

Identificazione ed Analisi di Biomarcatori dai Dati di Espressione Genica e Proteica

Corso di laurea in Informatica
Valerio Mesiti

Anno Accademico 2022-2023



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'informazione, Informatica e Statistica
Dipartimento di Informatica

Relatore: Prof. Maurizio Mancini
Correlatore: Prof. Enrico Tronci

Un **espressione** è una forma di comunicazione che consente di esprimere idee, valori o calcoli



Il nostro DNA funziona come un **archivio**, contiene non solo le istruzioni genetiche derivate dalle generazioni passate, ma anche le espressioni delle nostre **funzioni vitali**.

Indice

1. Introduzione e Contestualizzazione



- Introduzione e Contestualizzazione
- Revisione e Background
- Metodologia ed Implementazione
- Risultati e Conclusioni

Panoramica Generale

1. Introduzione e Contestualizzazione



- **Contesto e Motivazione:** dati di espressione genica, forniscono un'opportunità senza precedenti per acquisire una visione dettagliata delle basi biologiche delle malattie.
- **Obiettivo della ricerca:** individuare segnali biologici distintivi e rilevanti per la diagnosi e il trattamento delle malattie oncologiche.

Obiettivo della Ricerca

1. Introduzione e Contestualizzazione



Il progetto si è posto l'obiettivo di **esplorare** il ricco terreno dei dati di espressione genica e proteica disponibili in rete al fine di creare una base di dati per identificare ed analizzare biomarcatori associati a malattie specifiche tramite l'uso di **alberi decisionali**. Sono state realizzate quindi una **base di dati** con il rispettivo script di **download e popolazione** ed uno script per formare alberi decisionali a partire dai dati scaricati.

Nella ricerca, infatti, sono state affrontate **sfide** come la raccolta e la gestione di grandi quantità di dati che richiedono **un'organizzazione attenta** e l'implementazione di **standard di qualità**. Inoltre, le analisi dei dati biomedici possono essere complesse, richiedendo l'uso di algoritmi avanzati e risorse di calcolo.



Indice

2. Revisione e Background

- Introduzione e Contestualizzazione
- Revisione e Background
- Metodologia ed Implementazione
- Risultati e Conclusioni

Importanza dei Biomarcatori

2. Revisione e Background



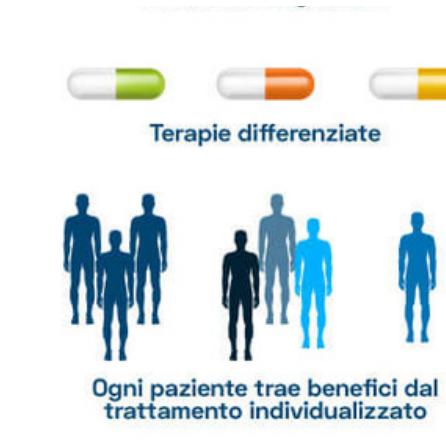
L'identificazione dei biomarcatori correlati alle malattie svolge un ruolo cruciale in una serie di contesti clinici e di ricerca:



Diagnosi



Prognosi



Terapia Personalizzata



Ricerca

GDC

2. Revisione e Background



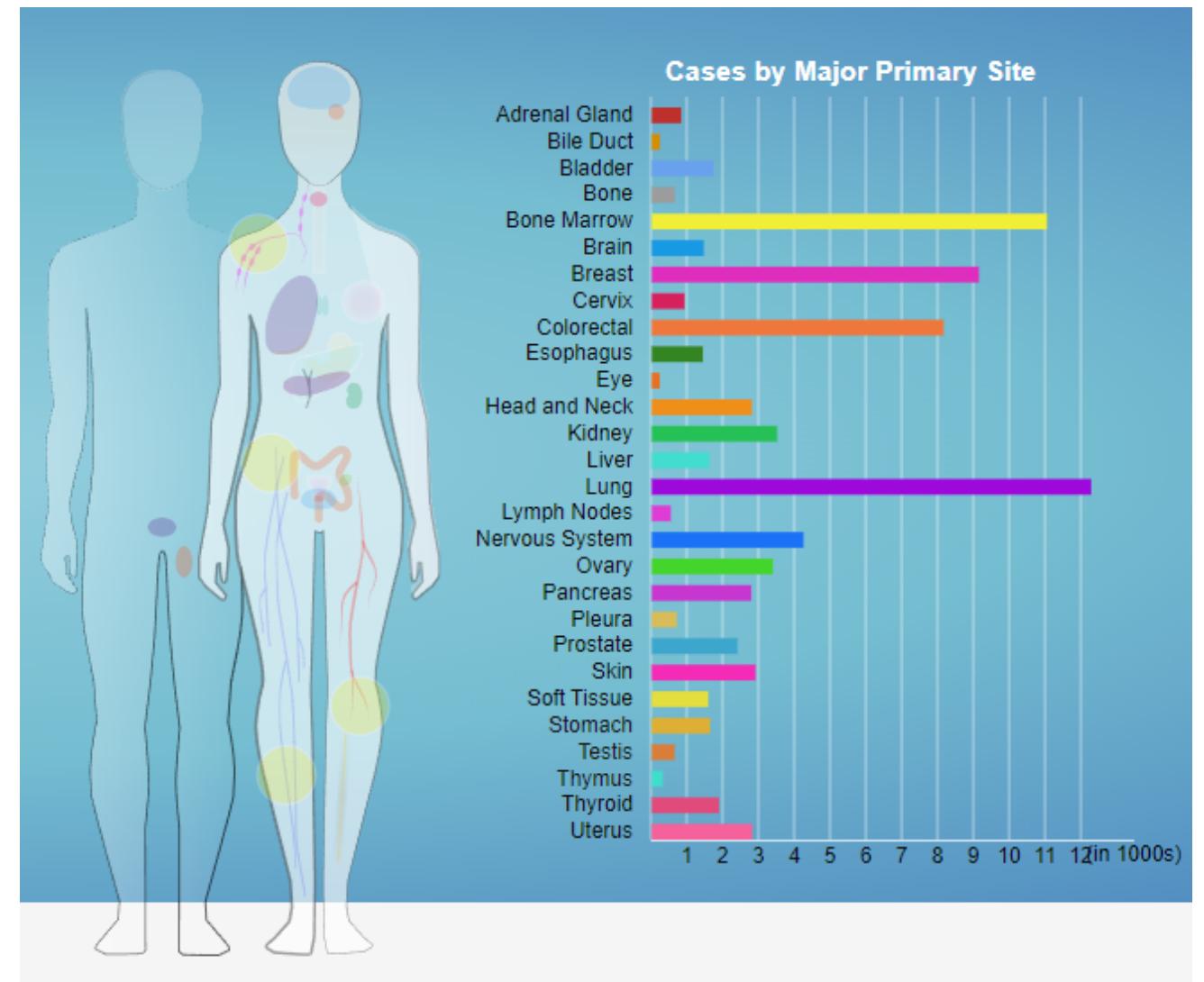
Il Genomic Data Commons rappresenta una fonte inestimabile di informazioni genetiche, inclusi dati di **espressione genica, sequenziamento genomico, copy number variation (CNV) e molto altro**. La nostra analisi si concentra su come raccogliere, organizzare e sfruttare questi dati in modo efficiente per scopi di ricerca scientifica.





