ICVS Life and Health Sciences Research Institute

INSTITUTO DE INVESTIGAÇÃO E INOVAÇÃO EM SAÚDE
UNIVERSIDADE DO PORTO

Universidade do Minho
Escola de Engenharia

# Identification of *Mycobacterium tuberculosis* Genetic Determinants of Disease Severity

**Rita Nóbrega Amaral Martins**
PG46733

Work supervised by Dr. **Nuno Ósório** and Dr. **Tiago Beites**

Project Presentation
Master's in Bioinformatics, University of Minho

29th May 2024

# *Mycobacterium tuberculosis (Mtb)*

***Mtb*** is a pathogenic bacteria

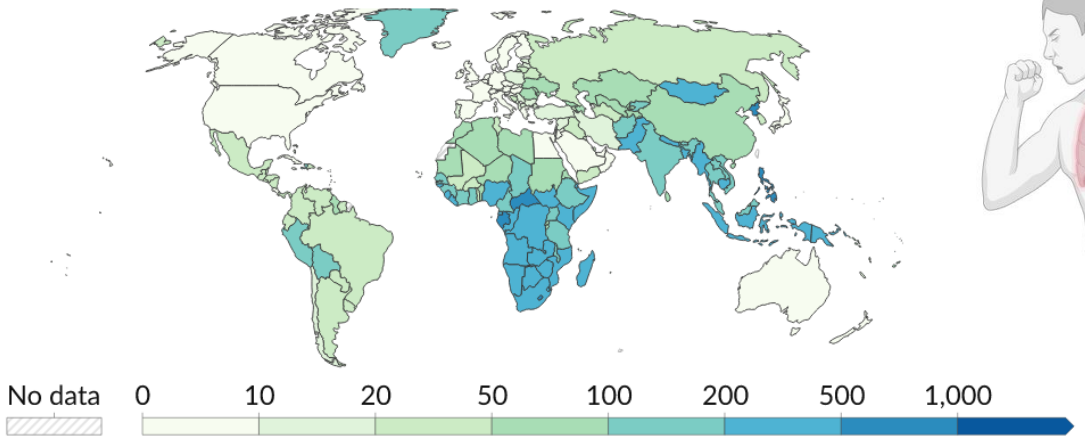
***Mtb*** is the etiological agent of **Tuberculosis (TB)**

- Spectrum of clinical manifestations
- Influenced by pathogen, host and environment

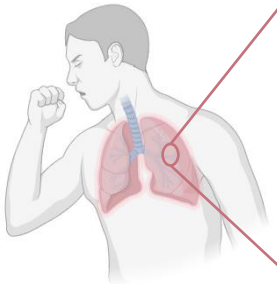
In 2022, WHO estimates:

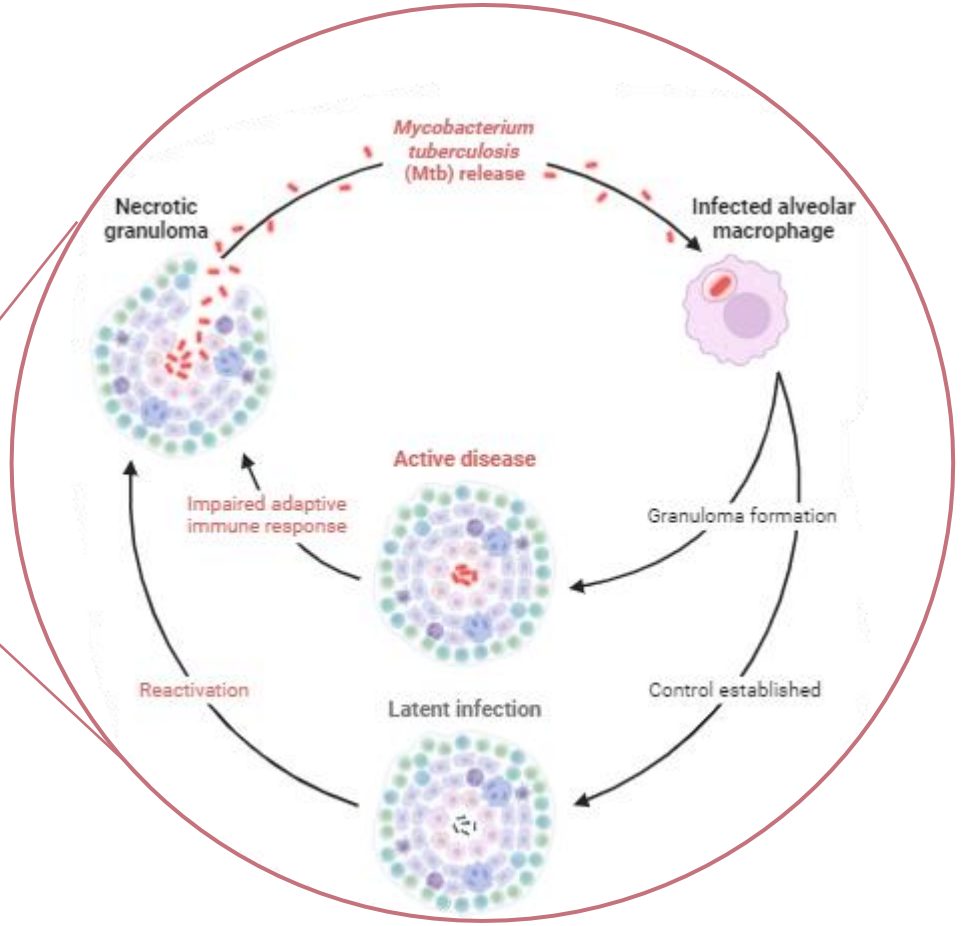
**7.5 million**
**newly diagnosed**

**1.3 million**
**TB deaths**




Rate of new TB cases per 100 000 people
Source: WHO, 2023






Adapted from Chandra et al., 2022

# Therapeutical Strategies

## Combined Antibiotics

- Pathogen's fast adaptation
- Resistant strains
- Side effects
- Co-infections Co-morbidities

**Potential novel strategy: Antivirulence drugs**

Target pathogens' **ability to cause disease**

**without** directly affecting **viability**

Chandra et al., 2022 Young et al., 2020 Dartois et al., 2022

# Genetic variants of *Mtb*

**Potential targets** for antivirulence drugs

↓

Is there an **association** between **genetic diversity within *Mtb*** and **TB severity ?**

↓

How can we find these genetic variations?

Pre-processing of sequence reads → Alignment/Mapping to Reference Genome → Variant Detection and Filtering → Variant and lineage annotation

Sousa et al., 2020

# Variant Calling for *Mtb*

Examples of existing pipelines for *Mtb*: MAGMA, MTB-VCF, MycoVarP and TBProfiler

Tools' **lack** of flexibility and robustness

Tools' **limited** configurability

*Mtb*'s **complexicity**

*Mtb*'s **repetitive** regions

**Removal** of important regions

Missing or **misidentified** genetic variants

**Discrepancy between phenotypic differences** among *Mtb* isolates and the **absence of corresponding genetic differences**

Sousa et al., 2020

# Scientific question and Aims

**Is there an association between genetic variations in *Mtb* and TB severity ?**

**1. Identify genetic variations of *Mtb* overlooked by existing pipelines**

Development of an **optimized pipeline** for the **identification of genetic variants in *Mtb***

