



Consiglio Nazionale  
delle Ricerche



# TRAINING COURSE IN Computational Methods for Epitranscriptomics

Bari, 11th-13th September 2024



**Title:** Detection of rare C-to U RNA editing events: C-to-U RNA editing unveiled by Nanopore Direct RNA sequencing

**Date:** Thursday 12th September – 15.30-16.30 (day 2) Bari

**Instructor:** Adriano Fonzino, Ph.D.



## Detection of rare C-to U RNA editing events: C-to-U RNA editing unveiled by Nanopore Direct RNA sequencing

- Brief introduction to the **biological topic**;
- Technical aspects related to **Oxford Nanopore Technology** (ONT) third generation sequencing platform;

- Main Topics:

General overview of **available strategies** for **epitranscriptomics modifications** detection via **ONT**.

**Case Study:** C-to-U editing signal amelioration in ONT direct-RNA sequencing assays (*Fonzino et al., 2024*).

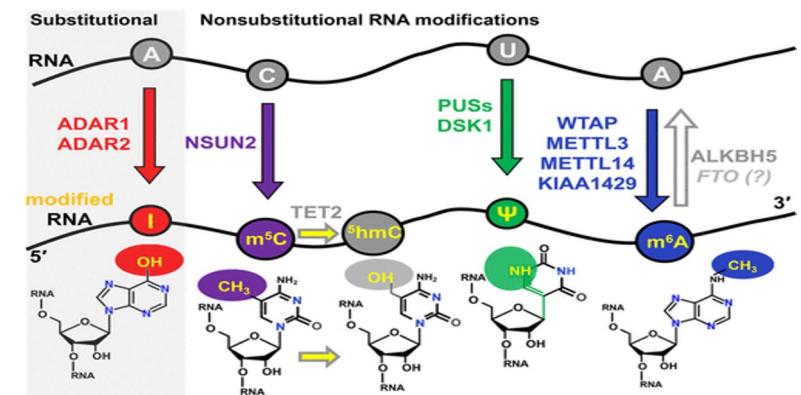
**Practical Session** with CtoUclassifier package for the denoising of C-to-U signal in **ONT**.

# Complexity of Transcriptomes and RNA modifications



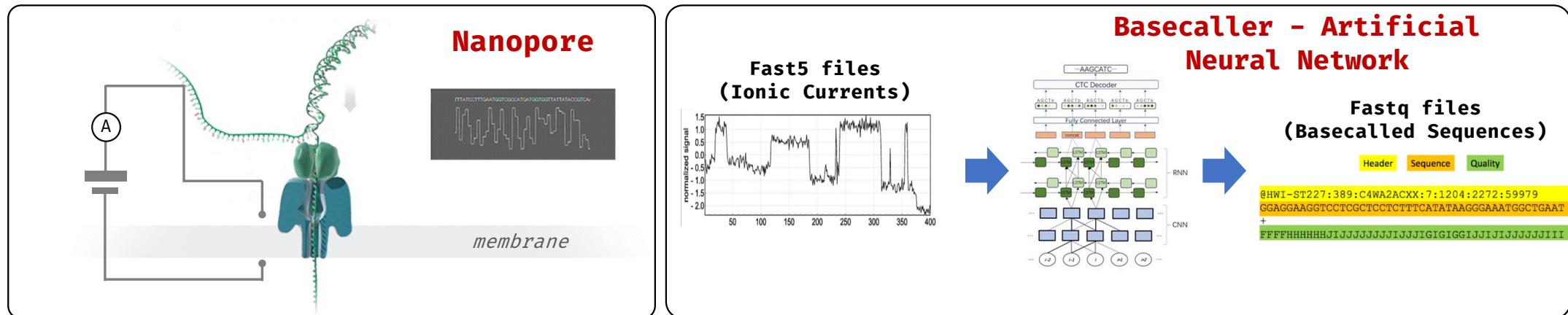
As we know, **RNA modifications** are both **transient** and **non-transient** alterations of ribonucleotides which contribute to increase the complexity of transcriptomes. To date, in the **epitranscriptomics** field, several important discoveries have been achieved but many problems remain to be solved:

- More than **170 types of RNA modifications** have been identified (**m6A**, **Inosine** the most studied);
- **Heterogeneity** of techniques for RNA modifications detection
- Diversity in **resolution** of the technique used and sometimes necessity of *in-vitro/in-vivo* pre-treatment of the samples (feeding of cell cultures with modified nucleotides).
- Lack of **standards** for the detection and classification of these modifications.
- **Copyright** of existing platform/portals.

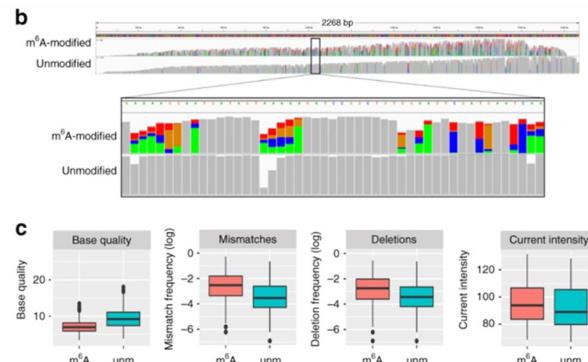


Taken and modified from Gatsiou and Stellos, 2018

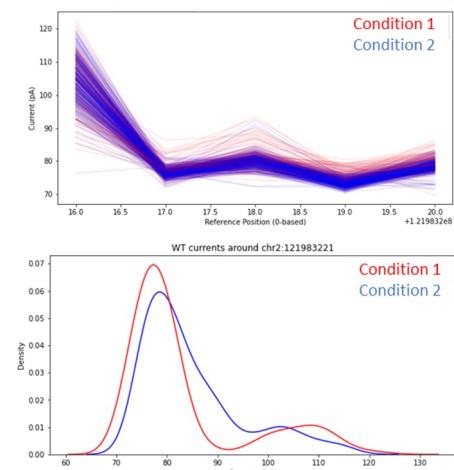
# Oxford Nanopore Sequencing Technology (ONT): Different approaches to detect modified bases



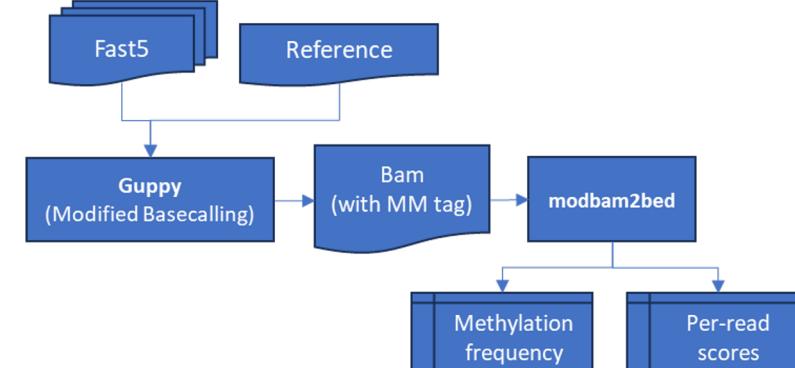
**A**  
Non-random Basecalling “errors”  
can be used to detect m6A modified base (EPINANO)



