

# Graphical interfaces in MetaCentrum

Aleš Křenek

e-Infra CZ conference, Apr 30, 2024

# Motivation

Tony Mitchell's view



# Motivation

Reality for many of us

```
bilouisII:~ liocha$ ssh -K zuphux
ljocha -- ssh -K zuphux -- 80x24
echo -n "$f" >>$out-time.tsv
for c in ${nc[@]}; do
  if [ \(${c} -lt $((f / 4)) -a $c -lt $((maxcores / 2)) \) -o $c -gt $maxcores -o $c -gt $f ]; then
    echo -n '          nan' >>$out-ret.tsv
    echo -n '          nan' >>$out-mem.tsv
    echo -n '          nan' >>$out-time.tsv
    continue
  fi
  echo ${dirname $0}/memory_single_run.sh -c $c -o $out-${f}_${c}.out
  t -e $out-${f}_${c}.err ${files[@]:0:$f}
  res=(-1 -2 -3)
  res=${$(dirname $0)/memory_single_run.sh -c $c -o $out-${f}_${c}.out -e $out-${f}_${c}.err ${files[@]:0:$f)}
  echo -n "    ${res[0]}" >>$out-ret.tsv
  echo -n "    ${res[1]}" >>$out-mem.tsv
  echo -n "    ${res[2]}" >>$out-time.tsv
done
echo >>$out-ret.tsv
echo >>$out-mem.tsv
echo >>$out-time.tsv
done
[ljocha@zuphux apLCMS-measure-memory]$
this frontend runs OS Centos (unl
if you want to use the Debian env
run_in_os debian11 /bin/bash -l
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
-bash: warning: setlocale: LC_CTY
Tip of the day: Need to work with
can be vastly faster than using
[ljocha@zuphux ~]$ [qsub -q gpu -l walltime=2:0:0 -l select=1:ncpus=12:mem=32gb:scratch_local=100gb:ngp
us=1 -- $PWD/predict-pbs.sh nist-train-small_01
balkis.ics.muni.cz
name: zuphux.cerit-sc.cz
em: CentOS 7.9.2009 on x86_64
ljocha -- ssh -K zuphux -- 80x24
ull-50_12.err full-60_56.out full-80_40.err
ull-50_12.out full-6_1.err full-80_40.out
ull-50_16.err full-6_1.out full-80_48.err
ull-50_16.out full-6_2.err full-80_48.out
ull-50_20.err full-6_2.out full-80_56.err
ull-50_20.out full-6_4.err full-80_56.out
ull-50_24.err full-6_4.out full-80_64.err
ull-50_24.out full-70_20.err full-80_64.out
ull-50_32.err full-70_20.out full-80_80.err
ull-50_32.out full-70_24.err full-80_80.out
ull-50_40.err full-70_24.out full-8_2.err
ull-50_40.out full-70_32.err full-8_2.out
ull-50_48.err full-70_32.out full-8_4.err
ull-50_48.out full-70_40.err full-8_4.out
ull-60_16.err full-70_40.out full-8_8.err
ull-60_16.out full-70_48.err full-8_8.out
ull-60_20.err full-70_48.out full-mem.tsv
ull-60_20.out full-70_56.err full-ret.tsv
ull-60_24.err full-70_56.out full-time.tsv
ull-60_24.out full-70_64.err full-70_64.out
ull-60_32.err full-70_64.out full-80_20.err
ull-60_32.out full-80_20.out full-80_20.out
ull-4_1.err full-4_1.out full-60_40.err
ull-4_1.out full-60_40.out full-80_20.out
```

# Motivation

## Beyond qsub

- Batch jobs still achieve the best resource utilization
  - and the resources are expensive
- The user wants better experience nowadays
- **Graphical interfaces** are a trade-off
- Become more and more FAIR

