Galaxy for Bioinformaticians

What is Galaxy

- A **Open-Source** platform that allows to run complex bioinformatics analysis with a graphics interface
- More than 350 public Galaxy resources:
  - 120+ public servers, many more non-public
  - Both general-purpose and domain-specific
- “Easy” to install and maintain local instances

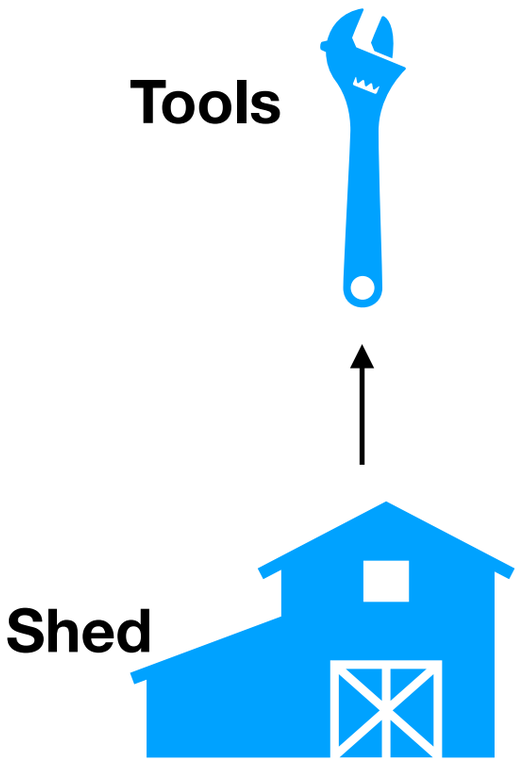
- **Accessibility**
    - Users without programming experience can easily upload/retrieve data, run complex tools and workflows, and visualize data
- **Reproducibility**
    - Galaxy captures information so that any user can understand and repeat a complete computational analysis
- **Transparency**
    - Users can share or publish their analyses (histories, workflows, visualisations)
    - Pages: online Methods for your paper

