Scaling Librarian to the client-server model

Background

- There are many different operations that produce FASTQ files.
  - Chromatin Analysis (ATAC-Seq, ChIP-Seq)
  - Transcriptome Sequencing (total RNA-Seq, mRNA-Seq)
  - Etc.
- Once reduced to a FASTQ file, they may be mislabelled later.

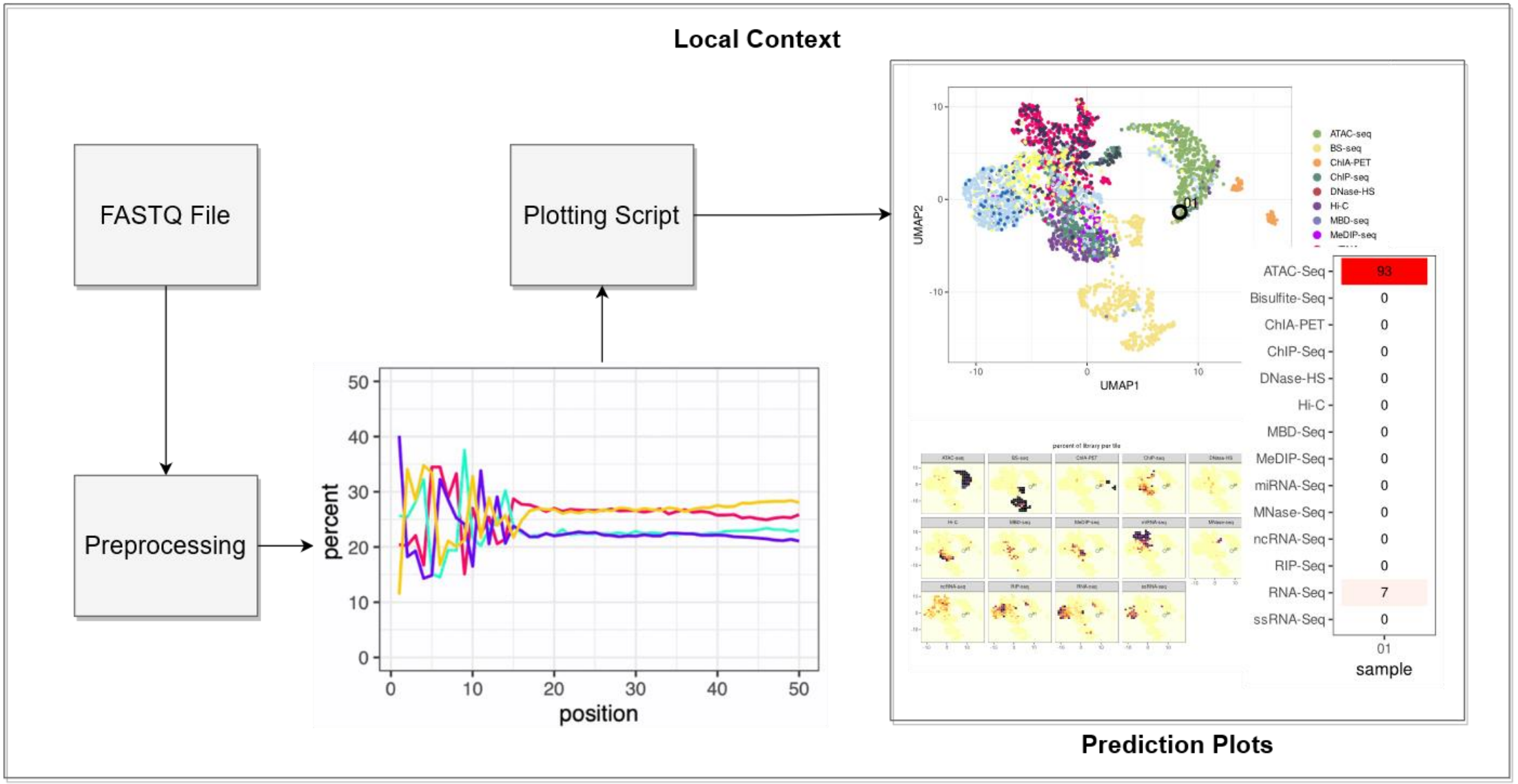
Background

Librarian is a quality-control tool that:

- Takes FASTQ files that contain biological data.
- Outputs predictions of the operations that produced the files.

