Bioinformatics projects

Politecnico di Torino
Bioinformatics
2017/18

POLITECNICO DI TORINO

Project #1

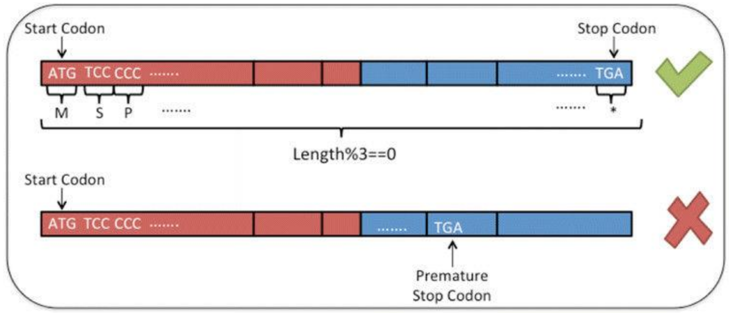
Post-processing of Gene Fusion detection tools results

Introduction

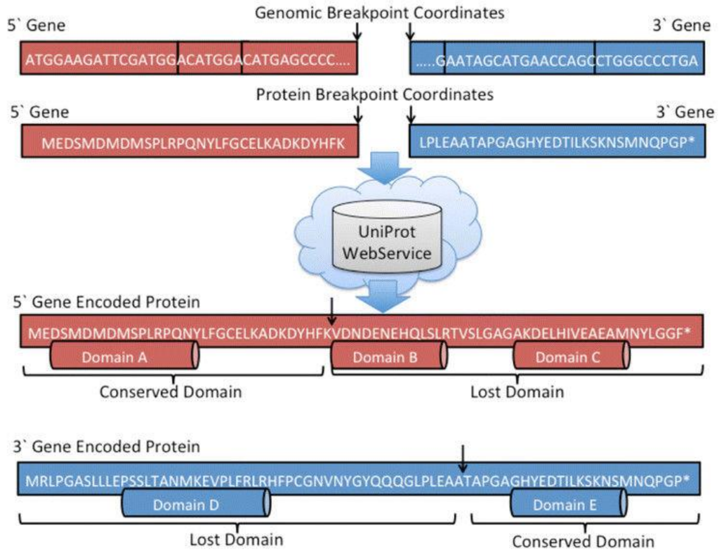
Gene fusions are the result of genetic aberrations (translocations, deletions, amplifications and inversions) involving the **juxtaposition of two genes** that can generate a single hybrid transcript. Since 1960, gene fusions have been known to play a major role in **tumorgenesis**.

However, the heterogeneity of filtering strategies implemented in gene fusion tools often yields **poorly overlapping sets** of candidate transcripts **between algorithms.**

In addition, very often the number of fusion candidates is too large to experimentally validate all putative fusion transcripts.



4(a)


Genomic Breakpoint Coordinates
5` Gene
3` Gene
ATGGAAGATTCGATGGACATGGACATGAGCCCC....
.....GAATAGCATGAACCAGCCTGGGCCCTGA
Protein Breakpoint Coordinates
5` Gene
3` Gene
MEDSMDMDMSPLRPQNYLFGCELKADKDYHFK
LPLEAATAPGAGHYEDTILKSKNSMNQPGP*
UniProt WebService
5` Gene Encoded Protein
MEDSMDMDMSPLRPQNYLFGCELKADKDYHFKVDNDENEHQLSLRTVSLGAGAKDELHIVEAEAMNYLGGF*
Domain A
Domain B
Domain C
Conserved Domain
Lost Domain
3` Gene Encoded Protein
MRLPGASLLLEPSSLTANMKEVPLFRLRHFPCGNVNYGYQQQGLPLEAATAPGAGHYEDTILKSKNSMNQPGP*
Domain D
Domain E
Lost Domain
Conserved Domain

4(b)

Gene fusion detection tools analysis and implementation of a new prioritization approach

Input: .fasta/.fastq file of a cancerogenic sample downloaded from SRA database https://www.ncbi.nlm.nih.gov/sra

Method:

1. Review of differences between gene fusion tools
2. Run at least three gene fusion tools to identify fusion candidates (e.g. deFuse, ChimeraScan, mapsplice, fusionmap, etc.)
3. Convert output file formats to be correct input to tools in point 4.
4. Run Pegasus and OncoFuse tools
5. Obtain protein sequences form gene fusions in point 2.
6. Define and implement a different classification approach for ranking gene fusions exploiting their protein sequence
7. Compare results on points 4 and 6

Output: complete pipeline + detailed report

References: Abate F.. et al, ***Pegasus****: a comprehensive annotation and prediction tool for detection of driver gene fusions in cancer*, BMC Systems Biology 2014, 8:97 Shugay M et al, ***Oncofuse****: a computational framework for the prediction of the oncogenic potential of gene fusions,* Bioinformatics. 2013 Oct 15;29(20):2539-46.

Project #2

Gene fusion structure retrievement and analysis

Note: Project not available for ICT students

Analysis Workflow (1)

- Input Files:
- - ChimeraScan output file (will be provided soon)
- - hg19 sequence (in fasta format)

- Output:

- **Consensus fusion sequence and**
- **information about the portion of the partner genes retained in the fusion**.

