Projects for Bioinformatics class

Final Exam rules

Students can choice between:

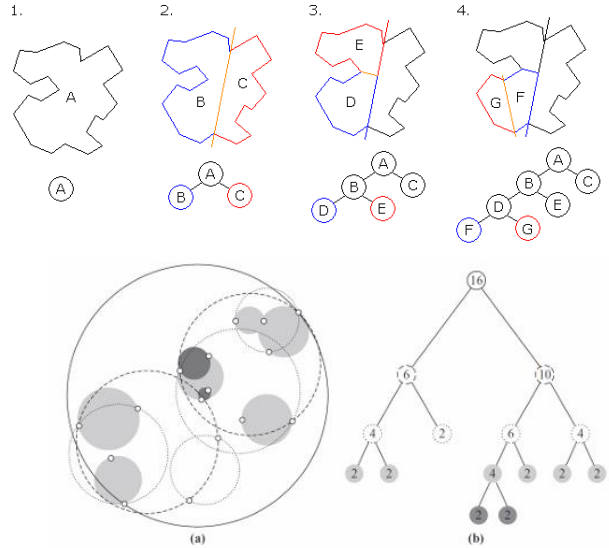
- Written test (*theory* questions + *programming* exercises)

  WARNING: the exam is passed only if **BOTH** *theory* **AND** *programming* exercises are sufficient

- Project (implementation + discussion with presentation + report + theory)
  a. Can be performed in group
  b. Oral examination will be focused only on the theory topics related to the project
  c. Students can choose the month they prefer for the oral presentation
  d. The exam is passed only if BOTH theory related to the project AND coding parts are sufficient
  e. The choice of the project must be communicated by e-mail with the names of **ALL** the members of the group. Please insert as recipients all the contact people specified in the project.
  f. There is a limit in the number of groups selecting the same project.

Multi-dimensional indexing for TGS long reads - #1

- Third-Generation Sequencing long reads enable new strategies for sequences representation. Some of those features extraction algorithms generates a fixed number of features, no matter the sequence properties.
- Multi-dimensional indexing data structures define policies for space partitioning. They allow to efficiently look for indexed data, assuming data are points in an *N*-dimensional *metric-space*.
- The project aims at analyzing the effectiveness of multi-dimensional indexing for the sake of long-reads mapping.

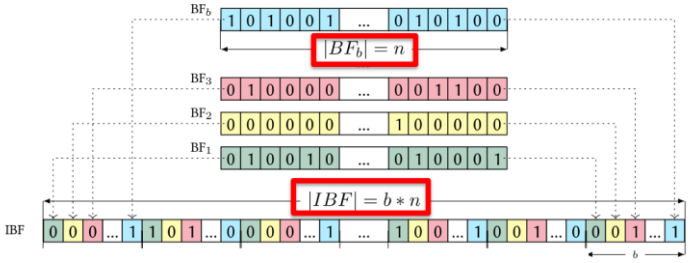


Contacts: {emanuele.parisi, gianvito.urgese, elisa.ficarra}@polito.it

Skills: Python programming

Sequences clustering with DREAM framework - #2

- The DREAM framework is an environment for partitioning genomic sequences which aims at clustering similar sequences together, enabling fast approximate search operations over large datasets.
- The project aims at testing the effectiveness of DREAM for partitioning genomes of various sizes. In particular:
  - Investigating the relationship between the genome properties and the DREAM structure hyper-parametrization.
  - Benchmarking the search procedure in terms of *false positives* and *false negatives*.

