snakefile (rules), schema (definition of wildcards), conda environment (dependencies), and configuration file (module config). The module config defines the input/output paths, tool parameters, computational resource requirements, supported sequencing types (seq_type), and toggles for handling data with or without a matched normal. Optional components include a reference subworkflow (automated reference assets with user overrides) and shared custom scripts [provided within src directory or via the companion LCR-scripts (<https://github.com/LCR-BCCRC/lcr-scripts>) repository]. This structure standardizes behavior, recycles common resources, and minimizes code duplication. To

improve handling of routine tasks—like subsetting sample table, generating tumor-normal pairs, creating directory structure for the module run, symlinking files, etc.—LCR-modules utilize a custom Python-based oncopipe module (<https://pypi.org/project/oncopipe/>), developed by our group.

The current collection includes 50 workflows designed to support both short-read Illumina and long-read PromethION sequencing data. The scope of supported sequencing types (seq_type) currently includes genome, capture, mrna, and long-read sequencing data (Oxford Nanopore). Most of the modules support more than one seq_type, often with automatic configurations tailored to the data (Fig. 1C). WGS (genome seq_type) has the highest number of supporting modules, and somatic variant calling and identification of significantly mutated genes are the analysis types with the highest representation of the developed modules.

2.2 Assembling modules into pipelines

Any individual modules with compatible inputs or outputs can be combined to facilitate multi-step genomic analysis workflow. Built using Snakemake (Mölder *et al.* 2021), modules can be directly imported into a project Snakefile, exposing their individual rules to the master (project) workflow (Fig. 1D). To establish a project workflow, each of the modules with its associated config constitutes an individual step, and the individual jobs that will be executed are dictated by the contents of the sample table, the presence of compatible input files, and the target rule definitions. The default configuration values of each of the modules can be replaced with the values specified in the project-level config. Such architecture allows the entire project to be concise, easy to maintain, scalable, and reproducible.

The LCR-modules repository provides a demo project where individual modules are organized into a representative workflow based on their sequencing type. The demo project illustrates the concept of the project for users while also ensuring the maintenance of functionality for the entire codebase with every new update. In addition, the demo contains custom wrapper scripts to launch the workflows as a dry run with local execution or with cluster execution using schedulers and workload managers such as Sun Grid Engine and Slurm (Yoo *et al.* 2003).

One example of such workflow is the simple somatic variant calling pipeline in the demo project (SLMS-3; https://github.com/LCR-BCCRC/lcr-modules/tree/master/modules/slms_3). Within both capture- and WGS-based pipelines (<https://github.com/LCR-BCCRC/lcr-modules/tree/master/demo>), SLMS-3 independently runs mutect2 (Benjamin *et al.* 2019), strelka2 (Kim *et al.* 2018), SAGE (<https://github.com/hartwigmedical/hmftools/tree/master/sage>), LoFreq (Wilm *et al.* 2012), and manta (Chen *et al.* 2016) variant callers. Starfish (<https://github.com/dancooke/starfish>) is used to intersect the resulting variant calls from each individual tool such that only variants supported by three or more variant callers are retained. Custom scripts, customized versions of individual callers, and all other required components of SLMS-3 are directly supplied with LCR-modules and pre-configured for the user.

Reproducibility in cancer genomics depends on clear, inspectable pipelines, yet pipeline management is challenged by large datasets, variable genome references and assays, heterogeneous tools, and scaling needs. LCR-modules address each of