- Output example:
- gene5p: <gene name> gene3p: <gene name> chr5p: <chromosome number> chr3p: <chromosome number> BP5p: <breakpoint coordinate> BP3p: <breakpoint coordinate> Fusion Sequence: AATCCCGTTAAACGTATGG(etc.) Fusion structure: < Promoter-

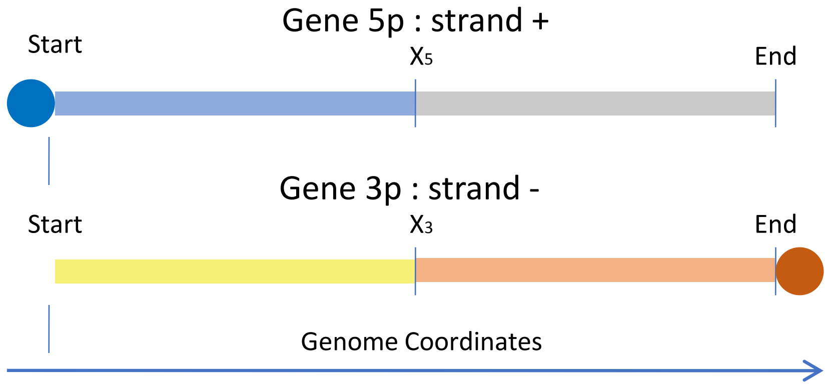
● = Promoter

X_5 , X_3 = breakpoint's coordinate

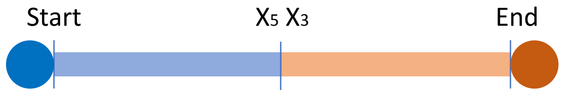
Remember: genomic coordinates grow from left to right, and

Strand +/- = transcription direction forward/reverse

The fusion structure is biological meaningful according to the direction of the transcription and the type of fusion mechanism (as seen in gene fusion lectures)

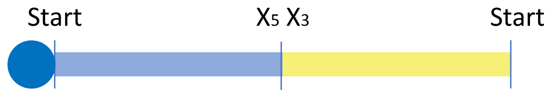


Promoter-Promoter



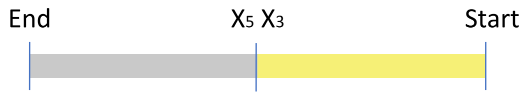
Mechanism 1

Promoter-End



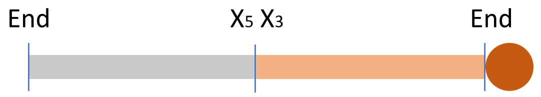
Mechanism 2

End-End



Mechanism 3

End-Promoter



Mechanism 4

Analysis Workflow (2)

Isolate in ChimerScan output file those fusions supported by reads (spanning_frags!=0) and extract the following partner genes information: chrom5p, chrom3p, start5p, start3p,end5p, end3p, strand5p, strand3p, breakpoint_spanning reads

For each fusion, use the spanning reads in order to reconstruct the longest consensus region. Remember that the breakpoint position is not indicated on the reads, so you have to implement an ad hoc solution to find regions of similarity among reads.

For each fusion reconstruct all the virtual references described by the data provided in ChimeraScan output file. Remember that you do not know which between start5p and end5p is the breakpoint on the first partner gene and which between start3p and end3p is the breakpoint on the second partner gene.

Remap the consensus sequence on the virtual references to detect which portion of the genes have been retained in the fusion

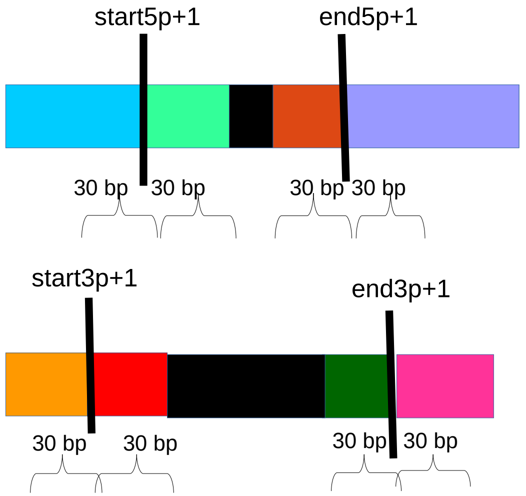
Consensus Region

```
GCGGAGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGC
CGGAGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCC
CGGAGGCGGAGGGCGAGGGGCGGGGAGAGCCGCCTGGAGCGCGGCAGGAAGCC
CGGCGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCCGGAAGCC
GGAGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCT
GGAGGCGGAGGGCGAGGGGCGGGGAGAGCCGCCTGGAGCGCGGCAGGAAGCCT
GGTGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCT
GGAGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCATGAAGCCT
GGAGGCGGAGGGCGAGGGGCGGCGAGCGCCGCCTGGAGCGCGGCAGGGAGCCT
GAGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTT
AGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTA
GCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATC
CGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCA
GGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAG
AGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAGTT
GGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAGGTC
GCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAGTTGTG
GGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAGTTGTGAGTGA
GGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAGTTGTGAGTGAG
```

```
GCGGAGGCGGAGGGCGAGGGGCGGGGAGCGCCGCCTGGAGCGCGGCAGGAAGCCTTATCAGTTGTGAGTGAG
```