1. @SEQ_ID
2. GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGT
3. +
4. !''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC
5. …

Background


Local Context
FASTQ File
Plotting Script
Preprocessing
percent
position
UMAP2
UMAP1
ATAC-seq
BS-seq
ChIA-PET
ChIP-seq
DNase-HS
Hi-C
MBD-seq
MeDIP-seq
ATAC-Seq 93
Bisulfite-Seq 0
ChIA-PET 0
ChIP-Seq 0
DNase-HS 0
Hi-C 0
MBD-Seq 0
MeDIP-Seq 0
miRNA-Seq 0
MNase-Seq 0
ncRNA-Seq 0
RIP-Seq 0
RNA-Seq 7
ssRNA-Seq 0
sample
Prediction Plots

Proposal

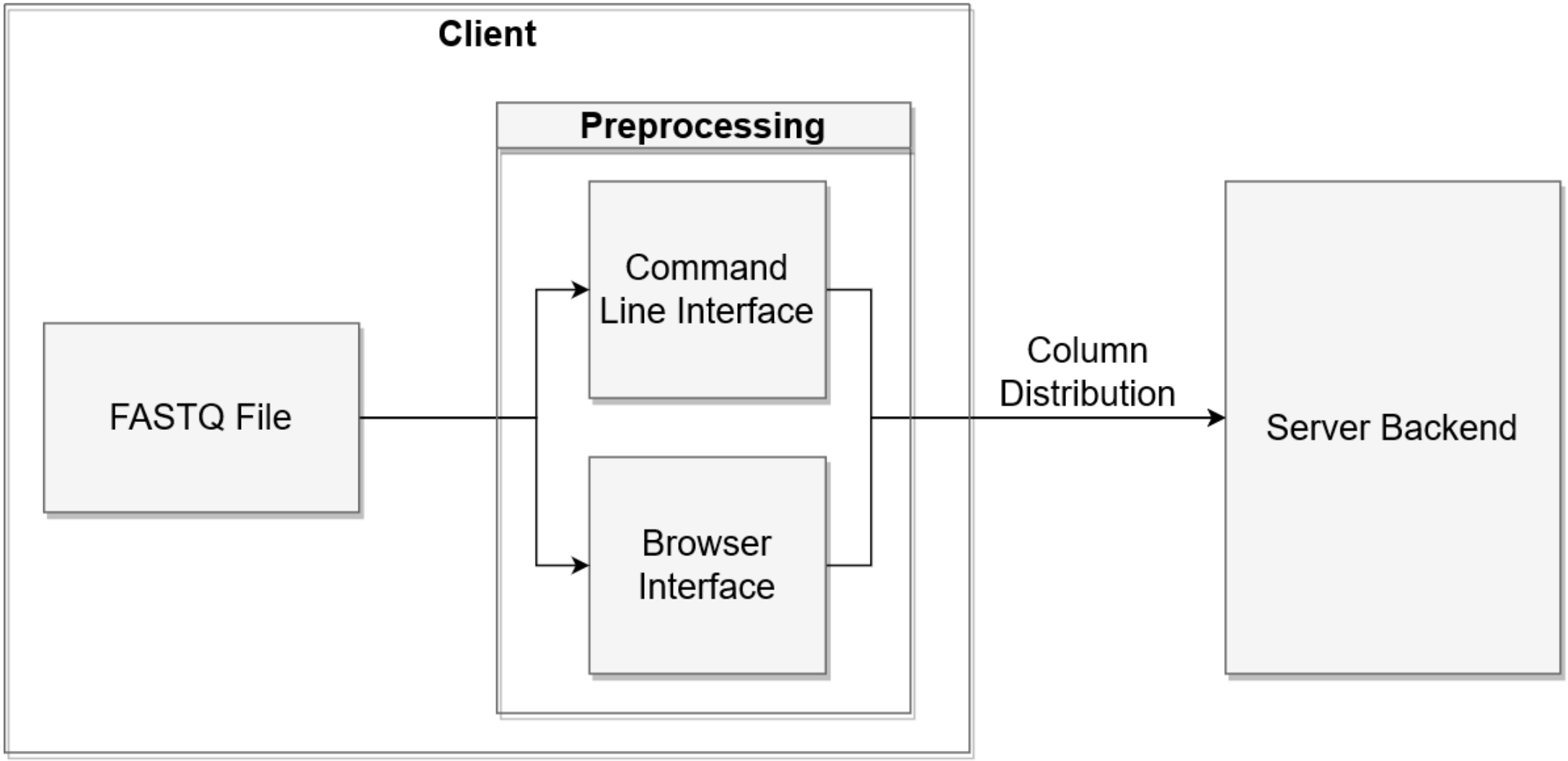
- Want to build web service to analyze user-provided files.