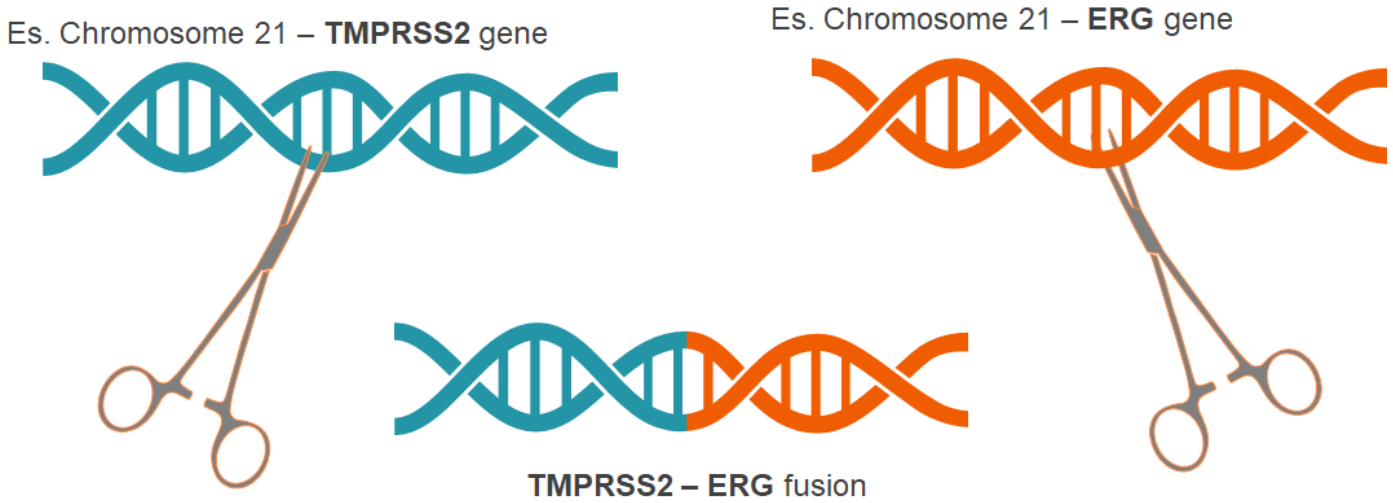


Contacts: {emanuele.parisi, gianvito.urgese, elisa.ficarra}@polito.it

Skills: Python programming

LSTM for gene fusions classification - #3

Idea: Gene fusions are the result of the juxtaposition of two non consecutive DNA regions (usually two distinct genes). This type of alteration can result as a driver of tumor processes (Oncogenic class) or generate proteins without any oncogenic effect (NotOncogenic class).



Contacts: {marta.lovino, gianvito.urgese, elisa.ficarra}@polito.it

LSTM for gene fusions classification

Provided data: list of ~ 2000 gene fusions sequences in ATCG format (nitrogenous bases)

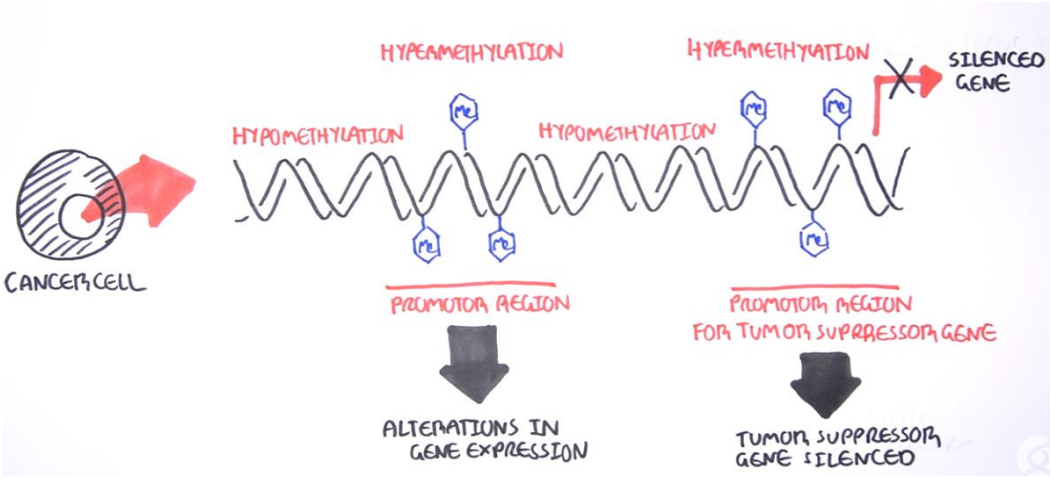
Expected work:

1. Create a new dataset translating the above mentioned sequences into protein alphabet
2. Create a Long Short Term Memory (**LSTM**) classifier able to classify gene fusions into **Oncogenic** and **NotOncogenic.** You have to build two classifiers, one with the dataset provided by us and one with the dataset you have built at step 1.
3. Implement a **bidirectional LSTM** classifier able to classify gene fusions into **Oncogenic** and **NotOncogenic.** You have to build two classifiers, one with the dataset provided by us and one with the dataset you have built at step 1.

Skills: Python 3 with **Keras** or **Pytorch** libraries, usage of Hactar cluster if the models become computationally challenging, deep learning techniques

Contacts: {marta.lovino, gianvito.urgese, elisa.ficarra}@polito.it

Methylation sites as a biomarker for cancer types - #4



DNA **methylation** is a process by which methyl groups are added to the DNA molecule (CH3 added to a base). Methylation can change the activity of a DNA segment without changing the sequence. When located in a gene promoter, DNA methylation typically acts to repress gene transcription

Contacts: {marta.lovino, gianvito.urgese, elisa.ficarra}@polito.it

- **Idea**: alterations in the methylation patterns seems to be cancer dependent.
- **Data**: tabular data with methylation sites
- **Methods**: create a classifier (with machine/deep learning methods) able to predict cancer type from methylation data

