next
Artificial
Intelligence in
Medicine

Status of computing resource usage and SW repository 2024

Francesca Lizzi & Ian Postuma

Computing Resources

https://baltig.infn.it/nextaim/code

There is a **wiki** containing all the informations on how to connect to and use the available computing resources.

National resources: **AI_INFN**, Recas.

Local resources: Pavia, Pisa.

For those new to **GIT** there is a brief description on how to interact with a git repository.

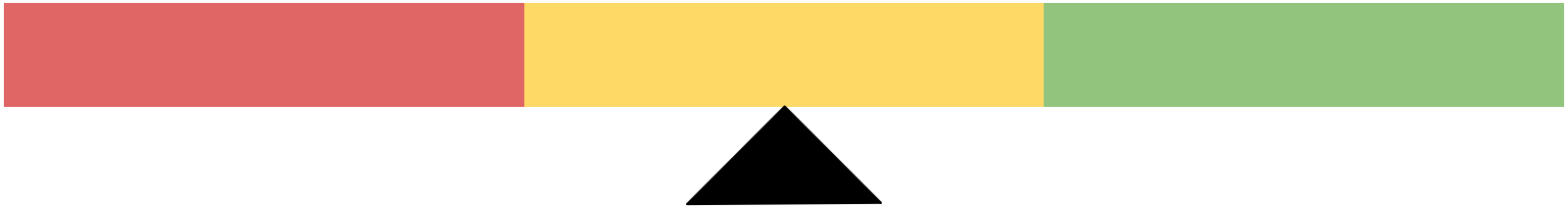
Mission

As WP4 conveners, we plan to:

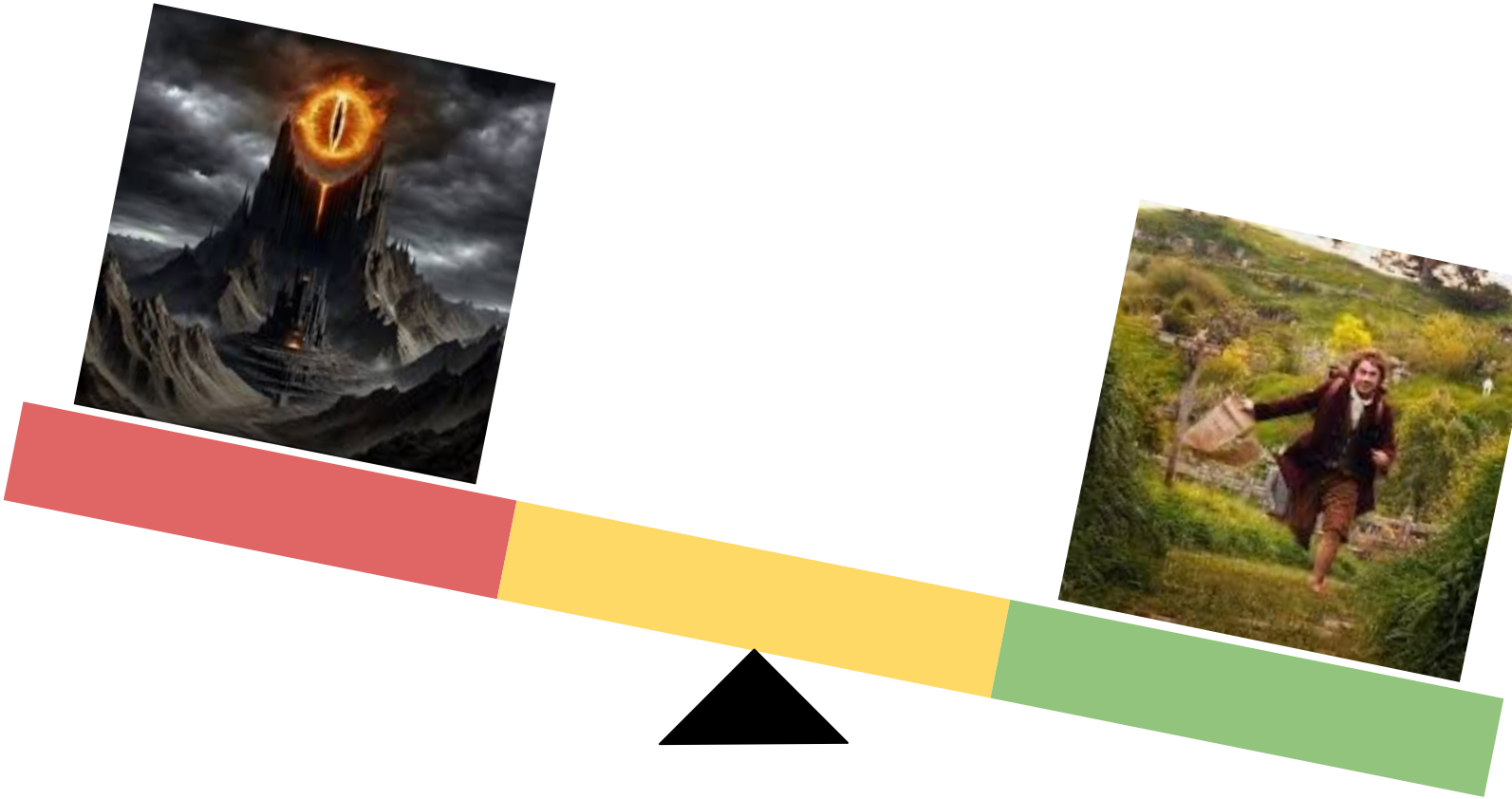
- Organize tutorials to use computing resources and baltig
- Help in building the shared software repository
- Grant access to machines and computing resources in the collaboration.