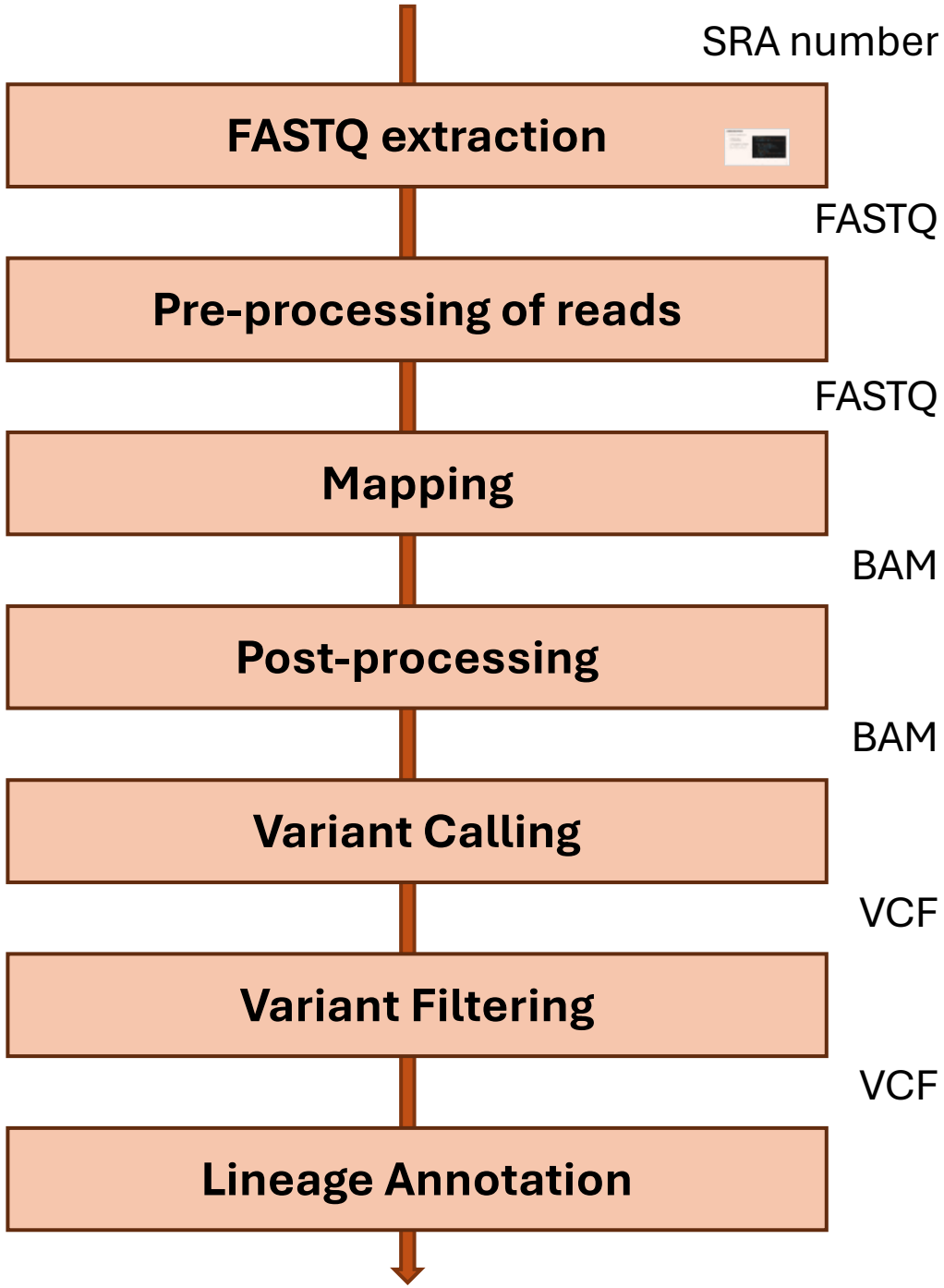
**2. Correlate the genetic variants with disease severity**

Study the **correlation** between the **genetic variations** identified and **TB clinical outcome** through **statistical and machine learning approaches**

# Task 1
# Development of the pipeline

- Jupyter notebook with Bash language
- A cell/function for each step of the pipeline

- Input: Text file with SRA numbers
- Output: List of variants identified



# FASTQ extraction

Input: List of 149 isolates of *Mtb*

- Folder per sample
- Downloads SRA data
- Converts SRA data into FastQ files

Output: 2 FastQ files per sample

```
extract_SRA_files () {
    echo "Extracting FastQ files"
    local sample="$1"
    local sra_diretoria="$HOME/Projeto_Mtb/Sample_reads"

    while IFS= read -r sample; do
        echo -e "\nSample: $sample"
        mkdir -p "$sra_diretoria/$sample"
        cd "$sra_diretoria/$sample" || exit 1

        prefetch "$sample"
        fasterq-dump "$sample"
        rm $HOME/Projeto_Mtb/SRA_files/sra/"$sample".sra

    done < "$sample"
}
```