**B** Currents Intensities Comparison  
(Nanopolish, Tombo, f5c)



**C** Modified Basecallers  
(Guppy or Dorado with extended vocabulary  
for methylation detection)



taken and modified from Liu et al., 2019

# Case Study:

**Unraveling C-to-U RNA editing events from direct RNA sequencing**

(Fonzino et al., 2024, *RNA Biology*)

RNA BIOLOGY  
2024, VOL. 21, NO. 1, 1–14  
<https://doi.org/10.1080/15476286.2023.2290843>



Taylor & Francis  
Taylor & Francis Group

RESEARCH PAPER

OPEN ACCESS



**Unraveling C-to-U RNA editing events from direct RNA sequencing**

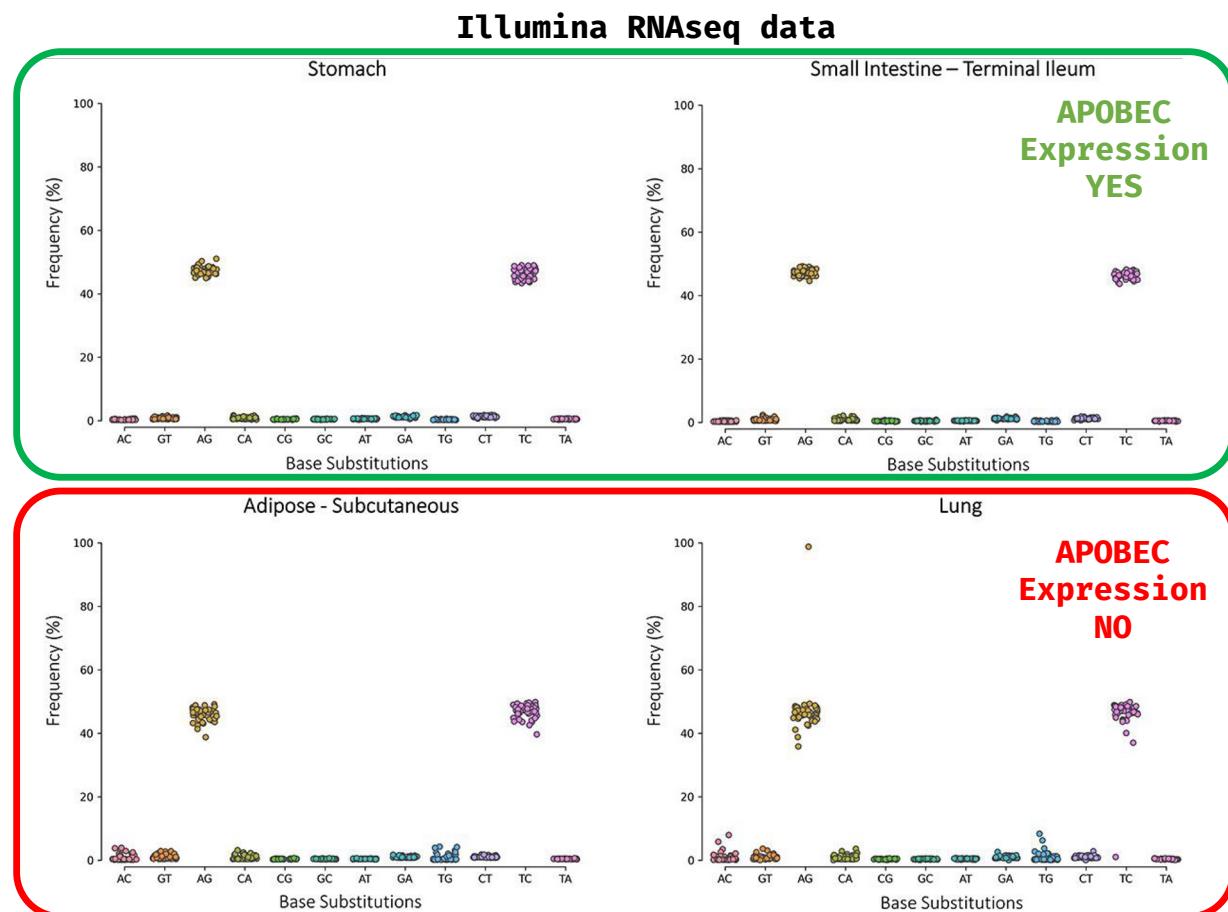
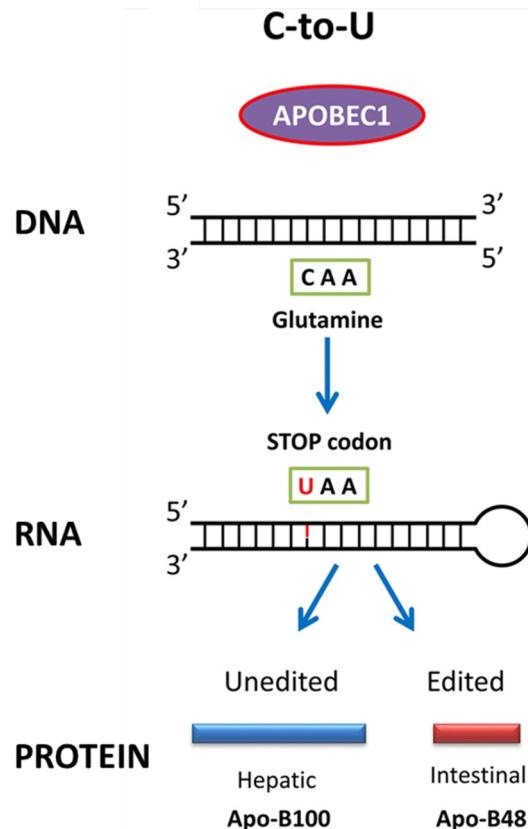
Adriano Fonzino<sup>a</sup>, Caterina Manzari<sup>a</sup>, Paola Spadavecchia<sup>a</sup>, Uday Munagala<sup>b</sup>, Serena Torrini<sup>b</sup>, Silvestro Conticello<sup>b,c</sup>, Graziano Pesole<sup>a,d,e</sup>, and Ernesto Picardi<sup>a,d,f</sup>

GitHub repository: [https://github.com/F0nz0/C\\_to\\_U\\_classifier](https://github.com/F0nz0/C_to_U_classifier)



## C-to-U editing detection challenges

Another uncommon **non-transient epitranscriptomic modification** occur via the hydrolytic deamination of **Cytosine (C)** in **Uracil (U)** mediated by the **APOBEC** family of enzymes (**C-to-U editing**). Despite the **A-to-I** detection can be carried out by **Illumina RNAseq technology**, the transcriptome-wide identification of **C-to-U editing** is not straightforward due to their **very low frequency**: background noise coming from sequencing errors and PCR artifacts can represent a challenge in this peculiar task.