Each application and/or software needs specific computing resources, hardware, software, virtual environments and documentation, so that you can write us to discuss how to provide the best configuration for your scopes.

Status of the repository upload



Status of the repository upload



Status of the repository upload

Sedi partecipanti	Task	Topic
FE, LNS, NA, PI	T1	Radiomics in Digital Breast Tomosynthesis (DBT)
BO, FE, NA, PI	T2	Super-Resolution in Medical Imaging
BO, CT	T3	Radiomics in prostate cancer
BO, CT	T4	Radiomics and DL in tcMRgFUS
BO, FI, GE, LNS	T5	Nuclear Imaging Quantification and Radiomics
BA, CA, CT, PD, PI	T6	Connectivity in functional MRI and EEG
CA, CT, FE, FI, GE, MI, PI, PV	T7	Radiomics and Deep Learning analysis of CT and patients' data in COVID-19
MI, PI, PV	T8	Radiomics and ML-segmentation on Facio-Scapulo-Humeral dystrophy (FSHD), lung and liver tumor
PV	T9	ML on Imaging data of 10B uptake tracks and dose monitoring by Compton cameras
FI, PI	T10	Artificial intelligence for monitoring RT response in soft-tissue sarcomas
FE, PD	T11	Machine Learning techniques for cardiological applications
FI, PI	T12	Application of NLP techniques to clinical notes towards the automated reading of instrumental data

Code on other repositories

As many of us uses Github to develop their algorithm, here we list the link to reach the code on github:

1. DNetPro Official implementation of the DNetPRO algorithm published on Scientific Reports by Curti et al. Scientific Reports (WP1 and WP2)
2. ClearLung Official Implementation of the ClearLung algorithm for radiomic analysis of CT Lung scans.
3. delta-BIT DELTA-BIT stands for Deep-learning Local TrActography for BraIn Targeting, it comes from the idea to make faster the FSL pipeline for probabilistic tractography.
4. PVSquared2 Machine learning for screening and predicting the best surface modifiers for a rational optimization of efficient perovskite solar cells. (https://doi.org/10.1063/5.0214736)

T1 - Radiomics in Digital Breast Tomosynthesis (DBT)

https://baltig.infn.it/nextaim/dbt_classifier

DBT_classifier

Star 0 | Fork 0

main | dbt_classifier / | + | History | Find file | Edit | Code

Minor updates
Gianfranco Paternò authored 6 months ago — 96c528e4

Name	Last commit	Last update
utils	Minor updates	6 months ago
.gitignore	Added gitignore	6 months ago
DBT_classifier.py	Minor updates	6 months ago
LICENSE	Initial commit	7 months ago
README.md	Minor updates	6 months ago
dataset_creator_DUKE.py	Minor updates	6 months ago
dataset_patient_folder_cr...	Minor updates	6 months ago

Project information

- 17 Commits
- 1 Branch
- 0 Tags
- 131 KiB Project Storage

README
European Union Public License 1.2
+ Add CHANGELOG
+ Add CONTRIBUTING
+ Add Kubernetes cluster
+ Set up CI/CD
+ Add Wiki
+ Configure Integrations

