Establishing a Galaxy Server and building a (local) Galaxy User Community from scratch.

A field report from the UFZ Leipzig

Matthias Bernt

UFZ Leipzig

19.11.2018

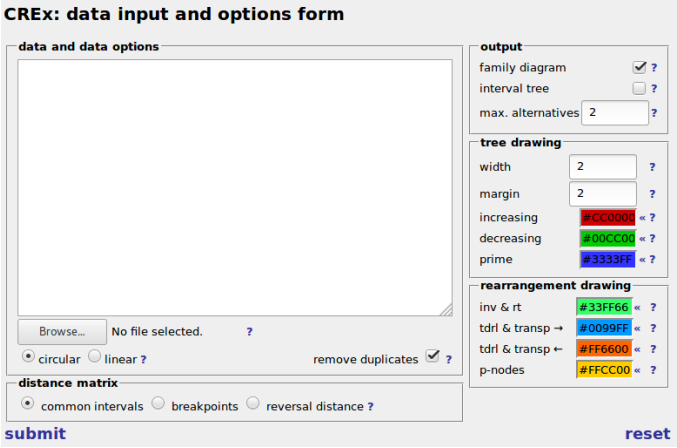
My pre Galaxy days ($\approx$ 2 years ago)

PostDoc @ University of Leipzig:

- Algorithmic Bioinformatics
  - Genome rearrangements (CREx)
  - Phylogeny
- Mitochondrial genetics
  - Annotation (MITOS)
  - Comprehensive comparative analyses
- Parallel Computing, Cellular Automata

My pre Galaxy days (≈ 2 years ago)

Some experience: providing software via (ugly) web interfaces



- MITOS:
  - Self written cluster connection
  - 100.723 jobs ≡ 1.276 MB → 900 citations

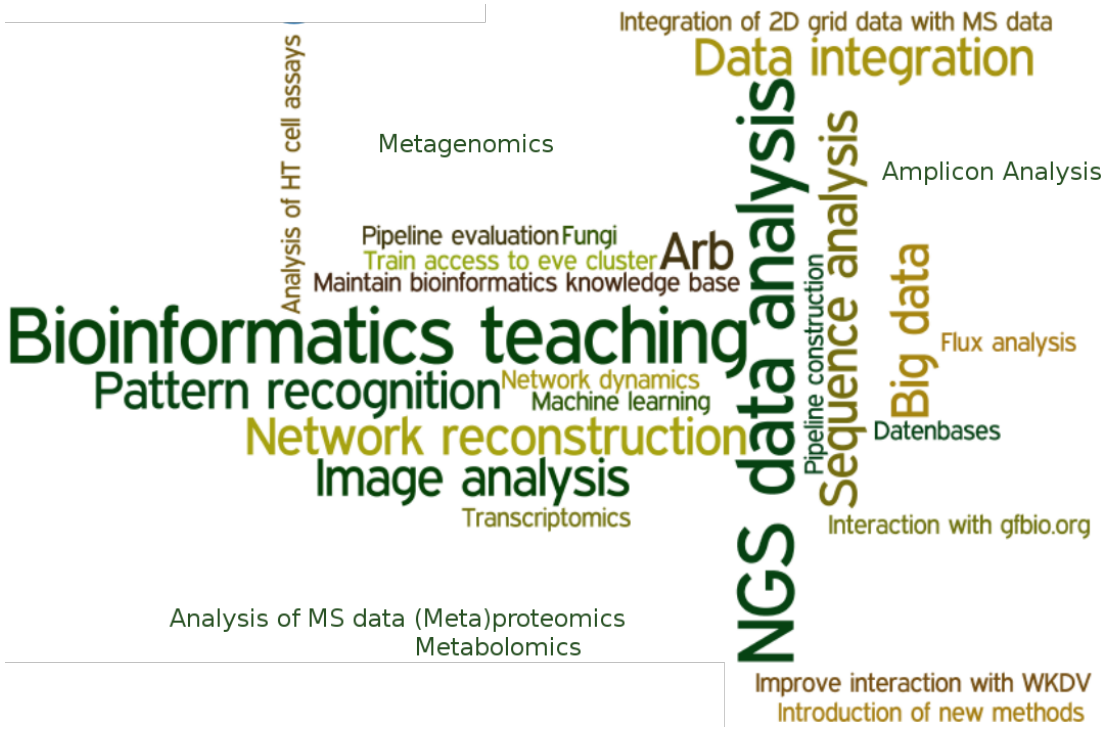
Only little experience with the analysis of NGS / MS data