2. Revisione e Background

In questo contesto, questa tesi si propone di utilizzare i dati di espressione genica e proteica dal **GDC** per creare una base di dati utile ad identificare biomarcatori correlati a malattie specifiche, quali i **68 siti** più comuni dove si può contrarre un **tumore**, contribuendo così ad un aiuto per future analisi in merito.



2. Revisione e Background



Tutta la ricerca ruota attorno ad un componente principale, delle **analisi** archiviate in file .tsv detti **Expression File**. Un File di Espressione è un insieme strutturato di dati che rappresenta le quantità di espressione genica o proteica di specifici geni in **campioni biologici**.

File Properties		Data Information	
Name	874e5c90-f93c-4295-a9ba-4e35172f91d6.ma_seq.augmented_star_gene_counts.tsv	Data Category	Transcriptome Profiling
Access	open	Data Type	Gene Expression Quantification
UUID	56455c13-a5bb-46c3-9e84-e90905ed597b	Experimental Strategy	RNA-Seq
Data Format	TSV	Platform	--
Size	4.24 MB		
MD5 Checksum	c054d9abcf17c21925a6af0ad53f3234		
Archive	--		
Project	CGCI-HTMGP-CC		

Showing 1 - 1 of 1 associated cases/biospecimen

Associated Cases/Biospecimen				
Entity ID	Entity Type	Sample Type	Case UUID	Annotations
HTMCP-03-06-02266-01A-01R-9006	aliquot	Primary Tumor	37941fa0-0167-4bfb-a609-29521ee854d8	0

Show 10 entries

Analysis		Reference Genome	
Analysis ID	fcd36eac-f0b3-49ec-b900-9dbd276b5499	Genome Build	GRCh38.p0
Workflow Type	STAR - Counts	Genome Name	GRCh38.d1.vd1
Workflow Completion Date	2021-12-22		
Source Files	1		

File Properties		Data Information	
Name	TCGA-10-0925-01A-11-20_RPPA_data.tsv	Data Category	Proteome Profiling
Access	open	Data Type	Protein Expression Quantification
UUID	e1cb5dfb-d999-4676-8c71-fc1ec92d1e5a	Experimental Strategy	Reverse Phase Protein Array
Data Format	TSV	Platform	RPPA
Size	24.05 KB		
MD5 Checksum	c6ff36371cef9fb6d15f335560f49ff0		
Archive	--		
Project	TCGA-QV		

Showing 1 - 1 of 1 associated cases/biospecimen

Associated Cases/Biospecimen				
Entity ID	Entity Type	Sample Type	Case UUID	Annotations
TCGA-10-0925-01A-11	portion	Primary Tumor	869c9aae-85eb-4faa-8893-b3f487e8981c	0

Show 10 entries

File Versions				
Version	File UUID	Release Date	Release Number	
1	e1cb5dfb-d999-4676-8c71-fc1ec92d1e5a	Current Version	2021-09-23	30.0

2. Revisione e Background



Questa è la parte centrale del file e consiste in una tabella in cui le righe rappresentano geni e proteine specifiche e le colonne contengono le quantità di espressione (ad esempio, **TPM** - Transcripts Per Million, **FPKM** - Fragments Per Kilobase per Million) associate a ciascun gene o proteina nei campioni.

gene_id	gene_name	gene_type	unstranded	stranded_first	stranded_second	tpm_unstranded	fpkm_uq_unstranded
ENSG00000000003.15	TSPAN6	protein_coding	6082	2	6080	55.4834	16.1430
ENSG00000000005.6	TNMD	protein_coding	5	0	5	0.1402	0.0408
ENSG00000000419.13	DPM1	protein_coding	1543	21	1522	52.8991	15.3911
ENSG00000000457.14	SCYL3	protein_coding	2698	1583	2449	16.2281	4.7193
ENSG00000000460.17	C1orf112	protein_coding	2244	619	2994	15.5539	4.5254
ENSG00000000938.13	FGR	protein_coding	783	2	781	9.5803	2.7874
ENSG00000000971.16	CFH	protein_coding	13922	5	13919	72.1919	21.0044
ENSG00000001036.14	FUCA2	protein_coding	3723	37	6368	54.5916	15.8836
ENSG00000001084.13	GCLC	protein_coding	6633	1	7821	31.8488	9.2665
ENSG00000001167.14	NFYA	protein_coding	4624	5	5184	50.2075	14.6080
ENSG00000001460.18	STPG1	protein_coding	1661	41	1658	8.0757	2.3496
ENSG00000001461.17	NIPAL3	protein_coding	7571	8	7599	33.3426	9.7011
ENSG00000001497.18	LAS1L	protein_coding	4857	6	4885	16.0081	4.6576
ENSG00000001561.7	ENP4	protein_coding	2090	1	2089	18.6227	5.4183
ENSG00000001617.12	SEMA3F	protein_coding	22324	9	22347	191.4142	55.6925
ENSG00000001626.16	CFTR	protein_coding	509	19	500	2.1185	0.6164
ENSG00000001629.10	ANKK1B1	protein_coding	4510	7	4520	25.2569	7.3486
ENSG00000001630.17	CYP51A1	protein_coding	48	1	49	0.5628	0.1638
ENSG00000001631.16	KRIT1	protein_coding	544	15	544	3.6284	1.0557
ENSG00000002016.18	RAD52	protein_coding	1290	168	1130	11.8543	3.4491
ENSG00000002330.14	BAD	protein_coding	598	48	2145	14.4878	4.2153
ENSG00000002549.12	LAP3	protein_coding	3268	1	3267	35.7277	10.3951
ENSG00000002586.20	C99	protein_coding	16559	16	16547	141.0477	41.0382
ENSG00000002587.10	HS3ST1	protein_coding	270	7	263	1.5027	0.4372
ENSG00000002726.21	AOC1	protein_coding	7641	65	7576	83.5140	24.2986
ENSG00000002745.13	WNT16	protein_coding	33	11	22	0.4184	0.1217
ENSG00000002746.15	HECW1	protein_coding	109	5	113	0.3237	0.0942
ENSG00000002822.15	MAD1L1	protein_coding	37	0	37	0.2118	0.0616
ENSG00000002834.18	LASP1	protein_coding	23157	15	27179	135.7080	39.4846
ENSG00000002919.15	SNX11	protein_coding	2646	43	2606	26.4919	7.7079
ENSG00000002933.9	TMEM176A	protein_coding	3513	72	3486	41.8083	12.1642
ENSG00000003056.8	M6PR	protein_coding	6722	163	8155	77.5888	22.5747