You have to select, in case of discordance among reads, the most supported base

Virtual References



gene5p

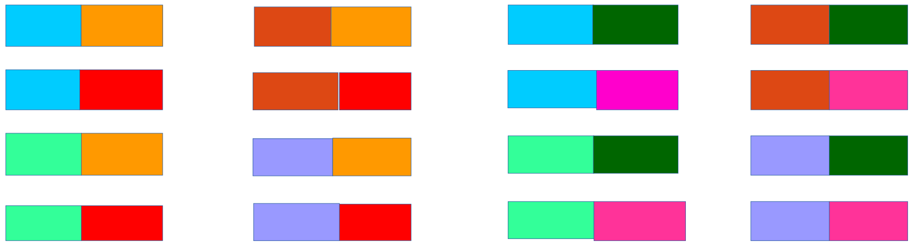
Depending on the STRAND the different portions of the gene are named Promoter or End

gene3p

Using Chimerascan tool, there is no information about the BP position (i.e. it can be the start or the end point)

Consequently, there are 16 combinations instead of 4 only

<u>VIRTUAL REFERENCES:</u>



Virtual reference (2)

- **NOTE**: when you construct the virtual reference for all the possible combinations consider that the BPs of the genes must be faced.
- Thus, consider the possibility to reverse (and complement) the original DNA sequence of one gene, or both the genes.

Open questions….

- After the alignment on the virtual references, check if the regions between start and end points are conserved in the fusions.
- Can you avoid in principle some of the 16 combinations?

Project #3

micro-RNA search on viral genomes

microRNAs

micro-RNAs, or **miRNA**s, are short sequences involved in post-transcriptional **regulatory processes**.

See slides on alignment for miRNA & isomiR detection.pdf

In certain biological contexts can be interesting to detecting the presence of known miRNA sequences in viral genomes, to better understand the interactions between a **virus** and its host.

miRNA sequences already identified and annotated for different organisms can be downloaded from available **miRNA databases**.

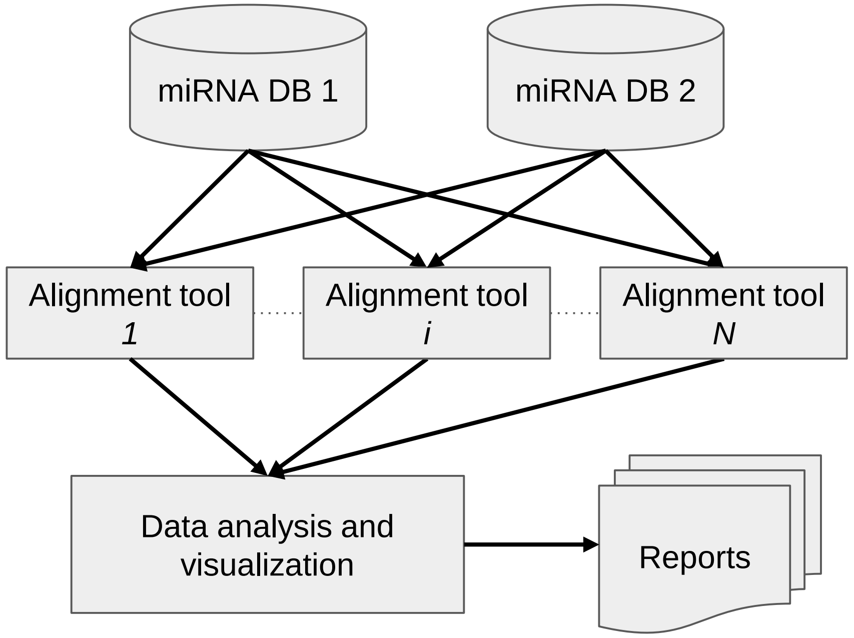
Micro-RNA search on viral genomes

The main tasks the project focuses on are:

- Download miRNA sequence from miRBase and MirGeneDB

- Perform alignment over all the available viral genomes (given input to the project) using standard NGS alignment tools such as Bowtie, BWA, Yara, Mega-Blast.

- Report about identified virus, positions for the alignmente, expression



References
miRBase: https://academic.oup.com/nar/article/42/D1/D68/1057911
MirGeneDB: https://www.biorxiv.org/content/early/2018/02/05/258749?rss=1
Bowtie2: https://www.ncbi.nlm.nih.gov/pubmed/22388286
Yara: http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000099827

