# BUILDING EVERYTHING WITH NOTHING

Harnessing Nix for Bioinformatics

László Kupcsik

2026-01-31

# MY SETUP

```
❯ R
fish: Unknown command: R

❯ python
fish: Unknown command: python

❯ blastn
fish: Unknown command: blastn

❯ bowtie2
fish: Unknown command: bowtie2

❯ kallisto
fish: Unknown command: kallisto
```

```
~
❯ cd ~/nix_presentation/

nix_presentation via ⊕ v1.7.32 via R v4.5.1
❯ Rscript -e "getRversion()"
[1] ‘4.5.1’
❯ cd ~/aoc2025/day1

aoc2025/day1 via R v4.5.2
❯ Rscript -e "getRversion()"
[1] ‘4.5.2’
❯ , blastn -version
blastn: 2.14.1+
```

# THE JOURNEY

- Using R for plotting / statistics
- Self-contained scripts
- Define environment with {renv/rv}
- Use docker + CWL for non-R workflows
- Let’s use docker for analysis environments too!

# THE DOCKER MAZE

- `FROM image` → dependency maze
- Doesn’t scale well
- Doesn’t compose well

**Directory Tree**

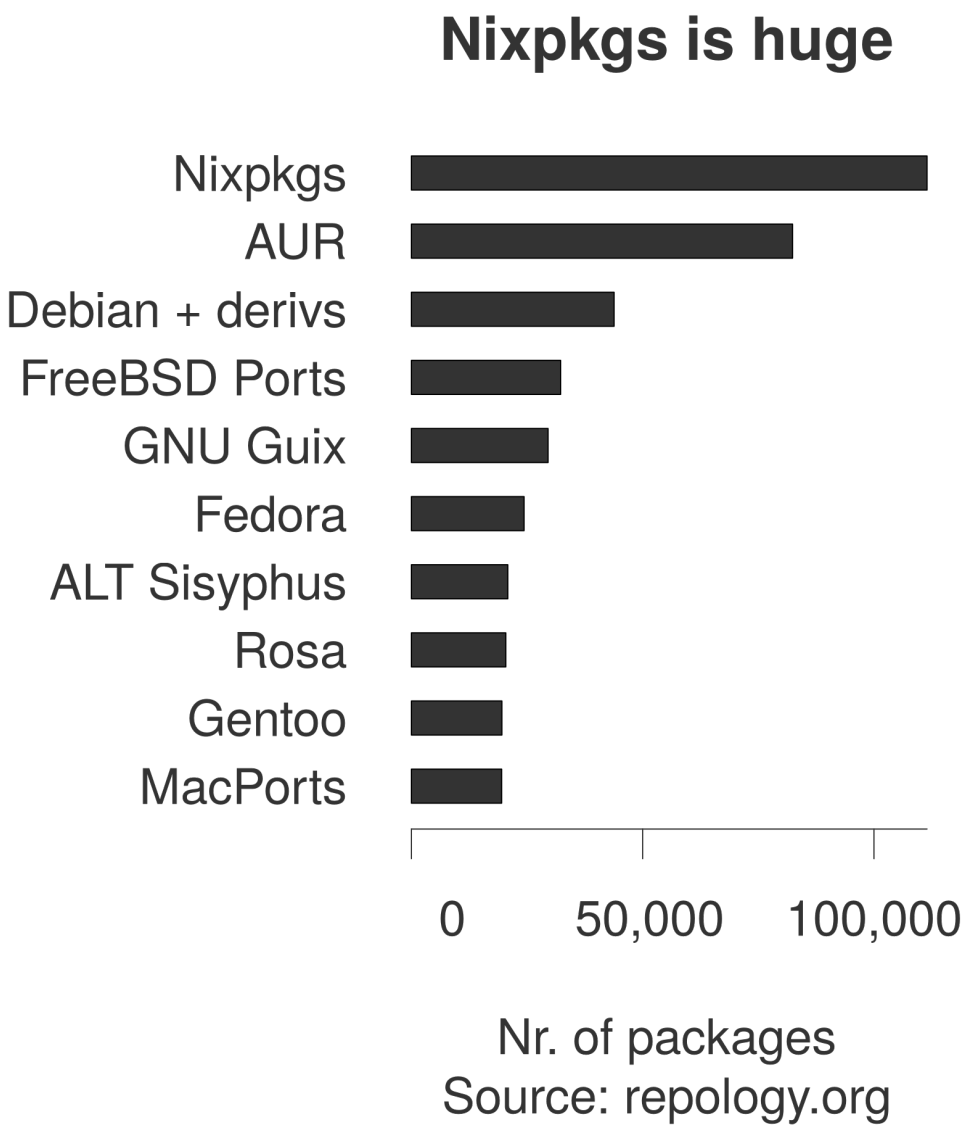
```
bioinf-envs/
├── base-python
│   ├── Dockerfile
│   └── install_scripts
│       ├── env_base.yml
│       └── install_python_base.sh
├── base-r
│   ├── Dockerfile
│   └── install_scripts
│       ├── install_bioinf_base.sh
│       ├── install_custom_packages.sh
│       └── install_r_base.sh
├── bulk-onco
│   ├── Dockerfile
│   └── install_scripts
│       └── install_bulk_onco.sh
├── bulk-rnaseq
│   ├── Dockerfile
│   └── install_scripts
│       └── install_bulk_rnaseq.sh
├── flow-cytometry
│   ├── Dockerfile
│   └── install_scripts
│       └── install_flow_cytometry.sh
├── minimal
│   ├── Dockerfile
│   └── install_scripts
```