# C-to-U editing detection with ONT: Experimental Design



**Oxford Nanopore Sequencing** technology (**ONT**) offers the possibility to investigate **direct-RNA** sequences without the need of a cDNA intermediate.

## Curlcakes (IVT)

***in-vitro transcribed synthetic construct*** with only **canonical unmodified nucleotides**.

4 sequences containing all the possible 5-mers combinations (*Liu et al., 2019*) with 2533.75 +/- 199.89 nt in lengths (mean +/- st.dev.).

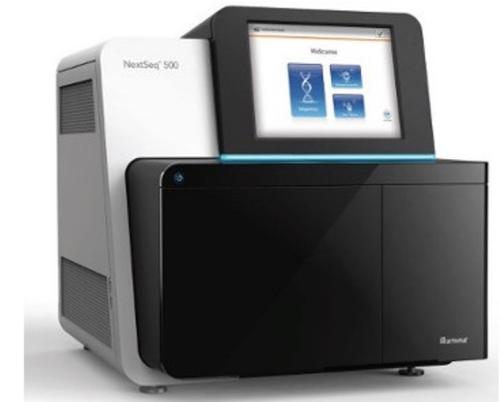
**Direct-RNA ONT sequencing**



## Murine dataset

Total RNA from **WT** and **APOBEC1 KO** cell line **RAW 264.7** murine macrophage cell line, was supplied by the Papavasiliou lab.

**Direct-RNA ONT sequencing & Illumina sequencing (for “ground-truth” sites)**



## Human dataset

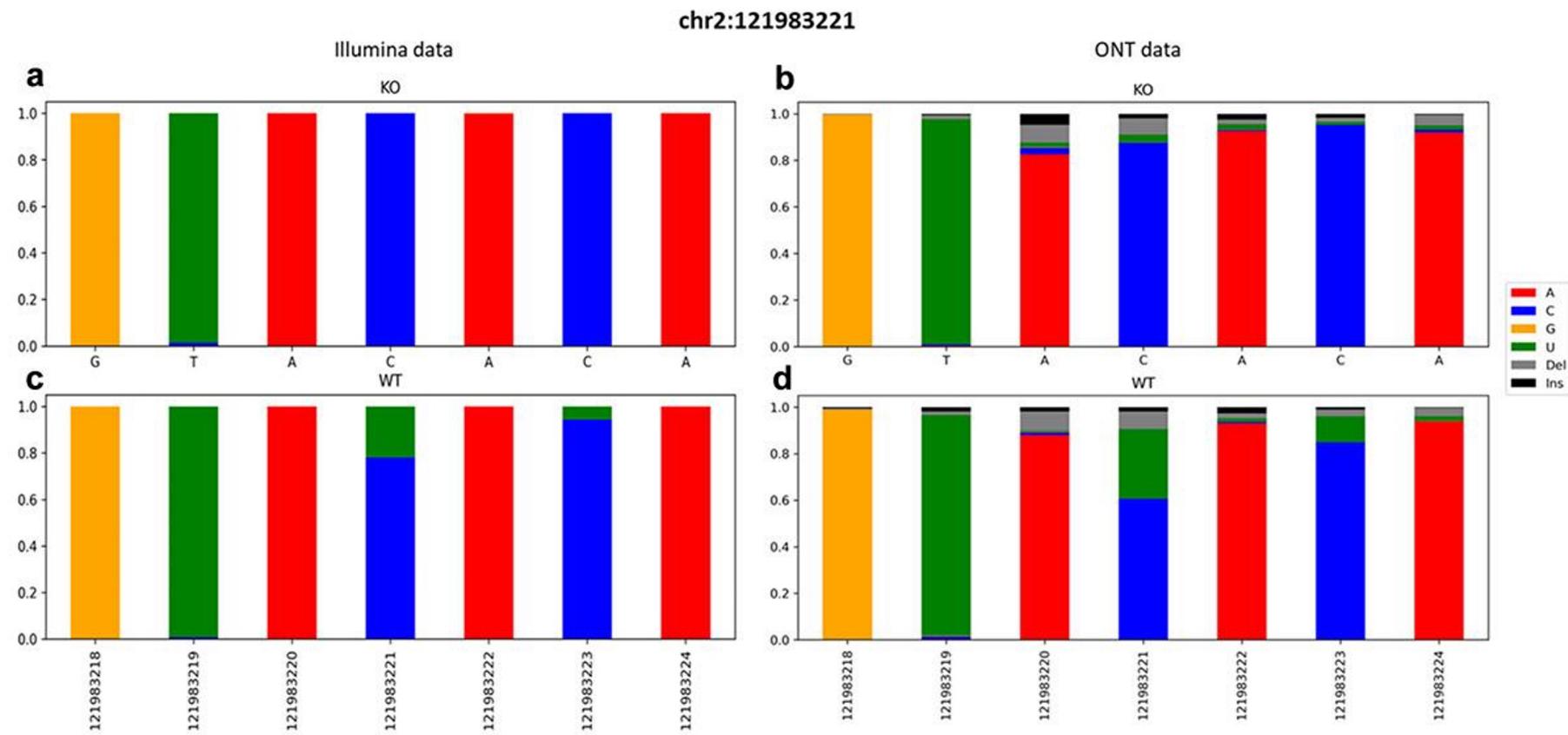
Total RNA from from **wild-type HEK293T** cells (**WThek**) and from cells **expressing APOBEC1** and **RBM47 (OVhek)** provided by Conticello Lab.