AGID	lab_id	catalog_number	set_id	peptide_target	protein_expression
AGID00100	882	sc-628	Old	1433BETA	0.042929
AGID00100	882	sc-628	Old	1433BETA	0.042929
AGID00111	913	sc-23957	Old	1433EPSILON	0.085853
AGID00101	883	sc-1019	Old	1433ZETA	0.12268
AGID00001	2	9452	Old	4EBP1	0.011953
AGID00002	3	9456	Old	4EBP1_pS65	-0.12592
AGID00003	6	9459	Old	4EBP1_pT37T46	-0.3184
AGID00443	8	9455	Old	4EBP1_pT70	-0.08528
AGID00120	985	4937	Old	53BP1	-0.12767
AGID00004	13	3661	Old	ACC_pS79	-0.25097
AGID00005	14	1768-1/ab45174	Old	ACC1	-0.52794
AGID00408	2372	3658	Set164	AceCS1	-0.710064738339178
AGID00473	1182	5335	Old	ACETYLATUBULINLYS40	0.78307
AGID00404	2367	9189	Set164	ACSL1	-0.865744418773428
AGID02144	2450	PA5-27081	Old	ACVRL1	0.024425
AGID00186	1198	ab88574	Old	ADAR1	-0.4196
AGID00146	1084	4691	Old	AKT	-0.53732
AGID00028	230	9271	Old	AKT_pS473	-1.2893
AGID00170	1154	2965	Old	AKT_pT308	-0.71388
AGID00316	1800	3063	Set164	Akt2	-0.275718718864172
AGID00347	2009	8599	Set164	Akt2_pS474	1.27083863155985



Indice

3. Metodologia ed Implementazione

- Introduzione e Contestualizzazione
- Revisione e Background
- Metodologia ed Implementazione
- Risultati e Conclusioni

Creazione della Base di Dati



3. Metodologia ed Implementazione

Richiesta dei Requisiti

Si vuole realizzare una base di dati per analizzare la mole di dati derivanti dai maggiori siti di biomedicina riguardanti la “**expression**”.

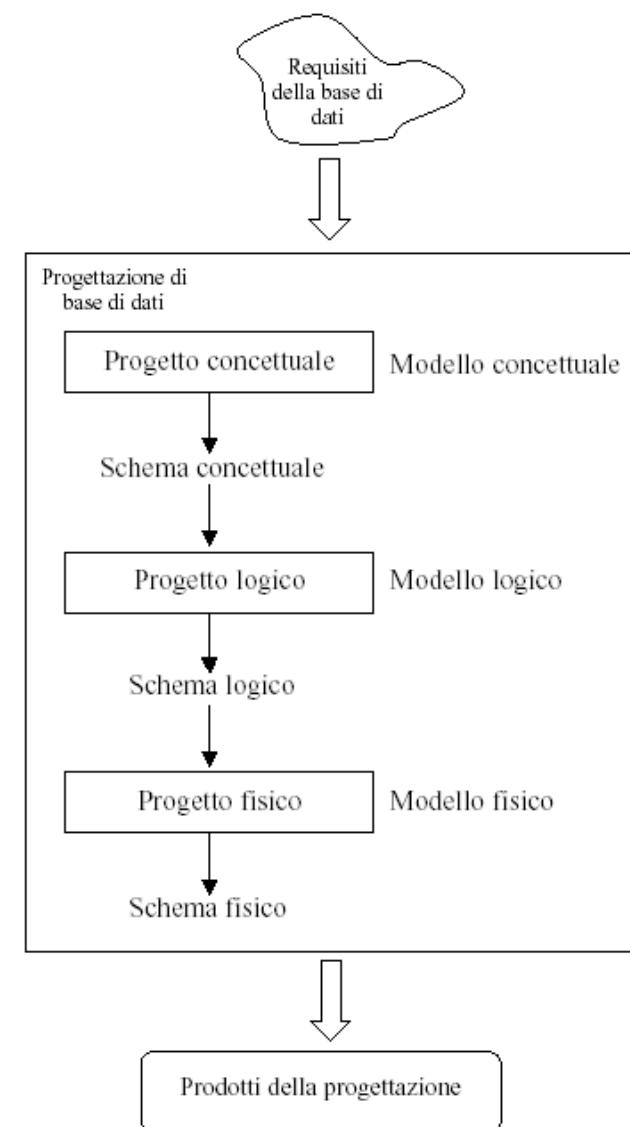
Tutto il database è diviso in **progetti** che hanno un codice univoco ed un nome, i progetti contengono tutti le **analisi** sotto forma di file e i casi che ne fanno parte.

Di un caso sappiamo l'id, il **tipo di malattia** ed il sito primario dove risiede la malattia. Da questo paziente scaturiscono più **campioni biologici**. Questi prelievi primari, chiamati **sample**, possono essere trattati e resi **portion**, quest'ultime possono diventare **analyte** che infine con determinati trattamenti diventano **aliquote**.

