WORKING WITH DATA

Karin Lagesen

karin.lagesen@vetinst.no

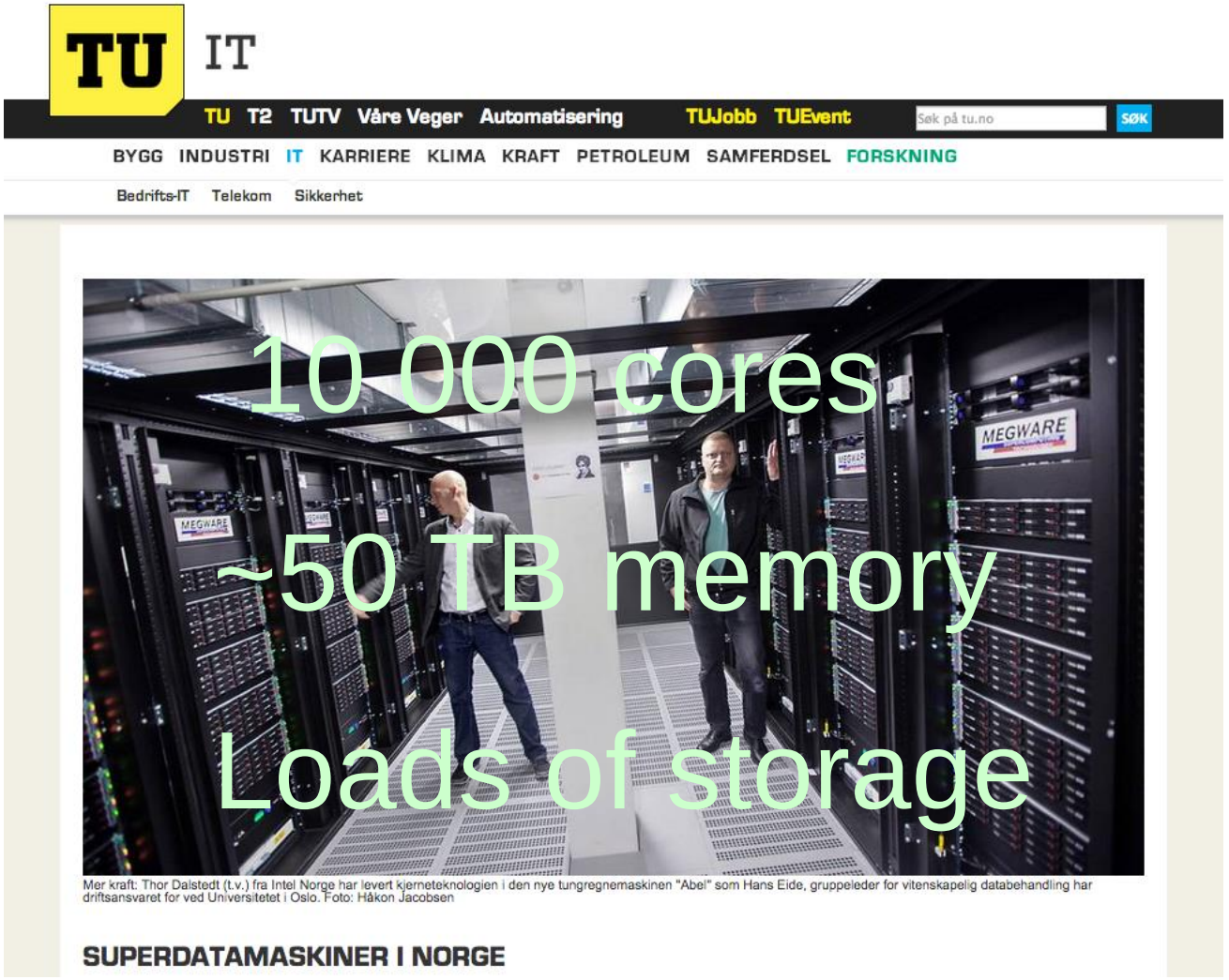
Overview

- Available computational resources
- Working with computer clusters
- Figuring out what to do
- Getting help

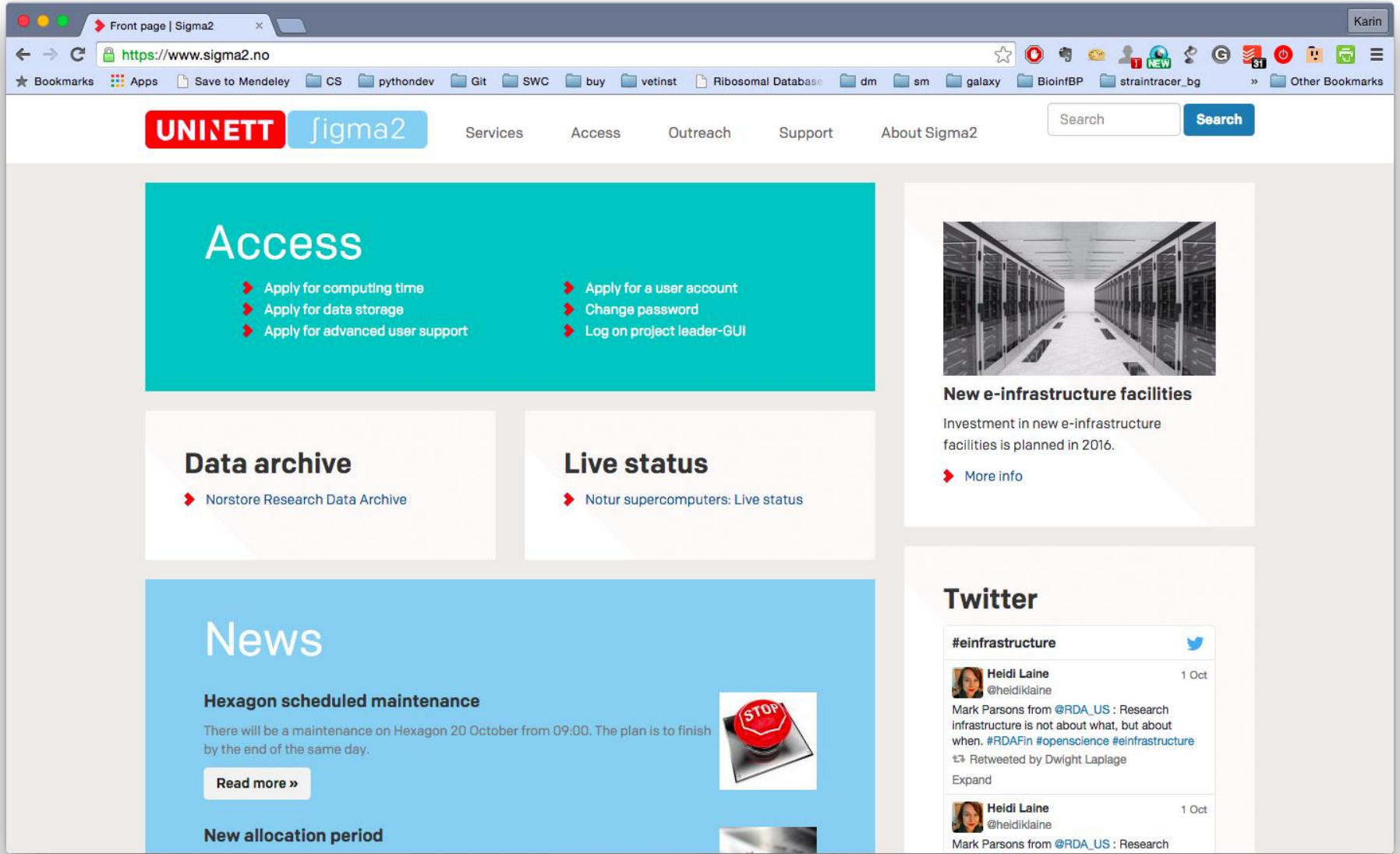
Size of data can become very big

TBs of HTS data

Abel computer cluster



www.sigma2.no



Sigma2

- The Norwegian metacenter for computational science
- Goal: provide a modern national High Performance Computing infrastructure
- Offers HPC services to Norwegian universities, colleges, research institutes and industry
- Funded by RCN, UiO, NTNU, UiB and UiT
- Gain resources by applying for them