Methylation sites as a biomarker for cancer types

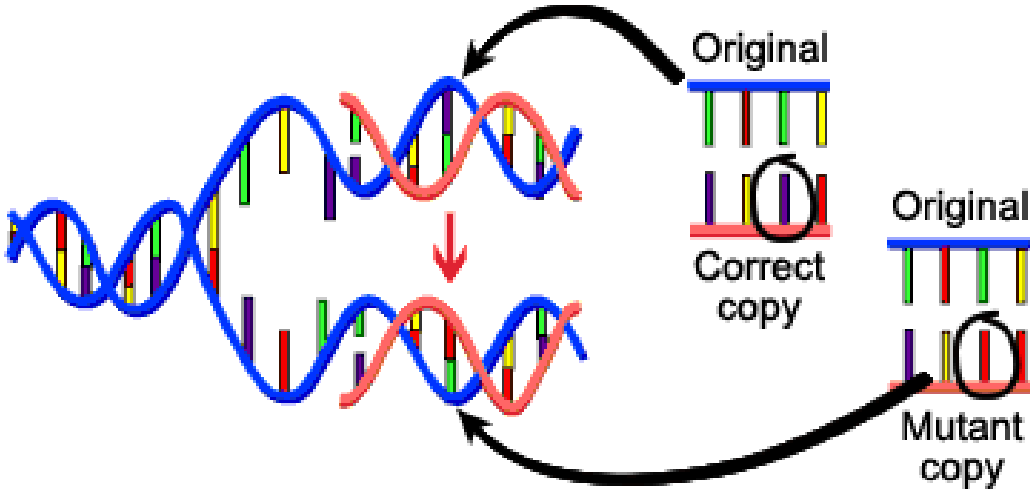
Expected work:

1. Access GDC (Genomic Data Common) to download via API methylation data of kidney and lung tumors
2. Calculate statistics about methylation profiles in these two cancer types (the distribution for each chromosome, the distribution for protein coding region, the distribution only on specific oncogenic/tumor suppressors genes)
3. Implement three machine/deep learning models at your choice to classify methylation data into kidney or lung cancer (paying attention to feature selection/dimensionality reduction, class imbalance, parameters' optimization, performance evaluation criteria, characteristics of ML model/deep learning architecture)

Skills: Python 3 with **Keras** or **Pytorch** libraries, usage of Hactar cluster if the models become computationally challenging, machine and deep learning techniques

Contacts: {marta.lovino, gianvito.urgese, elisa.ficarra}@polito.it

Mutation data as a biomarker for cancer types and cancer progression - #5



- **Idea**: alterations in genomic sequence can be involved in oncogenic processes.
- **Data**: Single Nucleotide Variation (SNV)/Copy Number Variation (CNV) from GDC
- **Methods**: create a classifier (with machine/deep learning methods) able to predict cancer type and/or cancer progression from mutation data

Contacts: {marta.lovino, gianvito.urgese, elisa.ficarra}@polito.it

Mutation data as a biomarker for cancer types and cancer progression

Expected work:

1. Access GDC (Genomic Data Common) to download via API SNV/CNV data of kidney and lung tumors
2. Calculate statistics about SNV/CNV profiles in these two cancer types (the distribution for each chromosome, the distribution for protein coding region, the distribution only on specific oncogenic/tumor suppressors genes)
3. Implement three machine/deep learning models at your choice to classify SNV/CNV data into kidney or lung cancer (paying attention to feature selection/dimensionality reduction, class imbalance, parameters' optimization, performance evaluation criteria, characteristics of ML model/deep learning architecture)

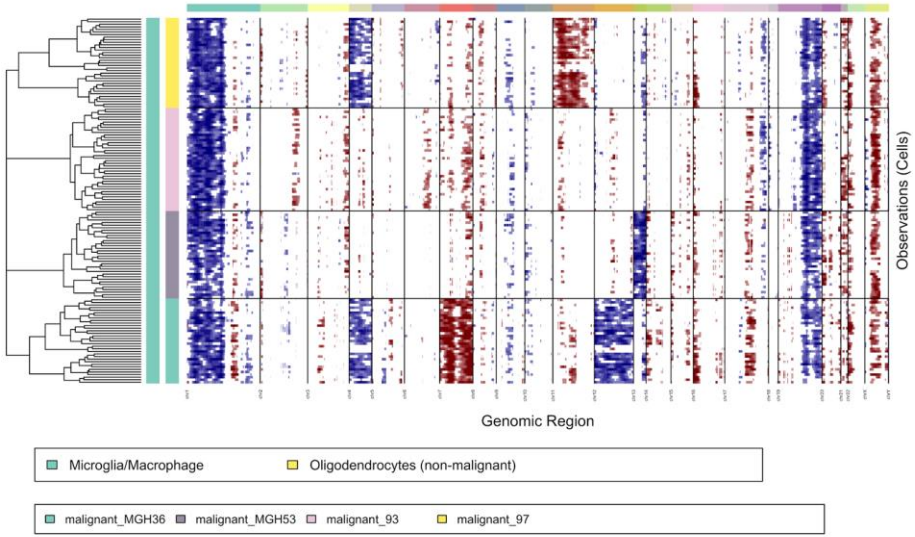
Skills: Python 3 with **Keras** or **Pytorch** libraries, usage of Hactar cluster if the models become computationally challenging, machine and deep learning techniques