Bioinformatics Service at the UFZ Leipzig

UFZ:

- 1.093 employees (292 PhDs, 382 visiting scientists)
  - Very diverse research fields and methods
  - Mainly wet lab

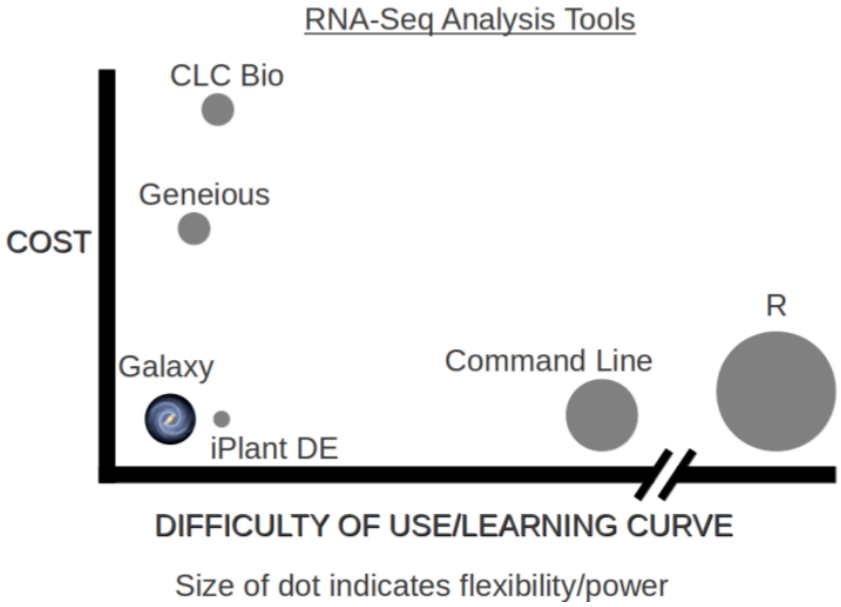
Bioinformatics at the UFZ:



Bioinformatics Service at the UFZ Leipzig

Tasks:

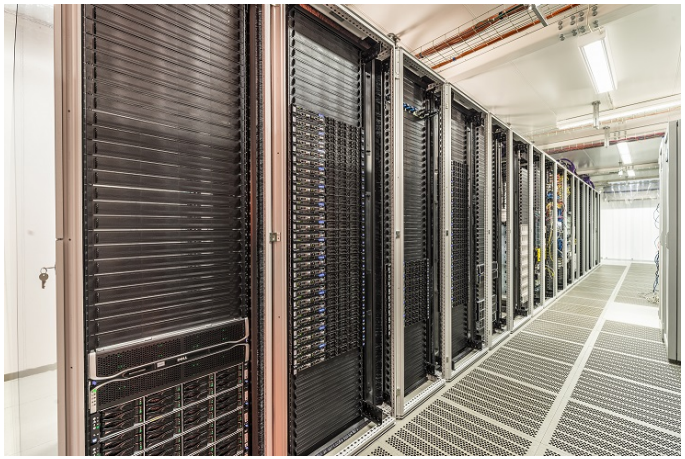
- Development of Pipelines and Workflows
  - Focus: Solve problems occurring for many
  - **Establish a Galaxy** Server at the UFZ
- Teaching
- Improve cooperation: user groups
- Demand analysis



HPC @ UFZ

Integrate Galaxy into:

- EVE (shared between UFZ and iDiv)
  - 99 compute nodes (2,532 cores, 21,496 GB)
  - Plain CentOS 6
  - UNIVA Grid engine (need to specify runtime and memory)
- LDAP, long term archiving system



Obstacles:

- No super user access → weekly meetings with admins
- No connection from the outside

Running Galaxy Step 1: local

- Install in a local VirtualBox with plain CentOS6
- Setup via Ansible
- Successful integration in LDAP

Helpful:

- Public slides from Galaxy admin workshops
- Galaxy wiki
- galaxy-dev mailing list

Problems:

- The Galaxy documentation was moved during this time

Running Galaxy Step 2: EVE

- Moving to the EVE HPC cluster
  - Handed the Ansible scripts to the admins
  - Installed a PostgresDB in a VirtualBox
  - Job submission via Galaxy's drmaa runner
    - as Galaxy system user
    - run everything as jobs (uploads, . . .)
- Setup of two Galaxy servers with identical setup
  - production (latest release)
  - testing (dev)

So far everything went really well :)

Idea: setup of `job_conf`:

- Escalation strategy to minimize administration efforts
- Destinations with combinations of
  - (10min, 1day, 1week) $\times$ (6G, 18G)
  - same for parallel jobs

Running Galaxy Step 2: Problems

Problems:

1. Admins required that jobs run as real user
   - Galaxy used email for submission, but EVE needs user name
   - $\Rightarrow$ #4096

Running Galaxy Step 2: Problems

Problems:

1. Admins required that jobs run as real user
   - Galaxy used email for submission, but EVE needs user name
   - ⇒ #4096
2. drmaa runner uses:
   `drmaa_session.job_status(external_job_id)`
   - only distinguishes DONE / FAILED (deleted?, killed because violation of run time / memory?)
   - idea: use `drmaa_session.wait(external_job_id)`
   - started to work on #7004 (formerly #4275, #4857)

Running Galaxy Step 2: More Problems

3. `job_status` and `wait` only work in the same drmaa session
   - in the real user setting jobs are started by an external script

Running Galaxy Step 2: More Problems

3. `job_status` and `wait` only work in the same drmaa session
   - in the real user setting jobs are started by an external script

Solution for 2+3: #7004 (inspired by the slurm runner):

- `job_status` + `qstat` to distinguish:
  - queued/running from finished/failed
- `wait` + `qacct` to get detailed infos on finished jobs
- Now running on our cluster for 1 1/2 years and only 1 last known bug
- Pretty sure that it should work on other drmaa systems (SGE, torque, . . .)

Running Galaxy Step 2: More Problems

3. `job_status` and `wait` only work in the same drmaa session
   - in the real user setting jobs are started by an external script

Solution for 2+3: #7004 (inspired by the slurm runner):

- `job_status` + `qstat` to distinguish:
  - queued/running from finished/failed
- `wait` + `qacct` to get detailed infos on finished jobs
- Now running on our cluster for 1 1/2 years and only 1 last known bug
- Pretty sure that it should work on other drmaa systems (SGE, torque, . . .)

4. OOM detection
   - not many tools implement oom checks
   - Galaxy does not check OOM if exit code is present (#6338, now John Chilton's #6685)

Running Galaxy Step 3: Lessons learned

- Messing with the Galaxy sources is difficult
  - in particular for newbies
  - only little source code documentation
  - but extremely helpful community
- Its very time consuming because lots of restarts are necessary

Running Galaxy Step 2: More Problems

JAVA

- `Could not reserve enough space for ...KB object heap`
- `Could not allocate metaspace: ... bytes`
- `There is insufficient memory for the Java Runtime Environment to continue.`
- ...

"Solution":

- specialized JAVA destinations setting -Xmx to a fraction of the available memory
- but now all tools that use java need to be specified manually in `job_conf`

Running Galaxy Step 3: Get users

After $\approx$ 3 month there was a running Galaxy, but only a few people at the UFZ knew it.

- Newsletter
- Announcements

Little success

- At GCC2017 I learned about the GTN

1. Let's try training
2. Push key projects with motivated users

Running Galaxy Step 3: Get users by training

Strategy:

- Choose the most 6 most fancy tutorials from GTN
- Start a poll and schedule sufficient courses
- Two rounds (full day Galaxy 101 + Specialization)
  - Transcriptomics
  - Amplicon Data Analysis
  - Genome assembly

Tips:

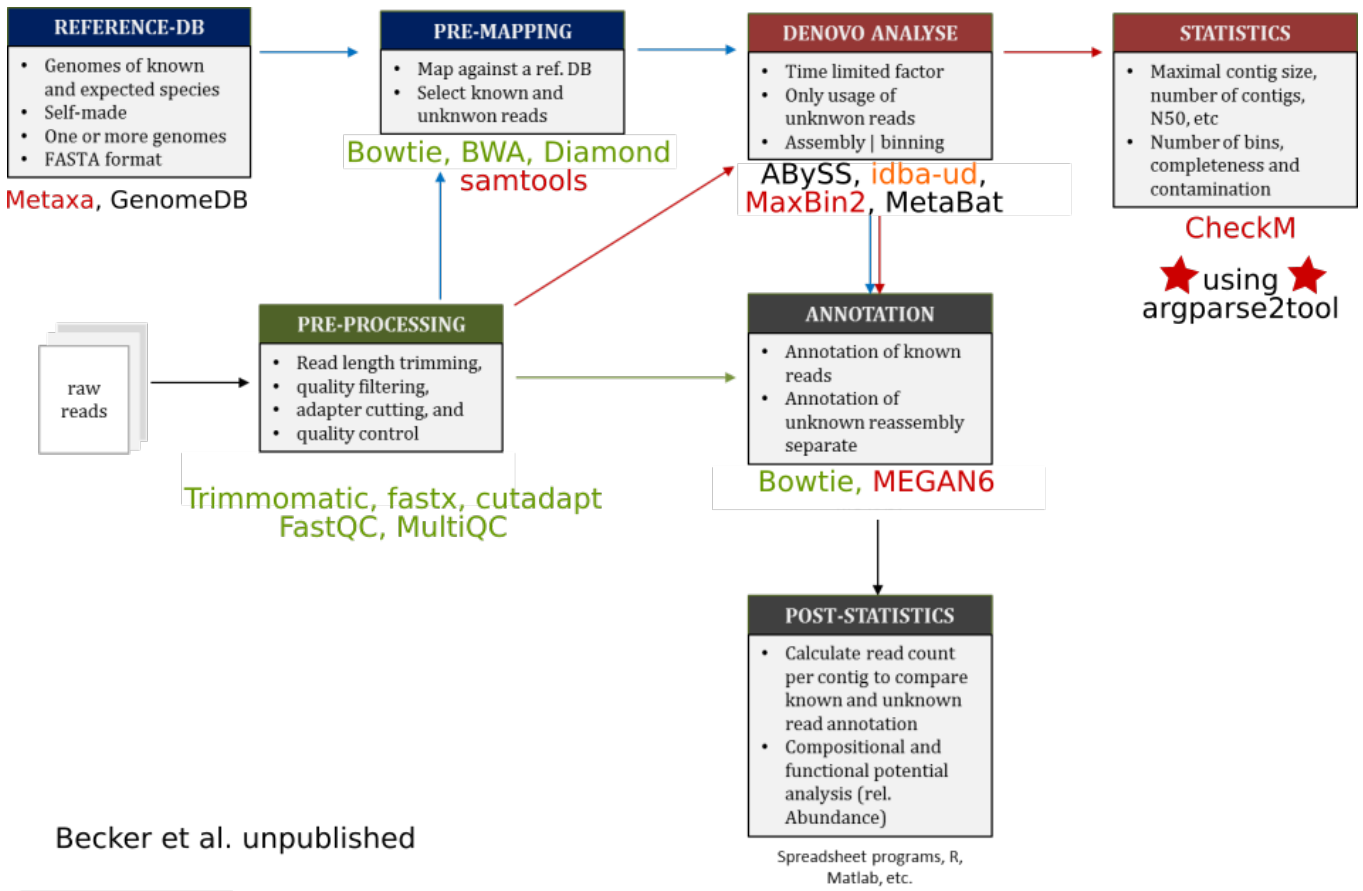
- Do a test course (e.g. Galaxy 101) with close colleagues
  - check if your Galaxy scales to a dozen simultaneous users
- Check the course early enough on your own :)
  - Some tools might not work / recent versions differ
  - Now much better due to automatic testing of the tutorials

Running Galaxy Step 3: Get users by key projects

- Workflow4Metabolomics
  - actually just worked
  - but colleague left before establishing it entirely
  - $\Rightarrow$ Training
- Proteomics Workflows (MetaProSIP) from Knime
  - OpenMS bugfixes
- (Meta)proteomics + transcriptomics workflows
  - SearchGUI/Peptideshaker bugfixes
- RADSeq pipelines
  - Stacks upgrade

Running Galaxy Step 3: Get users by key projects

- Metagenomics Workflow



Becker et al. unpublished

Running Galaxy Step 3: Local community

User group meetings

- 1-2 talks
- exchange of information and ideas
- So far only happened once (amplicon analysis)

Communication platform

- For target oriented communication (eg. announcements on Galaxy)
- Currently slack

Running Galaxy Step 3: Local community

Success?

- Number of (registered) Galaxy users approaching 100.
- It's so hard to break habits. Some still prefer:
  - to move data in Excel
  - to use expensive vendor black boxes
- We have quite good R courses. R users come when they need HPC.

More plans

- Get the new PhD students
  - Galaxy 101 for everyone :)
- Would like to share our Galaxy with iDiv
  - joint administration

Running Galaxy Step 4: Give back to the community

- Tool building training
  - Bioinf Leipzig retreat (w Joerg Fallmann and Stephanie Kehr)
- CGGUG: Central German Galaxy User Group (w Steffen Neumann)
  - So far 1 meeting in Halle to find out the training needs of admins/users
  - Tool building workshop in planning

Thank you for your attention