



Real-world Reproducibility

Lessons learned from implementing a GWAS pipeline

Felix Thalén

felix.thalen@cardio-care.ch

Medizincampus Davos

Cardio-CARE

Basel, 12.04.2024

Disclaimer

This talk concerns reproducibility from a data-centric point of view.
The material discussed is not necessarily applicable to a wet lab.

Cardio-CARE

- ▶ Location: Davos
- ▶ Non-profit, funded by the Kühne Foundation
- ▶ Improve cardiovascular disease diagnosis and prognosis
 - ▶ **Whole-genome sequencing**
 - ▶ Clinical studies



The Cardio-CARE team

Hamburg City Health Center

- ▶ Population-based cohort study
- ▶ 45,000 participants planned
- ▶ Age 45-74
- ▶ Rich phenotypic data
- ▶ Sequencing data from 9,000 samples



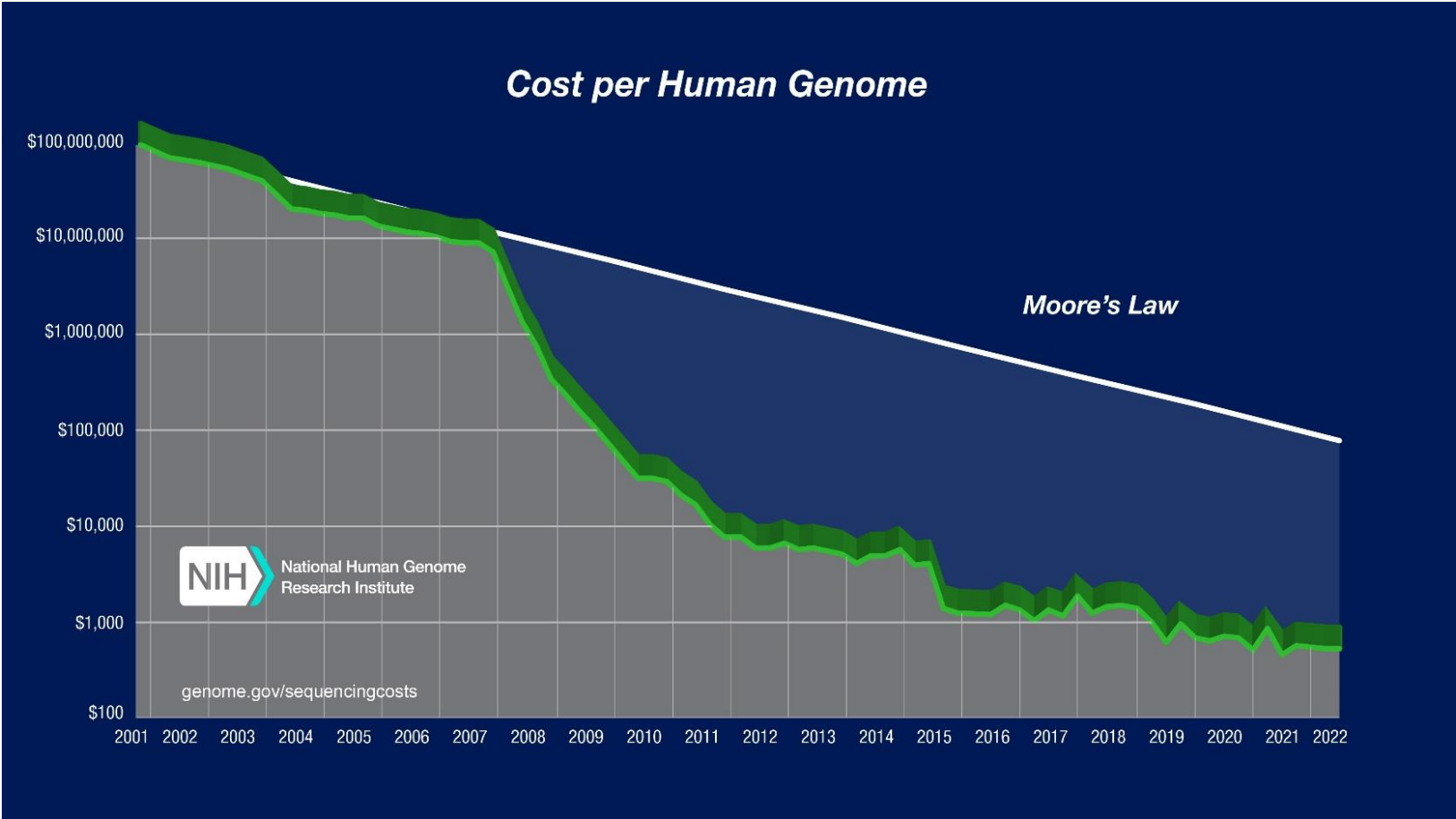
Source: hchs.hamburg

Our environment

- ▶ Petabytes worth of sequencing data
 - ▶ ~65 GB / sample
- ▶ Highly-flexible environment
 - ▶ DRAGEN updated every 6 months
 - ▶ New reference genomes
 - ▶ Other software updates



Decreasing sequencing costs



Source: National Human Genome Research Institute.

Why reproducibility?

Nature survey:

- ▶ >70% failed to reproduce others' experiment
- ▶ >50% failed to reproduce their own

- ▶ Computational sciences well-equipped for reproducibility
- ▶ Large biological datasets are different
- ▶ New technologies aid reproducibility



Reproducibility vs. repeatability

- ▶ **Reproducibility:** consistent results using same input
- ▶ **Repeatability:** consistent results across studies



Tech stack @ Cardio-CARE

Name	Type	Use case
Git	Version control	Track source code changes
Nextflow	Workflow management	Create + run workflows, control environment
Apptainer/Singularity	Containerization	Run containers
Quarto/Markdown	Documentation	Create reports

General strategy

- ▶ Automate
- ▶ Reduce dependencies
- ▶ Time-tested tools
- ▶ Maintenance strategy

Ensuring data integrity

- ▶ Data corruption possible during transfers
- ▶ Checksums verify data integrity
- ▶ **MD5** most common checksum algorithm
- ▶ **SHA-1** newer and more secure than MD5

Workflow management systems (WFMS)