Contacts: {marta.lovino, gianvito.urgese, elisa.ficarra}@polito.it

Explore relationships between CNV calls on scRNA and scDNA data - #6

Scenario:

- methods to detect CNVs on scRNA sequencing data are routinely used on human tumor samples to distinguish tumor and normal cells and identify tumor subclones
- new experimental procedures to sequence single cells DNA at large scale through microfluidics are becoming available



Goal: we would like to compare the resolution and sensibility/specificity of these two approaches, with the final aim to determine if scDNA is cost effective or not. The work will start searching and testing existing tools to obtain CNV calls on these data, then will focus on implementing sound metrics to compare the calls and could possibly end in new algorithmic ideas for single cell CNV calls.

Contacts: elena.grassi@ircc.it {marilisa.montemurro, gianvito.urgese, elisa.ficarra}@polito.it

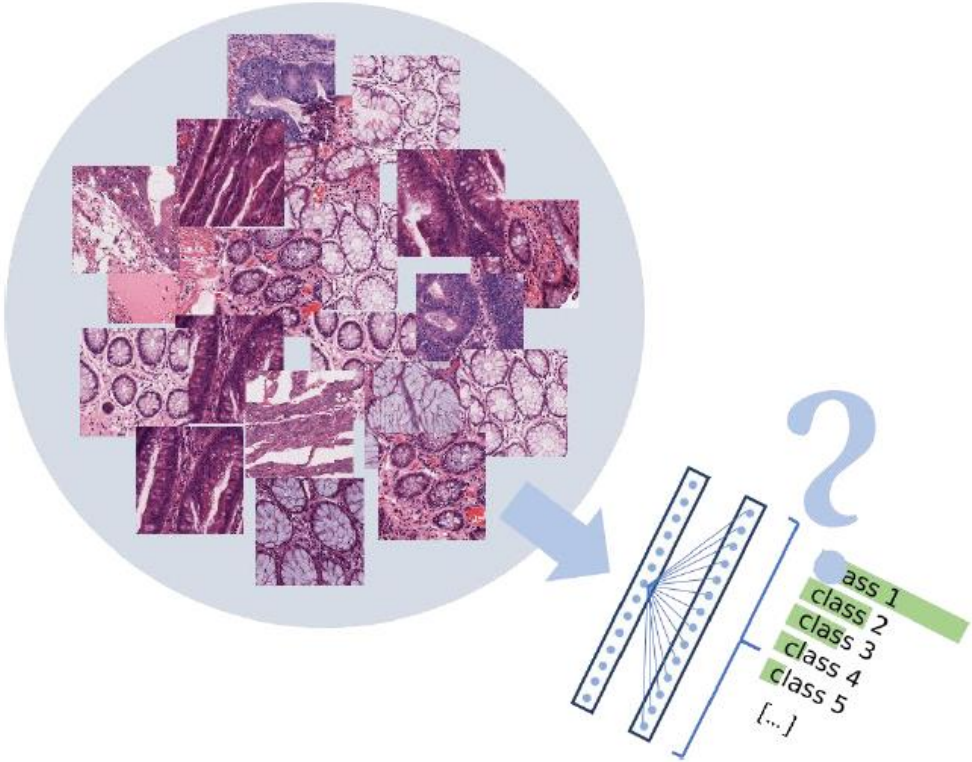
Explore relationships between CNV calls on scRNA and scDNA data

- **Pre-requisites**: proficient manuality with unix command line, basic programming skills (required), some knowledge of the biological context (optional, but suggested)
- **Tools**: bioinformatic pipeline management systems, pre-existing tools to work on scRNA and scDNA data (InferCNV, 10x cellranger-dna, SCOPE), python or R data analysis and plotting libraries
- **Data**: scRNA and scDNA public datasets (e.g. SRP114962)

Contacts: elena.grassi@ircc.it {marilisa.montemurro, gianvito.urgese, elisa.ficarra}@polito.it