- Submit files with either website or CLI.
  - CLI intended for automated tools and power users.
  - Website serves as demo for first-time users.

Task

- Split local computation pipeline into client-server.
- **Problem:** FASTQ files are large. An issue comment mentions: *“The file [...], is 23GB in size and contains 1,204,065,472 lines.”*
- **Solution:** preprocess file into column distribution on the client. (column distribution is ~6kb for any file)
- **Challenge:** implementing preprocessing on browser and CLI.

Architecture



Challenge: Preprocessing on disparate platforms

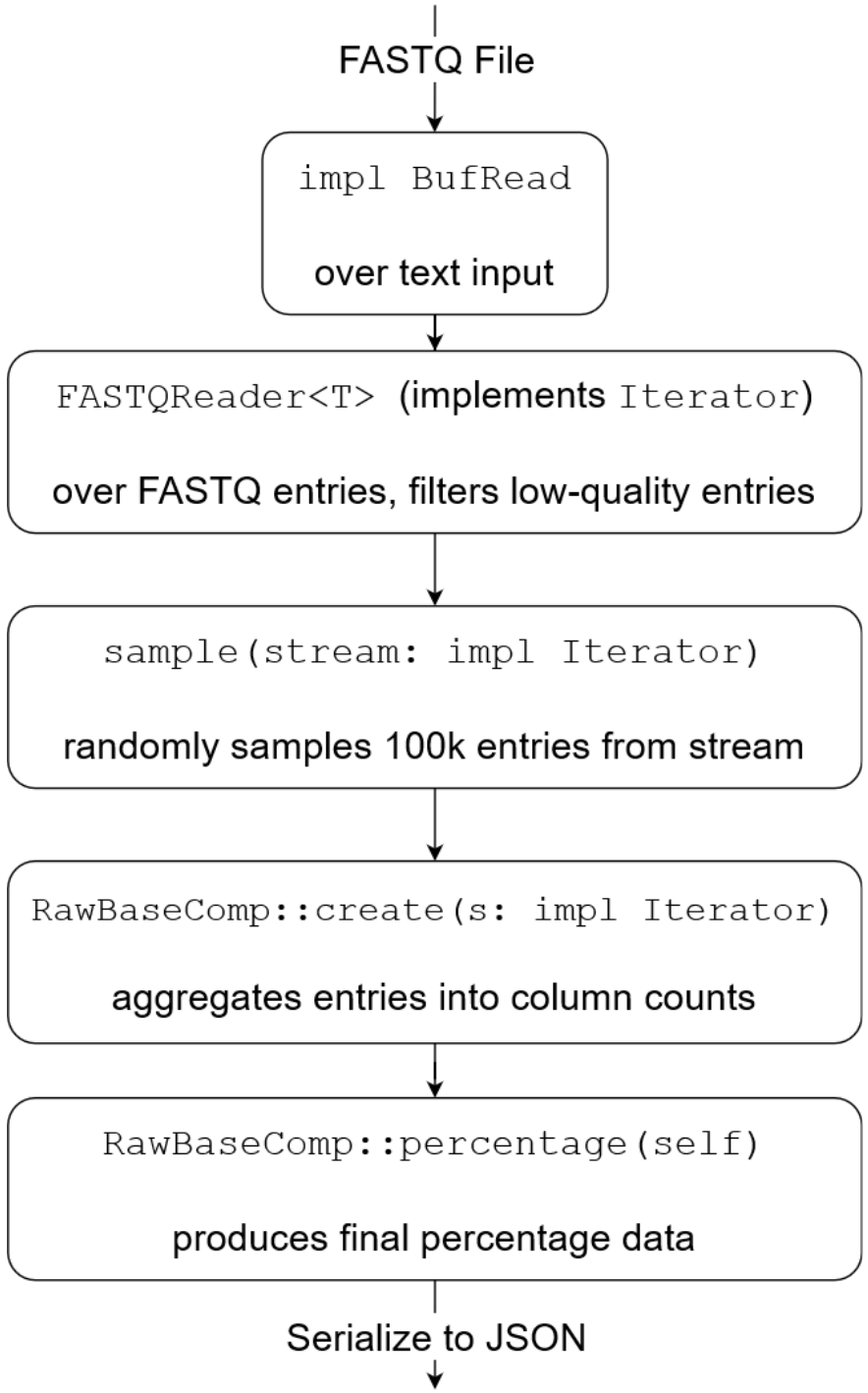
1. Built custom library to preprocess FASTQ file in Rust.
   - Rust is a systems-level programming language (like C, C++).
   - Can be deployed in CLI as-is (CLI also written in Rust).