Data sets

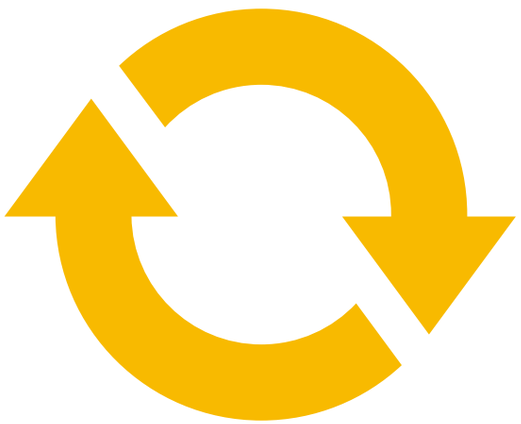
Data sets

Tools

Data sets



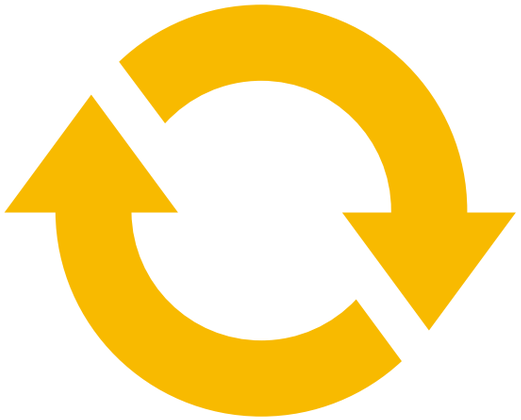
Data sets

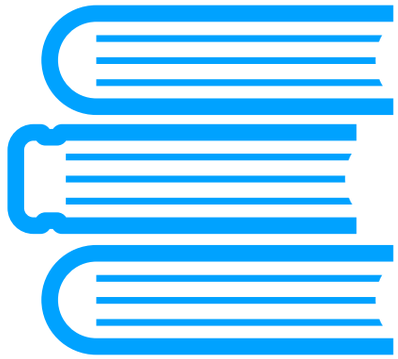


Tools

Shed

Data sets





History

Tools

Shed

Data sets

Tools

Shed

History

Workflow

Data sets

Visualisation

Tools

History

Shed

Workflow

Data sets

Visualisation

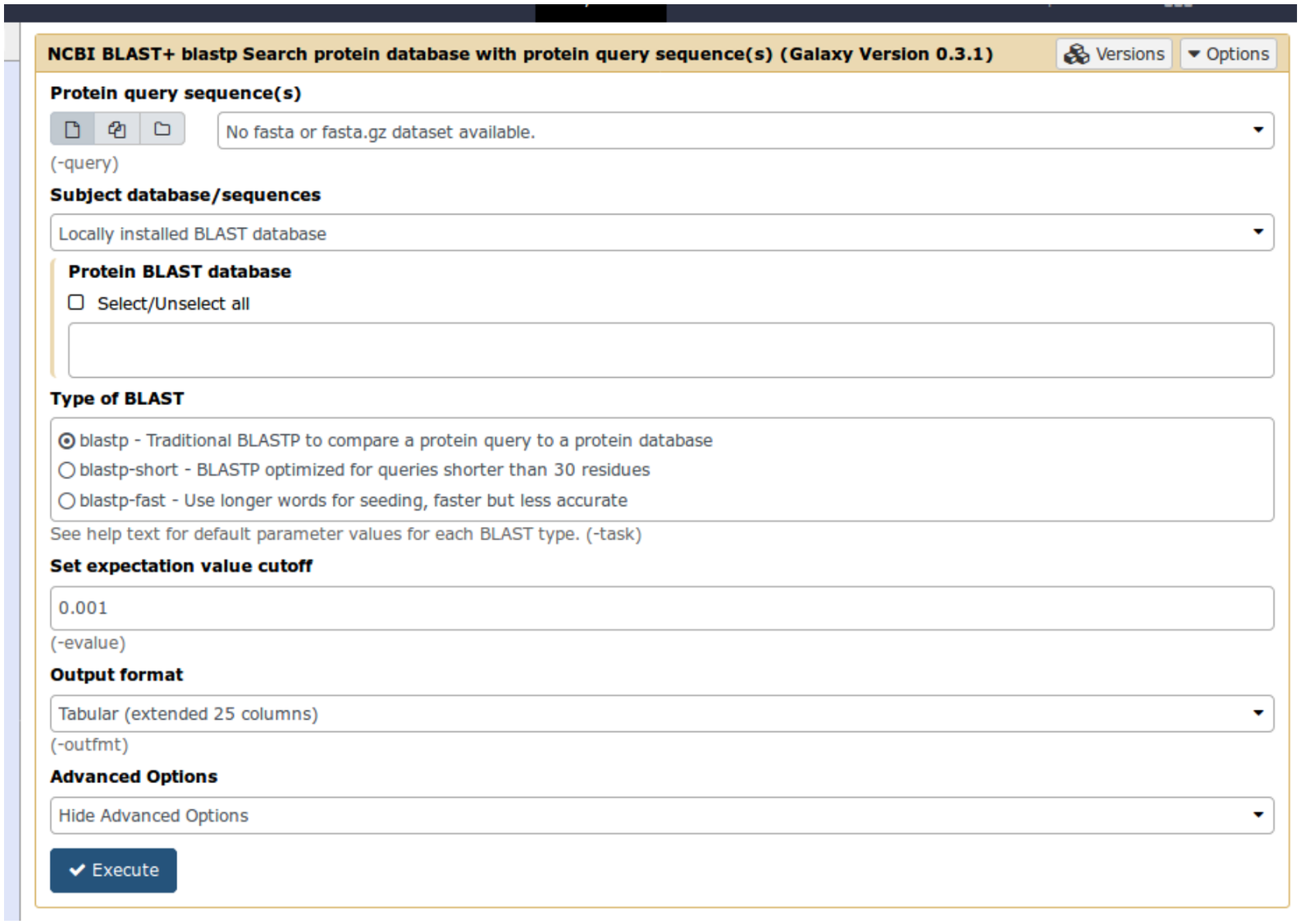
Tools

History

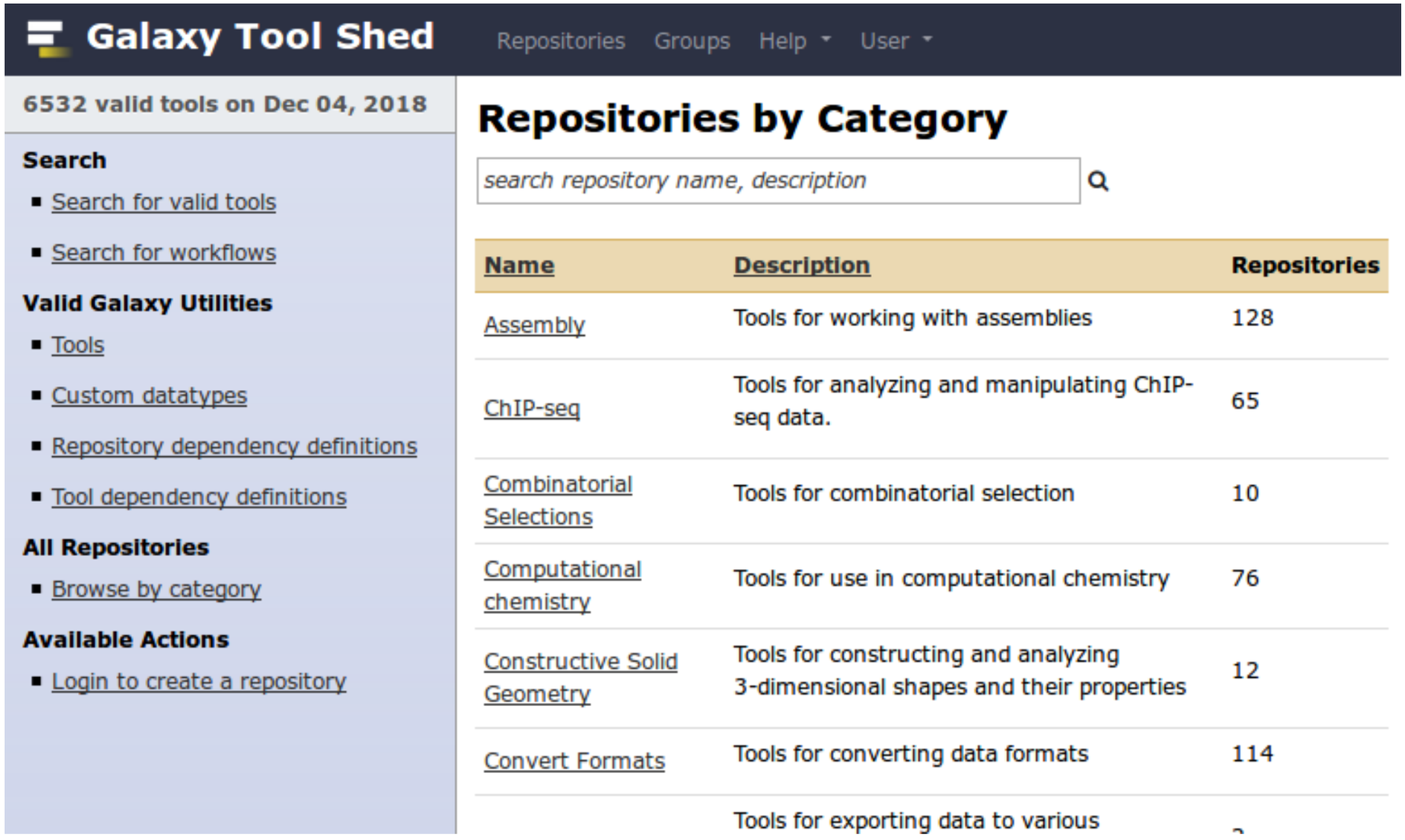
Shed

Workflow

Tools



- A tool form contains:
  - input datasets and parameters
  - help, citations, metadata
  - an Execute button to start a job, which will add some output datasets to the history
- New tool versions can be installed without removing old ones to ensure reproducibility



- Free "app" store: Galaxy Tool Shed
    - Thousands of tools already available
    - Most software can be integrated
    - If a tool is not available, ask the Galaxy community for help!
        - Only a Galaxy admin can install tools

Data

- Copy/paste from a file
- Upload data from a local computer
- Upload data from internet using URL
- Upload data from online databases: UCSC, BioMart, ENCODE, modENCODE, Flymine etc.
- Import from Shared Data (libraries, histories, pages)
- Upload data from FTP

Data types

- Tools only accept input datasets with the appropriate datatypes
- When uploading a dataset, its datatype can be either:
  - automatically detected
  - assigned by user
  - Dataset produced by a tool: datatype assigned by the tool