# Pre-processing of reads

## Quality Control

Input: List of isolates

Uses: 2 FASTQ files

Output: 2 FASTQC files – Quality Reports

```
quality_control () {
  echo 'Quality control'
  local sra_diretoria="$HOME/Projeto_Mtb/Sample_reads"

  local sample_file="$1"
  sed -i 's/\r$//' "$sample_file"

  if [ ! -f "$sample_file" ]; then
      echo "Arquivo de amostra '$sample_file' não encontrado."
      return 1
  fi

  parallel -j "$(nproc)" "
      sample={};
      if [ -f \"$sra_diretoria/\$sample/\${sample}_1.fastq\" ] \
      && [ -f \"$sra_diretoria/\$sample/\${sample}_2.fastq\" ]; then
          fastqc \"$sra_diretoria/\$sample/\${sample}_1.fastq\" \
          \"$sra_diretoria/\$sample/\${sample}_2.fastq\";
      else
          echo \"Arquivo fastq não encontrado para a amostra '\$sample'.\";
      fi
    " < "$sample_file"
}
```

## Trimming with BBDuk

- Trimming FastQ file to remove adapters and low-quality reads

Output: 2 trimmed FastQ files

```
trimming_bbduk () {
  echo 'Read trimming with BBDuk'
  local sample_file="$1"
  sed -i 's/\r$//' "$sample_file"
  cat "$sample_file" | xargs -I {} -P "$(nproc)" bash -c '
    sample={};
    echo "Sample: $sample";
      if [ -f "$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}_1.fastq" ] \
      && [ -f "$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}_2.fastq" ]; then
        bbduk.sh \
          in1="$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}_1.fastq" \
          in2="$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}_2.fastq" \
          out="$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}_R1_bbduk.fastq" \
          out2="$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}_R2_bbduk.fastq" \
          overwrite=t \
          ref=~/Projeto_Mtb/NGS_helper_files/adapters_combined_256_unique.fasta \
          ftm=5 ktrim=r k=19 mink=8 editdistance=1 editdistance2=1 \
          trimpairsevenly=f removeifeitherbad=t \
          qtrim=r trimq=20 trimpolygright=10 \
          minavgquality=20 minlength=20 ottm=t \
          rename=t ziplevel=1 showspeed=t ;
        rm "$HOME/Projeto_Mtb/Sample_reads/$sample/${sample}"_[1-2].fastq ;
        echo "Trimming for $sample completed"
      else
        echo "Arquivo fastq não encontrado para a amostra $sample.";
      fi
      '
}
```

# Mapping

Input: List of isolates

- Mapping using **‘bwa mem’**
  - Trimmed FastQ files
  - FASTA of a reference genome: ***Mtb H37Rv***
- Sorts and compresses BAM files using **‘samtools’**

Output: BAM file

```
map_bwa() {
  echo "Mapping genomes to MTB_anc with bwa mem + sorting BAM by read name"

  local sample_file="$1"
  local genome_reference="$HOME/Projeto_Mtb/NGS_helper_files/MTB_anc.fasta"
  local sra_diretoria="$HOME/Projeto_Mtb/Sample_reads"

  while IFS= read -r sample; do
    echo -e "\nMapping Sample: $sample"

    bwa_threads=$(nproc)
    bwa mem -t "$bwa_threads" "$genome_reference" \
    "$sra_diretoria/$sample/${sample}_R1_bbduk.fastq" \
    "$sra_diretoria/$sample/${sample}_R2_bbduk.fastq" \
    | samtools sort -n -l 1 -@ 1 -o "$sra_diretoria/$sample/$sample.bam"

    samtools view -@ 1 -b "$sra_diretoria/$sample/$sample.bam" \
    | samtools sort -@ 1 -o "$sra_diretoria/$sample/${sample}_sorted.bam"
    samtools index -@ 1 "$sra_diretoria/$sample/${sample}_sorted.bam"

    echo "Mapping for $sample completed."
  done < "$sample_file"
}
```

# Post-Processing

## Duplicates Marking with Samtools

Input: List of isolates

- Marks duplicates and indexes BAM files per sample with **'samtools markup'**

```
mark_duplicates() {
  echo "Marking duplicates with samtools markdup and indexing BAM files"
  local sample_file="$1"
  local sra_diretoria="$HOME/Projeto_Mtb/Sample_reads"
  local threads=$(nproc)

  while IFS= read -r sample || [[ -n $sample ]]; do
    samtools fixmate -m -@ 2 "$sra_diretoria/$sample/$sample.bam" -u - \
    | samtools sort -u -@ 2 - \
    | samtools markdup --include-fails -S --mode s -@ 2 - -O bam,level=1 \
    "$sra_diretoria/$sample/${sample}_markdup.bam" \
    && samtools index -@ 2 "$sra_diretoria/$sample/${sample}_markdup.bam" \
    && rm "$sra_diretoria/$sample/$sample.bam"

  done < "$sample_file"

  echo "Duplicate marking and indexing finished"
}
```