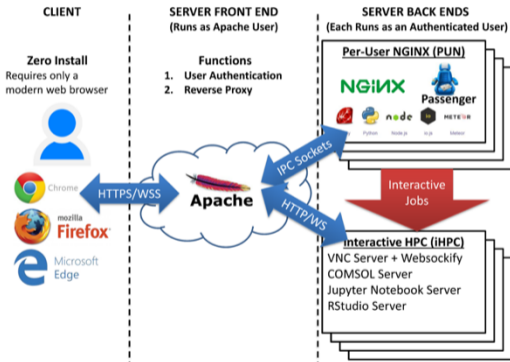
# Open OnDemand

## User experience

- I need 64 cores, 1 TB RAM, 4 hi-end GPU for my Matlab calculation
- I can't spend so much money to purchase this computer
- Well, I need it 2 hours per month either
- Go to `https://ondemand.metacentrum.cz/`
  - say what you need for how long

# Open OnDemand

## Architecture



# Open OnDemand

## 3D accelerated remote desktop

- VirtualGL, <https://virtualgl.org>
  - intercept OpenGL calls, render on headless GPU, copy image
- TurboVNC, <https://turbovnc.org>
  - VNC server optimized for transmitting rendered 3D images
- noVNC, <https://novnc.com>
  - zero install VNC client (HTML5)

# Open OnDemand

## Optimized scheduling

- dedicated PBS queue
- handled by dedicated scheduler – fast turnaround
- dedicated compute nodes
- “run-or-move” strategy
  - in average, 50 % jobs are run immediately ( $< 1$  minute)
  - otherwise moved to “normal” queue



# Open OnDemand

## Available applications

- Ansys/Fluent, Ansys/Workbench
- CLCgenomcsWB
- VMD
- Matlab, RStudio, Jupyter notebook
- generic desktop (“module add anything”)
- plain shell

# Rancher

- <https://rancher.cloud.e-infra.cz>
- Similar purpose, different technology (Kubernetes)
- Far more flexible – less intuitive sometimes
- Slightly different set of prepared applications

# Galaxy

## Overview

- <https://usegalaxy.org>, <https://usegalaxy.eu>, <https://usegalaxy.cz>
- “Galaxy is a free, open-source system for analyzing data, authoring workflows, training and education, publishing tools, managing infrastructure, and more.”  
(<https://galaxyproject.org>)

recetox-aplcms - generate feature table generate feature table from noise-removed HRMS profile data (Galaxy Version 0.12.0+galaxy2)

**Input profile data**

73: RECETOX apLCMS Hybrid updated\_known\_table on data 25, data 2...

Mass spectrometry profile data.

**Bandwidth factor**

0.5

Parameter used to scale down the overall range of retention times (the bandwidth) assumed in the kernel smoother used for peak identification. The value is between zero and one. The minimal and maximal bandwidth can be limited by explicit values.

**Minimal bandwidth [unit corresponds to the retention time]**

The lower limit on the resulting bandwidth. If not given, it is estimated based on the overall range of retention times in the profile.

**Maximal bandwidth [unit corresponds to the retention time]**

The upper limit on the resulting bandwidth. If not given, it is estimated based on the overall range of retention times in the profile

**BIC factor**

2.0

A factor influencing Bayesian information criterion (BIC) in estimation of RT peak shape. If the value is larger than 1, models with more peaks are penalized more.

Advanced

Execute

**Galaxy UMSA**
Using 39.5 GB

Workflow
Visualize
Shared Data
Admin
Help
User

**Tools**

Upload Data

Collection Operations

Interactive Tools

Visualization

GENERAL TEXT TOOLS

Text Manipulation

Filter and Sort

Join, Subtract and Group

UMSA

RECETOX

CAMERA

ChemicalToolBox

ISA tools

XCMS

SECIMTools

metaMS

W4M

MetFrag

SIRIUS

### Dataset Information

Number	165
Name	matchms output formatter (get-thresholded-data) on data 164 and data 163
Created	Friday Jul 15th 9:10:44 2022 UTC
Filesize	1.2 KB
Dbkey	?
Format	tsv
File contents	contents
History Content API ID	c748b79d4648e9a4 (101658)
History API ID	2d75b8c756d9b206 (9899)
UUID	ad4f50b5-a8d8-4e2e-9bc0-3c3e4a1bc4cc
Full Path	/mnt/volume/shared/ces- nya/hfs4/home/umsa/cache/object_store_cache/067/dataset_67867.dat

### Tool Parameters

Input Parameter	Value
Scores Table	163 CosineHungarian scores of data 24 and data 162
Matches Table	164 CosineHungarian matches of data 24 and data 162
Formatting method	get-thresholded-data
Scores Threshold	0.6
Matches Threshold	3

### Job Outputs

Tool Outputs	Dataset
matchms output formatter (get-thresholded-data) on	165 matchms output formatter (get-thresholded-data) on data 16...