Sigma2 services

- High Performance Computing – NOTUR
  - Offers CPU hours on five different clusters at UiB, UiT, NTNU, Iceland Univ, and UiO (Abel)
  - Access to useful software
  - Advanced user support
  - Training
- Data Storage – NORSTORE
  - Project data storage
  - Research data archive
  - Sensitive data storage

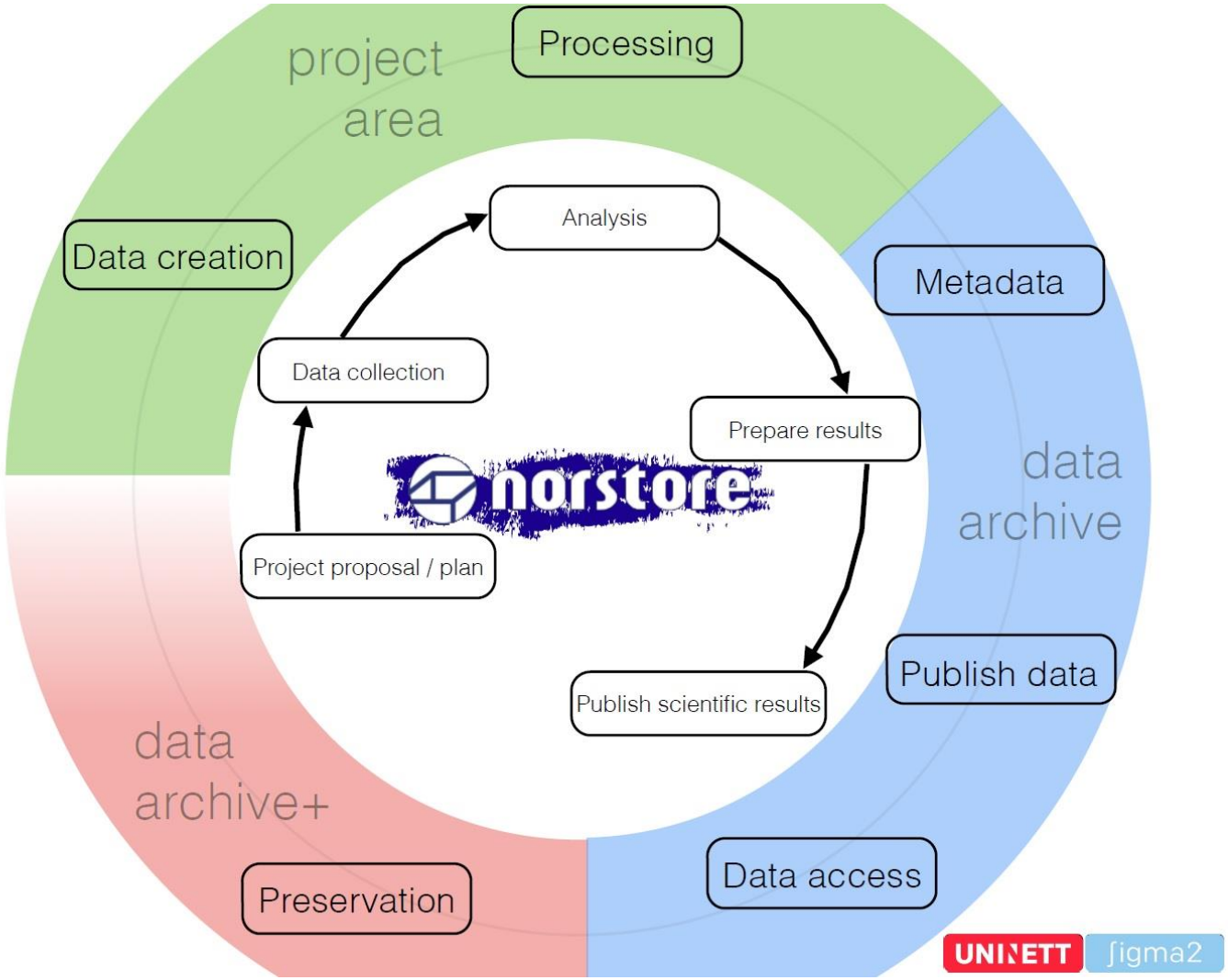
Accessing NOTUR resources

- UiO users have by default access to Abel
  - Access with normal UiO login id
  - Access to UiO CPU hours - ~10% of cluster
  - Additionally: access to freebee.abel.uio.no
  - Freebee can be used for testing purposes
- Larger scale UiO users and all others can apply to NOTUR for access
- **NO SENSITIVE (HUMAN) HTS DATA ON ABEL!**

NORSTORE – data storage

- National infrastructure for management, curation and long-term archiving of digital scientific data
- Can get storage that can be used for both
  - Active computing - /projects
  - Sensitive data
  - Archival storage
- Also, long-time archive storage with possibility for publishing research data