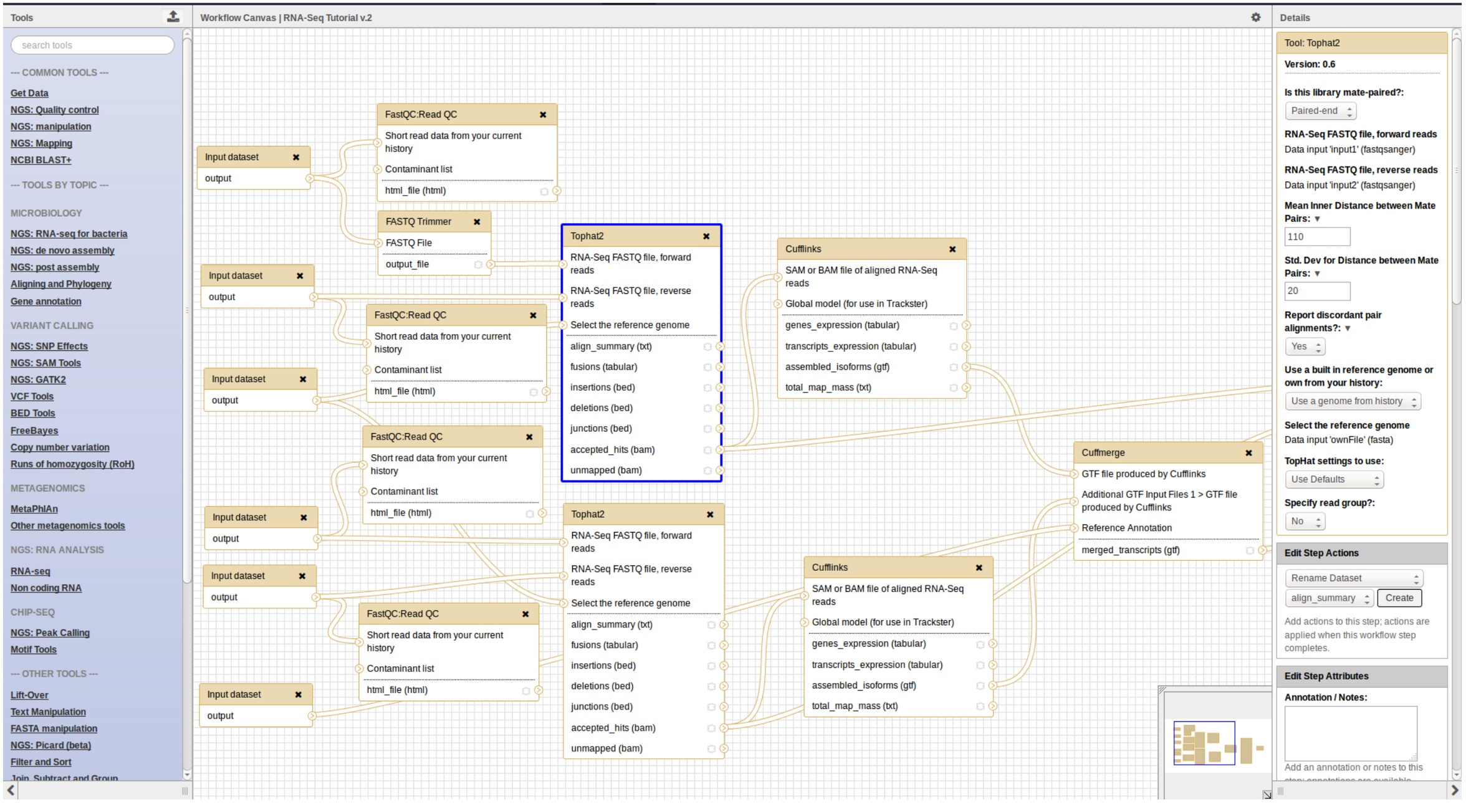
History

- Location of all analyses
  - collects all datasets produced by tools
  - collects all operations performed on the data

Workflows

- Workflows are build by:
  - **Extracting from a history**
  - manually by adding and configuring tools using the canvas
  - Importing an existing **shared workflow**
- Workflows allow:
  - Create sub-workflows: a workflow inside another workflow
  - Share workflows for publication and with the community