**Direct-RNA ONT sequencing & Illumina sequencing (for “ground-truth” sites)**



# C-to-U editing detection with ONT

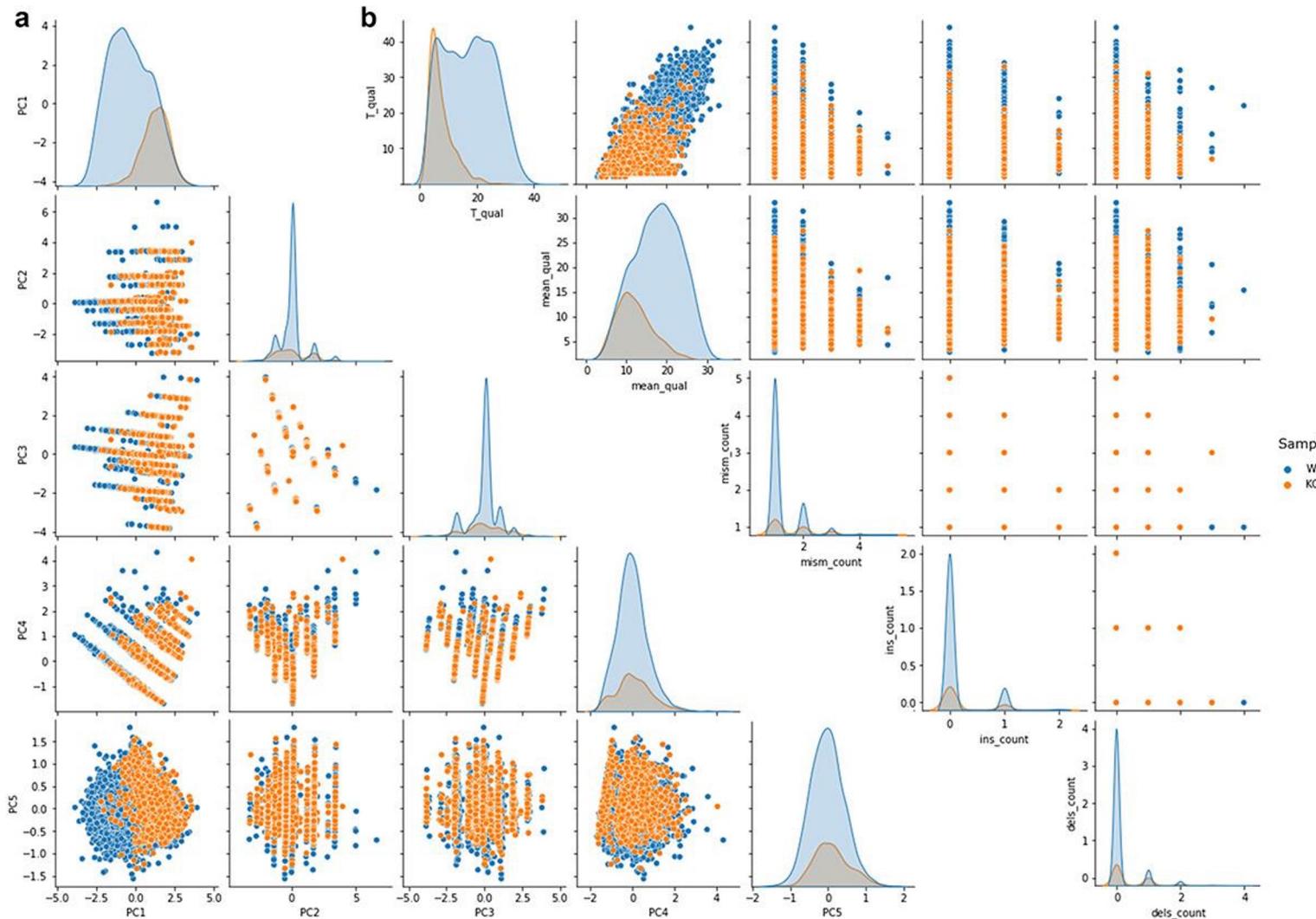
Looking at C-to-U nucleotides changes in ONT direct RNA reads, we found a basal noise affecting their correct detection consisting in **unexpected C-to-U substitutions**. We demonstrated here that, the huge amount of information provided by this innovative technology can be exploited to let the **C-to-U signal to emerge** after an **anomaly detection-based polishing** of native data.



# C-to-U changes in ONT murine dataset – Basecalling Features



Exploratory data analysis for CU context reads mapped on bona-fide editing sites



## Basecalling features

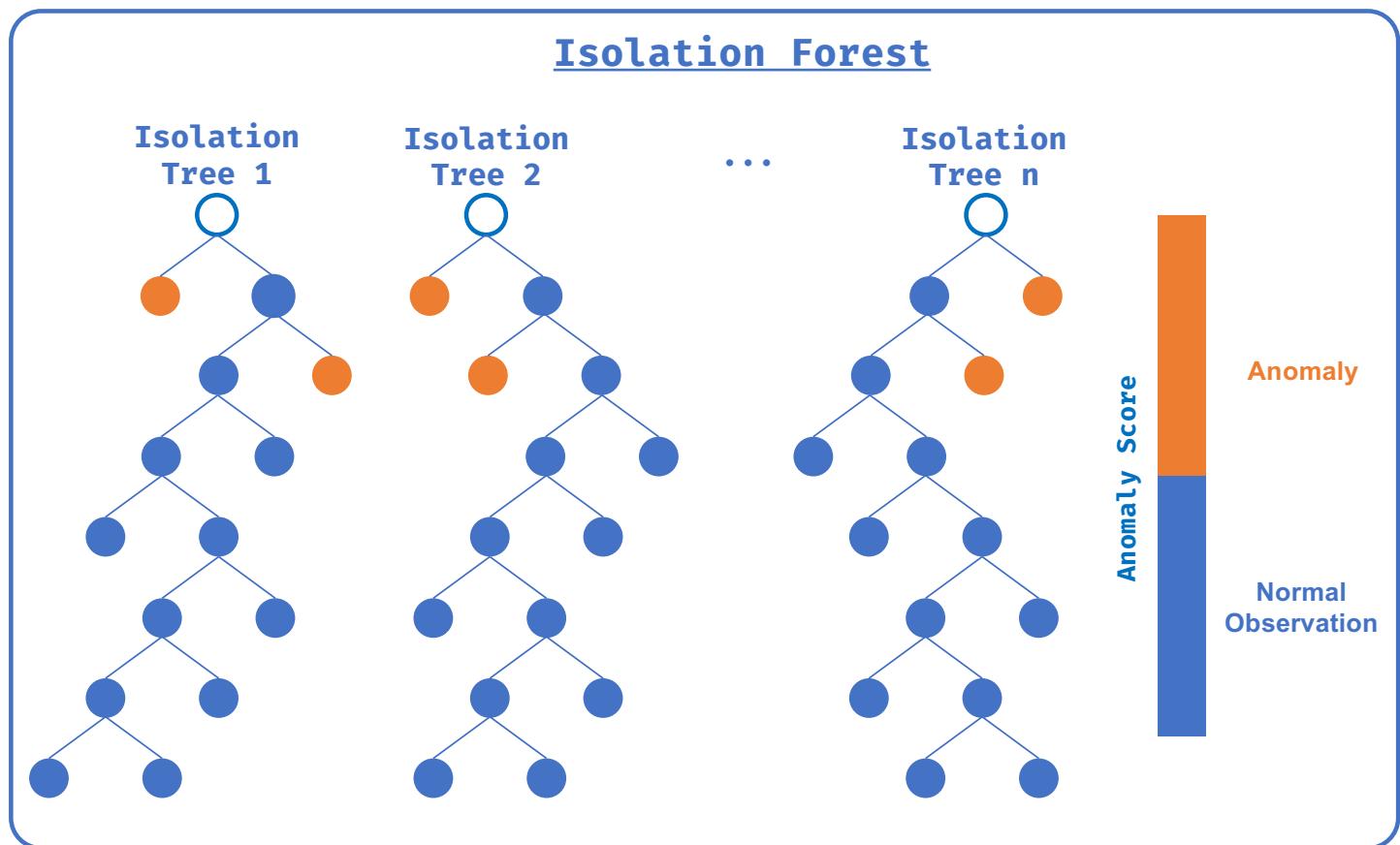
- U Base Quality
- Mean Quality
- Mismatches
- Deletions
- Insertions

## C-to-U editing detection with ONT: CtoUclassifier package and Isolation Forest



Harnessing these alignment and basecalling glitches, we developed a machine learning-based tool, the **CtoUclassifier**, able to denoise alignment profiles maintaining only high confident C-to-U changes. This tool leverages on the Isolation Forest (iForest), an anomaly detection algorithm.