T2 - Super resolution in medical imaging

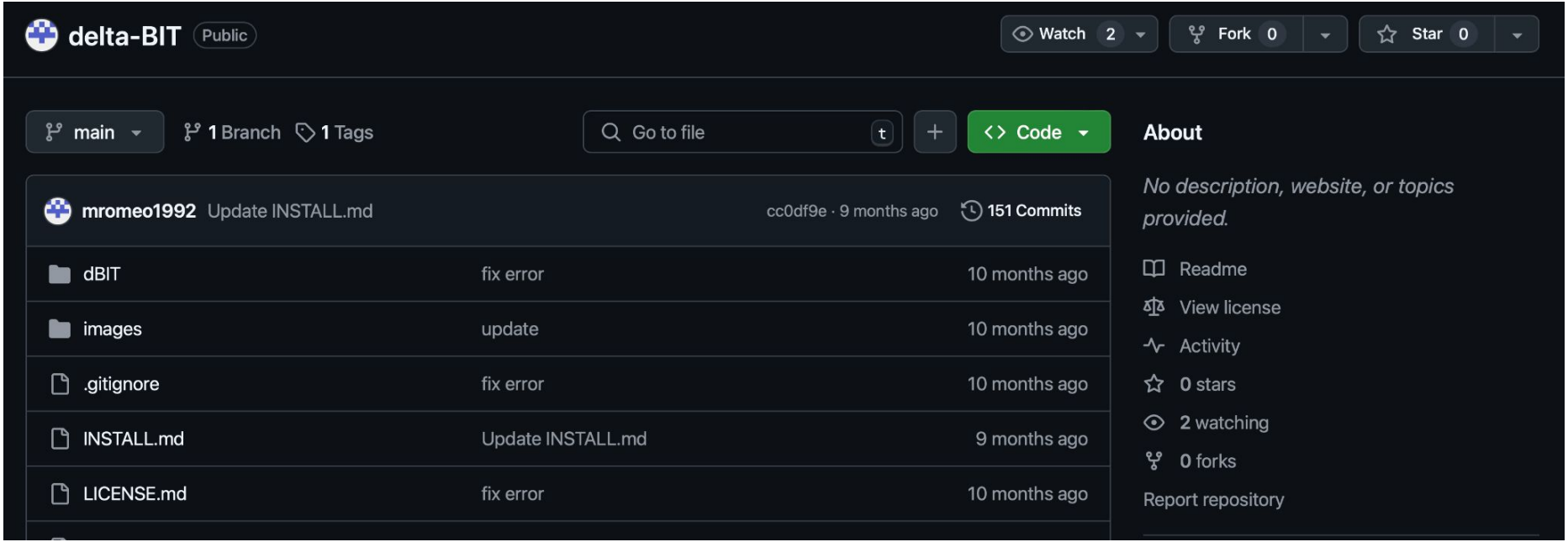
?

T3 - Radiomics in prostate cancer.

?

T4 - Radiomics and DL in tcMRgFUS

https://github.com/mromeo1992/delta-BIT



T5 - Nuclear imaging quantification and radiomics

https://baltig.infn.it/nextaim/radiomics_matlab_cnrinfn

Radiomics_matlab_CNRINFN

Star 0 Fork 0

main radiomics_matlab_cnrinfn / + History Find file Edit Code

Add new file
Alessandro Stefano authored 7 months ago fc6c3524

Name	Last commit	Last update
DA.m	Upload New File	7 months ago
feature_selection.m	Upload New File	7 months ago
license.txt	Upload New File	7 months ago
main_radiomics.m	Upload New File	7 months ago
next_AIM.xlsx	Upload New File	7 months ago
readme.md	Add new file	7 months ago
train_test.m	Upload New File	7 months ago

Project information

- 7 Commits
- 1 Branch
- 0 Tags
- 1.5 MiB Project Storage

- README
- BSD 2-Clause "Simplified" License
- Wiki
- + Add CHANGELOG
- + Add CONTRIBUTING
- + Add Kubernetes cluster
- + Set up CI/CD
- + Configure Integrations

T6 - Connectivity in functional MRI and EEG

https://baltig.infn.it/nextaim/func_abide

func_ABIDE

Star 0 Fork 0

main func_abide / + History Find file Edit Code

Merge branch 'saponaro-main-patch-99137' into 'main'
Francesca Lizzi authored 8 months ago

