## **1) (MSigDB ∩ DE) + NCBI + tagovanje**

**Skripta 1:** step1\_overlap\_ncbi\_tag.py   
**Ulazi:**

* B\_cell\_sample\_1.csv
* 6 × Hallmark .gmt fajlova

**Izlazi:**

* B\_cell\_sample\_1\_ImmuneMeta\_overlap.csv
* B\_cell\_sample\_1\_ImmuneMeta\_counts\_by\_set.csv
* B\_cell\_sample\_1\_Tagged\_All.csv ← sa kolonama molecule\_class, ncbi\_summary\_short itd.

Folder gde se nalazi B\_cell\_sample\_1.csv je:

D:\1 Rad\BioINGLab\scRNA-Seq\Marko Genes Analyses\MSigDB\B Cell v2

HALLMARKS se nalaze u folderu:

D:\1 Rad\BioINGLab\scRNA-Seq\Marko Genes Analyses\MSigDB\B Cell v2\HALLMARKS

Želim da mi se prvi rezultat ovog poređenja nađe u folderu:

D:\1 Rad\BioINGLab\scRNA-Seq\Marko Genes Analyses\MSigDB\B Cell v2

Za + NCBI + tagovanje koristi moju mail adresu: [marko.zivanovic@uni.kg.ac.rs](mailto:marko.zivanovic@uni.kg.ac.rs)

Kada se dobije fajl B\_cell\_sample\_1\_ImmuneMeta\_overlap.csv u koloni „molecule\_class“ dobićemo razne klase, među kojima verovatno najviše „other/immune“.

Ovo sređujem automatski pomoću **skripte 2** (subclassification.py), tako što sve tekstove iz kolone „ncbi\_summary“ prepišem u note (Gene description.txt), koji se nalazi u folderu: D:\1 Rad\BioINGLab\scRNA-Seq\Marko Genes Analyses\MSigDB\B Cell v2\Subclassification Na ovaj način šaljem GPT da mi dodeli podklasifikacije.

Ovako dobijem dokument: B\_cell\_sample\_1\_ImmuneMeta\_overlap\_subclassed.csv

**Rezultat** ovog prvog koraka je spisak gena koji se poklapa sa Hallmarks i kojima je dodeljena precizna klasa i podklasa.

Sada je sve spremno za STRING

## **2) STRING**

**Skripta 3:** string.py

Ulazni fajl je B\_cell\_sample\_1\_Tagged\_All.csv iz foldera B Cell v2.

Drugi ulazni fajl je B\_cell\_sample\_1\_ImmuneMeta\_overlap\_subclassed.csv da bi klasifikaciju prepisao odatle.

Dakle, string.py će:

1. **Učitati gene** iz B\_cell\_sample\_1\_Tagged\_All.csv (kolona gene).
2. **Učitati podklasifikacije** iz Subclassification\B\_cell\_sample\_1\_ImmuneMeta\_overlap\_subclassed.csv (kolone gene, molecule\_subclass).
3. Poslati listu gena STRING API-ju (Homo sapiens).
4. Sačuvati rezultate u folderu **STRING**:
   * Bcell\_STRING\_edges.tsv – interakcije između gena.
   * Bcell\_STRING\_nodes.tsv – čvorovi sa STRING atributima.
   * Bcell\_STRING\_nodes\_annot.tsv – čvorovi + molecule\_subclass iz tvoje podklasifikacije.

📌 **Šta dobijaš kad pokreneš skriptu:**

1. **Bcell\_STRING\_edges.tsv** – parovi gena koji interaguju + STRING score.
2. **Bcell\_STRING\_nodes.tsv** – lista svih gena koji su u mreži.
3. **Bcell\_STRING\_nodes\_annot.tsv** – isto kao gore, ali sa tvojim molecule\_subclass oznakama (npr. transcription factor, cytokine, receptor...).

📍 **Ulazni fajl:**  
B\_cell\_sample\_1\_ImmuneMeta\_overlap\_subclassed.csv iz foldera **Subclassification**.

📍 **Izlazni folder:**  
D:\1 Rad\BioINGLab\scRNA-Seq\Marko Genes Analyses\MSigDB\B Cell v2\STRING

Slika na kojoj se nalazi krug, linija, dijagram

Sadržaj generisan AI-jem može biti netačan.

## **3) Cytoscape**

Folder Cytoscape

Kada uradim Cytoscape kreiranje mreže, onda vidim koji su mi geni pouzdani.

**Prikaz “snage” interakcije debljinom linija**

- **Uvoz mreže**

* *File → Import → Network from File…*
* Izaberi B\_cell\_sample\_1\_STRING\_edges.tsv.
* Postavi **preferredName\_A** kao *Source Node*, **preferredName\_B** kao *Target Node*.
* Klikni **OK**.

- **Provera kolone sa snagom**

* U *Edge Table* proveri da postoji kolona score (0–1).