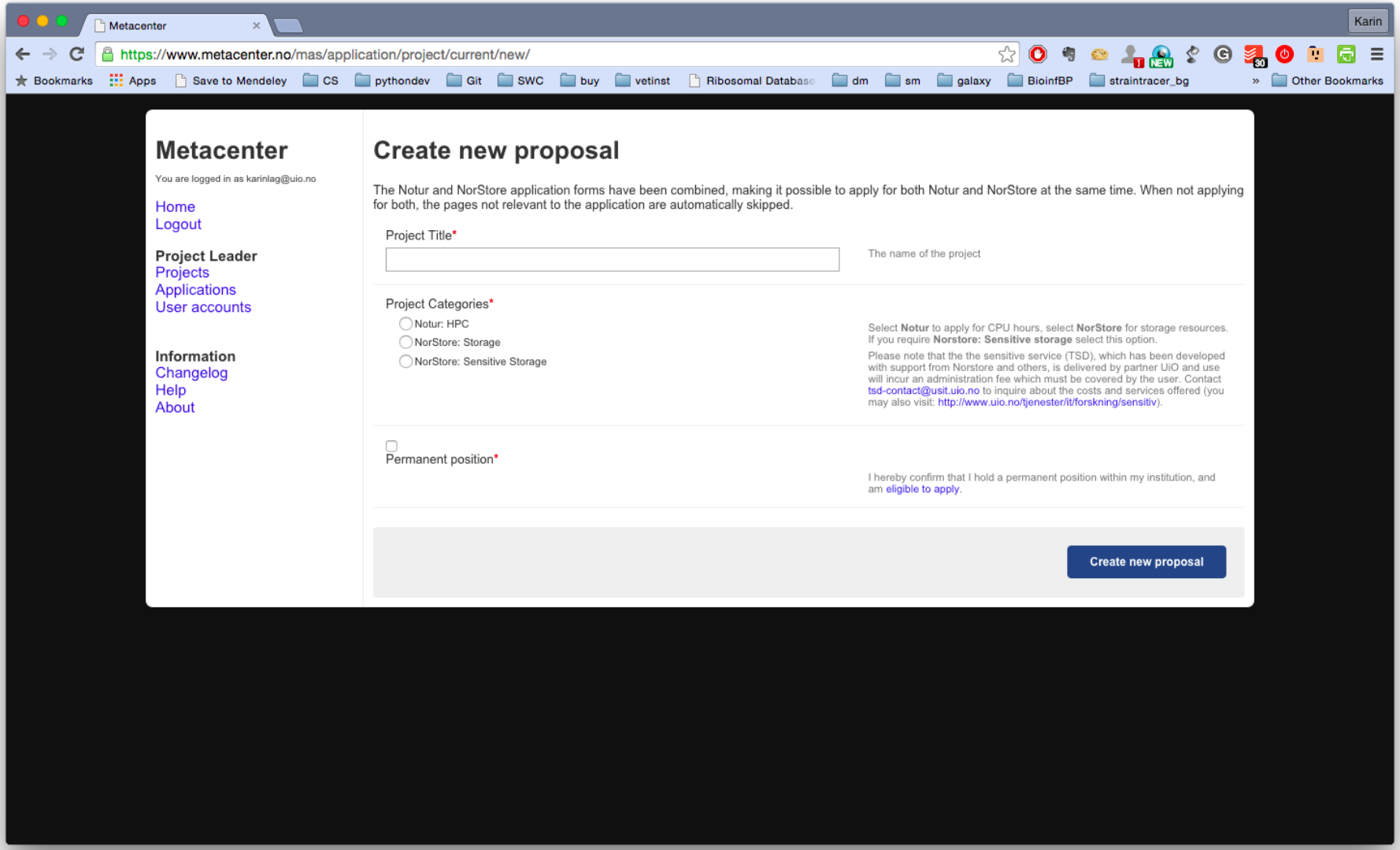
Norstore data storage



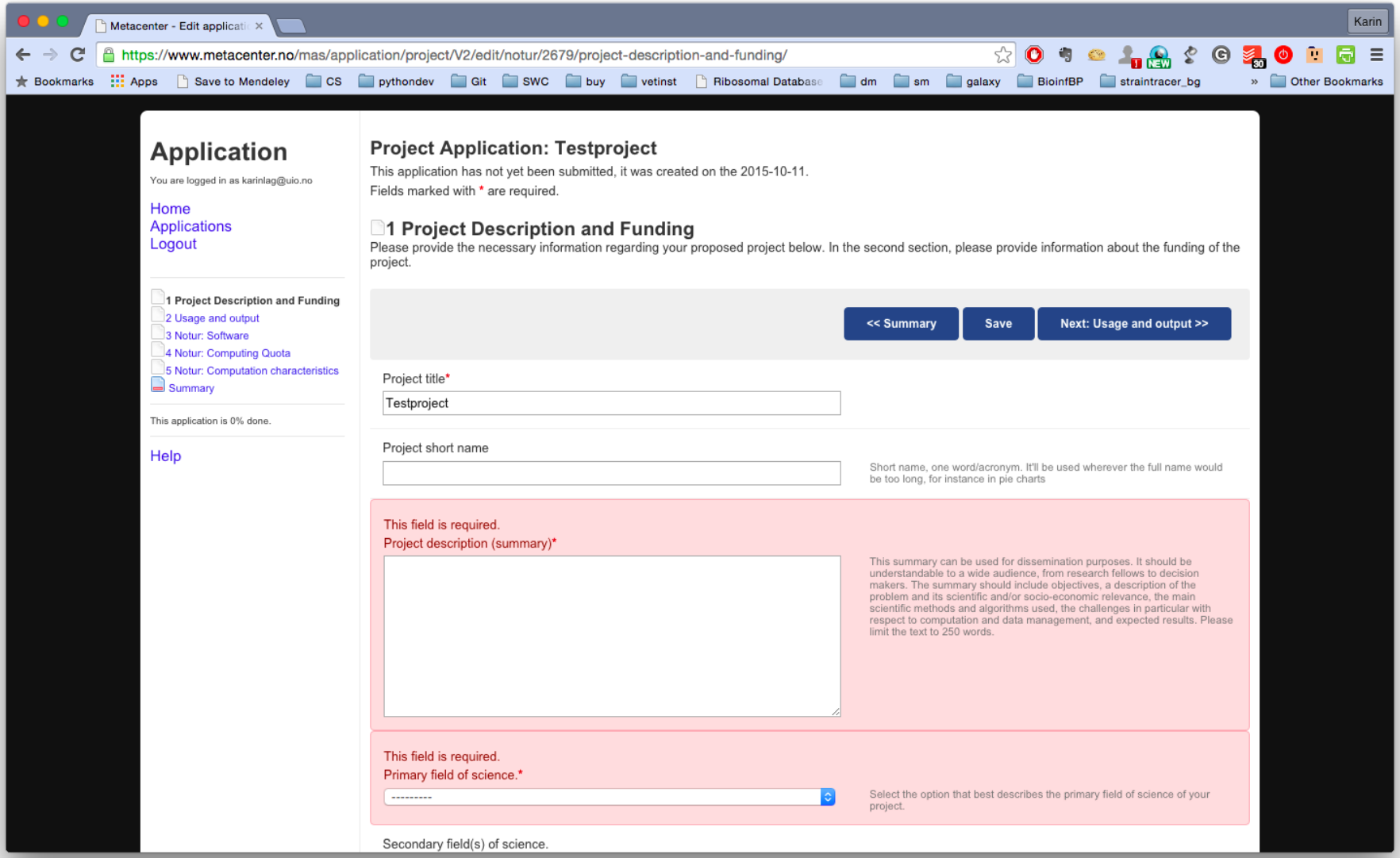
Applying for access

- Application deadline Feb/Aug
  - Same application form for storage and CPUs
- Application includes project description, describing what the resources will be used for
- Applications evaluated on scientific merit
- New users/projects given priority
- Main applicant must hold permanent position