# WHAT IS NIX?

- A package manager and build system
- Declarative, *truly* reproducible
- Manages all dependencies
- Isolated, atomic and immutable builds
- A life-changer
- Also a language and an OS (btw)

# HOW DOES NIX MANAGE PACKAGES?

- Monorepo of build recipes
- Builds to `/nix/store`
- Prepending unique hash
- On-demand linking
- Caching: local and remote



# R IN NIX

- CRAN, Bioconductor are part of Nixpkgs
- Follows R and Bioconductor release cycles
- Helpful maintainer community (Matrix)
- Unofficial daily snapshots: rstats-on-nix
- Multiple IDEs: R, radian, rstudio, *vim, *emacs, *codium, …

> @jbedo, @tomasajt @b-rodrigues @kupac @baumann-philipp et al.

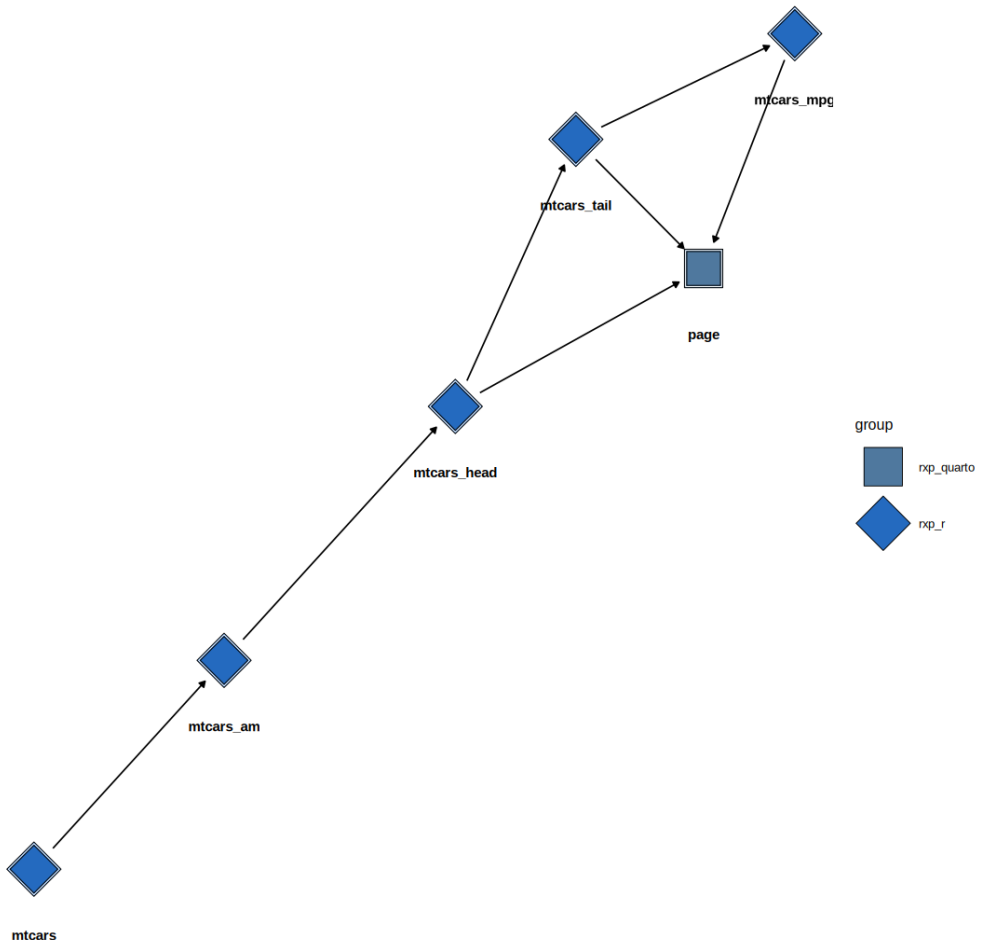
# “WHAT IF I ONLY KNOW R?”