CNNs projects - Uncertainty for BCNNs - #7.1

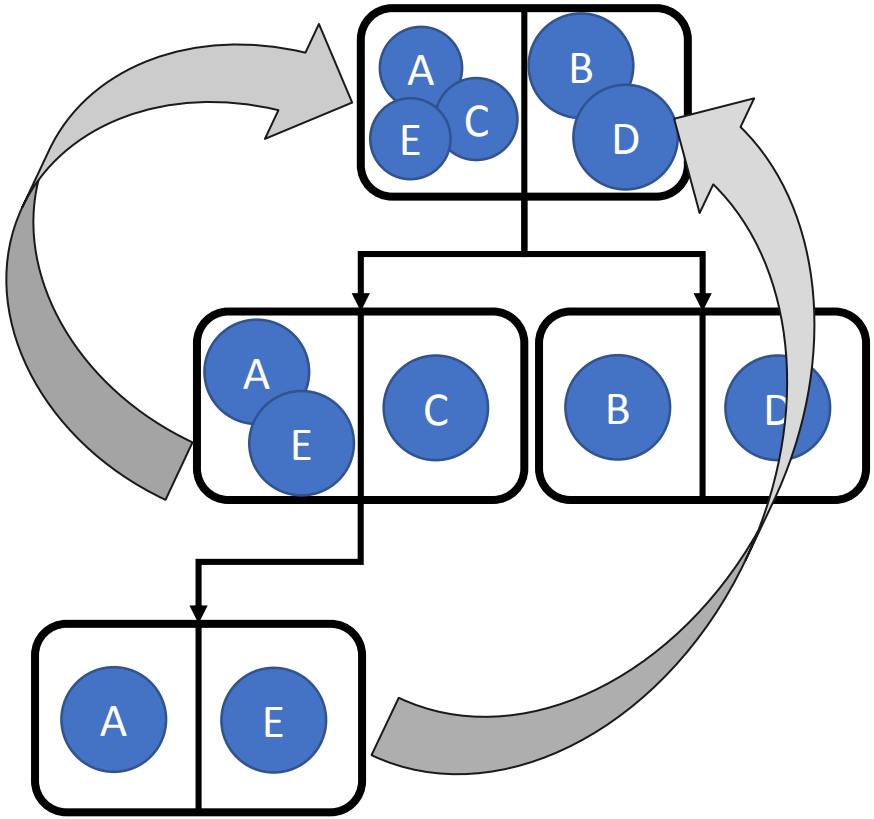
- **Aim:** analysis of uncertainty modelling in ConvNets.
- **Task:** classification and data cleaning.
- **Applications:** Everyday images (cifrar10) and pathology images.
- **Teams**: 2 people.
- **Technical requirments**:
  - Tensorflow/Tensorboard



Contacts: {francesco.ponzio, santa.dicataldo, elisa.ficarra}@polito.it

CNNs projects - Bayesian DeepTree - #7.2

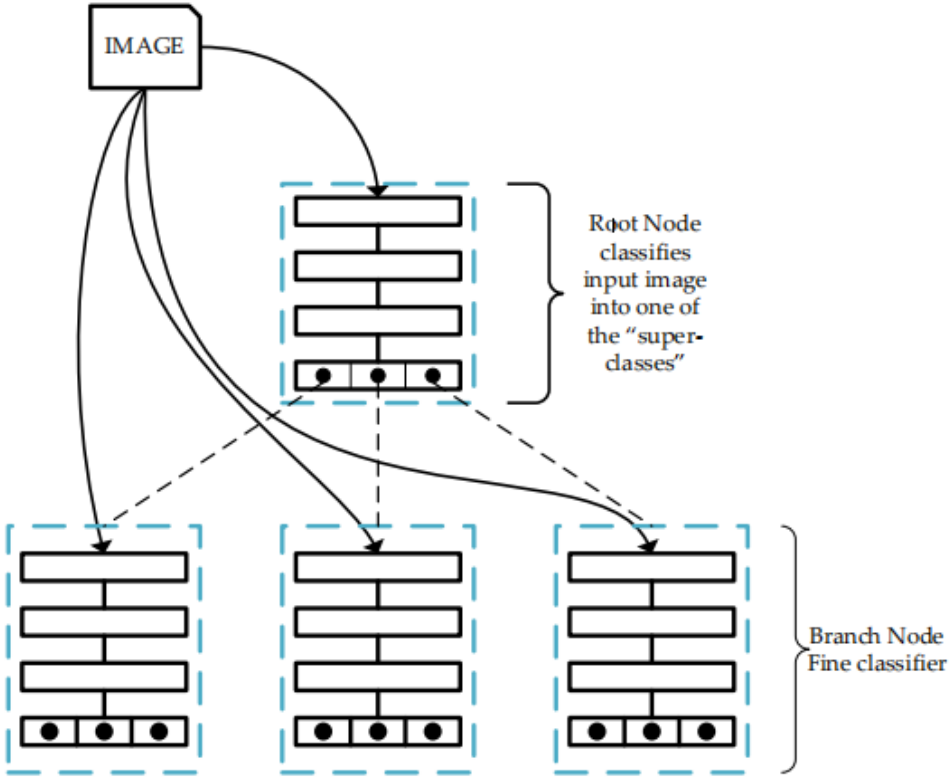
- **Aim:** incremental learning in the context on a **fixed** tree-based framework with recovery based on bayesian uncertainty.
- **Task:** classification.
- **Applications:** Everyday images (cifrar10) and pathology images.
- **Teams**: 3 people.
- **Technical requirments**:
  - Tensorflow/Tensorboard