- Small projects often given resources at once



One application form for both NOTUR and NORSTORE

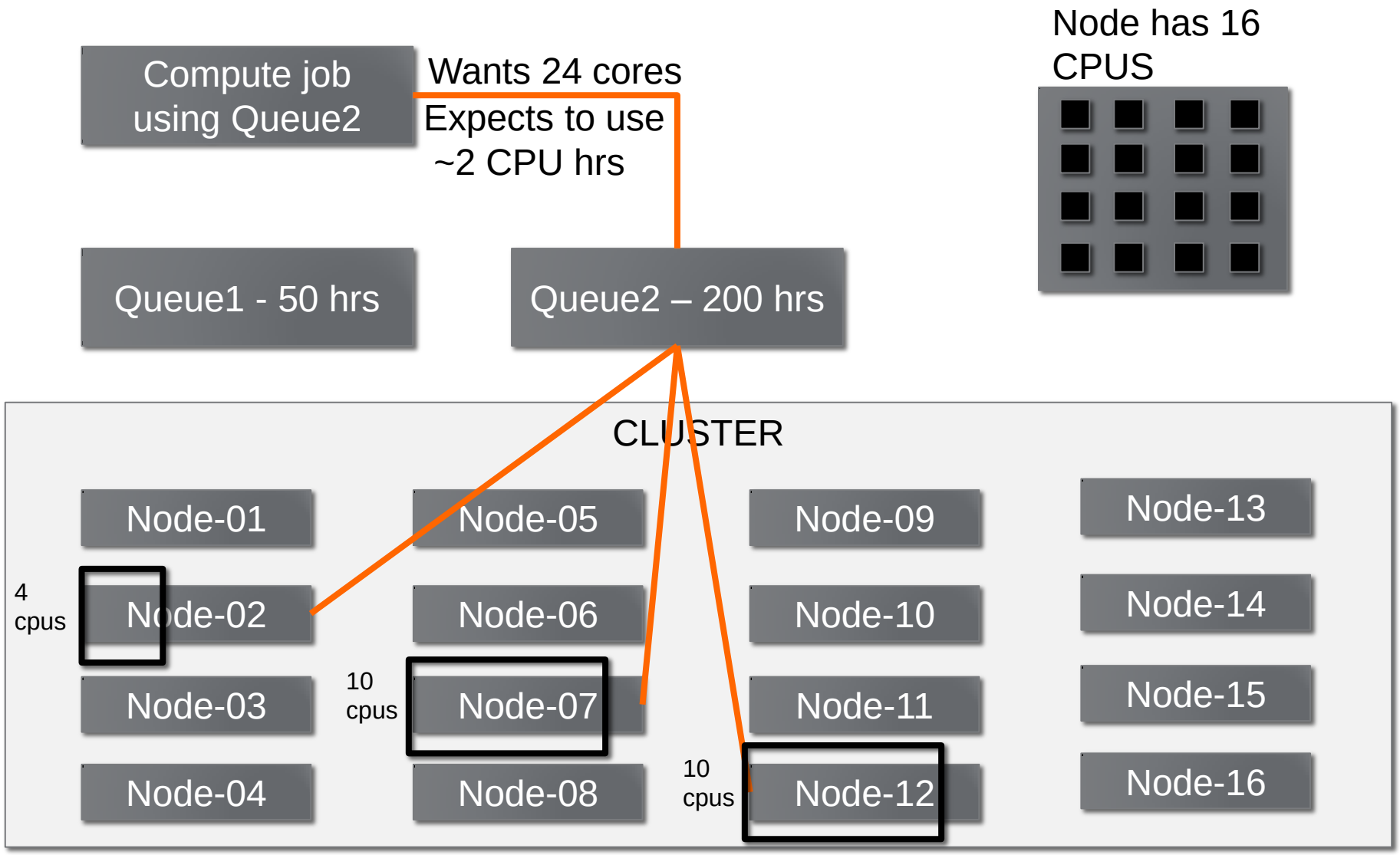

Metacenter - Edit applicati
Karin
https://www.metacenter.no/mas/application/project/V2/edit/notur/2679/project-description-and-funding/
Bookmarks
Apps
Save to Mendeley
CS
pythondev
Git
SWC
buy
vetinst
Ribosomal Database
dm
sm
galaxy
BioinfBP
straintracer_bg
Other Bookmarks
Application
You are logged in as karinlag@uio.no
Home
Applications
Logout
1 Project Description and Funding
2 Usage and output
3 Notur: Software
4 Notur: Computing Quota
5 Notur: Computation characteristics
Summary
This application is 0% done.
Help
Project Application: Testproject
This application has not yet been submitted, it was created on the 2015-10-11.
Fields marked with * are required.
1 Project Description and Funding
Please provide the necessary information regarding your proposed project below. In the second section, please provide information about the funding of the project.
<< Summary
Save
Next: Usage and output >>
Project title*
Testproject
Project short name
Short name, one word/acronym. It'll be used wherever the full name would be too long, for instance in pie charts
This field is required.
Project description (summary)*
This summary can be used for dissemination purposes. It should be understandable to a wide audience, from research fellows to decision makers. The summary should include objectives, a description of the problem and its scientific and/or socio-economic relevance, the main scientific methods and algorithms used, the challenges in particular with respect to computation and data management, and expected results. Please limit the text to 250 words.
This field is required.
Primary field of science.*

Select the option that best describes the primary field of science of your project.
Secondary field(s) of science.

Using a compute cluster

- Large computer clusters often have queue systems
- Queue systems feed compute jobs to the computer, ensuring that it is optimally used
- Queue system used is named Slurm

Slurm and the compute cluster

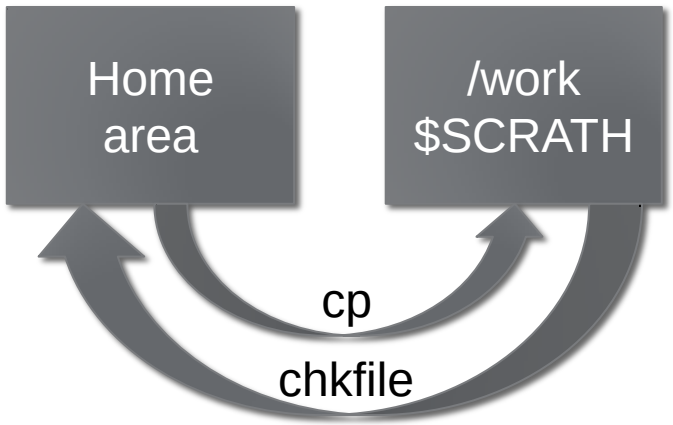


Specifying slurm scheduled job

- Need to specify:
  - Estimated time
  - Queue to run in
  - # nodes
  - # cores (cpus)
  - Amount of memory
  - Program(s) to run
- Specifications saved in Slurm job script
- Use command *sbatch* to submit job to slurm

Example slurm script

```
#!/bin/bash
#
# Job name:
#SBATCH --job-name=YourJobname
#
# Project:
#SBATCH --account=YourProject
#
# Wall clock limit:
#SBATCH --time=hh:mm:ss
#
# Number of cpus/cores
#SBATCH –ntasks=#of_cpus
#
# Max memory usage:
#SBATCH --mem-per-cpu=Size

## Set up job environment
source /cluster/bin/jobsetup

## Copy input files to the work directory:
cp MyInputFile $SCRATCH

## Make sure the results are copied back to the submit directory:
chkfile MyResultFile

## Do some work:
cd $SCRATCH
YourCommands
```



Directory created in /work with job id – directory alias is $SCRATCH. All files local to that job are saved there. Should copy job input there to begin with