- {rix} is for you
- Define R and package versions
- Use custom packages from git hosting or ROpenSci
- Convert renv with `renv2nix()`
- Supports various IDE-s

```r
library("rix")

# Path to your project
# Creates: .Rprofile, default.nix
my_path <- "."

rix(
  r_ver = "4.3.3",
  r_pkgs = c("dplyr", "ggplot2"),
  system_pkgs = NULL,
  git_pkgs = NULL,
  ide = "code",
  project_path = my_path,
  overwrite = TRUE
)
```

# HOW ABOUT WORKFLOWS?

- Inputs → recipe → outputs
- Software product ≈ Data product
- Is Nix also a workflow manager?
- Examples: Bionix, {rixpress}
- But:
  - No resource management
  - No built-in permission management
  - Everything is copied to `/nix/store`



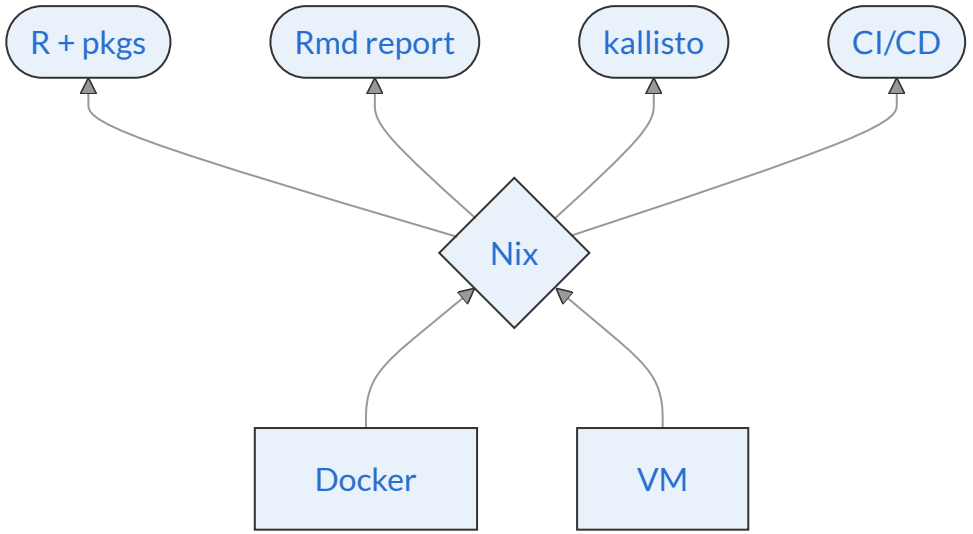

Workflow DAG by {rixpress}

# DOCKER – NIX


```
{
  description = "Nix docker demo";

  inputs = {
    nixpkgs.url = "github:NixOS/nixpkgs/nixpkgs-uns
  };

  outputs = { self, nixpkgs }:

  let
    system = "x86_64-linux";
    pkgs = nixpkgs.legacyPackages.${system};
    dock = pkgs.dockerTools;
  in

  {

    packages.x86_64-linux.kallisto = pkgs.kallisto;

    packages.x86_64-linux.docker =
      dock.buildLayeredImage {
        name = "kallisto";
        tag = pkgs.kallisto.version;
        contents = with pkgs; [
          kallisto
```

# DOCKER – NIX


Docker
Nix
Docker
Nix


R + pkgs
Rmd report
kallisto
CI/CD
Nix
Docker
VM

# THE BENEFITS

- Peace of mind (reporoducibility)
- Your environment as code
- Collaborations, CI/CD, containers, anywhere!
- Be part of the community, contribute
- Learn something new and awesome

# THANK YOU!

# ADVICE

- Add your own trusted cache
- Take care not to add data/creds to nix store
- Forget Dockerfile, nix → container image
- For R, use {rix}
- For simple workflows -> try {rixpress}