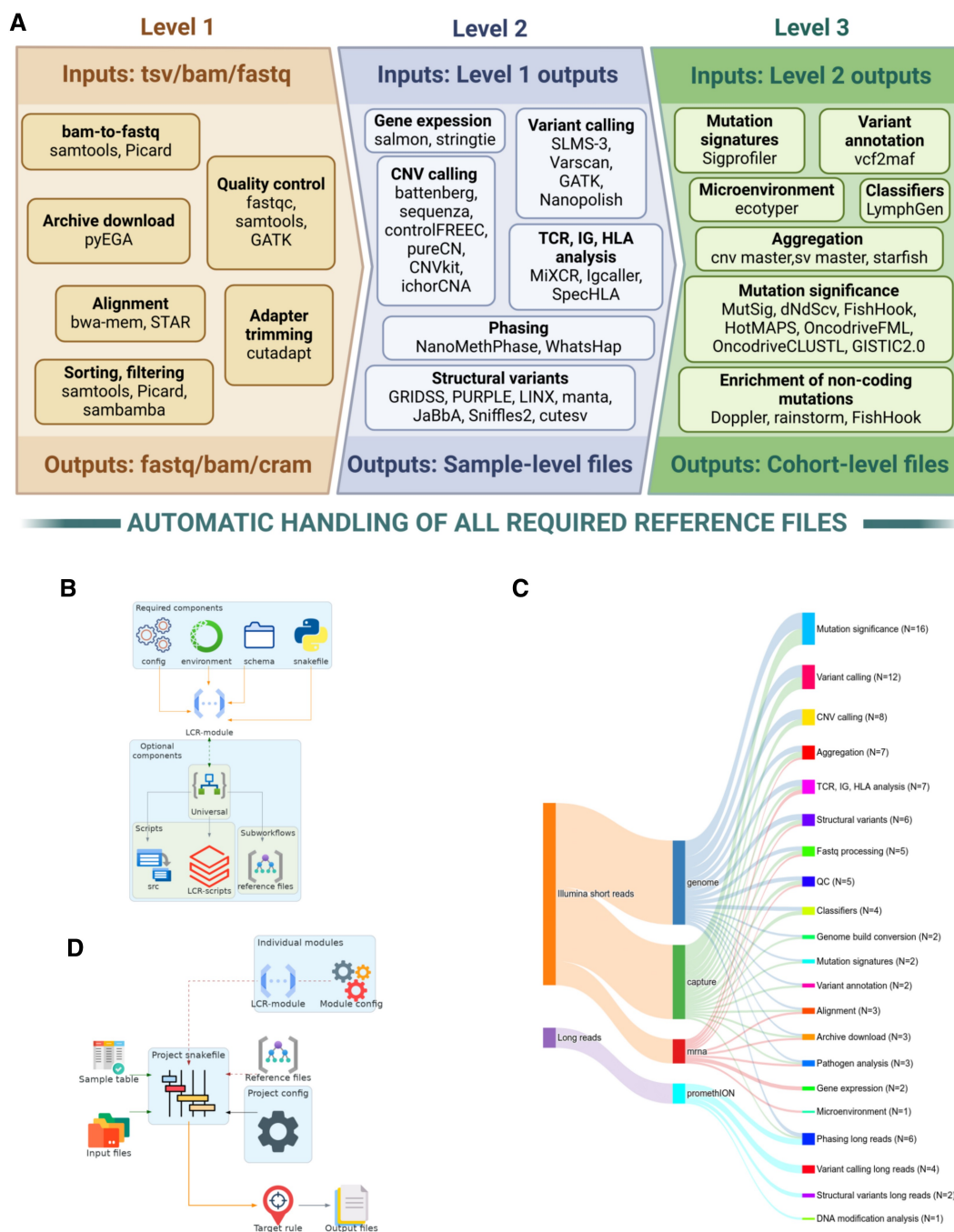


Figure 1 LCR-modules overview. (A) Schematic representation of modules as organized into different levels according to the complexity of performed analyses. Only representative individual modules are shown. The full list of modules is available at <https://github.com/LCR-BCCRC/lcr-modules/tree/master/modules>. (B) The general structure of the main components of individual modules. (C) Sankey plot showing the relationship between supported sequencing type and generalized function of each module. The number in parentheses illustrates the number of supported sequencing types and individual modules for the specific type of analysis. (D) Schematic representation of the main components in a project workflow organized with LCR-modules.

these issues with modular Snakemake workflows, pinned conda environments, and a reference subworkflow that automates genome assets. Declarative configs (not ad hoc scripts) control inputs, resources, reference files, and seq_type handling, reducing developer effort while preserving transparency. Combined with reusable utilities and an included demo for end-to-end testing, this design promotes consistent results across cohorts and

infrastructure, and delivers accuracy on FFPE samples while scaling to large studies. To date, LCR-modules has powered analyses in numerous publications (Dreval et al. 2022, 2023, Hilton et al. 2023, Thomas et al. 2023, 2025, Sahasrabudde et al. 2025) to perform sequencing sample QC, identify somatic mutations, significantly mutated genes, genetic subgroup assignment—and has processed more than 2060 whole genomes,

4500 exomes, and 4130 RNA-seq datasets, highlighting both its reliability and broad applicability.

Acknowledgements

This work was supported by a Terry Fox New Investigator Award (No. 1043) and by an operating grant from the Canadian Institutes for Health Research and a New Investigator Award from the Canadian Institutes for Health Research (R.D.M.). R.D. M. is a Michael Smith Foundation for Health Research Scholar, and D.W.S. is a Michael Smith Foundation for Health Research Health Professional-Investigator.

Author contributions

Kostiantyn Dreval (Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Validation [Equal], Visualization [Equal], Writing—original draft [Equal], Writing—review & editing [Equal]), Laura K Hilton (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Project administration [Equal], Resources [Equal], Software [Equal], Validation [Equal], Visualization [Equal], Writing—review & editing [Equal]), Bruno M Grande (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Giuliano Banco (Software [Equal], Writing—review & editing [Equal]), Krysta M Coyle (Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Manuela Cruz (Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Sierra Gillis (Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Software [Equal], Writing—review & editing [Equal]), Luke Klossok (Software [Equal], Writing—review & editing [Equal]), Prasath Pararajalingam (Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Christopher K Rushton (Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Haya Shaalan (Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Nicole Thomas (Data curation [Equal], Formal analysis [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Helena Winata (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Methodology [Equal], Writing—review & editing [Equal]), Jasper Wong (Data curation [Equal], Formal analysis [Equal], Investigation [Equal], Methodology [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Jacky Yiu (Formal analysis [Equal], Resources [Equal], Software [Equal], Writing—review & editing [Equal]), Christian Steidl (Conceptualization [Equal], Funding acquisition [Equal], Project administration [Equal], Supervision [Equal], Writing—review & editing [Equal]), David W. Scott (Conceptualization [Equal], Funding acquisition [Equal], Project administration

[Equal], Supervision [Equal], Writing—review & editing [Equal]), and Ryan Morin (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Funding acquisition [Equal], Investigation [Equal], Methodology [Equal], Project administration [Equal], Resources [Equal], Software [Equal], Supervision [Equal], Validation [Equal], Visualization [Equal], Writing—original draft [Equal], Writing—review & editing [Equal])

Conflict of interests

None declared.

