485 RAFT: A FRAMEWORK TO SUPPORT RAPID AND REPRODUCIBLE IMMUNO-ONCOLOGY ANALYSES

Steven Vensko*, Benjamin Vincent, Dante Bortone. *University of North Carolina, Chapel Hill, NC, USA*

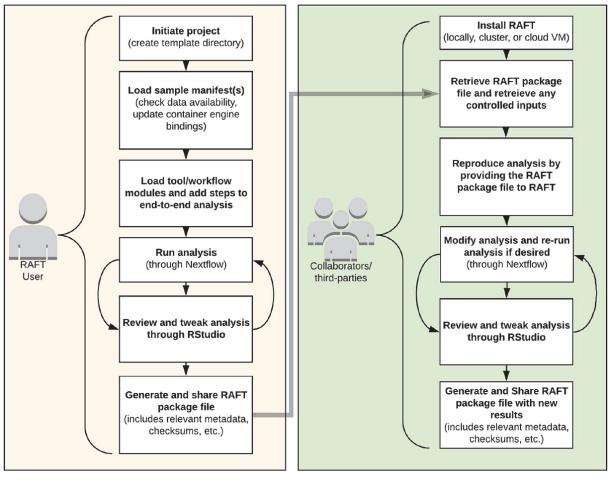
Background Analysis reproducibility and transparency are pillars of robust and trustworthy scientific results. The dependability of these results is crucial in clinical settings where they may guide high-impact decisions affecting patient health. Independent reproduction of computational results has been problematic and can be a burden on the individuals attempting to reproduce the results. Reproduction complications may arise from: 1) insufficiently described parameters, 2) vague methods, or 3) secret scripts required to generate final outputs, among others. Here we introduce RAFT (Reproducible Analyses Framework and Tools), a framework for immuno-oncology biomarker development built with Python 3 and Nextflow DSL2 which aims to enable end-to-end reproducibility of entire computational analyses in multiple contexts (e.g. local, compute cluster, or cloud) with minimal overhead through a focus on usability (figures 1 and 2).

Methods RAFT builds upon Nextflow's DSL2 module-based approach to workflows by providing a 'project' context upon which users can add metadata, load references, and build up their analysis step-by-step. RAFT also has pre-built modules with workflows commonly utilized in immuno-oncology analyses (e.g. TCR/BCR repertoire reconstruction and HLA typing) and aids users through automatic module dependency resolution. Transparency is gained by having a single end-to-end script containing all steps and parameters as well as a single configuration file. Finally, RAFT allows users to create and share a package of project metadata files including the main script, all input and output checksums, all modules, and the RAFT steps required to create the analysis. This package, coupled with any required inputs files, can be used to recreate the analysis or further expand an analysis with additional datasets or alternative parameters.

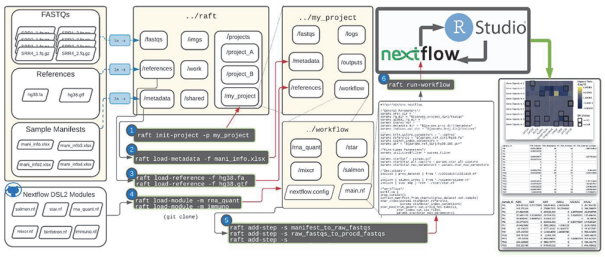
Results RAFT has been used by our computational team to create an immuno-oncology meta-analysis submitted to SITC 2020. A simple, proof-of-concept analysis has been used to establish RAFT's ability to support reproducibility by running locally on laptop computers, on multiple research compute clusters, and on the Google Cloud Platform.

Conclusions The RAFT platform shows promising capabilities to support rapid and reproducible research within the field of immuno-oncology. Several features remain in development and testing, such as incorporation of additional immunogenomics feature modules such as variant/fusion detection and HLA/peptide binding affinity estimation. Other functionality in development will enable collaborators to use remote Git repository hosting (e.g. GitHub or GitLab) to jointly and iteratively modify an analysis.

http://dx.doi.org/10.1136/jitc-2020-SITC2020.0485



Abstract 485 Figure 1 Example RAFT Usage
Users define their required inputs, build their analysis, and run their analysis using the RAFT command-line interface. The metadata from the analysis can then be shared through a RAFT package with collaborators or interested third-parties in order to reproduce or expand upon the initial results.



Abstract 485 Figure 2 End-to-end RAFT
RAFT supports end-to-end analysis development through a 'project' structure. Users link local required files (e.g. FASTQs, references or manifests) into their appropriate/raft subdirectory. (1) Projects are initiated using the raft init-project command which creates and populates a project-specific directory. (2–3) Users then load required metadata (e.g. sample manifests or clinical data) and references (e.g. alignment references) into the project using the raft load-metadata or raft load-reference commands, respectively. (4) Modules consisting of tool-specific and topical workflows are cloned from a collection of remote repositories into the project using raft load-module. (5) Specific processes and workflows from previously loaded modules are added to the analysis (main.nf) through raft add-step. Users can then modify main.nf with their desired parameters and execute the workflow using raft run-workflow. (6) Additionally, RAFT allows an iterative approach where results from RAFT can be analyzed and modified through RStudio and re-run through Nextflow.

486 IRECEPTOR PLUS: A DATA INTEGRATION PLATFORM TO SHARE, COMPARE AND ANALYZE ADAPTIVE IMMUNE RECEPTOR REPERTOIRE (AIRR-SEQ) DATA FROM ANTIBODY/B- AND T-CELL REPERTOIRES

Felix Breden*. *Simon Fraser University, Pasadena, CA, USA*

Background Over the past few years, next-generation sequencing technologies have been developed to characterize 'adaptive immune receptor repertoires' (i.e., antibody/B-cell and T-cell receptor repertoires or AIRRs) in exquisite detail. AIRR sequencing (AIRR-seq) has enormous promise for understanding the dynamics of the immune repertoire in vaccinology, infectious diseases, autoimmunity, and cancer biology. While AIRR-seq data is important, it is also very large, complex, and requires specialized tools and services to curate, analyze, and share. In response to these challenges, The AIRR Community was formed in 2015 (www.airr-community.org). The AIRR Community comprises immunologists, immunogeneticists, computer scientists, bioinformaticians, and experts in legal, ethics and IP issues who are developing shared protocols and