Workflows


Tools
search tools
--- COMMON TOOLS ---
Get Data
NGS: Quality control
NGS: manipulation
NGS: Mapping
NCBI BLAST+
--- TOOLS BY TOPIC ---
MICROBIOLOGY
NGS: RNA-seq for bacteria
NGS: de novo assembly
NGS: post assembly
Aligning and Phylogeny
Gene annotation
VARIANT CALLING
NGS: SNP Effects
NGS: SAM Tools
NGS: GATK2
VCF Tools
BED Tools
FreeBayes
Copy number variation
Runs of homozygosity (RoH)
METAGENOMICS
MetaPhlAn
Other metagenomics tools
NGS: RNA ANALYSIS
RNA-seq
Non coding RNA
CHIP-SEQ
NGS: Peak Calling
Motif Tools
--- OTHER TOOLS ---
Lift-Over
Text Manipulation
FASTA manipulation
NGS: Picard (beta)
Filter and Sort
Join, Subtract and Group
Workflow Canvas | RNA-Seq Tutorial v.2
Input dataset
output
FastQC:Read QC
Short read data from your current history
Contaminant list
html_file (html)
FASTQ Trimmer
FASTQ File
output_file
Tophat2
RNA-Seq FASTQ file, forward reads
RNA-Seq FASTQ file, reverse reads
Select the reference genome
align_summary (txt)
fusions (tabular)
insertions (bed)
deletions (bed)
junctions (bed)
accepted_hits (bam)
unmapped (bam)
Cufflinks
SAM or BAM file of aligned RNA-Seq reads
Global model (for use in Trackster)
genes_expression (tabular)
transcripts_expression (tabular)
assembled_isoforms (gtf)
total_map_mass (txt)
Cuffmerge
GTF file produced by Cufflinks
Additional GTF Input Files 1 > GTF file produced by Cufflinks
Reference Annotation
merged_transcripts (gtf)
Details
Tool: Tophat2
Version: 0.6
Is this library mate-paired?:
Paired-end
RNA-Seq FASTQ file, forward reads
Data input 'input1' (fastqsanger)
RNA-Seq FASTQ file, reverse reads
Data input 'input2' (fastqsanger)
Mean Inner Distance between Mate Pairs:
110
Std. Dev for Distance between Mate Pairs:
20
Report discordant pair alignments?:
Yes
Use a built in reference genome or own from your history:
Use a genome from history
Select the reference genome
Data input 'ownFile' (fasta)
TopHat settings to use:
Use Defaults
Specify read group?:
No
Edit Step Actions
Rename Dataset
align_summary
Create
Add actions to this step; actions are applied when this workflow step completes.
Edit Step Attributes
Annotation / Notes:
Add an annotation or notes to this

