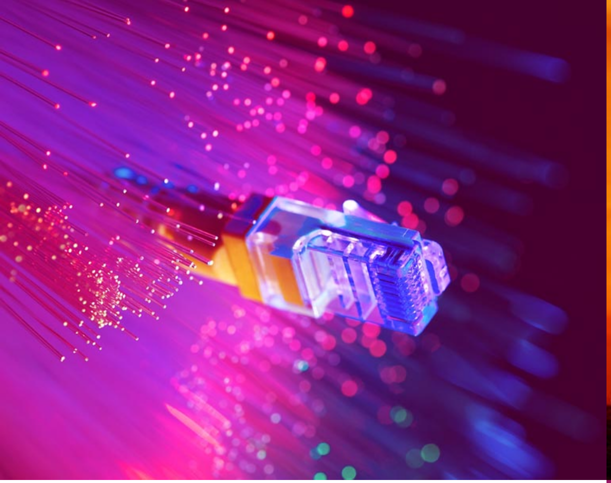


Delivering bioinformatics software as virtual machine image

Workshop on Nordic Big Biomedical Data for Action

Petri Klemelä, CSC – IT Center for Science, Finland

CSC – Finnish research, education, culture and public administration ICT knowledge center

Me

- Petri Klemelä
- CSC – IT Center for Science, Finland
- Software developer
- Developing Chipster software since 2007

Outline

- Challenges of setting up training environments and production servers
- What is a virtual machine image
- Where do they come from?
- Chipster as an example of the bioinformatics software
- How do we produce an use images?
- What have we learnt?

What is the problem in setting up a training environment?

- Typically a lot of software and reference data is needed → installation takes time, you need somebody with admin rights
- Students need to have identical installation → if they come with their own laptops, this is difficult to achieve
- Your course will be repeated in different location → the same installation hassle again!
- NGS analysis tools change rapidly → need to update the tools used in training
- Students need access to an identical environment after the course
- Analysis jobs can require a lot of CPU and memory → laptop might not suffice
- 20-30 people run the analysis job at the same time → need a lot of computing resources temporarily

What is the problem in running an analysis server?

- Typically a lot of software and reference data is needed
- You need many identical installations
- You have to be able to reinstall any server anytime (updates, migrations, hardware issues)
- NGS analysis tools change rapidly
- Analysis jobs can require a lot of CPU and memory → scaling
- You don’t want to wake up to do server maintenance →Fault-tolerance

What is an image and can it help?

- Snapshot of the whole computer (operating system, settings, programs, data)
- Ready-made package of analysis tools and their dependencies, reference data,…
- Provides reproducibility: allows you to create exactly the same environment again
- Runs on your computer or in the cloud (easy to start many)
- Two types of images
  - Virtual machine image (called VM instance when it is running)
  - Container image (called container when it is running)