Data availability

No new data were generated in support of this research. The source code for the LCR-modules is openly available at <https://github.com/LCR-BCCRC/lcr-modules>.

References

- Bal E, Kumar R, Hadigol M *et al.* Super-enhancer hypermutation alters oncogene expression in b cell lymphoma. *Nature* 2022; **607**:808–15.
- Benjamin D, Sato T, Lichtenstein L *et al.* 2019. Calling somatic SNVs and indels with Mutect2. bioRxiv, 861054, preprint: not peer reviewed.
- Chen X, Schulz-Trieglaff O, Shaw R *et al.* Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* 2016;**32**:1220–2.
- Cremin CJ, Dash S, Huang X *et al.* Big data: historic advances and emerging trends in biomedical research. *Curr Res Biotechnol* 2022;**4**:138–51.
- Dash S, Shakyawar SK, Sharma M *et al.* Big data in healthcare: management, analysis and future prospects. *J Big Data* 2019;**6**:54.
- Davis-Turak J, Courtney SM, Hazard ES *et al.* Genomics pipelines and data integration: challenges and opportunities in the research setting. *Expert Rev Mol Diagn* 2017;**17**:225–37.
- Dobin A, Davis CA, Schlesinger F *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 2013;**29**:15–21.
- Dreval K, Boutros PC, Morin RD *et al.* Minimal information for reporting a genomics experiment. *Blood* 2022;**140**:2549–55.
- Dreval K, Hilton LK, Cruz M *et al.* Genetic subdivisions of follicular lymphoma defined by distinct coding and noncoding mutation patterns. *Blood* 2023;**142**:561–73.
- Hilton LK, Ngu HS, Collinge B *et al.* Relapse timing is associated with distinct evolutionary dynamics in diffuse large b-cell lymphoma. *J Clin Oncol* 2023;**41**:4164–77.
- Houssein EH, Hosney ME, Emam MM *et al.* Soft computing techniques for biomedical data analysis: open issues and challenges. *Artif Intell Rev* 2023;**56**:2599–649.
- Hübbschmann D, Kleinheinz K, Wagener R *et al.* Mutational mechanisms shaping the coding and noncoding genome of germinal center derived b-cell lymphomas. *Leukemia* 2021;**35**:2002–16.
- Kim S, Scheffler K, Halpern AL *et al.* Strelka2: fast and accurate calling of germline and somatic variants. *Nat Methods* 2018; **15**:591–4.

- Liu Z, Zhu L, Roberts R *et al.* Toward clinical implementation of next-generation sequencing-based genetic testing in rare diseases: where are we? *Trends Genet* 2019;**35**:852–67.
- Mölder F, Jablonski KP, Letcher B *et al.* Sustainable data analysis with snakemake. *F1000Res* 2021;**10**:33.
- Mustafa AS. Whole genome sequencing: applications in clinical bacteriology. *Med Princ Pract* 2024;**33**:185–97.
- Patro R, Duggal G, Love MI *et al.* Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods* 2017;**14**:417–9.
- Sahasrabuddhe AA, Chen X, Ma K *et al.* The FBXO45–GEF-H1 axis controls germinal center formation and b-cell lymphomagenesis. *Cancer Discov* 2025;**15**:838–61.
- Satam H, Joshi K, Mangrolia U *et al.* Next-generation sequencing technology: current trends and advancements. *Biology (Basel)* 2023;**12**:997.
- Thomas N, Dreval K, Gerhard DS *et al.* Genetic subgroups inform on pathobiology in adult and pediatric Burkitt lymphoma. *Blood* 2023;**141**:904–16.
- Thomas N, García-Prieto CA, Dreval K *et al.* DNA methylation epitypes of burkitt lymphoma with distinct molecular and clinical features. *Blood Cancer Discov* 2025;**6**:325–42.
- Ulrich T. 2017. *Harnessing the Flood: Scaling up Data Science in the Big Genomics Era*. Cambridge, MA: Broad Institute.
- Weinstein JN, Collisson EA, Mills GB *et al.* The Cancer Genome Atlas Pan-Cancer Analysis Project. *Nat Genet* 2013;**45**:1113–20.
- Wilm A, Aw PPK, Bertrand D *et al.* LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Res* 2012;**40**:11189–201.
- Wratten L, Wilm A, Göke J *et al.* Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nat Methods* 2021;**18**:1161–8.
- Yoo AB, Jette MA, Grondona M. SLURM: Simple Linux Utility for Resource Management. In: *Workshop on Job Scheduling Strategies for Parallel Processing*. Berlin, Germany: Springer, 2003, 44–60.