Example: ATAC Seq

1. Upload libraries

2. Check Quality

3. Map to human genere

4. Filter out low quality alignments, unpair reads, and mitochondrial reads

5. Remove duplicates

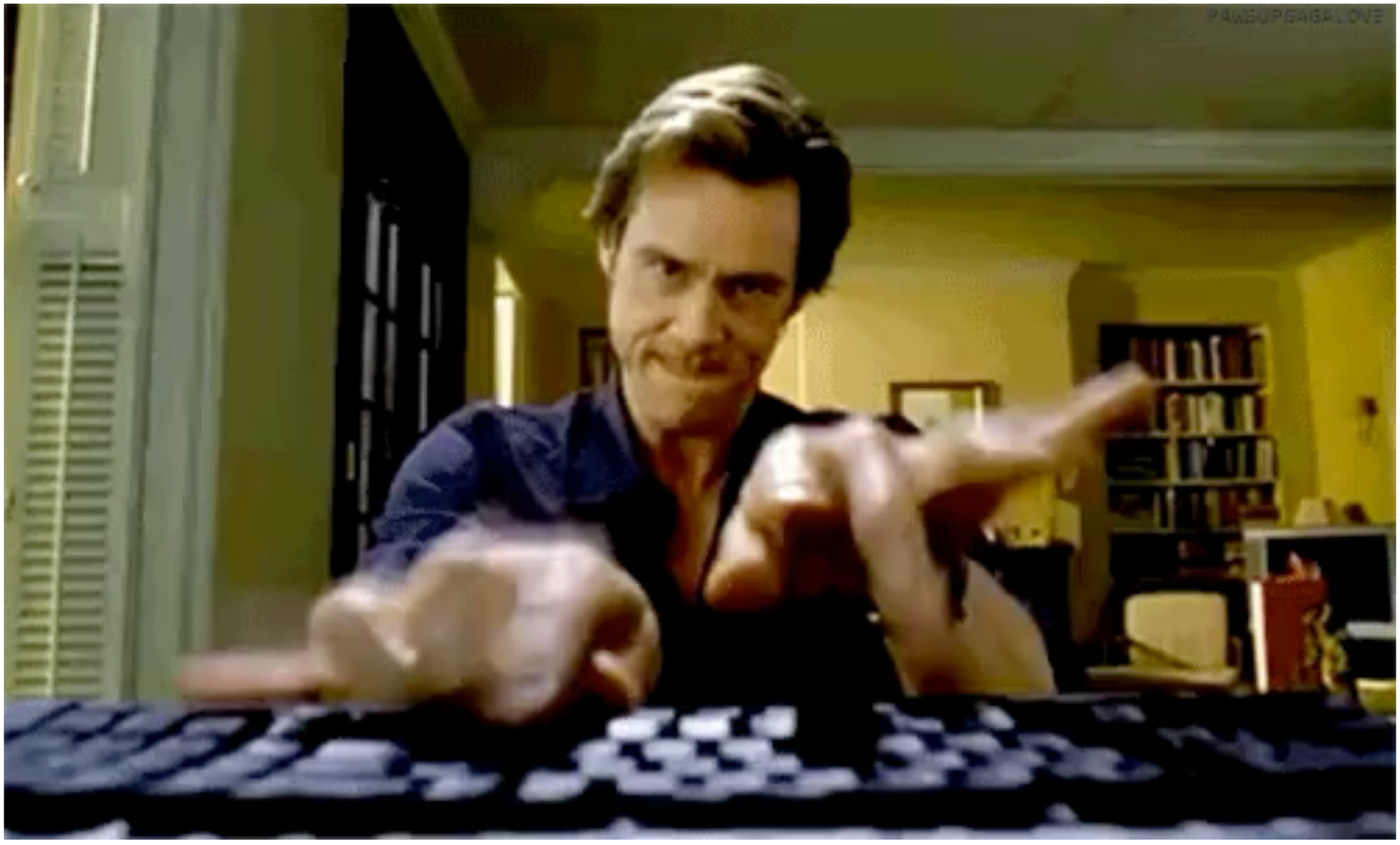
6. Visualise insert size (QC)