Project #4

Web interface for bioinformatics tools

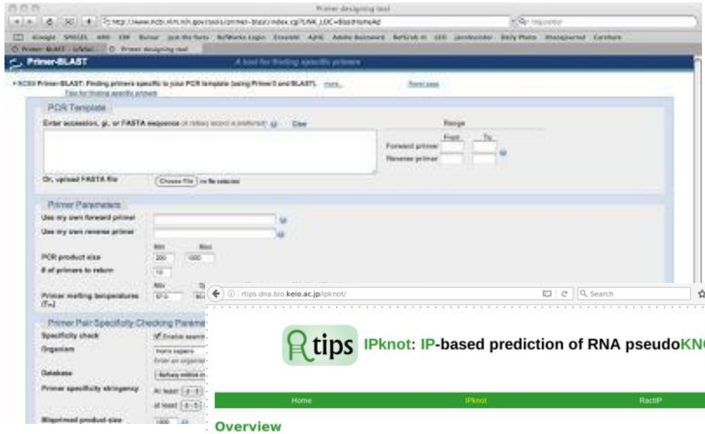
WEB available bioinformatics tools

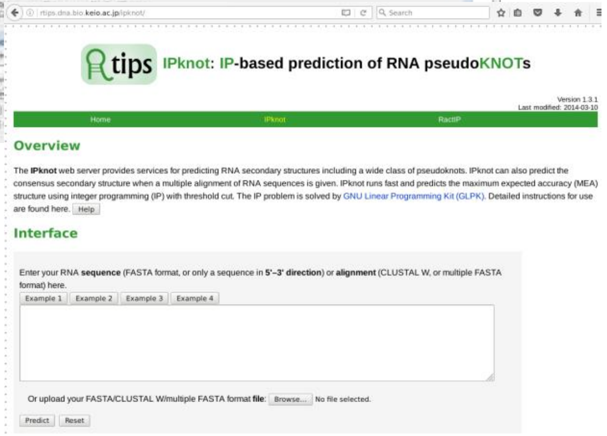
Bioinformatics tools are very often released as stand-alone command line applications; however, some of the most known and utilized tools, like *BLAST*, have a **WEB interface**, for making their utilization easier. A further example is *IPknot*, for RNA secondary structure analysis.

Some analysis pipeline like *QuickRNASeq* not only make analysis tool accessible from the WEB, but also allows for **online result visualization**.

The project aims at producing a working pipeline for the analysis and visualization of miRNA data, using an already existing tool called *isomiR-SEA*, making it web accessible, similarly to the examples cited above.

References
BLASTn: https://blast.ncbi.nlm.nih.gov/Blast.cgi
IPknot: http://rtips.dna.bio.keio.ac.jp/ipknot/
QuickRNASeq: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-015-2356-9
IsomiR-SEA: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-0958-0





IPknot: IP-based prediction of RNA pseudoKNOTs

Version 1.3.1
Last modified: 2014-03-10

Overview

The **IPknot** web server provides services for predicting RNA secondary structures including a wide class of pseudoknots. IPknot can also predict the consensus secondary structure when a multiple alignment of RNA sequences is given. IPknot runs fast and predicts the maximum expected accuracy (MEA) structure using integer programming (IP) with threshold cut. The IP problem is solved by GNU Linear Programming Kit (GLPK). Detailed instructions for use are found here.

Interface

Enter your RNA **sequence** (FASTA format, or only a sequence in **5'–3' direction**) or **alignment** (CLUSTAL W, or multiple FASTA format) here.

Or upload your FASTA/CLUSTAL W/multiple FASTA format **file**: No file selected.

POLITECNICO DI TORINO

Project(s) #5

Sequences features extraction strategies on TGS data

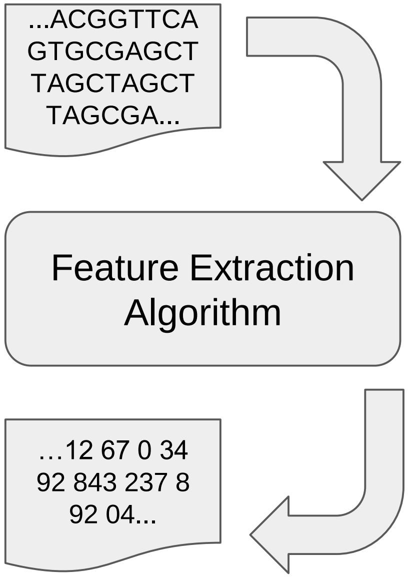
Sequences features extraction

Third-Generation Sequencing technologies enabled the development of different strategies for sequence **alignment**, relying on features extraction algorithm giving sequences a different concise representation, and defining different distance functions over such representations (see TGS alignment slides).

A set of projects are available for proposing non-investigated, or novel, approaches for performing features extraction on biological sequences. Some of the tasks available are:

- Choosing and test **features extraction algorithms**, defining distance measures for deciding how different two sequences are, looking at their features list.
- Explore how and if existing algorithms can be vectorized exploiting **SIMD** instruction set extensions (i.e. Single Instruction Multiple Data).
- Define **indexing strategies** for different features representations.

References
Winnowing: https://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf
COSINE: https://academic.oup.com/nar/article/45/14/e132/3861609


...ACGGTTCA GTGCGAGCT TAGCTAGCT TAGCGA...
Feature Extraction Algorithm
…12 67 0 34 92 843 237 8 92 04...

Project #6

Classification of gene expression data

Introduction to the problem

Impact of classes imbalances on classification performance

Gene expression and Machine Learning

- For concepts related to gene expression see slides in *Gene expression analysis in a glance.pdf* file
- For concepts related to Machine Learning please follow appropriate lectures in the second part of the Bioinformatics course and refer to corresponding materials.