Per i campioni identificati da un id, vogliamo sapere il **tipo del campione** e di che **tumore** si tratta.

Rappresentiamo quindi l'analisi sotto forma di file, dove è contenuto un determinato insieme di dati di espressione presi da **uno o più biospecie** e per ognuno di essi da chi proviene. Le analisi hanno: un codice, un nome, la **categoria dei dati**, il peso del file, la data di creazione del file, la data di ultimo aggiornamento del file, il **tipo di dati** e la **strategia sperimentale**.

Per le due espressioni, contenute in più file e identificate da un codice univoco, si vuole memorizzare il nome e le misurazioni che ne derivano.



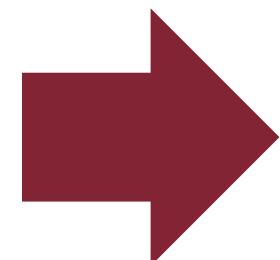
Progettazione Concettuale

3. Metodologia ed Implementazione



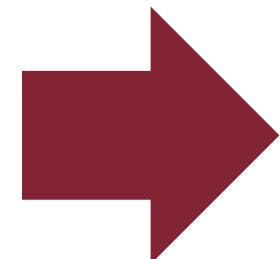
La progettazione concettuale è il **fondamento** su cui verranno costruite le fasi successive

- Glossario dei termini



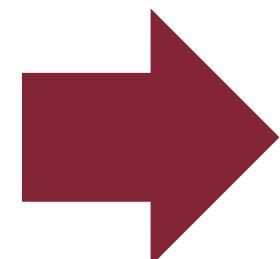
Un elenco di **definizioni** chiare e concise dei principali termini e concetti utilizzati nel contesto del database biomedico.

- Dizionario dei dati

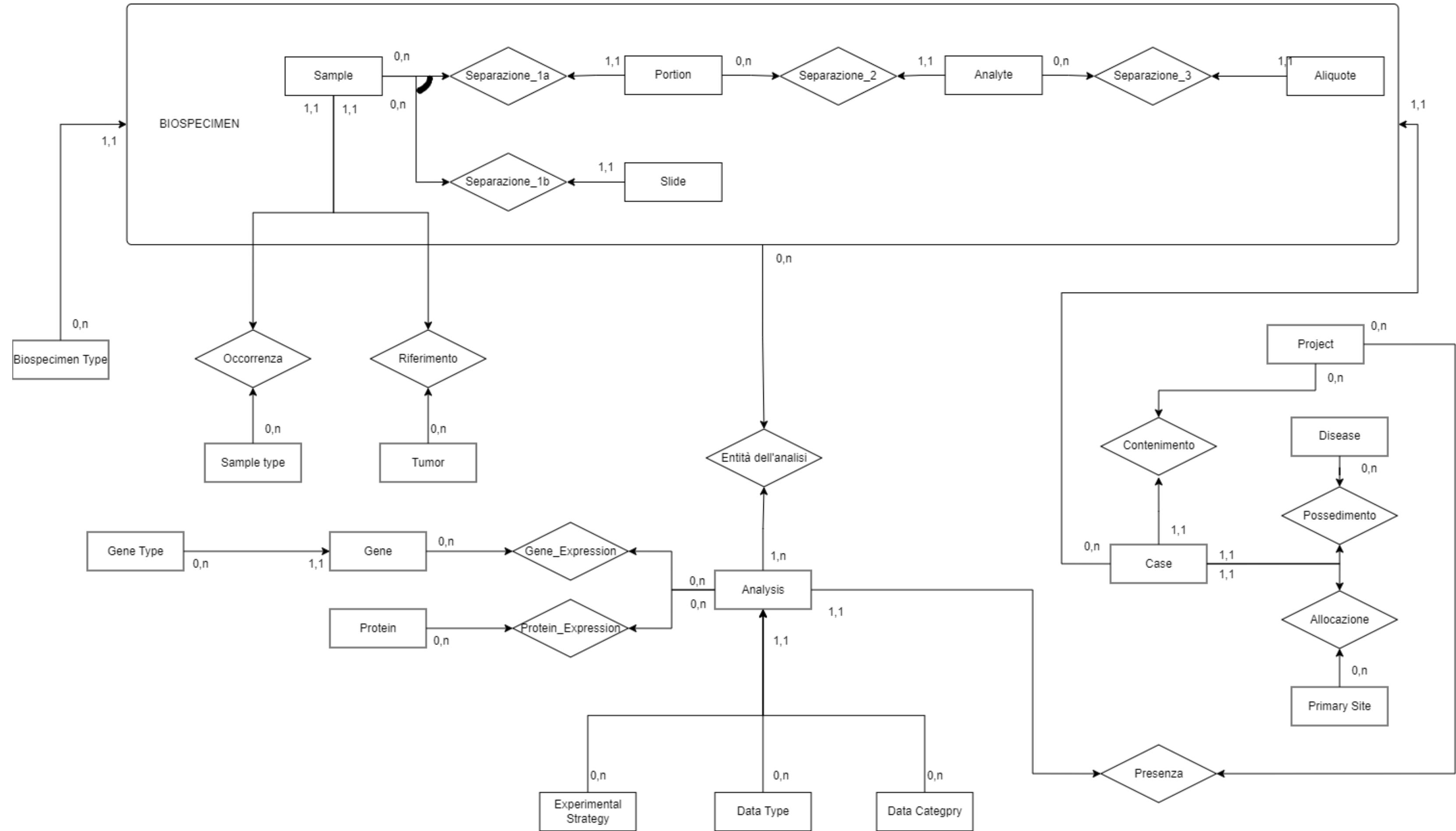


Un documento o una risorsa che elenca e descrive in dettaglio le diverse **entità, tabelle o categorie di dati** presenti nel sistema.

- Tavola dei volumi



Una risorsa che fornisce una panoramica delle **dimensioni dei dati** memorizzati nel sistema.