78152f6d

Name	Last commit	Last update
joint_fusion_DL_model	g	8 months ago
.gitignore	Update .gitignore	1 year ago
my_abide_library.py	Revert "modifica prima frase"	1 year ago
struct_func.py	programma classificazione	1 year ago

Project information

24 Commits
2 Branches
0 Tags
83.4 MiB Project Storage

+ Add README
+ Add LICENSE
+ Add CHANGELOG
+ Add CONTRIBUTING
+ Add Kubernetes cluster

T6 - Connectivity in functional MRI and EEG

https://baltig.infn.it/nextaim/multimodal-neuroimaging

Multimodal Neuroimaging

Star 0 | Fork 0

main | multimodal-neuroimaging / | History | Find file | Edit | Code

added optional save/load flags for filter functions and improved documentation
Gianmarco Tiddia authored 2 months ago — a90aab44

Name	Last commit	Last update
multimodal_neuroimaging	added optional save/load flags for ...	2 months ago
phenotipic_data_informat...	updated notebook with the instruct...	3 months ago
.gitignore	Removed .DS_Store + Updates in	3 months ago
LICENSE	Initial commit	3 months ago
README.md	Updates in README	3 months ago

Project information

- 43 Commits
- 5 Branches
- 0 Tags
- 408 KiB Project Storage

- README
- GNU AGPLv3
- Add CHANGELOG
- Add CONTRIBUTING
- Add Kubernetes cluster

T7 - Radiomics and DL of CT and patients data of C19

https://baltig.infn.it/nextaim/lungquant … and UI

ORIGINAL ARTICLE Open Access

A multicenter evaluation of a deep learning software (LungQuant) for lung parenchyma characterization in COVID-19 pneumonia

Camilla Scapicchio[1,2*†], Andrea Chincarini[3†], Elena Ballante[4,5], Luca Berta[6,7], Eleonora Bicci[8], Chandra Bortolotto[9,10], Francesca Brero[5], Raffaella Fiamma Cabini[5,11], Giuseppe Cristofalo[12], Salvatore Claudio Fanni[13], Maria Evelina Fantacci[1,2], Silvia Figini[4,5], Massimo Galia[12], Pietro Gemma[14], Emanuele Grassedonio[12], Alessandro Lascialfari[5], Cristina Lenardi[7,15], Alice Lionetti[9], Francesca Lizzi[1,2], Maurizio Marrale[16,17], Massimo Midiri[12], Cosimo Nardi[8], Piernicola Oliva[18,19], Noemi Perillo[14], Ian Postuma[5], Lorenzo Preda[9,10], Vieri Rastrelli[8], Francesco Rizzetto[20,21], Nicola Spina[13], Cinzia Talamonti[22,23], Alberto Torresin[6,7,15], Angelo Vanzulli[6,24], Federica Volpi[13], Emanuele Neri[13,25] and Alessandra Retico[2]

T8 - FSHD, lung and liver tumour

Computer Methods and Programs in Biomedicine 256 (2024) 108399

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

Myo-regressor Deep Informed Neural NetwOrk (Myo-DINO) for fast MR parameters mapping in neuromuscular disorders

Leonardo Barzaghi [a,b,c,*], Francesca Brero [c,d], Raffaella Fiamma Cabini [a,c,e], Matteo Paoletti [b], Mauro Monforte [f], Francesca Lizzi [g], Francesco Santini [h,i], Xeni Deligianni [h,i], Niels Bergsland [j,k], Sabrina Ravaglia [n], Lorenzo Cavagna [m], Luca Diamanti [n], Chiara Bonizzoni [b], Alessandro Lascialfari [c,d], Silvia Figini [o,p], Enzo Ricci [f], Ian Postuma [c,d], Anna Pichiecchio [l,b]

T8 - FSHD, lung and liver tumour

https://github.com/niguardateam/covid-classifier

README License

CLEARLUNG

Clinical Extraction And Radiomics on LUNGs

Summary

Welcome to the CLEARLUNG framework! This package provides both clinical and radiomic analysis of lung CT scans. It was developed as a Master Thesis project.