Note: binary classification is affected by classes imbalance

- Unbalanced classes size can affect performance of downstream classifiers on the **testing dataset**
  - Classifiers tend to be biased towards majority class
- Pre-processing of **training dataset** is needed

Note (continue): How to reduce imbalance?

- Two main approaches:
    - Under-sampling of majority class (e.g. random subsampling)
    - Over-sampling of minority class (e.g. synthetic samples generation)
- A mixed approach (over-, then under-sampling) may be the best solution in most cases

Cancer subtypes classification from gene expression data

Cancer is composed of multiple subtypes with distinct morphologies and clinical implication (e.g. response to therapy and overall survival).

Input: gene expression data of breast cancer subtypes downloaded from GDC database
https://gdc.cancer.gov/

Method:

1. Handling the problem of classes imbalance (where the classes are the cancer subtypes)
2. Handling the problem of the huge number of features (genes) for each sample (see feature selection/extraction)
3. Choose, then train and test some supervised machine learning techniques to predict breast cancer subtypes
4. Choose and then apply some unsupervised machine learning techniques to cluster breast cancer samples

Output: complete pipelines (one for classification, one for clustering) + detailed report

ProjectS #7

Projects on Convolutional Neural Networks (CNNs) – deep learning

POLITECNICO DI TORINO

Introduction to the problem

Task

- Task of assigning a label from a fixed set of categories to an input image
- It seems quite trivial … but computer points of view is rather different
- Our task is to turn this quarter of a million numbers into a single label, such as *“cat”*

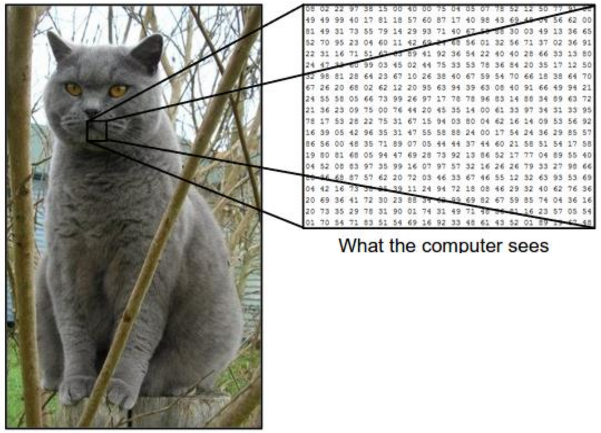
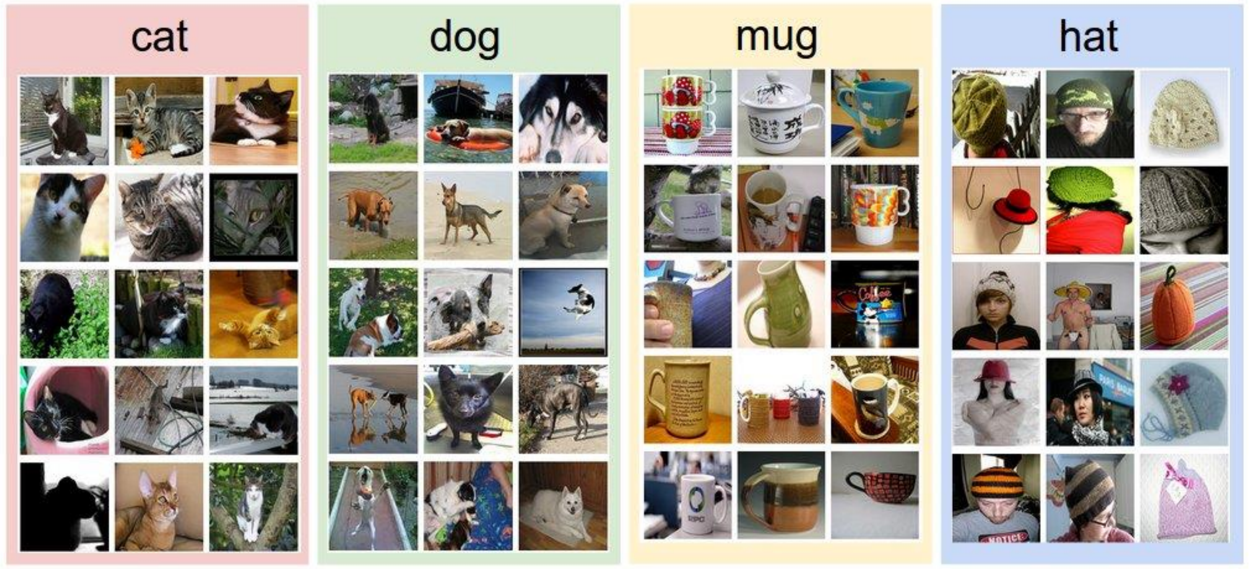


Image classification

Data driven approach

- Algorithms that look at ***many examples*** and learn to classify a given image to a certain class