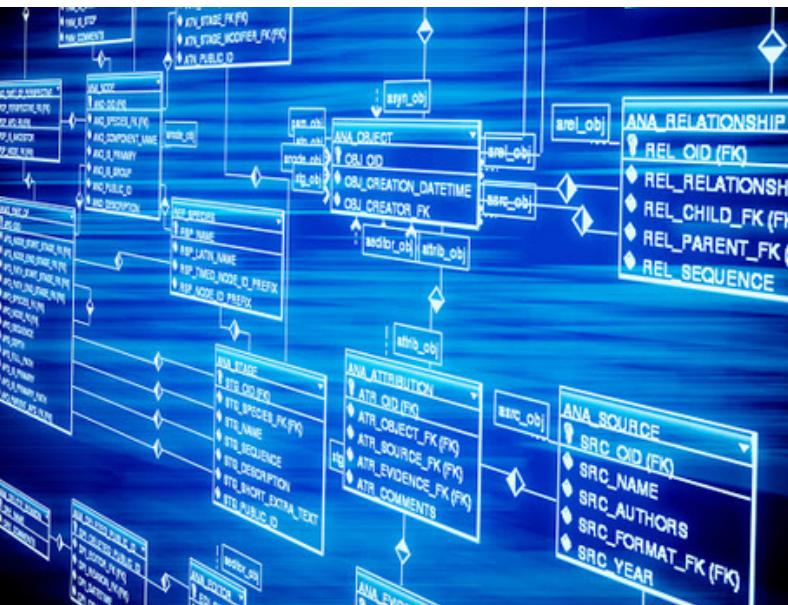




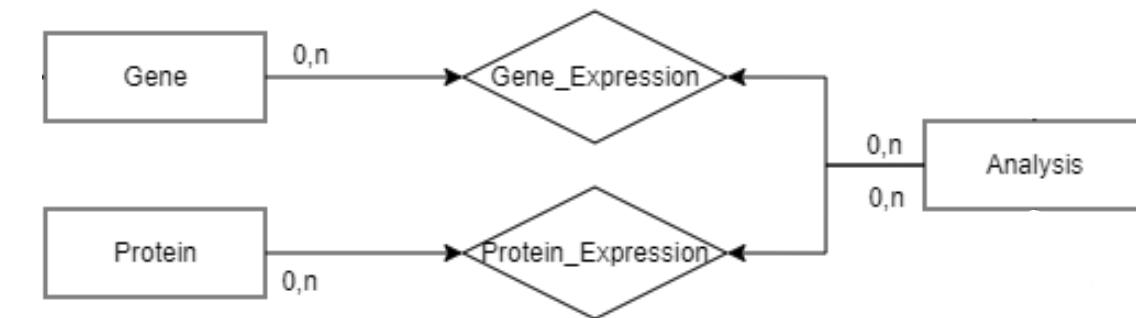
Progettazione Logica

3. Metodologia ed Implementazione

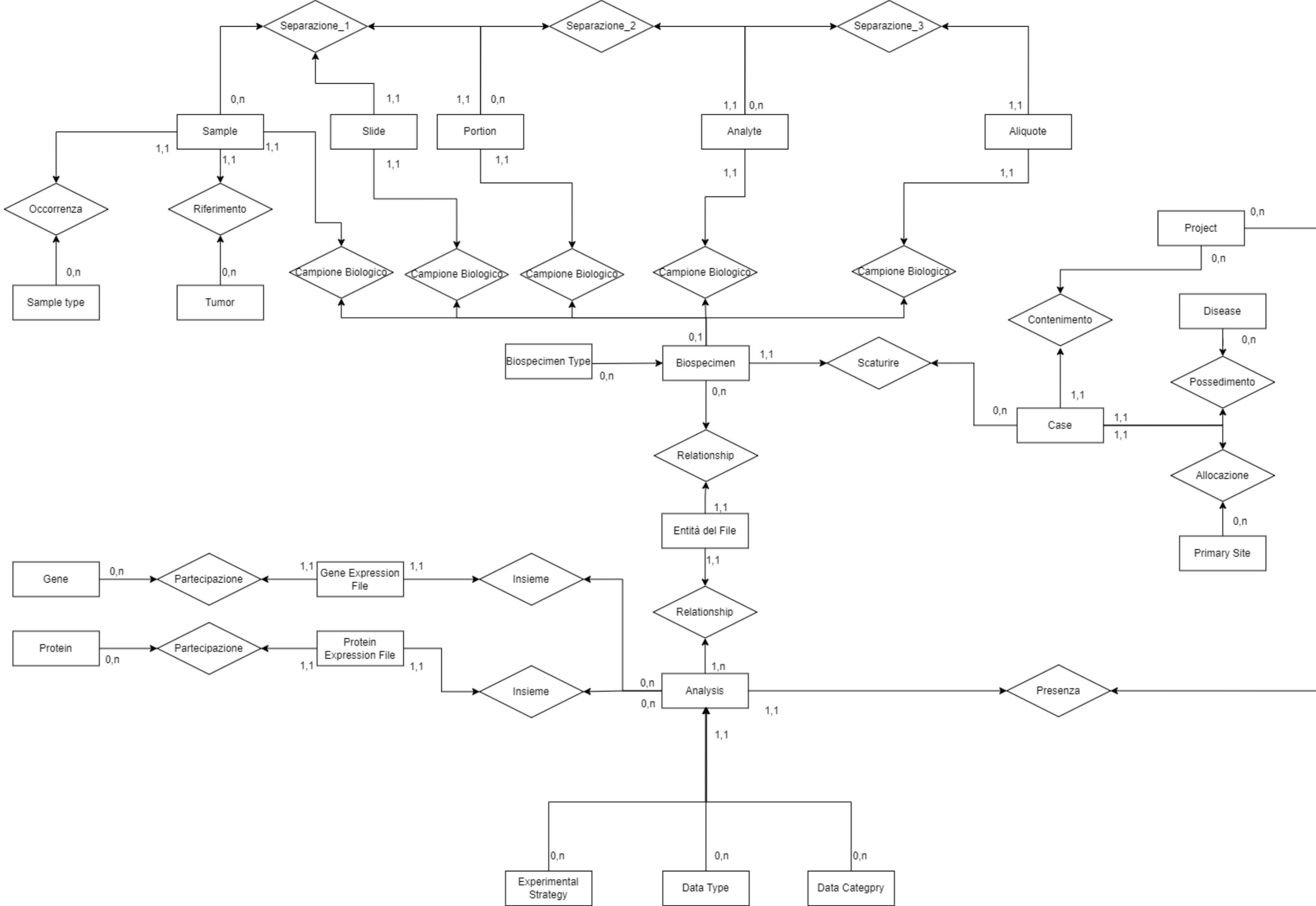
L'obiettivo principale di questa ristrutturazione è semplificare il modello dei dati, **eliminare ridondanze, migliorare le prestazioni e garantire una migliore gestibilità del database.**