Using cluster resources

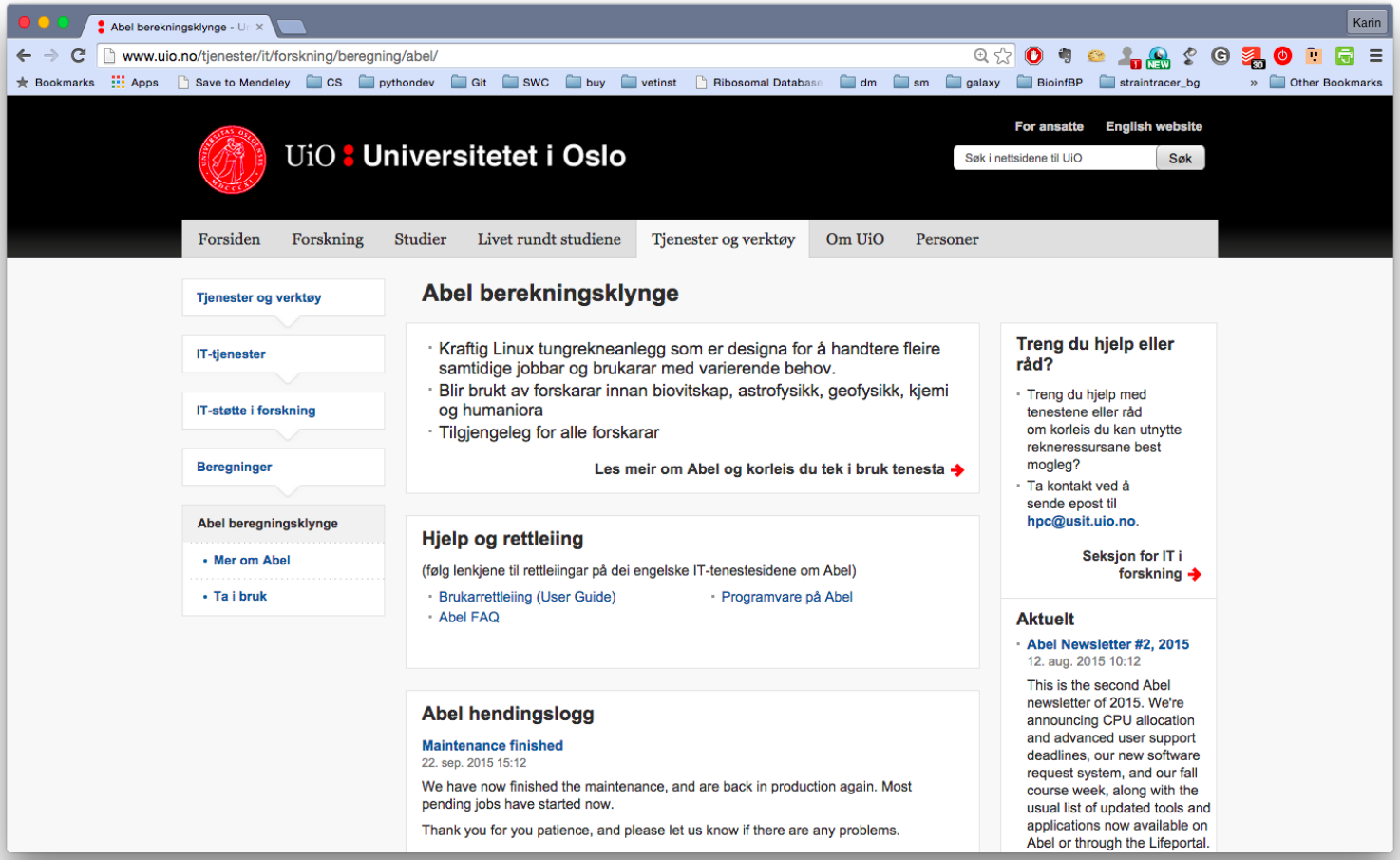
- To use more than 1 CPUs:
  - Need to ask for # CPUs in slurm script
    - --ntasks - # CPUs requested
  - Need to specify # CPUs in command to program
    - e.g. fastqc –t #cpus
  - Note: make sure --ntasks corresponds to # cpus used
- Note: can often run out of memory
  - Per node (16 cpus) – 61.5 Gb of RAM
  - Job will be killed if it uses more than allocated
  - May have to give task upper limit of memory, esp common with java programs

Useful commands

- sbatch <my_slurm_script>: put script in queue
- squeue –u <username>: list all jobs in queue for a specific user
- scancel <job id>: cancel job
- scancel –u <username>: cancel all jobs for user
- cost: how much of the allocation is used
- dusage: how much storage is used

Help on Abel and Slurm

- www.uio.no/tjenester/it/forskning/beregning/abel/



Bioinformatics software on abel

- Lots of software available
- Different people need different kinds of software
- Solved this by packaging SW in modules
- > 400 modules available
- Useful commands:
  - *module avail*: lists all available modules
  - *module load modulename*: loads that module
  - *module list*: shows all currently loaded modules
- Use in slurm script: include module load in script