Contacts: {francesco.ponzio, santa.dicataldo, elisa.ficarra}@polito.it

CNNs projects - Bayesian DeepTree - #7.3

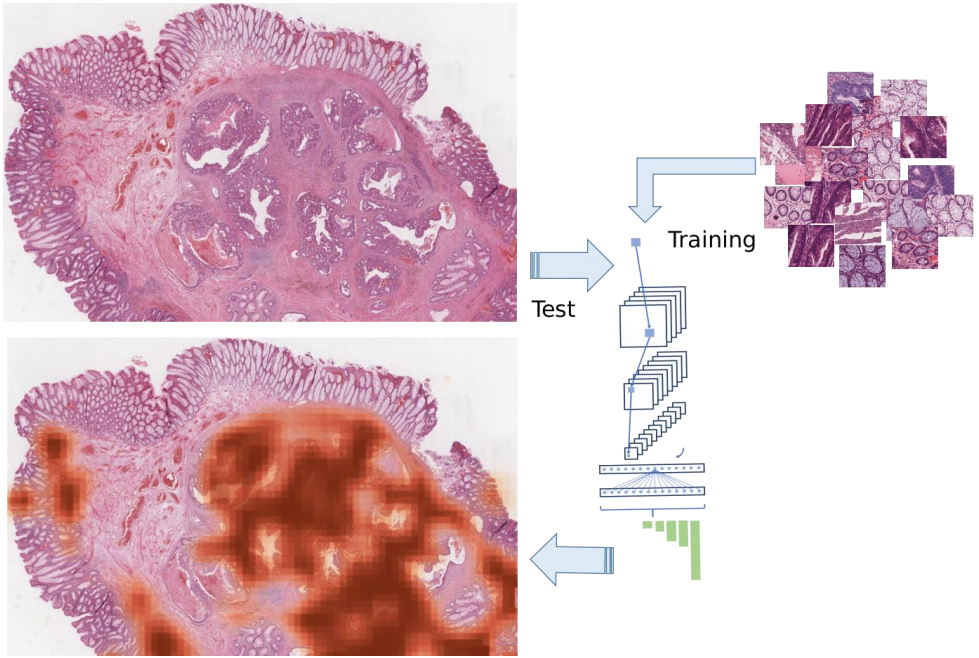
- **Aim:** incremental learning in the context on a **growing** tree-based bayesian framework.
- **Task:** classification.
- **Applications:** Everyday images (cifrar10) and pathology images.
- **Teams**: 4 people.
- **Technical requirments**:
  - Tensorflow/Tensorboard



Contacts: {francesco.ponzio, santa.dicataldo, elisa.ficarra}@polito.it

CNNs projects - Computer Aided Diagnosis Tool - #7.4

- **Aim:** investigation of uncertainty modelling in deep neural networks.
- **Task:** segmentation to obtain cancer probabilities map.
- **Applications:** pathology images.
- **Teams**: 3 people.
- **Technical requirments**:
  - GUI development
  - Tensorflow/Tensorboard



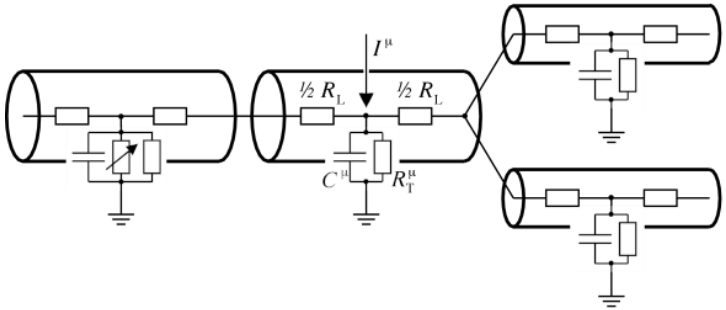
Contacts: {francesco.ponzio, santa.dicataldo, elisa.ficarra}@polito.it