Images can be made in two different ways

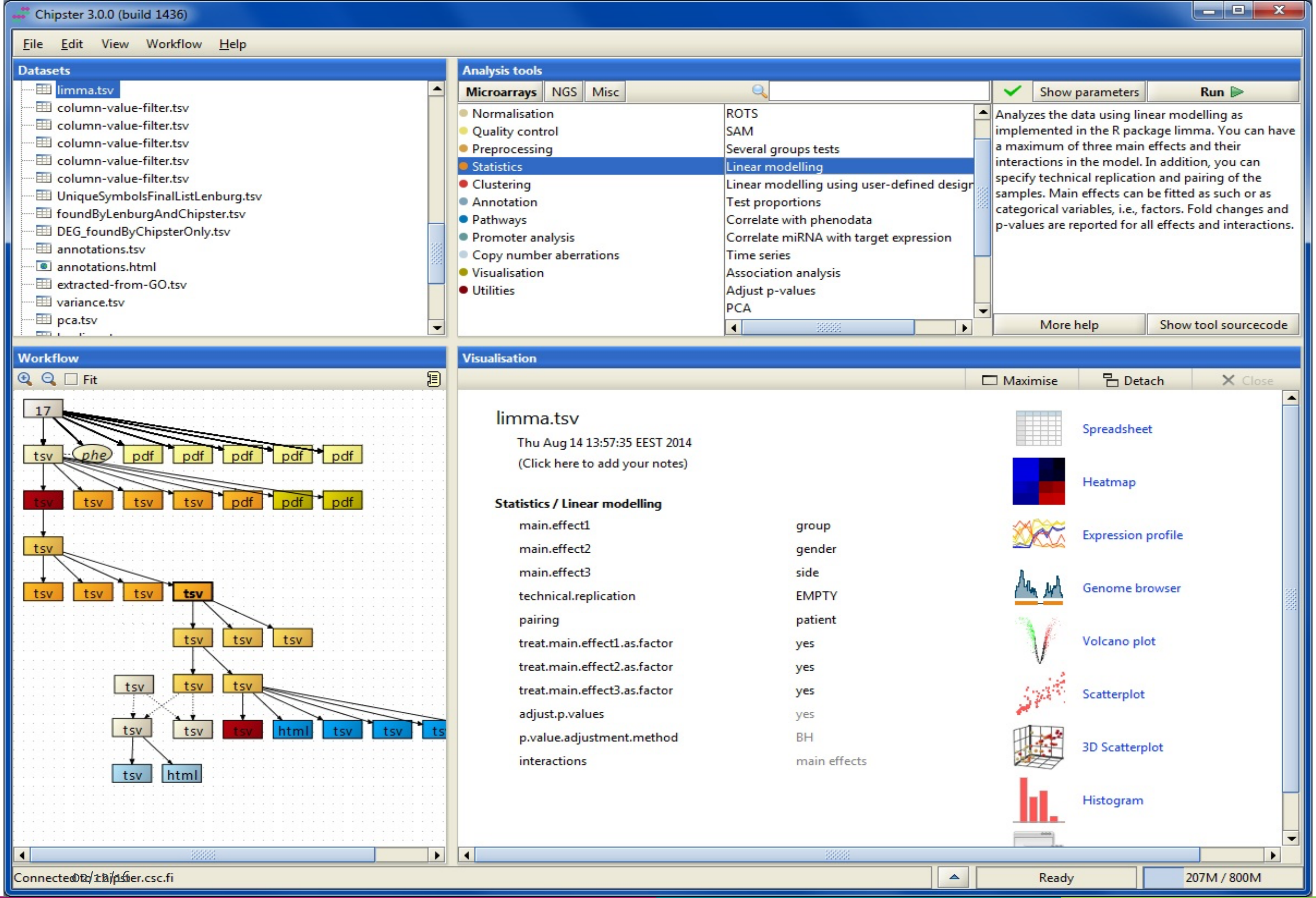
- Build the image manually
  - Take a base image (e.g. Ubuntu), install the analysis tools etc, and take a snapshot
  - Pros: Easy to understand and do
  - Cons: Large image file, hard to version, different VM file format needed for different clouds,…
- Write a recipe for building the image and build it automatically
  - E.g. Ansible file for VM, Dockerfile for Docker containers
  - Pros: Small file, easy to version and update, easy for others to see what exactly goes to your image (admins will love you)
  - Cons: Need more expertise

Virtualized resources offer many options

- Run on laptop, internal cloud or external cloud?
- Virtual machine instance or a container?
- Preconfigured instances or scripts for configuring?
- Separate VM/container for each software, course, user or job?
- Keep pool of instances running or start on demand?
- Reuse instances or start always new?
- How to access?

Example: Chipster in a nutshell

- Free and open source analysis software for high-throughput data
  - Over 350 analysis tools
  - reference data, inc precalculated genome indexes for aligners, etc
- Tools can be used via Chipster GUI or on command line
  - GUI allows users to visualize data interactively and share sessions and workflows
- Lot of training material available
- Easy to scale and tailor
- Available as a ready-to-run virtual machine image
- http://chipster.csc.fi



Datasets:

- limma.tsv
- column-value-filter.tsv
- column-value-filter.tsv
- column-value-filter.tsv
- column-value-filter.tsv
- column-value-filter.tsv
- UniqueSymbolsFinalListLenburg.tsv
- foundByLenburgAndChipster.tsv
- DEG_foundByChipsterOnly.tsv
- annotations.tsv
- annotations.html
- extracted-from-GO.tsv
- variance.tsv
- pca.tsv

Tool categories:

- Normalisation
- Quality control
- Preprocessing
- Statistics
- Clustering
- Annotation
- Pathways
- Promoter analysis
- Copy number aberrations
- Visualisation
- Utilities

Statistics tools:

- ROTS
- SAM
- Several groups tests
- Linear modelling
- Linear modelling using user-defined design
- Test proportions
- Correlate with phenodata
- Correlate miRNA with target expression
- Time series
- Association analysis
- Adjust p-values
- PCA

Analyzes the data using linear modelling as implemented in the R package limma. You can have a maximum of three main effects and their interactions in the model. In addition, you can specify technical replication and pairing of the samples. Main effects can be fitted as such or as categorical variables, i.e., factors. Fold changes and p-values are reported for all effects and interactions.