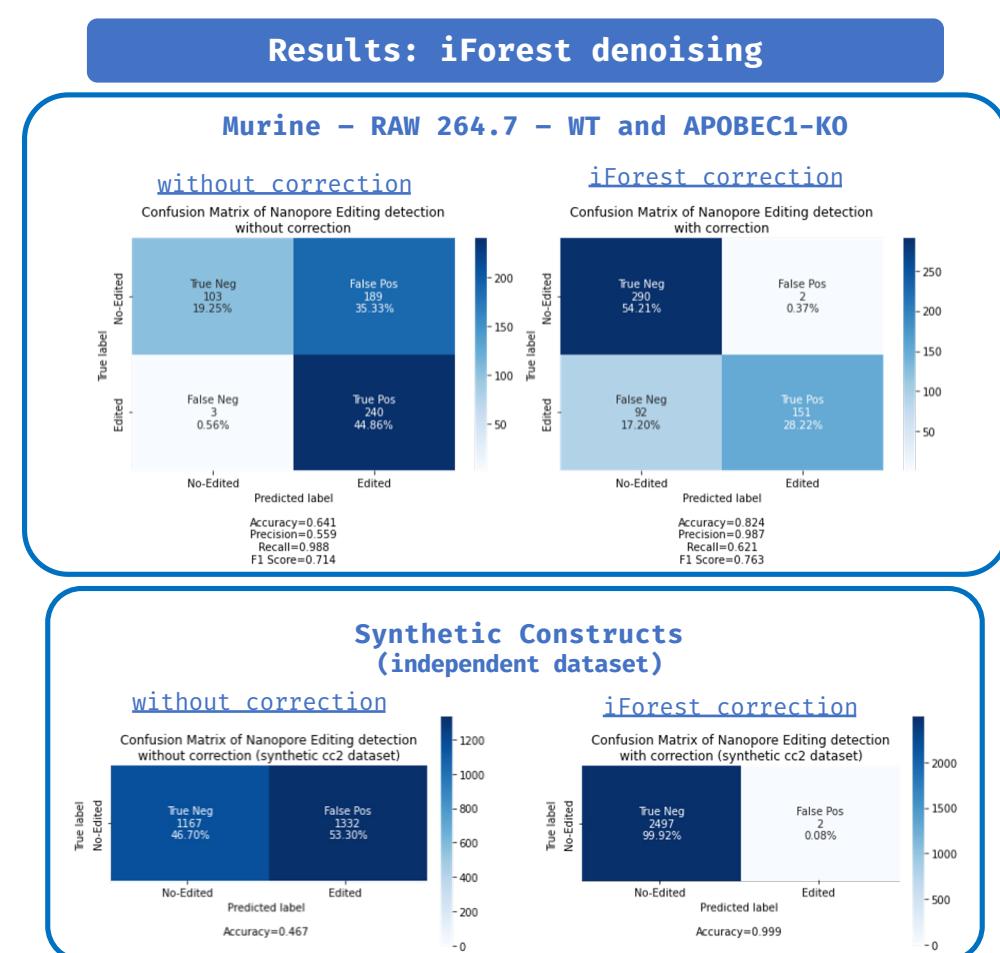
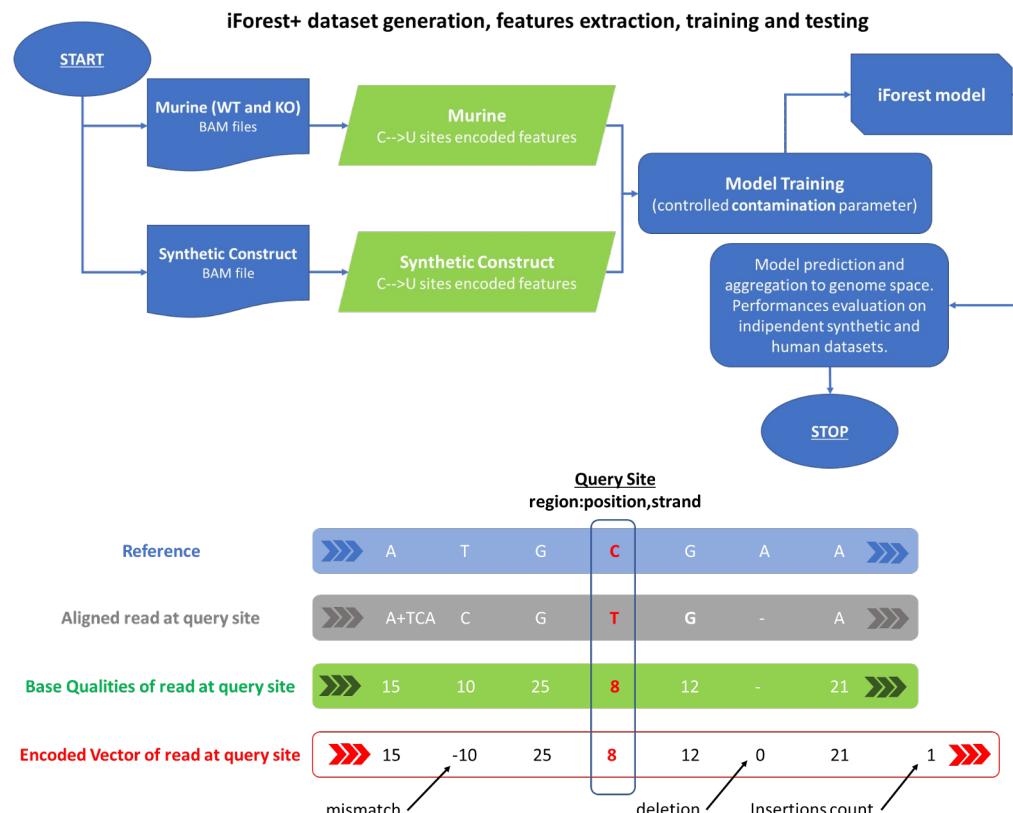
The **iForest** uses a **tree-based approach** to detect **anomalies** since these tend to be few in number and to isolate in a multidimensional space. A **random forest** in which decision trees are grown selecting **randomly** at each **node, features, threshold values** and a **subdataset**. The less the number of nodes required to isolate an observation, the more the probability to be an anomalous datapoint.