2. Compile Rust to WebAssembly to use in the browser.
   - WebAssembly is a bytecode format that provides an alternative method of executing code in a browser.
   - JavaScript code can call into the WebAssembly module to run library code.
   - More details later.

Preprocessing Deep Dive

Considerations:

- One solution for multiple interfaces (CLI, browser)
- Scalability to further interfaces.



Preprocessing Deep Dive

As a result, porting to WebAssembly only required rewriting file logic (the highlighted “create_reader” function).

```
let fastq_reader =
        FASTQReader::new(SampleArgs::default(),
                        create_reader(file));
run_json(fastq_reader)
```

WASM Deep Dive

WebAssembly is a bytecode format (like the Java .class format)

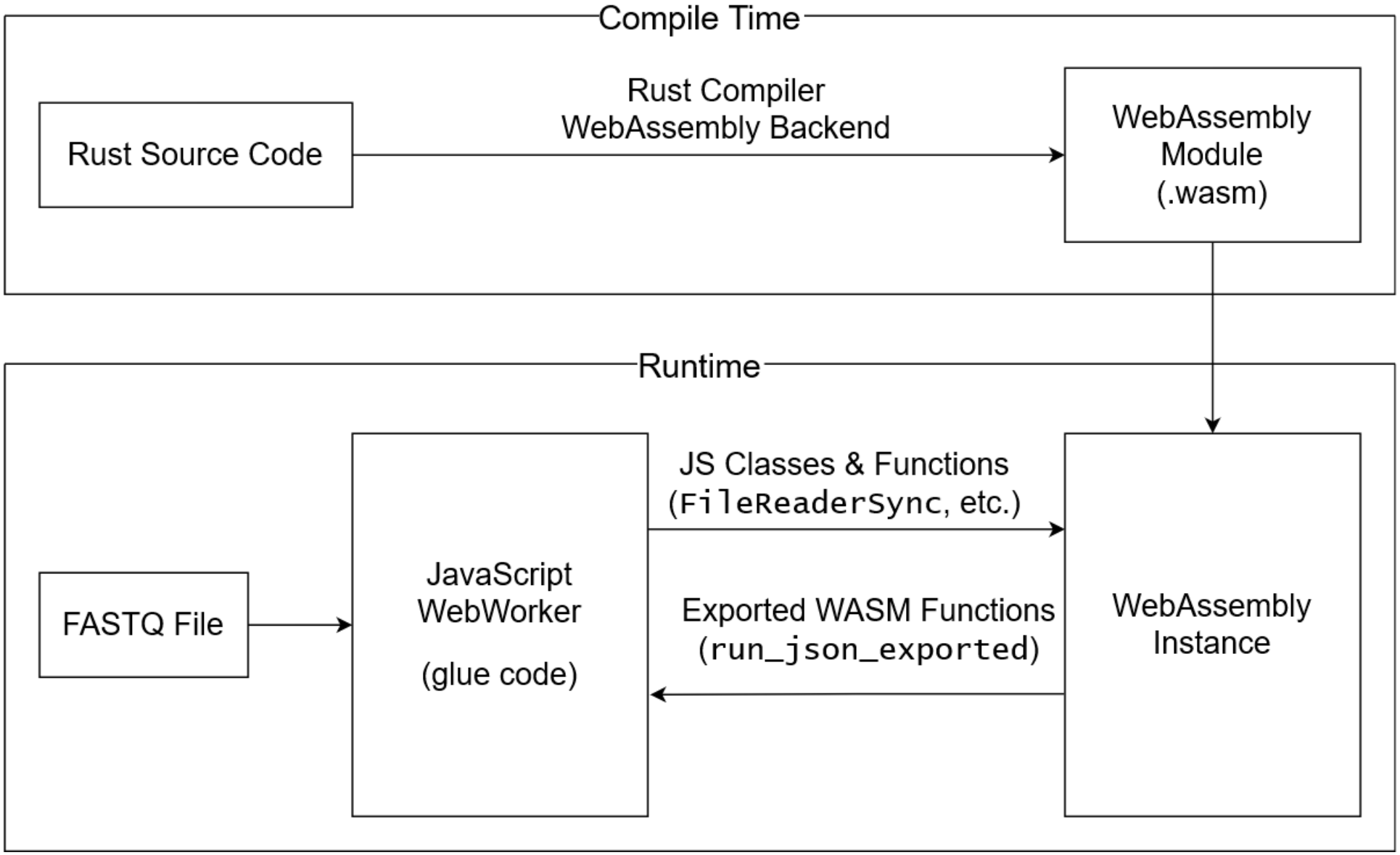
- Runs on a stack-based Virtual Machine with access to linear memory (load/store).
- Think of it as assembly language for an imaginary architecture.

```
(module
    (import "console" "log" (func $log (param i32)))
        (func $main
        ;; load `10` and `3` onto the stack
        i32.const 10
        i32.const 3
        i32.add ;; add up both numbers
        call $log ;; log the result

    ) (start $main)
)
```