Challenges

- Challenges involved from the perspective of a Computer Vision algorithm

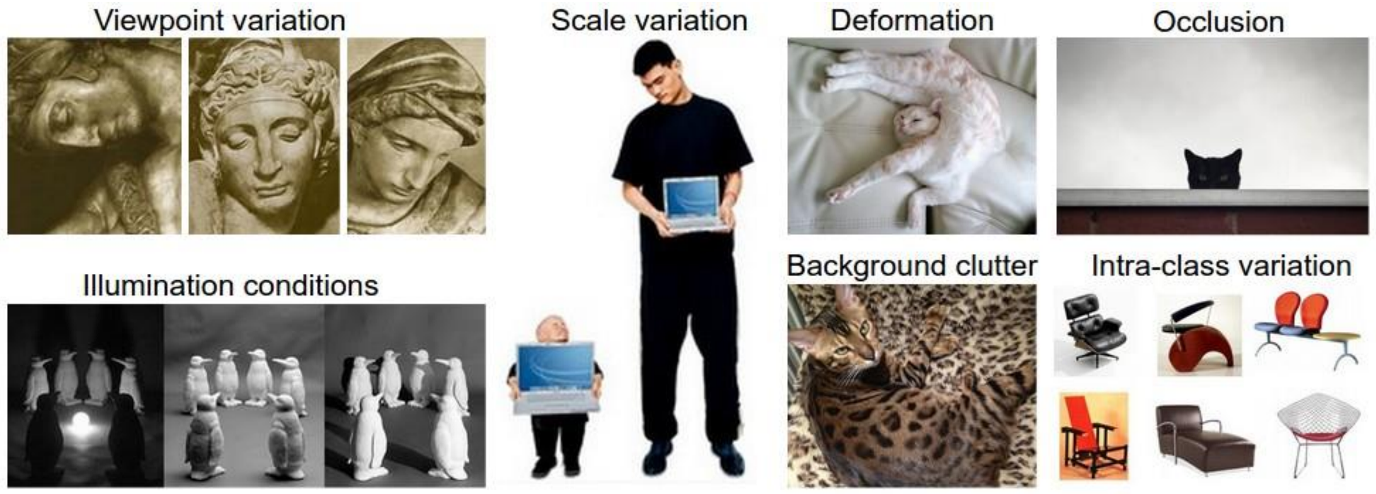


Image classification

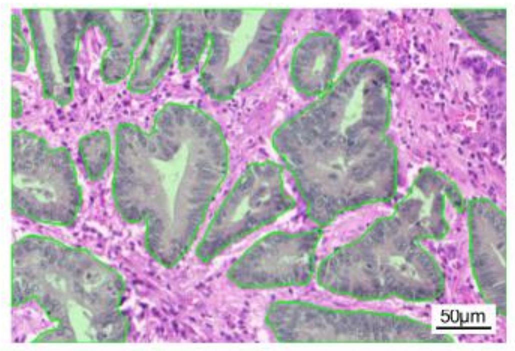
What we need (basically) ?

1. Training set: *N* images, each labeled with one of *K* different classes
2. Test set: ground truth to evaluate the quality of the classifier (i.e. labelled different images set)
3. Classification algorithm

Image classification

Method (typical)

- The chosen machine learning algorithm is fed with a certain number of features extracted from the images.
- Suppose we want to classify some colonic **histological images** in **cancerous** or **not cancerous**.



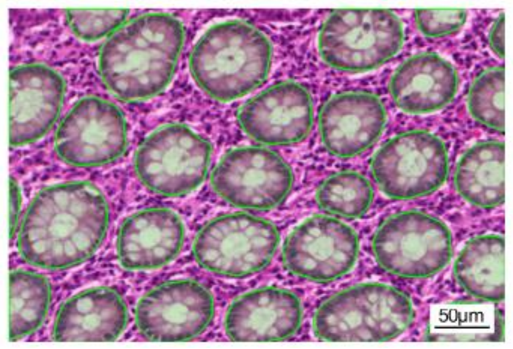
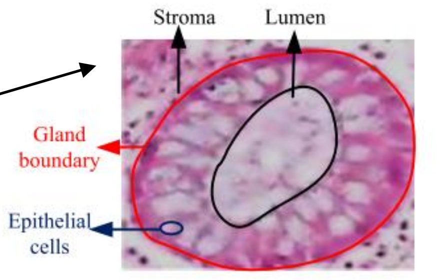
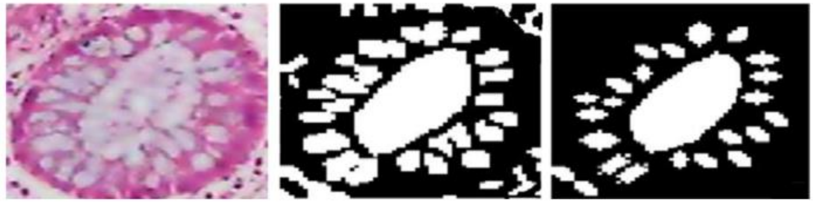


Image classification
Method (typical)

- We must:
  i. Segment colonic glands
  ii. Extract some **features** (morphological, textural, …) **to describe the object**





  iii. Eventually reduce redundancy of information
  iv. Feed the machine learning algorithm with such extracted and selected features
  v. Evaluate performance on new images (test set)