# C-to-U editing detection with ONT: CtoUclassifier



To unveil **APOBEC1 editing events** in **ONT direct RNA** sequencing experiments, we have developed a **novel machine-learning strategy** based on the **iForest** algorithm in which C-to-U editing changes are treated as **sequencing “anomalies”** and discriminated from ONT errors without the need of computational demanding analyses based on ionic currents and dwell times.





```
# connect via ssh to your RECAS area
ssh -i /path/to/private_key <username>@wn-gpu-8-3-2.recas.ba.infn.it

# print your current working directory
pwd

# go to traning-session directory and view its content
cd ~/CORSO_EPITRANSCR_2024_CTOU
ls -lh .

# visualize datasets: data directory with fastq sequences to map against reference sequences
ls -lh data/

# visualize datasets: reference sequences directory
ls -lh refs/
```

```
CORSO_EPITRANSCR_2024_CTOU/
    C_to_U_classifier/
    data/
        gBlocks.subsample.fastq      → Basecalled sequences for IVTs *
        KO_region_B2m.fastq       → Basecalled sequences for APOBEC-KO macrophagic mouse cell line (RAW 264.7)
        WT_region_B2m.fastq       → Basecalled sequences for WT murine cell line
    refs/
        gBlock_ref.fa              → Reference sequence for IVTs
        gBlock_ref.fa.fai
        GRCm38.primary_assembly.genome.filtered.fa      → Reference mouse genome
        GRCm38.primary_assembly.genome.filtered.fa.fai   (filtered to have only chr* contigs)
```

\* from Nguyen et al., 2022 (Nat. Methods)



```
# show CtoUclassifier package content  
ls -lh C to U classifier/
```

```
CORSO_EPITRANSCR_2024_CTOU/
  └── C_to_U_classifier/ → Main Package Directory (cloned from GitHub)
    ├── aggregate_results.py
    ├── pipe_basecalling.py → Python Script for the «Basecalling» pipeline
    ├── apobec1_models/ → Directory with APOBEC1 models
    │   (human or murine models to compute
    │   the APOBEC1 signature probability)
    ├── pipe_currents_extraction_step.py
    ├── cc1_cc2_freqs_thrs/
    ├── pipe_only_currents_extraction_step.py
    ├── predict_CT_iForest_cc.py
    ├── currents_dwell_retriever_multiprocessing.py
    ├── eventalign_splitter.py
    ├── README.md
    ├── filter_bam_file_for_training_dataset.py
    ├── requirements.txt
    ├── filters_currents_dwells_features.py
    ├── train_model_from_scratch.py
    ├── iForest_pretrained_model/ → Directory with iForest models for
    │   inference and correction steps on the
    │   extracted basecalling features
    ├── TT_CT_CC_basecalling_features_retriever.py
    ├── __init__.py.
    ├── utils.py
    └── venv/ → Directory with required virtual environment

  └── data/
  └── refs/
```



**PLEASE DON'T RUN THESE COMMANDS! WE HAVE ALREADY DONE FOR YOU TO SPARE TIME!**

**PLEASE, DON'T THANKS US! :D**

```
### PRELIMINARY STEPS ###
### ALREADY DONE FOR YOU TO SPARE TIME ###

# 1. Download the source code from GitHub repository at the url:
git clone https://github.com/F0nz0/C_to_U_classifier

# 2. Enter the C_to_U_classifier folder and decompress the iForest_pretrained_models folder:
cd C_to_U_classifier/
tar -xf iForest_pretrained_model.tar.gz
rm iForest_pretrained_model.tar.gz

# 3. Create a new virtual environment
# (it's suggested to create, use and activate a base conda environment with all the required software):
# 3.1. create a new conda environment adding required conda channels
conda config --add channels conda-forge
conda config --add channels bioconda
conda create --name CtoU_2024 python=3.8
# 3.2. activate the conda environment
conda activate /home/instructor_3/.conda/envs/CtoU_2024
# 3.3. install samtools
conda install -c bioconda samtools==1.6
```



**PLEASE DON'T RUN THESE COMMANDS! WE HAVE ALREADY DONE FOR YOU TO SPARE TIME!**

**PLEASE, DON'T THANKS US! :D**

```
# 3.4. install minimap2 aligner
conda install -c bioconda minimap2==2.24
# 3.5. install f5c (optimised re-implementation of the call-methylation and eventalign modules in
Nanopolish)
conda install -c bioconda f5c==1.1

# 3.6. create virtual environment starting from the conda CtoU_2024 environment
python3 -m venv venv

# 4. Activate the venv
source ~/CORSO_EPITRANSCR_2024_CTOU/C_to_U_classifier/venv/bin/activate

# 5. Upgrade pip version:
python3 -m pip install --upgrade pip

# 6. Install wheel package via pip:
pip install wheel

# 7. Install required Python packages using the requirements.txt file:
python -m pip install -r requirements.txt
```



## HANDS ON! It's your turn! ☺

Check whether everything is working fine:

```
### HANDS ON! ###
# 8.1. Test installation and package
# 8.2. Activate conda and venv virtual environments
conda activate /home/instructor_3/.conda/envs/CtoU_2024
source ~/CORSO_EPITRANSCR_2024_CTOU/C_to_U_classifier/venv/bin/activate

# 8.3. Launch help message of CtoUclassifier basecalling pipeline main script
python3 ~/CORSO_EPITRANSCR_2024_CTOU/C_to_U_classifier/pipe_basecalling.py -h
usage: pipe_basecalling.py [-h] -B BAM_FILEPATH -R REFERENCE_FILEPATH [-T THRESHOLD] [-threads THREADS]
                           [-aligner ALIGNER] [-O ORGANISM]
```

The **C\_to\_U\_classifier** is able to ameliorate the C-to-U editing signal in direct-RNA Nanopore runs.

The **Basecalling Pipeline** extracts from the BAM file **C-to-U signals** at **per-read** level for every position with a depth higher than a given **threshold** (default 50 reads).

Main available options are:

- B -----> BAM file path
- R -----> Reference file path (filtered in order to contain only chr\* like contigs)
- T -----> a threshold for the minimum number of aligned reads to analize [50]
- threads --> Number of threads to process contigs/chromosome in parallel
- O -----> Organisms (<murine>, <human> or omitted) to be used for the computation of the APOBEC1 signature with a Likelihood-ratio test



Start alignments with minimap2 of basecalled reads.

Alignment of WT RAW 264.7 murine reads against reference genome (only chr\* contigs):

```
# 9. Create a new directory to store results
cd ~/CORSO_EPITRANSCR_2024_CTOU
mkdir CtoU_2024
cd CtoU_2024

# 10. ##### WT #####
# 10.1. alignment to reference genome with minimap2 in splice mode (~3 min)
minimap2 -t 2 -ax splice -uf -k14 --secondary=no
~/CORSO_EPITRANSCR_2024_CTOU/refs/GRCm38.primary_assembly.genome.filtered.fa
~/CORSO_EPITRANSCR_2024_CTOU/data/WT_region_B2m.fastq > WT_region_B2m.sam

# 10.2. filtering of unmapped reads sorting and conversion to binary alignment file
samtools view -b -F 2308 WT_region_B2m.sam | samtools sort -O BAM > WT_region_B2m.bam

# 10.3. index the BAM file
samtools index WT_region_B2m.bam

# 10.4. remove unused SAM file
rm WT_region_B2m.sam
```