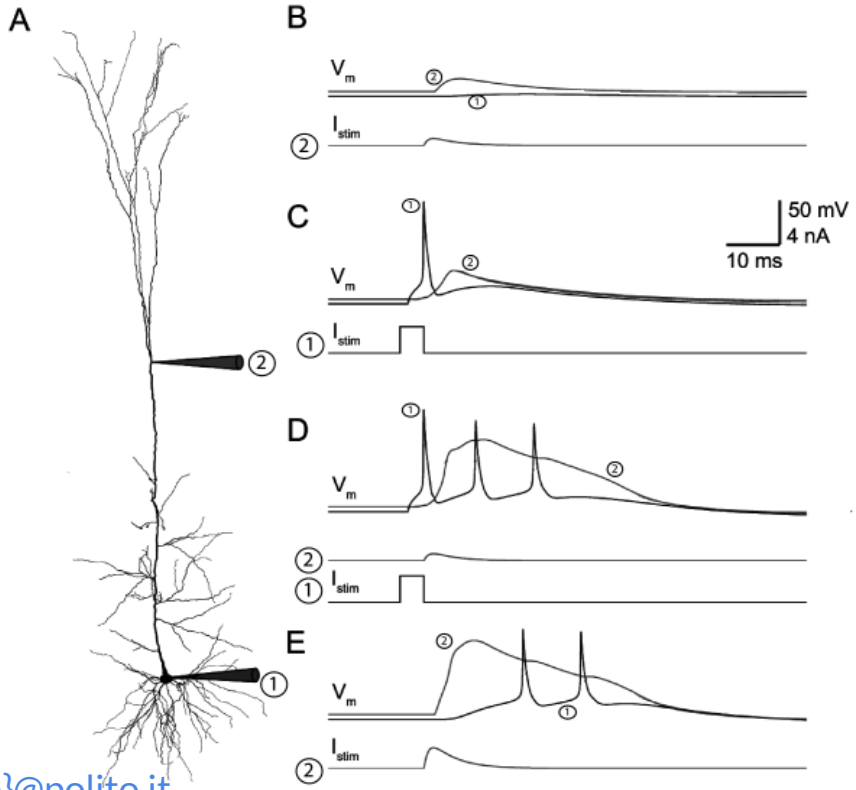
SpiNNaker - Multicompartment Neuron Model - #8

Topic: Develop of a multicompartment neuron model on a multicore *neuromorphic* hardware (SpiNNaker)

Idea: Usage of a MPI (message passing interface) programming model for implementing a modular and reconfigurable environment for Spiking Neural Network simulations

Pre-requisites: Code development in C and Python



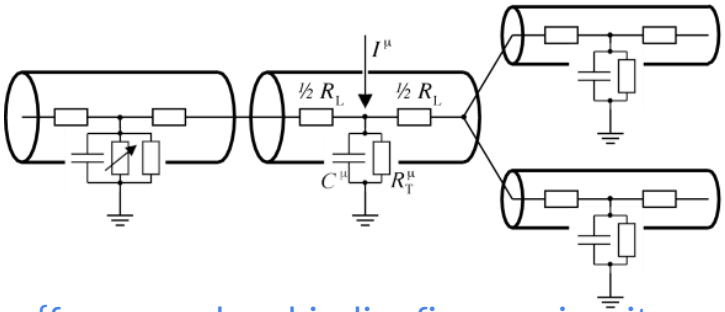


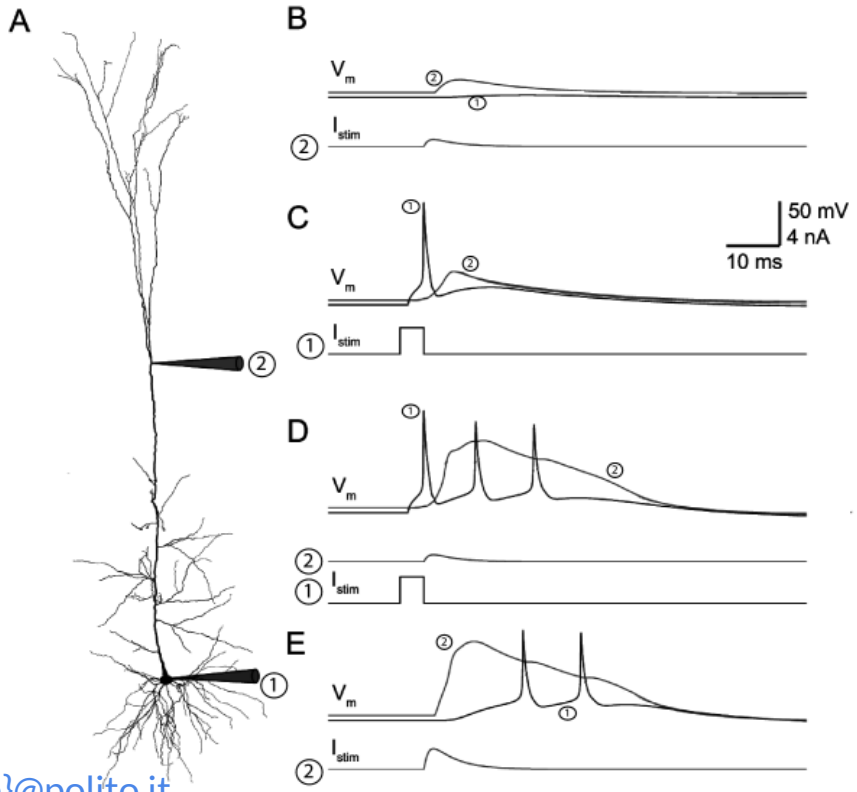
Contacts: {francesco.barchi, elisa.ficarra, gianvito.urgese}@polito.it