Modules…

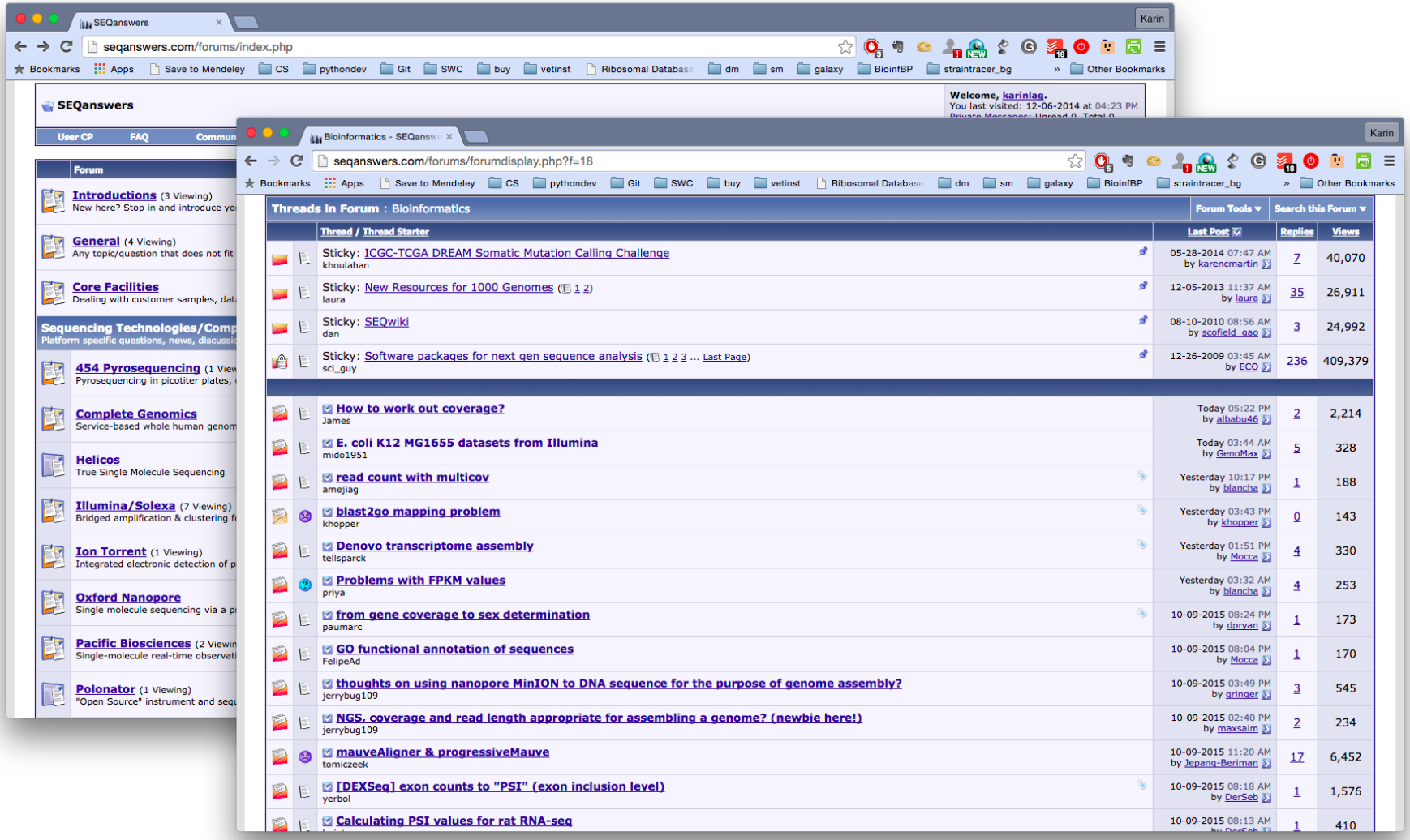
```
[karinlag@titan ~]$ module avail

------------------------------------------------------------------ /usr/share/Modules/modulefiles -------------------
dot     module-git  module-info modules   null    use.own

------------------------------------------------------------------------ /cluster/etc/modulefiles ----------------------
454apps/1.1.03.24           gaussian/g09b01                ncview/2.1.2(default)         prottest/3.2(default)
454apps/2.0.01.02           gaussian/g09c01                netcdf/4.2.1.1(default)       pypar/2.1.5(default)
454apps/2.3                 gaussian/g09d01(default)       netcdf.gnu/4.2.1.1(default)   python2/2.7.3(default)
454apps/2.5.3               gcc/4.7.2                      netcdf.intel/4.2.1.1(default) python2/2.7.6
454apps/2.6                 gcc/4.8.0                      netcdf.pgi/4.2.1.1(default)   python3/3.2.3(default)
454apps/2.7                 gcc/4.8.2                      newbler/1.1.03.24             python3/3.4.0
454apps/2.8(default)        gcc/4.9.0                      newbler/2.0.01.02             qiime/1.5.0(default)
454apps/2.9                 gcc/4.9.1                      newbler/2.3                   qiime/1.8.0
454apps/3.0                 gdal/1.9.1(default)            newbler/2.5.3                 quast/2.3(default)
abyss/1.3.4(default)        geneid/1.4.4(default)          newbler/2.6                   R/2.15.2
adf/2010.02b                genemark-es/2.3e               newbler/2.7                   R/2.15.2.shlib
adf/2012.01b                genemarks/19032014             newbler/2.8                   R/3.0.2.shlib
adf/2013.01(default)        geos/3.3.5(default)            newbler/2.9                   R/3.0.3
adf/2014.01                 ghc/7.4.2(default)             newbler/3.0(default)          R/3.0.3.profmem
adf_gpu/2014.01             gmap/2013-09-30                nfuse/0.2.1(default)          R/3.0.3.shlib
allpathslg/48777(default)   gmap/2013-11-27(default)       nltk/2.0.1(default)           R/3.1.0
amber/12(default)           gnu_parallel/20131022(default) notur/0.1(default)            R/3.1.0.profmem
amos/3.1.0(default)         gnuplot/4.6.0(default)         novocraft/V3.02.05(default)   R/3.1.0.shlib
ampliconnoise/1.25(default) gnuplot/4.6.3                  ocaml/4.00.0(default)         R/3.1.1(default)
ampliconnoise/1.29          graphviz/2.28.0(default)       octave/3.6.3(default)         R/3.1.1.gnu
aragorn/1.2.36(default)     grib_api/1.12.3                open64/5.0(default)           R/3.1.1.profmem
asreml/2.00ah               gsl/1.15(default)              openifs/38r1v04               R/3.1.1.shlib
```

How to figure out how to do things

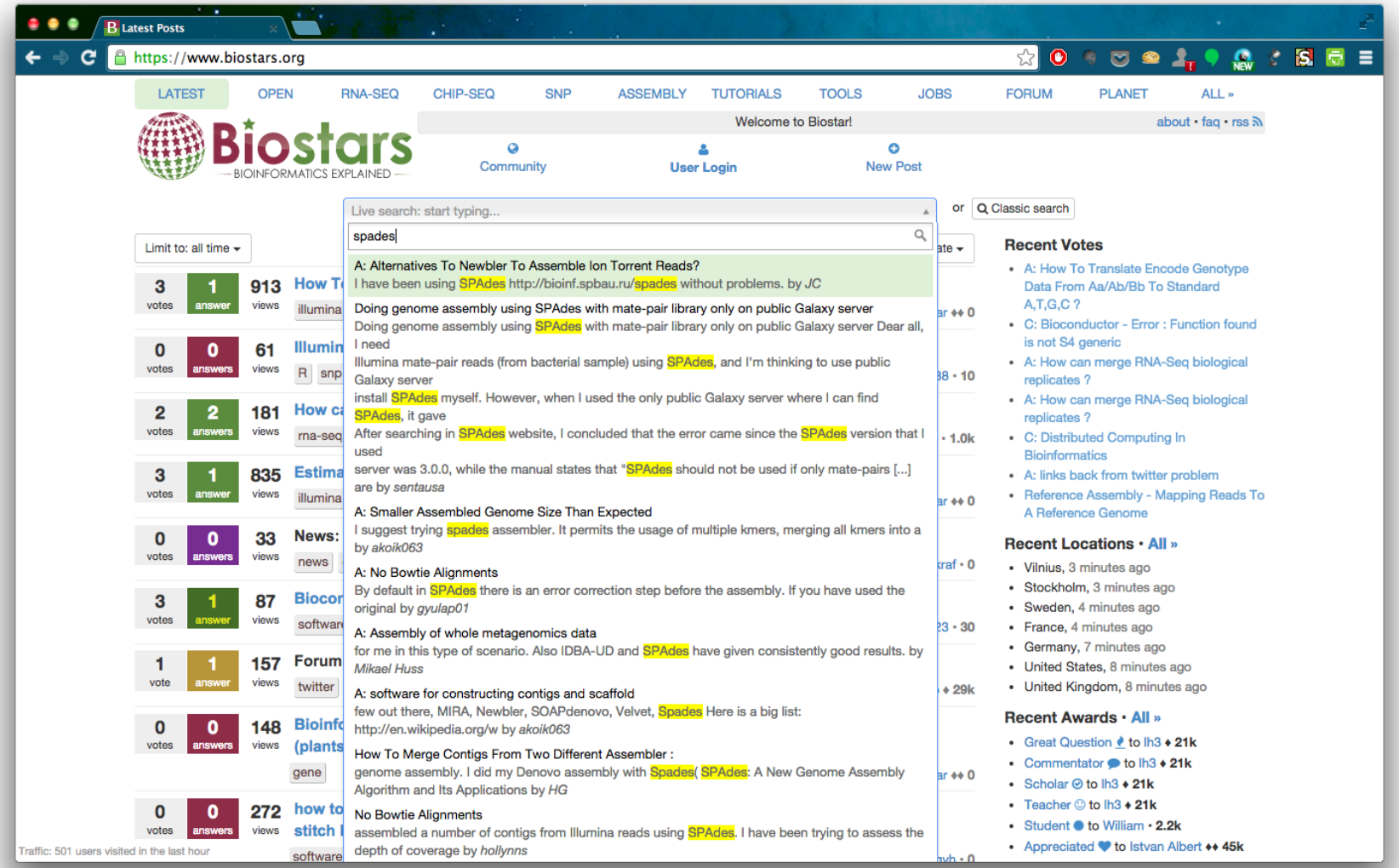
- Read papers that do what you want to do – repeat their methods on your data
- Signup and use:
  - seqanswers.com
  - biostars.org
- Find blogs from people who do what you do
- Read tutorials/manuals on the software you use
- Find email lists to software