Contributors 3

andreasala98 Andrea Sala

GiuliaZorzi Giulia Zorzi

scarrazza Stefano Carrazza

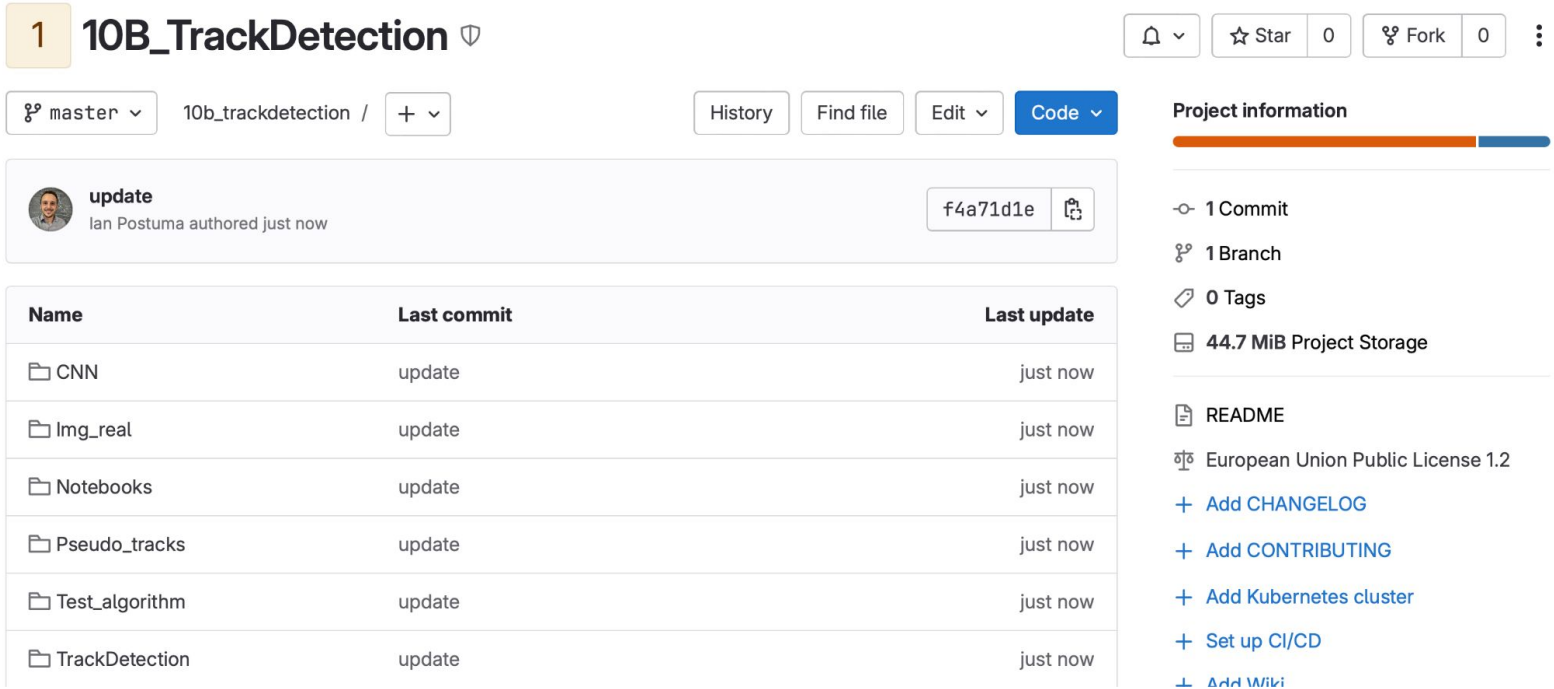
Languages

Python 87.4% HTML 12.4%

Shell 0.2%