- **Mapiranje debljine linije**

* Otvori *Style* panel → izaberi **Edges**.
* Kod svojstva **Width** klikni ikonicu mape → izaberi **score** i *Continuous Mapping*.
* Dvaput klikni na “Current Mapping” da otvoriš **Continuous Mapping Editor**.
* Postavi:
  + **Min Value** = 0.50 → **Edge Width** = 0.5
  + **Max Value** = 0.95 → **Edge Width** = 3.0
* Klikni **OK**.

- **Rezultat**

* Linije su sada proporcionalno deblje za jače interakcije i tanje za slabije.

Export png iz Cytoscape podesi tako da je širina 2000 px.

Slika na kojoj se nalazi dijagram, linija, snimak ekrana, dizajn

Sadržaj generisan AI-jem može biti netačan.

## **4) Bar Plot Up-DownRegulation**

Gene koji su ispod skora 0.7, tj. gene koje ne vidim u Cytoscape obrišem, tako što fajl B\_cell\_sample\_1\_ImmuneMeta\_overlap.csv iskopiram u Folder „Cytoscape“ i imenujem ga u „B\_cell\_sample\_1\_ImmuneMeta\_overlap\_STRING.csv“ i u njemu obrišem ručno gene koji mi nedostaju u string-u i u cytoscape-u.

Slika na kojoj se nalazi tekst, Font, snimak ekrana, dijagram

Sadržaj generisan AI-jem može biti netačan.

## **5) Pathway Enrichment**

* Koristi **Enrichr**, **g:Profiler**, **Reactome**, ili **KEGG**:
  + unese selektovane gene
    - B\_cell\_sample\_1\_ImmuneMeta\_overlap\_STRING.csv
  + proveri koje signalne i metaboličke putanje su najviše obogaćene.

U folderu „Pathway Enrichment“ je csv fajl B\_cell\_sample\_1\_ImmuneMeta\_overlap\_STRING.csv (iskopiran iz foldera Cytoscape) iz kojeg sam obrisao **3 gena** koji su ispali iz **STRING/Cytoscape mreže** zbog niskog poverenja,  
jer:

1. **Povećavamo tačnost** – u enrichment analizi će ostati samo geni za koje znamo da su funkcionalno povezani.
2. **Izbegavamo “šum”** – nepovezani geni mogu veštački razvodniti statistiku i smanjiti značajnost pravih puteva.
3. **Lakše tumačenje** – dobijeni putevi će biti u skladu sa mrežom interakcija, što će kasnije olakšati povezivanje vizuelizacija.

Praktično, sada ću imati **11 gena** kao ulaz za **Pathway Enrichment**.

Kreirao sam novu skriptu: enrichment\_bcell\_genes.py

Dobio sam rezultat: B\_cell\_sample\_1\_PathwayEnrichment\_Enrichr.csv

Za dobijanje:

* Identifikaciju ključnih puteva → vraća listu top Reactome, KEGG, GO procesa.
* Klasifikaciju procesa → organizuje rezultate po tipu (signalni putevi, metabolički, imunološki). Rangiranje po statističkoj značajnosti → automatski dobijaš p-value i FDR.
* Povezivanje sa literaturom i bolestima → možemo uključiti i baze kao što su Disease Ontology i Human Phenotype Ontology.
* Bolju vizuelizaciju → g:Profiler može da vrati podatke koje lako pretvaramo u bubble plot ili network.

Koristio sam skriptu: gprofiler\_enrichment\_allinone.py

Dobio sam rezultat:

Bcell\_genes\_gprofiler\_all.csv

* svi termini sa FDR‑om (adj\_p\_value), brojem pogodaka i klasifikacijom (Category).

Bcell\_genes\_gprofiler\_results.xlsx

* u jednom fajlu:

 Input\_genes (geni, sa logFC ako postoji),

 Enrichment\_all (kompletni rezultati, sortirani po FDR),

 Top\_by\_source (Top10 po svakoj bazi),

 GeneTerm\_matrix (1/0 matrica pogodna za dodatne ploter-e).

DEBUG\_gprofiler\_raw\_columns.csv

Bcell\_genes\_gprofiler\_bubble.png

Bcell\_genes\_TermGene\_edges.tsv

Bcell\_genes\_TermTerm\_overlap.tsv

Onda sam kreirao skriptu: make\_enrichment\_summary.py

Za dalju obradu rezultata i dobio sledeće:

Bcell\_genes\_enrichment\_summary.csv

Bcell\_genes\_enrichment\_summary.xlsx

FigA\_Bcell\_bar\_logFC.png

FigB\_Bcell\_bubble\_Top5.png

FigC\_Bcell\_term\_gene\_network\_Top5.png

Slika na kojoj se nalazi tekst, snimak ekrana, Font

Sadržaj generisan AI-jem može biti netačan.

Slika na kojoj se nalazi tekst, snimak ekrana, dijagram, linija

Sadržaj generisan AI-jem može biti netačan.

Slika na kojoj se nalazi tekst, snimak ekrana, dijagram

Sadržaj generisan AI-jem može biti netačan.

Slika na kojoj se nalazi linija, dijagram, snimak ekrana

Sadržaj generisan AI-jem može biti netačan.