- ▶ Reproducible
- ▶ Scalable
- ▶ Portable

- ▶ Examples: **Nextflow**, Snakemake

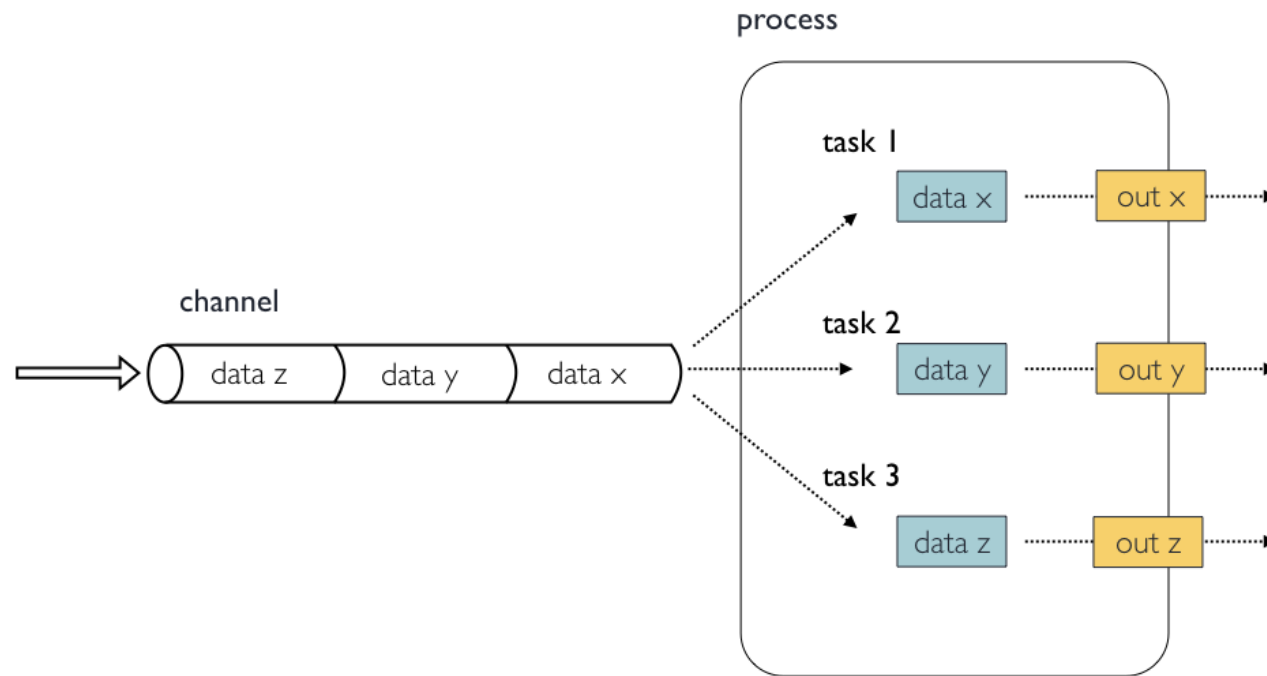
Nextflow

- ▶ Connects command-line tools
- ▶ HPC + cloud
- ▶ Container management
- ▶ Parallelization
- ▶ Resume long jobs

Nextflow

Processes and channels

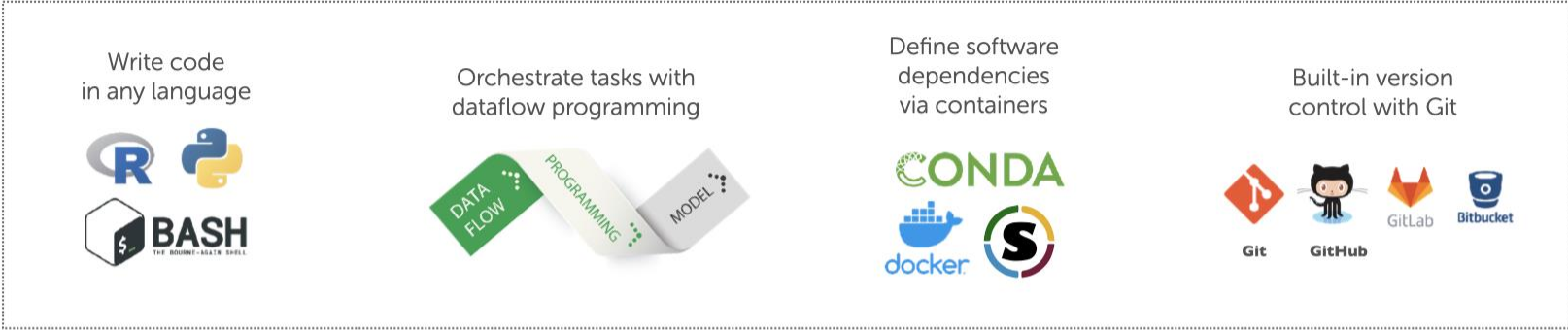
- ▶ Processes define tasks
- ▶ Communicates via asynchronous FIFO queues



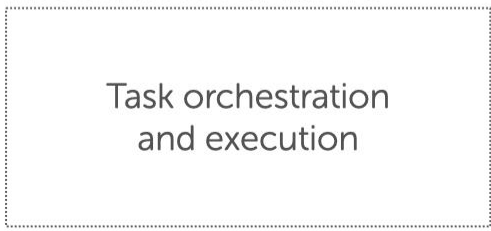
Nextflow

Execution model

nextflow pipeline



nextflow runtime



Supported Platforms



Nextflow

Parameter validation

The screenshot shows the Nextflow pipeline schema builder interface. The browser address bar indicates the URL: https://oldsite.nf-co.re/pipeline_schema_builder?id=1712757557_476615d9338a. The navigation menu includes Home, Pipelines, Modules, Tools, Docs, Events, and About. A green button labeled "Join nf-core" is visible in the top right.

The main interface features a toolbar with buttons for "Add parameter", "Add group", "Collapse groups", "Expand groups", and "Back to top". A "Finished" button with a checkmark is also present.

The central area displays a table of parameters for the "Input/output options" group. The table has columns for ID, Description, Type, Default, Required, and Hide. The "Required" column for the "outdir" parameter is checked.