- JavaScript can read/write to this linear memory to perform complex inter-op.

WASM Interop Deep Dive



Retroactive: Increasing Performance

- Investigate WebAssembly threading.
- Use AsyncFileReader to request data ahead of time.
- Investigate copying cost; there is currently an instance of a BufReader wrapping a BufReader wrapping a Read object, each of which maintains an internal buffer.

Summary

- Given a local data processing pipeline.
- Separated pipeline into client-server.
- Needed to support both CLI and browser interfaces.
- Built custom library and deployed to both interfaces, with ability to add more.

Impact

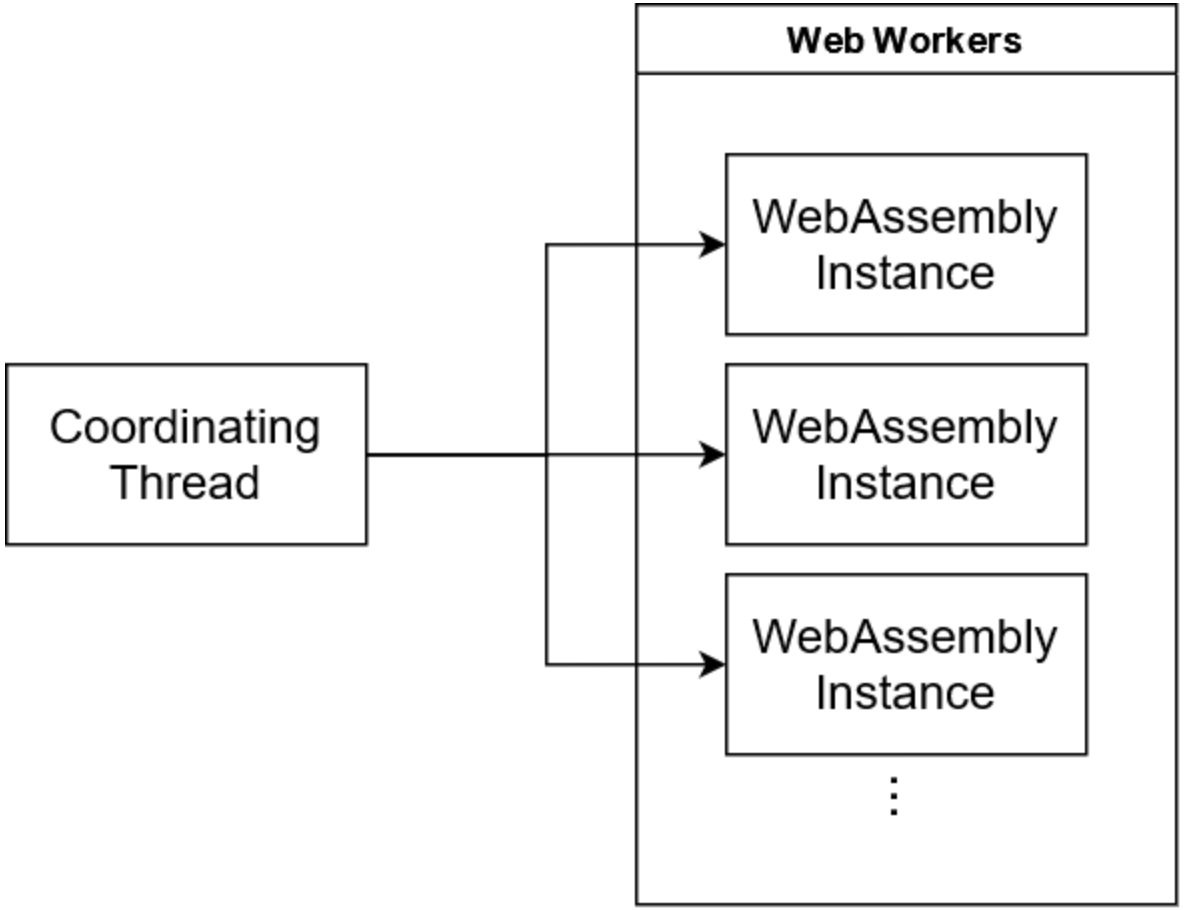
- CLI has a mode to run the rest of the computation locally instead of calling to the server. (i.e., slide 4 architecture)