7. Convert BAM to BED

8. Call Peaks with MACS2

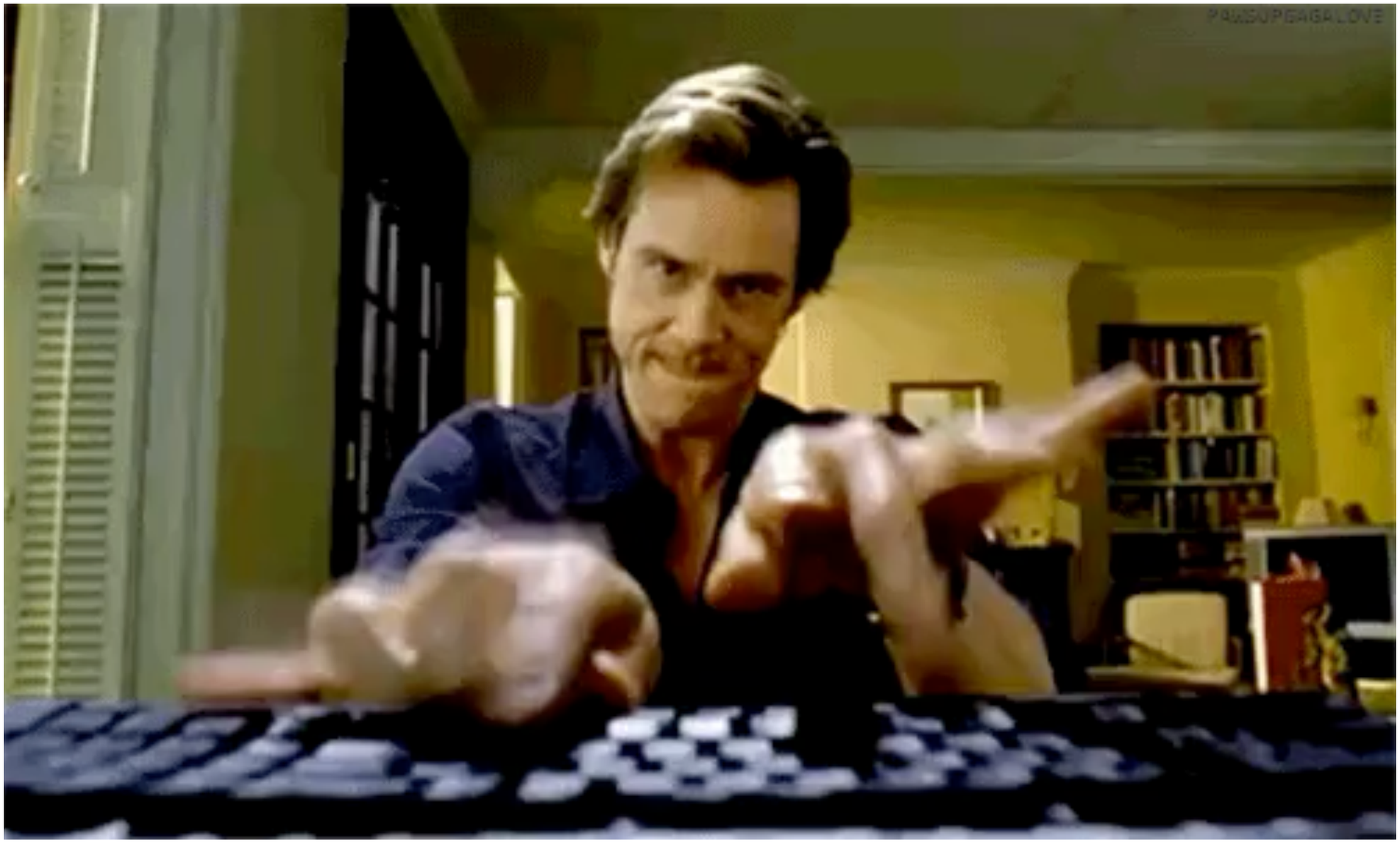
9. Convert BED to BigWig

10. Visualise Peaks in genome context



But… Who cares?

We can write code



But… Who cares?

We can write code

We still can benefit from…

- **Reproducibility**
  - For each produced dataset the history tracks:
  - name, format, size, creation time, datatype-specific metadata
  - tool id, version, inputs, parameters
  - standard output (stdout) and error (stderr)
- **Automatisation via Workflows**
  - Re-run the same analysis on different input data sets
  - Change parameters before re-running a similar analysis
  - Make use of the workflow job scheduling

The Galaxy API

“In addition to being accessible through a web interface, Galaxy can also be accessed programmatically, through shell scripts and other programs. The web interface is appropriate for things like exploratory analysis, visualisation, construction of workflows, and rerunning workflows on new datasets.

The web interface is less suitable for things like:

- Connecting a Galaxy instance directly to your sequencer and running workflows whenever data is ready
- Running a workflow against multiple datasets (which can be done with the web interface, but is tedious)
- When the analysis involves complex control, such as looping and branching.

The Galaxy API addresses these and other situations by exposing Galaxy internals through an additional interface, known as an Applications Programming Interface, or API.”

https://galaxyproject.org/develop/api/
https://docs.galaxyproject.org/en/latest/api_doc.html

we can code

we can code

Bindings

- The Galaxy code itself contains JavaScript bindings.
- BioBlend contains a set of Python bindings
- blend4j contains Java bindings largely modeled after BioBlend. JavaDocs
- blend4php contains PHP bindings.
- CLI Bindings: Parsec

CLI Bindings: Parsec

- https://github.com/galaxy-iuc/parsec/
- parsec is a set of wrappers for BioBlend's API. It builds a set of small, useful utilities for talking to Galaxy servers. Each utility is implemented as a subcommand of `parsec`

- https://parsec.readthedocs.org

Example

```
> parsec init

> parsec histories get_histories

> parsec histories show_history b381997063dbe2e8

> parsec workflows get_workflows

> parsec workflows show_workflow fee63819c4abec0e
```

Example

```
> HISTORY_ID=$(parsec histories create_history | jq .id)

> parsec histories update_history --name 'Parsec test’ $HISTORY_ID

> parsec tools upload_file my_file_R1 $HISTORY_ID

> parsec tools upload_file my_file_R2 $HISTORY_ID

> # create json input file

> WORKFLOW_ID=$(parsec workflows invoke_workflow  fee63819c4abec0e \
   --inputs inputs.json \
   --history_id $HISTORY_ID \
  | jq .workflow_id)

> parsec utils wait_on_invocation $WORKFLOW_ID
```

But… There are many workflow managers

- steeper learning curve

- You need to integrate the tools yourself, and take care off all data typing and compatibility

- Hard to share with other non-techie colleagues