ID	Description	Type	Default	Required	Hide
phenotypes_file	Phenotypes file	string	Default	<input type="checkbox"/>	<input type="checkbox"/>
covariates_file	Covariates file	string	Default	<input type="checkbox"/>	<input type="checkbox"/>
plink_files_dir		string	Default	<input type="checkbox"/>	<input type="checkbox"/>
chromosomes		string	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18	<input type="checkbox"/>	<input type="checkbox"/>
outdir	The output directory where the results will	string	Default	<input checked="" type="checkbox"/>	<input type="checkbox"/>
email	Email address for completion summary.	string	Default	<input type="checkbox"/>	<input type="checkbox"/>

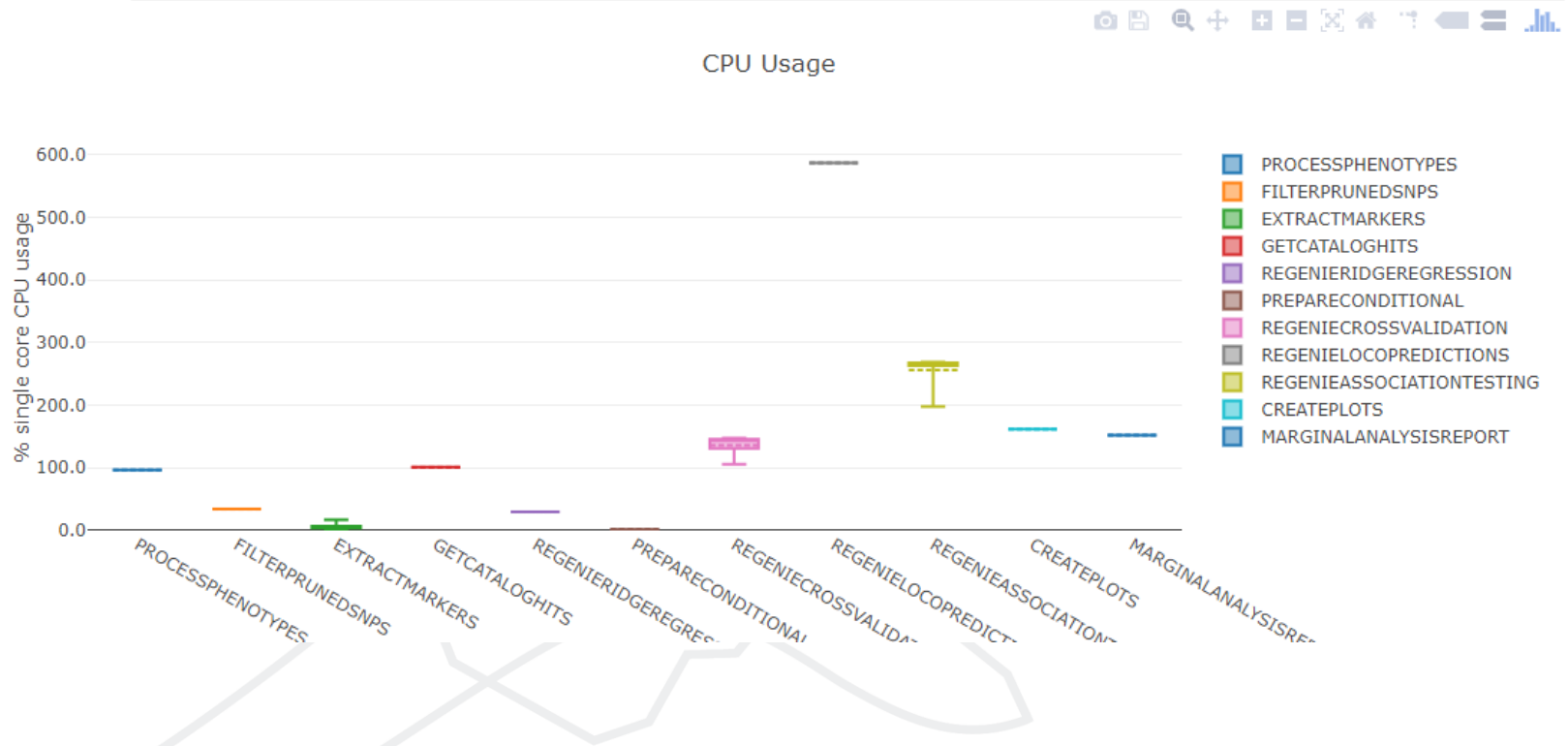
Nextflow Tracing

Resource Usage

These plots give an overview of the distribution of resource usage for each process.