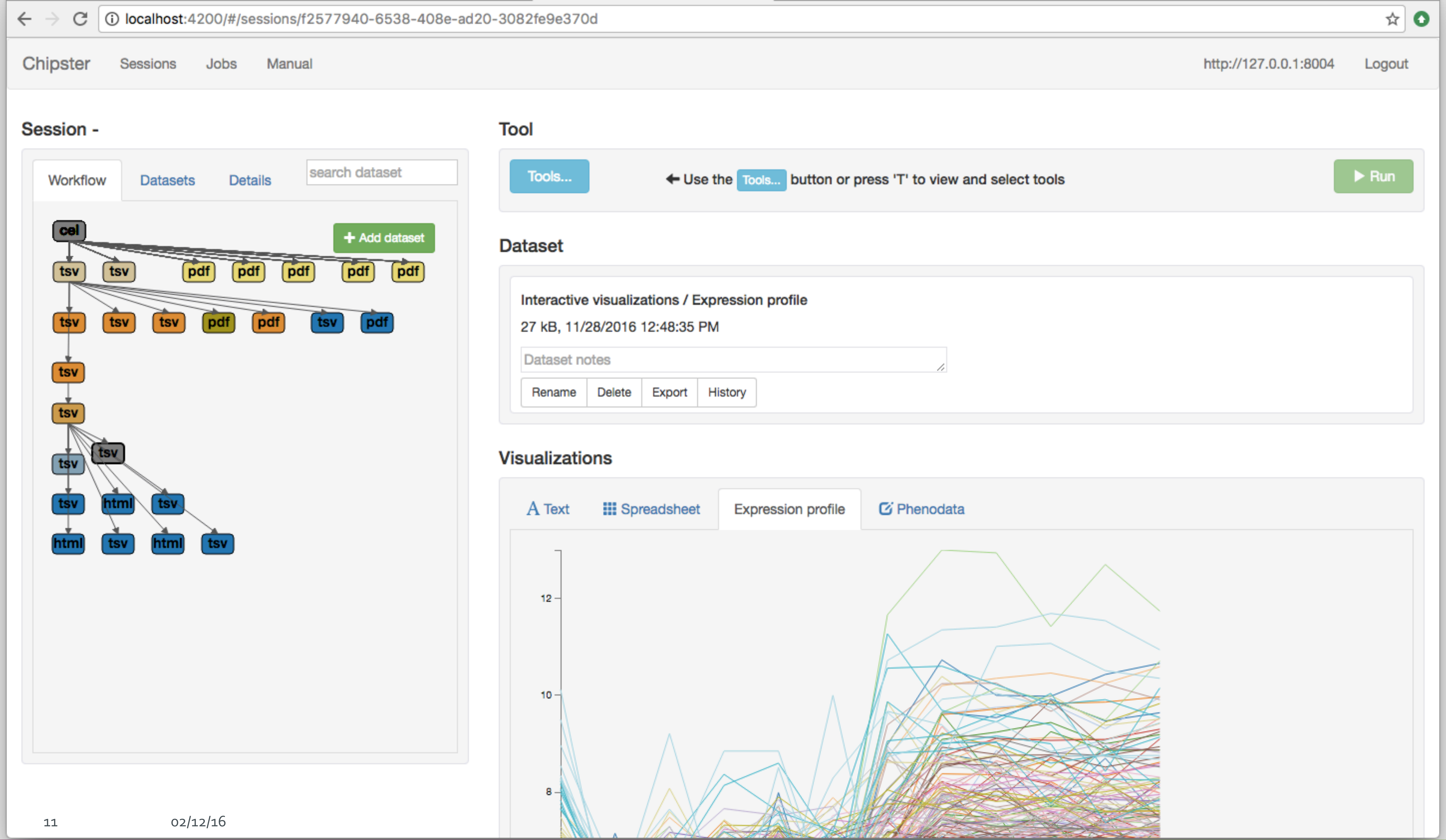
limma.tsv

Thu Aug 14 13:57:35 EEST 2014

(Click here to add your notes)

Statistics / Linear modelling

main.effect1	group
main.effect2	gender
main.effect3	side
technical.replication	EMPTY
pairing	patient
treat.main.effect1.as.factor	yes
treat.main.effect2.as.factor	yes
treat.main.effect3.as.factor	yes
adjust.p.values	yes
p.value.adjustment.method	BH
interactions	main effects



← Use the Tools... button or press 'T' to view and select tools

Interactive visualizations / Expression profile

27 kB, 11/28/2016 12:48:35 PM

The problem

- Free and open source analysis software for high-throughput data
  - Over 350 analysis tools
  - reference data, inc precalculated genome indexes for aligners, etc