- St. Jude’s Hospital is integrating the local CLI mode into their quality-control pipeline!

Q&A

WebAssembly Threading

- Library can be easily parallelized (by design).
- WebWorkers run WASM and have their own threads.
- Launching multiple WebWorkers to process portions of the file is possible.



Choose FASTQ file

The reference map is built on Illumina sequencing data from mouse and human from the following library types: ATAC-seq, BS-seq, ChIA-PET, ChIP-seq, DNase-HS, Hi-C, MBD-seq, MeDIP-seq, miRNA-seq, NMase-seq, ncRNA-seq, RIP-seq, RNA-seq and ssRNA-seq. Data obtained from other sequencing technologies, different organisms (in particular those with a substantially different genomic GC content) or other library preparation methods may produce unexpected results.

Browse... ATAC.example.fastq

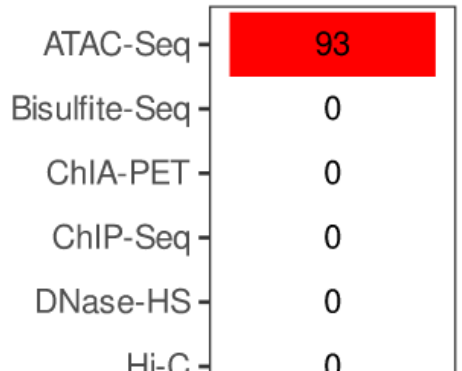
Run analysis

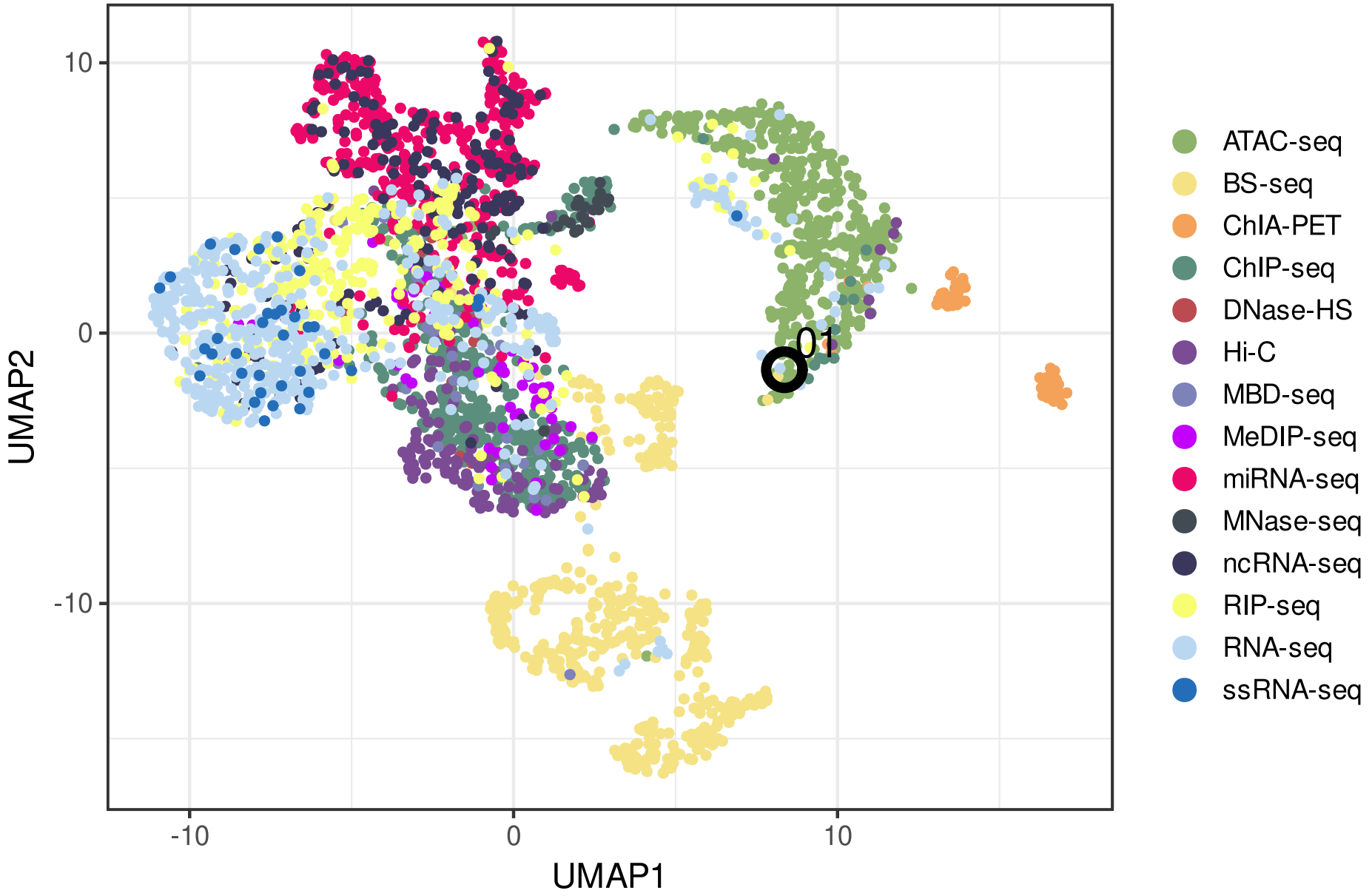
Download **sample FASTQ files** to try Librarian

Results

Download Plots

Sample name	Sample number
ATAC.example.fastq	01




10
0
-10
UMAP2
-10
0
10
UMAP1
01
ATAC-seq
BS-seq
ChIA-PET
ChIP-seq
DNase-HS
Hi-C
MBD-seq
MeDIP-seq
miRNA-seq
MNase-seq
ncRNA-seq
RIP-seq
RNA-seq
ssRNA-seq

```
Librarian CLI 1.3.0
A tool to predict the sequencing library type from the base composition of a supplied FastQ file. Uncompresses .gz files
when reading.

USAGE:
    librarian [FLAGS] [OPTIONS] <input>...

FLAGS:
    -h, --help
            Prints help information

    -l, --local
            Run all processing locally, replacing the need for a server. Requires Rscript and other dependencies to be
            installed, along with the `scripts` folder. See
            https://github.com/DesmondWillowbrook/Librarian/blob/master/cli/README.md for more details.

            This cannot be set together with `--api`.
        --raw
            Only output the librarian_heatmap.txt file used by MultiQC, and don't output any plots.
            This option requires `local` to be set.
    -V, --version
            Prints version information

OPTIONS:
        --api <api>
            Specifies query URL to send prediction request to. Defaults to Babraham Bioinformatic's server. Passed
            argument is given precedence over environment variable.

            This cannot be set together with `--local`. [env: LIBRARIAN_API_URL=]  [default:
            https://www.bioinformatics.babraham.ac.uk/librarian/api/plot_comp]
    -o, --output-dir <output-dir>
            Output directory (eg. `output_dir/`) [default: ]


ARGS:
    <input>...
            List of input files
```

Project Details

- Research Paper
  https://f1000research.com/articles/11-1122/v2

- Repository
  https://github.com/DesmondWillowbrook/Librarian