CPU

Raw Usage **% Allocated**



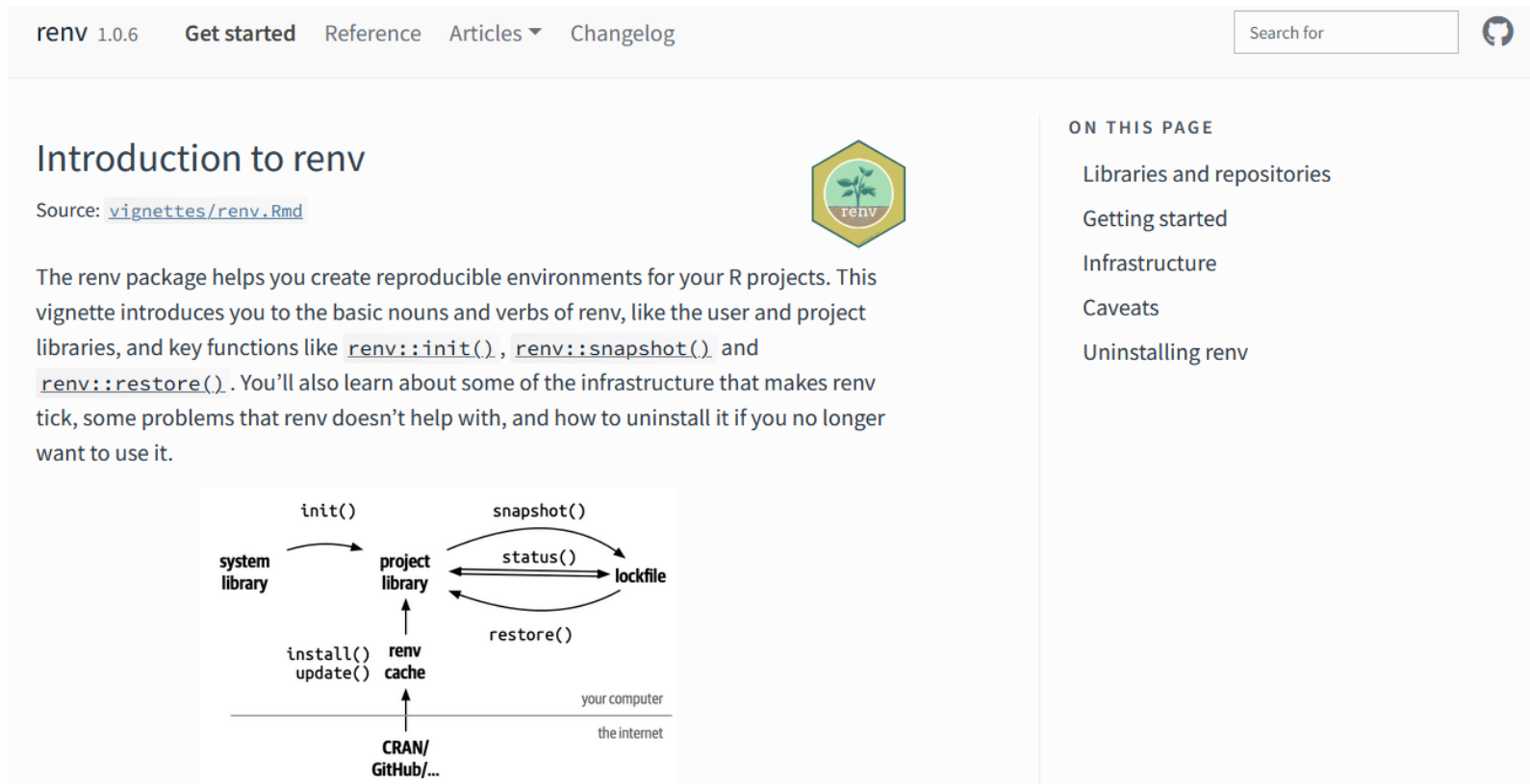
Environment encapsulation

- ▶ **Environment =**
 - ▶ Hardware
 - ▶ Operating system
 - ▶ Software installations

- ▶ Recreates environments:
 - ▶ Containers, virtual machines
 - ▶ Environment managers (e.g., Conda)