Creazione di Vincoli



Risoluzione relazioni

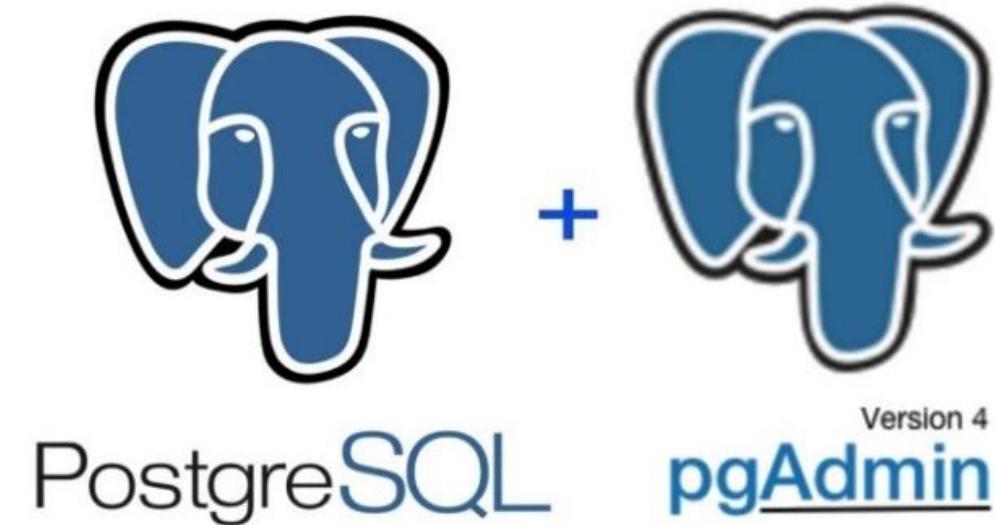


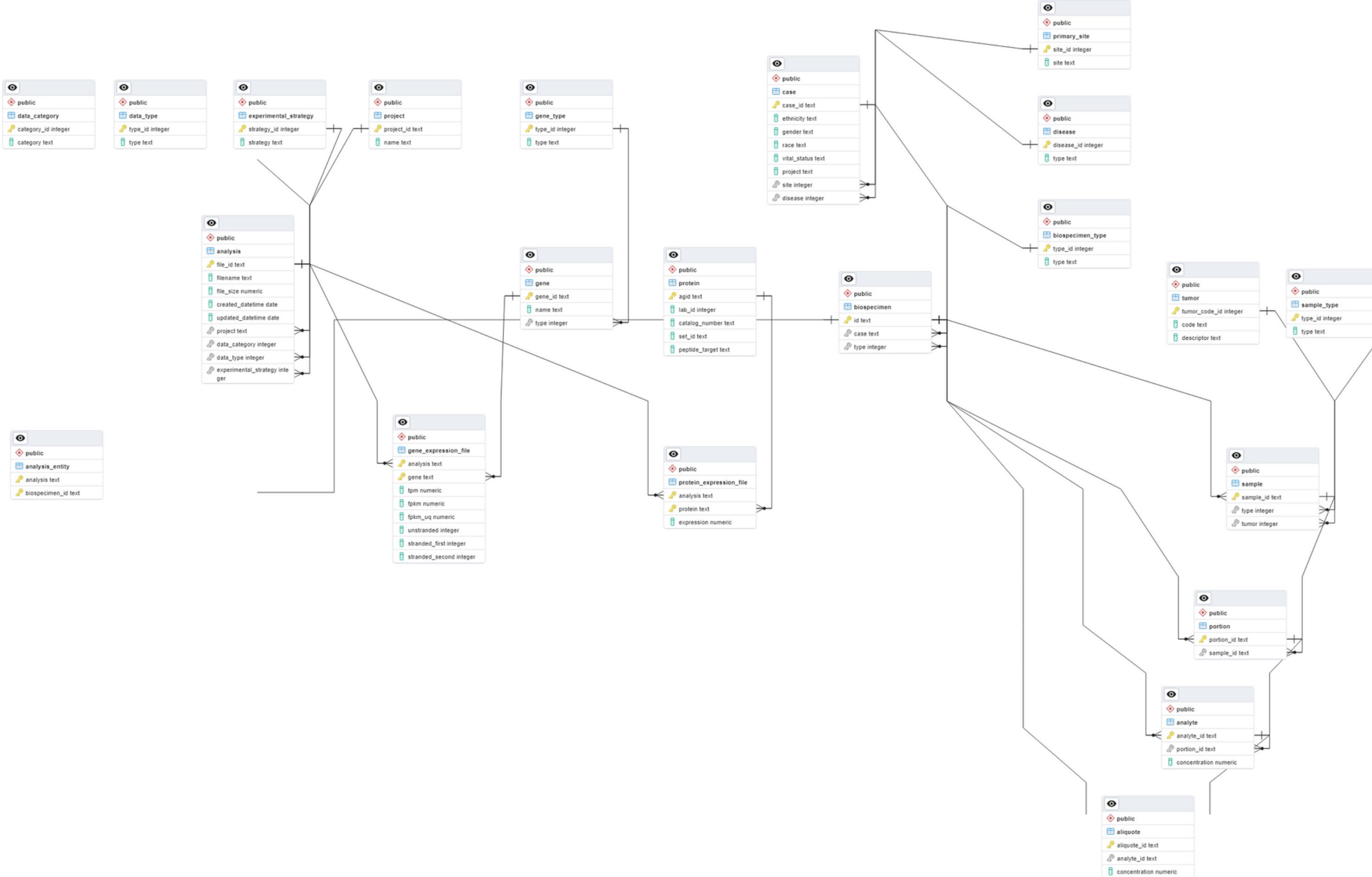
Progettazione Fisica

3. Metodologia ed Implementazione



Questo schema dati è stato strutturato per fornire **flessibilità e scalabilità** in modo che ulteriori informazioni possano essere **aggiunte in futuro** senza dover ridisegnare completamente il database.