T9 - 10B uptake measurements and compton cameras

https://baltig.infn.it/nextaim/10b_trackdetection

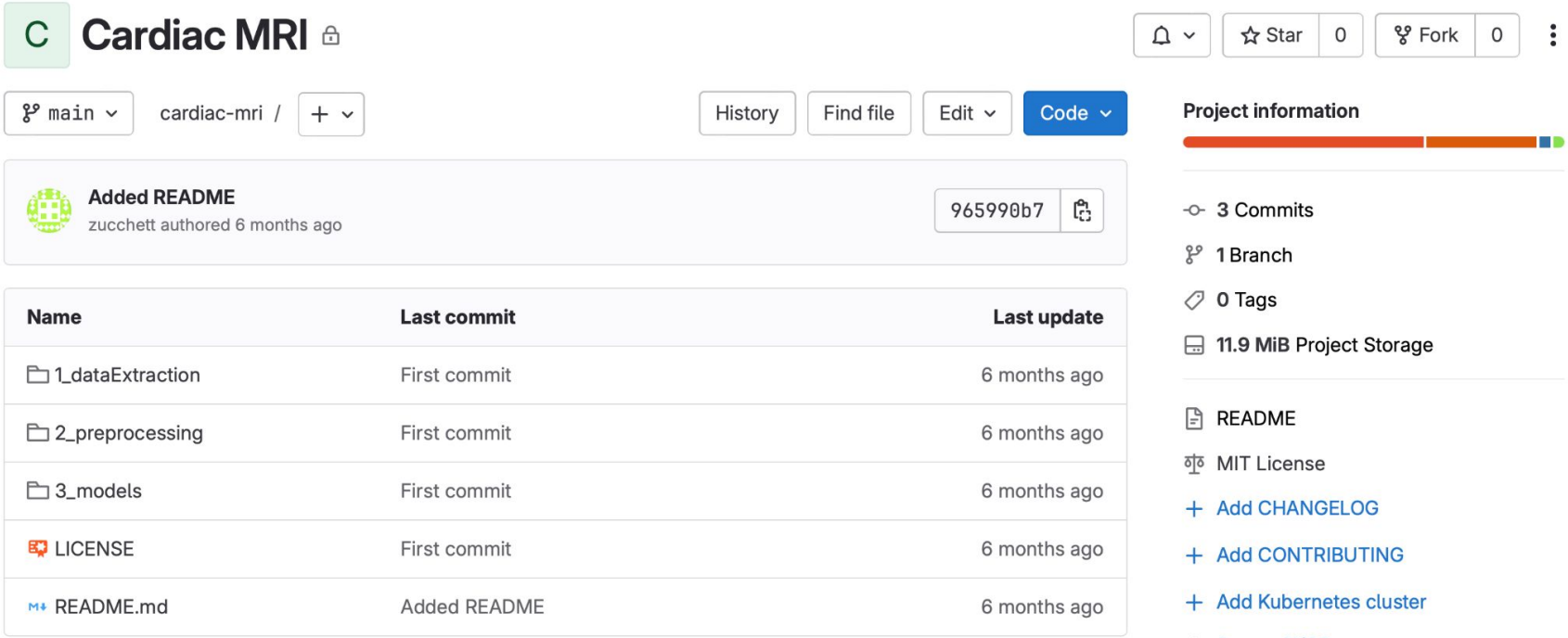


T10 - AI for monitoring RT response in soft tissue sarcoma

?