SpiNNaker - Multicompartment Neuron Model

Tutors: Francesco Barchi, Gianvito Urgese, Elisa Ficarra

References:
http://apt.cs.manchester.ac.uk/projects/SpiNNaker
https://neuronaldynamics.epfl.ch/online/Ch3.S4.html





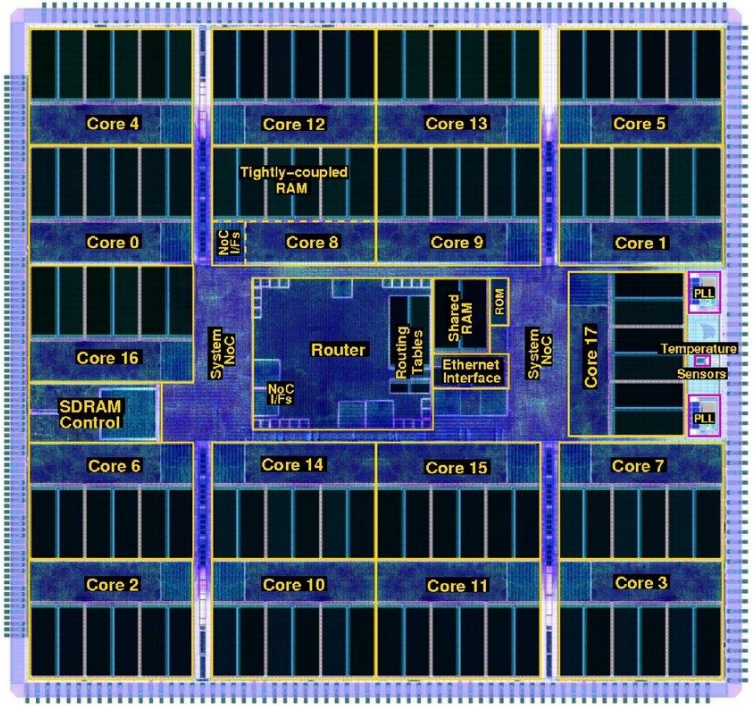
Contacts: {francesco.barchi, elisa.ficarra, gianvito.urgese}@polito.it

SpiNNaker - Communication Middleware - #9

Topic: Develop and profile a communication middleware able to exploit the SpiNNaker architecture multicast connectivity

Idea: The multicore and neuromorphic SpiNNaker architecture has a very efficient connectivity, it make use of ad-hoc routers to spread packets in a multicast way. The student will be involved in the developing of a communication middleware exploiting the SpiNNaker routers. The middleware will expose a low level API in order to implement Point to Point, Broadcast, Multicast and Synchronization feature.

Pre-requisites: Code development in C and Python.

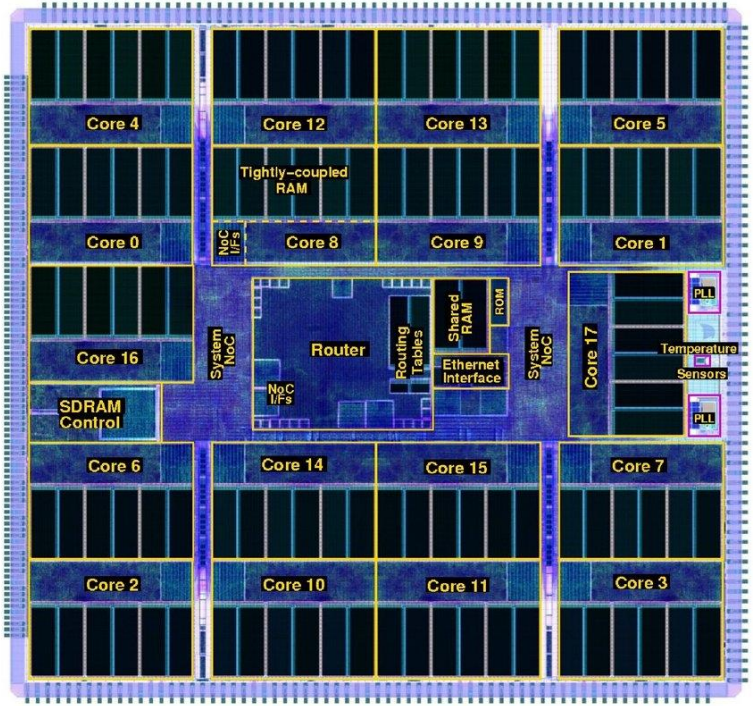


Contacts: {francesco.barchi, elisa.ficarra, gianvito.urgese}@polito.it

SpiNNaker - Communication Middleware

Tutors: Francesco Barchi, Gianvito Urgese, Elisa Ficarra

References:
http://apt.cs.manchester.ac.uk/projects/SpiNNaker
https://ieeexplore.ieee.org/abstract/document/8052322



Contacts: {francesco.barchi, elisa.ficarra, gianvito.urgese}@polito.it