- Twitter can also be very helpful

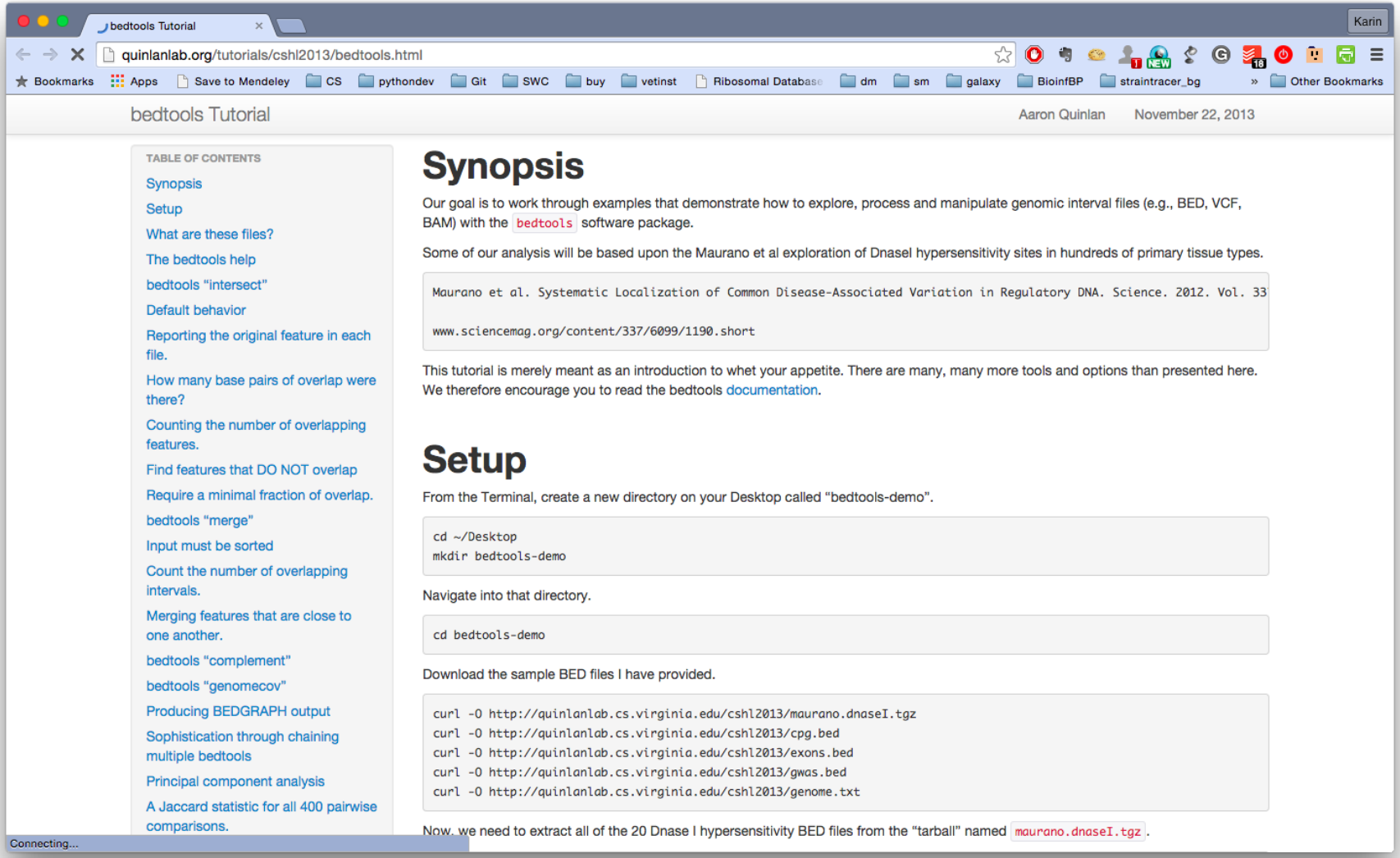
seqanswers.com



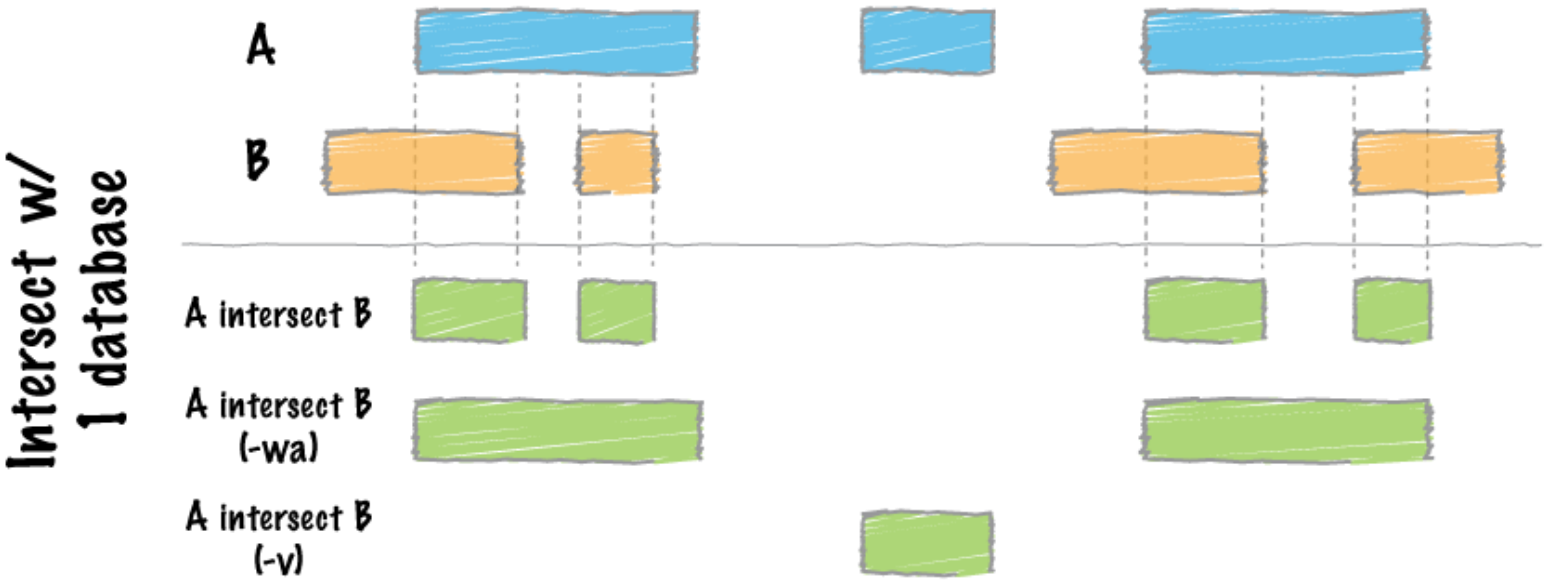
biostars.org



Reading blogs



Reading blogs



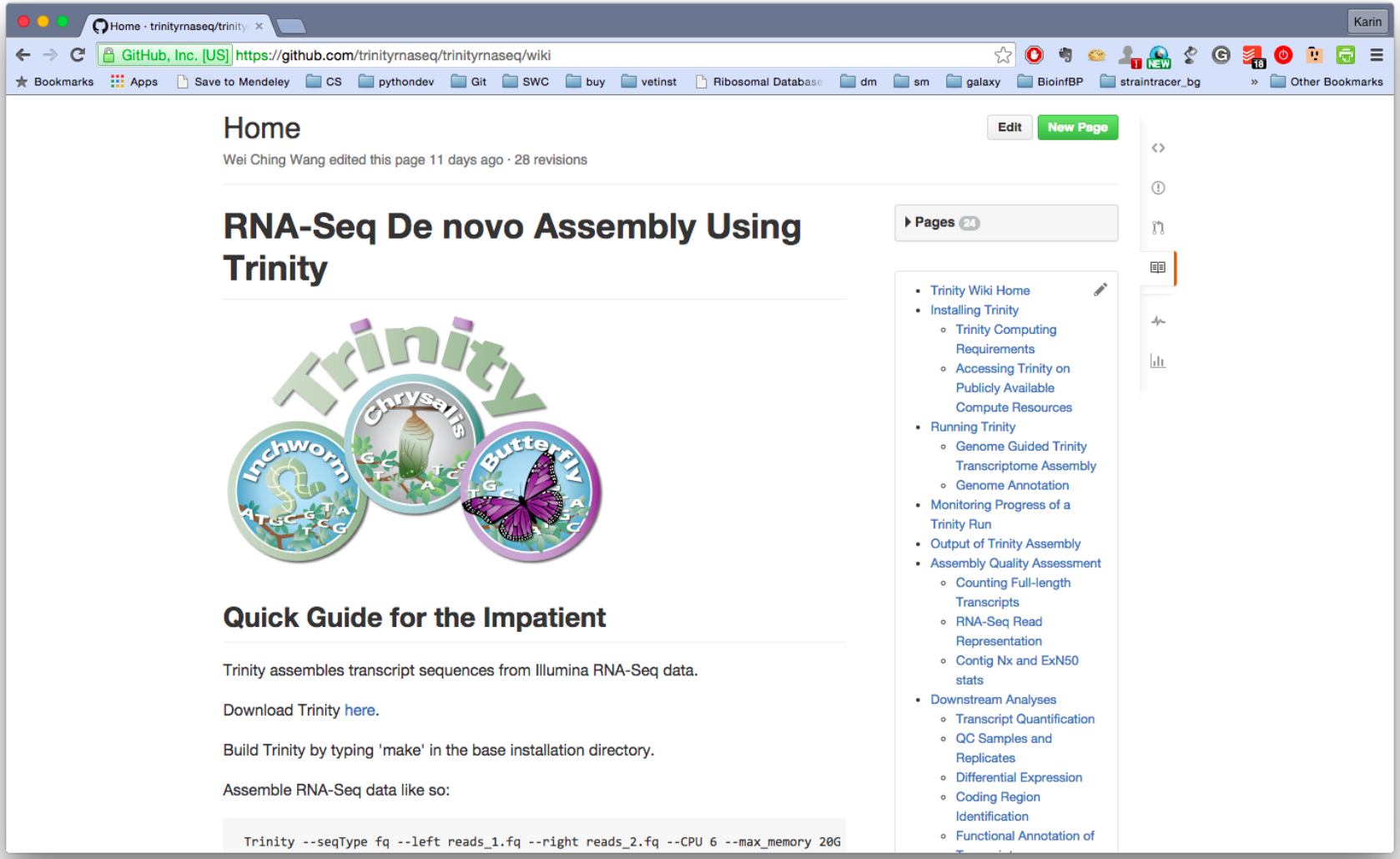
Default behavior

By default, `intersect` reports the intervals that represent overlaps between your two files. To demonstrate, let's identify all of the CpG islands that overlap exons.

```
bedtools intersect -a cpg.bed -b exons.bed | head -5
chr1    29320   29370   CpG:_116
chr1    135124  135563  CpG:_30
chr1    327790  328229  CpG:_29
chr1    327790  328229  CpG:_29
chr1    327790  328229  CpG:_29
```

http://quinlanlab.org/tutorials/cshl2013/bedtools.html

Find tutorials and manuals for sw



What to do if you get an error?

- Try with a different data set – often good to try with a smaller, well-known data set
- Change version of program if another one are available

- Google is your friend – google error message
- Check seqanswers and biostars
- Look at the webpage for the software – is your error mentioned?
- Check email list for software – is error mentioned?
- Write to software authors and/or sw email list, have they seen this before?
- Also – if on Abel/TSD: email the helpdesk

What to include in an error report

(0) What is my environment?

1. What did I do?

2. What result did I expect?

3. What result did I get?

(4) Why is this incorrect?

Error report - translated

- (Shortly) explain purpose of analysis
- Name of program, incl. version
- Full command line, incl. all options
- Copy-paste of error from start of program
- For USIT: include file system location

- Goal: help person should be able to recreate the bug, without having to ask you more questions

Finding your way

- Many ways to the same goal
  - But: which one is “right”?
- Top trick: keep your biological question in mind
  - Software often designed to solve one particular biological problem
  - Keep your biological expectations in mind – if result breaks with that, stop and rethink
- Figure out expectations by using a “known” entity
  - either a small data set or
  - data set you already know the biological answer for

The bioinformatical process

- Working with bioinformatics *is* experimental work
- Not all software are appropriate for
  - Your biological problem
  - Your data
- Need to experiment to find the right combination of software for your problem
- Bioinformatics is an iterative process:
  - Try program on data
  - Evaluate result
  - Decide if results are appropriate

Redo several times

Log your work

- Bioinformatics is an iterative process
  - May not remember what solution you ended up with, or for that matter, why
- Keep a log with:
  - Today's date
  - What you are doing
  - Program version
  - Commandline
  - A little bit of output
  - Evaluation of output

Questions?