Alignment of **APOBEC1-KO** RAW 264.7 murine reads, against the mouse reference genome:

```
# 11. ##### K0 #####
# 11.1. alignment to reference genome with minimap2 in splice mode (~3 min)
minimap2 -t 2 -ax splice -uf -k14 --secondary=no
~/CORSO_EPITRANSCR_2024_CTOU/refs/GRCm38.primary_assembly.genome.filtered.fa
~/CORSO_EPITRANSCR_2024_CTOU/data/K0_region_B2m.fastq > K0_region_B2m.sam

# 11.2. filtering of unmapped reads sorting and conversion to binary alignment file
samtools view -b -F 2308 K0_region_B2m.sam | samtools sort -O BAM > K0_region_B2m.bam

# 11.3. index the BAM file
samtools index K0_region_B2m.bam

# 11.4. remove unused SAM file
rm K0_region_B2m.sam
```



Alignment of **IVT** reads (no CtoU events are expected), against the reference sequences:

```
# 12. ##### gBlocks #####
# 12.1. alignment to reference sequence with minimap2 (no polarity expected)
minimap2 -t 2 -ax map-ont -k14 --secondary=no --for-only ~/CORSO_EPITRANSCR_2024_CTOU/refs/gBlock_ref.fa
~/CORSO_EPITRANSCR_2024_CTOU/data/gBlocks.subsample.fastq > gBlocks.subsample.sam

# 12.2. filtering of unmapped reads sorting and conversion to binary alignment file
samtools view -b -F 2308 gBlocks.subsample.sam | samtools sort -O BAM > gBlocks.subsample.bam

# 12.3. index the BAM file
samtools index gBlocks.subsample.bam

# 12.4. remove unused SAM file
rm gBlocks.subsample.sam

# 13. List of produced output files
ls -lh .

# 14. view some alignments statistics using a simple for loop
for bam in $(ls *.bam); do echo Stats for BAM: $bam; samtools flagstat $bam; echo ""; done
```



We can now launch the **CtoUclassifier basecalling pipeline** to detect, at a **per-read level**, **CU contexts reads** and their position. After the **denoising** by iForest model, these intervals will be classified as **real CU substitutions** (due to either editing or genomic variations) or as **errors**. Then the data will be **aggregated** onto the «**genome-space**» to evaluate sites with no editing evidence on the **APOBEC1-KO** sample but with a residual signal on the **WT**. The pipeline will be launched also on **IVT reads**, as well.

```
# 15. launch CtoUclassifier for denoising on WT murine sample (~2 minutes)
python3 ~/CORSO_EPITRANSR_2024_CTOU/C_to_U_classifier/pipe_basecalling.py -B WT_region_B2m.bam -R
~/CORSO_EPITRANSR_2024_CTOU/refs/GRCm38.primary_assembly.genome.filtered.fa -O murine

# 16. launch CtoUclassifier for denoising on WT murine sample (~2 minutes)
python3 ~/CORSO_EPITRANSR_2024_CTOU/C_to_U_classifier/pipe_basecalling.py -B K0_region_B2m.bam -R
~/CORSO_EPITRANSR_2024_CTOU/refs/GRCm38.primary_assembly.genome.filtered.fa -O murine

# 17. launch CtoUclassifier for denoising on IVT sample. It will be a negative control (few seconds)
python3 ~/CORSO_EPITRANSR_2024_CTOU/C_to_U_classifier/pipe_basecalling.py -B gBlocks.subsample.bam -R
~/CORSO_EPITRANSR_2024_CTOU/refs/gBlock_ref.fa
```

## CtoUclassifier: Description of Outputs 1

### Basecalling Features



Visualize **output folders** and files generated by the **CtoUclassifier**. Understanding of file structure of the extracted **basecalling features** at a per-read level:

```
### 18. Description of OUPUTS (IVT as example)
# two folders produced: 1) <bam_name>.basecalling_features 2) <bam_name>.model_iForest_pretrained_results
ls -lh

# 18.1. see basecalling features (TT, CC and CT context reads with basecalling features)
ls -lh gBlocks.subsample.basecalling_features

# 18.2. see CT context reads/trascript which will be denoised (one file per reference contig)
ls -lh gBlocks.subsample.basecalling_features/gBlocks.subsample.CTcontext_reads_features_forward_rev

# 18.3. view basecalling features of IVTs reads mapped on chr2
cat
gBlocks.subsample.basecalling_features/gBlocks.subsample.CTcontext_reads_features_forward_rev/CTcontext_reads_
_features_forward_rev_chr2.tsv | head -3
```

# CtoUclassifier: Description of Outputs 2

## Basecalling Features



```

CORSO_EPITRANSCR_2024_CTOU/
└── CtoU_2024/
    ├── gBlocks.subsample.bam
    ├── gBlocks.subsample.bam.bai
    └── gBlocks.subsample.basecalling_features/
        ├── gBlocks.subsample.CCcontext_reads_features_forward/
        └── gBlocks.subsample.CTcontext_reads_features_forward_rev/
            ├── CTcontext_reads_features_forward_rev_chr1.tsv
            └── CTcontext_reads_features_forward_rev_chr2.tsv
            └── CTcontext_reads_features_forward_rev_chr3.tsv
        └── gBlocks.subsample.TTcontext_reads_features_forward/
    └── gBlocks.subsample.model_iForest_pretrained_results/
        ├── KO_region_B2m.bam
        ├── KO_region_B2m.bam.bai
        └── KO_region_B2m.basecalling_features/
    └── KO_region_B2m.model_iForest_pretrained_results/
        ├── WT_region_B2m.bam
        ├── WT_region_B2m.bam.bai
        └── WT_region_B2m.basecalling_features/
    └── WT_region_B2m.model_iForest_pretrained_results/
    └── C_to_U_classifier/
    └── data/
    └── refs/

```

### Stranded Basecalling Features:

+/-3 nt interval with encoded *base quality*,  
 negative values for *mismatched bases*,  
 0 value for *deletions* and  
*insertions count* in the last column

The table displays five rows of stranded basecalling features for different genomic regions and positions. The columns represent the region (Reg), position (Pos), read ID (Read-ID), strand (Str), and quality scores for positions P-3 through P+3, along with an insertion count (Ins).