# RESOURCES

- Packages: Search
- Development: gh://NixOS/nixpkgs
- Documentation: NixOS Wiki
- Alternative: Guix
- R: {rix}
- Python: devenv, Jupyenv
- Kubernetes: Kubenix
- Me :) bioinfo@biobits.be

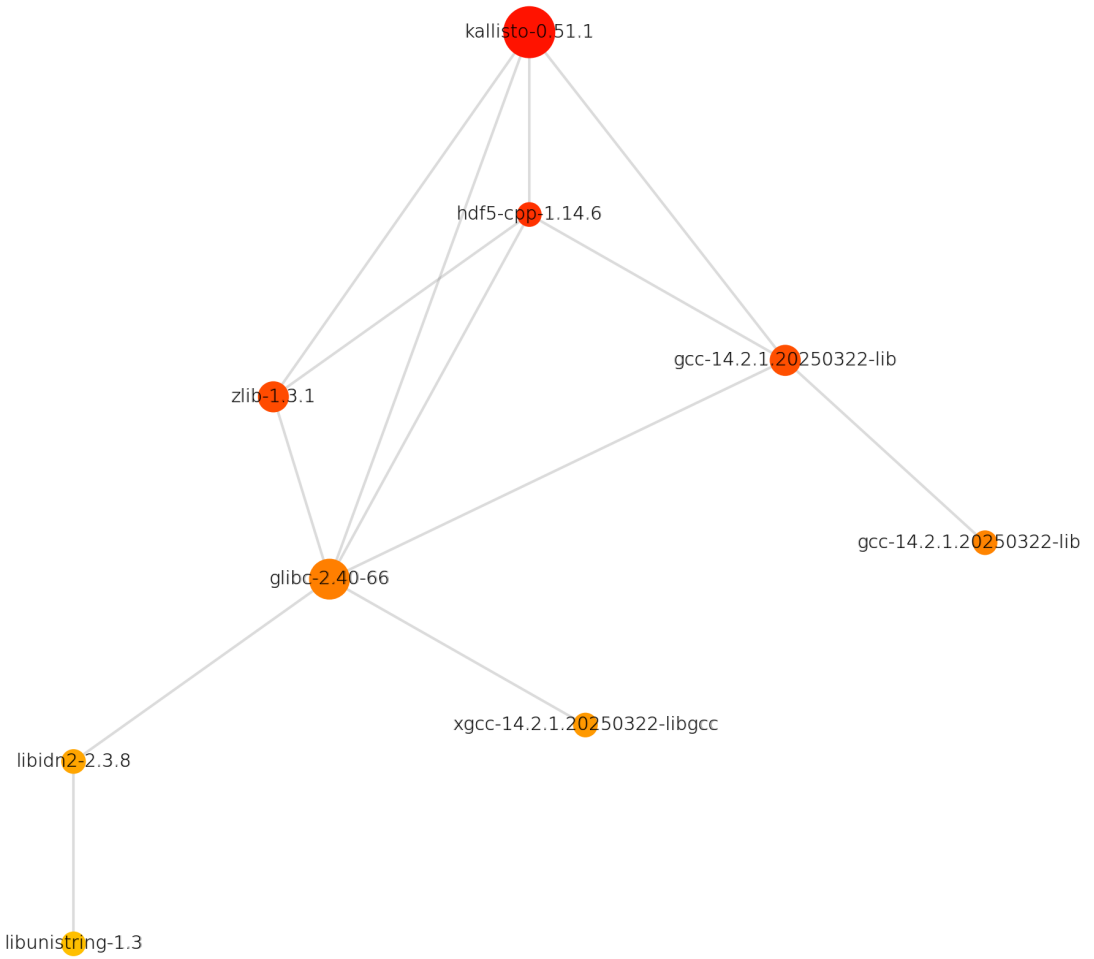
# NIX STORE

```
> ls -U --hide "*.drv" /nix/store | less

...
y3q303knk4vylr15bq0ry9a0wa62653d-R-4.5.1
b2nry70yph8p60yp8xc0jpr76v9dhj3h-python3.12-boto3-1.36.21
gvmw57lc866p7lrlfpxywzd5dbf2hvz1-python3.13-pexpect-4.9.0
mzh3dbzzl2izzgm47i3gf1w36zriknib-python3.13-h11-0.16.0
v7ihfx32zv7bdha6i9dd6a3r0knzs8j3-setup-hook.sh
0slin6ikfi1il6y2s06vrmc2iz2hzx4p-python3.12-rfc3986-validator-0.1.1
skn98v8krqlqnxjzxvvjd1xigf9cyl6h-postgresql-16.9-man
a0zi6y22s899sbvwaf7a1ivak1kvcsr0-r-boot-1.3-31
...
```