Downloading e Gestione Dati

3. Metodologia ed Implementazione

È stata implementata una procedura per scaricare e gestire i dati relativi all'espressione dalla piattaforma GDC. Sono stati creati script e utilizzate API per acquisire questi dati. Illustriamo alcuni componenti chiave:

Connessione al Database

```
# Crea una connessione al database PostgreSQL
connection = psycopg2.connect(**db_params)

# Crea un cursore per eseguire query SQL
cursor = connection.cursor()

# Inizia la transazione
connection.autocommit = False
```

Un esempio di inserimento di Expression File

```
if type_id == 1:
    for data_row in expression_data:
        # Inserimento dei dati di espressione genica nel database
        gene_id = data_row["gene_id"]
        stranded_first = data_row["stranded_first"]
        stranded_second = data_row["stranded_second"]

        if stranded_first != 0 and stranded_second != 0:
            cursor.execute(inserisci_espressione_genica, (file_id, gene_id,
data_row["tpm_unstranded"], data_row["fpkm_unstranded"],
data_row["fpkm_uq_unstranded"], data_row["unstranded"], stranded_first,
stranded_second))
            connection.commit()
```



Downloading e Gestione Dati

3. Metodologia ed Implementazione

Nel codice sono incluse diverse funzioni ausiliarie per la gestione di **progetti, casi, campioni e dati di espressione**. Queste funzioni consentono di inserire dati di supporto nel database.

Sono stati anche pensati e realizzate: funzioni che: **gestiscono gli errori, evitano l'inserimento di duplicati e filtrano** selezionando solo i file desiderati

```
# Funzione per inserire un nuovo progetto nel database
def project(id, cursor):
    project_url = "https://api.gdc.cancer.gov/projects/" + id
    inserisci_progetto = "INSERT INTO public.project VALUES (%s, %s) ON CONFLICT
(project_id) DO NOTHING;"

    params = {
        #Puoi aggiungere altri campi che danno più info relative al progetto
        "fields": "name",
        "format": "JSON",
        "pretty": "true"
    }

    response = requests.get(project_url, params=params)

    if response.status_code == 200:
        data = json.loads(response.content.decode("utf-8"))["data"]

        cursor.execute(inserisci_progetto, (id, data["name"]))
        print("Progetto inserito nel database")
    else:
        print(f"Errore durante il download del progetto: {response.status_code}")
    return []
```

Alberi Decisionali

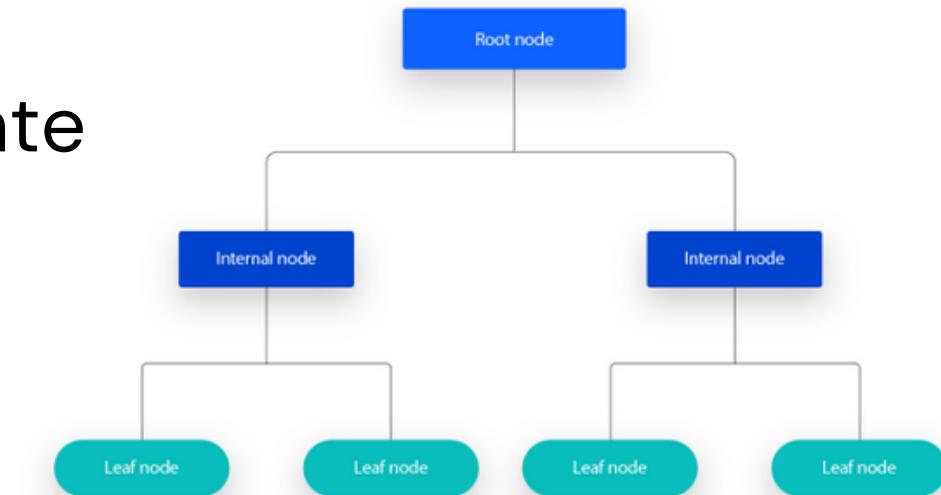
3. Metodologia ed Implementazione



Un albero decisionale è un modello di apprendimento automatico utilizzato nell'ambito dell'analisi dei dati e della classificazione. Questo modello rappresenta un'analisi dei dati sotto forma di una struttura ad albero, in cui ogni nodo rappresenta **un test** su un attributo dei dati, ciascun ramo rappresenta **l'outcome di una decisione** e ogni foglia rappresenta **una classe o una previsione**.

L'albero decisionale costruisce una **serie di domande** basate sugli attributi, questo processo continua ricorsivamente fino a quando viene soddisfatta una condizione di stop.

Gli alberi decisionali sono apprezzati per la loro **semplicità e interpretabilità**. Tuttavia, possono diventare complessi con dati ricchi di attributi.





Codice ed Implementazione

3. Metodologia ed Implementazione

Viene implementato un codice che utilizza il modulo scikit-learn per creare un albero decisionale a partire dai dati di espressione presenti nel database locale. Elenchiamo le principali componenti:

1. Connessione al Database

2. Query per estrazione dei Dati

3. Estrazione dei dati

4. Creazione e Addestramento del Modello

5. Valutazione del Modello

```
# Esecuzione della query e ottenimento dei dati
cursor.execute(query2)
data = cursor.fetchall()
column_names = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=column_names)
```

```
# Dividi i dati in set di addestramento e test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Crea e addestra il modello di albero decisionale
model = DecisionTreeClassifier(random_state = 2, criterion = "entropy",
min_samples_split = 100, min_samples_leaf = 60)
model.fit(X_train, y_train)

# Valuta il modello
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
```

Indice

4. Risultati e Conclusioni



- Introduzione e Contestualizzazione
- Revisione e Background
- Metodologia ed Implementazione
- Risultati e Conclusioni



Dati caricati

4. Risultati e Conclusioni

I dati caricati rappresentano una risorsa preziosa per una vasta gamma di analisi e ricerche nel campo della **genetica, dell'oncologia** e di altre discipline correlate.

- **Tipo di Dati;** TPM (Transcripts Per Million), FPKM (Fragments Per Kilobase per Million), ecc.
- **Origine dei Campioni;** dettagli sul paziente, etnia, genere ed il sito primario del tumore.
- **Dettagli delle Analisi;** codice univoco, nome, identificativo del caso, ecc.
- **Annotazioni di Geni e Proteine;** annotazione dettagliate sui geni e le proteine coinvolte
- **Concentrazioni;** concentrazioni misurate nelle diverse analisi.