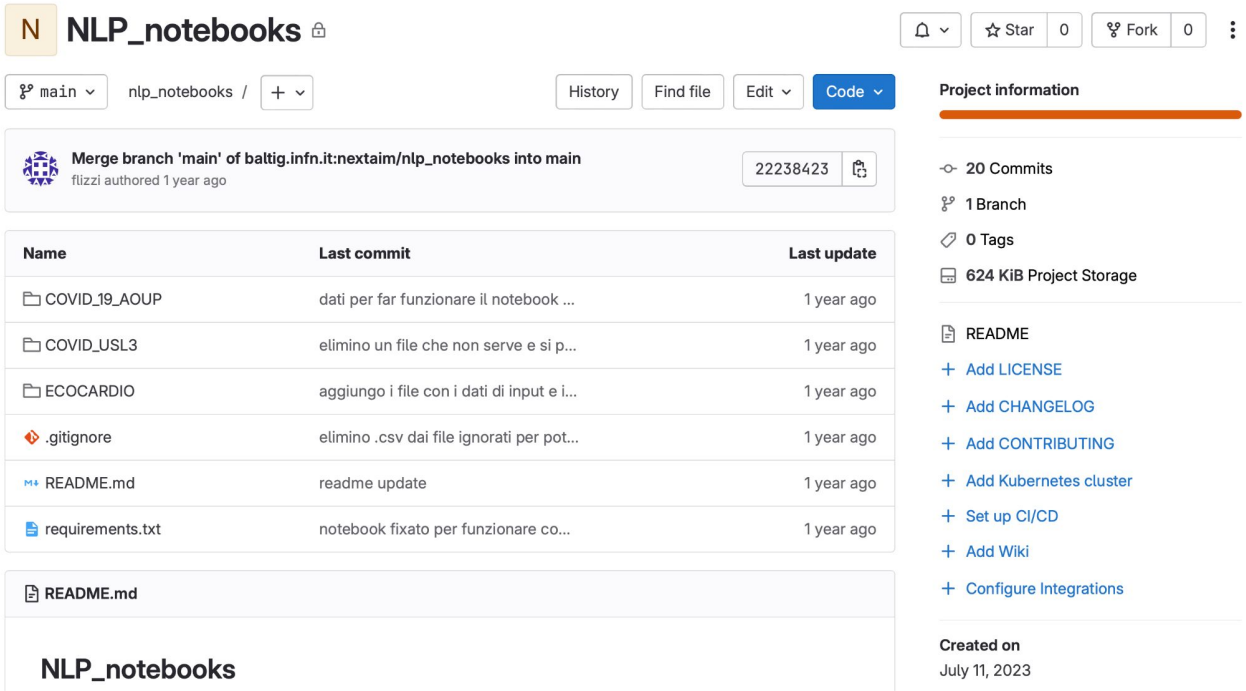
T11 - ML for cardiological applications

https://baltig.infn.it/nextaim/cardiac-mri



T12 - NPL application to clinical notes

https://baltig.infn.it/nextaim/nlp_notebooks



Thank you for the attention!

francesca.lizzi@sns.it
ian.postuma@pv.infn.it

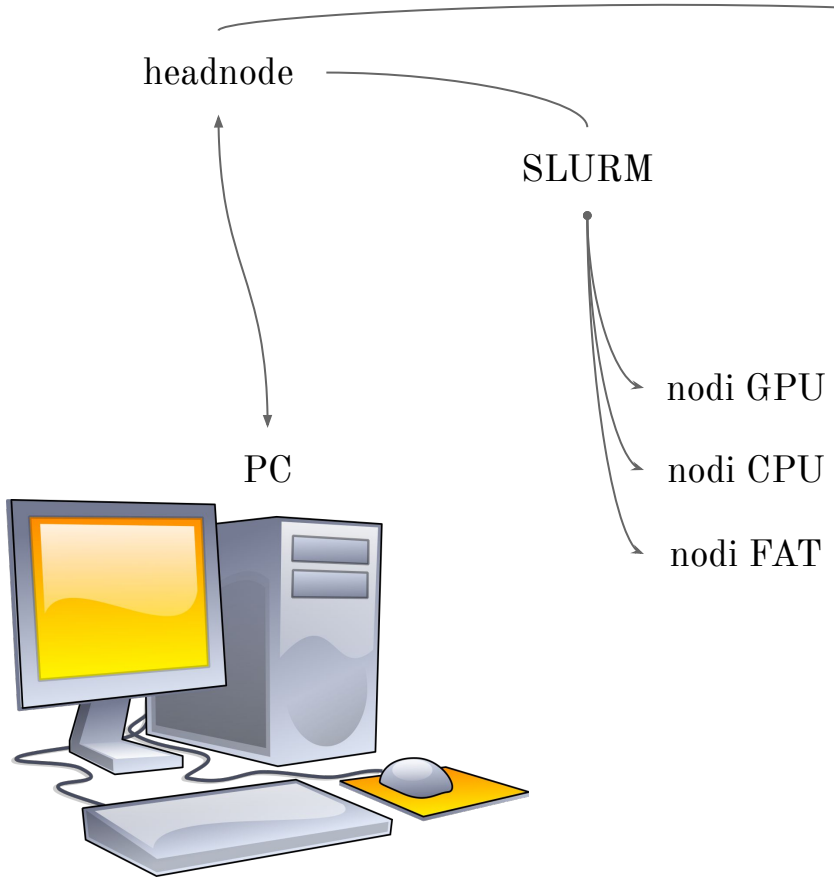
Computing Resources (PV)

- 2 Desktop Workstation
  - WS1: Ubuntu, CPU i9-10850, 1.5TB storage, RTX3060 12GB, 64GB RAM
  - WS2: Linux Mint, CPU AMD FX-8350, 2TB storage, RTX 2070 8GB, 32GB RAM
- 1 remote Server - EOS di UniPv
  - Sistema Linux (CentOs 7)
  - 160 TB di storage
  - 7 nodi FAT (ognuno con 768 GB RAM)
  - 7 nodi GPU (ognuno con 128GB RAM e 2 NVIDIA Tesla V100 da 32GB RAM)
  - 7 nodi standard (ognuno con 128GB RAM)

Accessing Computing Resources (PV)