# SOME CONS

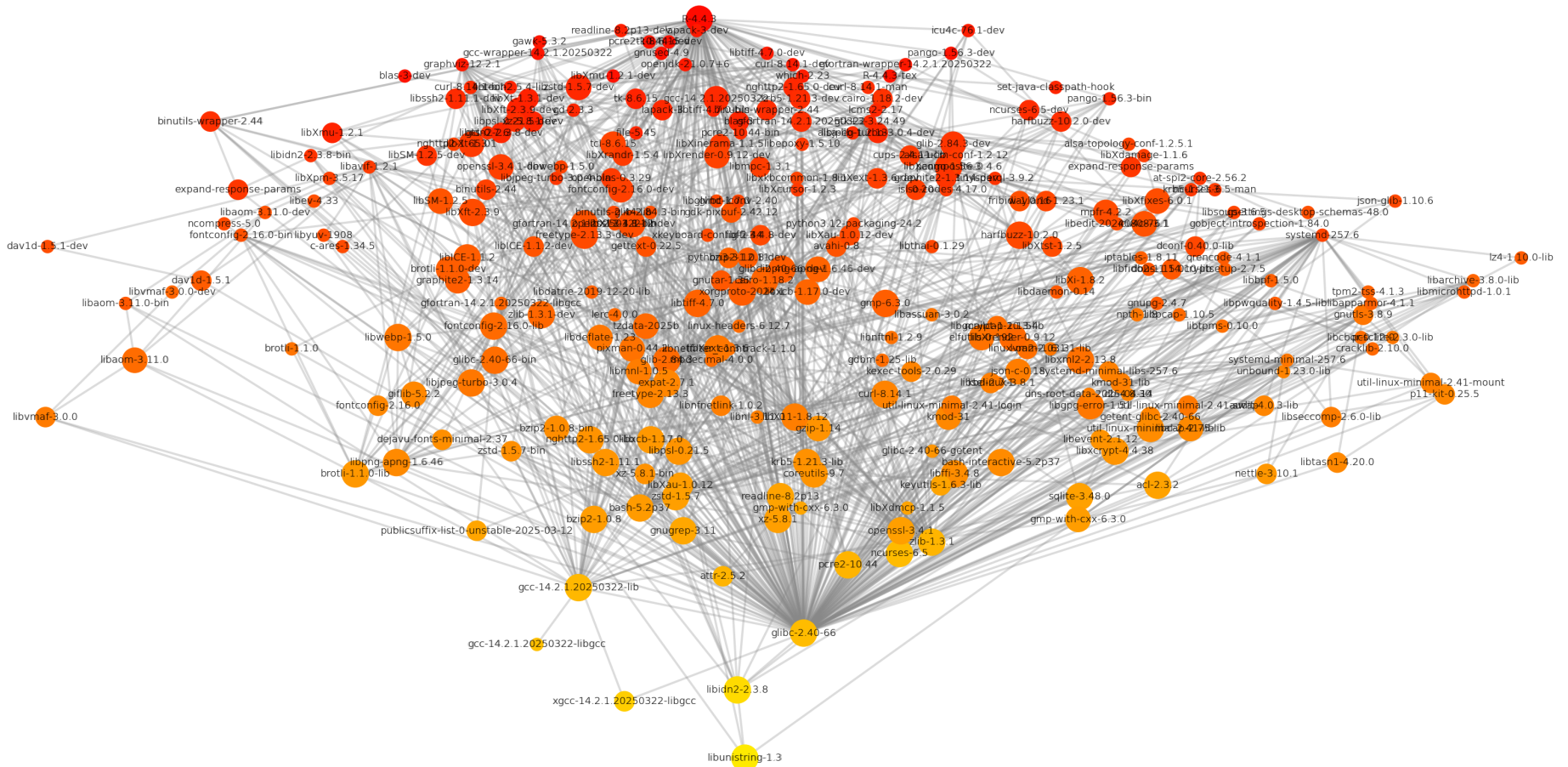
- The language
- Learning curve: conventions, tooling, “implicit” documentation
- Disk space (gc FTW)
- Flakes are “experimental”
- Not all R packages are cached

# SOFTWARE IS COMPLEX - KALLISTO




github:craigmbooth/nix-visualize

# SOFTWARE IS COMPLEX - R




github:craigmbooth/nix-visualize