### Job Information

Galaxy Tool ID: [toolshed.g2.bx.psu.edu/repos/recetox/matchms\\_formatter/matchms\\_formatter/0.1.3](https://toolshed.g2.bx.psu.edu/repos/recetox/matchms_formatter/matchms_formatter/0.1.3)

**History**

Missing peaks #5

15 shown, 44 deleted, 131 hidden

9.84 GB

**History**

Missing peaks #5

15 shown, 44 deleted, 131 hidden

9.84 GB

165: matchms output f  
ormatter (get-threshol  
ded-data) on data 164 and data 16  
3

164: CosineHungarian  
matches of data 24 and  
data 162

163: CosineHungarian  
scores of data 24 and d  
ata 162

162: RI using kovats of  
Mass spectra from RA  
MClustR on data 159

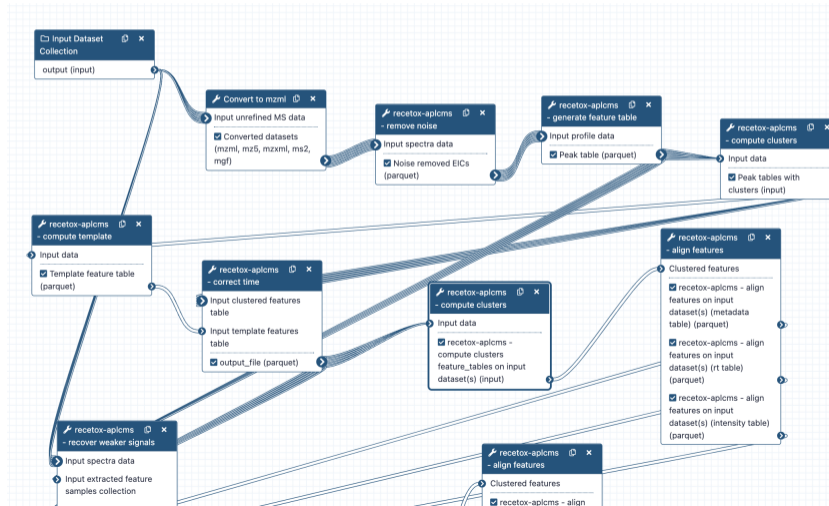
161: Mass spectra from  
RAMClustR on data 159

160: Spec Abundance o  
f data 159

159: Feature-by-sampl  
e RECETOX apLCMS Hy  
brid recovered\_feature\_sample\_ta  
ble on data 25\_data 21\_and others t

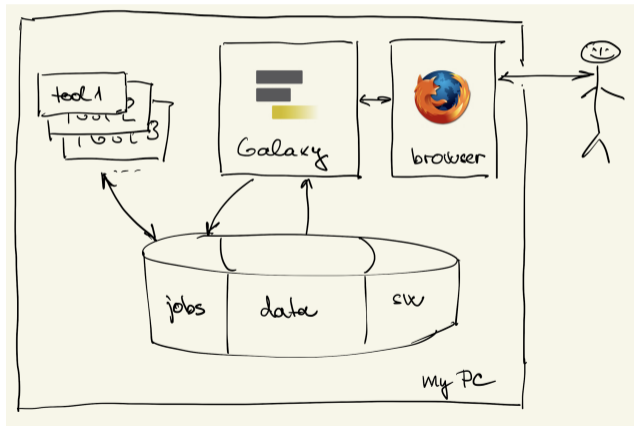
# Galaxy

## Workflows



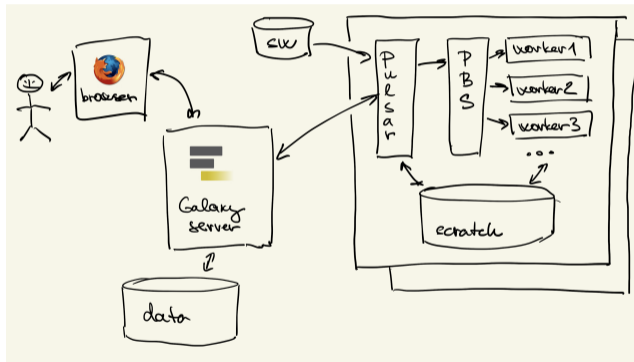
# Galaxy

Install your own



# Galaxy

## Pulsar network





# Galaxy

[usegalaxy.cz](http://usegalaxy.cz)