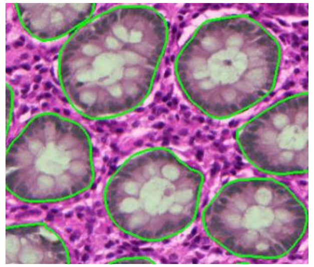


Image classification

Method (Convolutional Neural Networks)

- Delegate features extraction to a Convolutional Neural Networks (CNNs)

i. **What is a CNNs and, more in general, deep learning?**
  - please follow appropriate lectures in the last part of the Bioinformatics course and refer to corresponding materials.
  - *LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.*

ii. **How it works?**
  - *http://cs231n.github.io/ is a very well-done tutorial to understand CNNs (Module 2, at the end of the web-page)*
  - *http://neuralnetworksanddeeplearning.com/*

iii. **How can I use CNNs as features extractors and why?**
  - *Donahue, Jeff, et al. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition." Icml. Vol. 32. 2014.*
  - *Tajbakhsh, Nima, et al. "Convolutional neural networks for medical image analysis: full training or fine tuning?." IEEE transactions on medical imaging 35.5 (2016): 1299-1312.*

Application: **digital pathology**

- Management of information generated from a digital slide coming from a tissue sample

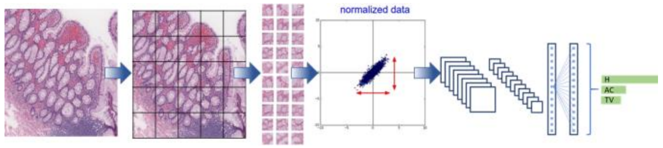
- Tools that can provide a faster, cheaper and more accurate **diagnoses**

CNNs projects - #1 Whole slides attention maps

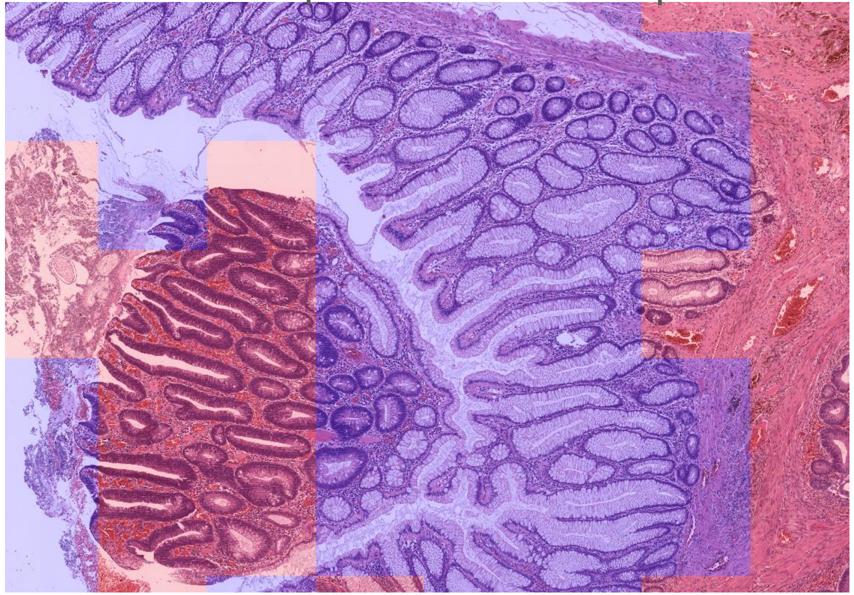
Aim: Implement a software able to produce a tiled multi-resolution image depicting attention maps for cancer detection.

Method: CNN training from scratch used for inference on samples from test set in order to create attention map.

Input data: Whole histological slides are provided. First step of the pipeline must be dataset generation via WSIs cropping.



Example of attention map



CNNs projects - #2 CNNs: insight in classification

Aim: This project aims at comparing CNNs full training versus transfer learning (both features extraction and fine tuning) at increasing numerosity of the training set. The outline is image classification.

Method: Three strategies of machine learning must be implemented and compared. i) CNNs trained from scratch; ii)CNNs fine tuned from different layers; ii)CNNs used as features extractor for a traditional ML method. Result must be analyzed at different numerosity of the training set.

Input data: Whole histological slides are provided. First step of the pipeline must be dataset generation via WSIs cropping.

CNNs projects - #3 Data augmentation for CNNs

Aim: The focus of this project is evaluating CNNs full training and fine tuning with an increasing numerosity of the dataset through the so-called data augmentation. Different kinds of data augmentation must be analyzed

Method: Two strategies of machine learning must be implemented and compared. i) CNNs trained from scratch; ii)CNNs fine tuned from different layers. You must change the training set size via data augmentation and evaluate the impact of different strategies of data augmentation.