Reg	Pos	Read-ID	Str	P-3	P-2	P-1	P0	P+1	P+2	P+3	Ins
chr2	13	0eb...5c6	+	16.0	10.0	4.0	3.0	6.0	-0.0	17.0	0.0
chr2	33	1d5...b18	+	26.0	25.0	19.0	19.0	24.0	22.0	12.0	0.0
chr2	33	6db...c00	+	8.0	4.0	14.0	7.0	11.0	-0.0	-0.0	0.0
chr2	38	1b7...3a8	+	25.0	23.0	25.0	11.0	11.0	16.0	16.0	0.0
chr2	38	0eb...5c6	+	25.0	17.0	5.0	5.0	12.0	12.0	18.0	1.0

**Positional Info**

## CtoUclassifier: Description of Outputs 3

### iForest denoising



Exploration of output tables produced by **iForest predictions** and denoising on **CU-context reads basecalling features**.

```
# 18.4. to see iforest predictions performed on basecalling features of CU-context reads
ls -lh gBlocks.subsample.model_iForest_pretrained_results/

# 18.5. see per-read predictions
cat gBlocks.subsample.model_iForest_pretrained_results/df_CT_predicted.tsv | head -5

# 18.6. see genome-space aggregated predictions
cat gBlocks.subsample.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv | head -5
```

# CtoUclassifier: Description of Outputs 4

## iForest denoising



region	position	strand	Tnative	Tcorrected	depth	Tfreq	Tfreq	5mer	y_hat
					stranded	native	corrected		
chr2	13	+	1	0	145	0.0068	0.0	CACGA	0
chr2	33	+	2	0	145	0.0137	0.0	AGCAA	0
chr2	38	+	30	0	135	0.2222	0.0	TTCTT	0
chr2	42	+	2	0	140	0.0142	0.0	TGCAC	0
chr2	44	+	1	0	140	0.0071	0.0	CACTT	0

CORSO\_EPITRANSCR\_2024\_CTOU/

```

  └── CtoU_2024/
      ├── gBlocks.subsample.bam
      ├── gBlocks.subsample.bam.bai
      ├── gBlocks.subsample.basecalling_features/
      ├── gBlocks.subsample.model_iForest_pretrained_results/
          ├── df_CT_predicted_aggregated.tsv
          └── df_CT_predicted.tsv
      ├── KO_region_B2m.bam
      ├── KO_region_B2m.bam.bai
      ├── KO_region_B2m.basecalling_features/
      ├── KO_region_B2m.model_iForest_pretrained_results/
      ├── WT_region_B2m.bam
      ├── WT_region_B2m.bam.bai
      ├── WT_region_B2m.basecalling_features/
      ├── WT_region_B2m.model_iForest_pretrained_results/
      └── C_to_U_classifier/
          ├── data/
          └── refs/

```

region	position	read_name	strand	pred
chr2	13	0eb...f5c6	+	0.0
chr2	33	1d5...bb18	+	0.0
chr2	33	6db...c00	+	0.0
chr2	38	1b7...3a8	+	0.0

## CtoUclassifier: Description of Outputs 5

### iForest denoising



```
# 18.7. count CU sites before and after iForest denoising
# 0: no real CU site; 1: real CU site
cat gBlocks.subsample.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv | cut -f 10 | sort | uniq -c
Out:
 189 0
   1 1
   1 y_hat
```

```
# 18.8. the same thing but for WT and APOBEC1-KO murine cell lines
cat WT_region_B2m.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv | cut -f 10 | sort | uniq -c
Out:
 175 0
  11 1
   1 y_hat

cat K0_region_B2m.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv | cut -f 10 | sort | uniq -c
Out:
 184 0
   1 1
   1 y_hat
```

## CtoUclassifier: Description of Outputs 6

### iForest denoising



```
# 18.9. See B2m known C-to-U editing site at chr2:122152740(+) from murine WT and K0 data
awk '$1=="chr2" && $2==122152740 || $1=="region"'
WT_region_B2m.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv
```

**Out:**

```
region position strand T_native T_corrected depth_stranded Tfreq_native Tfreq_corrected 5mer y_hat apobec1_pvalue
chr2 122152740 + 280 85 848 0.330188 0.100235 TACAC 1 0.00441884749
```

```
awk '$1=="chr2" && $2==122152740 || $1=="region"'
K0_region_B2m.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv
```

**Out:**

```
region position strand T_native T_corrected depth_stranded Tfreq_native Tfreq_corrected 5mer y_hat apobec1_pvalue
chr2 122152740 + 28 1 817 0.034271 0.001223 TACAC 0 0.00441884749
```

## CtoUclassifier: Description of Outputs 7

### iForest denoising



```
# 18.10. See B2m known SNP site chr2:122147789(+) from murine WT/KO data
awk '$1=="chr2" && $2==122147789 || $1=="region"'
WT_region_B2m.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv
```

**Out:**

```
region position strand T_native T_corrected depth_stranded Tfreq_native Tfreq_corrected 5mer y_hat apobec1_pvalue
chr2 122147789 +      696      224          792          0.878787      0.282828      GCCTG 1      1.0
```

```
awk '$1=="chr2" && $2==122147789 || $1=="region"'
KO_region_B2m.model_iForest_pretrained_results/df_CT_predicted_aggregated.tsv
```

**Out:**

```
region position strand T_native T_corrected depth_stranded Tfreq_native Tfreq_corrected 5mer y_hat apobec1_pvalue
chr2 122147789 +      687      225          785          0.875159      0.286624      GCCTG 1      1.0
```

**Final note:** CtoUclassifier model was trained on R9 legacy chemistry, so an update have to be released for the portability on the newest 004 version of ONT pores. For further details, visit GitHub repository at: [https://github.com/F0nz0/C\\_to\\_U\\_classifier](https://github.com/F0nz0/C_to_U_classifier)



# Thank you for your attention...

## Questions time!

e-mail: [adriano.fonzino@uniba.it](mailto:adriano.fonzino@uniba.it)

## Student ID ----> Public Key

- |                                  |                                     |
|----------------------------------|-------------------------------------|
| 1: barbara_perugini.pub          | 20: leda_katopodi.pub               |
| 2: biancaantonica_rsa.pub        | 21: livio.muccillo_ssh_key_win.pub  |
| 3: carlotta_spattini_ub.pub      | 22: lucilla-crudele-key.pub         |
| 4: Chiara_Arcuri-key.pub         | 23: martina_terrusa-key.pub         |
| 5: cilli_public_key.pub.pub      | 24: matilde_conte_epitrans.pub      |
| 6: claudiaKey_win.pub            | 25: mimonti-ssh-key.pub             |
| 7: coffa_martina.pub             | 26: monica_marra.pub                |
| 8: dematteis_publickey.pub       | 27: moras_beatrice.pub              |
| 9: derisog_ssh_key.pub           | 28: picarone_key.pub                |
| 10: elena_sq_ssh.pub             | 29: pizzulli_cristina.pub           |
| 11: fabio_lauria.pub             | 30: roberto_petruzzi.pub            |
| 12: fb-ssh-key-bari.pub          | 31: semente_miriam.pub              |
| 13: federica.viola.pub           | 32: SSHkey_bari_tanada.pub          |
| 14: FrazzettoKey_rsa.pub         | 33: steaud_key.pub                  |
| 15: giada_spadavecchia.pub       | 34: tavellat_key.pub                |
| 16: ighlidibari.pub              | 35: Maria_Arconzo_my_ssh_key.pub    |
| 17: ilaria_monaco.pub            | 36: luciaSSHKey_win.pub             |
| 18: key_bari_wengeralice.pub     | 37: key_ssh.pub                     |
| 19: KEY_margherita-degasperi.pub | 38: _C_maricacariello_gmail_com.pub |