- Pulsar architecture, submit jobs to Metacentrum PBS
- Mirrored set of tools from [usegalaxy.eu](http://usegalaxy.eu)
- More resources for e-Infra CZ users, including GPU

## Successful example: Metadynminer (Jan Beránek, UCT)

python\_metadynminer.ipynb

### 5. Identify the local minima

You have two options: If you set the keyword `precise=True`, the local minima will be identified by an algorithm which:

1. finds all local minima, even very shallow and probably unimportant minima,
2. each point on the FES will be assigned to the minimum the system would most likely go to, if it only follows the gradient of free energy, and
3. free energy value of minima will be calculated from each point on FES assigned to the respective minima. This results in more precise free energy values, as it accounts for the width of the minimum as well. For this calculation the unit of free energy (`energy_unit="kJ/mol"` or `energy_unit="kcal/mol"`) and the thermodynamical temperature (`temp`) of the simulation must be supplied. This algorithm doesn't use the `nbins` keyword.

Example:

```
minisa = mn.Minima(fes, precise=True, temp=300.0, energy_unit="kJ/mol")
```

If you set `precise = False`, the method will use the original algorithm from the metadynminer package for R. In this algorithm the FES is first divided to number of bins (can be set with option `nbins`, default is 8) and the absolute minima is found for each bin. Then the algorithm checks if this point is really a local minimum by comparing to the surrounding points of FES. This algorithm only accounts for the depth of each minima, which is less precise, but usually sufficient.

In some cases, this algorithm is the preferred one, because on some free energy landscapes the total number of local free energy minima can reach tens of thousands, which makes the calculation using precise algorithm slow and impractical.

Example:

```
minisa = mn.Minima(fes, precise=False, nbins=8)
```

```
[9]: minisa = mn.Minima(fes, precise=True, temp=300.0, energy_unit="kJ/mol")
Calculating gradients for FES with 6536 bins...
Searching for the nearest local minima...
Done.

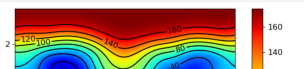
Print the list of the local minima:
```

```
[10]: minisa.e minima
```

	Minimum	free energy	CV1bin	CV2bin	CV1 - torsion_SS	CV2 - torsion_cycle
0	A	0.0	193.0	161.0	1.0953400194010667	0.8517301829032395
1	B	6.963876206871731	61.0	177.0	-1.6444274046134073	1.1935812021613703

You can visualise the FES with the local minima as letters. All keywords are analogous as for the `fes.plot()` function. The color of the letters changes automatically to ensure their good visibility, but if you want to override this behaviour, provide a matplotlib color like this: `color="black"`.

```
[11]: # plot the free energy surface with minima
minisa.plot(contours_spacing=20)
```



# Jupyter Notebooks

## One size doesn't fit all

1. `https://ondemand.metacentrum.cz`
  - standard PBS job
  - singularity container (predefined or custom)
  - almost full Metacentrum environment
2. `https://hub.cloud.e-infra.cz`
  - in Kubernetes, with all its flexibility and complexity
  - easy to make custom clones (Metadynminer)
3. `https://notebooks.egi.eu`
  - more standardized for international collaborations
  - support for community installations
  - integration with OneData, B2DROP
  - EGI Check-in authentication

## Summary

- Still lagging 20 years behind the scientists in movies
- Metacentrum offers something more than qsub:
  - generic graphical environments: OnDemand and Rancher
  - Galaxy – de facto standard in large international community
  - Jupyter notebooks – mix text and code, record experiments
- Access to the whole infrastructure
- Better user experience, support for open science