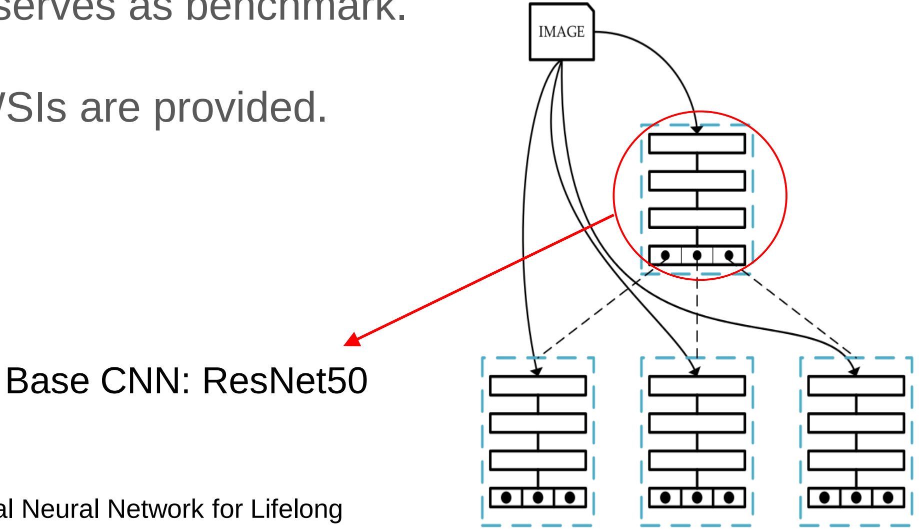
Input data: Whole histological slides are provided. First step of the pipeline must be dataset generation via WSIs cropping.

CNNs projects - #4 Tree-CNN

Aim: Develop a framework for CNNs continuous learning and compare with fine-tuning model. Here the interest is preserve the knowledge obtained in previous learning procedure.

Method: Develop a deep CNNs tree which grows with respect to new classes as input. The comparison with a fine tuning model serves as benchmark.

Input data: Patches obtained from WSIs are provided.



D. Roy, P. Panda, K. Roy, “Tree-CNN: A Deep Convolutional Neural Network for Lifelong Learning” https://arxiv.org/abs/1802.05800

POLITECNICO DI TORINO

Project #8

Image classification using Spiking Neural Networks (SNNs)

Image classification using Spiking Neural Networks

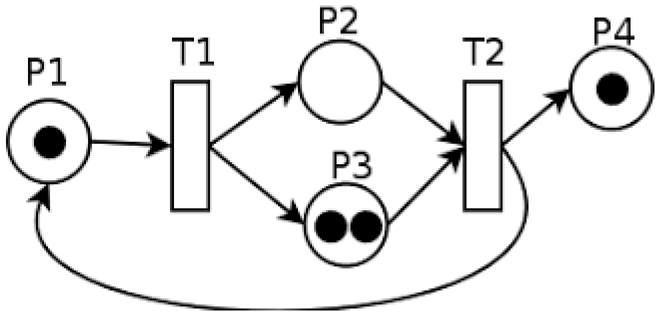
https://link.springer.com/article/10.1186/s13640-015-0059-4

POLITECNICO DI TORINO

Project #9

Simulation of biological networks on the multicore parallel neuromorphic architecture named SpiNNaker

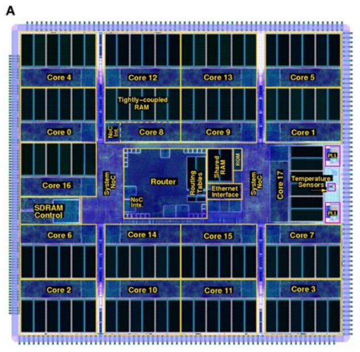
Simulation of biological networks with Petri Nets on Multicore Platforms



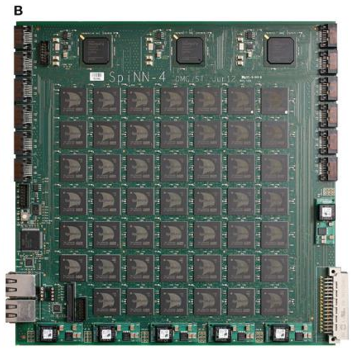
Petri Nets (PN) offer a framework to analyse the dynamical properties of concurrent systems, from either a qualitative or a quantitative point of view (see [1] for an introduction to PNs). Indeed, PNs have already been applied to various types of biological networks (see [2] for a review), such as gene regulatory and cell-to-cell communication networks.

Petri Nets simulation is thus used for simulation of complex dynamic systems, however, simulation time can prevent their scalability to meaningful biological studies. For this reason, the exploration of alternative HW architectures to speed-up PN simulation is an intriguing research problem.

The purpose of this project is to explore the performance of Petri Nets simulation on a multicore parallel architecture named **SpiNNaker** (http://apt.cs.manchester.ac.uk/projects/SpiNNaker/). SpiNNaker has been designed to simulate biological neural networks, which from a computational model perspective present characheristics similar to PN.
The goal of the project is to implement a (simple) Petri Net simulation model on SpiNNaker and report issues, limitations and relevant performance figures.


A

The SpiNNaker chip


B

A 48-node SpiNNaker board (SpiNN-4)

Requirements: **Python** and **C**

References: [1] H. Alla, R. David Continuous and hybrid Petri nets JCSC, 8 (1) (1998), pp. 159-188
[2] S. Hardy, P.N. Robillard Modeling and simulation of molecular biology systems using Petri nets: Modeling goals of various approaches J. Bioinform. Comput. Biol., 2 (4) (2004), pp. 595-613