Analisi di Esempio

4. Risultati e Conclusioni

Questa interrogazione su una vista ci da una panoramica di quanti casi su un determinato tumore abbiamo. Così da poter identificare quelli meno comuni e rendere le analisi più mirate

```
CREATE VIEW TumorFrequencies AS (
    SELECT primary_site.site, COUNT(*) AS frequency
    FROM public.case c JOIN primary_site ON c.site = primary_site.site_id
    GROUP BY site_id
)

SELECT DISTINCT tf.*
FROM TumorFrequencies tf
```

Biomarcatori ed Alberi

4. Risultati e Conclusioni

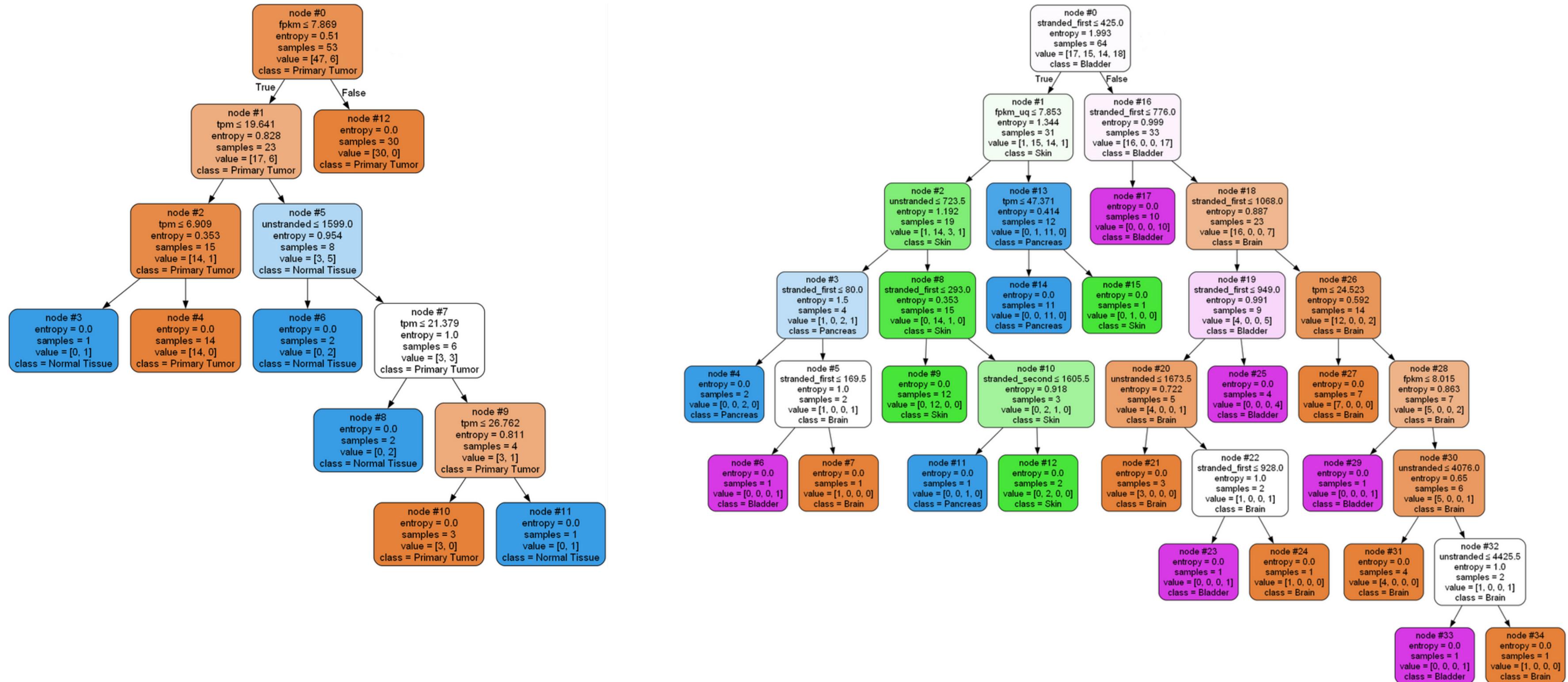


Gli alberi decisionali sono stati impiegati per identificare biomarcatori utili per la personalizzazione delle terapie. Questi, due esempi:

- **Classificazione dei Campioni Biologici;** Abbiamo utilizzato alberi decisionali per classificare i campioni biologici in base ai loro profili di espressione
- **Identificazione dei Fattori Chiave;** per identificare i geni, le proteine o le features di maggiore importanza nell'analisi dei dati.



Due alberi di esempio che mostrano, utilizzando come marcatore il gene oncogeno KRAS, come si possa distinguere rispettivamente: un tessuto **tumorale** da uno **sano** oppure dei **vari tipi di tumore**



Sviluppi Futuri

4. Risultati e Conclusioni



Una delle principali limitazioni del database è la completezza dei dati. Nonostante gli sforzi per acquisire un ampio spettro di dati di espressione genica e proteica, ci sono ancora **lacune nelle informazioni disponibili**. Quindi, potrebbe essere necessario **integrare i dati** del nostro database con altre risorse biomediche.



International
Cancer Genome
Consortium



HUMAN PROTEOME MAP

Conclusione

4. Risultati e Conclusioni



Il progetto ha portato alla creazione di un database **solido e ben strutturato** che ospita una **vasta quantità di dati di espressione**. Questi dati sono fondamentali per comprendere come i geni e le proteine sono regolati in risposta a **diversi stimoli, condizioni o malattie**.

Questo database è una risorsa preziosa e può essere utilizzato per una varietà di scopi, tra cui **l'identificazione di biomarcatori, studi di malattie e personalizzazione della terapia**.

Inoltre ho anche condotto analisi approfondite sui dati del database, dimostrando l'efficacia degli **strumenti e metodi**.

Identificazione ed Analisi di Biomarcatori dai Dati di Espressione Genica e Proteica

Corso di laurea in Informatica
Valerio Mesiti

Anno Accademico 2022-2023

Grazie per la Vostra Attenzione! Domande?



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'informazione, Informatica e Statistica
Dipartimento di Informatica

Relatore: Prof. Maurizio Mancini
Correlatore: Prof. Enrico Tronci