Binaries

- 91G R-3.2.3
- 1.9G VirusDetect-1.62
- 155M R-3.3.2
- …

Reference data

- 41G genomes/fasta
- 7.1G genomes/gtf
- …

Indexes

- 48G genomes/indexes/bwa
- 41G genomes/indexes/bowtie2
- 28G genomes/indexes/bowtie
- 12G genomes/indexes/tophat2

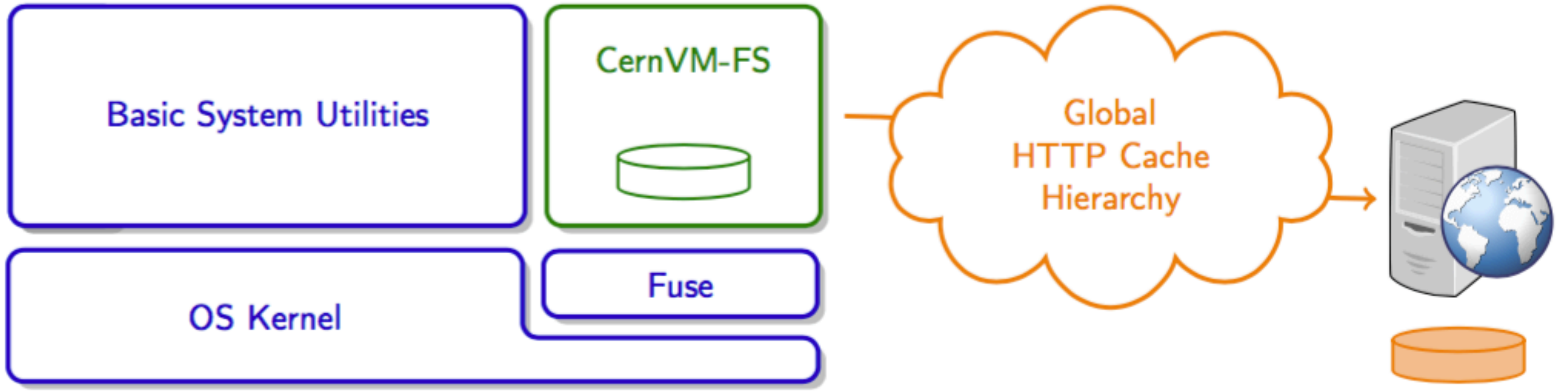
Solution 1: Package everything as a virtual machine image

- Start the previous version
- Install new software
- Take a snapshot
- Build image files
- Copy it to a web server
- Launch a virtual machine
- Wait for hypervisor to load the image

- Every step will take several a hours (e.g. 4) resulting very slow troubleshooting

Solution 2: CernVM File System

- Transfers data on demand and caches it

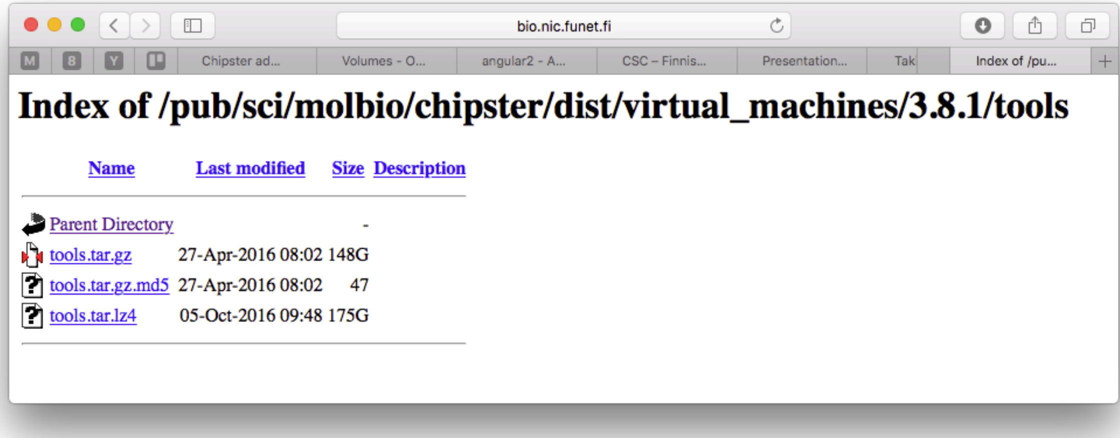


https://cernvm.cern.ch/sites/cernvm.web.cern.ch/files/hep-cloud.pdf

- Data import takes time
- Additional servers to maintain
- Instructions for setting up your own server in EGI Federated Cloud, but usually a version or two behind