## BAM Coverage with mosdepth

Input: List of isolates

- Calculates coverage using '**mosdepth**' per base and per region

```
calculate_bam_coverage() {
  local sample_file="$1"
  local sra_diretoria="$HOME/Projeto_Mtb/Sample_reads"
  local threads=$(nproc)
  local mosdepth_bin="$CONDA_PREFIX/bin/mosdepth"

  echo "Calculating bam coverage with mosdepth in parallel"

  while IFS= read -r sample || [[ -n $sample ]]; do
    if [ -z "$sample" ]; then
      continue
    fi

    pushd "$sra_diretoria/$sample" > /dev/null || continue
    echo "Calculating coverage for ${sample}_markdup.bam"
    echo "$bam_file" | parallel -j0 --colsep="\t" \
    "${mosdepth_bin}" --flag 3844 --mapq 20 --use-median --threads "$threads" \
    "${sample}_markdup.bam"

    echo "BAM coverage calculated for $sample"

    popd > /dev/null || continue
  done < "$sample_file"
  echo "Coverage calculation completed"
}
```

# Variant Calling

## Variant Calling with bcftools

Input: List of isolates

- Variant Calling using **'bcftools'**
  - FASTA of a reference genome: ***Mtb H37Rv***
  - BAM file per sample

Output: VCF file per sample

➢ **Variant Calling with Filters and Annotations**


➢ **Simple Variant Calling**

```
bcftools mpileup -Ou -f ~/Projeto_Mtb/NGS_helper_files/MTB_anc.fasta ~/Projeto_Mtb/Sample_reads/DRR130093/DRR130093_markdup.bam \
| bcftools call -Ov -vc > ~/Projeto_Mtb/Sample_reads/DRR130093/DRR130093.raw.vcf
```

# Variant Calling with Filters and Annotations

## 1. Creation of Intervals in the reference genome

- Using '**bedtools**'
- Output: BED file with the intervals

```
create_intervals() {
    local threads=24
    local output_file="$HOME/Projeto_Mtb/NGS_helper_files/intervals_${threads}threads.bed"

    echo "Creating equally-sized intervals file for the reference genome"
    bedtools makewindows -g ~/Projeto_Mtb/NGS_helper_files/MTB_anc.fasta.fai -n "$threads" -i winnum \
    | awk '{print $1"\t"$2+1"\t"$3}' > "$output_file"
    cat -n "$output_file"
}
```