Environment encapsulation

renv

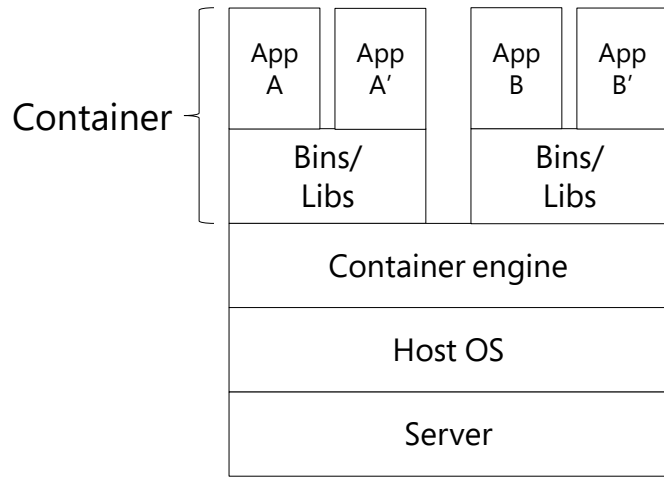
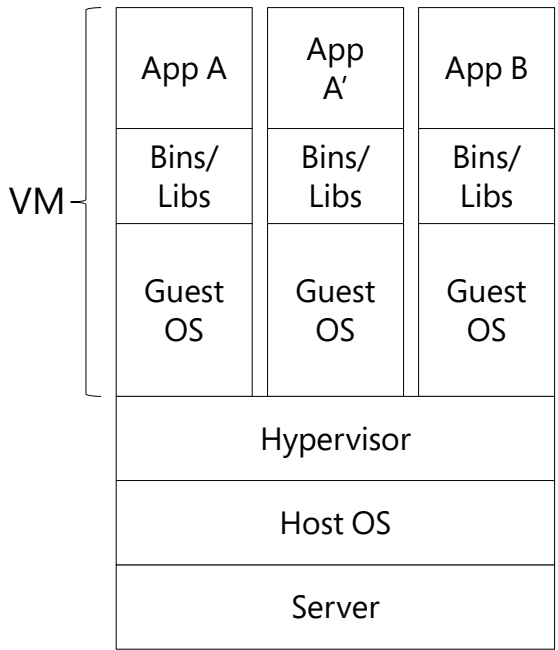


The screenshot shows the top navigation bar of the renv website with links for 'renv 1.0.6', 'Get started', 'Reference', 'Articles', and 'Changelog'. A search bar and a GitHub icon are also present. The main heading is 'Introduction to renv' with a source link to 'vignettes/renv.Rmd' and a hexagonal logo. The introductory text explains that renv helps create reproducible environments for R projects, mentioning functions like `renv::init()`, `renv::snapshot()`, and `renv::restore()`. A sidebar on the right lists 'ON THIS PAGE' with links to 'Libraries and repositories', 'Getting started', 'Infrastructure', 'Caveats', and 'Uninstalling renv'. A diagram at the bottom illustrates the workflow: 'system library' leads to 'project library' via `init()`; 'project library' and 'lockfile' interact via `snapshot()`, `status()`, and `restore()`; 'renv cache' leads to 'project library' via `install()` and `update()`; and 'CRAN/GitHub/...' leads to 'renv cache' via 'the internet'.

Containers vs. virtual machines

- ▶ Virtual machines:
 - ▶ Secure, isolated
 - ▶ Strict hardware limits
- ▶ Containers:
 - ▶ Low overhead
 - ▶ Increased performance
 - ▶ Efficient resource sharing

Containers vs. virtual machines (continued)



Adapted from Trisovic, Ana 2018.

Choice at Cardio-CARE: Apptainer

- ▶ Build and run containers
- ▶ Integrates with Nextflow
- ▶ HPC
- ▶ Secure



Container best practices

- ▶ Use public images
- ▶ Multi-stage builds → reduce image size
- ▶ Non-root user → increase security
- ▶ Fewer dependencies
- ▶ Document versions

Analysis preservation

- ▶ CapriceCockpit
 - ▶ Manages analysis runs
 - ▶ ***Quick demonstration***

Lessons learned

- ▶ WFMS
 - ▶ Connects:
 - ▶ CL tools
 - ▶ Containers/environment managers
 - ▶ HPC/cloud
 - ▶ Platforms
 - ▶ Preserves:
 - ▶ Workflows
 - ▶ Input
 - ▶ Versions
 - ▶ Resource usage

Questions?

 @fethalen

 github.com/fethalen