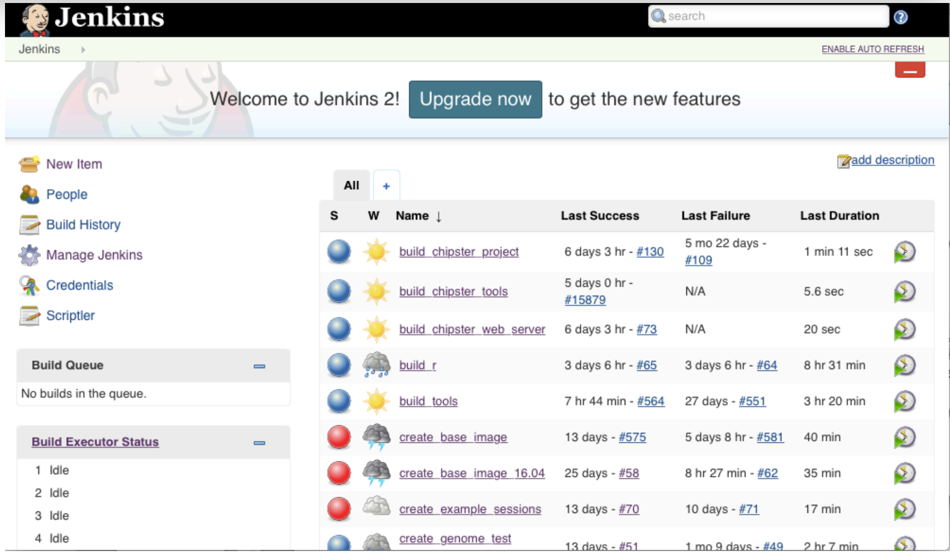
Solution 3: Compromise

- Distribute the difficult parts (i.e. the bootable operating system image) in a small VM image
- Download the tools package later
  - 200GB .tar.gz package on a web server
  - Still difficult to download
  - Optimization is possible with gnu-parallel, lftp, lz4 etc.
- Share locally with a NFS server
  - Can be tested and deployed immediately
- Continuous integration server Jenkins and lot of custom Ansible scripts to do all this



Index of /pub/sci/molbio/chipster/dist/virtual_machines/3.8.1/tools

Name	Last modified	Size	Description
Parent Directory		-	
tools.tar.gz	27-Apr-2016 08:02	148G	
tools.tar.gz.md5	27-Apr-2016 08:02	47	
tools.tar.lz4	05-Oct-2016 09:48	175G	



Welcome to Jenkins 2! Upgrade now to get the new features

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle
3 Idle
4 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	build_chipster_project	6 days 3 hr - #130	5 mo 22 days - #109	1 min 11 sec
●	☀	build_chipster_tools	5 days 0 hr - #15879	N/A	5.6 sec
●	☀	build_chipster_web_server	6 days 3 hr - #73	N/A	20 sec
●	☂	build_r	3 days 6 hr - #65	3 days 6 hr - #64	8 hr 31 min
●	☀	build_tools	7 hr 44 min - #564	27 days - #551	3 hr 20 min
🔴	⛈	create_base_image	13 days - #575	5 days 8 hr - #581	40 min
🔴	⛈	create_base_image_16.04	25 days - #58	8 hr 27 min - #62	35 min
🔴	☁	create_example_sessions	13 days - #70	10 days - #71	17 min
●	☁	create_genome_test	13 days - #51	1 mo 9 days - #49	2 hr 7 min

Next steps: Choose the options that improve turnaround time

- Allow any part to be build and tested independently
  - Now it takes 3 hours to install the tool binaries, another 3 hours to build the indexes etc.
- Better tools for bioinformaticians for tool script development and building reference genomes and indexes
- Production services that can be updated without a service break

Summary

- Whether you are running a course or a server, you have to rebuild the environment every now and then
- Especially when you are developing it
- Configuration of the environment is as difficult and important as the software development in general (but the tools are inferior)
  - Best practices still pay off (code readability, code reuse etc.)
- Virtual machine image is a nice package for a software, but you must be able to update any part of it easily
- Make your configuration scripts public
  - May improve code quality
  - Build trust
  - Easier to apply for something that you couldn’t even imagine

CSC – IT Center for Science Ltd

+3589 457 2821 (service requests)
+3589 457 2001 (call center/ contacts)

servicedesk@csc.fi

www.csc.fi

https://www.facebook.com/CSCfi

https://twitter.com/CSCfi

https://www.youtube.com/c/CSCfi

https://www.linkedin.com/company/csc---it-center-for-science

Kuvat CSC:n arkisto ja Thinkstock