## 2. Variant Calling using bcftools (mlineup, call, norm and annotate)

- Only reads with a quality of alignment and mapping superior to 20
- Normalization of the variants' representation
- Add important informations (e.g. Mapability, lineage, excluded regions)

```
run_variant_calling() {
    local sample="$1"
    local sra_diretoria="$HOME/Projeto_Mtb/Sample_reads"
    local output_dir="$sra_diretoria/$sample"
    local output_prefix="${sample}_bcftools_varsonly"
    local ref_file="$HOME/Projeto_Mtb/NGS_helper_files/MTB_anc.fasta"
    local threads=$(nproc)
    local intervals_file="$HOME/Projeto_Mtb/NGS_helper_files/intervals_24threads.bed"

    echo "Running variant calling for sample: $sample"
    bcftools mpileup -f "$ref_file" "$sra_diretoria/$sample/${sample}_markdup.bam" \
    --count-orphans \
    --no-BAQ --min-MQ 20 --min-BQ 20 \
    --regions-file "$intervals_file" \
    --annotate AD,ADF,ADR,DP,SP,SCR,INFO/AD,INFO/ADF,INFO/ADR,INFO/SCR \
    --threads "$threads" --output-type u \
    | bcftools call --ploidy 1 \
    --keep-alts --keep-masked-ref \
    --multiallelic-caller \
    --variants-only \
    --threads "$threads" --output-type u \
    | bcftools norm --fasta-ref "$ref_file" \
    --multiallelics - --keep-sum AD \
    --threads "$threads" --output-type v \
    | bcftools annotate \
    --annotations ~/Projeto_Mtb/NGS_helper_files/excludedloci_RLC2021_annot.tab.gz \
    --header-lines ~/Projeto_Mtb/NGS_helper_files/excludedloci_RLC2021_annot.header \
    --columns CHROM,FROM,TO,RLC_tag \
    --threads "$threads" --output-type u \
    | bcftools annotate \
    --annotations ~/Projeto_Mtb/NGS_helper_files/blindspots_mappability_marin2021_annot.tab.gz \
    --header-lines ~/Projeto_Mtb/NGS_helper_files/blindspots_mappability_marin2021_annot.header \
    --columns CHROM,FROM,TO,Mappability \
    --threads "$threads" --output-type u \
    | bcftools annotate \
    --annotations ~/Projeto_Mtb/NGS_helper_files/lineagesnps_annot.tab.gz \
    --header-lines ~/Projeto_Mtb/NGS_helper_files/lineagesnps_annot.header \
    --columns CHROM,POS,REF,ALT,Lineage_tag \
    --threads "$threads" --output-type u \
    | bcftools annotate \
    --annotations ~/Projeto_Mtb/NGS_helper_files/iedbepitopes_annot.tab.gz \
    --header-lines ~/Projeto_Mtb/NGS_helper_files/iedbepitopes_annot.header \
    --columns CHROM,POS,REF,ALT,IEDB_tag \
    --merge-logic IEDB_tag:unique \
    --threads "$threads" --output-type v \
    | bcftools annotate --set-id +'%CHROM:%POS' \
    | bgzip > "$output_dir/${output_prefix}_annotated.vcf.gz"
    tabix -p vcf "$output_dir/${output_prefix}_annotated.vcf.gz"
}
```

# Variant Calling – First Results

## Simple Variant Calling

Total number of variants listed: 1571

| #CHROM | POS | **ID** | REF | ALT | **QUAL** | FILTER | INFO | FORMAT | DRR130093.bam |
|---|---|---|---|---|---|---|---|---|---|
| MTB_anc | 39158 | **.** | C | G | **225.007** | . | DP=103;VDB=0.0672406;SGB=-0.693147;MQSB=1;MQ0F=0;AF1=1;AC1=1;DP4=0,0,55,26;MQ=60;FQ=-999 | GT:PL | 05:15,0 |

## Version with filter and annotations

Total number of variants listed: 1645

| #CHROM | POS | **ID** | REF | ALT | **QUAL** | FILTER | INFO | FORMAT | DRR130093.bam |
|---|---|---|---|---|---|---|---|---|---|
| MTB_anc | 39158 | **MTB_anc:39158** | C | G | **225** | . | DP=105;ADF=0,56;ADR=0,29;AD=0,85;SCR=19;VDB=0.0479958;SGB=-0.693147;MQSB=1;MQ0F=0;AC=1;AN=1;DP4=0,0,56,29;MQ=60;**Mappability=1;Lineage_tag=!lineage,2,tbprofiler** | GT:PL:**DP:SP:ADF:ADR:AD:SCR** | **1:255,0:85:0:0,56:0,29:0, 85:19** |

# Future work

**Pipeline Optimization**

- Additional **filtering** in pre-processing
- Test **other parameters** in variant calling
- **Integrate** variant calls from **different tools** to increase coverage

**Filtering and selection of variants**

**Task 2: Correlation between the variants and the disease severity**

- Random Forests
- Logistic Regression

ICVS Life and Health Sciences Research Institute

INSTITUTO DE INVESTIGAÇÃO E INOVAÇÃO EM SAÚDE
UNIVERSIDADE DO PORTO

Universidade do Minho
Escola de Engenharia

# Identification of *Mycobacterium tuberculosis* Genetic Determinants of Disease Severity

**Rita Nóbrega Amaral Martins**
PG46733

Work supervised by Dr. **Nuno Ósório** and Dr. **Tiago Beites**

Project Presentation
Master's in Bioinformatics, University of Minho

29th May 2024