- 2 Desktop Workstation
  - ssh
  - locally maintained - **if** needed we can share.
- 1 remote Server - EOS di UniPv
  - there are some restrictions for people outside UniPv
  - registration -> https://forms.gle/tiH9KDPakGPpGz2H8
  - ssh
  - SLURM batch system

EOS system overview (PV)



EOS cluster

Example workflow on EOS batch (PV)

- Create python virtual environment with CONDA
- Write code and update a GIT repository (baltig.infn.it)
- Load the data
- Create a slurm submission file & Train
  - Tell slurm which and how much resources are needed
  - Activate system modules for GPU and CONDA
  - Load the specific virtual environment
  - Execute the python scripts
  - if job terminates badly slurm log files and error outputs are useful
- Use trained network locally

While EOS batch is running (PV)

- Check job status: squeue -u USERNAME
- Check job verbosity in slurm log and error files
- Check some output files generated during script execution
- Log into the machine where the job is running
  - Check GPU usage
  - Check CPU usage
  - Check RAM usage
  - Evaluate general state of machine and job

How to access to computing resources (PI)

- The access is made through public login machines, called User Interface (ui).
- Every employee or associate can access to this machines thanks to the AAI infrastructure.
- The use of the infrastructure is made with batch LSF.
- The first login should be made to setup.pi.infn.it to initialize the account.
- In order to access to medical physics exclusive resources, users should be added to fismed/arianna group.

How to use computing resources (PI)

- The User Interface machines are made to interface with the whole computing infrastructure of INFN Pisa and they cannot be used to directly compute or execute scripts.

- Once you have logged in the user interface (localui.pi.infn.it or gridui.pi.infn.it), you can submit jobs using docker.

- To use GPU you need to login to gridui.pi.infn.it

Available GPUs and how to use them (PI)

- As Medical Physics group, we have:
  1. CPU 32 core Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz
  2. 128 GB RAM
  3. 2x V100 PCIe 16GB
  4. 8x Tesla K80 8GB
- The CUDA installation on this machine follows the docker approach so that it is possible to use only CPU or CPU+GPU.
- Beside hardware, it is possible to choose the software environment:
  1. sl6 - Scientific Linux 6 senza supporto NVidia
  2. cs7 - CentOS 7 senza supporto NVidia
  3. 91_rtm_cs7 - CentOS 7 con supporto NVidia runtime ver. 9.1
  4. 91_dev_cs7 - CentOS 7 con supporto NVidia sviluppo ver. 9.1
  5. 90_tfks_cs7 - CentOS 7 con supporto NVidia runtime ver. 9.0 + Tensorflow + Keras

How to use GPU/CPU (PI)

- To submit an interactive job:
  bsub -Is -q gpuari -n 1 -R "select[defined(V100)] rusage[ngpus=1]" -a "docker-90_tfks_cs7" /bin/bash
- To submit a batch job:
  bsub -q gpuari -n 10 -R "select[defined(K80m)] rusage[ngpus=1]" -o out.out -e err.err -a "docker-90_tfks_cs7" script-to-be-executed

- As a best practice, please save error and output files.
- You can: visualize your active jobs with bjobs (-q queuename -u user), print the output (batch) with bpeek jobnumber, kill the job with bkill... as any docker job on a lsf system!

Software and virtualenv (PI)

- Machine and deep learning show a strong dependence on software packages and CUDA versions. If you want to use Tensorflow and Keras on GPUs, please note that there are tested build configurations (https://www.tensorflow.org/install/source?hl=en)

- You can create your own virtual environment with the desired packages and versions, using Miniconda, which is a free minimal installer for conda (choose the right version).

- Once you have your base environment, you can create the virtual environment you want!!

Pros and cons

- Drawbacks:
    1. Long queues (PV)
    2. Git issues (PI)
    3. CUDA interfaces in python virtual environments (PI,PV)
    4. Not usable when under maintenance (PI,PV)

- Advantages:
    1. Storage (PV,PI)
    2. Batch system (PV,PI)
    3. Sufficiently VRAM for our needs (but not for our dreams!)
    4. Dedicated hardware (PI)
    5. Desktop machines for tests (PV)

Software repository

- We can use baltig to store the code we develop.
- Baltig is equal to gitlab but it is provided by INFN.
- We can create the nextAIM group and share our code and scripts.
- Goal: have and maintain a repository with useful code for people in the collaboration.